T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ SINIFLANDIRMADA YİNELEYEN DERİN AĞ VE GÖRÜ DÖNÜŞTÜRÜCÜ MODELLERİNİN KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Oğuzhan BUBO

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektrik Mühendisliği Bilim Dalı

HAZİRAN 2023

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ SINIFLANDIRMADA YİNELEYEN DERİN AĞ VE GÖRÜ DÖNÜŞTÜRÜCÜ MODELLERİNİN KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Oğuzhan BUBO

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektrik Mühendisliği Bilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Burhan BARAKLI

HAZİRAN 2023

Oğuzhan BUBO tarafından hazırlanan "Görüntü Sınıflandırmada Yineleyen Derin Ağ ve Görü Dönüştürücü Modellerinin Karşılaştırılması" adlı tez çalışması 07.06.2023 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı Elektrik Mühendisliği Bilim Dalı'nda Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı :	**Doç. Dr. Muhammed Kürşad UÇAR** Sakarya Üniversitesi	.....
Jüri Üyesi :	**Dr. Öğr. Üyesi Burhan BARAKLI**(Danışman) Sakarya Üniversitesi	.....
Jüri Üyesi :	**Dr. Öğr. Üyesi Ali Furkan KAMANLI** Sakarya Uygulamalı Bilimler Üniversitesi	.....

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum "Görüntü Sınıflandırmada Yineleyen Derin Ağ ve Görü Dönüştürücü Modellerinin Karşılaştırılması" başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(07/06/2023).

(imza)

Oğuzhan BUBO

Aileme

TEŞEKKÜR

Çalışma boyunca yanımda olan, bilgilerinden, tecrübelerinden yararlandığım ve beni her konuda yönlendiren değerli danışman hocam Dr. Burhan Baraklı hocama ve aileme sundukları destek adına sonsuz minnet ve teşekkürlerimi sunarım.

Oğuzhan BUBO

İÇİNDEKİLER

Sayfa

KISALTMALAR

BLEU : Bilingual Evaluation Understudy
CBAM : Convulutional Block Attention Module
CIFAR : Canadian Institute For Advanced Resarch
CNN : Convulutional Neural Network
CPU : Central Process Unit
EDRAM : Enrich-Deep Recurrent Attention Model
GeLU : Gaussian Error Linear Unit
GM : Grand Master
GPU : Graphics Process Unit
LSTM : Long Short Term Memory
MLP : Multi-layer Perceptron
MNIST : Modified National Institute Of Standards And Technology
MSA : Multi-Layer Self-Attention
NLP : Natural Language Processing
RAM : Recurrent Attention Model
ReLU : Rectified Linear Units
RNN : Recurrent Neural Network
SGD : Stochastic Gradient Descent
STL : Spatial Transformer Layer
TDK : Türk Dil Kurumu
ViT : Vision Transformer
YSA : Yapay Sinir Ağı

SİMGELER

$\boldsymbol{A_{(t)}}$: Aksiyon

$\boldsymbol{a_{ij}}$: Öğrenme oran faktörü

$\boldsymbol{a_t}$: Aksiyon

$\boldsymbol{b}$: Yanlılık değeri

$\boldsymbol{c_t}$: Hafıza hücreleri

$\boldsymbol{D}$: Sabit gizli vektör boyutu

$\boldsymbol{d_k}$: Anahtar vektörün boyutu

$\boldsymbol{E}$: Gömme matrisi

$\boldsymbol{e_{ij}}$: Karakteristik uygunluk

$\boldsymbol{f_t}$: Unut kapısı

$\boldsymbol{f_a(\theta_a)}$: Aksiyon ağı

$\boldsymbol{f_l(\theta_l)}$: Konum ağı

$\boldsymbol{G_k}$: Yörüngenin sonuna dek beklenen toplam ödül

$\boldsymbol{g_t}$: Temsili bakış

$\boldsymbol{h_t}$: Dahili temsil

H : Uzunluk

$\boldsymbol{i_t}$: Giriş kapısı

K : Anahtar matrisi

$\boldsymbol{L}$: Özdeş katman

$\boldsymbol{l_{t-1}}$: Konum merkezi

$\boldsymbol{o_t}$: Çıkış kapısı

$\boldsymbol{p_j}$: Tahmini değerler

$\boldsymbol{\rho(X_t, l_{t-1})}$: Temsili retina

Q : Sorgu matrisi

$\boldsymbol{R_{(t)}}$: Ödül

$\boldsymbol{S_{(t)}}$: Durum

tanh : Hiperbolik tanjant aktivasyon fonksiyonu

$\boldsymbol{\tau}$: Yörünge

$\mathbf{U}(\boldsymbol{\theta})$: Ağırlık fonksiyonu

$\mathbf{V}$: Değer matrisi

$\mathbf{w}$: Ağırlık

$\mathbf{X}$: YSA girdisi

$\boldsymbol{X_t}$: Giriş görüntüsü

$\boldsymbol{y_j}$: Reel değerler

$\boldsymbol{z_0}$: Transformatör kodlayıcıya ilk giriş

$\boldsymbol{\gamma}$: İndirim faktörü

$\boldsymbol{\sigma}$: Sigmoid aktivasyon fonksiyonu

$\boldsymbol{\theta}$: Maksimum ödülü getirecek ağırlık

$\boldsymbol{\pi}$: Politika

TABLO LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

GÖRÜNTÜ SINIFLANDIRMADA YİNELEYEN DERİN AĞ VE GÖRÜ DÖNÜŞTÜRÜCÜ MODELLERİNİN KARŞILAŞTIRILMASI

ÖZET

Görüntü sınıflandırma, görüntülerde veya videolarda bulunan belli nesneleri algılayıp, görüntü işleme alanında kullanan bir teknoloji aracıdır. Bu işlem, aynı zamanda bir dizi sınıflandırma sürecinin bir parçasıdır. Makineler aracılığıyla yapılan bu sınıflandırma işlemlerinde uzun zamandır çeşitli teoriler ve yöntemler ortaya atılmış ve uygulanmıştır. Geometrik, bölgesel tabanlı, mantıksal nitelikli, dikkat tabanlı çok sayıda modelleme çeşitli uygulamalarda sıklıkla kullanılmaktadır. İhtiyaç ve gereksinimlere göre çeşitli modeller ortaya atılmaya devam etmektedir. Son yıllarda, sinir ağları ile görüntü sınıflandırma çalışmaları gitgide artmaktadır. Bu çalışmada yineleyen sinir ağları, pekiştirmeli öğrenme, görü dönüştürücü kullanılarak görüntü sınıflandırma işleminin gerçekleştirilmesi hedeflenmektedir. Çalışmada dikkat tabanlı iki model olan, yineleyen dikkat modeli ile görü dönüştürücü aracılığıyla görüntü sınıflandırma işleminin gerçekleştirilmesi ve her iki dikkat modelinin karşılaştırılması ana hedeftir. Sinir ağı, sunulan nesneleri önceki öğrendikleriyle karşılaştırarak aradaki ilişkiyi tespit edip, nesneleri sınıflandırmaktadır. Yineleyen dikkat modelleri, son dönemlerde farklı tanıma ve sınıflandırma çalışmalarında üstün başarı göstermiştir. Çalışmada, yineleyen görsel dikkat modeli sınırlı bir sensör vasıtasıyla görsel bir ortamla ilişkili, hedef odaklı bir araçtır. Model, pekiştirmeli öğrenme ile görsel bir çevreyle ilişkiye giren hedefe yönelik bir ajanın birbirini izleyen karar verme prosesidir. Yineleyen dikkat modelinin çalışma prensibi, görüntünün sadece ilgilenilen bölümüne odaklanmaya dayanmaktadır. Bu sayede zamandan tasarruf sağlanarak, ilgilenilen piksel sayısında düşüş oluşur. Pekiştirmeli öğrenmedeki ajan, eylemleri yürütüp, ortamın reel gidişatını da etkiliyebilir. Çevre sınırlı gözlemlenebildiğinden, ajanın nasıl bir yol izleyeceğini ve sensörünü en etkin bir biçimde nasıl yerleştireceğini tespit etmek için bilgiyi zamanla entegre etmesi gerekmektedir. Her adımda, ajan bir ödül almaktadır. Ajanın ana amacıda, ödüllerin maksimuma çıkarılmasıdır. Ödüllerin maksimum olduğu haller için derin ağ modeli oluşturulmuş olmaktadır. Karşılaştırmada kullanılan bir diğer dikkat tabanlı model ise görü dönüştürücüdür. Görü dönüştürücü, yineleyen dikkat modeline göre daha yeni ortaya çıkmış bir modeldir. Dönüştürücü sinir ağları, ilk olarak NLP adı verilen dil işleme çalışmalarında sıklıkla kullanılmıştır. Bu tip çalışmalarda otomatik çeviri ile insan çevirisi arasındaki değişikliğin ölçüldüğü BLEU puanı adı verilen sistem kullanılmaktadır ve başarılı sonuçlar elde edilmektedir. Bu çalışmaların sonrasında, görü dönüştürücüler çeşitli sınıflandırma uygulamalarında da kullanılmaya başlamış ve harcanan zaman ve doğruluk açısından iyi sonuçlar elde edilmiştir. Görü dönüştürücü modelinde önce giriş görüntüleri sabit boyutlu parçalara ayrılarak, düzleştirilip bir vektör elde edilir. Bu vektörlerden doğrusal gömme dizileri oluşturulur. Oluşturulan bu gömme dizileri transformatör kodlayıcıya giriş olur. Modelde, çok başlı öz dikkat katmanı, çok katmanlı algılayıcı ve norm katmanları

bulunmaktadır. Modelde, kişisel dikkat katmanı görüntü bilgilerinin gömülmesi ve yeniden yapılandırma görevlerine sahiptir. Yineleyen sinir ağları önemli iki probleme sahiptir. Bunlar; zaman olarak eğitimin uzunluğu ve kaybolan gradyan problemidir. Eğitim süresinin uzunluğu, güçlü GPU ve CPU'lar ile çözüme kavuşturulabilir. Ancak kaybolan gradyan problemi daha önemli bir sorundur. Geçmişten gelen bilgileri kullanmada uzun vadede yineleyen sinir ağlarında sıkıntılar oluşmaktadır. Bu kısmen uzun kısa süreli bellek aracılığıyla çözülmeye çalışılsada optimum düzeyde performans sağlayamamaktadır. Uzun kısa süreli bellek geçmişten gelen bilginin kullanımında yineleyen ağlara göre çok daha başarılı davranabilmektedir. Dönüştürücü sinir ağları kullanıldığında, bilgiler gömülerek belirlenebilir. Görü dönüştürücü dikkat tabanlı bir model olmakla beraber, bu gömülü bilgiler saklanarak, değişik ağırlık değerleri ile öncelik sırası belirlenerek kullanılmaktadır. Sınıflandırma uygulamalarında, sunulan nesnenin üstün bir başarıyla tespiti için kullanılan veri seti belli aşamalardan geçmelidir. Çalışma, PyTorch üzerinde gerçekleştirilmiştir. Bu kütüphanenin erişim sağlayabildiği datasetlerinden Fashion-MNIST ve CIFAR-10 datasetleri her iki dikkat tabanlı modeller üzerinde test edilerek, çıkan sonuçlar karşılaştırılmıştır. CIFAR-10 dataset çeşitli ulaşım araçları ve hayvanlar olmak üzere on farklı sınıftan oluşmaktadır. Görüntü boyutu 32x32'dir ve 60000 görüntüden oluşmaktadır. Fashion-MNIST dataseti ise on farklı giyim ürünü türünden oluşan ve 70000 görüntüye sahip bir datasettir. Deney kısmında, google colab'in kullanıcılarına sunduğu Tesla T4 GPU kullanılmıştır. İlk olarak Fashion-MNIST datasetinin yineleyen dikkat modeli üzerinde eğitimi gerçekleştirilmiştir. Modelin yüksek duyarlı olduğu patch boyutu, bakış sayısı, iterasyon sayısı, öğrenme oranı gibi parametreleri değiştirilerek tepkileri ölçülmüş ve birçok eğitim gerçekleştirilmiştir. Görüntü boyutları göz önüne alınarak patch boyutunun sekiz olması uygun görülmüştür. Bakış sayısı yükseltildiğinde nispeten ufak artışlar olmasına rağmen eğitim süresinin ciddi bir oranda arttığı gözlemlenmiştir. Bu nedenle altı olarak uygun görülmüştür. İterasyon sayısı 300 olarak alınsada yeterli gelişme görülmediğinden eğitim 256. iterasyonda sonlandırılmıştır. Öğrenme oranı zamanla azalan biçimdedir. Eğitim sonuç grafikleri incelendiğinde, eğitim ve test kayıp oranları azalırken; eğitim ve test doğruluk oranları ise artmaktadır. Daha sonrasında ise aynı işlemler CIFAR-10 dataseti üzerinde gerçekleştirilmiştir. Görüntü boyutuna göre patch boyutu on iki olarak alınmıştır. Bakış sayısı sekize yükseltilmiştir. Fashion-MNIST datasetindeki iterasyon ve öğrenme oranı ise sabit kalmıştır. Kayıp ve doğruluk oranları bir önceki datasetle benzer çizgidedir. Bir sonraki aşamada görü dönüştürücü modelinin Fashion-MNIST ve CIFAR-10 datasetleri üzerinde eğitimi gerçekleştirilmiştir. İlk olarak, Fashion-MNIST datasetinde, görü dönüştürücü modelin duyarlı olduğu patch boyutu, derinlik, MLP boyutu gibi parametreleri değiştirirek çeşitli eğitimler gerçekleştirilmiştir ve sonuçlar incelenmiştir. Seçilen parametre değerleri tablo halinde verilmiştir. İterasyon sayısında kırk esas alınmıştır, çünkü bu değer sonrasında kaybın yükseldiği gözlemlenmiştir. Test ve eğitim kayıp oranları azalırken, doğruluk oranları ise artmaktadır. Daha sonrasında benzer işlemler CIFAR-10 dataset üzerinde de gerçekleştirilmiştir. Çeşitli denemelerden sonra optimum parametre değerleri belirlenmiştir. Kayıp ve doğruluk oranları bir önceki datasetle benzer çizgidedir. Fashion-MNIST datasetinin yineleyen dikkat modeli üzerinde gerçekleştirilen eğitimi sonucu, eğitim ve test kayıp oranları azalırken; eğitim ve test doğruluk oranları ise artmıştır. Bu eğitim süreci 256. iterasyonda sonuçlanmıştır. Bunun akabinde, Fashion-MNIST datasetinin görü dönüştürücü modeli üzerinde eğitimi sonucunda, yine aynı şekilde eğitim ve test kayıp oranları azalırken; eğitim ve test doğruluk oranları ise

artmıştır. Görü dönüştürücü modelinde kırk iterasyon gibi bir süreçte, yineleyen dikkat modeli ile gerçekleştirilen eğitim sonucunda ulaşılan değerlere yaklaşmıştır. Fashion-MNIST datasetinin yineleyen dikkat modeli ile görü dönüştürücü modeli arasında zamansal bir kıyaslama yapıldığında, görü dönüştürücü modelinin yineleyen dikkat modeline göre daha avantajlı olduğu görülmektedir. CIFAR-10 datasetinin yineleyen dikkat modeli üzerinde gerçekleştirilen eğitimi sonucu, eğitim ve test kayıp oranları azalırken; eğitim ve test doğruluk oranları ise artmıştır. Bu eğitim süreci 293. iterasyonda sonuçlanmıştır. Bunun akabinde, CIFAR-10 datasetinin görü dönüştürücü modeli üzerinde eğitimi sonucunda, yine aynı şekilde eğitim ve test kayıp oranları azalırken; eğitim ve test doğruluk oranları ise artmıştır. Görü dönüştürücü modelinde 200 iterasyon gibi bir süreçte, yineleyen dikkat modeli ile gerçekleştirilen eğitim sonucunda ulaşılan değerlerden çok daha iyi sonuçlar elde edilmiştir. CIFAR-10 datasetinin yineleyen dikkat modeli ile görü dönüştürücü modeli arasında zamansal bir kıyaslama yapıldığında, görü dönüştürücü modelinin yineleyen dikkat modeline göre daha avantajlı olduğu görülmektedir. Sonuç olarak deney bölümünde, her iki dataset eğitim sonuçlarına ait grafik ve tablolar sunulmuştur. Elde edilen sonuçlara göre görü dönüştürücü modeli, yineleyen dikkat modeline oranla eğitim süresi bakımından daha hızlı sonuçlanırken, elde edilen doğruluk ve kayıp oranlarında da daha iyi sonuçlar verdiği gözlemlenmiştir. Önceden eğitilmiş verisetleri kullanıldığında daha iyi sonuçlar elde edileceği öngörülmektedir.

COMPARISON OF RECURRENT DEEP NETWORK AND VISION TRANSFORMER MODELS IN IMAGE CLASSIFICATION

SUMMARY

Image classification is a technology tool that detects certain image in images or videos and uses them in image processing. This process is also part of a series of classification processes. Various theories and methods have been put forward and applied for a long time in these classification processes made by machines. A large number of geometric, regional-based, logical, attention-based models are frequently used in various applications. Various models continue to be introduced according to needs and requirements. In recent years, image classification studies with neural networks have been increasing. In this study, it is aimed to perform image classification by using recurrent neural networks, reinforcement learning, and vision transformer. The main objective of the study is to perform the image classification process through the recurrent attention model and the vision transformer, which are two attention-based models, and to compare both attention models. The neural network compares the presented images with what it has learned before, detects the relationship between them and classifies the images. Recurrent attention models have shown superior success in different recognition and classification studies in recent years. In the study, the recurrent visual attention model is a goal-oriented tool associated with a visual environment through a limited sensor. The model is the sequential decision-making process of a goal-directed agent that engages with a visual environment through reinforcement learning. The working principle of the recurrent attention model is based on focusing only on the part of the image of interest. This saves time and reduces the number of pixels of interest. The agent in reinforcement learning can also carry out the actions and affect the real course of the environment. Since the environment is limitedly observable, the agent needs to integrate the information over time to determine what path to take and how to deploy its sensor most effectively. At each step, the agent receives a reward. The main purpose of the agent is to maximize the rewards. For the cases where the rewards are maximum, the deep network model is created. Another attention-based model used in comparison is the vision transformer. The vision transformer is a more recent model than the recurrent attention model. Transformer neural networks were first used frequently in language processing studies called NLP. In such studies, a system called BLEU score, which measures the change between automatic translation and human translation, is used and successful results are obtained. After these studies, vision transformer started to be used in various classification applications and good results were obtained in terms of time and accuracy. In the vision transformer model, the input images are first divided into fixed-size pieces, flattened and a vector is obtained. Linear embedding sequences are created from these vectors. These built-in sequences are input to the transformer encoder. In the model, there are multi-headed self-attention layer, multi-layer perceptron and norm layers. In the model, the personal attention layer has the tasks of embedding and

reconstructing image information. Recurrent neural networks have two important problems. These; is the length of the training in time and the vanishing gradient problem. The length of training time can be solved with powerful GPUs and CPUs. But the vanishing gradient problem is a more important one. Problems occur in recurrent neural networks in the long term in using information from the past. Although this is partially solved through long short-term memory, it cannot provide optimum performance. Long short-term memory can act much more successfully than recurrent networks in the use of information from the past. When using transformer neural networks, information can be determined by embedding. Although the vision transformer is an attention-based model, this embedded information is stored and used by determining the priority order with different weight values. In image classification applications, the data set used to detect the presented image with superior success must go through certain stages. The study was carried out in the PyTorch library. Fashion-MNIST and CIFAR-10 datasets, which this library can access, were tested on both attention-based models and the results were compared. The CIFAR-10 dataset consists of ten different classes, including various transportation vehicles and animals. The image size is 32x32 and consists of 60000 images. The Fashion-MNIST dataset, on the other hand, is a dataset consisting of ten different clothing product types and has 70000 images. In the experiment part, Tesla T4 GPU offered by google colab to its users was used. First, training on the recurrent attention model of the Fashion-MNIST dataset was carried out. By changing the parameters such as patch size, number of glimpses, number of iterations, learning rate, which the model is highly sensitive to, its responses were measured and many trainings were carried out. Considering the image dimensions, it was deemed appropriate to have a patch size of eight. Although there were relatively small increases when the number of views was increased, it was observed that the training time increased significantly. Therefore, six was considered appropriate. Although the number of iterations was taken as 300, the training was terminated at the 256th iteration, since there was not enough improvement. The learning rate is in a decreasing form over time. When the training result graphs are examined, while training and test loss rates are decreasing; training and test accuracy rates are increasing. Afterwards, the same operations were performed on the CIFAR-10 dataset. According to the image size, the patch size was taken as twelve. The number of views has been increased to eight. The iteration and learning rate in the Fashion-MNIST dataset remained constant. Loss and accuracy rates are in line with the previous dataset. In the next step, training of the vision transformer model was carried out on Fashion-MNIST and CIFAR-10 datasets. First, in the Fashion-MNIST dataset, various trainings were carried out by changing the parameters such as patch size, depth, MLP size that the vision transformer model is sensitive to, and the results were examined. Selected parameter values are given in the table. The number of iterations was based on forty, because it was observed that the loss increased after this value. While test and training loss rates are decreasing, accuracy rates are increasing. Later, similar operations were performed on the CIFAR-10 dataset. Optimum parameter values were determined after various trials. Loss and accuracy rates are in line with the previous dataset. As a result of the training of the Fashion-MNIST dataset on the recurrent attention model, the training and test loss rates decreased; training and test accuracy rates have increased. This training process is concluded in the 256th iteration. Subsequently, as a result of training the Fashion-MNIST dataset on the vision transformer model, the training and test loss rates decreased; training and test accuracy rates have increased. In a process such as forty iterations in the vision transformer

model, it approached the values reached as a result of the training carried out with the recurrent attention model. When a temporal comparison is made between the recurrent attention model and the vision transformer model of the Fashion-MNIST dataset, it is seen that the vision transformer model is more advantageous than the recurrent attention model. As a result of the training of the CIFAR-10 dataset on the recurrent attention model, the training and test loss rates decreased; training and test accuracy rates have increased. This training process concluded in the 293rd iteration. Subsequently, as a result of the training of the CIFAR-10 dataset on the vision transformer model, the training and test loss rates decreased; training and test accuracy rates have increased. In the vision transformer model, in a process such as 200 iterations, much better results were obtained than the values reached as a result of the training carried out with the recurrent attention model. When a temporal comparison is made between the recurrent attention model and the vision transformer model of the CIFAR-10 dataset, it is seen that the vision transformer model is more advantageous than the recurrent attention model. As a result, graphics and tables of both dataset training results are presented in the experiment section. According to the results obtained, it has been observed that the vision transformer model results faster in terms of training time compared to the recurrent attention model, while it gives better results in terms of accuracy and loss rates. It is predicted that better results will be obtained when pre-trained datasets are used.

1. GİRİŞ

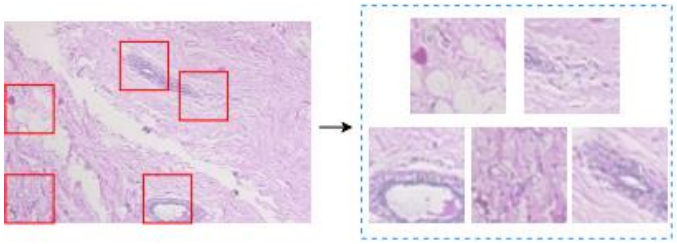
Canlılar, istek ve gereksinimlerine bağlı olarak görsel olarak karşılaştıkları nesneleri farklı özellikler açısından sınıflandırmak veya tanımak isterler. Sınıflandırma ve tanıma işleminin makineler aracılığıyla gerçekleştirilmesi için son 50-60 yılda çeşitli teoriler ve yöntemler ortaya atılmış ve halen geliştirilmeye devam edilmektedir [1].

Sayısal görüntü işleme alanında; sınıflandırma, görüntülerden elde edilen bazı özellikler ile görüntülerin birbirinden ayırt edilmesi işlemi olarak tanımlanır. Nesne tanıma ve sınıflandırma uygulamaları, çoğunlukla üretim ve dağıtım yapan çeşitli sanayi organizasyonlarında askeri sanayide, tıp sektöründe, güvenlik alanında ve hayatımızın birçok alanında sıklıkla kullanılmaktadır [2].

