T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ 5G ÖTESİ AĞLARDA GÖRME DESTEKLİ IŞIN İZLEME PROBLEMİNE YUMUŞAK DİKKAT MEKANİZMASI KULLANARAK UZUN KISA SÜRELİ BELLEK UYGULANMASI

Nasir SİNANİ

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Doç.Dr. Ferkan YILMAZ

Temmuz, 2022

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ 5G ÖTESİ AĞLARDA GÖRME DESTEKLİ IŞIN İZLEME PROBLEMİNE YUMUŞAK DİKKAT MEKANİZMASI KULLANARAK UZUN KISA SÜRELİ BELLEK UYGULANMASI

Nasir SİNANİ tarafından hazırlanan tez çalışması 07.07.2022 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç.Dr. Ferkan YILMAZ
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç.Dr. Ferkan YILMAZ, Danışman
Yıldız Teknik Üniversitesi _____

Doç.Dr. Ahmet SERBES, Üye
Yıldız Teknik Üniversitesi _____

Prof. Dr. Oğuz KUCUR, Üye
Gebze Teknik Üniversitesi _____

Danışmanım Doç.Dr. Ferkan YILMAZ sorumluluğunda tarafımca hazırlanan Kablosuz 5G Ötesi Ağlarda Görme Destekli Işın İzleme Problemine Yumuşak Dikkat Mekanizması Kullanarak Uzun Kısa Süreli Bellek Uygulanması başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Nasir SİNANİ

İmza

Değerli aileme

TEŞEKKÜR

Babaları 1944 yılında Makedon güçlerince Blace Katliamında suçsuzca katledilen, yetim olarak büyüyen dedem Sinan ve babaannem İmriye'nin emeklerini hatırlatmak isterim, öğrettikleri hayat dersleri ve bilgi peşinde koşmak için kalbime yerleştirdikleri arzu için kendilerine minnettarım. Şu anda aramızda olmasalar da bu başarıdan dolayı mutlu ve gururlu olacaklarını biliyor ve tüm emek ve fedakarlıklar için çok teşekkür ediyorum.

Bu çalışma, danışmanım Profesör Ferkan Yılmaz olmadan mümkün olmayacaktı. Yüksek Lisansımı takip ederken, kendisi bana birçok belirsizlik ve zorlukla karşı karşıya kalan yeni bir araştırma alanına adım atmam için farklı fırsatlar verdi. Desteği ve cesaretlendirmesi, araştırmama devam etmem için muazzam bir yardım sağladı ve beni her zaman motive etti. Sayısız konuşmamız ve onun rehberliği, bilimdeki yerimi bulmama yardımcı olmuştur. Kendileri ile gelecekteki olası işbirlikleri benim için onur olacaktır.

Türkiye'nin en iyi üniversitelerinden biri olan Yıldız Teknik Üniversitesi'nde eğitim görme fırsatı sunduğu için Türkiye Bursları programına ayrıca teşekkür etmek istiyorum. Burslu olmak bir onurdu ve İstanbul'da okumak ve yaşamak büyük bir deneyimdi. Ailemle birlikte, tüm bunları mümkün kılan herkese her zaman minnettar kalacağız.

Technoperia'da çalışan ekibime, özellikle Ervin Domazet ve Affan Hasan'a bu tezin tamamlanmasındaki sürekli destekleri için teşekkür ederim. Ayrıca, tüm zorluklarda beni güçlü bir şekilde destekleyen arkadaşlarıma teşekkür ederim (eğer buradalarsa muhtemelen kim olduklarını bileceklerdir).

Ailem ve sevdiklerim, yüksek lisansım boyunca benim omurgam olmuşlardır. Beni yurt dışında ağırlamakta gösterdikleri sabır, moralimi yüksek tutan ve her konuda bana destek olan babam Şeval ve annem Mamure'ye en masum ve en özel teşekkürlerimi sunmaktayım. En umutsuz anlarımda, kardeşim Samir ve eşi Şefiye bana güç vermek ve daha ileri gitmem için beni cesaretlendirmek için her zaman hazırdılar. Ayrıca tatlı yeğenlerim Bora ve Arban'a da yanlarında olamadığım ve onlarla birlikte vakit geçiremediğim için çok üzgün olduğumu belirtmek isterim. Umarım bir gün bu eseri

okuma fırsatına sahip olup, gelecekte onlarla birlikte olmak yerine çalışmam için harcadığım zaman için beni affederler.

Son olarak, harika eşim Djellza'ya, özellikle hafta sonları dizüstü bilgisayarımın başında oturup daha iyi derin öğrenme modelleri bulmaya çalıştığım zamanlarda, yüksek lisansımı sürdürme konusundaki sabrı için çok teşekkür etmek istiyorum. Onun varlığı, tüm zorlukların üstesinden gelmem için beni daha da güçlendirdi.

Nasir SİNANİ

İÇİNDEKİLER

SİMGE LİSTESİ

$g(z)$	Activation Function
f_u	Beamforming Vector
$\mathscr{F}$	Codebook
$\mathbb{1}\{\bullet\}$	Indicator Function
$p(x)$	Probability Distribution
W	Weights

KISALTMA LİSTESİ

5G	The fifth generation of cellular networks
AI	Artifical Intelligence
CNN	Convolutional Neural Network
Convnet	Convolutional Network
CPU	Central Processing Unit
CS	Camera Surveillance
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
FC	Fully Connected
GHz	Gigahertz
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
ICS	Intelligent Camera Surveillance
IoT	Internet of Things
IS	Intelligent Surveillance
LiDAR	Light Detection and Ranging
LOS	Line of Sight
NLOS	Non-Line of Sight
LSTM	Long Short-Term Memory
MEC	Mobile Edge Computing
MHz	Megahertz
mmWave	Millimeter wave

MNIST	Modified National Institute of Standarts and Technology
NN	Neural Network
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
SNR	Signal to Noise Ratio

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

Kablosuz 5G Ötesi Ağlarda Görme Destekli Işın İzleme Problemine Yumuşak Dikkat Mekanizması Kullanarak Uzun Kısa Süreli Bellek Uygulanması

Nasir SİNANİ

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç.Dr. Ferkan YILMAZ

5G teknolojisinin büyümesi ve sağlık, otonom arabalar, görüntü tanıma ve diğer birçok alanda çeşitli bilgisayarlı görü görevleri için derin öğrenmenin sürekli başarısı, kablosuz iletişim alanında yeni zorluklar getirmiştir. Ayrıca, 5G-ağlar ötesi, temel olarak baz istasyonları ve mobil kullanıcılar arasındaki görüş hattı (LOS) bağlantılarının nasıl sürdürüleceğine dayanmaktadır. Bu nedenle, 5G-Ötesi ağlarındaki ana zorluklardan biri, en iyi performans için en iyi ışın biçimlerdirmesini arama gecikmesinden kaçınmak için, blokajlar mobil kullanıcıların iletişim kurmasını engellemeden önce mobil kullanıcılar için devir mekanizmasının proaktif olarak nasıl korunacağıdır. Buna göre, görme destekli milimetre dalga (mmWave) ışın ve blokaj tahmini, proaktif aktarım ve kaynak tahsisi için yeni araştırmalara kapı açmıştır. Bu tezin amacı, Vision-Wireless ViWi-BT veri kümesinde değerlendirilen derin öğrenme yaklaşımını kullanarak mmWave bantlarında kablosuz ışın izlemeyi incelemektir [1]. Temel tahmin yöntemi olarak uzun kısa süreli bellek (LSTM) ağı kullanarak önceden gözlemlenen ışın dizilerinden ve görüntülerden gelecek ışın dizilerinin nasıl tahmin edileceğini sunmaktayız. En önemli özellikleri akıllıca seçmek için yumuşak dikkat mekanizmasını kullanıyoruz ve bu nedenle gradyan kaybolma sorununu ortadan kaldırmak için softmax dikkat işlevini farklı periyodik dikkat işlevleriyle değiştirilmesini önermekteyiz.

Anahtar Kelimeler: Derin Öğrenme, bilgisayarlı görü, kablosuz iletişim, ışın tahmini, uzun kısa süreli bellek

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ABSTRACT

On the Vision-Beam Aided Tracking for Wireless 5G-Beyond Networks Using Long Short-Term Memory with Soft Attention Mechanism

Nasir SİNANİ

Department of Computer Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Ferkan YILMAZ

The growth of 5G technology and the continuous success of deep learning for various computer vision tasks in healthcare, self-driving cars, visual recognition, and many other areas, brought new challenges in the field of wireless communication. Moreover, 5G-Beyond networks primarily rely on how to maintain line-of-sight (LOS) links between base stations and mobile users. As such, one of the main challenges in 5G-Beyond networks is how to proactively maintain the hand-over mechanism for mobile users before blockages prevent mobile users from communicating, so as to avoid the latency of searching the best beamforming for the best performance. Accordingly, vision-aided millimeter-wave (mmWave) beam and blockage prediction has opened the door for new research for proactive hand-off and resource allocation. The purpose of this thesis is to study wireless beam tracking on mmWave bands using deep learning approach evaluated on the Vision-Wireless ViWi-BT dataset [1]. We present how to predict future beam sequences from previously observed beam sequences and images using a long short-term memory (LSTM) network as a base predictive method. As such, we utilize the soft attention mechanism to intelligently choose the most important features and thus we suggest replacing the softmax attention function with different periodic attention functions to eliminate the gradient vanishing problem.

Keywords: Deep Learning, computer vision, wireless communication, beam prediction, long short-term memory

YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

1
GİRİŞ

1.1 Literatür Taraması

5G sorunlarını çözmek için derin öğrenmeyi uygulamanın en umut verici zorluklarını ele almak için literatürde çeşitli anketler sunulmaktadır. [2] ve [3] yazarları, derin öğrenme ve makine öğrenimini kullanarak mmWave ağlarında en iyi ışın biçimlendirmeyi aramak için bir literatür araştırması sunar ve CNN, RNN, DRL ve LSTM gibi kapsamlı farklı derin öğrenme modellerinin uygulama listesini sağlar. Daha ayrıntılı olarak, [3] devir yönetimi için makine öğrenimi uygulamalarına odaklanılmaktadır.

Daha geniş bir kablosuz-görü uygulama yelpazesinde (makine öğrenimi görevleri) araştırmayı ilerletmek adına CS veya ICS'den gelen görsel verilerden yararlanmak için, [1]'de görsel destekli bir kablosuz ışın izleme (ViWi-BT) veri kümesi tanıtılmıştır. Bu, zengin dinamiklere sahip birden fazla baz istasyonunu ve mobil kullanıcıları veya nesneleri (arabalar, otobüsler ve yayalar) içermektedir. Ayrıca [1]'de sunulan ViWi-BT yarışmasında ana görev, önceden gözlemlenen ışınları ve RGB görüntülerini kullanarak bir kullanıcının gelecekteki mmWave ışınlarını tahmin etmektir. Görü destekli ışın izleme konusundaki araştırma ilgisi son yıllarda çok çekici hale geldi. Katılımcı yaklaşımlarının kamuya açık olmadığı, dokuz takımın katıldığı ViWi-BT yarışmasında elde edilen sonuçlar resmi web sitesinde bulunabilmektedir [4]. Kamuya açık olan ancak yarışma kapsamında olmayan yaklaşımlar şu şekilde özetlenebilir.

İlk olarak [1]'de GRU modeli kullanılarak ışın izleme ve tahmin için temel çözüm önerilmiştir. [1]'deki ana soru, görsel verilerin tahmin sonuçlarını iyileştirip iyileştiremeyeceğiydi. [5]'de Alrabeiah ve Alkhateeb, mmWave bağlantı blokajı tahmini ve ışın tahmini problemini araştırdı. Yeterince derin olan sinir ağı modellerinin kullanılmasının, bire yakın bir başarı olasılığıyla mmWave ışın ve blokajlarını tahmin edebileceğini kanıtladılar. [6]'daki yazarlar, yalnızca çevresel verilere dayanan görüntüleri yakalamak için bir kamera kullanarak 3D yeniden yapılandırma görüntülerini kullanan bir ışın seçim yöntemi önerdiler. ViWi-BT veri

kümesi tarafından sağlanan ışın oluşturma ve görüntüleri kullanarak dinamik bağlantı blokaj tahmini için CNN'ler ve RNN tabanlı tekrarlayan tahmin ağı kullanan yeni bir çerçevenin uygulanması Charan vd. tarafından [7] sağlanmıştır. Ek olarak, [8]'deki yazarlar, blokaj tahminlerini merkezi birime akıtarak iletişimin belirli bir kullanıcı için farklı bir kameraya iletilmesi gerekip gerekmediğini belirlemek için ve blokaj tahmin problemini çözmek için CNN ve GRU'nun iki bileşeninden oluşan yeni derin öğrenme mimarisini sundular. ViWi-BT sorgulama veri kümesini blokaj tahmini ve nesne algılama veri kümelerini sunan iki veri kümesine genişlettiler. Reus-Muns vd. [9] mobil kullanıcıların konum, hız ve çevreleyen sahne görüntüleri gibi uzamsal bilgilerinde kullanılmış ve herhangi bir zamanda tüm olası kullanıcı konumlarını içeren hareketli bölgeyi tahmin etmek ve doğru tahmin edilmeyen kullanıcı konumlarının hatasını azaltma yöntemi için kanal kovaryans matrisi adı verilen bir yöntem önermişler ve daha sonra derin öğrenme tabanlı gürültü gidermeyi tanıtmışlardır. [10]'da, Roy vd. görsel verilerin kombinasyonu ile GPS konum bilgisinden yararlanan derin öğrenme tabanlı füzyon çerçevesi ile ilgili bir anket sunmuştur. Ayrıca Salehi vd., [11]'te bir araç senaryosunda mmWave bağlantıları için ışın seçimini, LiDAR gibi sensörlerden, kamera görüntüleri ve GPS' ten toplanan verilerden yararlanarak sunmuştur. Temel katkı, bölgesel olarak veya mobil uç bilgi işlem merkezinde (MEC) yürütülebilen füzyon tabanlı derin öğrenmeyi kullanan ışın tahmin çerçevesini sunmak ve önerilen optimizasyon yöntemine dayalı olarak bağlantıyı kurmak adına en iyi çift ışını seçmek için mmWave bandında çalışan ışın süpürme işlemini sunmaktır. [12]'te, Tian vd. farklı ışın gömme, görüntü tanıma, görüntü özelliği gömme ve sıralı modellere dayalı ışın izleme yaklaşımı sunmuştur. Buradaki temel katkı, eğitim ve doğrulamadaki görüntülerin birbirini dışlayan yeniden formüle edilmiş ViWi veri kümesidir. Hu ve Han [13]'da, eğitim sırasında hesaplamayı kolaylaştırmak için görüntülerin boyutunu çift doğrusal ölçekleme yoluyla küçülterek iki temel öğrenme modeli ve kaynaşmış öğrenme modelinden oluşan ViWi ışın izleme problemini çözmek için yeni bir algoritma sunmuşlardır.

1.2 Tezin Amacı

Bu çalışmada, yumuşak dikkat mekanizmasını kullanarak ViWi-BT sorununu çözmenin bir yöntemini araştırmaktayız. Bildiğimiz kadarıyla, ışın izleme takip problemini çözmek için şimdiye kadar literatürde yumuşak dikkat mekanizması açık bir şekilde kullanılmamıştır. Ayrıca, makale aşağıdaki noktalarla da özetlenebilir:

- **Özellik çıkarma modülü**: Odak noktası, görsel verilerden ResNet-50 ve 3D Resnext-101 CNN mimarilerini kullanarak özellikleri çıkarmaktır.

ResNet-50'nin, özellik çıkarma görevlerinde büyük başarı göstermesi nedeniyle 2B özniteliklerin çıkarılması için kullanıldığı nitelendirilmiştir.

- **Genel bilgi yakalama modülü**: Amaç, çıkarılan özelliklerden yararlanmak ve yumuşak dikkat mekanizmasının önemli bir rol oynadığı özellik çıkarma modülünden çıkarılan en önemli özellikleri akıllıca seçmektir. Alternatif olarak, kaybolan gradyan problemini ortadan kaldırmak için softmax dikkat fonksiyonunu Taylor softmax, soft-margin softmax ve benzeri gibi alternatif dikkat fonksiyonlarıyla değiştirilmesi önerilmektedir (Bkz. 3.2.3).

- **Dizi üreteci modülü**: Dizi üreteci modülünün rolü, yumuşak dikkat mekanizması ve ışın indeksleri tarafından akıllıca seçilen birleştirilmiş özellikleri kullanarak gelecekteki ışın dizilerini tahmin etmektir.

- Modelimizin sonuçlarını, %10'luk bir doğruluk artışı gösterdiğimiz temel çözümle değerlendiriyoruz.

Bu çalışma şu şekilde organize edilmiştir. Bölüm 2 tezde kullanılan teorik kavramlar ve metodolojiler kısa bir şekilde açıklanacaktır. Bölüm 3.1'de, ViWi-BT zorluğu için gelecekteki mmWave ışınlarını tahmin etme problemi yeniden ele alınmakta ve sunulmaktadır. Sorunu sunarak, Bölüm 3.2'de sunulan zorluğa bir çözüm olarak kaldıraçlı yöntemin modüllerini tanıtıyoruz. Daha sonra, Bölüm 3.2.3'te, kullanılan yumuşak dikkat mekanizmasının ana hatlarını çiziyoruz ve softmax dikkatin yerini alacak alternatif dikkat fonksiyonları öneriyoruz. Ayrıca ek olarak, Bölüm 3.3'te, kaldıraçlı yöntemin eğitimi ve test edilmesiyle ilgili ayrıntılı bilgiler sunuyoruz. Bölüm 4'te sonuçları alıyoruz ve temel çözümle ilkel karşılaştırma sağlıyoruz. Son olarak, Bölüm 5'de sonuçlarımızı çıkartıp ve daha ileri araştırma önerilerini ele almaktayız.