Sayısal görüntüde sınıflandırma alanında ilk olarak geometrik tabanlı yaklaşım kullanılmış, ancak çeşitli açılardan yeterli görülmeyip daha iyi algoritmalara ihtiyaç duyulmuştur [3]. Daha sonrasında nesne parçalarının yardımıyla yapılan sınıflandırma çalışmaları yaygınlaşmış, nesnelerin bütününü oluşturan parçalardan nesnenin tamamına ulaşılmaya çalışılmıştır [4]. Bunların yanında; lokal ve global modelleme, imgesel ve bölgesel tabanlı, mantıksal niteliği temel alan çeşitli sınıflandırma yaklaşımları görüntü sınıflandırma uygulamalarında sıklıkla kullanılmıştır [1].

Yaşayan her canlı varlıkta beyin veya beyin gibi çalışan yapılar mevcuttur. Canlı varlık, hayatını düzenli olarak devam ettirebilmek için bazı savunma ve saldırı teknikleri geliştirmiştir. Örneğin bazı bitkiler koku ve dokunma ile hayatını devam ettirirken; hayvanlar koku, dokunma, görme ve hisleriyle davranmaktadırlar. İnsan bebeğini düşünürsek; annesi ve babasını tanırken, başka birinin kucağındayken ağlamaya başlar. Her canlı hayatının bazı alanlarında bazı olgulara dikkat etmektedir. Her canlının nesnelere bakışı farklıdır. İnsanoğlunun nesneleri algılaması çoğunlukla nesnenin alanlarını tanıması ile gerçekleşir. Örneğin, bir bebeğe veya çocuğa, inek ile köpeği öğretirken; aslında bu iki canlı arasındaki farklar öğretilir. Kısaca bir nesneyi veya canlıyı tanırken; nesnenin bazı bölümlerine dikkat ederler. Dikkatin TDK anlamı; ilgi, özen, detaylı olma, kılı kırk yarma, ince eleyip sık dokuma olarak belirtilmektedir. Görüntü sınıflandırma uygulamalarında; dikkat, önemli bir çalışma alanı olmuştur.

Yapay zeka çalışmalarında dikkat konusuna dikkat mekanizması adı verilmiştir [5]. Şekil 1.1'de meme kanseri görüntülerinin dikkat tabanlı sınıflandırılmasının yapıldığı bir çalışmadan dikkat mekanizmasının nasıl seçim yaptığına dair örnek görüntü gösterilmektedir [6].



Şekil 1.1. Dikkat mekanizmasının nasıl seçim yaptığına dair örnek görüntü [6].

Sınıflandırmada veri kümeleri kullanılır. Çoğunlukla veri kümesindeki her bir elemanın sınıfı bellidir. Ancak son yıllarda geliştirilen pekiştirmeli öğrenme ile sınıfının belli olmasına bazı uygulamalarda gerek yoktur. Ayrca veri kümesindeki eleman ve özellik sayısının uygulamaya bağlı olarak belli sayıda olması istenir. Son yılların en yaygın konularından biri de yapay zeka alanıdır. Yapay zekanın bir alt kolu olan makine öğrenmesi tabanlı bazı yaklaşımlar [7, 8] aşağıda şekil 1.2'de gösterilmektedir.

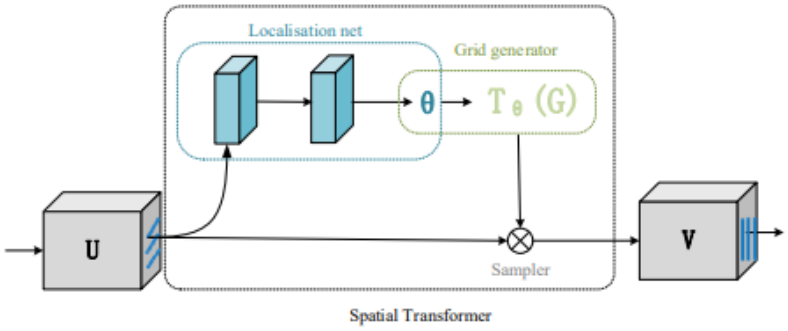
Şekil 1.2. Makine öğrenmesinde kullanılan teknikler.

Yapay zekanın bir alt dalı olan ve yaygınlığı giderek artan çeşitli derin öğrenme yöntemleri geliştirilmektedir. Konvolüsyonel sinir ağı (CNN), yineleyen sinir ağı (RNN), pekiştirmeli öğrenme ve transformerlar derin öğrenmedeki önemli yöntemlere örnek olarak verilebilir. Derin öğrenme tabanlı uygulamalara ise doğal dil işleme, görüntü ve video işleme, biyomedikal sinyal ve görüntü işleme gibi alanlar örnek verilebilir [9].

Bu tezde görsel dikkat mekanizması kullanılmıştır. Görsel dikkat modellerinde, canlılar gibi sadece ilgilenilen alana odaklanarak gereksiz yükten tasarruf sağlanmış ve geçmiş sınıflandırma yöntemlerine göre başarılı sonuçlar elde edilmiştir. Derin öğrenme alanında çeşitli dikkat mekanizmaları geliştirilmiştir. Dikkat modelleri yumuşak dikkat ve sert dikkat olarak ayrılmaktadır. Yumuşak dikkatin derin öğrenme bakımından uygulanması kolaydır ve görüntünün değişik bölgelerine odaklanılarak işlem yapılmaktadır. Sert dikkatte ise bir bölgeye odaklanılır, bölgeler rastgele seçilir ve derin öğrenme yöntemleri kullanılarak gerçekleştirilir [10].

Yumuşak dikkat, genel olarak sınıflandırma, video-görüntü işleme, segmentasyon ve algılama gibi çeşitli alanlarda kullanılmaktadır ve mekanizmaları uzamsal, kanal, karma ve öz dikkat olarak gösterilmektedir [10].

Uzamsal dikkat, Uzamsal dönüştürücü katmanı (STL) adlı çalışma ile DeepMind tarafından tasarlanmıştır ve bu ağın temsili yapısıda şekil 1.3'te gösterilmektedir [11].

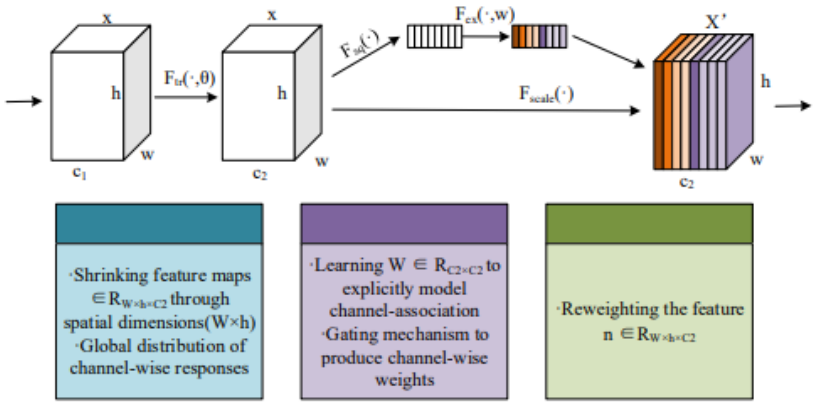


Şekil 1.3. STL model yapısı [9].

Bu dikkat mekanizmasında; yerelleştirme ağı, giriş görüntüsünden bir θ elde etmektedir. Daha sonrasında, giriş ve çıkış görüntüsünün koordinat bilgilerini θ'ya dayanarak hesaplamaktadır [10].

Son olarak ise örnekleyici bazı kurallar kullanarak görüntü V'yi oluşturur. Böylece girişte verilen görüntü, çıkışta STL aracılığıyla istenen şekilde elde edilebilmektedir [10].

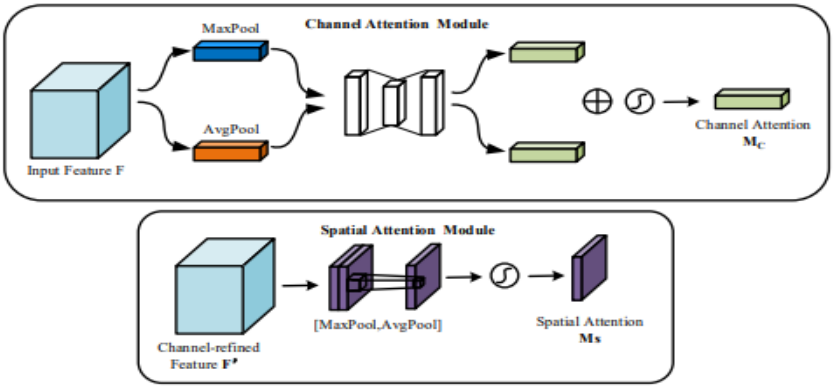
Kanal dikkat mekanizması modeline, 2017'de ödül alan SENet modeli örnek olarak verilebilmektedir [12]. Modelin prensibi şekil 1.4'te gösterilmektedir.



Şekil 1.4. SENet prensibi [10].

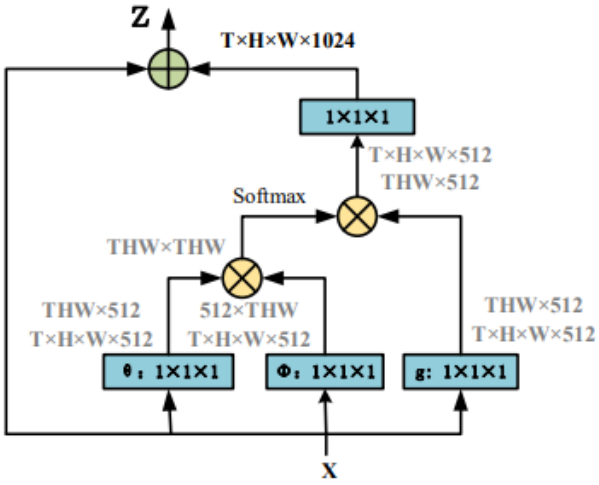
Kanal dikkat mekanizmasında, derin öğrenme ağırlıkları her kanala (R,G,B) uygulanarak, belli görevler aracılığıyla kanalların önemi ortaya çıkarılmıştır. Kısaca bir CNN ağı ile sınıflandırma gerçekleştirilmektedir. Bu dikkat mekanizması modeli, hesaplama gücü gerektirse de, verim oldukça arttırılmıştır [10].

Karma dikkat, birçok dikkat mekanizmasını birleştirerek, performans artışı sağlamaktadır. CBAM [13], buna bir örnektir. Bu modelde kanal ve uzamsal dikkat birleştirilerek uygulanmıştır [10]. Şekil 1.5'te CBAM yapısı verilmektedir.



Şekil 1.5. CBAM yapısı [10].

Öz dikkat, eğitim ve tahmin gerçekleştirilirken; benzer pikselleri kullanma, buna karşın farklı pikselleri görmezden gelme prensibine dayanmaktadır [10]. Her piksel düzeyinde tahmin için, yerel olmayan sinir ağları teorisi ortaya atılmıştır [14]. Şekil 1.6'da yerel olmayan sinir ağları yapısı gösterilmektedir.

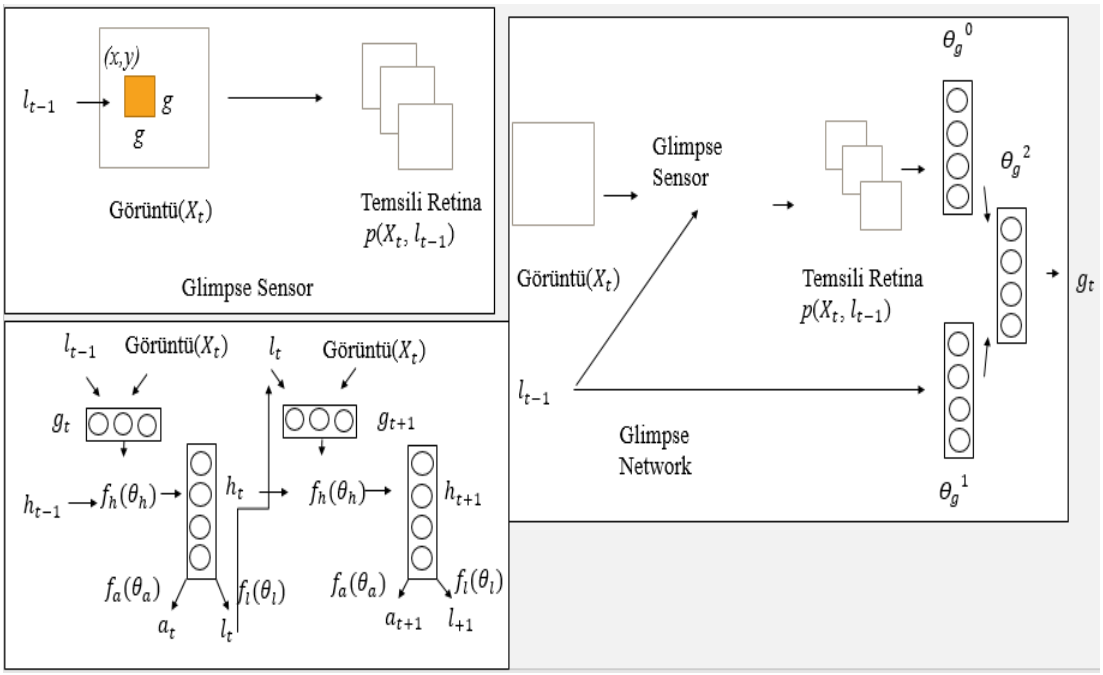


Şekil 1.6. Yerel olmayan sinir ağları [10].

Sert dikkat mekanizmasıyla alakalı yapılan çalışmalar, yumuşak dikkat mekanizmasına oranla nispeten daha azdır. Bunun nedeni pekiştirmeli öğrenme ve monte carlo yöntemi gibi değişken modelleri kullanmasından kaynaklanır. Bu mekanizma, giriş bilgilerinden dikkat çekici özellikleri seçebildiği için, verimli bir modeldir [10].

Sert ve yumuşak dikkat ile ilgili son yıllarda çıkan en başarılı yöntemler aşağıda verilmiştir.

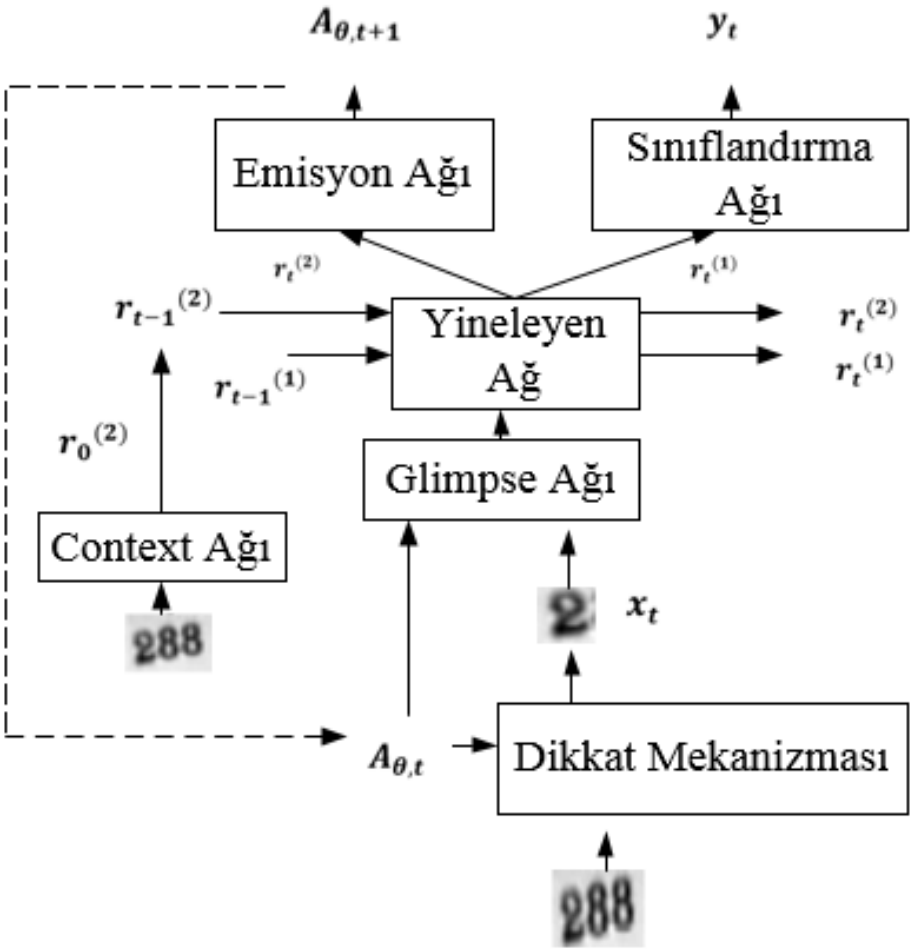
Mnih ve arkadaşları (2014), yaptıkları çalışmada giriş görüntüsünün tamamına odaklanmak yerine spesifik olarak ilgilenilen merkezi bir bölgeye odaklanmanın hesaplama açısından avantaj oluşturacağını belirterek, yineleyen dikkat modelini (RAM) sunmuşlardır [15]. Modelin, ilgilenilen bölgeyle etkileşime girdiğinden karmaşık bir görüntüdeki dağınıklıktan etkilenmeyeceğini ve sınıflandırma uygulamalarında karşılaştırılabilir düzeyde parametreyle evrişimsel bir modelden daha yüksek performans gösterebildiğini belirtmişlerdir [15]. Şekil 1.7'de sunulan gösterilmektedir.



Şekil 1.7. Yineleyen dikkat modeli [15].

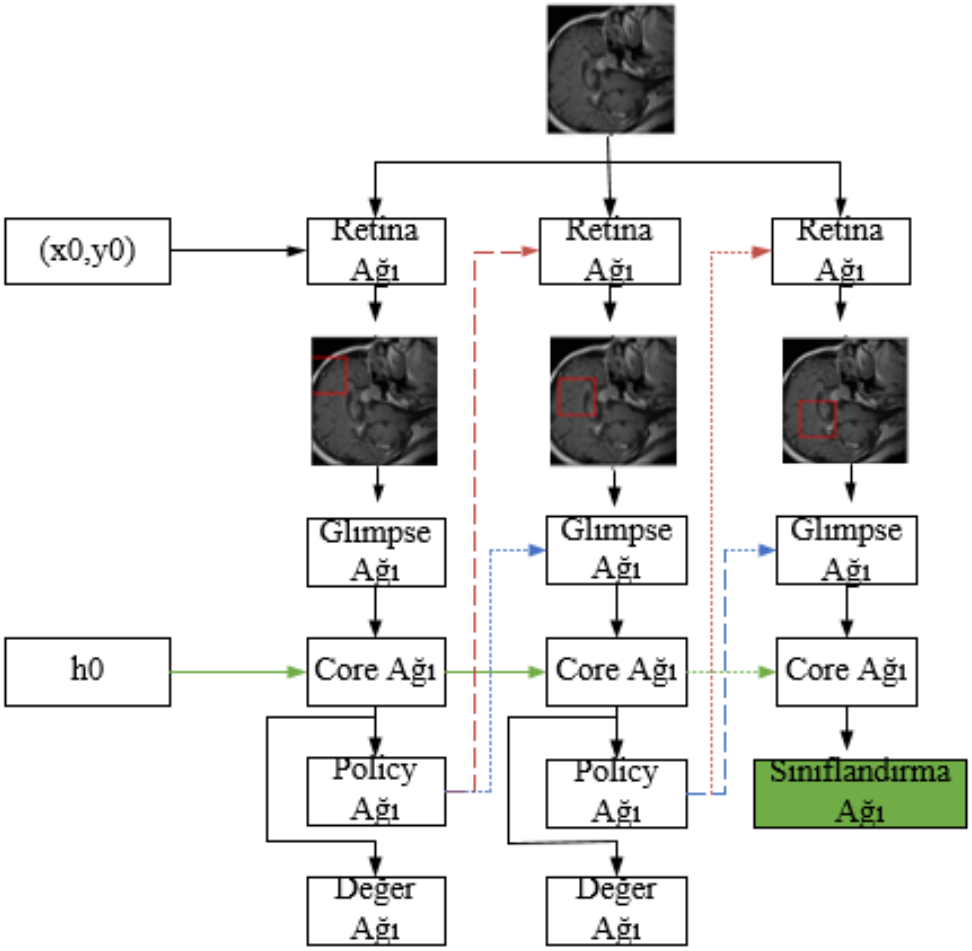
Vakanski ve arkadaşları (2020), 510 adet ultrason görüntüsünden oluşan bir veri seti kullanarak, yüksek belirginlik seviyelerine odaklanan meme tümörü için dikkatle zenginleştirilmiş bir derin öğrenme modeli sunmuş ve başarılı sonuçlar elde etmişlerdir. Çalışmanın diğer farklı organların tıbbi görüntülerinde de başarı sağlayabileceğinin beklendiği vurgulamıştır [16].

Ablavatski ve arkadaşları (2017), nesne tanıma ve sınıflandırma uygulamalarında kullanılması için zenginleştirilmiş derin yineleyen dikkat modelini (EDRAM) sunmuşlardır. EDRAM, yineleyen derin ağları ve esnek bir dikkat mekanizmasını kullanarak SGD ile bütün bir ağın uçtan uca eğitilmesini sağlamayı amaçlamaktadır. Çalışmada; sinir ağı, SGD ve Adam optimizasyon algoritması aracılığıyla eğitilmiş, hesaplama ve performans açısından avantajlı bir model olduğu belirtilmiştir [17]. Şekil 1.8'de EDRAM model mimarisi gösterilmektedir.



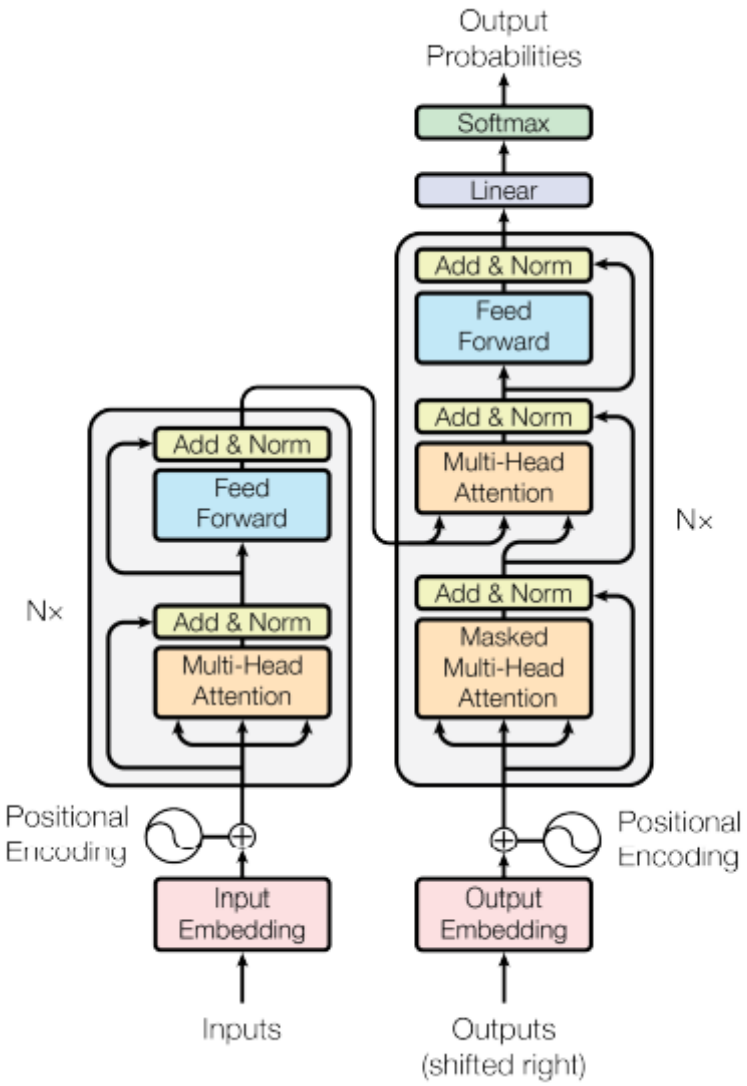
Şekil 1.8. EDRAM model mimarisi [17].

Shaikh ve arkadaşları (2015), biyomedikal alanındaki tıbbi görüntülerin sınıflandırılmasında, bilgi kaybından ve evrişimli sinir ağlarındaki hesaplama yükünden kurtulmak amacıyla yineleyen bir dikkat mekanizması kullanmayı önermişlerdir. Çalışma manyetik rezonans görüntülerinden beyin tümörünün sınıflandırılması ve fundus görüntülerinden diyabetik maküler ödemin ehemmiyetinin tespit edilmesi olarak iki farklı sınıflandırma görevini içermektedir. Modelin, her iki sınıflandırma görevinde de üstün performans gösterdiği belirtilmiştir [18]. Şekil 1.9'da beyin tümörü sınıflandırması için dikkat mekanizması gösterilmektedir.



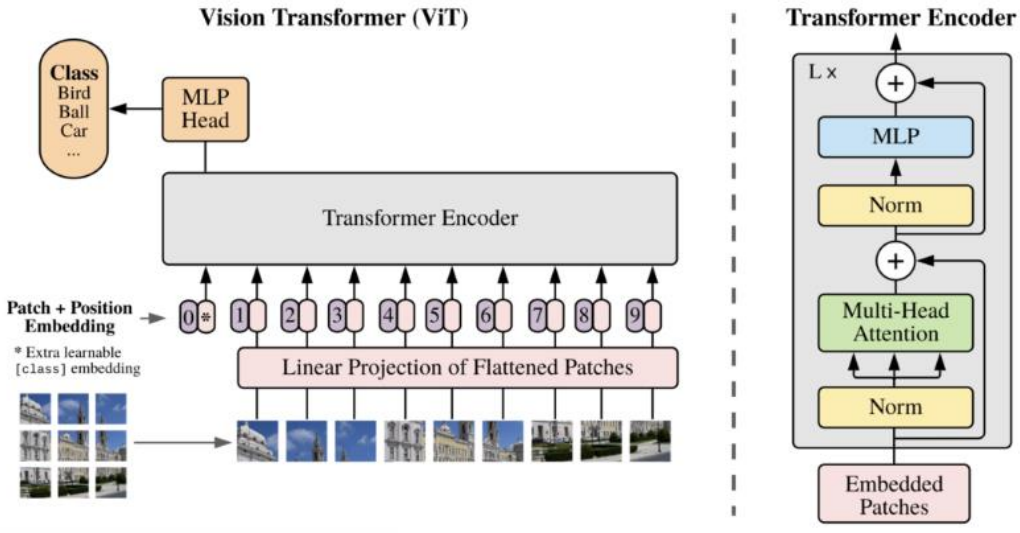
Şekil 1.9. Beyin tümörü sınıflandırması için dikkat mekanizması [18].

Vaswani ve arkadaşları (2017), sadece dikkat tabanlı, dezavantajlarından dolayı yineleyen ağlardan vazgeçerek, yeni bir ağ modeli olan transformer (dönüştürücü) ağ yapısını sunmuşlardır. Çeviri çalışmalarında, otomatik çeviri ile insan kaynaklı çevirilerin farkının puansal değeri olan BLEU puanı bakımından İngilizce-Almanca ve İngilizce-Fransızca çevirilerindeki performansının yanında, dönüştürücü ağların eğitim süresi bakımından da üstün performans gösterdiği belirtilmiştir [19]. Şekil 1.10'da dönüştürücü ağ mimarisi gösterilmektedir.



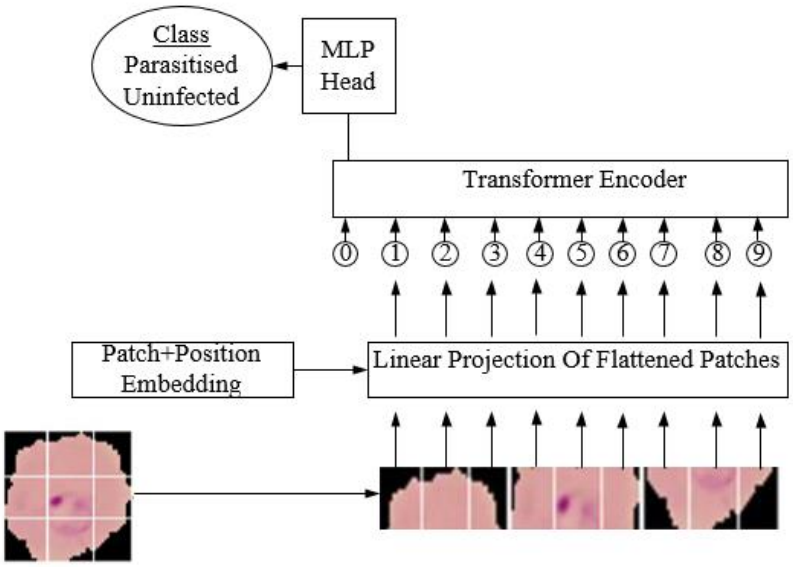
Şekil 1.10. Dönüştürücü ağ mimarisi [19].

Dosovitskiy ve arkadaşları (2020), yaptıkları çalışmada, NLP ve çeviri uygulamalarında popüler olan dönüştürücü ağların görüntü sınıflandırma çalışmalarında üstün performans gösterebileceğini öngörerek, dönüştürücü ağları temel alan görü dönüştürücü (vision transformer) modeli sunmuşlardır. Model, zaman tasarrufu açısından büyük avantaj oluşturmuştur. Model, çeşitli veri setleri üzerinde test edilerek, başarılı sonuçlar elde edilmiş, önceden eğitilmiş veri setlerinde ise performansı belirgin ölçüde artmıştır [20]. Şekil 1.11'de görü dönüştürücü modeli gösterilmektedir.



Şekil 1.11. Görü dönüştürücü modeli [20].

Tuncel ve arkadaşları (2022), son dönemde NLP ve çeviri uygulamalarında oldukça iyi performans gösteren görü dönüştürücü modelini kullanarak sıtma hastalığının teşhisi için bir çalışma gerçekleştirmişlerdir. Bu yönde literatürdeki diğer ağlarla yapılan çalışmalardaki sonuçlar ile elde edilen sonuçlar karşılaştırılmıştır. Daha yüksek boyutlu veri setlerinde görü dönüştürücü modeli ile elde edilen başarının yükseleceği öngörülmektedir [21]. Şekil 1.12'de çalışmada önerilen model gösterilmektedir.



Şekil 1.12. Önerilen model [21].

Literatürden görüleceği üzere nesne tanıma ve sınıflandırma çalışmaları giderek artmaktadır. Son çalışmalar evrişimli sinir ağları üzerine yoğunlaşmıştır. Ancak yüksek boyutlu görüntülerde evrişimli sinir ağı yapısını kullanmak, hesap gücü ve piksel sayısıyla doğru orantılı bir yapıya sahip olduğundan, belli bir yük getirmektedir. Bu çalışmada genel amaç, yineleyen dikkat modelini etkin bir biçimde kullanarak bu hesaplama yükünden sıyrılma düşüncesi ve alternatif bir model olan görü dönüştürücü modeli ile kıyaslamasını yapmaktır. Yineleyen dikkat modeli, pekiştirmeli öğrenme yöntemi kullanılarak eğitilmektedir. Model, görsel bir yapıyla etkileşen klasik bir pekiştirmeli öğrenme ajanının karar aşamasıdır. Ajan her t adımında band genişliği sınırlı bir sensör yardımıyla çevreyi gözlemleyerek, sensör kaynaklarının dağıtımından sorumlu bir görev edinmektedir. Ajan, çevre tam olarak gözlemlenemediğinden, ajanın çalışma biçimini ve bakış sensörünü verimli bir şekilde nasıl yerleştireceğini tespit etmek için bilgiyi zaman içinde entegre etmelidir. Ajan her basamakta bir ödül almaktadır. Ajanın amacı; bu ödüllerinin toplam değerini en yüksek düzeye çıkarmaktır. Ödüllerin bu maksimum olduğu haller için derin ağ modeli oluşturulmuş ve görüntü sınıflandırma işlemi gerçekleştirilmiştir. Görü dönüştürücü ise RNN'lerin eğitim süresinin yavaşlığı ve kaybolan gradyan probleminin çözümü için alternatif olan dönüştürücü sinir ağlarını kendine odak noktası seçmiş bir modeldir.

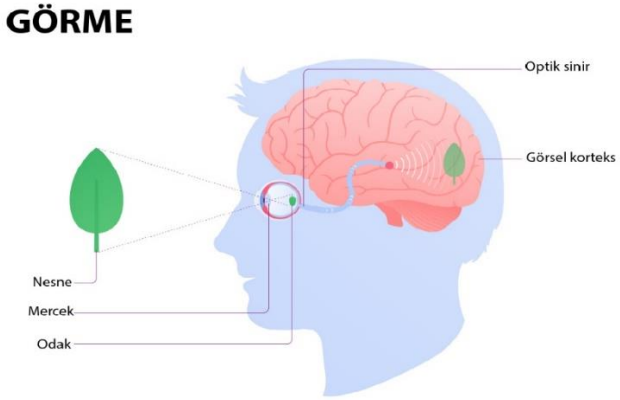
Çalışmada, derin ağ yöntemlerinden yinelemeli sinir ağı, makine öğrenmesi yöntemlerinden pekiştirmeli öğrenme algoritması ve görsel dikkat modeli kullanılarak, görüntü sınıflandırma işleminin gerçekleştirilmesi işlemi yapılmaktadır. Ayrıca son üç yılın sınıflandırma alanındaki en popüler konusu olan görü dönüştürücü (vision transformer) modeli incelenerek, iki model arasında kıyaslama yapılmaktadır. Tez, derin öğrenme ile yineleyen dikkat modelini ve görü dönüştürücü modelini kendisine odak noktası haline getiren görüntü sınıflandırma çalışmasını içermektedir. Tezin devamında "modeller" kelimesi derin öğrenme modelini kast etmektedir.

2. MATERYAL VE YÖNTEM

Bu bölümde irdelenen iki modelin yapısında kullanılan yöntemler ve unsurlar hakkında bilgiler verilmektedir. Yineleyen dikkat modelinde, RNN, CNN, reinforce algoritması gibi yöntemler kullanılırken; optimizasyon algoritması olarak Adam, aktivasyon fonksiyonu olarak ReLU, kayıp fonksiyonu olarak ise hibrit kayıp fonksiyonu kullanılmaktadır. Görü dönüştürücü modelinde; kayıp fonksiyonu olarak çapraz entropi kaybı, aktivasyon fonksiyonu olarak softmax ve GeLU, optimizasyon algoritması olarak ise yine Adam optimizasyon algoritması kullanılmaktadır. Çalışma ortamı ise Pytorch'tur. Ayrıca çalışmada kullanılan hiperparametreler ile ilgili bilgiler de bu başlık altında verilmektedir.

2.1. Görüntü Sınıflandırma

Göz, beş duyu organından birisidir ve aynı zamanda dış çevreyle etkileşim halinde olarak görme olayının gerçekleştirildiği yapıdır. Bir nesnenin görülebilmesi için nesnenin yansıttığı ışık ışınlarının önce göze ve belli aşamalardan geçtikten sonra beyne ulaşması gerekmektedir. Beyne sinir ağı aracılığıyla ulaşan ters görüntü, beyin tarafından düz olarak algılanır ve görme olayı gerçekleşir [22]. Şekil 2.1.'de insan tarafından bir nesnenin görülmesi olayı aktarılmaktadır.



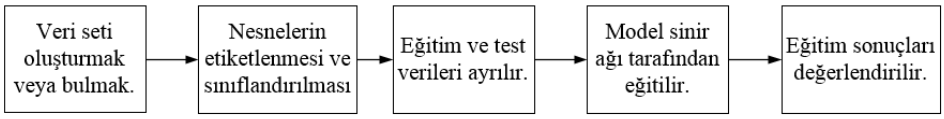
Şekil 2.1. Nesnenin görülmesi olayı [23].

Görüntü işleme alanının popüler konularından olan görüntü sınıflandırma çalışmalarında ise bu görme olayı bilgisayar destekli olarak gerçekleştirilir. Görüntü sınıflandırma çalışmalarında, bir görüntü içindeki özel bir nesnenin varlığı ve konumu irdelenmektedir. Uygulamalarda, yüksek data boyutu ve karmaşıklığı hesap gücü sorunu ortaya çıkarırken; bu sorun günümüzün teknolojisi ile gelişen, güçlü GPU ve CPU'larla aşılmaktadır. Derin öğrenme projelerinde; GPU'ların, CPU'lara göre daha yüksek performans gösterdikleri söylenebilir [24].

Görüntü sınıflandırmada, bir görüntü veya görüntü kümesi arasında belli niteliklere dayalı sınıflandırma çalışması işlemi gerçekleştirilmektedir. Bu alandaki uygulamalara; plaka tanıma, parmak izi tanıma, imza tanıma, tıp ve biyomedikal alanındaki uygulamalar örnek gösterilebilir [25].

2.2. Sınıflandırma Aşamaları

Sınıflandırmada uygulanan genel birkaç adım vardır. Bu adımlardan ilki bir veri seti oluşturmak veya bulmaktır. Daha sonra etiketleme ve sınıflandırma işlemi gerçekleştirilmektedir. Veri seti, eğitim ve test olarak ayrılır. Daha sonra görsel dikkat modelinin eğitilmesi aşamasına geçilir. Eğitilme işleminin tamamlanmasıyla sonuçlar değerlendirilmektedir. Şekil 2.2.'de görüntü sınıflandırma aşamaları gösterilmektedir.



Şekil 2.2. Görüntü sınıflandırma aşamaları.

2.3. Sınıflandırma Alanındaki Gelişmeler

Amerikalı bilim adamları Papert ve Minsky (1966) , bilgisayar aracılığıyla nesnelerin tanınmasını hedefleyen bir çalışma başlatmışlardır ama tüm nesneler için elle kural yazmanın zorluğundan başarıya ulaşılamamıştır. LeCun, hala görüntü sınıflandırma popüler olarak kullanılan CNN'leri tanıtmıştır [26].

Japon bilim adamı Fukushima (1979), görsel örüntü tanıma için bir sinir ağı modeli ortaya atarak, "neocognitron" adını vermiştir [27].

Viola ve Jones (2001), yüz tanıma çalışmalarında sıklıkla kullanılan Viola-Jones modelini ortaya attılar [28].

2.4. Yapay Zeka

İnsanoğlu, beyni aracılığıyla düşünür, muhakeme eder, yaptığı ve yapacağı davranışların getiri ve götürülerini hesap ederek hareket eder. Bu, insan ile diğer canlılar arasındaki en büyük farkı oluşturur.

Geçmişten günümüze çeşitli filozoflar ve bilim adamları zeka kavramı hakkında değişik tanımlamalarda bulunmuştur. Platon, insanın yapısında doğuştan gelen bir yetenek ve bu yeteneğin sonucunda ortaya çıkan bilgi ihtiyacı olarak yorumlarken; Aristoteles ise belirli olaylar üzerinden çıkarım ve yorum yapabilme kabiliyeti olarak tanımlamıştır [29, 30].

İnsanlar, yaptığı mantıklı ve faydalı davranışlar sonucu diğer insanlar tarafından zeki sıfatıyla adlandırılırlar. Yapay zeka ise insanın yaptığı bu mantıklı hareketlerin makineler aracılığıyla gerçekleştirilmesi işlemidir. Bu konuya örnek verilmesi gerekirse; ABD'li büyük teknoloji şirketi IBM tarafından 1952 yılında ilk kez satranç oyunu oynayan bir yazılım geliştirilmiştır. 1996 senesinde ise yine aynı şirket tarafından geliştirilen "Deep Blue" isimli bilgisayar ile dünya satranç şampiyonu ve kimilerine göre satranç tarihinin en büyük oyuncusu olan GM Garry Kasparov arasında 6 oyundan oluşan bir maç gerçekleştirilmiştir. Garry Kasparov bu maçı kazanmıştır ancak 1997 yılında ise geliştirilen Deep Blue ile bir rövanş maçı daha gerçekleştirilmiş ve bu maçı Deep Blue kazanmıştır. Böylece, insan zekasını taklit eden bir bilgisayar, ilk kez bir dünya satranç şampiyonunu yenerek, yapay zeka alanında yapılan çalışmaların önemini bir kez daha kanıtlamıştır [31, 32].

2.5. Makine Öğrenmesi

Bilgisayar aracılığıyla, insan tarzı fikir yürütebilme kabiliyeti olarak adlandırılan yapay zeka için oluşturulan algoritmalar ve uygulamalar makine öğrenmesi olarak adlandırılan bir alt dalı oluşturmaktadır.

Bilgisayar programlama çalışmalarında datalar ve kurallar bilgisayara aktarılarak sonuç istenirken; makine öğrenmesinde bilinen data ve sonuçlardan bilgisayarın kurallar üretmesi beklenmektedir [33]. Makine öğrenmesinin; gözetimli öğrenme, gözetimsiz öğrenme ve pekiştirmeli öğrenme olmak üzere çeşitli türleri mevcuttur. Şekil 2.3'te yapay zekanın, makine öğrenmesinin ve derin öğrenmenin ilişki şeması gösterilmektedir.

Şekil 2.3. Alt dal şeması.

2.5.1. Gözetimli öğrenme

Gözetimli öğrenme uygulamalarında giriş olarak verilen veri ya da veri grubu ve çıkış, uygulayıcı tarafından bilinmektedir ve makinenin bunu öğrenmesi amaçlanmaktadır. Yani girişler bellidir ve çıkışlar etiketlidir. Lineer regresyon ve sınıflandırma çalışmalarında sıklıkla kullanılmaktadır [34].

2.5.2. Gözetimsiz öğrenme

Gözetimsiz öğrenmede ise gözetimli öğrenmenin aksine, etiketlenmiş çıkışlar bulunmaz ve yalnızca giriş olarak bir veri ya da veri grubu makineye verilerek eğitim yapılır. Giriş olarak verilen veri ya da veri grubu, makine aracılığıyla veriler arasında belli özelliklere göre sınıflandırma çalışması yapmaktadır [34].

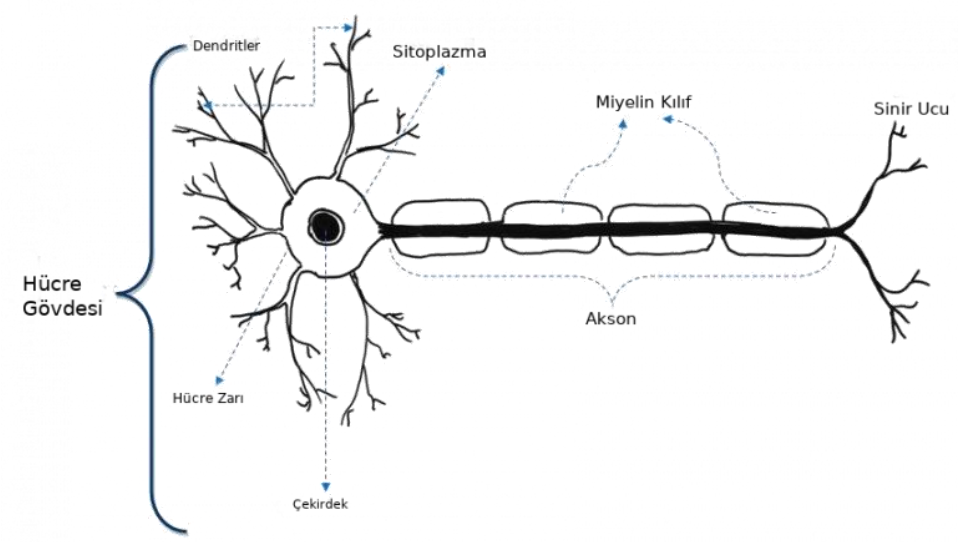
2.6. Derin Öğrenme

Makine öğrenmesi, yapay zekanın bir alt dalıyken, derin öğrenme ise makine öğrenmesinin bir alt dalıdır ve yapay sinir ağlarını temel olarak almaktadır. Derin öğrenme, insan düşünsel yapısı içindeki dil çevirisi ve görüntü sınıflandırma gibi görevleri çok katmanlı sinir ağları aracılığıyla gerçekleştirmektedir. Örnek verilmek gerekirse google diller arası çeviri işleminde derin öğrenme kullanılmaktadır [34, 35, 36].

Derin öğrenme uygulamalarında, makine öğrenmesi uygulamalara göre daha fazla veri kullanılmaktadır. Veri sayısıyla orantılı olarak daha yüksek güçlü makinelere ihtiyaç duyulmaktadır. Çok katmanlı yapısı nedeniyle derin öğrenme uygulamalarında eğitim daha uzun sürmektedir [36].

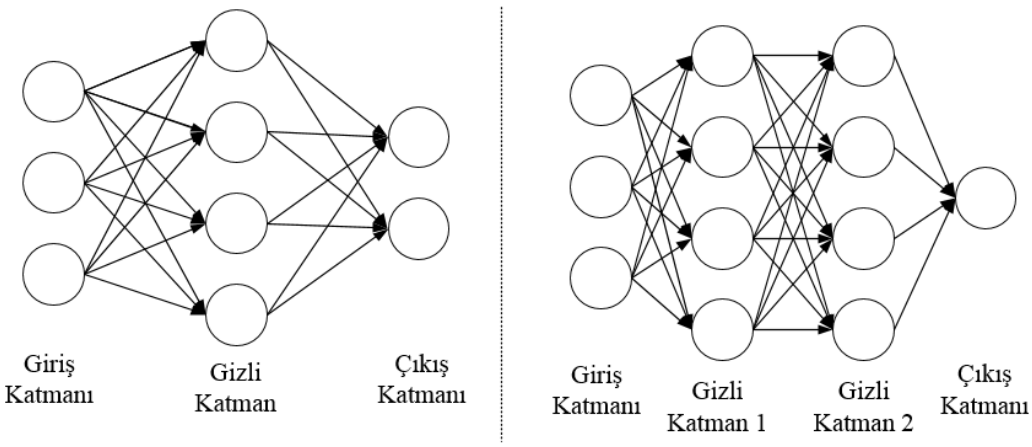
2.6.1. Yapay sinir ağları

İnsan sinir sistemi ve yapay sinir ağları birbirine benzer yapıdadır. İnsanlar davranışlarda bulunurken beynini kullanırken, makineler ise yapay sinir ağlarını kullanarak bu davranışları taklit ederler. İnsan sinir hücrelerinde dendritler giriş bilgilerini alırken;aksonlar ise bu giriş verilerine dayanarak çıkış verilerini oluşturmaktadır. Sinir sistemi aracılığıyla yaşamsal aktiviteler gerçekleştirilmektedir [33, 37]. Şekil 2.4’te insan sinir hücresi (nöronu) yapısı gösterilmektedir.



Şekil 2.4. İnsan sinir hücresi (nöron) yapısı [38].

Derin öğrenmede yapay sinir ağları, insan sinir sistemini kendine örnek alarak, giriş olarak aldığı veriyi çeşitli merhalelerden geçirerek bir çıkış oluşturur. Yapay sinir ağı; giriş, gizli katman, çıkış olmak üzere üç kısımdan oluşurken; tek veya çok katmanlı bir yapıya sahip olabilir [33]. Şekil 2.5'te tek ve çok katmanlı sinir ağı yapısı gösterilmektedir.

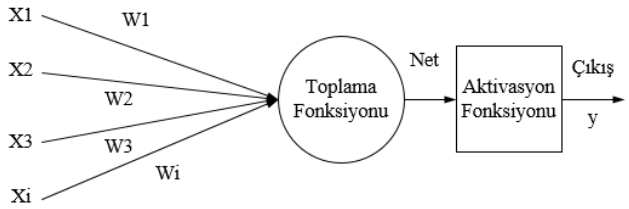


Şekil 2.5. Tek ve çok katmanlı sinir ağı yapıları.

Yapay sinir ağlarında, girdi değerleri belli ağırlıklarla çarpılıp, toplama fonksiyonuyla toplanarak, çıktı katmanına iletilmektedir. Denklem 2.1'de girdiler x olarak gösterilirken, ağırlıklar w ve yanlılık değeri ise b ile temsil edilmektedir [33].

$$y = W * x + b \tag{2.1}$$

Şekil 2.6'da ise temsili yapay sinir hücresi modeli gösterilmektedir.



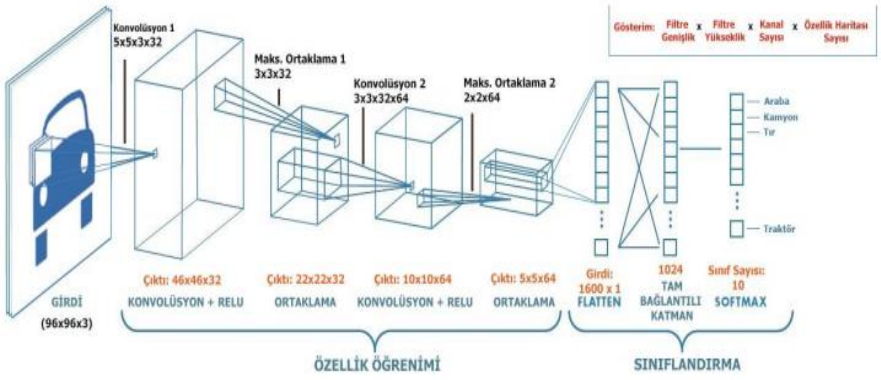
Şekil 2.6. Yapay sinir hücresi temsili.