1.3 Hipotez

Kablosuz veri trafiği, abone başına yılda %50'nin üzerinde bir oranda artmakta ve bu eğilimin, videonun sürekli kullanımı ve Nesnelerin İnterneti'nin (IoT) yükselişiyle önümüzdeki on yılda hızlanması beklenmektedir [14]. Mobil veri talebinin patlayıcı büyümesiyle, beşinci nesil (5G) mobil ağ, milimetre dalgasındaki (mmWave) muazzam miktardaki spektrumdan yararlanacaktır [15]. Bu nedenle, son araştırmalarda mmWave frekanslarının, kablosuz iletim için halihazırda derinleştirilmiş 700 MHz ila 28 GHz ve 38 GHz radyo spektrum frekanslarını artırmak için kullanılabileceği öne sürülmüştür [16]. mmWave frekans bantlarında mevcut çok büyük bant genişliğinin olması gerçeği bile ultra yüksek veri hızı iletişimini sağlıyormuş gibi görünse de mmWave sinyallerinin yayılmasının özellikler

ve karakteristikleri hem fiziksel hem de ağ katmanlarında yeni zorluklar ortaya çıkarır. En önemli zorluklardan biri, mobil kullanıcıların devri ve en iyi performans için en iyi ışın şekillendirmeyi arama gecikmesinin nasıl azaltılacağıdır. Diğeri, blokajlar mobil kullanıcıların iletişim kurmasını engellemeden önce mobil kullanıcıları proaktif olarak nasıl devredeceğidir, arama gecikmesinden kaçınmak için en iyi performans için en iyi ışın şekillendirmesidir. Son yıllarda, makine öğrenmesi ve derin öğrenmenin yapay zeka (AI) oluşumları, sadece kamu için değil, aynı zamanda çok yaygın araçlar haline gelen kameralı gözetleme (CS) uygulamaları ve teknolojileri, trafik izleme, üretim izleme, yanlış sorumluluk iddialarını azaltma, yapı ve varlıkların korunması vb. gibi özel güvenlik amaçları video ve görüntü işlemede önemli bir araştırma alanı haline geldi. Zaman içinde kameraların (ISC) akıllı gözetleme (IS) sayısı artmakta ve bu durum çeşitli alanlarda umut verici uygulamalar ortaya çıkarmaktadır. Derin öğrenme temelli yaklaşımların kullanımının hızla gelişmesiyle birlikte birçok araştırmacı, sahnelerdeki gizli öznitelikler ve video dizilerinde parçalı hareketli nesneler hakkında modellerin öğrenme özelliklerine dayalı yöntemler önermiştir. ICS'nin arkasındaki aynı düşünceye dayalı olarak, görüntüleri ve video akışlarını kullanan AI tekniklerinin yardımıyla, Alrabeiah vd. tarafından birkaç araştırma yönüne dikkat çekilmiştir nitekim bu, [1]'de 5G-ötesi ağlardaki ana kablosuz iletişim problemlerini çözmek ve görsel verileri kullanarak kablosuz ağların güvenilirliğini arttırmak olarak listelenebilir.

2
DERİN ÖĞRENMEYE GENEL BAKIŞ

Bu bölümde, tezde kullanılan teorik kavramlar ve metodolojiler açıklanacaktır. Tezin daha iyi anlaşılması için derin öğrenme terimleri tanıtılacak, ardından özellikle evrişimli ve tekrarlayan modeller için tanımlar ve metodolojiler ile devam edilecektir.

2.1 Makine Öğrenmesi

Şu anda, Makine Öğrenimi, etkisini gösterdiği ve en acil sorunları çözmek için büyük bir potansiyel gösterdiği dünyadaki en güçlü teknolojilerden biridir. Verilerin günlük hayatımızda büyümesiyle birlikte Makine Öğreniminin önemi artmaktadır. Son yıllarda, verilerin etkili miktarda artmasına tanık oluyoruz ve verileri analiz etmeden, büyük bir kısmı işe yaramaz kalacaktır. Temel olarak Makine Öğrenimi, verileri analiz eden ve verileri bilgiye dönüştüren bir araçtır.

Makine Öğrenimi ile ilgili ilk fikirler ve tartışmalar, L. Pitts ve W. McCulloch [17], insan düşünce süreçlerini taklit etmek için sinir ağlarının ilk matematiksel modellemesini "Matematiksel sinir ağı modellemesini" yayınlamasıyla 1943'e kadar sürmektedir. Daha sonra, 50'li yıllarda, makine bir insanı "makinenin de insan olduğuna" inandırabiliyorsa, makineye "akıllı" denilebilir fikriyle Turing'in önerdiği Turing Testinden bahsedebiliriz. Daha sonra, A. Samuel, bir IBM bilgisayarı için, dama oyunları oynama ve oyun her oynandığında gelişme gösterme özelliğine sahip bir program sunmuştur. 70'li yıllarda K. Fukushima'nın el yazısı karakter tanıma için kullanılan hiyerarşik çok katmanlı bir yapay sinir ağı türü olan nörobiliş üzerinde yaptığı yeni çalışması öne çıkan çalışmalar arasındadır. 90'lar, bugün bile araştırmaların temel aldığı en önemli katkılardan birini kaydeden Makine Öğrenimi dönemidir. 1995 yılında T. K. Ho, rastgele karar ormanı üzerine bir makale yayınladı. Bunu, destek vektör makineleri hakkında etkili bir makale yayınlayan V. Vapnik ve C. Cortes izlemiştir. 1996 yılında, IBM'in satranç oynayan bir bilgisayarı olan Deep Blue, dünyada ilk kez o dönemin satranç şampiyonu Garry Kasparov'u mağlup etti. El yazısı tanıma değerlendirme kriteri olarak benimsenen Y. LeCun liderliğindeki ekipten

MNSIT (Modifiye Ulusal Standartlar ve Teknoloji Enstitüsü) veri kümesi geliştirmeyle birlikte bir başka güçlü katkı daha geldi. Bu tezdeki kaldıraçlı model çoğunlukla 1997 yılında S. Hochreiter ve J. Schmidhuber tarafından yayınlanan LSTM (uzun kısa süreli bellek) mimarisine dayanmaktadır.

Makine öğreniminin son dönemi, 2006 yılında "derin" ağları eğitme fikriyle başlamıştır. Literatürde aşağıdaki önemli gelişme derin öğrenmenin başarısına büyük ölçüde katkıda bulunmuştur:

- **Donanım**: Donanım yetenekleri, CPU ve GPU kaynaklarında büyük bir büyüme kaydetti ve böylece daha derin sinir ağları yönetmemize izin vermiştir.
- **Dijital Veri**: Dijital sensörlerin ve veri depolama kapasitesinin büyümesiyle, araştırmacıların modelin sonuçlarını değerlendirmek ve araştırmak için gerekli veri kütlesini elde etmesi kolaylaşmiştir.
- **Yazılım Geliştirme**: TensorFlow, Keras, PyTorch, Numpy, Pandas vb. gibi yazılım ve kütüphanelerin evrimi, makine öğrenimi yaklaşımını kolaylaştırmış ve farklı makine öğrenimi algoritmalarının daha da geliştirilmesine yardımcı olmuştur.

Makine Öğreniminin yararlı bir tanımı, aşağıdakilerin belirtildiği T. Mitchell "Makine Öğrenimi" kitabında bulunabilir: *Her makine öğrenimi problemi, bir tür eğitim deneyimi (E) yoluyla, bir görevi (T) yerine getirirken bir performans ölçüsünü (P) iyileştirme problemi olarak kesin olarak tanımlanabilir*.

Görev *(T)*, olayın nasıl işlendiğini temsil eden tarafımızdan tanımlanan algoritmanın öğrenme hedefini temsil eder. Makine Öğrenimi, benzer olayların sınıflandırması için farklı bölümlere ayrıldığı sınıflandırma ve çıktının bir olay grubu yerine bir olay olduğu regresyon olarak sınıflandırılan çeşitli sorunları çözebilir. Bu tezde, öğrenme hedefi, önceki görsel olayları ve kablosuz olayları gözlemleyerek bir sonraki gelecekteki ışın dizisi olayını tahmin etmek olarak belirlenmiştir. Bu tezdeki *Görev T* 3.1 bölümünde tanımlanmıştır.

Performans *(P)*, eğitime dahil olmayan bağımsız verilerden elde edilen başarıya dayalı olarak algoritmanın yeterliliğini ölçen bir metriktir. Metrik görev T'den sonra seçilmelidir. Bu tezde ele alınan kayıp fonksiyonu 3.3 bölümünde sunulmuştur.

Deneyim *(E)*, makine öğrenimi modelinin nasıl eğitildiğini gösterir.

Modelleri eğitirken Makine Öğrenimi'nde uygulanabilecek birçok teknik vardır. Denetimli öğrenme, girdi verilerinin ve çıktı verilerinin bilindiği ve iyi yapılandırıldığı

bir yaklaşımdır. Denetimli öğrenmenin amacı, x girdi verisini çıktı verisi y ile eşlemektir. Diğer bir yaklaşım, durumlarda yalnızca girdi verilerinin sağlandığı denetimsiz öğrenmedir. Buradaki amaç, çıktı verisi y hakkında ön bilgi sahibi olmadan girdi verisi x içindeki en ilginç gizli modelleri bulmaktır. Makine Öğrenimi, giriş verisi x çıktı verisinden çok daha büyük olarak bulunduğu sorunları da çözebilir, burada model, çıktı y'nin nasıl üretileceğini öğrenmek için bunları bir araya getirmeye çalışır. Makine Öğreniminde bu, yarı denetimli öğrenme olarak bilinir. Takviye gibi diğer makine öğrenimi yaklaşımlarında modeller, modelin iyi davranışlarının ödüllendirildiği ve kötü davranışların cezalandırıldığı ödül güdümlü davranışlarda öğrenir.

Bu tezde deneyim E, girdi verilerinin ve çıktı verilerinin bilindiği ve iyi yapılandırılmıştır. Temel olarak, model bir olasılık dağılımı üretecektir $p(x)$ çıktı verilerini y tahmin etmek için x verilerinin eşleştirmesini öğrenecektir.

2.2 Yapay Sinir Ağları

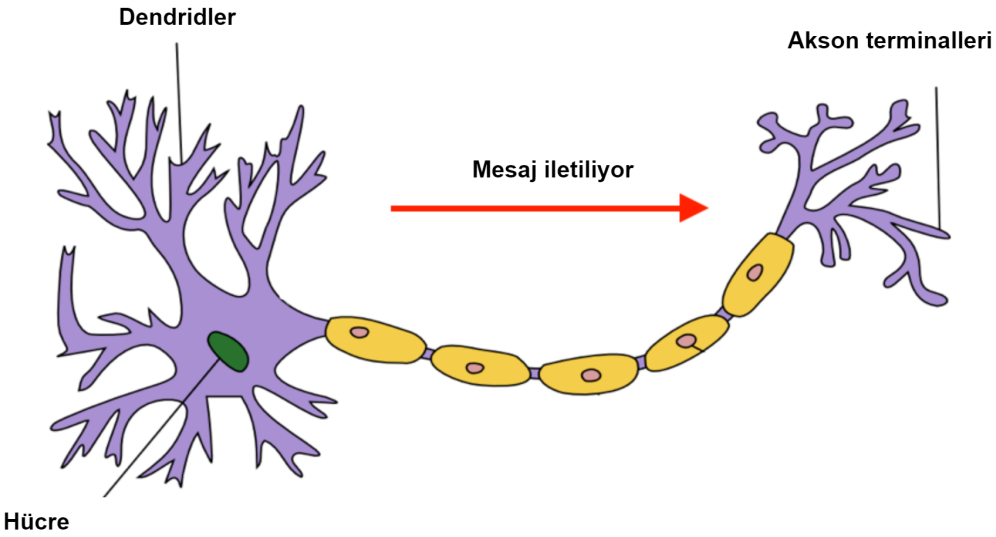
Yapay Sinir Ağları, en ünlü denetimli öğrenme algoritmalarıdır ve Derin Öğrenme sanatı ve bilimi onların üzerine inşa edilmiştir. Derin Öğrenme modelleri, aşağıdakiler gibi çok karmaşık görevleri çözmek için kullanılabilir:

- Derin Sinir Ağları (DNN'ler) çoğunlukla sınıflandırma görevleri için kullanılır
- Evrişimli Sinir Ağları (CNN'ler) bilgisayarlı görü görevlerini çözer
- Yinelemeli Sinir Ağları (RNN'ler) zaman serisi tabanlı görevleri çözer

Bu modeller, yüksek tahmin doğruluğu gösteren ve çoğu durumda insan performansına eşit olan beynin insan yapısını taklit etmeye çalışır. İnsan beyninin temel hesaplama birimi bir nörondur. Nöron, girdiyi dendritler aracılığıyla kabul ettiği, tüm sinyallerin toplandığı bir hücreye giren süreçleri ve daha sonra akson yoluyla sinyal ürettiği bir hesaplama birimi olarak işlev görür. Akson dallanır ve diğer nöronların dendritlerine sinapslar yoluyla farklı nöronlara bağlanır. Nöronun temel yapısı şekil 2.1'de verilmiştir.

Temel olarak, sinir ağlarının arkasındaki ana fikir:

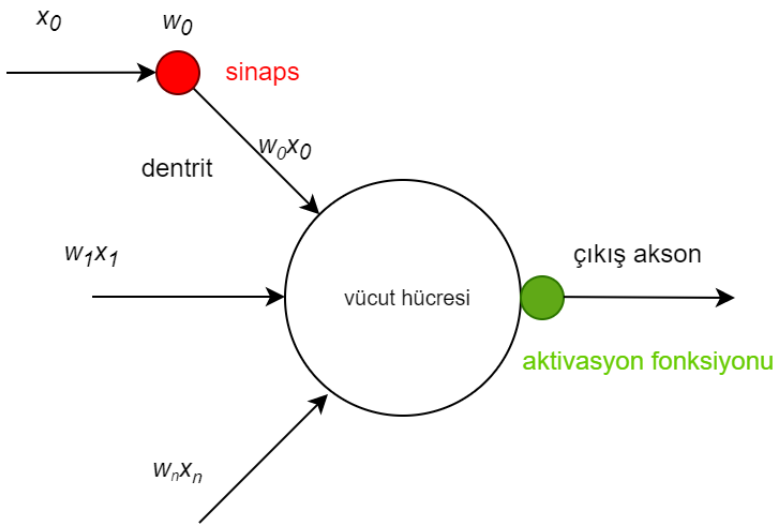
- Fonksiyon olarak görev yapan hücreyi temsil etmek
- Nöronları anlamlı bir şekilde birbirine bağlamak



Şekil 2.1 Bir nöronun temel yapısının temsili. Kaynak: Google Görseller

2.2.1 Perceptron (Algılayıcı)

Sinir ağlarının amacının insan beynini modellemek olmadığını, bu araştırmalarının çoğunlukla bir hesaplama modeli temsil eden matematik ve mühendislik disiplinlerine dayandığını belirtmek önemlidir. Sinir ağlarının daha önceki bir model temsili bir Perceptron'dur. Perceptron, ikili sınıflandırma görevlerini çözmek için bir giriş, bir çıkış ve aradaki en fazla bir gizli katmandan oluşan basit bir algoritmadır. Temel olarak, perceptron, girdinin belirli bir kategoriye ait olup olmadığını tahmin eder. Şekil 2.2'de görülebileceği gibi, Perceptron modeli nöron modeline çok yakındır.



Şekil 2.2 Bir perceptronun temel yapısının temsili. Kaynak: Google Görseller

Sinir ağlarının temel özelliği, algoritmalarının matematiksel problemleri, doğru çözüm

için sembolik bir ifade sağlayan bir formülü analitik olarak türetmek yerine, yinelemeli bir süreç yoluyla çözüm tahminlerini güncelleyen yöntemlerle çözmesidir [18]. Sinir ağlarının kesin çözümü bulması için çok boyutlu fonksiyon çeşitliliği nedeniyle fazla zaman harcayacaktır. Bu nedenle, sinir ağları algoritmasının amacı, en iyi çözümü veya kesin çözümü bulmak değil, danışmanlı deneme yanılma yöntemiyle sadece iyi bir çözüm bulmaktır. Nielsen, sinir ağlarını şu şekilde tanımlamaktadır: Sinir ağları hakkındaki en çarpıcı gerçeklerden biri, herhangi bir işlevi hesaplayabilmeleridir [19].

Bu, sinir ağlarının herhangi bir fonksiyonun yaklaşık değer tahmini için çok iyi algoritmalar olduğunu anlamamızı sağlar. Şekil 2.2'de gösterilen hücre gövdesi, özellik girdilerinin $(x_1, x_2, x_3...)$ bir vektörünü alan ve ağırlıklarla $(w_1, w_2, w_3...)$ çarpılan perceptronu temsil eder. Ağırlıklar, başlangıçta, sinir ağlarının her bir özellik için genel olanları öğrenmesi amacıyla rasgele ayarlandığı, öğrenilebilir parametrelerdir x_i. Daha sonra, x veri kümesinden bir örneğin tüm özellikleri çarpılır, toplanır ve bir aktivasyon fonksiyonuna aktarılır.

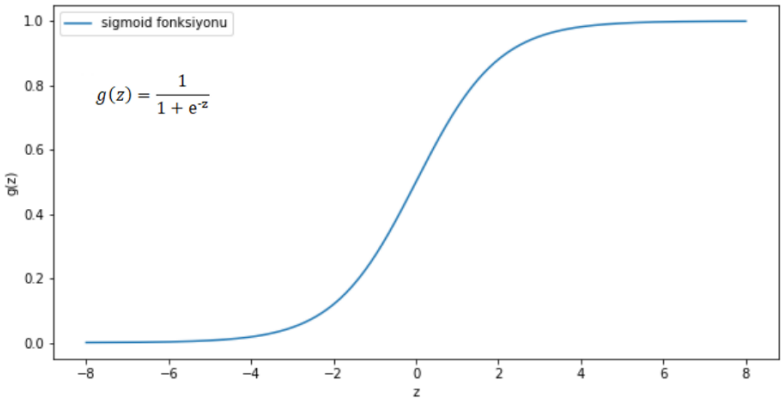
$$z = W^T x + b \tag{2.1}$$

$$g(z) = \begin{cases} 0 \;\; if \sum w_i x_i \leq \;\; threshold \;\; \theta \\ 1 \;\; if \sum w_i x_i > \;\; threshold \;\; \theta \end{cases} \tag{2.2}$$

Denklem 2.2, en basit aktivasyon fonksiyonunu temsil eder ve denklem 2.1'in sonucu seçilen sınırın üzerindeyse 1, değilse 0 verir. Modelin parametreleri yavaş öğrenmesini istediğimiz için bu etkinleştirme fonksiyonu sınırlıdır, bu nedenle ağdaki herhangi bir tek perceptronun ağırlıklarındaki veya yanlılığındaki küçük bir değişiklik, bazen bu algılayıcının çıktısının tamamen değişmesine neden olabilir, örneğin 0'dan 1'e. Bu çevirme daha sonra ağın geri kalanının davranışının karmaşık bir şekilde tamamen değişmesine neden olabilir [19].

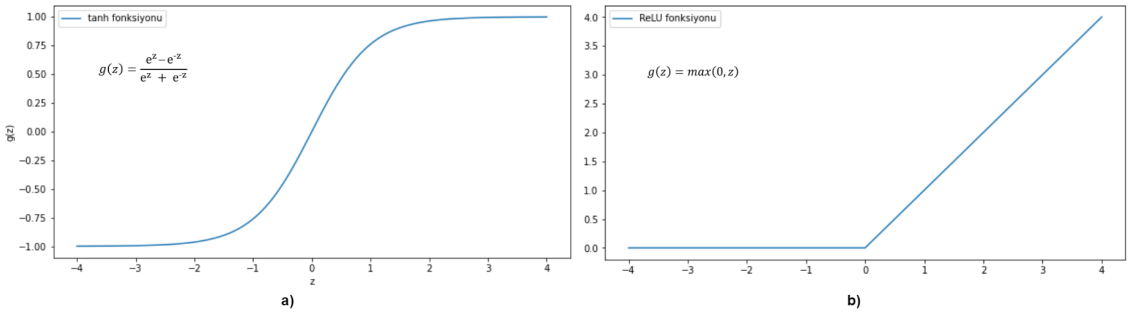
2.2.2 Aktivasyon Fonksiyonları

Perceptron aktivasyon fonksiyonu sorununu çözmek için doğrusal olmayan aktivasyon fonksiyonları sunulmuştur. Doğrusal olmayan aktivasyon fonksiyonları, modelin öğrenilebilir parametreleri yavaşça öğrenmesine yardımcı olur, böylece ağırlıklardaki küçük bir değişiklik, çıktı sonucunda da küçük bir değişikliğe neden olur. Literatürde en çok kullanılan aktivasyon fonksiyonlarından biri şekil 2.3'te sunulan Sigmoid aktivasyon fonksiyonudur.



Şekil 2.3 Sigmoid aktivasyon fonksiyonunun temsili

Sigmoid aktivasyon fonksiyonu çoğunlukla çıktının 1 ile 0 arasında olması beklenen ikili sınıflandırma problemlerinde kullanılır. Sık kullanılan diğer aktivasyon fonksiyonları hiperbolik tanjant ve ReLU aktivasyon fonksiyonlarıdır. Aktivasyon fonksiyonlarının grafiksel gösterimi şekil 2.4'te gösterilmiştir.



Şekil 2.4 a) Tanh aktivasyon fonksiyonunun ve b) ReLU aktivasyon fonksiyonunun gösterimi

Doğrusal olmayan aktivasyon fonksiyonunun en iyi seçimi hala tartışmalıdır ve aktif bir araştırma alanıdır.

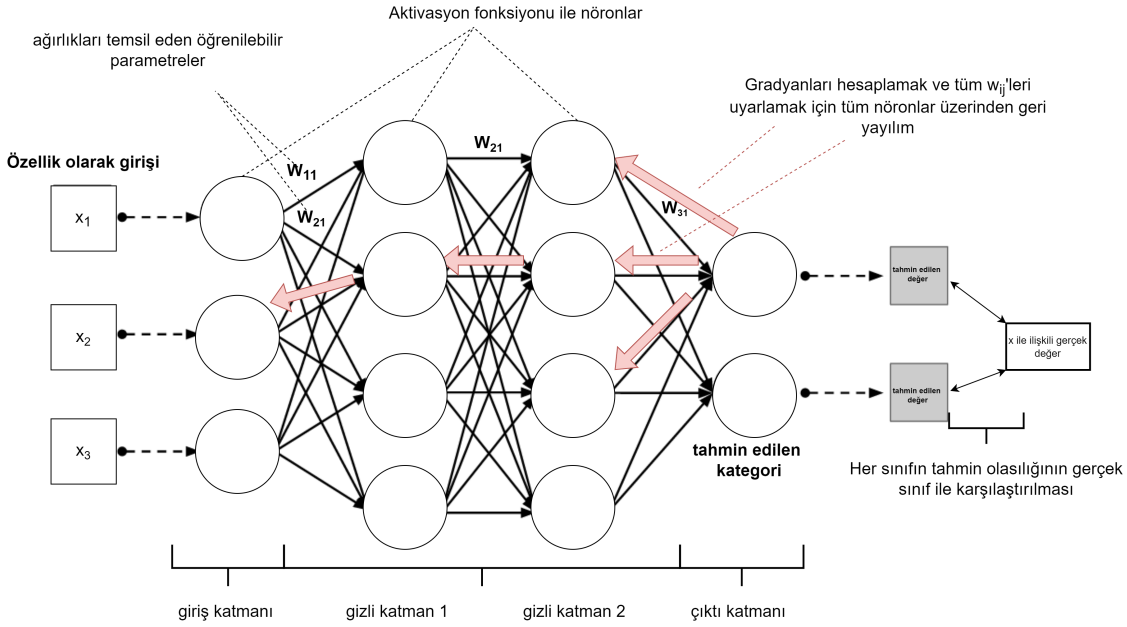
2.2.3 Derin Sinir Ağlarının Mimarisi

Bu bölüm sinir ağı mimarilerinin daha derin ve karmaşık bir yapıdayken verilerin ağ içindeki akışının nasıl gerçekleştiğiyle ilgili bir giriş sunacaktır. Derin Sinir Ağları (DNN) mimarisi üç unsurdan oluşmaktadır:

- **Giriş katmanı**: Bir veri örneğinin tüm özelliklerini $(x_1 ... x_i)$ içeren vektörü sunar
- **Gizli katman(lar)**: Verileri işleyen ve ileten bağlı nöronlar. Birden çok katman ve birden çok nöron içerebilir. Daha fazla katmana sahip, "daha derin" ağ

- **Çıktı katmanı**: Son katman, her zaman bir ağın geçişinin sonuçlarını içeren bir çıktı vektörü sunar

DNN'lerin öğrenmek için izledikleri adımlar, bir örnek verinin bir özelliğini temsil eden bir girdi vektörü elde etmeleri, girdi vektörünün başlangıç durumunda rastgele ayarlanmış ağırlıklarla çarpılması, sonucu bir aktivasyon fonksiyonuna beslemesi ve her nöronu diğer nöronlar ile birbirine bağlamasıdır. Daha sonra tahmin edilen değer ile beklenen değer karşılaştırılacak ve hataya göre geri yayılım kullanılarak ağırlıkların değerleri biraz değişecektir. Ağ içinde bilgi akışı süreci şekil 2.5'te sunulmaktadır.



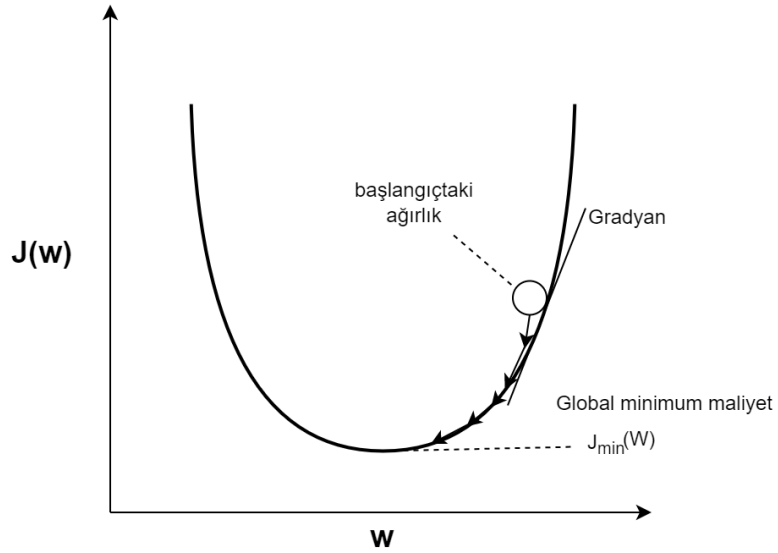
Şekil 2.5 Bilgi akışının ağ içinde nasıl sağlandığının temsili

2.2.4 Geri yayılım (Backpropagation)

Şekil 2.5'te sunulan ağ, ileri beslemeli ağ olarak da bilinir. Bu modellere ileri beslemeli ağlar denir, çünkü bilgi akışı, x'ten değerlendirilen fonksiyon ile, tanımlamak için kullanılan ara hesaplamalar aracılığıyla ve son olarak y çıktısına sağlanmaktadır [18].

Modelin eğitimi sırasında, ileri beslemeli yayılım, çıktı düğümleri için bir maliyet üretir. Genellikle geri yayılım algoritması, gradyanı hesaplamak için maliyetten gelen bilginin daha sonra ağ üzerinden geriye doğru akışına izin verir [18]. Amaç, her girdi vektörü için maliyet fonksiyonunun azalmasını ve son katmandaki çıktının istenen sonuca mümkün olduğunca yakın olmasını sağlayan bir ağırlıklar kümesi bulmaktır [20]. Bu, son x'e ulaşılana kadar grafikteki bazı x'e göre z'nin skaler gradyanını hesaplayarak elde edilir.

İleri beslemeli yayılım, geri yayılım, maliyet fonksiyonunun hesaplanması ve ağırlıkların güncellenmesi süreci, maliyet fonksiyonu değerlendirildiğinde her adımda yinelemeli bir süreci temsil eden dönem olarak bilinir. Şekil 2.7'de gösterildiği gibi, her bir nöronun ağırlıkları, minimumlar yönünde, maliyet fonksiyonundan aşağı doğru küçük adımlar atılarak güncellenecektir.



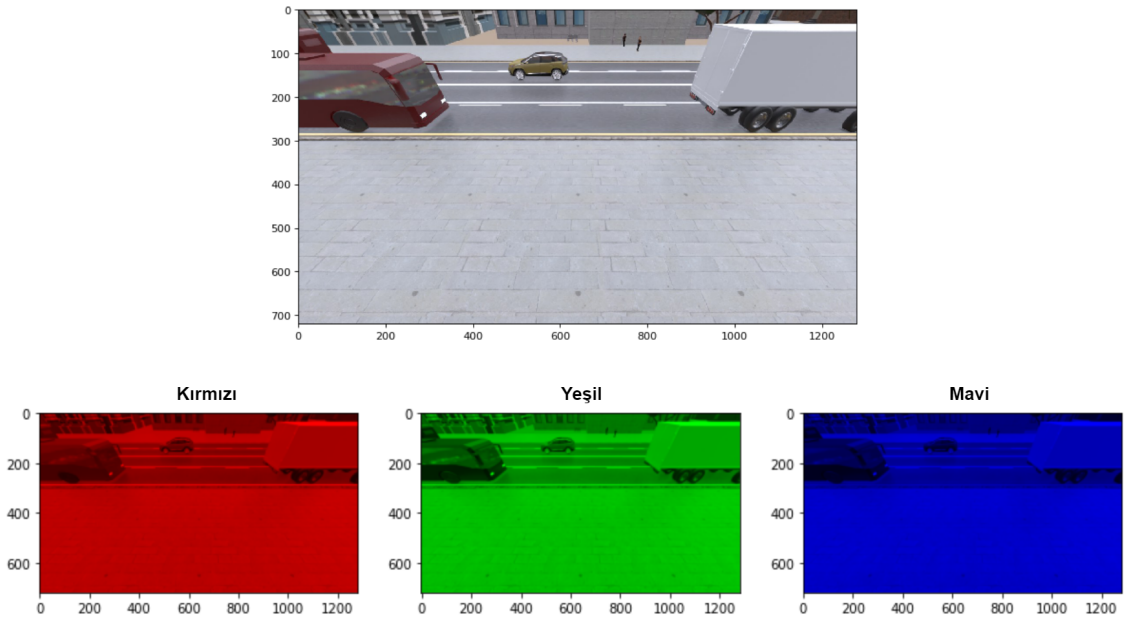
Şekil 2.6 Gradyan azalma temsili. $J(w)$ kaybı temsil eder ve w ağırlıkları temsil eder

Optimum çözüme ulaşmak için atılması düşünülen adım sayısı öğrenme oranı ile belirlenir. Yüksek değerlere sahip olmak minimuma ulaşmak zor olacağından öğrenme oranının düşük değerde olması önerilmektedir.

2.3 Evrişimli Sinir Ağları

Bir Evrişimli Sinir Ağı, bilgisayarlı görü görevlerini çözmek için kullanılan ve görüntüleri girdi olarak kabul edebilen bir Derin Öğrenme algoritmasıdır. CNN ağının bir görüntüdeki uzamsal ve zamansal bağımlılıkları yakalama yeteneği, CNN'nin bilgisayarlı görü ile ilgili çeşitli görevleri çözmede baskın olmasının bir nedenidir. İlk yayın LeCun vd. tarafından sağlanmıştır [21].

Eğitim için modele girdi olarak bir görüntünün teslim edilmemesinin önemi, günümüzde görüntülerin boyutlarının büyük boyutlara ulaşmasından kaynaklanmaktadır. RGB görüntüleri üç renge ayrılabilir: Kırmızı, Yeşil ve Mavi ve örneğin 8K (7680x4320) boyutlarında bir görüntüye sahip olmak, hesaplama açısından yoğun hale gelebilir. Bu nedenle, görüntü içindeki önemli bilgi ve özellikleri kaybetmeden, işlenmeden ve modele beslenmeden önce görüntülerin boyutunu küçültmek çok önemlidir.



Şekil 2.7 Örnek bir görüntü için RGB renk alanının temsili

Bu tezde, görüntülerden en önemli öznitelikleri çıkarmak için ResNet-50 ve 3D ResNext-101 CNN mimarilerini kullanmaktayız.

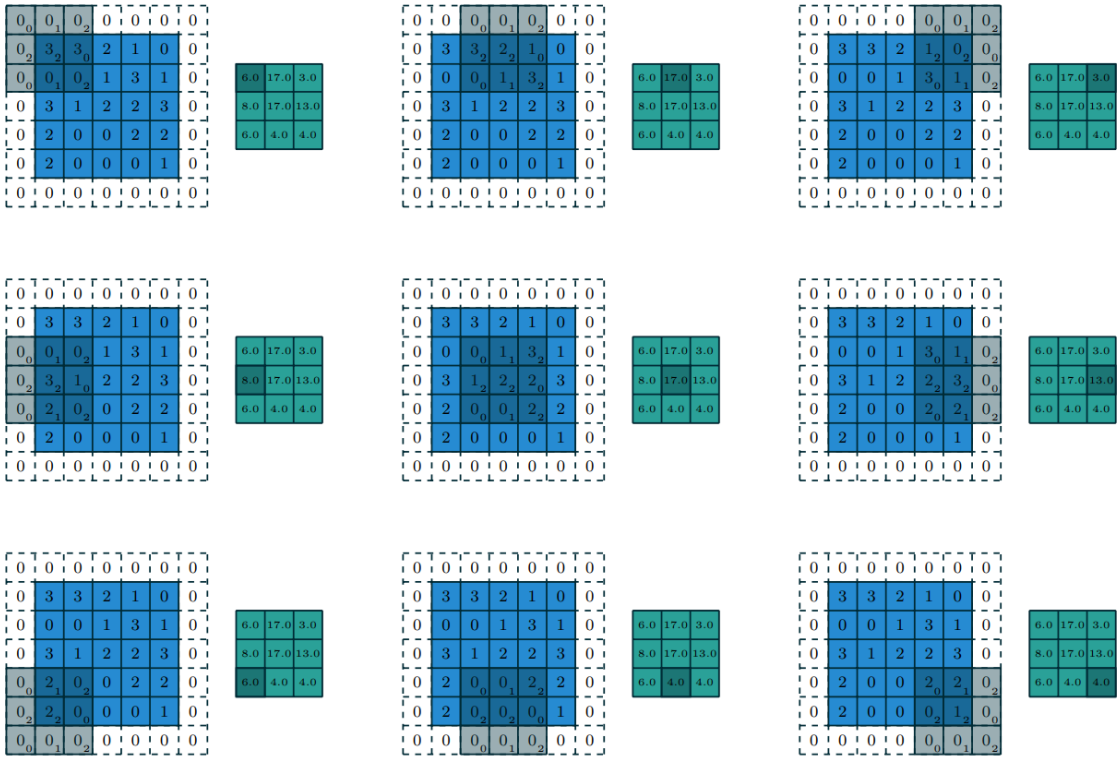
2.3.1 Evrişimsel Katman

Bir CNN ağı oluşturmak için temel blok bir evrişim katmanıdır. Evrişim katmanı, evrişim işlemiyle öznitelikleri çıkarmak için kullanılır. Bir evrişim işlemi, iki matris arasında bir nokta çarpımı gerçekleştiren doğrusal bir işlemdir ve denklem 2.3'te aşağıdaki biçimde gösterilmektedir:

$$y(i,j) = (x \star w)(i,j) = \sum_m \sum_n w(m,n)\, x(i-m, j-n) \tag{2.3}$$

İlk matris, genişlik i ve yükseklik j koordinatları ile 2B girdi görüntü verilerini sunar ve ikinci matris, çekirdek adı verilen bir dizi öğrenilebilir parametre sunar. Convnet'lerde, bir ileri beslemeli geçiş gerçekleştirirken, çekirdek girdi görüntü verisi boyunca kayar ve $y(i,j)$ görüntüsünün iki boyutlu bir temsilini üretir; burada yanıt çıktı özellik haritası olarak bilinir. Evrişim işleminin görselleştirilmesi şekil 2.8'de gösterilmiştir.

Bu örnekteki çekirdeğin değerleri denklem 2.4'te verilmiştir.