CNN ve RNN sıklıkla kullanılan sinir ağlarıdır ve aşağıdaki alt başlıklar halinde genel bilgileri verilmektedir.

2.6.1.1. Konvolüsyonel sinir ağları

CNN'ler en popüler YSA'larından biri olup, örüntü tanıma ve sınıflandırma çalışmalarında sıklıkla kullanılmaktadır. CNN modeli genel olarak beş temel unsurdan oluşmaktadır. Giriş katmanı, konvolüsyon katmanı, ortaklama katmanı, tam bağlantılı katman ve çıkış katmanı bu temel unsurlardır [39, 40].

Şekil 2.7'de CNN genel yapısı gösterilmektedir.



Şekil 2.7. CNN genel yapısı [39].

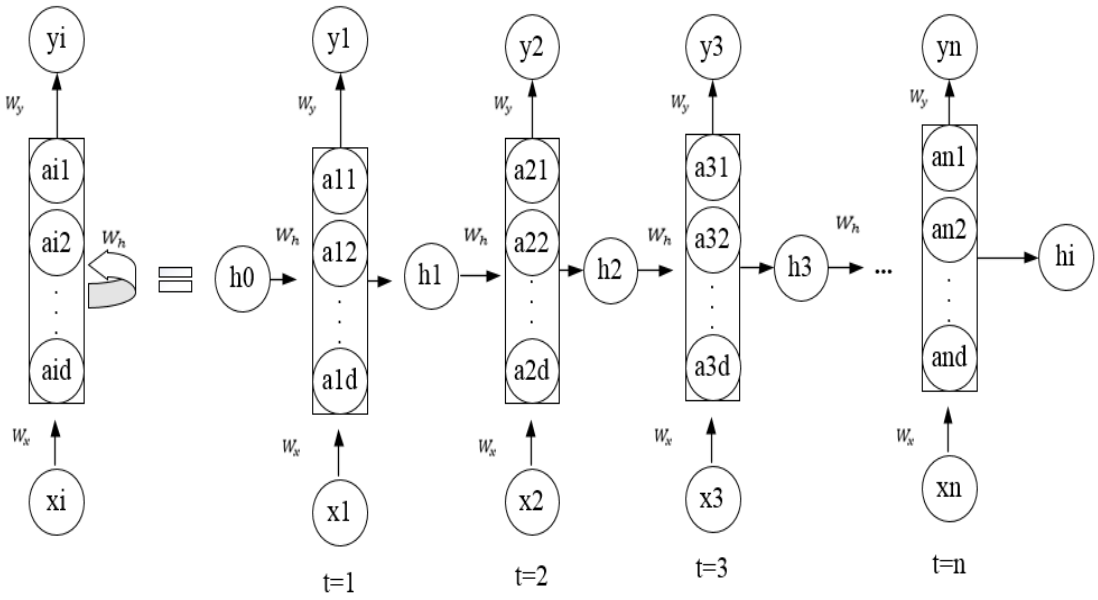
Giriş olarak bir veri ya da veri grubu kullanılır. Konvolüsyon katmanı, giriş olarak alınan verilere evrişimleme uygular ve çeşitli özellikler çıkarır. Ortaklama katmanları, özellik boyutunu azaltmak amacıyla girişlere alt örneklemeler uygulamaktadır.

Tam bağlantılı katmanlar ise evrişimsel katmanların çıkardığı özellikleri kullanarak sınıflandırma işlemi gerçekleştirir ve hata değeri hesaplar [33, 39, 41].

2.6.1.2. Yinelemeli sinir ağları

RNN'ler, dizili veriler ile optimum şekilde çalışmaktadır ve sınıflandırma, metin verileri üzerinde yapılan çalışmalarda sıklıkla kullanılmaktadır. RNN'lerde çıkış yalnızca giriş verilerine bağlı olarak değil, aynı zamanda geçmişten gelen durum bilgileri ile birlikte kullanılmaktadır. Bir önceki çıkış bilgisi, bir sonraki adımın girişi olmaktadır. Yani hafızalı bir sinir ağı olarak tanımlanabilir [42- 45].

Şekil 2.8'de RNN yapısı gösterilmektedir.

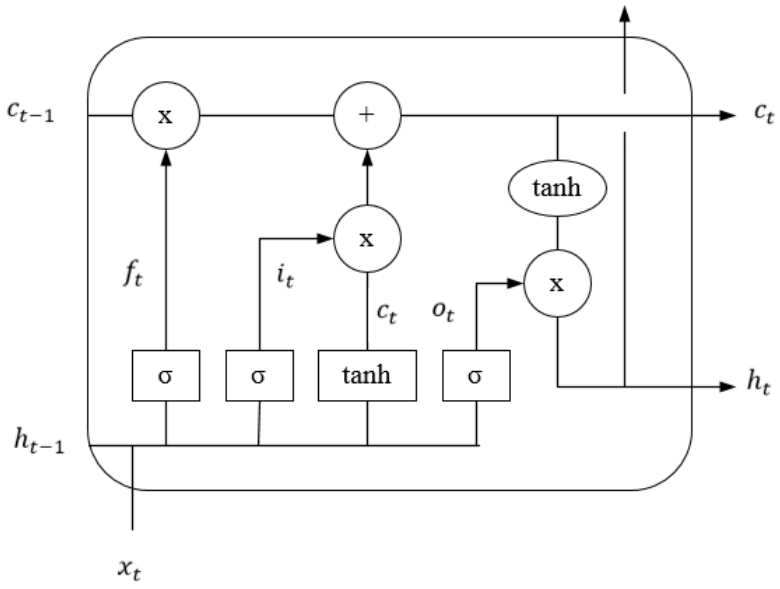


Şekil 2.8. RNN yapısı.

2.6.1.3. Uzun kısa süreli bellek (LSTM)

LSTM'ler, RNN'lerin özel bir türü olarak bilinmektedir. LSTM ağları, RNN'lerde ortaya çıkan gradyan problemleri nedeniyle ortaya çıkmıştır. Bu problem, uzun süren RNN'lerde türevlerin sıfıra yaklaşması olayı olarak açıklanmaktadır. Yapısında hem uzun hemde kısa süreli belleği birlikte tutar. Bu gradyan sorununu, bellek hücrelerini kullanarak aşmaktadır. Bu hücreler karar verme birimidir ve geçmiş verileri saklamaktadır. LSTM'ler biri giriş ve çıkış kapısı olmak üzere, hücre ve unut kapısı ile birlikte toplam dört temel unsurdan oluşmaktadır [44-46].

Bunlardan, giriş kapısı verilerin taşınmasında ve normalizasyonunda, unut kapısı hangi verinin tutulup tutulmayacağının kara verme aşamasında, çıkış kapısı ise tanh fonksiyonu aracılığıyla verilerin seçiminde görevlidir. LSTM'de kapılar, sigmoid aktivasyon fonksiyonu kullanır [44–46]. Şekil 2.9'da LSTM'nin genel yapısı gösterilmektedir



Şekil 2.9. LSTM yapısı.

LSTM denklemleri aşağıda gösterilmektedir. Bu denklemlerde i giriş, o çıkış, f unut kapılarını, c hafıza hücrelerini, h ise çıktı vektörünü temsil etmektedir [47, 48].

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i) \tag{2.2}$$

$$f_t = \sigma\left(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f\right) \tag{2.3}$$

$$c_t = f_t c_{t-1} + i_t \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \tag{2.4}$$

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + w_{co}c_t + b_o) \tag{2.5}$$

$$h_t = o_t \tanh(c_t) \tag{2.6}$$

2.7. Aktivasyon Fonksiyonları

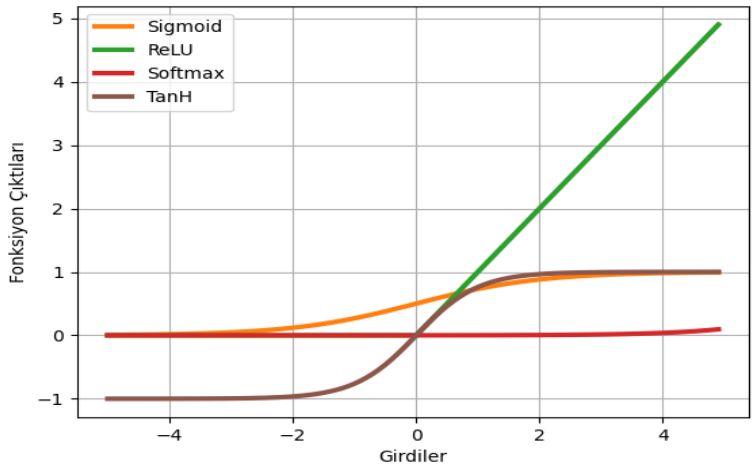
Bu başlık ve alt başlıklar altında her iki modelde kullanılan aktivasyon fonksiyonları tanıtılmıştır. Derin ağ uygulamalarında, bazı parametreler rastgele bir biçimde belirlenerek, sonuçlar değerlendirilip, modele uygun değerler seçilir [33, 49].

Ağırlıklar yenilenerek modelin optimum ağırlık değerleri saptanır. Çok düşük ve yüksek dalgalı değerler hatalı sonuçlara yol açmaktadır. Derin ağlarda, aktivasyon fonksiyonları lineer olmayan dönüşüm süreçleri için tercih edilirler [33, 49]. Tablo 2.1'de çeşitli aktivasyon fonksiyonları gösterilmektedir.

Tablo 2.1. Aktivasyon fonksiyonları.

Fonksiyon	Denklem	Sınır Aralığı
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	(0,1)
ReLU	$f(x) = 0, \quad x < 0$ $f(x) = x, \quad x \geq 0$	[0,∞)
Tanh	$f(x) = \frac{2}{1+e^{-2x}} - 1$	(-1,1)
Softmax	$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$	[0,1]
GeLU	$f(x) = \frac{x}{1+e^{-1.702x}}$	N(0,1)

Şekil 2.10'da aktivasyon fonksiyonlarının değer aralıkları gösterilmektedir.



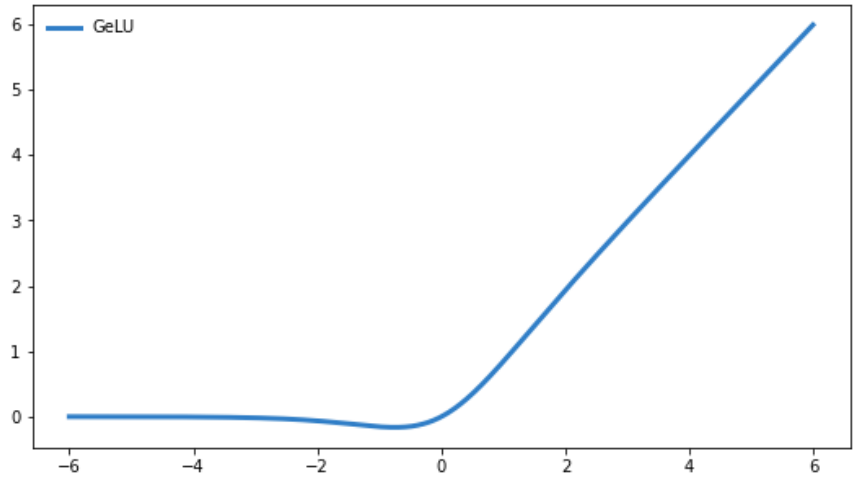
Şekil 2.10. Aktivasyon fonksiyonları değer aralıkları.

2.7.1. ReLU

Çalışmada, ReLU aktivasyon fonksiyonu kullanılmaktadır. ReLU aktivasyon fonksiyonları sıfırdan küçük her x değeri için 0, sıfırdan büyük ve eşit x değeri için x değerini almaktadır. Tablo 2.1 ve şekil 2.10'da relu aktivasyon fonksiyonu ile ilgili bilgiler ve grafik gösterilmektedir [50, 51].

2.7.2. GeLU

Çalışmanın görü dönüştürücü bölümünde GeLU aktivasyon fonksiyonu kullanılmaktadır. GeLU aktivasyon fonksiyonu girdileri 0 ile 1 arasında bir değer ile çarpıp bunları birleştirmektedir. MNIST, CIFAR-10 gibi datasetlerde denenerek diğer aktivasyon fonksiyonları ile karşılaştırıldığında iyi sonuçlar vermektedir [52, 53]. Şekil 2.11'de GeLU aktivasyon foksiyonu grafiği gösterilmektedir.



Şekil 2.11. GeLU aktivasyon fonksiyonu.

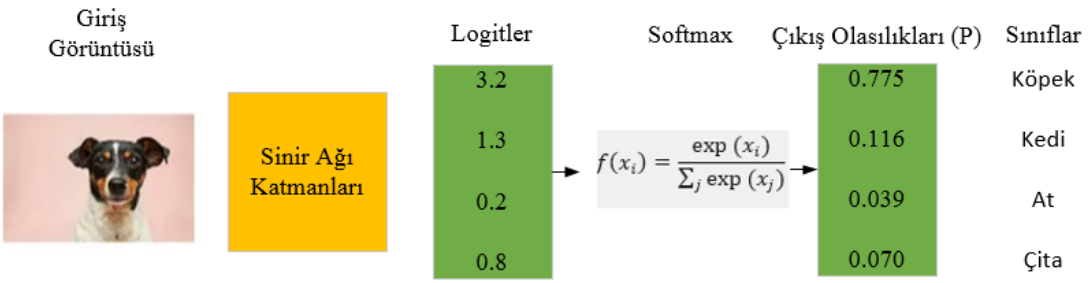
2.7.3. Softmax

Çoklu sınıfsal bir uygulama gerçekleştirilirken, kullanılan modelin çıktısının her bir sınıfının olasılıksal bir değeri vardır. Yani uygulamayı gerçekleştiren kişinin elinde ne kadar çok sınıf varsa, o kadar çok buna bağlı çıktı olacaktır. Softmax aktivasyon fonksiyonu bu gibi durumlarda kullanılmaktadır. Bu aktivasyon fonksiyonunda, olasılık [0,1] aralığındadır ve özellikler modele vektör halinde verilmektedir. Bu vektörü sınıf sayısı belirler ve yalnızca tek bir elemanı 1 değerine sahiptir. Örnek vermek gerekirse; bir çantanın, kahverengi, siyah, gri ve yeşil renkleri olduğu düşünülürse; burada dört sınıf olduğundan vektör uzunluk değeri de dört olacaktır.

Bu vektörlerde temsil edilen sınıf 1 olarak alınırken; bunun haricindekiler ise 0 olarak alınmaktadır. Buna göre kahverengi, siyah, gri ve yeşil renk sınıflarının vektörleri sırasıyla şu şekilde olmaktadır: (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1). Bu uygulama one hot encoding olarak adlandırılır ve denklem 2.7'de gösterilmektedir [54, 55].

$$y\epsilon\{(1,0,0,0),(0,1,0,0),(0,0,1,0),(0,0,0,1)\} \quad (2.7)$$

Örnek bir tahmin örneği vermek gerekirse; y^1, y^2, y^3, y^4 çıktılarının olasılıkları sırasıyla 0.1, 0.8, 0.1, 0.1 olduğu varsayıldığında, çanta siyah renk sınıfı olarak tahmin edilmektedir. Softmax aktivasyon fonksiyonunda, iki şarttan ilk olarak logitlerin negatif olmamasını sağlamak için her bir logitin üsseli alınmakta; daha sonrasında ise toplamlarının bir olması için toplamlarına bölümü gerçekleştirilmektedir [54, 55]. Yukarıda verilen örneğin bir başka hali görsel olarak şekil 2.12'de gösterilmektedir.



Şekil 2.12. Softmax fonksiyonun nasıl sınıflandırma yaptığına örnek görüntü [56].

2.8. Kayıp Ve Amaç Fonksiyonu

Bu başlık ve alt başlıklarda her iki modelde kullanılan kayıp fonksiyonları hakkında bilgiler verilmektedir.

Derin öğrenme uygulamalarında, ilk olarak bir kayıp fonksiyonu tanımlanmaktadır ve kayıpları minimize etmek amacıyla bir optimizasyon algoritması kullanılmaktadır. Bu kayıp fonksiyonu, optimizasyonun amaç fonksiyonu olarak isimlendirilmektedir. Burada temel amaç hatayı en aza indirmektir. En aza indirmek istenilen bu kayıp değerleri, optimizasyonda amaç fonksiyonu haline dönüşmektedir. Kayıp fonksiyonu seçerken spesifik olarak uygulamaya göre karar verilmektedir. Çoklu sınıflandırma uygulamalarında, genellikle çapraz entropi kaybı kullanılmaktadır [43, 57].

Kayıp fonksiyonu, eğitim sonucu ile ümit edilen çıktının ne oranla ayrıştığını belirten hata değerinin hesabı amacıyla kullanılmaktadır. Çeşitli sınıflandırma ve regresyon problemlerinde kullanılan çapraz entropi kaybı, ortalama kare hata kaybı, sigmoid kaybı, sotmax kaybı gibi sınıfsal örnekleri mevcuttur. Kayıp fonksiyonunun eğitim gerçekleştirilirken önemi büyüktür. Çünkü parametreler güncellenirken; bu fonksiyon gradyanı kullanılır ve bu yüzden eğitim başarısında belirleyici bir faktördür [58].

Yineleyen dikkat modelinde, hibrit kayıp fonkiyonu kullanılmaktadır ve bu diğer kayıp fonksiyonlarının karışım şeklindeki halidir. Bir diğer model olan görü dönüştürücüde ise yine çoklu sınıflandırma uygulamalarında sıklıkla tercih edilen çapraz entropi kaybı kullanılmaktadır.

2.8.1. Çapraz entropi kaybı

Çoklu sınıflandırma uygulamalarında sıklıkla kullanılan ve aynı sınıf elemanlarını belli bir uzay kümesinde birleştirmeyi amaç edinen kategorik çapraz entropi kaybı [59], log softmax ile negative log likelihood fonksiyonlarından türetilerek meydana getirilmiştir [58].

Genelde softmax aktivasyon fonksiyonun peşinden kullanılmaktadır ve bundan dolayıda softmax kaybı olarakta isimlendirilmektedir [60]. Denklem 2.8'de çapraz entropi kaybı gösterilmektedir.

$$Çapraz\ Entropi = -\sum_{j}^{c} y_j log(p_j) \tag{2.8}$$

Denklemde y_j reel değerleri, p_j ise tahmini değerleri sembolize etmektedir. Datasetin tümü için kayıp değerleri hesaplanmaktadır ve daha sonrasında ise ortalama alınarak ilgilenilen modelin dataset üstündeki kayıp değerleri hesap edilerek işlem gerçekleştirilir [60].

Denklem 2.9 bunu formülize eder. Denklem de m harfi örnek sayısını sembolize etmektedir [60].

$$Çapraz\ Entropi = -\frac{1}{m}\sum_{i}^{m}\sum_{j}^{c} y_j log(p_j) \tag{2.9}$$

2.8.2. Hibrit kayıp fonksiyonu

Hibrit kayıp fonksiyonları, diğer kayıp fonksiyonlarının karışımı şeklindeki halidir [58]. Yineleyen dikkat modelinde, ajanın gerçekleştirdiği hangi eylemin iyi hangi eylemin kötü olduğunun bilinmesi, toplam ödülle ilişkilidir. Bazı hallerde ise, yapılacak doğru aksiyon bilinebilir. Örnek vermek gerekirse, bir nesne algılama uygulamasında ajan, son aksiyon olarak nesnenin sahip olduğu etiketi çıkartmalıdır. Doğru etiketi, nesne ile ilişkilendirmek için genellikle ajan politikası optimizasyon yöntemleri ile optimize edilmektedir. Yineleyen dikkat modelinde, aksiyon ağı yani $f_a(\theta_a)$'yı eğitmek ve gradyanları geri yaymak amacıyla kayıp fonksiyonlarından çapraz entropi kaybı kullanılırken; konum ağı $f_l(\theta_l)$ ise reinforce algoritması ile eğitilmektedir [15]. Denklem 2.10 hibrit kayıp fonksiyonu denklemidir.

$$Hibrit\ Kayıp = loss_{aksiyon} + loss_{baseline} + loss_{reinforce} \tag{2.10}$$

2.9. Adam Optimizasyon Algoritması

Her iki modelde de adam optimizasyon algoritması kullanılmaktadır. Adam optimizasyon algoritması; hızı, rahat uygulanması ve düşük bellek ihtiyacı özellikleri ile derin öğrenme uygulamalarında sıklıkla kullanılmaktadır. RmsProb ve momentum optimizasyon algoritmalarının bir birleşimi olarak öğrenme oranının aşırı küçülme probleminin ortadan kaldırılması için ortaya atılmıştır [61, 62].

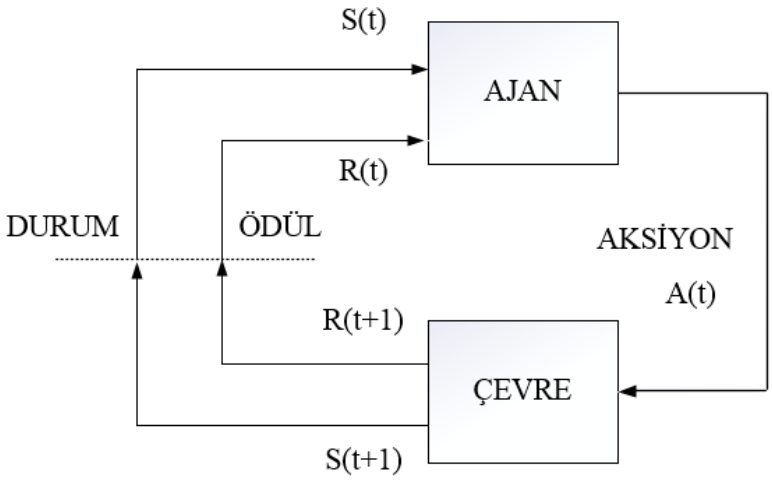
2.10. Pekiştirmeli (Takviyeli) Öğrenme

Pekiştirmeli öğrenme, çevreyle deneme yanılma yoluyla etkileşime girerek ve benzersiz geri bildirim olarak ödüller (olumlu veya olumsuz) alarak, çevrede; öğrenen aracılar (yapay zekâ) oluşturarak kontrol görevleri çözmek için kullanılan bir yöntemdir. Örneğin, hiç oyun oynamayan bir kişinin önüne oyun kumandası koyduğunuzu düşünün ve onu yalnız bırakın. Yön tuşlarından birine basarak, bir sanal para ödülü alsın. Bu artı bir ödüldür, dolayısıyla bu oyunda sanal paraların alınması gerektiğini öğrenmiştir. Ancak tekrar sağ tuşa bastığında bu sefer bir düşmana dokunabilir ve ölebilir, bunun sonucunda ise eksi bir ödül alır. Artık düşmana değmemesi gerektiğini öğrenmiştir. Dolayısıyla, oyuncu deneme yanılma yoluyla etkileşime girerek, bu ortamda para kazanmasını fakat düşmanlardan kaçınması

gerektiğini anlamıştır. Herhangi bir gözetim olmadan oyuncu devamlı oyunu oynarsa daha iyi hale gelebilir. İşte pekiştirmeli (takviyeli) öğrenme, aksiyonlardan öğrenmenin hesaplı bir yaklaşımıdır [63].

Takviyeli öğrenmede çevre, ajan, ödül ve aksiyon kavramları bulunmaktadır. Önceki örnekte çevre oyunun kendisi, ajan (oyuncu) yani karar verici, aksiyon ise oyuncunun bir aksiyon alması yani bir tuşa basması olarak düşünülebilir [63].

Takviyeli öğrenmede, bir ajan (beyin) karşısına gelen durumlara (S_t) kendi aksiyon (A_t) ve tecrübelerine dayanarak cevap vererek, bunun sonucunda bir ödül (R_t) almaktadır. Ajan aldığı ödülleri en yüksek seviyeye çıkarmaya çalışır. Takviyeli öğrenme sistemi genel olarak dört parçadan oluşurken, bunlardan birisi tercihe bağlıdır. Bunlardan ilki politikadır ve ajanın bulunduğu durumdaki aksiyonu seçerek, bu durum karşısında bir tepki vermesi olayıdır. Bir diğeri ise ödüldür. Ajan, gerçekleştirdiği aksiyonlar üzerinden ödül olarak bir puan alarak, bu ödülleri maksimum düzeye çıkarmaya çalışmaktadır. Ajan, düşük ödül aldığı durumlarda izlediği politikayı değiştirerek, bir dahaki seferde farklı bir aksiyon alarak ödül seviyesini yukarı çekmek istemektedir. Durum değeri ise uzun vadede ajanın aksiyonlar sonucu alacağı ödüllerin toplamını ifade ederek, hangi durumun iyi hangisinin kötü olduğunu göstermektedir [34, 64, 65]. Şekil 2.13'te pekiştirmeli öğrenmenin genel yapısı gösterilmektedir.

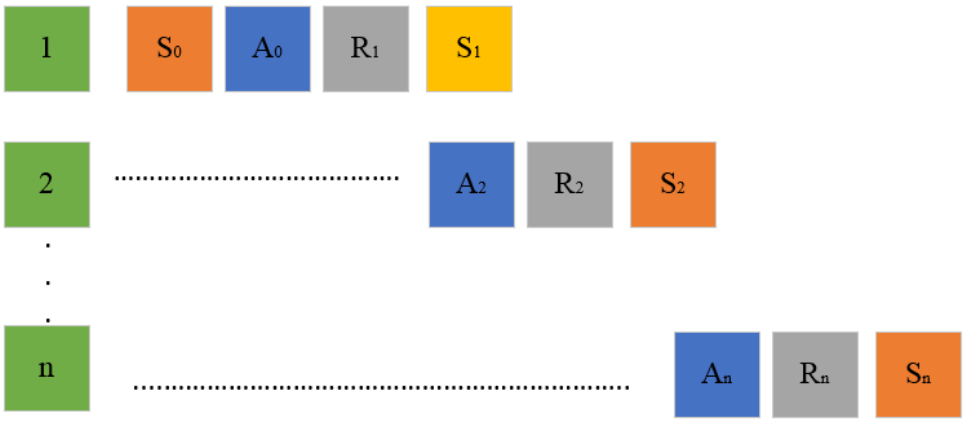


Şekil 2.13. Pekiştirmeli öğrenme yapısı.

Takviyeli öğrenmede; yukarıdaki verilen oyun-oyuncu (ajan) örneği düşünülecek olursa;

- Ajan oyundaki bir durumu (görüntü olsun) S_0 çevreden alır.
- S_0 durumu için, ajan A_0 aksiyonunu gerçekleştirir.
- Çevre bu duruma karşılık S_1 durumunu alır.

Dolayısıyla şöyle bir döngü bulunmaktadır ve şekil 2.14'te gösterilmektedir.



Şekil 2.14. Pekiştirmeli öğrenme döngüsü [63].

Ajanın amacı beklenen ödülü (toplam ödülü) maksimum yapmaya çalışmaktır. Yani takviyeli öğrenme beklenen ödülün maksimize edilebilmesi hipotezine dayanmaktadır [63].

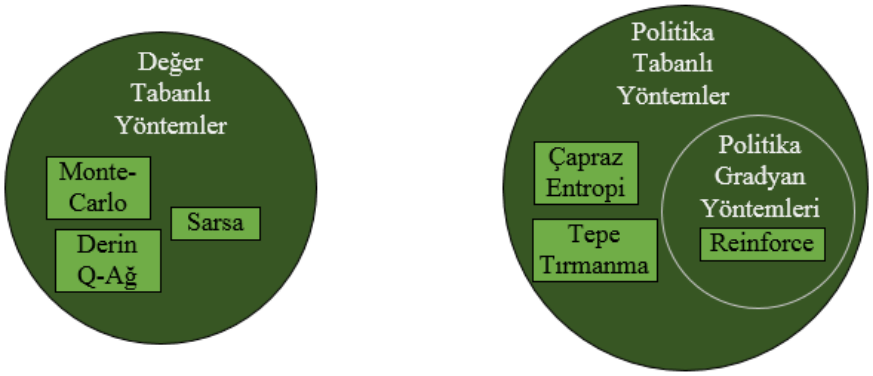
2.11. Politika-Gradyan Yöntemleri

Takviyeli öğrenmedeki en büyük sorun, toplam ödülü en üst düzeye çıkaran eylemleri seçebilen bir ajan nasıl oluşturulur problemidir. İşte burada politika π dediğimiz ajanın beyni devreye girmektedir. Politika π, bize içinde bulunduğumuz duruma göre hangi eylemi yapacağımızı söyleyen bir fonksiyondur. Dolayısıyla, ajanın belirli bir andaki davranışını politika π tanımlar. Bu politika öğrenmek istediğimiz bir fonksiyondur. Amaç; optimal politikayı bulmaktır. Ajan buna göre hareket ettiğinde beklenen getiriyi maksimize eden politikayı elde eder. Optimal politika, eğitim yoluyla elde edilir. Bunun için iki yaklaşım vardır [63].

- Doğrudan, ajanın bir durum için hangi aksiyonu gerçekleştirmesi gerektiğini öğreten politika-tabanlı yöntemler.
- Dolaylı, ajana hangi durumun daha değerli olduğu öğretilen ve daha değerli durumlara yol açan eylemler gerçekleştirilen değer tabanlı yöntemler.

Yineleyen dikkat modelinde kullanılan reinforce algoritması bir politika gradyan yöntemidir. Bu bölümdeki başlık ve alt başlıkta politika-gradyan yöntemleri ve reinforce algoritması hakkında bilgiler verilmektedir.

Politika gradyan yöntemleri, gradyan artışıyla en uygun prensibin ağırlık tahmini gerçekleştiren politika-tabanlı yöntemlerin bir alt dalıdır [66]. Şekil 2.15'de değer tabanlı ve politika tabanlı yöntemler gösterilmektedir.



Şekil 2.15. Değer tabanlı ve politika tabanlı yöntemler.

Politika-gradyanlarının ana düşüncesi, olumlu aksiyonları arttırmak, en uygun politikaya varana dek daha büyük ödül getiren aksiyonların arttırılması ve buna karşın daha az ödül getiren aksiyonlarının azaltılması olarak açıklanabilir. Bu olay, politika ağının ağırlıklarının güncellenmesiyle sağlanmaktadır. Pekiştirme algoritması, işte bu politika gradyan yönteminin bir çeşit türüdür [66].

2.11.1. Reinforce algoritması

"Reinforce algoritması", takviyeli öğrenmenin uygulandığı bir ağdan meydana gelmektedir. Algoritmada, r pekiştirme değerini (ödülü) ifade etmektedir ve bu değerin alınmasının ardından ağırlıklar güncellenir [67].

Algoritmada, w_{ij} ağın ağırlık değeri, a_{ij} öğrenme oranı faktörü, b_{ij} pekiştirme taban çizgisi (baseline), e_{ij} ise karakteristik uygunluk olarak temsil edilmektedir. Denklem 2.11 bu ifadeyi temsil etmektedir [67].

$$\Delta_{w_{ij}} = a_{ij}(r - b_{ij})e_{ij} \tag{2.11}$$

Öğrenme oranı faktörü a_{ij}, negatif değildir ve t'ye bağlıdır. Bu özel yapıdaki öğrenme tabanlı algoritma reinforce algoritması olarak adlandırılmaktadır. Bu adlandırılma, "Ödül artışı=Negatif olmayan faktör çarpı pekiştirme dengesi çarpı karakteristik uygunluk" ifadesinden kısaltılarak oluşturulmuştur [67].

Yörünge (yol), bir durumdan son ana gelene kadar alınan durum ve o durumda alınan aksiyonların sırasını ifade eder. Yörünge, bir durum-aksiyon-ödül üçlüsünden meydana gelir. Uzunluk olarak bir sınırlaması yoktur. Uzunluk H ile yörünge ise τ ile ifade edilmektedir. [66]. Denklem 2.12'de yörünge ifade edilmektedir.

$$\tau = (s_0, a_0, r_1, s_1, a_1, r_2, s_{2,\ldots} \alpha_H, r_{H+1}, s_{H+1}) \tag{2.12}$$

Reinforce yöntemi, yörüngeler üzerinden ödülü maksimuma çıkarmak için optimal politikayı aramaktadır [66]. τ yörüngesinin ödülü, denklem 2.13'te görüldüğü gibi R(τ) ile ifade edilir.

$$R(\tau) = (G_0, G_1, \ldots G_H) \tag{2.13}$$

$G_{(k)}$ parametresi, k geçişi için her k zaman adımından (s_k, a_k, r_{k+1}) yörüngenin sonuna dek beklenen toplam ödülü ifade eder [66]. Denklem 2.14'te G_k parametresi gösterilmektedir.

$$G_k \leftarrow \sum_{t=k+1}^{H+1} \gamma^{t-k-1} R_k \tag{2.14}$$

Reinforce algoritmasının genel amacı; $U(\theta)$ fonksiyonu ile gösterilen beklenen getiriyi maksimum duruma çıkaran θ ağırlıklarını elde etmektir [66]. Denklem 2.15'te maksimum ödülü getirecek θ ağırlıklarının olasılıksal formalize edilmiş hali gösterilmektedir.

$$U(\theta) = \sum_T P(\tau, \theta) R(\tau) \tag{2.15}$$

$U(\theta)$ fonksiyonunu (beklenen getiriyi olasılıksal olarak belirtir) maksimum düzeye çıkaran θ ağırlık değerini belirlemenin yöntemlerinden biri gradyan yükselişidir [66]. Formülasyon olarak, gradyan yükselişi için yenileme adımı denklem 2.16'daki gibi verilebilmektedir.

$$\theta \leftarrow \theta + \alpha \nabla_\theta U(\theta) \tag{2.16}$$

Burada α, genellikle zaman içerisinde azalan adım boyutunu ifade etmektedir. Bu yöntemi kullanabimek için; ∇_θ gradyanın tam değerini hesaplamak güçlükler içerdiğinden (hesaplama açısından çok pahalı olduğundan) yörüngeleri örnekleyerek, bu yörüngeler aracılığıyla gradyan tahmini yapılmaktadır. [66]

Bu örnekleme, Monte carlo yöntemi aracılığıyla olur ve bundan dolayı reinforce yöntemi literatürde Monte carlo politika gradyanları olarak da adlandırılmaktadır. Yöntem davranışını daha iyi anlamak için pseudo code (sözde kod) şekil 2.16'daki gibi verilebilir [66].

Input:a differentiable policy parameterization $\pi_\theta(a_t|s_t, \theta)$

Algorithm parameter: step size $a > 0$

Initiliaze the policy parameter θ at random

(1) Use the policy π_θ to collect a trajectory $\tau = (s_0, a_0, r_1, s_1, a_1, r_2, s_{2,\dots,} a_H, r_{H+1}, s_{H+1})$

(2) Estimate the Return for trajectory τ: $R(\tau) = (G_0, G_1, \dots G_H)$

where G_k is the expected return for transition k:

$$G_k \leftarrow \sum_{t=k+1}^{H+1} \gamma^{t-k-1} R_k$$

(3) Use the trajectory τ to estimate the gradient $\nabla_\theta U(\theta)$

$$\nabla_\theta U(\theta) \leftarrow \sum_{t=0}^{H} \nabla_\theta log \pi_\theta(a_t|s_t) G_t$$

(4) Update the weights θ of the policy

$$\theta \leftarrow \theta + \alpha \nabla_\theta U(\theta)$$

(5) Loop over steps 1-5 until not converged

Şekil 2.16. Sözde kod

Denklem 2.17'de ajanın t zaman adımında S_t durumundan aksiyonu seçme olasılığını değerlendirir. Burada π, θ yani ağırlıklar tarafından parametreleştirilen politikayı ifade etmektedir [66].

$$\pi_\theta(a_t|s_t) \tag{2.17}$$

Bu ifadenin tam şekli ise logaritmik olarak olasılığının alınmş halidir. Denlem 2.18 bunu temsil eder [66].

$$\nabla_\theta log\pi_\theta(a_t|s_t) \tag{2.18}$$

Denklem 2.18 aynı zamanda bize bir S_t durumunda aksiyon almanın logaritmik olasılığını yükseltmek için θ tarafından parametreleştirilen politikadaki ağırlıkları nasıl değiştirmemiz gerektiğini belirtmektedir [66].

2.11.2. Ödül ve taban çizgisi (baseline)

Reinforce algoritmasında, ana hedef en iyi ve makul politikaya ulaşana dek, alınan aksiyonlar sonucu elde edilen ödül miktarını arttırmaktır. θ parametresi ise algoritmada derin ağın başarımını belirten ağırlıklardır. Bu ağırlıklar her güncellendiğinde yeni bir ödül çıkarılmaktadır. Algoritmada modelin ne kadar iyi performans gösterdiğini belli eden bir taban çizgisi kullanılarak; yüksek değişken getirilerden kaynaklı yüksek varyans probleminin önüne geçilmek istenmiştir [66, 67]. Çünkü; yüksek varyans, kararsız bir yapıya ve başarısız sonuçlara neden olabilir. Kullanılan taban çizgisi varyansta azalmaya neden olarak, modelin daha hızlı bir biçimde tepki vermesine yardımcı olur [66, 67].

2.11.3. Monte carlo yöntemi

Monte carlo yöntemi, pekiştirmeli öğrenmede bir politikanın (π) değer fonksiyonunun tahmini konusunda kullanılmaktadır. Monte carlo yönteminde geçmişten gelen tecrübeler etkilidir. Bu yöntemde sadece alınan aksiyon ve içinde bulunan durum için bir değer verilmektedir. Ajan, aksiyon kümesi içerisinden rassal bir seçim yaparak, daha iyi bir politika belirlemek için geçmişten gelen tecrübelerden faydalanmaktadır [68].

2.12. PyTorch

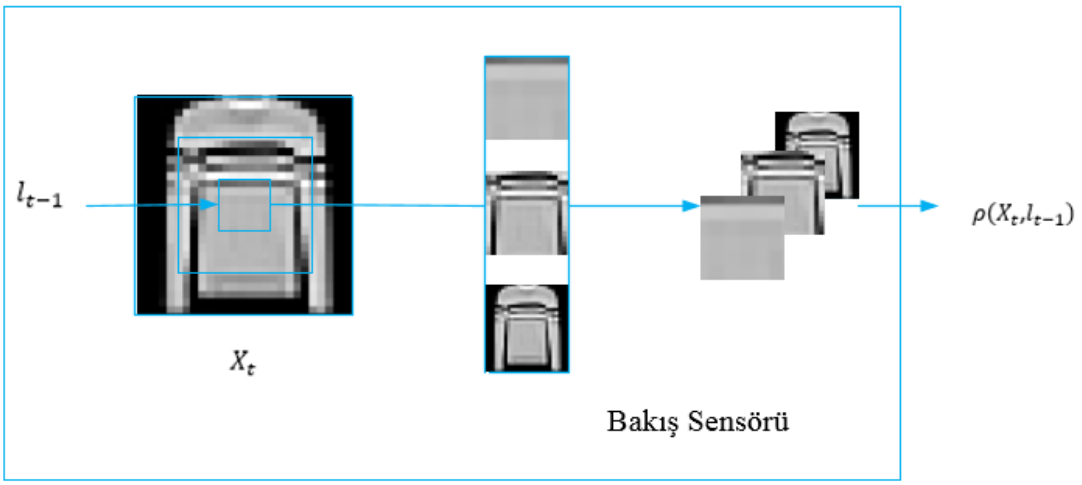
GPU'ları kullanarak derin öğrenme uygulamalarının gerçekleştirilmesinde sıklıkla kullanılan bir Python kütüphanesi olup, kullanımı her geçen gün artmaktadır. Derin ağ modelleri oluştururken;basit oluşu, dinamik hesaplamalı grafikler oluşturabilmesi, Python'ca sunulan tüm işlevlerden yararlanabilmesi açısından tercih sebebi olarak görülmektedir [69, 70].

3. YİNELEYEN DİKKAT MODELİ (RAM)

Yüksek boyutlu görüntülere CNN'ler ile yaklaşmak hesaplama açısından sıkıntı yaratmaktadır. Bunun yerine; Mnih ve arkadaşları (2014), çalışmalarında bu soruna karşı, sadece ilgilenilen, belirli bir bölüme odaklanan ve bu bölgede işlem yapan bir RNN modeli sunmaktadırlar [15].

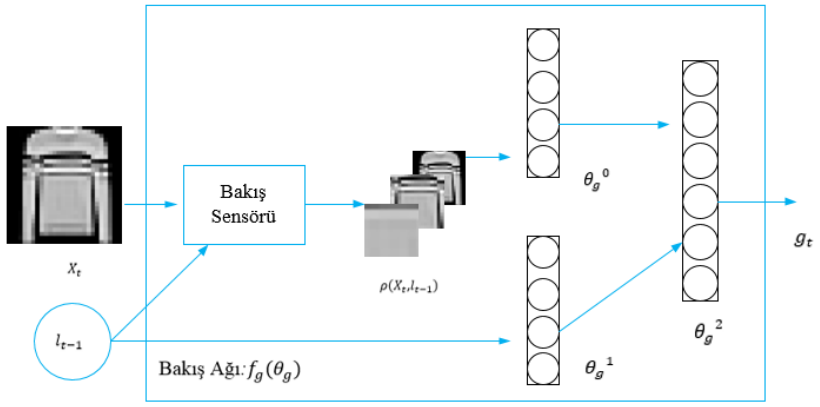
İnsanlar, bir ortama girdiklerinde ortamın tamamına odaklanmak yerine ihtiyaç ve gereksinimlerine göre spesifik bir bölgeye odaklanırlar. Görüntü işleme uygulamalarında da bir görüntünün tamamına değil, sadece ilgilenilen bölgeye göre işlem yapmak hesaplama açışından kolaylıklar sağlamaktadır. İlgilenilen piksel sayısı azaldıkça, hem zamandan tasarruf sağlanmakta hem de sorun basitleşmektedir. RAM'de, görüntünün sadece belli bir bölgesiyle ilgilenilir ve bu bölge geçmişten gelen bilgiler ve deneyimlere göre belirlenmektedir [15].

RAM'de bir ajan işlem yapacağı ortamı bant genişliği limitli bir sensör ile izler ve sadece hedeflenen bölgeyle ilgilenir. Şekil 3.1'de gösterilen bakış sensöründe, X_t giriş görüntüsüdür ve l_{t-1} merkezli retina temsili $\rho(X_t, l_{t-1})$ görüntü parçaları elde edilir [15].



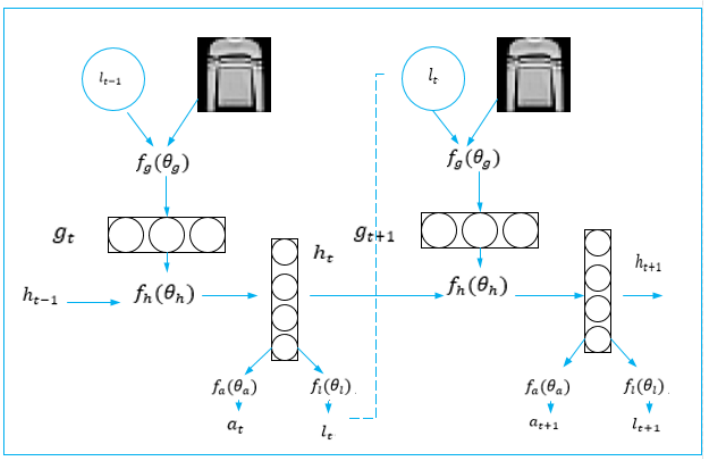
Şekil 3.1. Bakış sensörü.

Temsili retina $\rho(X_t, l_{t-1})$ oluştuktan sonra anlık konum ve bu temsili retina, $\theta_g{}^0$ ve $\theta_g{}^1$ bileşenlerinden gelen verileri birleştirmek adına diğer bir doğrusal katman $\theta_g{}^2$'yi kullanarak gizli bir alana eşler ve g_t temsili bakış oluşturulur. Bakış ağı, model için temsili bakış g_t'yi oluşturmaktadır. Şekil 3.2'de bakış ağı gösterilmektedir [15].



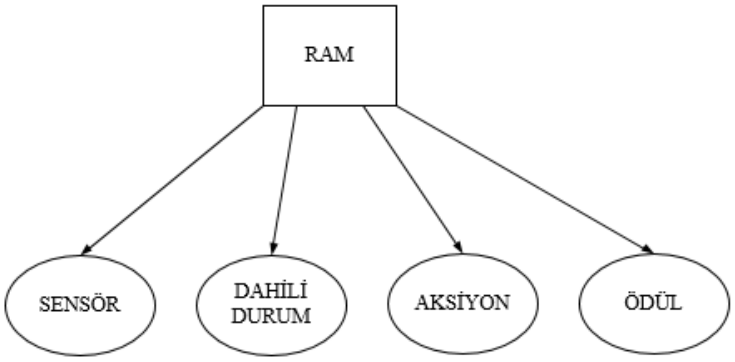
Şekil 3.2. Bakış ağı.

Model, yineleyen bir sinir ağından meydana gelmektedir. Temsili bakış g_t'yi giriş olarak alarak, h_{t-1} dahili temsil ile bir bütün halinde modelin yeni dahili durumu h_t'yi oluşturmaktadır. $f_l(\theta_l)$ konum ağı, $f_a(\theta_a)$ ise aksiyon ağıdır. Bunlar l_t'ye ve a_t'ye katılarak bir sonraki konumu ve aksiyonu oluşturmak adına h_t'yi yani dahili durumu kullanmaktadır. Bu aşamalar belirlenen zaman aralığında tekrarlanmaktadır [15]. Şekil 3.3'te model mimarisi gösterilmektedir.



Şekil 3.3. Model.

RAM, temel olarak dört temel unsur üzerine inşa edilmiştir. Bunlar; sensör, dahili durum, alınan aksiyon sonucu elde edilen ödüldür. Şekil 3.4'te RAM'nin temel unsurları gösterilmektedir [15].



Şekil 3.4. RAM temel unsurları.

Her zaman adımında sensör, X_t giriş görüntüsünden belli bölgelere odaklanarak sınırlı bir alandan bilgi almaktadır. Sensör, şekil 3.1'de gösterilen l merkezli bir alanı ele alarak, X_t giriş görüntüsünden daha az boyutlu bir vektörün elde edilmesini sağlamaktadır.

Bu, merkezi l bölgesine yakın yerlerin yüksek, uzak yerlerin ise düşük çözünürlükte kullanılması anlamını taşımaktadır. Ajan, geçmişte aldığı aksiyonları ve sonuçlarını içeren dahili durumu koruyarak, yeni adımda nasıl bir aksiyon alınacağına karar vermektedir. Ajan, her adımda almakta olduğu ödülleri yükseltmeyi amaçlar. Sınıflandırma aşamasında görüntünün doğru sınıflandırıldığı durumlarda ödül 1, diğer halde ise 0 olmaktadır [15]. Denklem 3.1 ödülleri sembolize etmektedir.

$$R = \sum_{t=1}^{T} r_t \tag{3.1}$$

RNN'ler uzun vadede ortaya çıkan gradyan problemlerine sahiptir. Denklem 3.2'de gradyan yaklaşımı gösterilmektedir [71].

$$\nabla_{\theta} J = \frac{1}{M} \sum_{i=1}^{M} \sum_{t=1}^{T} \nabla_{\theta} \log \pi\left({u_t}^i \middle| {s_{1:t}}^i ; \theta\right) R^i \tag{3.2}$$

Denklem 3.2 literatürde pekiştirme veya öğrenme kuralı olarak adlandırılmaktadır. Modelde, yüksek ödül alan aksiyonlar arttırılırken, aksi aksiyonlar azaltılır. Buna ek olarak, bir problemle karşılaşma durumu ortaya çıkar. Bu ödüllerin toplamı artarken sonsuzluğa yaklaşma problemidir. Bu sıkıntının bertaraf edilmesi için indirim faktörü (γ) kullanılır. Burada ödül toplamı 0 ile 1 arasında olan indirim faktörü ile her adımdan sonra daha da küçültülür. İndirimli ödül faktörü, pekiştirmeli öğrenme yönteminde iade olarak da adlandırılmaktadır [15, 72]. Denklem 3.3'te indirim faktörü denklemi gösterilmektedir.