Şekil 2.8 $N = 2$, $i_1 = i_2 = 5$, $k_1 = k_2 = 3$, $s_1 = s_2 = 2$ ve $p_1 = p_2 = 1$ için evrişim işlemi [22].

$$w = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix} \tag{2.4}$$

Aşağıdaki özellikler gösterilmiştir:

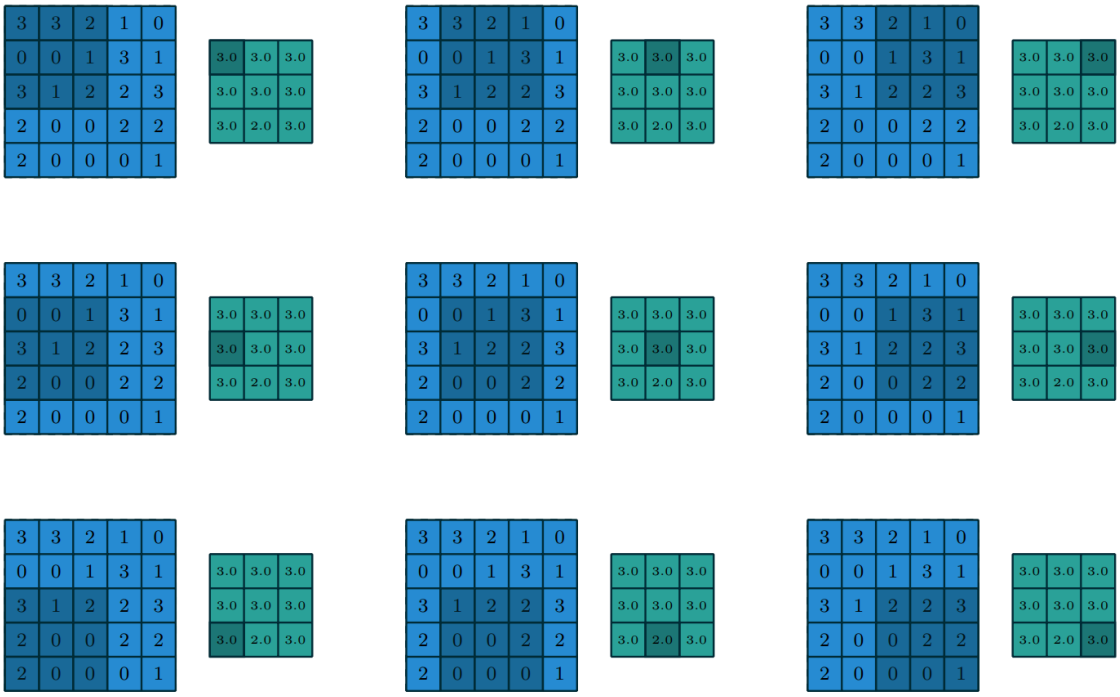
- i_j: j ekseni boyunca girdi boyutu,
- k_j: j ekseni boyunca çekirdek boyutu,
- s_j: j ekseni boyunca adım (çekirdeğin iki ardışık konumu arasındaki mesafe),
- p_j: j ekseni boyunca sıfır doldurma (eksenin başında ve sonunda birleşen sıfırların sayısı).

Görüntünün çıktı boyutu, adım ve dolgu değerine bağlıdır. Çıktı boyutunu hesaplamaya yaralan formül, görüntü $y(i, j)$ denklem 2.5'te verilmiştir.

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - f}{s} + 1 \right\rfloor \tag{2.5}$$

2.3.2 Havuzlama Katmanı (Pooling Layer)

Havuzlama katmanı, ConvNets için bir başka önemli yapı taşıdır. Genellikle, evrişim katmanıyla birlikte kullanılır ve özellik haritalarının çıktı boyutunu küçültmek için kullanılır. Havuzlama, giriş boyunca bir pencere kaydırarak ve pencerenin içeriği, ortalama veya maksimum değer alma gibi alt bölgeleri özetlemek için bir havuzlama fonksiyonuna çalışmaktadır [22]. Şekil 2.9, maksimum havuzlamaya ait bir örnek gösterilmiştir.



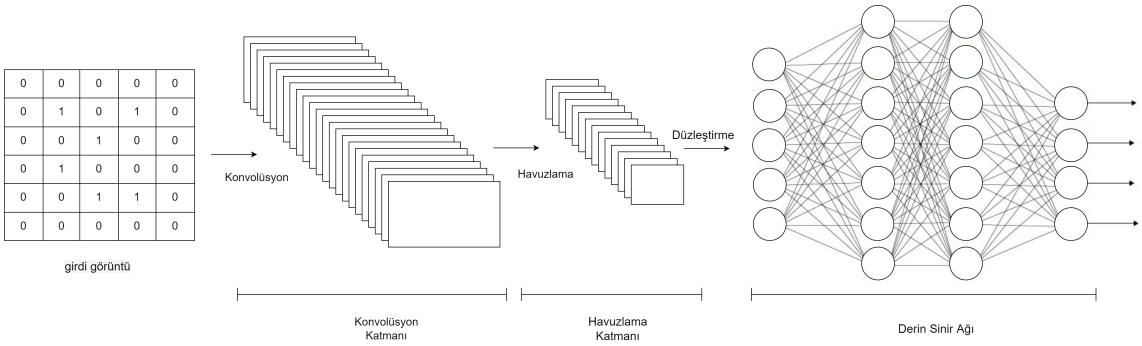
Şekil 2.9 1×1 adımlar kullanarak 5×5 girişte 3×3 maks havuzlama işleminin çıktı değerlerinin hesaplanması [22]

2.3.3 Tam bağlantılı katman (FC Katmanı)

Tam bağlı katman, Convnet'lerdeki son katmandır. Evrişimsel ve havuzlama katmanının çıktısı, girdi verilerinden çıkarılan öznitelikleri temsil eder. Tam bağlantılı katmanın amacı, görevleri sınıflandırmak için bu öznitelikleri kullanmaktır. Evrişimsel ve havuzlama katmanının çıktısı önce düzleştirilir ve daha sonra tam bağlantılı katmana girdi olarak verilir. Düzleştirme, evrişimli ve havuzlama katmanlarından 2 boyutlu dizileri tek bir vektöre dönüştürme işlemidir. Bu tek vektör daha sonra bir softmax etkinleştirme fonksiyonunu kullanan Çok Katmanlı Perceptronu temsil eden tam bağlantılı katmana beslenir.

CNN mimarisinin tüm yapı taşları şekil 2.10'da sunulmuştur.

Düzleştirilmiş çıktı FC katmanına beslenir ve bu nedenle eğitim sürecinin her



Şekil 2.10 CNN mimarisinin gösterimi

yinelemesinde geri yayılım kullanılır. Birkaç devirden sonra model, çekirdekler aracılığıyla daha önemli özellikleri öğrenmeye ve bunları softmax sınıflandırma tekniği ile sınıflandırmaya başlar.

2.4 Yinelemeli Sinir Ağları

Yinelemeli Sinir Ağları, dizileri modellemek için tasarlanmış farklı sinir ağları türleridir ve geçmiş bilgileri hatırlama ve buna göre yeni olayları işleme yeteneğine sahiptir [23]. DNN ve CNN mimarileri gibi, RNN'de nöronlar, maliyet fonksiyonları ve geri yayılım gibi benzer özelliklere sahiptir, ancak farkı, RNN'nin sıralı verileri bir girdi olarak kabul etmesidir. Tekrarlı Sinir Ağları ile ilgili ilk kaynak Rumelhart vd. tarafından yayımlanmıştır [20].

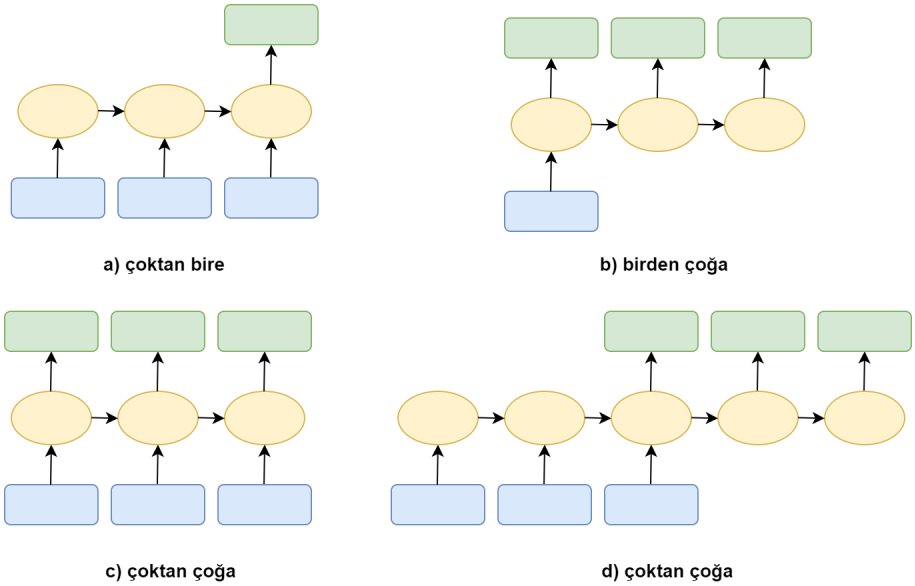
Bu verilere bir örnek, çıktı verilerinin sırayla veri olduğu konuşma tanıma, müzik oluşturma, duygu sınıflandırması ve makine çevirisi gibi metin verilerinde belirtilebilir. Bu tezde veriler görsel veriler ve kablosuz veriler olacaktır. Görsel veriler, ilgili kullanıcıyı belirten farklı zaman örneklerinde bir kablosuz ortamın sekiz görüntüsünü içerir ve kablosuz veriler ise görüntüye karşılık gelen ışın endekslerini içerir. Bu tür görüntüler ve ışın endeksleri, kullanıcıyı farklı konumlarda gösterir ve bunun iki büyük avantajı vardır. İlk olarak, bu dizi kullanıcının hareketini kaydetmeye izin verir. İkincisi, farklı zaman örneklerinde kullanıcı hareketini bilmeyi sağlayan bu gözlem, model için kullanıcı hakkında daha fazla bilgi sağlar. RNN, bir örneğin sadece bir takım özniteliklerine bakmakla kalmayıp aynı örneğin önceki durumuna ve mevcut durumuna da bakacak şekilde tasarlanmıştır.

Geleneksel sinir ağları modelinde, veri akışı bağımsız olarak ileri doğru gitmektedir, ancak RNN'de veriler çevrimler halinde akar. Bu çevrimler, bir değişkenin mevcut değerinin gelecekteki zaman adımlarında kendi değeri üzerindeki etkisini temsil eder [18].

RNN türleri, giriş veya çıkış verilerinin bir dizi veya kategorilerin aşağıda listelendiği tek bir vektör olması durumunda farklılık gösterir:

- **Çoktan bire**: Girdi verileri bir diziden oluşur ve çıktı verileri sabit boyutlu bir vektör veya skalerdir. Bu tür RNN, sınıflandırma problemlerini çözmek için kullanılabilir. Bir örnek, girdinin metin verisi ve çıktının bir sınıf etiketi olduğu duyarlılık analizi olabilir (örneğin, iyi veya kötü metin olsun, metin verilerinin amacını tahmin eder)
- **Birden çoğa**: Girdi verileri sabit boyutlu bir vektörden veya bir skalerden oluşur, ancak çıktı bir dizi biçimindedir. Birden çoğa bir örnek, müzik üretiminin problemlerini çözmek olabilir
- **Çoktan çoğa**: Hem giriş verileri hem de çıkış verileri bir dizi sunar. Bu tür RNN, girişin Türkçe dilinden veri olabileceği ve çıkışın çeviriyi İngilizce'ye sunduğu makine çevirileri için kullanılabilir. Bu tezde, girdinin bir ışın dizini dizisi olarak alındığı ve çıktının 1, 3 ve 5 gelecekteki ışın dizini dizisi olduğu durumlarda çoktan çoğa RNN tipi kullanılır.

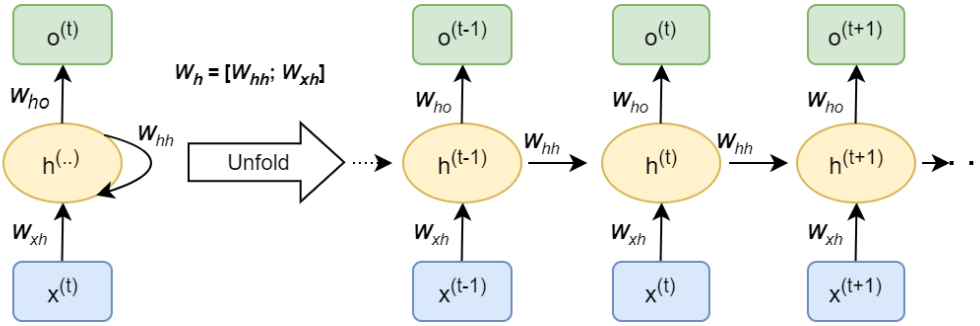
RNN türleri, her mavi dikdörtgenin bir girdi vektörü, okların bir fonksiyonu, sarı dairenin RNN'nin durumunu ve yeşil dikdörtgenin çıktı vektörünü belirttiği şekil 2.11'de gösterilmiştir.



Şekil 2.11 Girdi ve çıktı verilerine göre RNN türlerinin gösterimi [23]

2.4.1 Yinelemeli Sinir Ağlarının Mimarisi

RNN ağının mimarisi, ek olarak, RNN'nin tekrarlama içeren herhangi bir fonksiyona yaklaşabildiği geleneksel sinir ağlarına benzer. Mimari şekil 2.12 verilmiştir.



Şekil 2.12 RNN mimarisinin gösterimi [23]

Şekil 2.12'de $h_1^{(t)}$ gizli katmanı göstermektedir. $x^{(t)}$ girdi verilerini sunar. $h_1^{(t-1)}$ önceki gizli durumdur. W_{xh}, $x^{(t)}$ girdisi ile gizli katman, h arasındaki ağırlık matrisini sunar. W_{hh}, yinelenen kenarla ilişkili ağırlık matrisini sunar. W_{ho}, gizli katman ile çıktı katmanı arasındaki ağırlık matrisini sunar.

Geleneksel NN'nin aksine, şimdi RNN mimarisiyle, her gizli birimde, RNN'nin iki farklı girdi kümesi aldığını görebiliriz - girdi katmanından ön aktivasyon ve önceki zaman adımından aynı gizli katmanın aktivasyonu, $t-1$. İlk zaman adımında, gizli birimler sıfıra veya küçük rastgele değerlerle başlatılır [23].

İleri beslemeli işlem, geleneksel NN'ye çok benzer. Denklem 2.6, gizli katmanın hesaplanmasını, ağırlık matrislerinin karşılık gelen girdi vektörleriyle çarpımlarının toplamının bir hesaplamasını ve sapma birimini ekleyerek sunar.

$$z_h^{(t)} = W_{xh}x^{(t)} + W_{hh}h^{(t-1)} + b_n \tag{2.6}$$

Daha sonra, t zaman adımındaki gizli birimlerin aktivasyonları denklem 2.7'de gösterildiği gibi hesaplanır:

$$h^{(t)} = \phi_h\left(z_h^{(t)}\right) = \phi_h\left(W_{xh}x^{(t)} + W_{hh}h^{(t-1)} + b_n\right) \tag{2.7}$$

Daha sonra çıktı sonuçları denklem 2.8'de gösterildiği gibi hesaplanabilir.

$$o^{(t)} = \phi_o\left(W_{ho}h^{(t)} + b_o\right) \tag{2.8}$$

2.4.2 Geri Yayılım

RNN'deki geri yayılım işlemi, ağırlıkların azaltılmış veya ağırlıkların aynı değerleri ile birçok kez çarpılabileceğinden, her zaman damgasına geri dönmenin ve ağırlıkların güncellenmesinin zaman ve hesaplama gücü gerektirdiği bir zaman geri yayılımdır.

Bu sorun Hochreiter ve Shmidhuber tarafından şu şekilde anlatılmaktadır: *Zaman içinde geleneksel geri yayılım veya gerçek zamanlı yinelemeli öğrenme hata sinyalleri, zamanda geriye doğru akan (1) patlama veya (2) kaybolma eğilimindedir; geri yayılan hatanın zamansal gelişimi, ağırlıkların boyutuna üstel olarak bağlıdır. Durum 1, salınan ağırlıklara yol açabilir; 2. durumda, uzun zaman gecikmelerini kapatmayı öğrenmek çok fazla zaman alır veya hiç çalışmaz [24].*

Bu çalışmada, kaybolan gradyan problemini çözen Uzun Kısa Süreli Bellek ağı kullanıyoruz. Daha fazla bilgi bölüm 3.2.4'te verilmiştir.

2.5 Sinir Ağlarında Dikkat

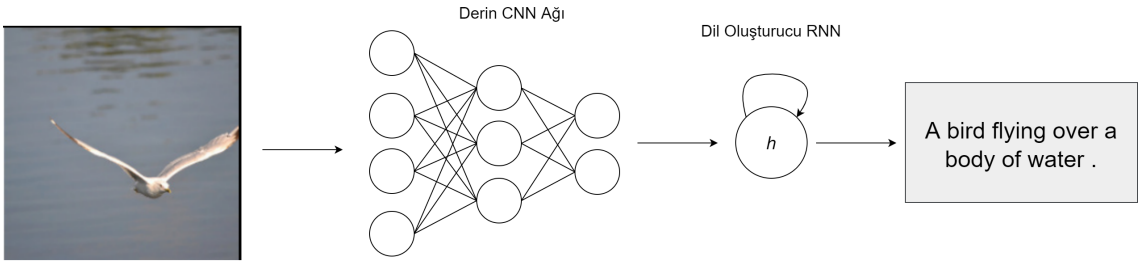
Dikkat mekanizması literatürde yaygın olarak çalışılmaktadır. Görsel dikkat sistemi, erken primatların [25] davranışlarından ve sinirsel mimarisinden esinlenmiştir ve iki ana ayrıntı üzerine inşa edilmiştir. Birincisi, bilgiyi işlemek için sınırlı kapasitedir. İnsan görüşüne dayalı olarak, herhangi bir zamanda, kontrol davranışında yalnızca bazı önemli bilgiler işlenebilir ve kullanılabilir. İkincisi ise seçiciliktir - istenmeyen bilgileri seçme veya filtreleme yeteneği. İnsan, dikkat edilen uyarıcıların bilincinde olabilir ve gözetimsiz uyarıcılardan büyük ölçüde habersiz olabilir [26].

Girdiden önemli özellikleri odaklama ve seçme konusunda benzer bir fikir, çeşitli bilgisayarla görme görevleri, konuşma tanıma, çeviri ve nesnelerin görsel olarak tanımlanması için Derin Öğrenme de uygulanmıştır. Literatürde farklı dikkat mekanizmaları sunulmaktadır ve bu çalışma yumuşak dikkat mekanizmasını kullanmayı göz önünde almaktadır [27].

2.5.1 Görüntü Altyazılama için Dikkat

Görüntü altyazısı oluşturma görevlerinde daha önceki çalışmalar, gizli durumu üreten önceden eğitilmiş CNN'yi kullanarak görüntüyü kodlar ve RNN, altyazının her bir kelimesini yinelemeli olarak üreten gizli durumun kodunu çözer. Bu yöntemi sunan makalelerden biri [28]'de belirtilmekte ve şekil 2.13'te gösterilmektedir.