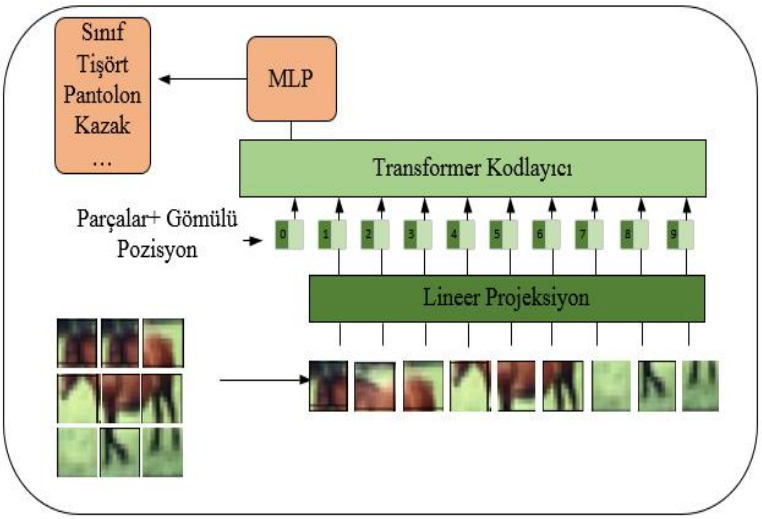
$$R = \sum_{t=1}^{T} \gamma^{t-1} r_t \, , T \to \infty \tag{3.3}$$

Ödülün, denklem 3.2'de yerine konulması ile maksimum ödülü sağlayan θ ağ parametreleri belirlenir. Bu şekilde optimum politika oluşturulmuş olur.

4. GÖRÜ DÖNÜŞTÜRÜCÜ (VİT)

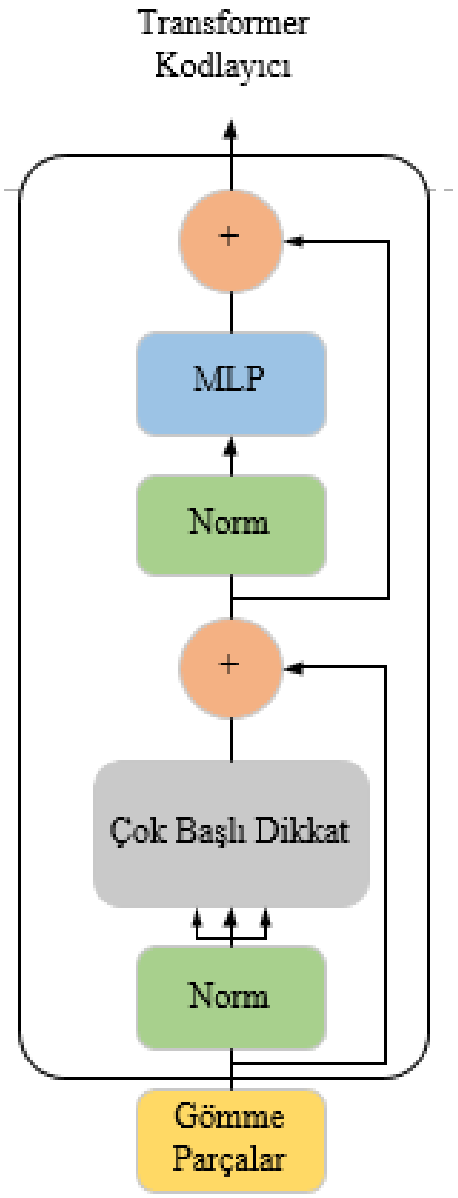
Son zamanlarda, görü dönüştürücü adı verilen model çeşitli çalışmalarda sıklıkla kullanılmaya başlanmıştır. Wasvani ve arkadaşlarınca (2017), sunulan model örnek alınarak oluşturulmuştur [19]. ViT modeli; Dosovitskiy ve arkadaşlarınca (2020), ilgili çalışmalarında tanıtılmıştır [20]. Harcanan zaman ve doğruluk kriterleri açısından avantajlı olduğu belirtilmiştir [20]. Dönüştürücü sinir ağları, daha çok NLP adı verilen dil işleme ve çeviri çalışmalarında kullanılmaktadır. Buna karşılık, Gheflani, B. ve Rivaz H. (2021), çalışmalarında meme kanseri görüntülerinin sınıflandırılması konusu üzerine eğilmektedirler [73, 74].

Şekil 4.1'de ViT model mimarisi temsili gösterilmektedir.



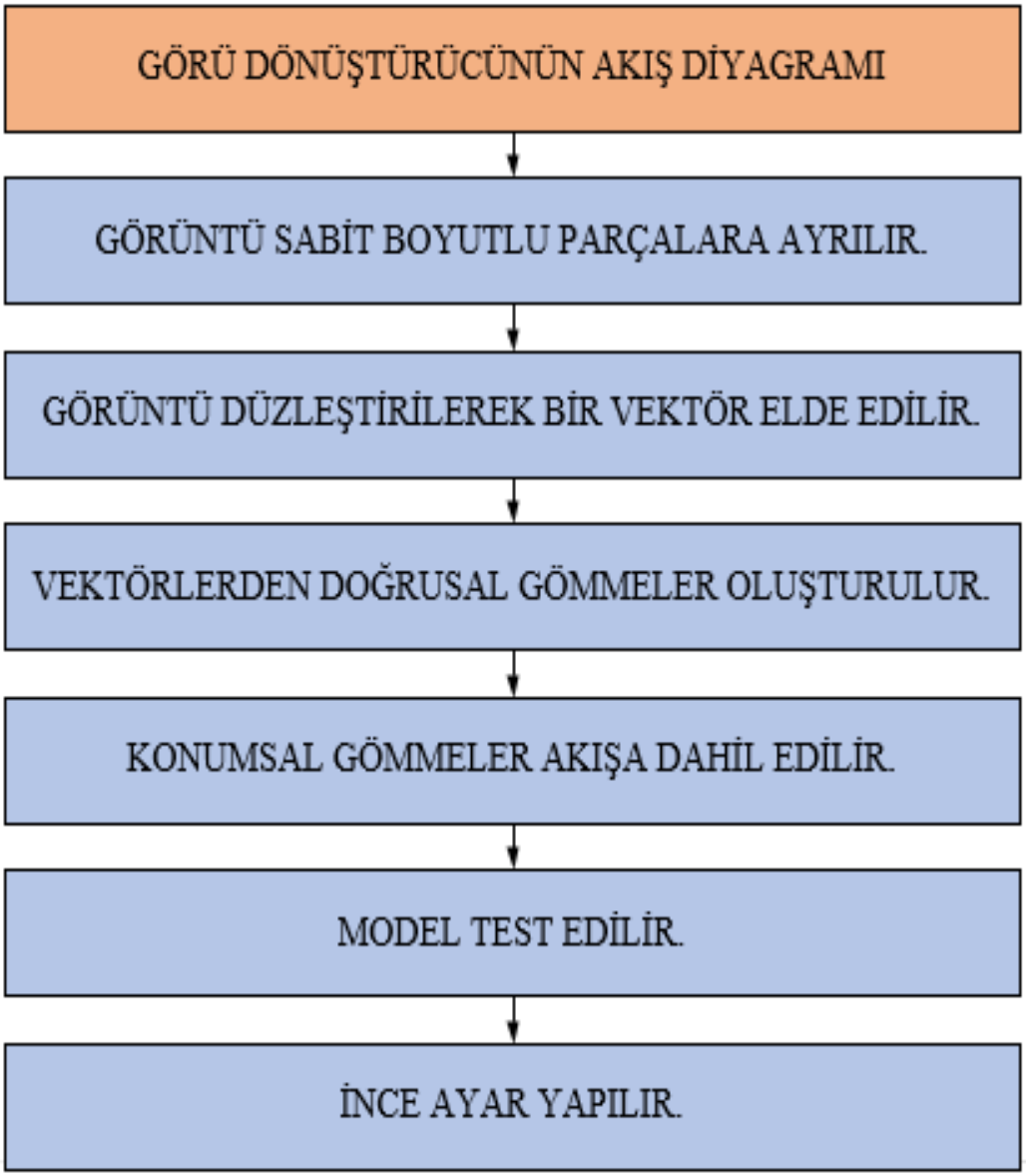
Şekil 4.1. Görü dönüştürücü mimarisi.

Modelde, giriş görüntüleri önce parçalara ayrılır ve her birinin bilgileri gömülür. Oluşturulan konumsal gömme dizisi, transformer kodlayıcıya giriş olur. ViT modelinde, kişisel dikkat katmanı görüntü bilgilerinin gömülmesini ve yeniden yapılandırma görevlerini yürütmektedir. Kodlayıcı (Encoder), yapısında bazı katmanlar barındırmaktadır. Bunlar; çok başlı öz dikkat katmanı, çok katmanlı algılayıcı ve norm katmanlarıdır [74]. Şekil 4.2'de transformatör kodlayıcı ve katmanlar gösterilmektedir.



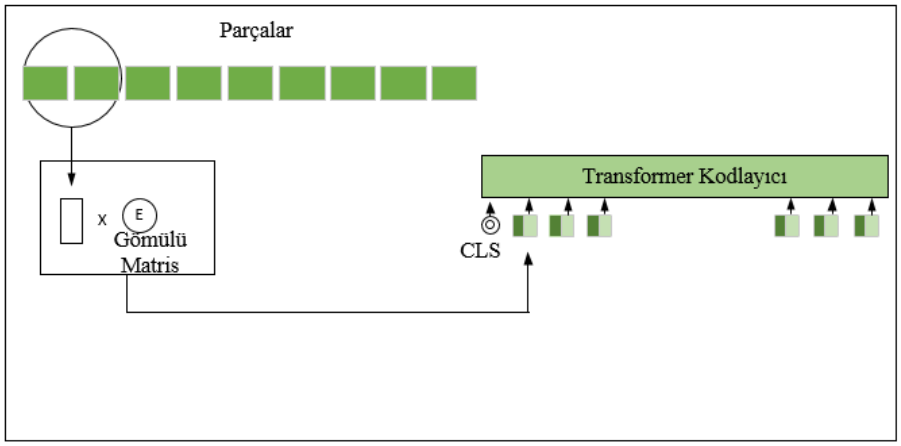
Şekil 4.2. Transformer kodlayıcı ve katmanlar.

Dönüştürücü sinir ağları, RNN'lerde yaşanan hesap zamanı uzunluğu ve kaybolan gradyan sorunu nedeniyle geliştirilmiştir. RNN'lerin bu sorunları LSTM'ler aracılığıyla çözülmeye çalışılsada belli bir seviyeye kadar idare etmektedir. Bu yüzden dönüştürücü sinir ağları bir alternatif olarak ele alınmıştır. RNN'lerde NLP çalışmaları örnek alındığında, her kelime birbiriyle bağlantılı olduğundan, her adımda kelimeyi beslemek gerekmektedir. Ancak dönüştürücü sinir ağları kullanıldığında, cümlenin her sözcük grubu gömülerek belirlenebilir [75]. Şekil 4.3'te görü dönüştürücünün akış diyagramı gösterilmektedir.



Şekil 4.3. Görü dönüştürücünün akış diyagramı.

ViT'te dönüştürücü bir boyutlu yerleştirme dizinini almakta ve iki boyutlu görüntüyü düzleştirme aşamasından önce görüntü parçalara ayrılmaktadır. Görüntünün ayrılan bu parçaları, bir E gömme matrisi aracılığıyla bir vektör oluşturur. Ayrılan parçaların konum verilerini saklamak adına transformatör kodlayıcı kısmına geçiş yapılır. Bu geçiş sırasında görüntüye ait sıra numaraları da bu kısma aktarılır. Konumsal gömme gerçekleştirilir ve tablo haline getirilir. Bu gömülü parçalar oluşturulduktan sonra transformatör kodlayıcı kısmına aktarılır [20, 21]. Şekil 4.4'te ViT kesit kodlayıcı bölümü gösterilmektedir.



Şekil 4.4. ViT kesit kodlayıcı bölümü.

Dönüştürücü, bütün katman işlemleri süresince sabit gizli vektör boyutu D'yi kullanmaktadır. GeLU ve tam bağlı iki katmandan meydana gelen MLP aracılığıyla, katmanlara sınıflandırma başlığı atanmaktadır [20, 21]. ViT ilgili denklemleri 4.1 -4.4 aşağıda gösterilmektedir.

$$z_0 = \left[x_{class}; {x_p}^1 E; {x_p}^2 E; \ldots; {x_p}^N E\right] + E_{pos} \tag{4.1}$$

$$E \epsilon R^{(P^2.C)XD} \tag{4.1a}$$

$$E_{pos} \epsilon R^{(N+1)XD} \tag{4.1b}$$

$$z'_l = MSA\left(LN(z_{l-1})\right) + z_{l-1}, \quad l = 1 \ldots . L \tag{4.2}$$

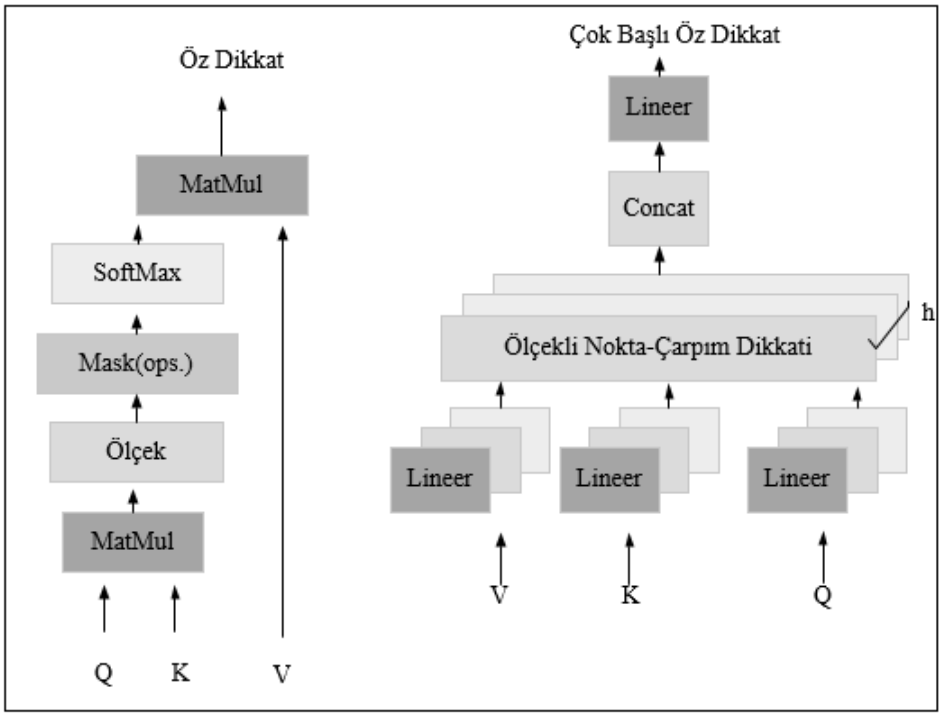
$$z_l = MLP\left(LN(z_l)\right) + z_l, \quad l = 1 \ldots . L \tag{4.3}$$

$$y_l = LN({z^0}_L) \tag{4.4}$$

ViT'te aynı RAM gibi dikkat tabanlı bir modeldir. Modelin dikkat mekanizması aracılığıyla, giriş görüntülerinin önemli kısımlarının bilgileri saklanarak, değişik ağırlık değerleri ile öncelik sırası belirlenmektedir. Dikkat mekanizmasındaki sorgu (Q), anahtar (K) ve değer (V) matrisleri giriş özelliklerine uygulanan doğrusal dönüşümler sonucu oluşturulur [19, 21]. Dikkat hesabının denklemi denklem 4.5'te gösterilmektedir.

$$Dikkat(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.5}$$

Şekil 4.5.'te öz dikkat ve çok başlı öz dikkat mekanizması gösterilmektedir.



Şekil 4.5. Öz dikkat ve çok başlı öz dikkat mekanizması.

5. DENEY VE SONUÇLAR

Çalışmada sınıflandırmada yaygın olarak kullanılan CIFAR-10 ve Fashion-MNIST veri seti kullanılmıştır. Bu bölümde, dikkat tabanlı görsel modeller olan RAM ve ViT modellerinin Fashion-MNIST ve CIFAR-10 datasetleri üzerinde gerçekleştirilen eğitim sonuçları gösterilmekte ve karşılaştırılmaktadır. ViT, RAM'e göre daha yeni bir modeldir, bu nedenle daha hızlı ve yüksek performans göstermesi beklenmektedir. ViT'ler, RNN'lerin dezavantajları olarak görülen eğitim süresi ve kaybolan gradyan probleminin bir alternatif çözümü olarak ön plana çıkmaktadır. İlk olarak çeşitli eğitimler gerçekleştirilerek, optimal sonuçların elde edildiği parametreler seçilmiştir.

5.1. Datasetler

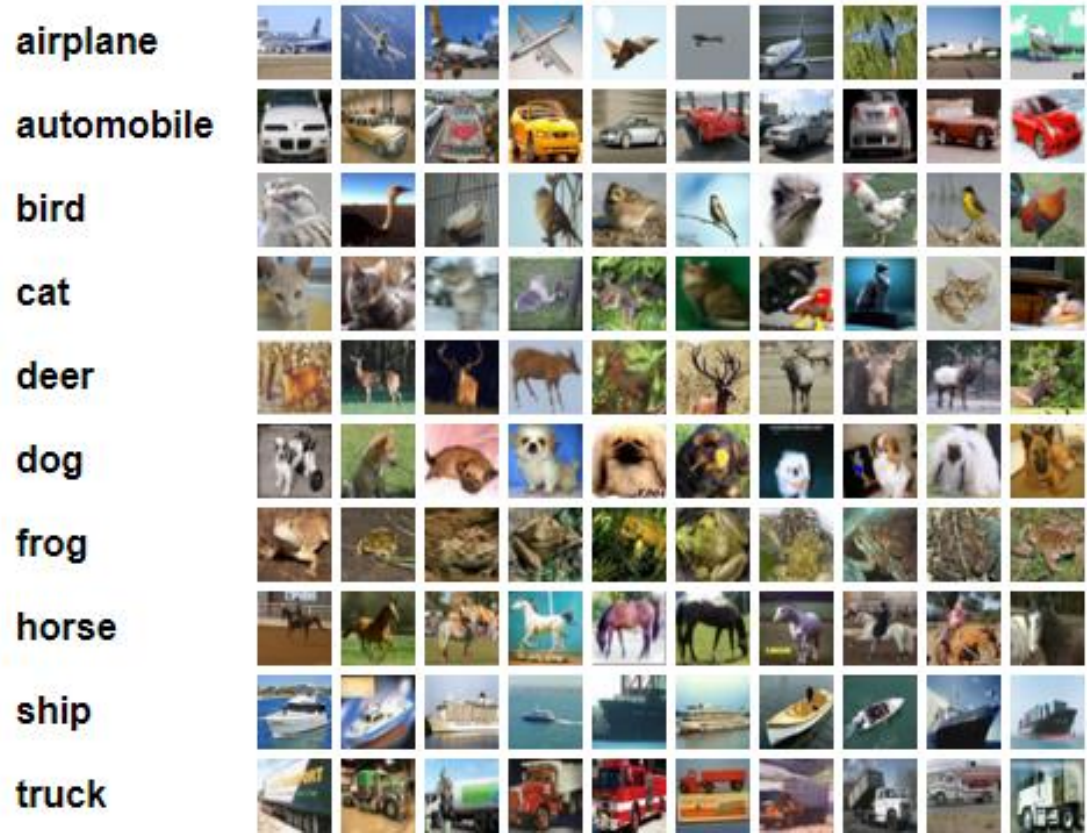
5.1.1. CIFAR-10 dataset

CIFAR-10 veri seti; uçak, otomobil, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyon olmak üzere toplam on ayrı sınıftan oluşan 60000 örneğe sahip, nesne tanıma ve görüntü sınıflandırma uygulamalarında kullanılan veri setlerinden biridir. Dataseti; Krizhevsky ve arkadaşları tarafından hazırlanarak oluşturulmuştur. Görüntüler, renklidir ve 32x32 boyutuna sahiptir. Tablo 5.1'de CIFAR-10 Datasetinin bazı özellikleri gösterilmektedir [76].

Tablo 5.1. CIFAR-10 dataset özellikleri.

Özellik	Değerler
Görüntü boyutları	32x32
Eğitim İçin Ayrılan Görüntülerin Sayısı	50000
Test İçin Ayrılan Görüntülerin Sayısı	10000
Görüntü Türü	Renkli

Şekil 5.1'de CIFAR-10 datasetine ait örnek bir bölüm gösterilmektedir [76].



Şekil 5.1. CIFAR-10 dataset'ten örnek görüntüler [76].

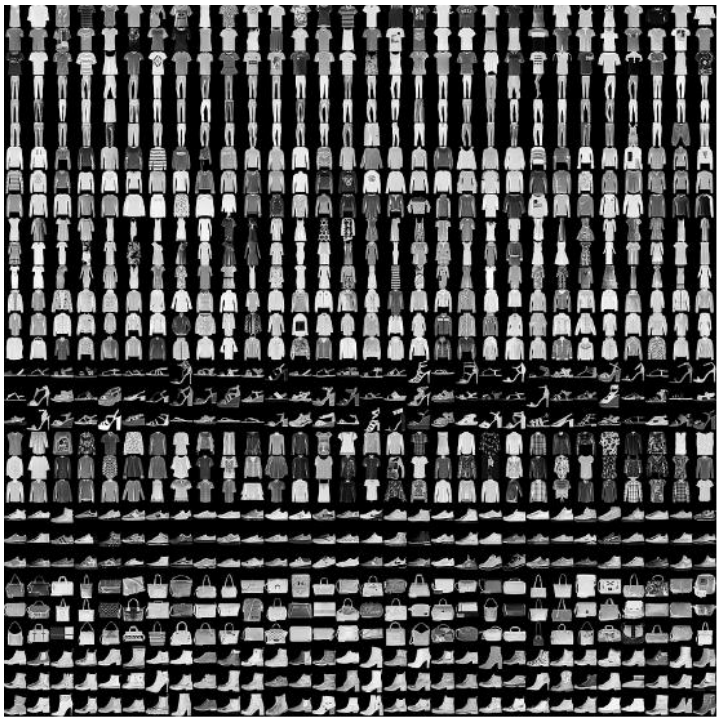
5.1.2. Fashion-MNIST dataset

Fashion-MNIST veri seti; tişört, pantolon, kazak, elbise, ceket, sandalet, gömlek, spor ayakkabı, çanta ve bilek boy bot olmak üzere toplam on ayrı sınıftan oluşan 70000 örneğe sahip, nesne tanıma ve görüntü sınıflandırma uygulamalarında kullanılan veri setlerinden biridir. Görüntüler, gridir ve 28x28 boyutuna sahiptir. Tablo 5.2'de Fashion-MNIST Datasetinin bazı özellikleri gösterilmektedir [77, 78].

Tablo 5.2. Fashion-MNIST dataset özellikleri.

Özellik	Değerler
Görüntü boyutları	28x28
Eğitim İçin Ayrılan Görüntülerin Sayısı	60000
Test İçin Ayrılan Görüntülerin Sayısı	10000
Görüntü Türü	Gri

Şekil 5.2'de Fashion-MNIST datasetine ait örnek bir bölüm gösterilmektedir.



Şekil 5.2. Fashion-MNIST dataset'ten örnek görüntüler [77].

5.2. Hiperparametreler

Bu bölüm ve alt başlıklarda her iki modelde kullanılan hiperparametreler ile ilgili bilgiler verilmektedir.

Derin ağ modellerinde, kullanılan datasete göre model parametreleri değiştirilerek, eğitim sonuçları optimize edilmeye çalışılır. Modelin sahip olduğu parametreler modeli tasarlayan kişiye ve seçilen datasete bağlıdır. Burada tasarımcının veya uygulayıcının tecrübe ve deneyimleri önem arz etmektedir [49].

5.2.1. Dataset boyutu

Datasetin boyutunun büyük olması doğruluk oranı açısından önemlidir. Datasetin boyutunun artması eğitim süresininde artmasına neden olur. Ancak zaman konusunda bir kısıtlılık yoksa bu olumsuzluk önemsenmeyebilir. Dataset boyutu arttırıldığında, doğruluk oranıda kesin olarak kayda değer şekilde artacak diye birşey söz konusu değildir, belli bir andan sonra doğruluk daha az şekilde artar ve kayda değer bir artış olmadığı durumlarda iterasyon sona erdilir [49].