Bu metodolojinin dezavantajı, RNN gizli durumu h tüm görüntü dikkate alınarak üretildiğinden, başlığın sonraki sözcüğü oluşturulurken alakasız bilgilerin

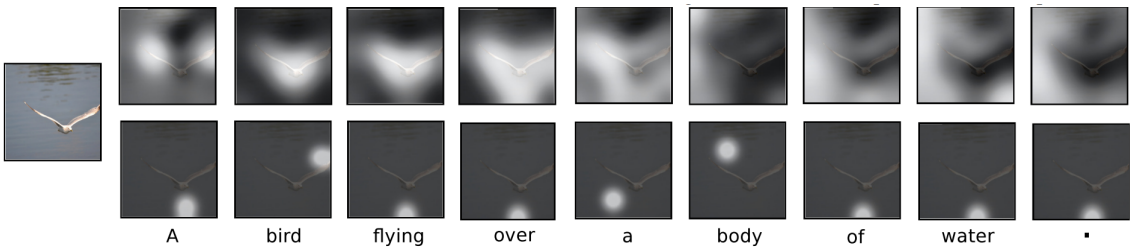


Şekil 2.13 [28]'ye dayalı klasik görüntü altyazılama modelinin temsili

çıkarılabilmesidir. Belirli kelimelerin görüntünün belirli bir alanı hakkında daha fazla bilgi verebileceği ve bu nedenle yumuşak dikkat mekanizmasının kullanılmasının somut bilgilere odaklanmaya yardımcı olabileceği bilinmektedir.

Yumuşak dikkat mekanizmasını kullanarak görüntü n parçaya bölünür ve her parçanın farklı temsil özellikleri h_1, h_2, ... h_n bir CNN modeli ile çıkarılır. Bu temsiller, bir başlık kelimesi oluştururken görüntünün ilgili kısımlarına odaklanacak ve böylece modelin önemli olan belirli alanlara dikkat etmesine izin verilecektir.

Şekil 2.14'te, yumuşak dikkat kullanıldığında bir resim yazısı sözcüğü oluşturmak için görüntünün hangi bölümünün kullanıldığı gösterilmektedir. Beyaz alan, resim yazısının sözcüğünü oluşturmak için görüntünün hangi bölümünün kullanıldığını gösterir.



Şekil 2.14 Üretilen her kelime için dikkatin görselleştirilmesi [27]

Beyaz alan, resim yazısının kelimesini oluşturmak için görüntünün hangi bölümünün kullanıldığını gösterir.

Bu çalışmada, çıkarılan en önemli öznitelikleri akıllıca seçmek için yumuşak dikkat kullanılmıştır. Bölüm 3.2.2, uygunluk puanlarının nasıl hesaplandığı hakkında daha fazla bilgi sunmaktadır.

3
MALZEMELER VE YÖNTEMLER

Görme destekli ışın izleme problemi ve zorluğu [1]'de kısaca açıklanmıştır. Takip eden bölümde, zorluk problemine kısaca tekrar değineceğiz ve daha sonraki bir bölümde, görsel verilerden yararlanarak görme destekli ışın takip probleminin yöntemlerini araştıracağız.

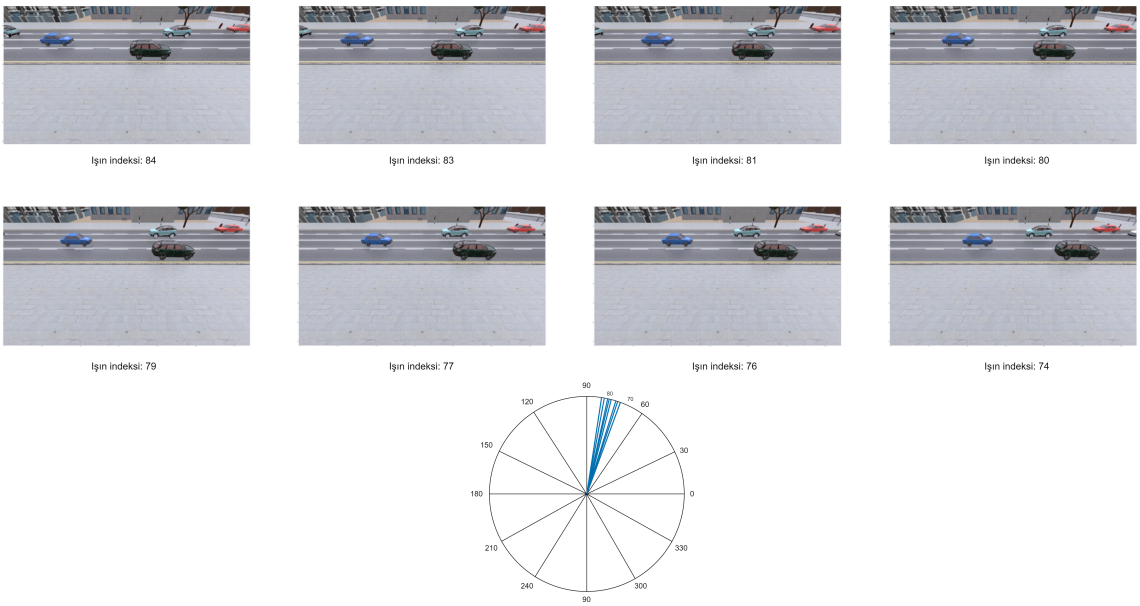
3.1 Problem Tanımı

ViWi-BT için temel zorluk, önceden gözlemlenen ışın dizileri ve RGB görüntülerine dayalı olarak gelecekteki ışın dizilerini proaktif olarak tahmin etmektir. ViWi-BT veri kümesinin yazarları, resmi web sitelerinde farklı senaryolar sundular, ancak ilgilendiğimiz senaryo, iki küçük hücreli mmWave baz istasyonu ile ilgili. İki baz istasyonunun, 28 GHz bandında çalışan bir mmWave tek tip doğrusal dizi anten ve her biri üç RGB kamera ile donatıldığı varsayılmaktadır [4]. Bu donanımlı kameraların, gözlemlenen sekiz çift görüntü ve karşılık gelen ışın indekslerini içeren ortamdaki nesnelerin hareketini kaydettiği dikkate alınmaktadır. Problem Denklem 3.1'deki gibi tanımlanabilir:

$$S_u(t) = [(f_u(t-\tau+1),\, X_u(t-\tau+1)), \ldots, (f_u(t),\, X_u(t))] \tag{3.1}$$

Burada f_u kod çizelgesindeki ışın biçimlendirme vektörüdür, $\mathscr{F}$ kullanıcıya μ zaman örneğinde m belirtmek için kullanılır. X_u ise, kamera tarafından yakalanan RGB görüntülerini temsil eden bir 3B tensördür. $S(t)$ ışın dizisi göz önüne alındığında amaç, $\hat{f}(t') = t+1\ldots,\ t+m$ ile gösterilen sonraki m zaman örneklerinde en iyi ışınları tahmin etmektir, burada zorluktaki m, üç aşamalı bir tahmin görevi 1, 3 ve 5'i tanımlar. Veri kümesindeki her kayıt, gözlemlenen bir diziden ve kullanıcıda alınan sinyal-gürültü oranını (SNR) maksimize etme anlamında karşılık gelen RGB görüntüsü ile beş temel gerçek gelecek ışın indeksinden oluşur.

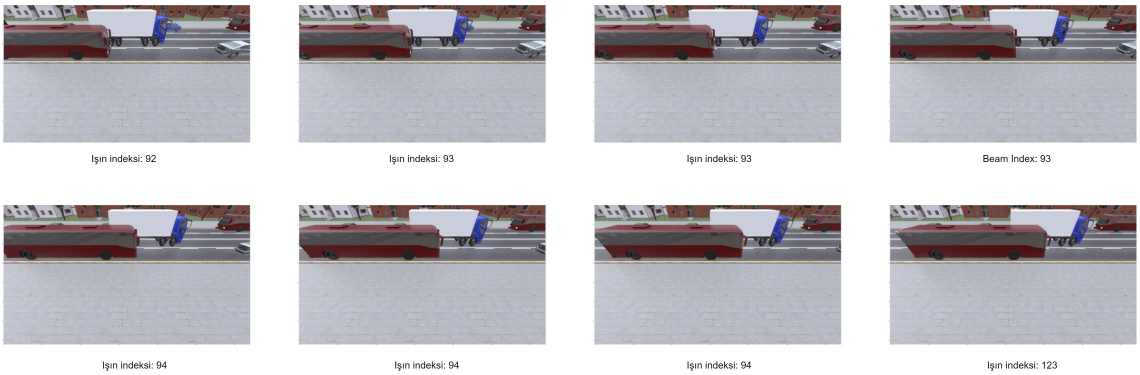
$\hat{f}(t')$ fonksyonunun tek bir kullanıcı için özelleştirilmesi beklenmez, ancak verilen kablosuz ortamdaki herhangi bir kullanıcının yerelleştirme bilgisi sağlanmadığından görüş hattı (LOS) veya görüş hattı olmayan (NLOS) olduğundan bağımsız olarak m kümesini tahmin edebilmelidir. Kullanıcının LOS'ta olduğu bir örnek şekil 3.1'de gösterilmektedir.



Şekil 3.1 Mobil kullanıcı LOS'tayken kablosuz ortam

Gösterildiği gibi, üç kamera ilgili kullanıcı ile iletişim için bağlantıyı korumaya çalışır. Her görüntüde, ışın indeksinin değeri ışın izleyicide mavi çizgi ile sunulur. Her görüntü, aynı boyutta 1280x720 piksele sahip bir RGB görüntü sunar ve tüm görüntüler, her nesne için tam konumlar sağlayan herhangi bir hareket bulanıklığı olmadan yakalanır.

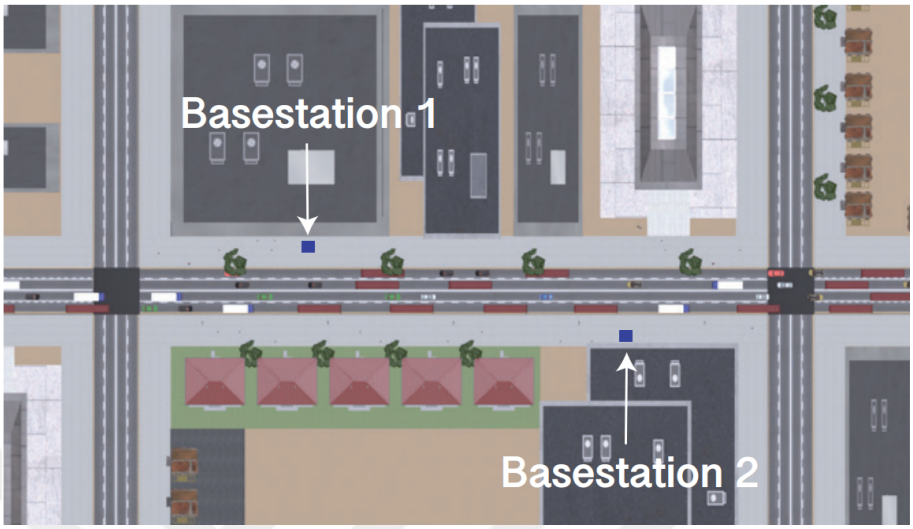
Şekil 3.2, ilgilenilen kullanıcının NLOS'ta olduğu ancak ışın indeksinin olası bir reflektöre göre seçildiği bir senaryo sunar.



Şekil 3.2 Mobil kullanıcı NLOS'tayken kablosuz ortam

ViWi-BT için 3D kablosuz ortam, ViWi-BT'de yürütülen senaryonun üstten

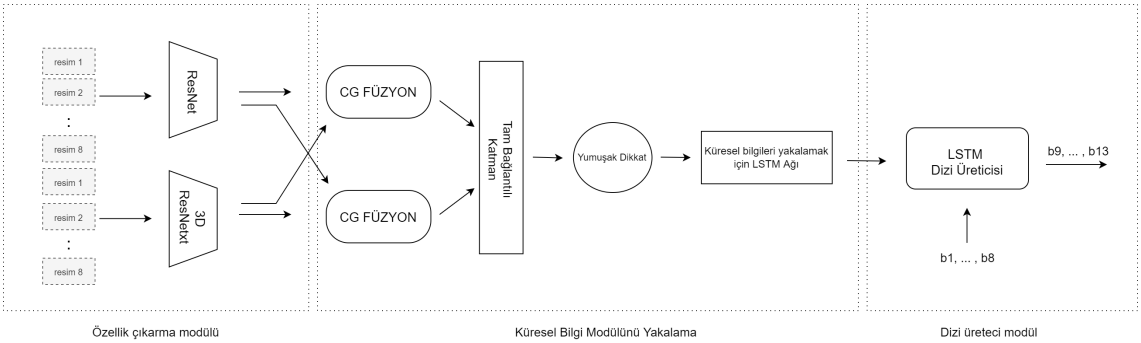
görünümünü gösteren şekil 3.3'te sunulmuştur. İki baz istasyonu arabalar, otobüsler, kamyonlar ve yayalar gibi dinamik nesnelerin yanı sıra [4], ağaçlar, çalılar, kaldırımlar, banklar ve binalar gibi statik nesnelerin farklı açılarından görüntü yakalayan üç kamerayla donatılmıştı.



Şekil 3.3 ViWi-BT'de yürütülen kablosuz ortam senaryosu [7]

3.2 Derin Öğrenme Çerçevesi

ResNet-50 [29] ve 3D ResNext-101'den [30] oluşan özellik çıkarma modülünden oluşan Şekil 3.4'de sunulan kaldıraçlı çerçeve, çapraz geçiş stratejisi [31] ve yumuşak dikkat mekanizmasından [32] ve dizi üreteci modüllerinden [33] oluşan ortak bilgi modülünü yakalar. Modüller aşağıdaki alt bölümlerde ayrı ayrı tanıtılacaktır.



Şekil 3.4 Kaldıraçlı derin öğrenme modelinin mimarisi [31]

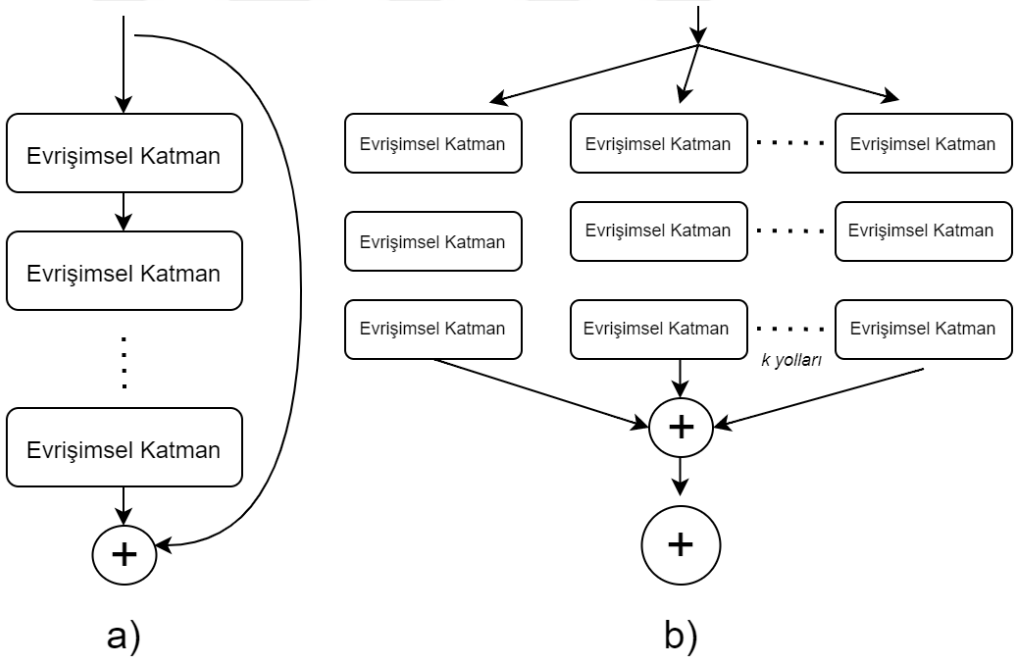
3.2.1 Özellik çıkarma modülü

Son zamanlarda, araştırma, çoklu görüntü özniteliklerinin çıkarılmasının ve çıkarılan özniteliklerin bir tahmine dayalı modele veya metin oluşturma modeli

ile güçlendirilmesi doğruluğun geliştirilmesinde önemli bir rol oynayabileceğini kanıtlamıştır [34], [35], [36].

Bunu dikkate alarak, çerçevede kablosuz veriyi modele iletmeden önce görüntülerin özniteliklerini çıkarmak için öznitelik çıkarma modülü kullanılır. Her görüntü, hareket eden kullanıcının bilgilerini ve çeşitli nesneler, binalar, ağaçlar, arabalar ve kamyonlar gibi çevre bilgilerini içerir. Zorluğun ortaya çıktığı veri kümesinde hangi kullanıcının ilgilendiği bilinmemekle birlikte, görüntünün öznitelik çıkarımının tahmine dayalı model için faydalı olduğu kanıtlanmıştır.

ResNet-50 ve 3D ResNext-101, çok sayıda katmana sahip olmaları nedeniyle güçlü özellik çıkarıcı modeller olarak gösterilmektedir. Bu modelleri sıfırdan eğitmek uzun zaman alır ve eğitim sürecinde çok fazla kaynak harcanır. Çerçevede, görüntülerden uzay-zamansal özellikleri çıkarmak için Kinetics veri kümesinde önceden eğitilmiş ResNet-50 ve 3D ResNext-101 modelini kullandık [37]. ResNet-50 ve 3D ResNext-101'in mimarisi Şekil 3.5'de verilmiştir.



Şekil 3.5 a) ResNet-50'nin mimarisini temsil eden bir blok [29]. b) Nicellik = k [30] olan bir 3D ResNeXt-101 bloğu

ResNet-50'nin görüntü sınıflandırmalarında en başarılı mimari olduğu kanıtlanmıştır [30]. ResNet-50, Şekil 3.5-a'da gösterildiği gibi bir sinyalin bir katmanı atlamasına ve sırayla sonraki katmana geçmesine izin veren kısayol bağlantıları sağlamaktadır. Bu bağlantılar ağlar üzerinden geçtiğinden, bu çok daha derin ağları eğitme yeteneği ile sonuçlanmaktadır.

3D ResNext-101, nicelliği tanıtmaktadır, nicellik ise dönüşüm kümesinin boyutunu

belirttiği ResNet-50'den miras alınır. Şekil 3.5-b'deki kardinalite k, özellik haritalarını küçük gruplara bölen evrişim gruplarını tanıtan daha derin ve daha geniş mimarilerden farklı boyutlar sunmaktadır. 3D ResNext-101 mimarisi, görüntülerden zaman-mekânsal 3D özelliklerin yakalanmasına yardımcı olmaktadır. Öznitelikleri çıkarmak için kullanılan ResNet-50 ve 3D ResNext-101 mimarisinin parametreleri 3.1'de gösterilmektedir.

Tablo 3.1 Görsel veri özelliklerini çıkarmak için kullanılan ağ mimarileri. Her evrişimli katmanı, toplu normalizasyon [38] ve ReLU [39] takip eder. F, özellik kanallarının sayısıdır ve N, her katmandaki blokların sayısıdır. conv1, girdilerin uzamsal olarak aşağı örneklenmesi işlemini temsil eder ve $conv2\_x$, $conv3\_x$, $conv4\_x$, $conv5\_x$ evrişim katmanını temsil eder

Model	conv1	*conv2_x*		*conv3_x*		*conv4_x*		*conv5_x*	
		F	*N*	*F*	*N*	*F*	*N*	*F*	*N*
ResNet-50	Conv, 7 x 7 x 7, 64, Temporal stride 1, Spatial stride 2	64	3	128	{4, 4, 4, 24}	256	{6, 23, 36, 35}	512	3
ResNeXt-101		128	3	256	24	512	36	{512,896}	{16,32}

3.2.2 Ortak bilgi yakalama modülü

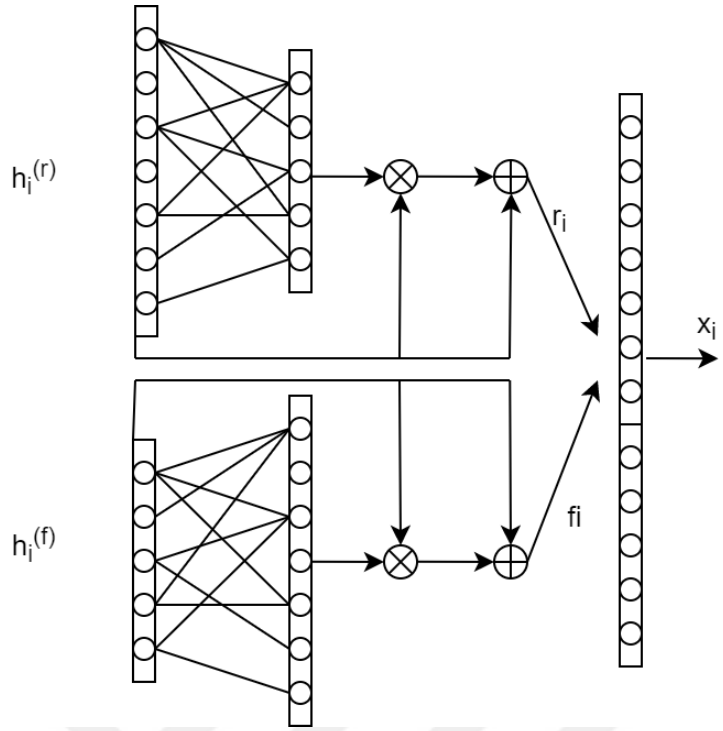
Ortak bilgi modülünü yakalama, ResNet-50 ve 3D ResNext-101 modellerinden çıkarılan öznitelikleri birleştirmek için kullanılır. Geçit füzyon ağı, [31]'da sağlandığı gibi iki aşamada gerçekleştirilir. İlk aşama, LSTM ağlarını kullanarak öznitelikleri toplamak iken, ikinci aşama, birleştirilmiş çerçevelerin ilişkilerini yakalamaktır, çünkü ilk aşamadaki çerçeveleri birleştirmek, ilişkileri yakalayamayacaktır. Çapraz geçit stratejisinin gösterimi Şekil 3.6'te verilmiştir.

Öznitelik temsillerini toplamak için kullanılan LSTM ağları, gizli durumları ve bellek hücrelerini temsil eden $h_i^{(r)}$ ve $h_i^{(f)}$ ile gösterilir. Çapraz geçitleme stratejisi, elde edilen ResNet-50 ve 3D ResNext-101 özniteliklerinin çarpma ve toplama işlemleriyle karşılık gelen anlamsal bilgi için verimli bir şekilde kullanılmasını sağlayabilir. r_i ve f_i, Resnet-50 ve 3D ResNext-101 çıkarılan öznitelikler için kapılı sonuçlardır.

$$r_i = \text{Gating}_r^{(E)}\left(h_i^{(f)},\ h_i^{(r)}\right) \tag{3.2}$$

$$f_i = \text{Gating}_r^{(E)}\left(h_i^{(f)},\ h_i^{(r)}\right) \tag{3.3}$$

burada Geçitleme fonksiyonu aşağıdaki gibi temsil edilir:



Şekil 3.6 Geçitli füzyon ağında önerilen çapraz geçit ağının çizimi. Çapraz geçit stratejisi, birbiriyle ilişkili bilgileri güçlendirir ve bunları birleştirir [31]

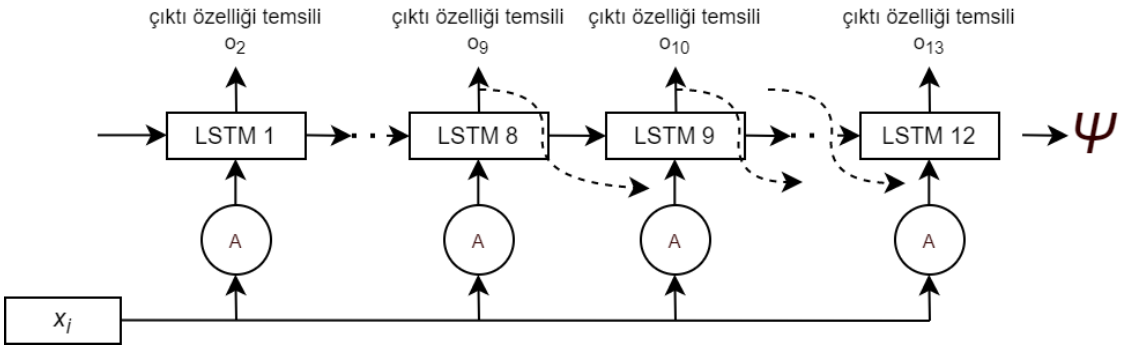
$$Gating\,(x,\ y) = \ \sigma\,(wx + b)\,y + y \tag{3.4}$$

burada y, x'nin öznitelikleri tarafından güncellenen hedef özelliği gösterir. w ve b öğrenilebilir parametreleri temsil eder ve σ, ReLU aktivasyon fonksiyonunu temsil eder. LSTM ağı ve çapraz geçiş stratejisi tarafından içerik bilgilerinin güçlendirilmesinden sonra, ResNet-50 ve 3D ResNext-101'in geçitli temsilleri, aşağıdaki gibi gösterilen tamamen bağlantılı bir katman tarafından birbirine kaynaştırılır:

$$x_i = w^{(E)}\left([r_i,\ f_i] + \ b^{(E)}\right) \tag{3.5}$$

x_i, CNN ağlarından birleştirilmiş özellikleri belirtirken, $w^{(E)}\ and\ b^{(E)}$ öğrenilebilir parametreleri temsil eder. Genel bilgileri yakalamak için LSTM ağı şekil 3.7'te verilmiştir.

Yakalanan özelliklerin sırası, $X = x_1,\ x_{2,} \ldots x_m$ kaynaşmış temsillere dayalı olacaktır. LSTM hücresi, genel bilgiyi yakalamak için tahmin edilen diziyi belirtir ve aşağıdaki gibi gösterilir:



Şekil 3.7 ResNet-50 ve 3D ResNext-101'den çıkarılan özniteliklerden ortak bilgileri yakalamak için LSTM ağı. *A* yumuşak dikkat mekanizmasını ifade eder

$$\begin{aligned} \{h_t,\ z_t\} &= LSTM\left(\left[E_{(o-1)},\ \phi_t(X,\ h_{t-1})\right],\ h_{t-1}\right) \\ Pr(c_o|c_{<t}, I; \theta) &= softmax\left(Wh_t\ +\ b\right) \end{aligned} \tag{3.6}$$

Burada h_t ve z_t sırasıyla gizli durumu ve hafıza hücresini gösterir. E, çıkarılan öznitelik temsil etiketleri için matrisi ifade eder ve $E_{(o-1)}$, çıkarılan özelliklerin gömme vektörünü belirtir. θ, W and b ağ tarafından öğrenilebilir parametreleri belirtir. $\Pr(c_o|c_{<t}, V; \theta)$, önceki etiketler $c_{<t} = c_1, c_2 \ldots c_{t-1}$ ve girdi görüntü dizisi I verilen doğru öznitelik belirtme etiketlerini c_t tahmin etme olasılığını gösterir.

3.6 denklemindeki ϕ_t sembolü, X üzerinde farklı ağırlıklara sahip bir vektör temsili veren [32]'ye dayalı yumuşak dikkat mekanizmasını belirtir:

$$\phi_t(X,\ h_{t-1}) = \sum_{i=1}^{m} a_{t,i} x_i \tag{3.7}$$

Burada $\sum_{i=1}^{m} a_{t,i} = 1$ ve $a_{t,i}$'ler her zaman adımında hesaplanır ve dikkat ağırlıklarını sunar. Dikkat ağırlıkları, geçerli adımda LSTM tarafından tahmin edilen en faydalı bilgileri seçmek için ResNet-50 ve 3D ResNext-101'den çıkarılan birleştirilmiş temsillerin alaka düzeyini yansıtır ve normalleştirilmemiş uygunluk puanını $e_{t,i}$ döndürür:

$$e_{t,i} = \mathrm{w}^T \tanh(Wh_{t-1} + Ux_i + b), \tag{3.8}$$

Burada w, W, U ve b öğrenilebilir parametrelerdir. Uygunluk puanları $e_{t,i}$ tüm görüntüler için $i = 1, \ldots, n$ hesaplandıktan sonra, softmax dikkati kullanılarak, $a_{t,i}$'leri

elde etmek için uygunluk puanlarını normalleştirmekteyiz:

$$a_{t,i} = \frac{\exp\left(e_{t,i}\right)}{\sum_{k=1}^{m} \exp\left(e_{t,i}\right)} \tag{3.9}$$

Dikkat mekanizması, karşılık gelen zamansal özelliğin dikkat ağırlıklarını artırarak LSTM ağının yalnızca önemli çerçeve alt kümesine odaklanmasına izin verir. Şekil 3.7'teki Ψ sembolü, son görünmeyen durumu gösterir ve ResNet-50 ve 3D ResNext-101'den çıkarılan özelliklerden ortak bilgiyi yakalaması beklenir ve daha sonra dizi üreteci LSTM ağını gelecekteki ışın indekslerini tahmin etmek için yönlendirmek için kullanılır.

3.2.3 Yumuşak dikkat mekanizması

Bu bölüm, görüntülerin çıkarılan özniteliklerinden ortak bilgiyi yakalamak için yumuşak dikkat mekanizmasının önemine ve Denklem 3.9'in yerini alacak farklı dikkat fonksiyonları önerilmesine ayrılmıştır. Çıkarılan özniteliklerin dizi üreteci modeline doğrudan beslenmemesinin nedeni, görüntünün tamamını eşit olarak ele almamasıdır. Dikkat mekanizması, görüntüyü girdi olarak kullanmak yerine, Denklem 3.7, Denklem 3.8 ve Denklem 3.9'de verildiği gibi girdi görüntüsü ile ilgili ilgi duyulan belirli alanlara veya nesnelere dikkat etmeyi düşünür.

Denklem 3.9'de sunulan softmax dikkat denklemi, dikkat mekanizmalarında kullanıldığında, ilgi puanlarını normal olarak dağıtmamaktadır, bu da eğitimi zorlaştıran kaybolan gradyan sorununa yol açmaktadır [40]. Bu nedenle, gradyan sorununu hafifletmek ve böylece özelliklere daha doğru bir önem vermek için softmax dikkatini değiştirmeyi düşünüyoruz. Bu değişim aşağıdaki gibi sıralanabilir:

- **Taylor softmax**: Taylor softmax fonksiyonu [41] tarafından önerilmiştir ve ikinci dereceden Taylor serisi yaklaşımını kullanır ve aşağıdaki gibi tanımlanabilir

$$a_{t,i} = \frac{1 + e_{t,i} + 0.5e_{t,i}^2}{\sum_{k=1}^{m}\left(1 + e_{t,k} + 0.5e_{t,k}^2\right)} \tag{3.10}$$

- **Soft-margin softmax**: Soft-margin (SM) softmax, logitlere bir mesafe marjı ekleyerek sınıf içi mesafeleri azaltır ancak sınıflar arası farklılığı geliştirir [41] ve aşağıdaki gibi tanımlanabilir

$$a_{t,i} = \frac{\exp\left(e_{t,i} - \beta\right)}{\sum_{k \neq i}^{m} \exp\left(e_{t,k}\right) + \exp\left(e_{t,k} - \beta\right)}, \tag{3.11}$$

burada β manuel olarak ayarlanır ve β sıfıra ayarlandığında, SM-Softmax orijinal softmax'ı basitleştirir.

- **Sin-Max-Constant/Cos-Max**: Bu dikkat fonksiyonu periyodik bir fonksiyondur ve aşağıdaki gibi tanımlanabilir [40]

$$a_{t,i} = \frac{1 + \sin\left(e_{t,i}\right)}{d + M + \sin\left(e_{t,i}\right)}, \quad \text{and} \quad M = \sum_{j \neq i}^{m} \sin\left(e_{t,i}\right). \tag{3.12}$$

- **Sin2-Max-Shifted**: Bu fonksiyon aşağıdaki gibi tanımlanır [40]

$$a_{t,i} = \frac{1 + \sin\left(e_{t,i}\right)}{d + M + \sin\left(e_{t,i}\right)}, \quad \text{and} \quad M = \sum_{j \neq i}^{m} \sin\left(e_{t,i}\right) \tag{3.13}$$

- **Sin-Softmax**: Bu dikkat fonksiyonu aşağıdaki gibi tanımlanır [40]

$$a_{t,i} = \frac{e^{\sin\left(e_{t,i}\right)}}{M + e^{\sin\left(e_{t,i}\right)}}, \quad \text{and} \quad M = \sum_{j \neq i}^{m} e^{\sin\left(e_{t,j}\right)}. \tag{3.14}$$

- **Siren-max**: Bu fonksiyon aşağıdaki şekilde tanımlanmıştır [40]

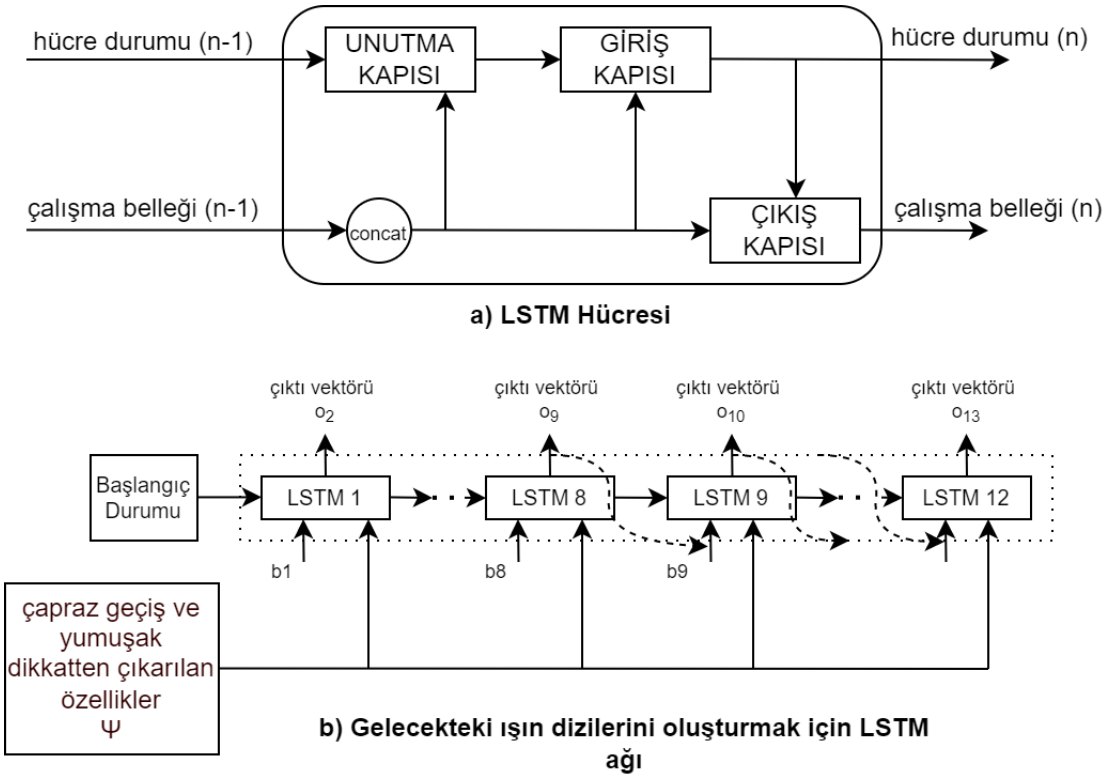
$$a_{t,i} = \frac{\frac{1+\sin\left(e_{t,i}\right)}{2-2\sin\left(e_{t,i}\right)}}{M + \frac{1+\sin\left(e_{t,i}\right)}{2-2\sin\left(e_{t,i}\right)}}, \quad \text{and} \quad M = \sum_{j \neq i}^{m} \frac{1 + \sin\left(e_{t,j}\right)}{2 - 2\sin\left(e_{t,j}\right)} \tag{3.15}$$

3.2.4 Dizi üreteci modülü

ViWi-BT zorluğunda, önceden gözlemlenen ışın dizileriyle 1, 3 ve 5 gelecekteki ışını tahmin etmek gerekir. LSTM ağı [24], tahmin ve metin oluşturma gibi zaman serisi verilerini içeren görevlerde büyük başarı göstermiştir ve bu nedenle bu yazıda tahmine dayalı model olarak kullanılması düşünülmüştür.

Şekil 3.8'te, tahmin görevi için bir LSTM hücresi ve LSTM ağı gösterilmiştir.