5.2.2. Batch size

Derin ağ çalışmalarında, zamanın ekonomik şekilde kullanımı ve hafıza açısından, kullanılan datasetteki verilerin komple bir biçimde değil, kısım kısım işlenmesi mantıklı bir çözümdür. Batch-size, datasetteki verilerin kaçar kaçar işleneceğini belitmektedir [49].

5.2.3. Öğrenme oranı

Öğrenme oranı, modelde ağırlıkların yenilenmesi aşamasında rolü olan bir parametredir. Öğrenme oranı belli bir değer olarak tutulabileceği gibi belirli adımda gitgide değişen bir değer olarakta ayarlanabilmektedir. Genel bir değer olarak, değişken bir biçimde 0.01-0.001 değerleri kullanılabilmektedir [49, 79].

5.2.4. İterasyon sayısı (epoch)

Model eğitime sokulup, doğruluk oranı tespit edildikten sonra ağırlıklar yenilerek model tekrar eğitilir ve optimum ağırlık değerleri hesap edilir. Bu her bir aşamaya epoch adı verilmektedir. İterasyon sayısının arttırılması doğruluğu arttıracaktır ancak belli bir aşamadan sonra doğruluk artışı azaldığından eğitim durdurulabilir [49].

5.2.5. Patch size

Yineleyen dikkat ve görü dönüştürücü modellerinde giriş görüntüsünden alınan parça boyutunu temsil eder. Yineleyen dikkat modelinde alınan parçanın boyutu ile ilişkiliyken; görü dönüştürücüde ise sadece alınan parça boyutunu temsil ile kalmayıp aynı zamanda giriş görüntüsünün ayrıldığı parça sayısını da doğrudan etkilemektedir.

5.2.6. Anlık bakış sayısı

Bu parametre ise yineleyen dikkat modelinin bakış ağı parametrelerinden biridir ve eğitim süresi ile ilişkili bir değerdir. Değer ile eğitim süresi arasında doğrusal bir ilişki söz konusudur.

5.2.7. MLP boyutu

Görü dönüştürücü modelinde kullanılan sinir ağı katmanıdır. GeLU aktivasyon fonksiyonu ile kullanılır.

5.3. RAM Sonuçları

RAM'de parça boyutu ve anlık bakış sayısı önemli parametrelerdir. Tablo 5.3'te parça boyutu ve anlık bakış sayısı değiştirilerek, Fashion-MNIST veriseti üzerinde gerçekleştirilen eğitim sonuçları ve eğitim süreleri gösterilmektedir. Bu tablodan optimal sonuç olarak parça boyutu 8 ve anlık bakış sayısı 6 olarak seçilerek karşılaştırmalı sonuçlarda bu eğitim sonuçları paylaşılmaktadır. Eğitimlerde; epoch 300, std 0,05, batch boyutu 128, öğrenme oranı 0,0003 olarak alınmıştır.

Tablo 5.3. Fashion-MNIST eğitim denemeleri sonuçları.

Patch Size	Bakış Sayısı	Eğitim Doğruluk	Eğitim Kayıp	Test Doğruluk	Test Kayıp	Eğitim Süresi
8	4	90,133	0,356	87,767	0,448	2.50h
8	6	92,531	0,263	88,583	0,416	3.33h
8	8	91,980	0,284	89,983	0,362	3.46h
10	4	91,800	0,307	87,933	0,441	2.38h
10	6	91,831	0,290	88,667	0,413	3.44h
10	8	92,210	0,275	88,450	0,421	3.54h
12	4	91,009	0,322	88,083	0,435	2.14h
12	6	92,410	0,268	88,225	0,430	3.33h
12	8	92,320	0,271	89,950	0,363	4.03h

Tablo 5.4'te parça boyutu ve anlık bakış sayısı değiştirilerek, CIFAR-10 veriseti üzerinde gerçekleştirilen eğitim sonuçları ve eğitim süreleri gösterilmektedir. Bu tablodan optimal sonuç olarak parça boyutu 12 ve anlık bakış sayısı 8 olarak seçilerek karşılaştırmalı sonuçlarda bu eğitim sonuçları paylaşılmaktadır.

Tablo 5.4. CIFAR-10 eğitim denemeleri sonuçları.

Patch Size	Bakış Sayısı	Eğitim Doğruluk	Eğitim Kayıp	Test Doğruluk	Test Kayıp	Eğitim Süresi
8	4	65,278	1,298	61,340	1,469	2.49h
8	6	68,758	1,182	64,300	1,340	3.38h
8	8	69,478	1,145	66,380	1,250	3.43h
10	4	64,709	1,323	62,160	1,434	2.52h
10	6	70,307	1,078	65,560	1,286	3.16h
10	8	70,296	1,079	65,860	1,273	3.28h
12	4	68,180	1,211	64,200	1,345	2.56h
12	6	70,093	1,088	65,280	1,298	3.24h
12	8	73,224	0,952	68,560	1,155	3,41h

Eğitimlerde; epoch 300, std 0,05, batch boyutu 128, öğrenme oranı 0,0003 olarak alınmıştır.

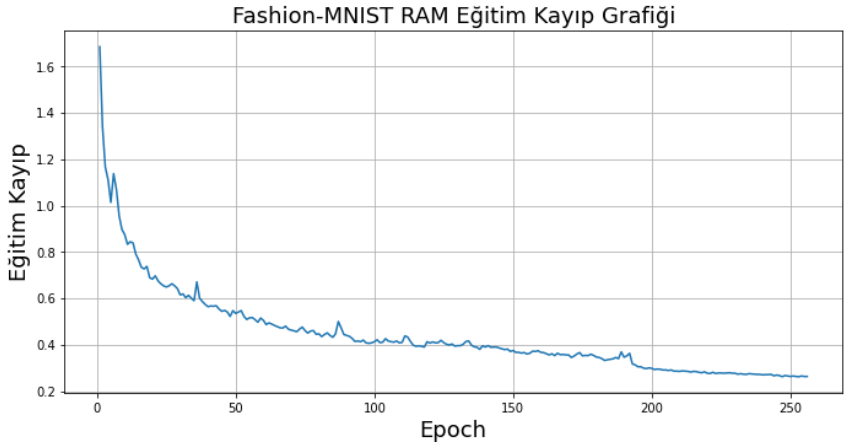
5.3.1. Fashion-MNIST dataset sonuçları

Fashion-MNIST datasetinin seçilen parametreler üzerinden RAM modeli eğitimi gerçekleştirilmiştir. Tablo 5.5'te Fashion-MNIST datasetinin eğitimi gerçekleştirilirken kullanılan parametreler gösterilmektedir.

Tablo 5.5. Fashion-MNIST RAM parametreleri.

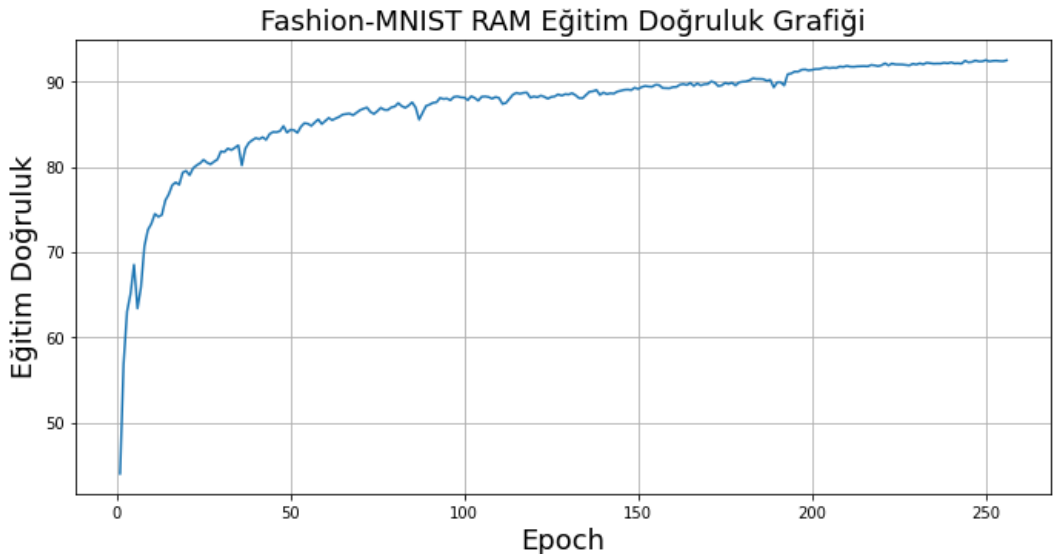
Parametre	Değerler
Patch Size	8
Bakış Sayısı	6
std	0,05
Batch Size	128
İterasyon Sayısı (Epochs)	256/300
Öğrenme Oranı	0,0003

Modelde; patch boyutu 8, anlık bakış sayısı 6, gauss standart sapma politikası 0,05, verilerin kaçar kaçar işleneceğini belirten batch boyutu 128, momentum 0,5, iterasyon sayısı 300, öğrenme oranı 0,0003 olarak uygulanmıştır. Daha fazla gelişim kaydedilmediğinden eğitim 256. iterasyonda sonlandırılmıştır. Öğrenme oranı 0,0003 olarak belirlenmiştir, fakat sabit bir değer değildir. Öğrenme oranı zamanla azalmaktadır. Şekil 5.3'te Fashion-MNIST datasetinin RAM üzerinde eğitimi sonucu ulaşılan eğitim kaybı gösterilmektedir.



Şekil 5.3. Fashion-MNIST RAM eğitim kayıp grafiği.

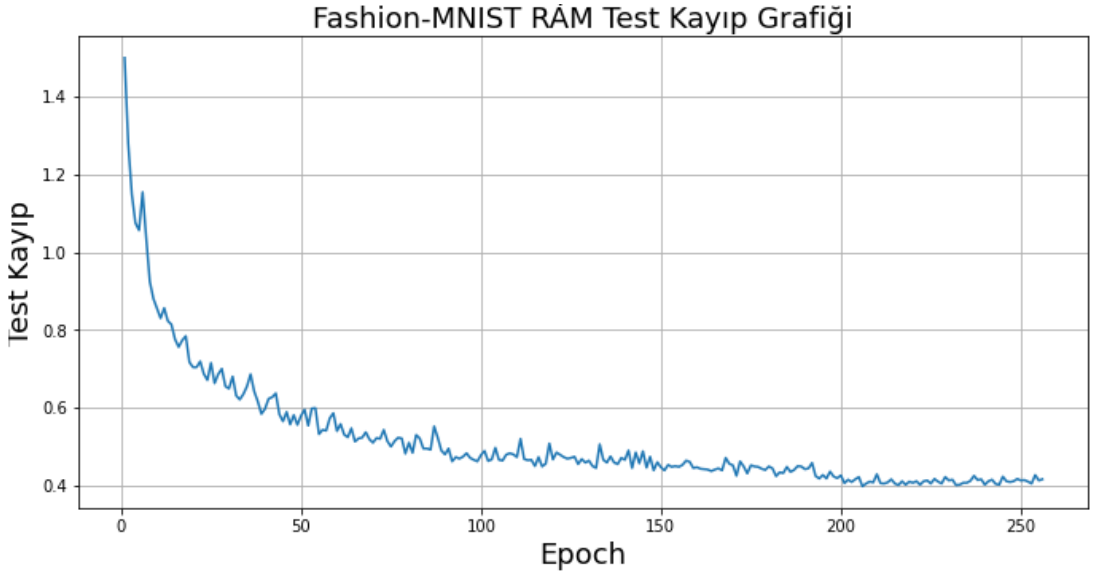
Şekil 5.4'te Fashion-MNIST datasetinin RAM üzerinde eğitimi sonucu ulaşılan eğitim doğruluğu gösterilmektedir.



Şekil 5.4. Fashion-MNIST RAM eğitim doğruluk grafiği.

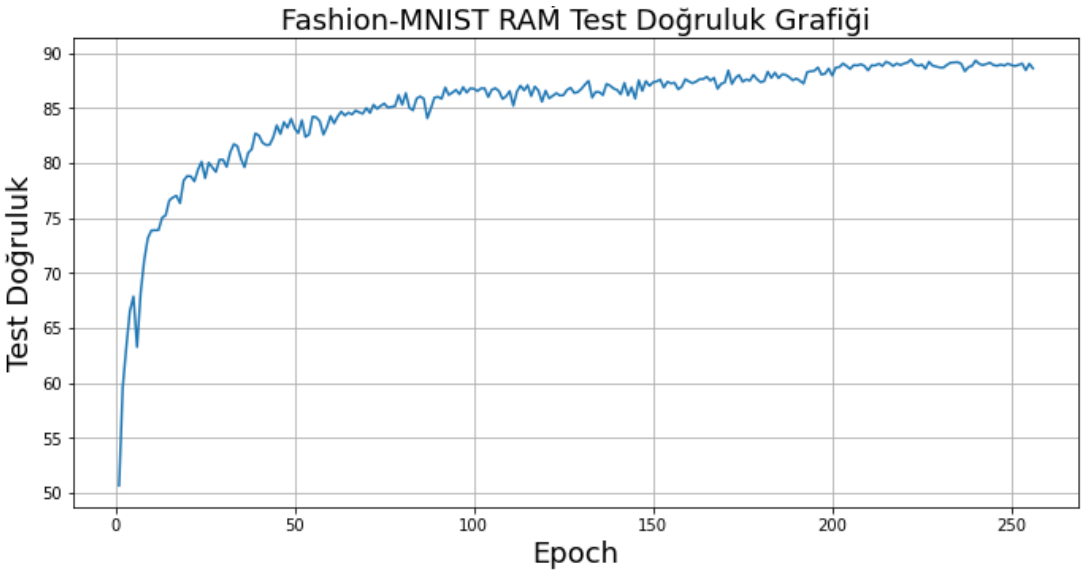
Şekil 5.3 ve şekil 5.4'te görüldüğü gibi eğitim kaybı zamanla azalan bir çizgide seyrederken, eğitim doğruluğunda ise tam tersi bir seyir söz konusudur.

Şekil 5.5'te Fashion-MNIST datasetinin RAM üzerinde eğitimi sonucu ulaşılan test kaybı gösterilmektedir.



Şekil 5.5. Fashion-MNIST RAM test kayıp grafiği.

Şekil 5.6'da Fashion-MNIST datasetinin RAM üzerinde eğitimi sonucu ulaşılan test doğruluğu gösterilmektedir.



Şekil 5.6. Fashion-MNIST RAM test doğruluk grafiği.

Şekil 5.5 ve şekil 5.6'da görüldüğü gibi test kaybı zamanla azalan bir çizgide seyrederken, test doğruluğunda ise tam tersi bir seyir söz konusudur. Tablo 5.6'da elde edilen sonuçlar tablo halinde gösterilmektedir.

Tablo 5.6. Fashion-MNIST RAM sonuçları.

Dataset	Eğitim Kayıp	Eğitim Doğruluk	Test Kayıp	Test Doğruluk
Fashion-MNIST	0,263	92,531	0,416	88,583

5.3.2. CIFAR-10 dataset sonuçları

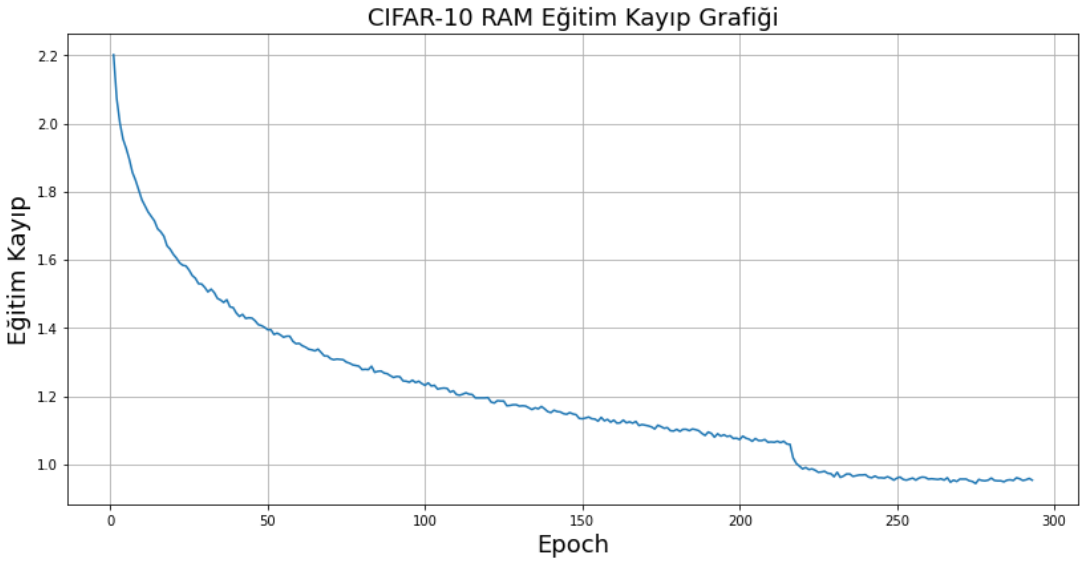
CIFAR-10 datasetinin seçilen parametreler ile RAM modeli üzerinde eğitimi gerçekleştirilmiştir. Tablo 5.7'de CIFAR-10 datasetinin eğitimi gerçekleştirilirken kullanılan parametreler gösterilmektedir.

Tablo 5.7. CIFAR-10 RAM parametreleri.

Parametre	Değerler
Patch Size	12
Bakış Sayısı	8
std	0,25
Batch Size	128
İterasyon Sayısı (Epochs)	293/300
Öğrenme Oranı	0,0003

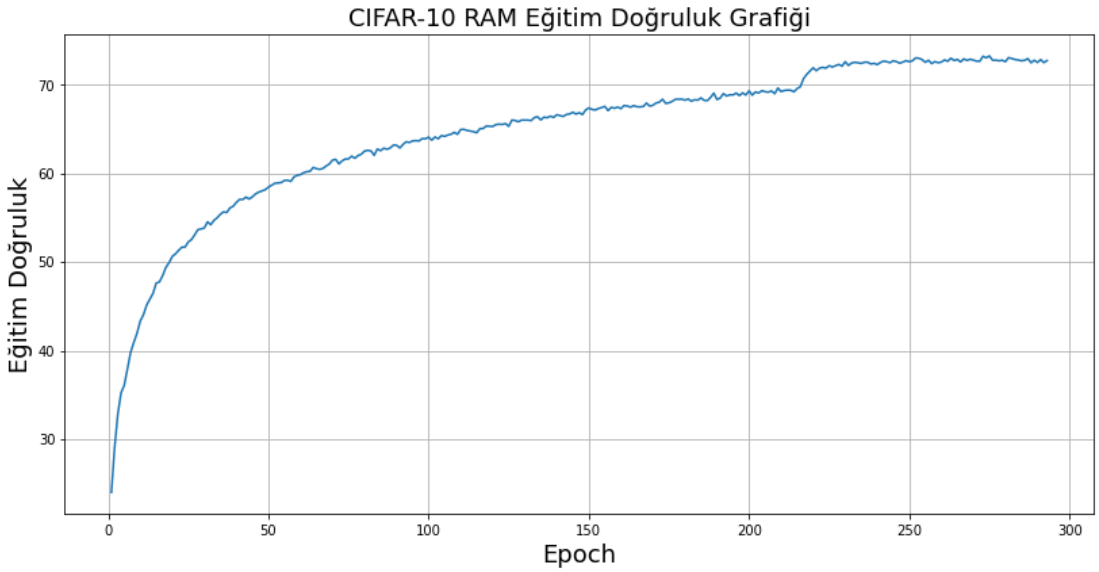
Modelde; görüntü boyutu ile ilişkili parça boyutu 12'ye çıkarılmıştır. Anlık bakış sayısı 6'dan 8'e çıkarılmış, gauss standart sapma politikası 0,25, verilerin kaçar kaçar işleneceğini belirten batch boyutu 128, momentum 0,5, iterasyon sayısı 300, öğrenme oranı 0,0003 olarak uygulanmıştır. Daha fazla gelişim kaydedilmediğinden eğitim 293. iterasyonda sonlandırılmıştır. Öğrenme oranı 0,0003 olarak belirlenmiştir, fakat sabit bir değer değildir. Öğrenme oranı zamanla azalmaktadır.

Şekil 5.7'de CIFAR-10 datasetinin RAM üzerinde eğitimi sonucu ulaşılan eğitim kaybı gösterilmektedir.



Şekil 5.7. CIFAR-10 RAM eğitim kayıp grafiği.

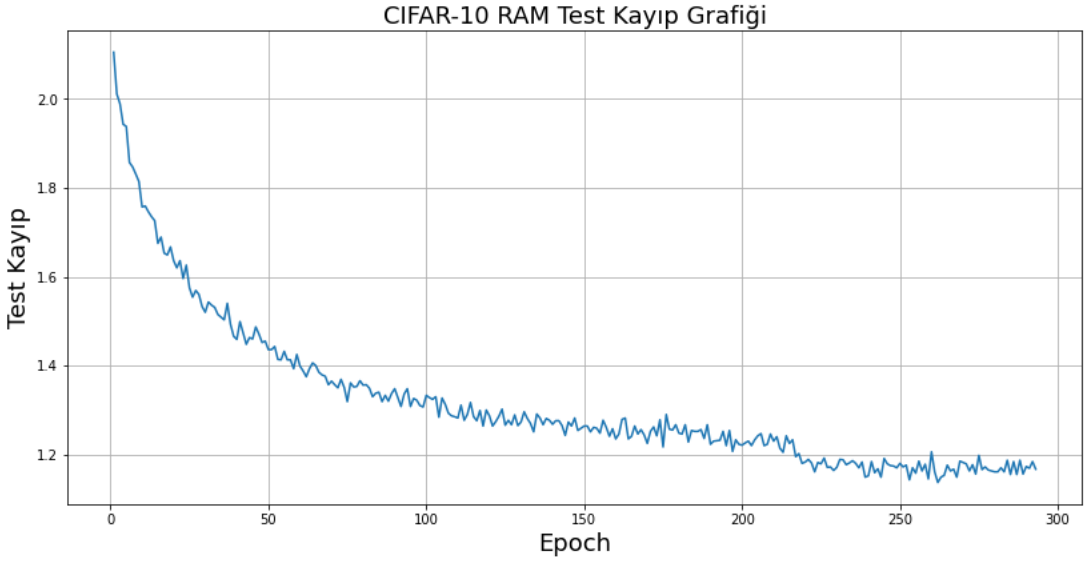
Şekil 5.8'de CIFAR-10 datasetinin RAM üzerinde eğitimi sonucu ulaşılan eğitim doğruluğu gösterilmektedir.



Şekil 5.8. CIFAR-10 RAM eğitim doğruluk grafiği.

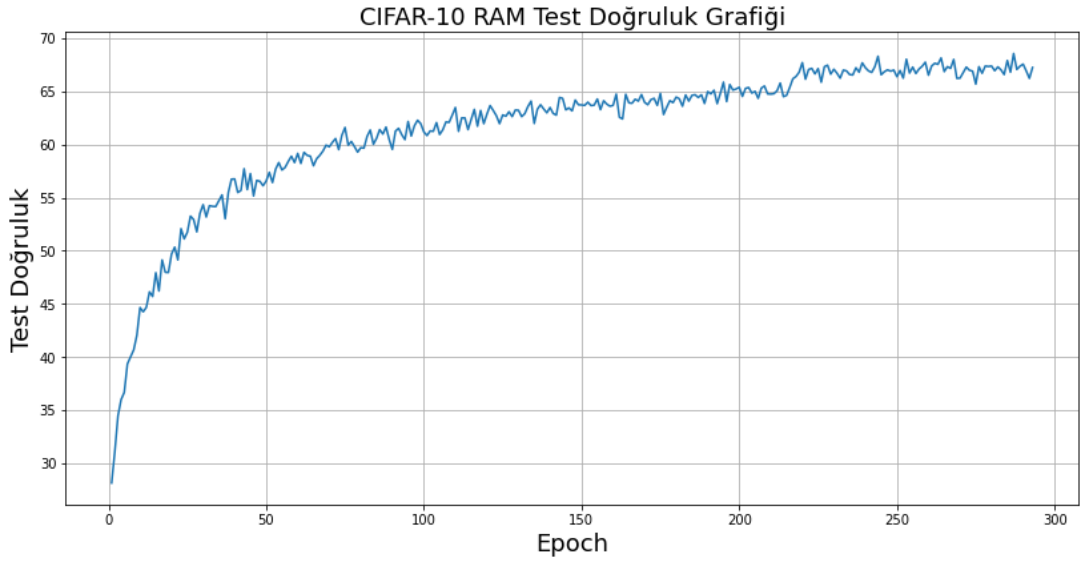
Şekil 5.7 ve şekil 5.8'de görüldüğü gibi eğitim kaybı zamanla azalan bir çizgide seyrederken, eğitim doğruluğunda ise tam tersi bir seyir söz konusudur.