Bir LSTM hücresinin ana bileşenleri, unutma kapısı, giriş kapısı ve çıkış kapısıdır. Hücre durumu, LSTM hücresinin temel fikri ve rolü, düğümün tüm yinelemeleri altında bilgiyi sürdürerek uzun süreli bir bellek olarak hareket etmektir. Hücre düğümü, gereksiz bilgileri kaldırmaya veya bilginin önemli olup olmadığına karar



Şekil 3.8 a) LSTM hücresi ve b) gelecekteki ışın dizilerini oluşturmak için LSTM ağının temsili

vermeye ve böylece onu tutmaya karar verebilir. Unutma kapısının rolü, hücre durumundan hangi bilgilerin çıkarılacağına karar vermede hücre durumuna yardımcı olmaktır. Bu işlem, birleştirilmiş girdi değerleri üzerinde hesaplamalar yapılarak ve bu işlem hücre durumuna uygulanarak tamamlanır. Birleştirilmiş giriş değerlerine dayalı girdi kapısı, hangi bilgilerin önemli olduğuna ve hücre durumuna nelerin dahil edilmesi gerektiğine karar vermektedir. Çıkış kapısı sorumluluğu, mevcut hücre durumu ve birleştirilmiş çalışma girdi değeri üzerindeki hesaplamalara dayalı olarak bir sonraki gizli durumun ne olması gerektiğine karar vermektir.

Işın dizisini oluşturmak için LSTM ağı, on ikinci LSTM hücrelerinden oluşur. LSTM hücresinin girdileri, başlangıç durumu, ortak bilgi modülünü yakalamadan birleştirilmiş özellik vektörü, hücre durumu (mevcut durum), unutma kapısı, giriş kapısı, çıkış kapısı hangi bilgilerin olması gerektiğine karar vermeye çalıştığı çalışma belleği (gizli durum) daha iyi ileri tahminler sağlamak için tutulur veya kaldırılır. Şekil 3.8'te b, her LSTM hücresinin yeni uzun süreli ve kısa süreli bellek ürettiği her LSTM hücresine girdi olarak verilen $B = \{b_1,\ b_{2,} \ldots b_{12}\}$ ışın indeksini temsil eder ve son çıktılar $o_9 \ldots o_{13}$ ışın dizisinin tahmini olarak kabul edilir.

3.3 Eğitim ve Test Etme

Bu bölümde, derin öğrenme çerçevesini eğitim ve değerlendirme süreci tanıtılmıştır. Derin öğrenme çerçevesini eğitmek için öncelikle sekiz görselden faydalanıp girdisini sağlıyoruz. Bu görüntüler, modelin konum, hareket ve blokaj bilgileri gibi görsel ve hareket özelliklerini elde etmesine yardımcı olmak için önceden eğitilmiş ResNet-50 ve 3D ResNext-101'e girdi olarak verilir. Her örnek için 2048 boyutlu görsel ve 8192 boyutlu öznitelikler çıkarıyoruz. İkinci adım, yakalama genel bilgi modülünü kullanarak ResNet-50 ve 3D ResNet-101'den çıkarılan öznitelikleri birleştirmektir. Çıkarılan öznitelikler birleştirildikten sonra model 463 boyutlu bir vektör oluşturur. Üçüncü adım, her LSTM hücresine ortak bilgi modülünün yakalanma çıktısını girmektir. İlk on iki ışın indeksinin birleştirilmiş özellik vektörleri, her LSTM hücresinin gizli durumunu güncellemek ve çıktı vektörlerini oluşturmak için her LSTM hücresini ziyaret eder. Son adım, temel doğruluk değerlerini kullanarak eğitim kaybını hesaplamak için on iki çıktı vektörünü kullanarak ağı eğitmektir.

Önerilen çerçevenin test süreci, eğitim sürecine benzer. İkinci adım, ilk sekiz ışın indeksinin birleştirilmiş vektörlerinin, gizli durumları güncellemek için birinciden yedinci LSTM hücrelerine kadar ilk sekiz ışın indeksini ziyaret etme biçiminde farklılık gösterir. Sekizinci ve on ikinci LSTM hücreleri, gelecekteki ışın indekslerinin tahminini elde etmeye yardımcı olur. Eğitim sürecinin son adımı atlanır ve test süreci için kullanılmaz.

Eğitim süreci nedeniyle, Adam iyileştirici [42] kullanarak modeli uygun hâle getiriyoruz ve öğrenme oranı her sekiz devirde yarı yarıya azaltılarak $4\ x\ 10^{-4}$ olarak ayarlanmaktadır. Eğitim süreci 256 parçalı boyuttan oluşur ve kayıp fonksiyonu çapraz entropi (cross-entropy) kaybına ayarlıdır.

4
DENEYSEL SONUÇLAR

Bu bölümde, çerçevenin ViWi-BT veri kümesi üzerindeki performansını değerlendirmek için deney yapılmıştır. Görsel verilerin eğitim süreci üzerindeki etkisini araştırmayı ve gelecekteki ışın tahminleri için çerçeveyi temel çözümle karşılaştırmayı amaçlıyoruz. Deneyler, bir adet NVIDIA 1060 6GB GPU kullanılarak PyTorch ortamında çok sınırlı donanım kümesi altında tamamlanmıştır.

ViWi-BT veri kümesi, 281100 kullanıcı örneğinden $\mathscr{D}_t$ ile gösterilen eğitim kümesini, 120468 kullanıcı örneğinden $\mathscr{D}_v$ ile gösterilen doğrulama veri kümesini ve 10000 kullanıcı örneğinden $\mathscr{D}_{test}$ ile gösterilen test veri kümesini içerir. Her örnek, 8 çift ışın indeksini ve sokak görünümünün karşılık gelen görüntülerini içeren $S_u(t)$ ile temsil edilir. $S_u(t)$ içindeki kamera görüntüleri, hedef kullanıcının görünümünü ve arabalar, kamyonlar, binalar, ağaçlar vb. gibi farklı nesnelerin görünümünü içerir. Hedef kullanıcının yerinin belirlenmesi $\mathscr{D}_t$, $\mathscr{D}_v$, ve $\mathscr{D}_{test}$'de sağlanmaz, bu nedenle hedef kullanıcının görünümü her olası t örneğinde olabilir. Ayrıca, ilk sekiz ışın indeksi ve karşılık gelen kamera görüntüleri, hedef kullanıcı için gözlemlenen bilgilerdir ve son beş ışın indeksi, gelecekteki ışın indekslerini ve karşılık gelen kamera görüntülerini içeren en temel verilerdir. Yürütülen deneyde amaç, ilk sekiz ışın endeksini ve karşılık gelen kamera görüntüsünü girdi olarak sağlayarak, son beş ışın endeksiyle karşılaştırıldığında oluşturulan beş ışın endeksinin olasılığını en üst düzeye arttırmak için bir tahmin fonksiyonu $f_{\odot}(S_u(t))$ bulmaktır.

Bu bölümde ayrıca, ilk 1 doğruluk ve üstel bozulma puanı metriklerini kullanarak, doğrulama veri kümesindeki temel çözüm [1] ile çerçevemizi karşılaştırıyoruz. İlk 1 doğruluk [1]'de sunulduğu gibi Denklem 4.1'da tanımlanabilir.

$$\text{Acc}_{top-1}^{(n)} = \frac{1}{U} \sum_{i=1}^{U} \mathbb{1}\left\{p_i^{(m)} = t_i^{(m)}\right\} \tag{4.1}$$

Burada U doğrulama veri kümesindeki örneklerin sayısıdır, $\mathbb{1}\{\bullet\}$ yalnızca koşul

karşılandığında 1 değeriyle gösterge fonksiyonunu sunar, $t_i^{(m)}$ gelecekteki doğru ışınları içeren i^{th} hedef ışın dizisidir ve $p_i^{(m)}$ tahmin edilen gelecek m ışınlarının dizisidir.

Üstel bozulma puanı metriği [1]'de sunulduğu gibi Denklem 4.2'da tanımlanmıştır.

$$\text{score}^{(m)} = \frac{1}{U}\sum_{i=1}^{U} exp\left(-\frac{\left|\left|p_i^{(m)} - t_i^{(m)}\right|\right|_1}{m\sigma}\right) \tag{4.2}$$

Burada $||.||_1$ birinci normu sunar ve σ 0,5'e ayarlanmış bir cezalandırma faktörüdür. Temel çözüm [1] ile karşılaştırmamız Tablo 4.1'de gösterilmiştir.

Tablo 4.1 Çerçevemizin temel çözümle karşılaştırılması

	İlk 1 Doğruluk			**Üstel Bozulma Skoru**		
	1 gelecek ışın	*3 gelecek ışın*	*5 gelecek ışın*	*1 gelecek ışın*	*3 gelecek ışın*	*5 gelecek ışın*
Kaldıraçlı yöntemimiz	0.8989	0.7043	0.6205	0.8229	0.7492	0.6909
Temel çözüm [1]	0.85	0.60	0.50	0.86	0.68	0.60

İlk 1 doğruluktan elde edilen sonuçlar, yöntemimizin 1 özellik ışını, 3 özellik ışını ve 5 gelecek ışın tahmininde temel çözümden [1] daha iyi performans gösterdiğini göstermektedir. Önerilen modelin toplam puanı 0.10 arttırılmıştır ve bu nedenle görsel verilerden yararlanmanın çerçevenin daha iyi sonuçlar elde etmesine büyük ölçüde yardımcı olabileceğini söyleyebiliriz.

Üstel bozulma skorundan elde edilen sonuçlar, şaşırtıcı bir şekilde, bir sonraki 1 gelecek ışın için tahminde daha kötü sonuçlar verdi. Bu performans hatası, görsel verileri 1 gelecek ışını tahmin etmek için kullanırsak, görsel veriler modelin daha iyi sonuçlar elde etmesine gerçekten yardımcı olabilirse, bununla ilgili gelecekteki tartışmaları ve zorlukları açabilir. 3 öznitelik ışını ve 5 öznitelik ışını tahmininde üstel bozulmadan elde edilen sonuçlar, temel çözümden daha iyi sonuçlar verir. Öznitelik çıkarma modülünün gelecek zaman örneklerinde daha güçlü olmaya başlaması ve modelin görüntüden gelen ortamın arkasındaki öznitelikleri öğrenmesine yardımcı olması ve böylece daha doğru tahminler sağlaması nedeniyle bu sonuçlar beklenmektedir.

Genel olarak, ilk 1 doğruluğu ve üstel azalma puanını gözlemleyerek, hedef kullanıcının konumunun ve blokajının ortam özelliklerini ve ortam ayrıntılarının

özniteliklerini çıkarmak için görsel verilerden yararlanmanın, LSTM'nin iyi öngörücü ağı ile birlikte, ışın izleme ve tahmin görevleri için üstün performanslar göstermektedir.

5
SONUÇ VE ÖNERİLER

Bu çalışmada, farklı öznitelik çıkarma teknikleri ve tahmine dayalı model olarak LSTM ağını kullanarak önceden gözlemlenen ışın indeksleri ve görüntüleri ile mmWave ışın tahminlerini gerçekleştirmek için derin öğrenmeye ek olarak yumuşak dikkat mekanizması dahil edilmiştir.

ViWi-BT veri kümesi üzerinde yürütülen deneysel sonuçlar, ışın tahmin görevleri için hem kablosuz hem de görsel verilerden yararlanmanın, yalnızca kablosuz verileri kullanmaktan daha iyi tahmin doğrulukları elde etmeye yardımcı olabileceğini göstermektedir.

Mevcut çalışmamızın birkaç olası uzantısı bulunmaktadır. İlk olarak, veri önişlemede veri temizleme ve önyükleme gibi ekstra kavramların tanıtılması, çerçevenin daha iyi sonuçlar elde etmesine yardımcı olabilir. İkincisi, görsel verileri bağlantıları olan LOS veya NLOS benzeri ortamlarına göre kümelemek, daha iyi eğitim stratejileri yürütmemize yardımcı olabilir. Üçüncüsü, farklı öznitelik çıkarım modellerini göz önünde bulundurmanın, çerçevenin kullanıcı yörüngeleri ve çevre ile ilgili daha önemli özniteliklerin çıkarılmasına yardımcı olabileceğini gözlemliyoruz. Sonuç olarak, kapılı füzyon ve dizi üreteci modülü ile değiştirilecek kodlayıcı-kod çözücü modelleri gibi daha iyi ağ mimarilerinin göz önüne alınması daha iyi sonuçlar verebilir.

KAYNAKÇA

[1] M. Alrabeiah, J. Booth, A. Hredzak, A. Alkhateeb, "Viwi vision-aided mmwave beam tracking: Dataset, task, and baseline solutions," *arXiv preprint arXiv:2002.02445*, 2020.

[2] A. Ly, Y.-D. Yao, "A review of deep learning in 5g research: Channel coding, massive mimo, multiple access, resource allocation, and network security," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 396–408, 2021.

[3] M. S. Mollel *et al.*, "A survey of machine learning applications to handover management in 5g and beyond," *IEEE Access*, vol. 9, pp. 45 770–45 802, 2021.

[4] M. Alrabeiah, A. Hredzak, Z. Liu, A. Alkhateeb, "Viwi: A deep learning dataset framework for vision-aided wireless communications," in *submitted to IEEE Vehicular Technology Conference*, Nov. 2019.

[5] M. Alrabeiah, A. Alkhateeb, "Deep learning for mmwave beam and blockage prediction using sub-6 ghz channels," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5504–5518, 2020.

[6] W. Xu, F. Gao, S. Jin, A. Alkhateeb, "3d scene-based beam selection for mmwave communications," *IEEE Wireless Communications Letters*, vol. 9, no. 11, pp. 1850–1854, 2020.

[7] G. Charan, M. Alrabeiah, A. Alkhateeb, "Vision-aided dynamic blockage prediction for 6g wireless communication networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2021, pp. 1–6.

[8] ——, "Vision-aided 6g wireless communications: Blockage prediction and proactive handoff," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 193–10 208, 2021.

[9] G. Reus-Muns *et al.*, "Deep learning on visual and location data for v2i mmwave beamforming," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, IEEE, 2021, pp. 559–566.

[10] D. Roy *et al.*, "Going beyond rf: How ai-enabled multimodal beamforming will shape the nextg standard," *arXiv preprint arXiv:2203.16706*, 2022.

[11] B. Salehi *et al.*, "Deep learning on multimodal sensor data at the wireless edge for vehicular network," *arXiv preprint arXiv:2201.04712*, 2022.

[12] Y. Tian, C. Wang, "Vision-aided beam tracking: Explore the proper use of camera images with deep learning," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, IEEE, 2021, pp. 01–05.

[13] Z. Hu, C. Han, "Image and index fused sequence-to-sequence algorithm for vision-aided millimeter-wave beam tracking," in *Proceedings of the 5th ACM Workshop on Millimeter-Wave and Terahertz Networks and Sensing Systems*, 2021, pp. 7–12.

[14] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, J. Zhang, "Overview of millimeter wave communications for fifth-generation (5g) wireless networks—with a focus on propagation models," *IEEE Transactions on antennas and propagation*, vol. 65, no. 12, pp. 6213–6230, 2017.

[15] Y. Niu, Y. Li, D. Jin, L. Su, A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5g: Opportunities and challenges," *Wireless networks*, vol. 21, no. 8, pp. 2657–2676, 2015.

[16] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5g cellular: It will work!" *IEEE access*, vol. 1, pp. 335–349, 2013.

[17] W. S. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[18] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*. MIT press, 2016.

[19] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, USA, 2015, vol. 25.

[20] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[21] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] V. Dumoulin, F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[23] S. Raschka, V. Mirjalili, "Python machine learning: Machine learning and deep learning with python," *Scikit-Learn, and TensorFlow. Second edition ed*, vol. 10, p. 3 175 783, 2017.

[24] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] L. Itti, C. Koch, E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[26] R. Desimone, J. Duncan, *et al.*, "Neural mechanisms of selective visual attention," *Annual review of neuroscience*, vol. 18, no. 1, pp. 193–222, 1995.

[27] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, PMLR, 2015, pp. 2048–2057.

[28] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[29] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[31] B. Wang, L. Ma, W. Zhang, W. Jiang, J. Wang, W. Liu, "Controllable video captioning with pos sequence guidance based on gated fusion network," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2641–2650.

[32] L. Yao *et al.*, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4507–4515.

[33] Y. Tian, G. Pan, M.-S. Alouini, "Applying deep-learning-based computer vision to wireless communications: Methodologies, opportunities, and challenges," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 132–143, 2020.

[34] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[35] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[36] C. Feichtenhofer, A. Pinz, A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[37] K. Hara, H. Kataoka, Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[38] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, PMLR, 2015, pp. 448–456.

[39] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[40] S. Wang, F. Liu, B. Liu, "Escaping the gradient vanishing: Periodic alternatives of softmax in attention mechanism," *IEEE Access*, vol. 9, pp. 168 749–168 759, 2021.

[41] K. Banerjee, R. R. Gupta, K. Vyas, B. Mishra, *et al.*, "Exploring alternatives to softmax function," *arXiv preprint arXiv:2011.11538*, 2020.

[42] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

A
ÖZELLİK ÇIKARACI MODEL KODLARI

A.1 ResNet-50 Modeli

Öznitelikleri çıkarırken, Kinetics veri kümesinde önceden eğitilmiş ResNet-50 modelini kullanarak tüm görüntüleri inceleyeceğiz ve öznitelikleri çıkaracağız. Bu, eğitim veri kümesi için toplam 281100 *.npy* dosyası ve doğrulama veri kümesi için 120468 *.npy* dosyası üretecektir. Kodlar [37]'e aittir.

```
def main(params):
    net = getattr(resnet, 'resnet50')()
    num_ftrs = net.fc.in_features
    net.load_state_dict(torch.load('./data/resnet-50-kinetics.
   pth'),False)
    my_resnet = myResnet(net)
    my_resnet.cuda()
    my_resnet.eval()


    train_set = hdf5storage.read(path='/img_path',
                          filename='./dev_dataset_h5/train_set.
   h5')

    print(len(train_set))

    seed(123)  # make reproducible
    for i in range(len(train_set)):
        outputs = []
        print('t',i)
        for j in range(8):
            I = skimage.io.imread('./dev_dataset_h5/' +
   train_set[i][j].replace('\\', '/'),as_gray=True)
            if len(I.shape) == 2:
                I = I[:, :, np.newaxis]
                I = np.concatenate((I, I, I), axis=2)
            I = I.astype('float32') / 255.0
            I = torch.from_numpy(I.transpose([2, 0, 1])).cuda()
            I = preprocess(I)
            with torch.no_grad():
                tmp_fc, tmp_att = my_resnet(I, params['att_size
   '])
            outputs.append(tmp_fc.data.cpu().float().numpy())
        np.save(os.path.join('/media/nasir/DATA/resnet50','
   t_video'+str(i+1)+'.npy'), outputs)
```

A.1.1 Model

```
class myResnet(nn.Module):
    def __init__(self, resnet):
        super(myResnet, self).__init__()
        self.resnet = resnet

    def forward(self, img, att_size=14):
        x = img.unsqueeze(0)

        x = self.resnet.conv1(x)
        x = self.resnet.bn1(x)
        x = self.resnet.relu(x)
        x = self.resnet.maxpool(x)

        x = self.resnet.layer1(x)
        x = self.resnet.layer2(x)
        x = self.resnet.layer3(x)
        x = self.resnet.layer4(x)

        fc = x.mean(3).mean(2).squeeze()
        att = F.adaptive_avg_pool2d(x,[att_size,att_size]).
   squeeze().permute(1, 2, 0)

        return fc, att
```

A.2 3D Resnext-101 Modeli

Aynı şekilde, eğitim ve doğrulama veri kümesi için 3D-Resnext 101 modelini kullanarak her görüntü için 3D öznitelikler çıkaracağız. Kodlar [37]'e aittir

```
if __name__=="__main__":
    opt = parse_opts()
    opt.mean = get_mean()
    opt.arch = '{}-{}'.format(opt.model_name, opt.model_depth)
    opt.sample_size = 112
    opt.sample_duration = 1
    opt.n_classes = 400
    print(opt)
    model = generate_model(opt)
    model_data = torch.load(opt.model)
    assert opt.arch == model_data['arch']

    model.load_state_dict(model_data['state_dict'],False)
    model.eval()

    class_names = []
    with open('class_names_list') as f:
        for row in f:
            class_names.append(row[:-1])

    if os.path.exists('tmp1111'):
        subprocess.call('rm -rf tmp1111', shell=True)

    train_set = hdf5storage.read(path='/img_path', filename='./
   dev_dataset_h5/train_set.h5')
    print(len(train_set))

```

```
    for i in range(len(train_set)):
        print('t ',i)

        subprocess.call('mkdir tmp1111', shell=True)
        for j in range(8):
            img_path='./dev_dataset_h5/' + train_set[i][j].
    replace('\\', '/')

            subprocess.call('cp '+img_path+' tmp1111/'+'image_
    {:05d}.jpg'.format(j+1),shell=True)

        result = classify_video('tmp1111', str(i), class_names,
     model, opt)
        outputs = []
        for kk in range(7):
            outputs.append(result['clips'][kk]['features'])

        np.save("/media/nasir/DATA/resnext101-64ftest"+'/
    t_video'+str(i+1)+'.npy', outputs)
        subprocess.call('rm -rf tmp1111', shell=True)

    if os.path.exists('tmp1111'):
        subprocess.call('rm -rf tmp1111', shell=True)

    with open(opt.output, 'w') as f:
        json.dump(outputs, f)
```

A.2.1 Model

```

class ResNet(nn.Module):

    def __init__(self, block, layers, sample_size,
    sample_duration, shortcut_type='B', num_classes=400, last_fc
    =True):
        self.last_fc = last_fc

        self.inplanes = 64
        super(ResNet, self).__init__()
        self.conv1 = nn.Conv3d(3, 64, kernel_size=7, stride=(1,
     2, 2),
                               padding=(3, 3, 3), bias=False)
        self.bn1 = nn.BatchNorm3d(64)
        self.relu = nn.ReLU(inplace=True)
        self.maxpool = nn.MaxPool3d(kernel_size=(3, 3, 3),
    stride=2, padding=1)
        self.layer1 = self._make_layer(block, 64, layers[0],
    shortcut_type)
        self.layer2 = self._make_layer(block, 128, layers[1],
    shortcut_type, stride=2)
        self.layer3 = self._make_layer(block, 256, layers[2],
    shortcut_type, stride=2)
        self.layer4 = self._make_layer(block, 512, layers[3],
    shortcut_type, stride=2)
        last_duration = math.ceil(sample_duration / 16)
        last_size = math.ceil(sample_size / 32)
        self.avgpool = nn.AvgPool3d((last_duration, last_size,
```

```
    last_size), stride=1)
        self.fc = nn.Linear(512 * block.expansion, num_classes)

        for m in self.modules():
            if isinstance(m, nn.Conv3d):
                n = m.kernel_size[0] * m.kernel_size[1] * m.
    out_channels
                m.weight.data.normal_(0, math.sqrt(2. / n))
            elif isinstance(m, nn.BatchNorm3d):
                m.weight.data.fill_(1)
                m.bias.data.zero_()

    def _make_layer(self, block, planes, blocks, shortcut_type,
    stride=1):
        downsample = None
        if stride != 1 or self.inplanes != planes * block.
    expansion:
            if shortcut_type == 'A':
                downsample = partial(downsample_basic_block,
                                     planes=planes * block.
    expansion,
                                     stride=stride)
            else:
                downsample = nn.Sequential(
                    nn.Conv3d(self.inplanes, planes * block.
    expansion,
                              kernel_size=1, stride=stride,
    bias=False),
                    nn.BatchNorm3d(planes * block.expansion)
                )

        layers = []
        layers.append(block(self.inplanes, planes, stride,
    downsample))
        self.inplanes = planes * block.expansion
        for i in range(1, blocks):
            layers.append(block(self.inplanes, planes))

        return nn.Sequential(*layers)

    def forward(self, x):
        x = self.conv1(x)
        x = self.bn1(x)
        x = self.relu(x)
        x = self.maxpool(x)

        x = self.layer1(x)
        x = self.layer2(x)
        x = self.layer3(x)
        x = self.layer4(x)

        x = self.avgpool(x)

        x = x.view(x.size(0), -1)
        if self.last_fc:
            x = self.fc(x)

        return x
```

B
MODELİ EĞİTMEK VE TEST ETMEK İÇİN KODLAR

Birçok kod satırı olduğu için sadece önemli olanları dahil etmeye çalışacağız. Kodlar [31] ve [33]'e dayanmaktadır. A üzerinde bahsedilen kodlar tarafından çıkarılan öznitelikleri girdi olarak almak ve 3.2.3 üzerinde bahsedilen yumuşak dikkat mekanizmasını uyarlamak için çeşitli değişiklikler uygulandı.

B.1 Eğitim Süreci

```
    minval_loss = 0.5 # can be specified to the desired value
    while not test_flag:
        model.train()

        update_lr_flag = True  # when a new epoch start, set update_lr_flag to True
        if update_lr_flag:
            # Assign the learning rate
            if epoch > opt.learning_rate_decay_start and opt.learning_rate_decay_start >= 0 and opt.optim != 'adadelta':
                frac = int((epoch - opt.learning_rate_decay_start) /
                           opt.learning_rate_decay_every)
                decay_factor = opt.learning_rate_decay_rate ** frac
                opt.current_lr = opt.learning_rate * decay_factor
                # set the decayed rate
                myutils.set_lr(optimizer, opt.current_lr)
                #print('epoch {}, lr_decay_start {}, cur_lr {}'.format(epoch, opt.learning_rate_decay_start, opt.current_lr))
            else:
                opt.current_lr = opt.learning_rate
            # Assign the scheduled sampling prob
            if epoch > opt.scheduled_sampling_start and opt.scheduled_sampling_start >= 0:
                frac = int((epoch - opt.scheduled_sampling_start) /
                           opt.scheduled_sampling_increase_every)
                opt.ss_prob = min(
                    opt.scheduled_sampling_increase_prob * frac, opt.scheduled_sampling_max_prob)
                model.ss_prob = opt.ss_prob
```

```
            # If start self critical training
            if opt.self_critical_after != -1 and epoch >= opt.
    self_critical_after:
                sc_flag = True
                myutils.init_cider_scorer(opt.reward_type)
            else:
                sc_flag = False
            update_lr_flag = False

        # loading train data
        myloader_train = DataLoader(
            mytrain_dset, batch_size=opt.batch_size,num_workers
    =6, collate_fn=data_io.collate_fn, shuffle=True)
        torch.cuda.synchronize()
        best_flag = False
        # print(model)
        for data, cap, cap_mask, feat1, feat2, feat_mask, lens
    in myloader_train:
            start = time.time()
            cap = Variable(cap, requires_grad=False).cuda()
            cap_mask = Variable(cap_mask, requires_grad=False).
    cuda()
            feat1 = Variable(feat1, requires_grad=False).cuda()
            feat2 = Variable(feat2, requires_grad=False).cuda()
            feat_mask = Variable(feat_mask, requires_grad=False
    ).cuda()

            input_cap = cap[:, :9]
            out_cap = cap[:, 1:10]
            groundtruth = cap[:, 9:].clone().cpu().numpy()

            optimizer.zero_grad()
            if not sc_flag:
                # (m,seq_len+1,n_words),(m, seq_len+1,
    n_classes)
                out = model(feat1, feat2, feat_mask, None, cap,
     cap_mask)
                if iteration % 50 == 0:
                    print('cap', cap[0])
                    print('out', 100, torch.argmax(out[0, :-1],
     dim=1))

                loss_language = crit(out, cap, cap_mask)
                # loss_classify = classify_crit(category,
    cap_classes, cap_mask, class_mask)
                # print(loss_language.data[0], loss_classify.
    data[0])
                loss = loss_language  # + opt.weight_class *
    loss_classify
            else:
                gen_result, sample_logprobs = model.sample(
                    feat1, feat2, feat_mask, {'sample_max': 0})

                reward = myutils.get_self_critical_reward(
                    model, feat1, feat2, feat_mask, groundtruth
    , gen_result)  # (m,max_length)
                loss = rl_crit(sample_logprobs, gen_result,
    Variable(
```

```
                    torch.from_numpy(reward).float().cuda(),
requires_grad=False))
            loss.backward()

            myutils.clip_gradient(optimizer, opt.grad_clip)
            optimizer.step()
            train_loss = loss.item()
            torch.cuda.synchronize()
            end = time.time()

            if not sc_flag and iteration % 10 == 0:
                print("iter {} (epoch {}), train_loss = {:.3f},
 loss_lang = {:.3f},  time/batch = {:.3f}".format(
                    iteration, epoch, train_loss, loss_language
.item(), end - start))

            # Write the training loss summary
            if (iteration % opt.losses_log_every == 0):
                writer.add_scalar('datas1/train_loss',
train_loss, iteration)
                writer.add_scalar('datas1/learning_rate',
                                  opt.current_lr, iteration)
                writer.add_scalar(
                    'datas1/scheduled_sampling_prob', model.
ss_prob, iteration)
                if sc_flag:
                    writer.add_scalar('datas/avg_reward',
                                      np.mean(reward[:, 0]),
iteration)

            # make evaluation on validation set, and save model
            if (iteration % opt.save_checkpoint_every == 0):
                # eval model
                print('validation and save the model...')
                time.sleep(3)
                eval_kwargs = {}
                # attend vars(opt) into eval_kwargs
                eval_kwargs.update(vars(opt))
                val_loss = eval_utils.eval_split(
                    model, crit, classify_crit, myvalid_dset,
eval_kwargs)
                print('validation is finish!')
                time.sleep(3)
                print('val_loss', val_loss)

                writer.add_scalar('datas1/validation loss',
                                  val_loss, iteration)
                # Save model if is improving o

                if val_loss < minval_loss:
                    print(val_loss, minval_loss)
                    minval_loss = val_loss
                    checkpoint_path = './model-best.pth'
                    torch.save(model.state_dict(),
checkpoint_path)
                    print("model saved to {}".format(
checkpoint_path))

```

```
                if not os.path.exists(opt.checkpoint_path):
                    os.mkdir(opt.checkpoint_path)

            if tmp_patience >= opt.patience:
                break
            iteration += 1
        if tmp_patience >= opt.patience:
            print("early stop, training is finished!")
            break
        if epoch >= opt.max_epochs and opt.max_epochs != -1:
            print("reach max epochs, training is finished!")
            break
        epoch += 1
```

Eğitim sürecinde, doğrulama kaybında bir gelişme gördüğümüzde, öğrenilebilir parametreler bir *.pth* dosyasına kaydedilecektir.

B.1.1 Genel Model

Genel modelin önemli bir kısmı aşağıdaki kodda gösterilmektedir.

```
class LSTMCore_two_layer_gate(nn.Module):

  def __init__(self,opt):
  # def __init__(self,input_encoding_size,rnn_size,drop_prob_lm
   ,feat_size,att_size):
    super(LSTMCore_two_layer_gate,self).__init__()
    torch.manual_seed(opt.seed)
    torch.cuda.manual_seed(opt.seed)

    #  word and rnn size
    self.input_encoding_size = opt.input_encoding_size  #  468
    self.rnn_size = opt.rnn_size  #  1000 or 3518

    #  feat and attention size
    self.visual_size = opt.rnn_size
    self.att_size = opt.att_size
    self.globalpos_size = opt.rnn_size

    #  drop out
    self.drop_prob_lm = opt.drop_prob_lm  #  0.5
    #  The first lstm layer
    self.gate = Gate(opt.seed, self.input_encoding_size, self.
   visual_size, self.drop_prob_lm)
    self.lstm_1 = two_inputs_lstmcell(self.input_encoding_size,
    self.visual_size, self.rnn_size, self.drop_prob_lm)
    #  The second lstm layer
    self.lstm_2 = two_inputs_lstmcell(self.rnn_size, self.
   visual_size, self.rnn_size, self.drop_prob_lm)
    #  Build a soft attention
    self.dropout = nn.Dropout(self.drop_prob_lm)
    self.v2a = nn.Linear(self.visual_size, self.att_size)  #
   Uv  (1536,att_size)
    self.h2a = nn.Linear(self.rnn_size , self.att_size)  #  Wh
    (1000,1536)
    self.a2w = nn.Linear(self.att_size, 1)  #  Wtanh(Wh+Uv+b)
   (1536,1)
```

```
30
31  def forward(self, xt, xt_mask, V, state): # xt:(m,
     input_encoding_size) xt_mask:(m,1) V:(m,28,visual_size)
     pos_feat:(m, pos_size)
32      '''state should be a list: [state_layer1, state_layer2]'''
33      assert len(state) == 2, "input parameters 'state' expect a
     list with 2 elements"
34      # the first LSTM only take "xt" as input
35      # state1, state2 = state[0], state[1]
36      state1=state[-1]
37      alpha = (self.h2a(state1[-1].squeeze()).unsqueeze(1)+self.
     h2a(state1[0].squeeze()).unsqueeze(1))/2 + self.v2a(V)   # (m
     , feat_K, att_size)
38      alpha = self.a2w(torch.tanh(alpha))   # (m, feat_K, 1)
39      alpha = F.softmax(alpha.transpose(1, 0), dim=0).transpose
     (1, 0)   # (m, feat_K, 1)
40      af = torch.sum(alpha * V, dim=1)   # (m, visual_size)
41
42      # two-layer lstm
43      output, state = self.lstm_1(xt, af, state1, xt_mask)
44      return output, [state, state]
```

B.2 Test Sureci

```
1           if test_flag:
2           model.eval()
3           batchsize = 12
4           with torch.no_grad():
5               validloader = DataLoader(
6                   myvalid_dset, batch_size=batchsize, collate_fn=
     data_io.collate_fn, shuffle=False,num_workers=6)
7               outbeam = np.zeros((len(validloader)*batchsize, 13)
     , dtype=int)
8               print(outbeam.shape)
9               m = 0
10              for data, cap, cap_mask, feat1, feat2, feat_mask,
     lens in validloader:
11
12                  tmp = [cap, cap_mask, feat1, feat2, feat_mask]
13                  tmp = [Variable(_).cuda() for _ in tmp]
14
15                  cap, cap_mask, feat1, feat2, feat_mask = tmp
16                  # forward the model to get loss
17                  inputs = cap[:, :8]
18
19                  for i in range(5):
20                      cap_mask = torch.ones_like(inputs).cuda()
21                      out = model(feat1, feat2, feat_mask, None,
     inputs, cap_mask)
22                      # print('out',out.shape,torch.argmax(out,
     dim=2)[:,-1])
23                      inputs1 = torch.zeros(batchsize, 8+i+1,
     dtype=int)
24                      inputs1[:, :-1] = inputs[:, :].clone()
25
26                      idx = torch.argmax(out, dim=2)
27                      inputs1[:, -1] = idx[:, -1]
```

```
28
29                      inputs = inputs1.cuda()
30                  outbeam[np.asarray(data) - 1, :] = inputs1.
    numpy()
31                  print(np.asarray(data) - 1)
32              print('m', m)
33
34          with open('test_resultvalid4.csv', 'wb') as file:
35              writer = csv.writer(file)
36              writer.writerows(outbeam)
```

.pth yüklendiğinde, eğitim sürecinde model gelecekteki 1, 3 ve 5 ışın indekslerini tahmin eder ve sonuçlar bir *.csv* dosyasına kaydedilir.

Doğruluğu 4.1 tablosunda gösterildiği gibi hesaplamak için *.csv* dosyasına kaydedilen sonuçlar doğru etiket ile karşılaştırılır. Kodlar aşağıdaki gibi gösterilir.

```
1  val_set = load('D:\dev_dataset_mat\dev_dataset_mat\dev_dataset.
       mat')
2  val_result=csvread('D:\test_resultvalid4.csv');
3  U=length(val_set.val_set.data);
4  val_gt=zeros(U,13);
5  for i=1:U
6      val_gt(i,:)=val_set.val_set.data(i).beam;
7  end
8  sc1=zeros(1,U);
9  sc3=zeros(1,U);
10 sc5=zeros(1,U);
11 sc1sum=0;
12 total_score=zeros(1,U);
13 for i =1:U
14     sc1(i)=exp(-norm(val_result(i,9)-val_gt(i,9),1)/1*0.5);
15     sc3(i)=exp(-norm(val_result(i,9:11)-val_gt(i,9:11),1)
       /0.5/3);
16     sc5(i)=exp(-norm(val_result(i,9:13)-val_gt(i,9:13),1)
       /0.5/5);
17     total_score(i)=(sc1(i)+3*sc3(i)+5*sc5(i))/9;
18 end
19 total_score0=mean(total_score)
20 abs_error=mean(mean(abs(val_result(:,9:13)-val_gt(:,9:13))))
21 sc1res=(1/U)*sum(sc1)
22 sc3res=(1/U)*sum(sc3)
23 sc5res=(1/U)*sum(sc5)
24
25 for i=1:U
26     tp1(i)=mean(val_result(i,9)==val_gt(i,9));
27     tp3(i)=mean(val_result(i,9:11)==val_gt(i,9:11));
28     tp5(i)=mean(val_result(i,9:13)==val_gt(i,9:13));
29 end
30 tp1res=(1/U)*sum(tp1)
31 tp3res=(1/U)*sum(tp3)
32 tp5res=(1/U)*sum(tp5)
```

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. N. Sinani and F. Yilmaz, “On the Vision-Beam Aided Tracking for Wireless 5G-Beyond Networks Using Long Short-Term Memory with Soft Attention Mechanism”, EJRnD, vol. 2, no. 2, pp. 505–520, Jun. 2022.