Şekil 5.9'da CIFAR-10 datasetinin RAM üzerinde eğitimi sonucu ulaşılan test kaybı gösterilmektedir.



Şekil 5.9. CIFAR-10 RAM test kayıp grafiği.

Şekil 5.10'da CIFAR-10 datasetinin RAM üzerinde eğitimi sonucu ulaşılan test doğruluğu gösterilmektedir.



Şekil 5.10. CIFAR-10 RAM test doğruluk grafiği.

Şekil 5.9 ve şekil 5.10'da görüldüğü gibi test kaybı zamanla azalan bir çizgide seyrederken, test doğruluğunda ise tam tersi bir seyir söz konusudur. Tablo 5.8'de elde edilen sonuçlar tablo halinde gösterilmektedir.

Tablo 5.8. CIFAR-10 RAM sonuçları.

Dataset	Eğitim Kayıp	Eğitim Doğruluk	Test Kayıp	Test Doğruluk
CIFAR-10	0,952	73,224	1,155	68,560

5.4. ViT Sonuçları

Görü dönüştürücüde parça sayısı parça boyutu ile ilişkili bir kavramdır. Giriş görüntüsünün boyutlarının çarpımının parça boyutlarının çarpımına bölünmesiyle transformatör kodlayıcıya giriş olan parça sayısı elde edilmektedir. Parça boyutu arttığında eğitim süresi azalmaktadır ancak doğruluk ve kayıp oranları da aynı şekilde düşmektedir. Bu nedenle her iki datasette de parça boyutu olarak 4 seçilmiştir. Eğitimlerde Fashion-MNIST datasette batch boyutu olarak 100, derinlik 6, epoch 40, başlık sayısı 8 olarak alınmıştır. CIFAR-10'da farklı olarak epoch sayısı 200'dür. Ayrıca MLP boyutunda değişiklik yapılarak 128 ve 512 değerleri için eğitim denemeleri gerçekleştirilmiştir.

Fashion-MNIST datasette çok büyük değişimler gözlemlenmemiştir. CIFAR-10 datasette ise MLP boyutunun düşürülmesi doğruluk oranlarını azaltırken; kayıp oranlarını arttırmıştır.

Tablo 5.9'da Fashion-MNIST dataseti üzerinde gerçekleştirilen eğitim sonuçları gösterilmektedir.

Tablo 5.9. Fashion-MNIST eğitim denemeleri sonuçları.

Patch Size	MLP Boyutu	Eğitim Doğruluk	Eğitim Kayıp	Test Doğruluk	Test Kayıp	Eğitim Süresi
4	128	89,2	0,290	88,35	0,3673	1.15h
7	128	88,950	0,313	88,025	0,397	1.06h
14	128	88,770	0,329	87,61	0,434	1.02h
4	512	88,570	0.349	87,91	0,413	1.36h
7	512	85,800	0,5338	85,21	0,553	1.21h
14	512	89,15	0,293	88,39	0,365	1.13h

Tablo 5.10'da CIFAR-10 dataset üzerinde gerçekleştirilen eğitim sonuçları gösterilmektedir.

Tablo 5.10. CIFAR-10 eğitim denemeleri sonuçları.

Patch Size	MLP Boyutu	Eğitim Doğruluk	Eğitim Kayıp	Test Doğruluk	Test Kayıp	Eğitim Süresi
4	512	93,500	0,184	79,130	0,834	2.16h
8	512	86,910	0,498	74,730	0,971	1.41h
16	512	79,928	0,784	67,290	1,105	1.44h
4	128	88,980	0,311	78,520	0,858	2.25h
8	128	80,386	0,695	71,850	1,036	2.08h
16	128	80,086	0,718	67,240	1,108	1.40h

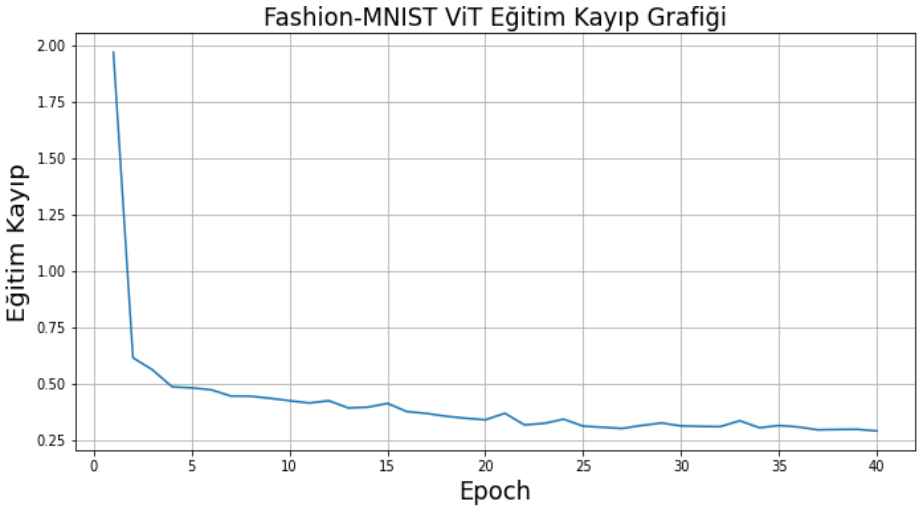
5.4.1. Fashion-MNIST dataset sonuçları

Fashion-MNIST datasetinin seçilen parametreler ile ViT modeli üzerinde eğitimi gerçekleştirilmiştir. Tablo 5.11'de Fashion-MNIST datasetinin eğitimi gerçekleştirilirken kullanılan parametreler gösterilmektedir.

Tablo 5.11. Fashion-MNIST ViT parametreleri.

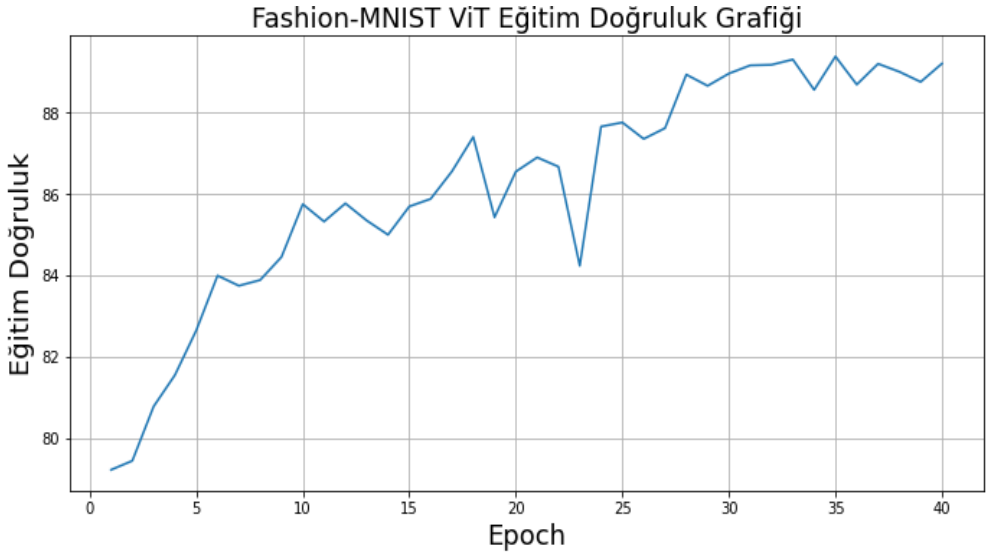
Parametre	Değerler
Patch Size	4
Batch Size	100
Depth	6
İterasyon Sayısı (Epochs)	40
Head	8
MLP boyutu	128

Modelde; patch boyutu 4, verilerin kaçar kaçar işleneceğini belirten batch boyutu 100, derinlik 6, iterasyon sayısı 40, head 8, çoklu katman algılayıcı 128 olarak uygulanmıştır.Eğitim süresi RAM'e göre hızlanmıştır. İterasyon sayısı 40 olarak uygun görülmüş bu değer sonrası test kaybının yükselip başlangıç değerine yaklaştığı gözlemlenmiştir. RAM'de ulaşılan değerlere 40 iterasyon sonrası yaklaşık olarak ulaşılmış ve buna göre ViT'in RAM'e göre daha hızlı tepki verdiği gözlemlenmiştir. Bunun ile birlikte renkli görüntü yapısına sahip olan CIFAR-10 dataseti üzerinde gerçekleştirilen eğitim sonucu bu özellikler daha belirgin şekilde görülmektedir. Şekil 5.11'de Fashion-MNIST datasetinin ViT üzerinde eğitimi sonucu ulaşılan eğitim kaybı gösterilmektedir.



Şekil 5.11. Fashion-MNIST ViT eğitim kayıp grafiği.

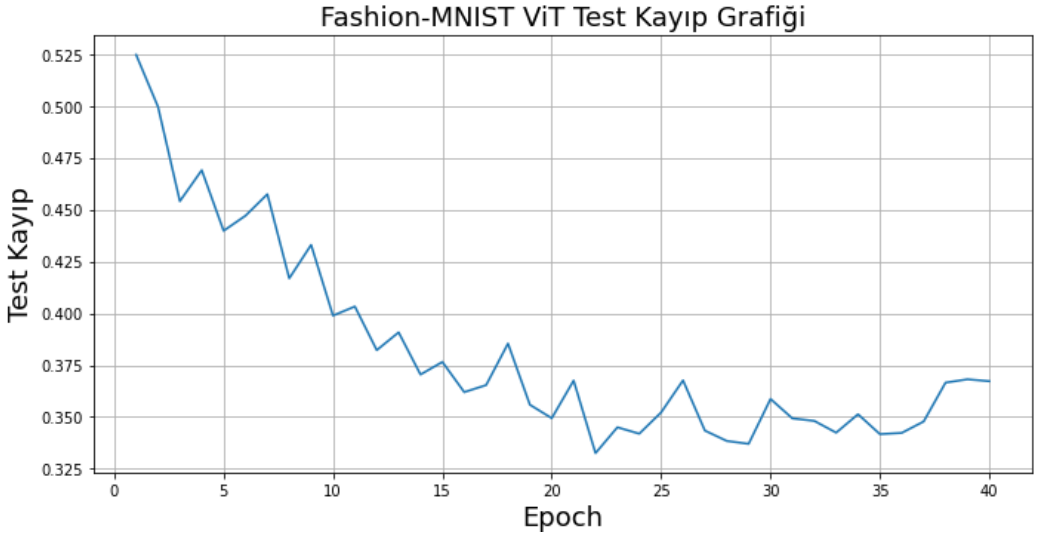
Şekil 5.12'de Fashion-MNIST datasetinin ViT üzerinde eğitimi sonucu ulaşılan eğitim doğruluğu gösterilmektedir.



Şekil 5.12. Fashion-MNIST ViT eğitim doğruluk grafiği.

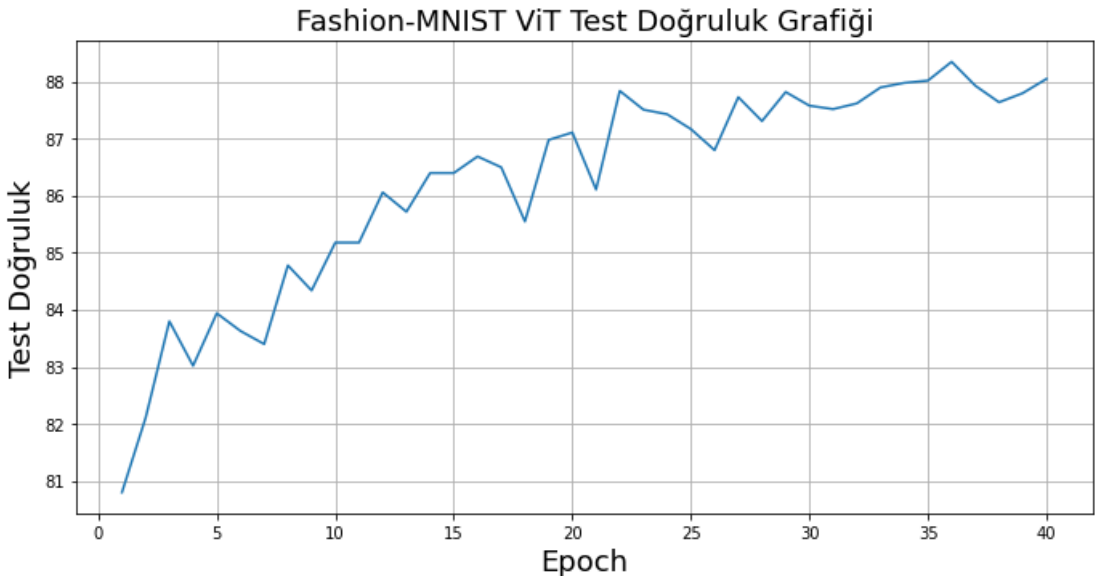
Şekil 5.11 ve şekil 5.12'de görüldüğü gibi eğitim kaybı zamanla azalan bir çizgide seyrederken, eğitim doğruluğunda ise tam tersi bir seyir söz konusudur.

Şekil 5.13'te Fashion-MNIST datasetinin ViT üzerinde eğitimi sonucu ulaşılan test kaybı gösterilmektedir.



Şekil 5.13. Fashion-MNIST ViT test kayıp grafiği.

Şekil 5.14'te Fashion-MNIST datasetinin ViT üzerinde eğitimi sonucu ulaşılan test doğruluğu gösterilmektedir.



Şekil 5.14. Fashion-MNIST ViT test doğruluk grafiği.

Şekil 5.13 ve şekil 5.14'te görüldüğü gibi test kaybı zamanla azalan bir çizgide seyrederken, test doğruluğunda ise tam tersi bir seyir söz konusudur. Tablo 5.12'de elde edilen sonuçlar tablo halinde gösterilmektedir.

Tablo 5.12. Fashion-MNIST ViT sonuçları.

Dataset	Eğitim Kayıp	Eğitim Doğruluk	Test Kayıp	Test Doğruluk
Fashion-MNIST	0,29	89,2	0,3673	88,350

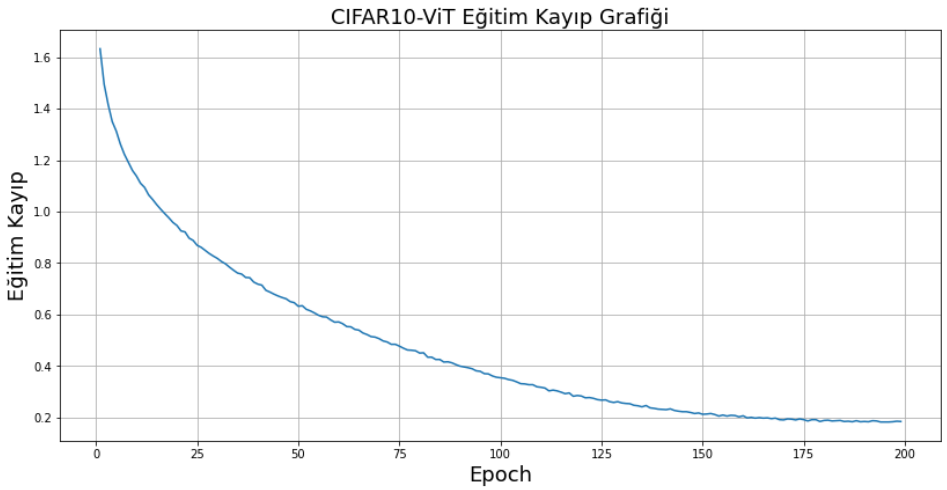
5.4.2. CIFAR-10 dataset sonuçları

CIFAR-10 datasetinin seçilen parametreler ile ViT modeli üzerinde eğitimi gerçekleştirilmiştir. Tablo 5.13'te CIFAR-10 datasetinin eğitimi gerçekleştirilirken kullanılan parametreler gösterilmektedir.

Tablo 5.13. CIFAR-10 ViT parametreleri.

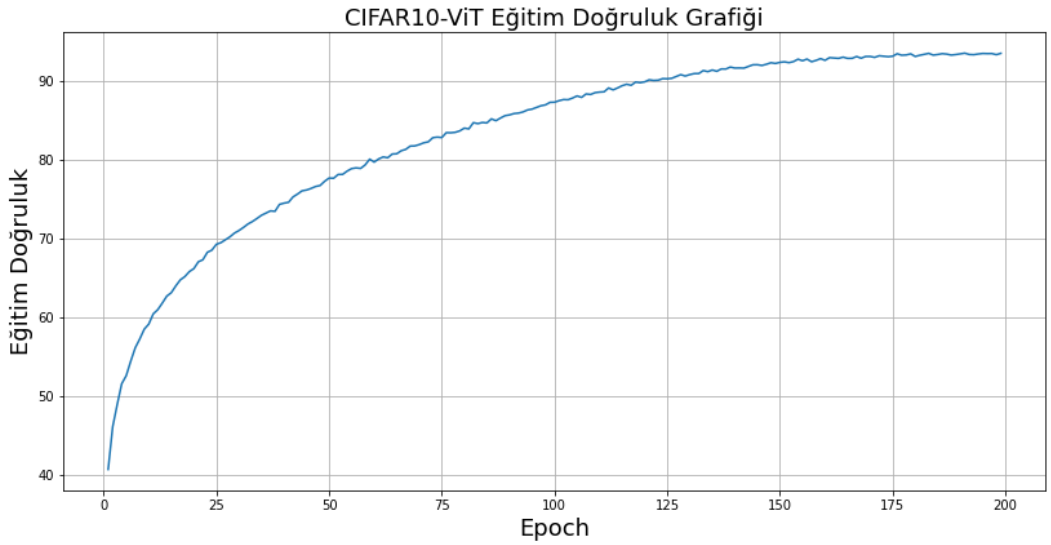
Parametre	Değerler
Patch Size	4
Batch Size	100
Depth	6
İterasyon Sayısı (Epochs)	200
Head	8
MLP boyutu	512

Modelde; patch boyutu 4, verilerin kaçar kaçar işleneceğini belirten batch boyutu 100, derinlik 6, iterasyon sayısı 200, head 8, çoklu katman algılayıcı 512 olarak uygulanmıştır. Eğitim süresi RAM'e göre hızlanmıştır. İterasyon sayısı 200 olarak uygun görülmüş bu değer sonrası test kaybının yükselip başlangıç değerine yaklaştığı gözlemlenmiştir. 200 iterasyon sonrası RAM'de ulaşılan değerlerden daha yüksek değerlere ulaşıldığı ve buna göre ViT'in RAM'e göre daha hızlı tepki verdiği gözlemlenmiştir. Bunun ile birlikte Fashion-MNIST datasetinden (gri) daha karmaşık bir set olan CIFAR-10 dataseti (renkli) üzerinde gerçekleştirilen eğitim sonucu bu özellikler daha belirgin şekilde görülmektedir. Şekil 5.15'te CIFAR-10 datasetinin ViT üzerinde eğitimi sonucu ulaşılan eğitim kaybı gösterilmektedir.



Şekil 5.15. CIFAR-10 ViT eğitim kayıp grafiği.

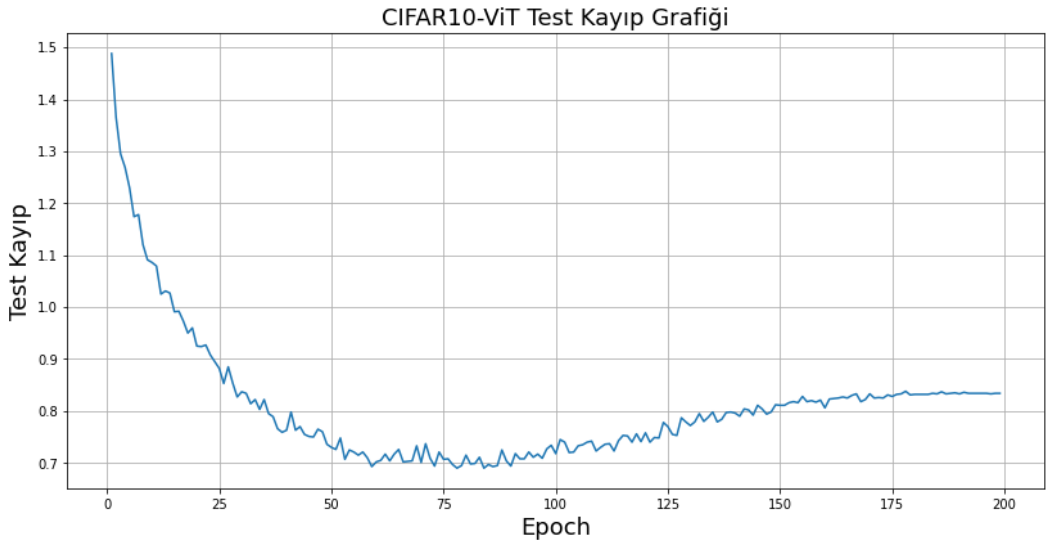
Şekil 5.16'da CIFAR-10 datasetinin ViT üzerinde eğitimi sonucu ulaşılan eğitim doğruluğu gösterilmektedir.



Şekil 5.16. CIFAR-10 ViT eğitim doğruluk grafiği.

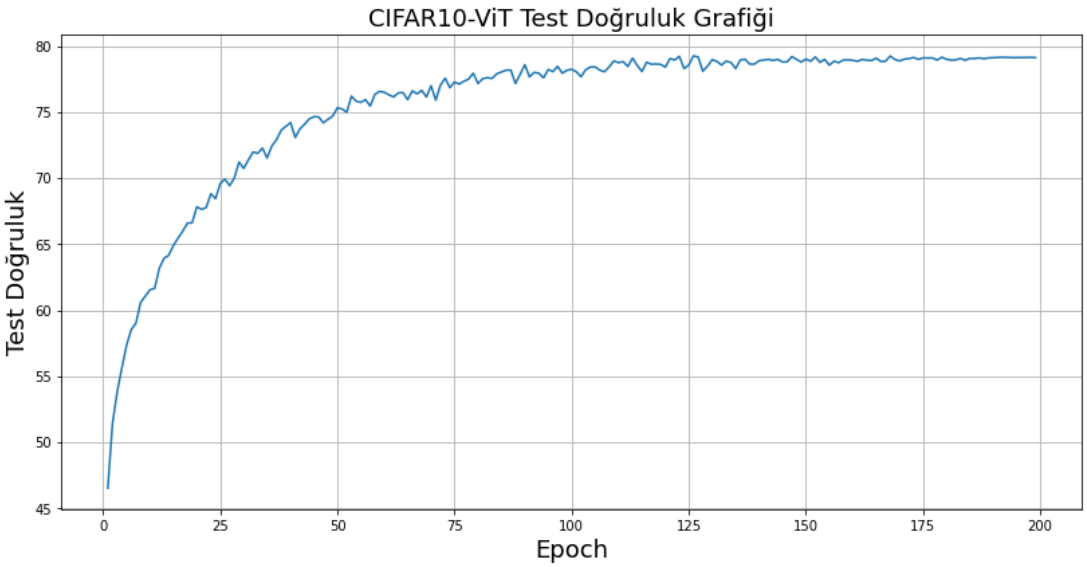
Şekil 5.15 ve şekil 5.16'da görüldüğü gibi eğitim kaybı zamanla azalan bir çizgide seyrederken, eğitim doğruluğunda ise tam tersi bir seyir söz konusudur.

Şekil 5.17'de CIFAR-10 datasetinin ViT üzerinde eğitimi sonucu ulaşılan test kaybı gösterilmektedir.



Şekil 5.17. CIFAR-10 ViT test kayıp grafiği.

Şekil 5.18'de CIFAR-10 datasetinin ViT üzerinde eğitimi sonucu ulaşılan test doğruluğu gösterilmektedir.



Şekil 5.18. CIFAR-10 ViT test doğruluk grafiği.

Şekil 5.17 ve şekil 5.18'de görüldüğü gibi test kaybı zamanla azalan bir çizgide seyrederken, test doğruluğunda ise tam tersi bir seyir söz konusudur. Tablo 5.14'te elde edilen sonuçlar tablo halinde gösterilmektedir.

Tablo 5.14. CIFAR-10 ViT sonuçları.

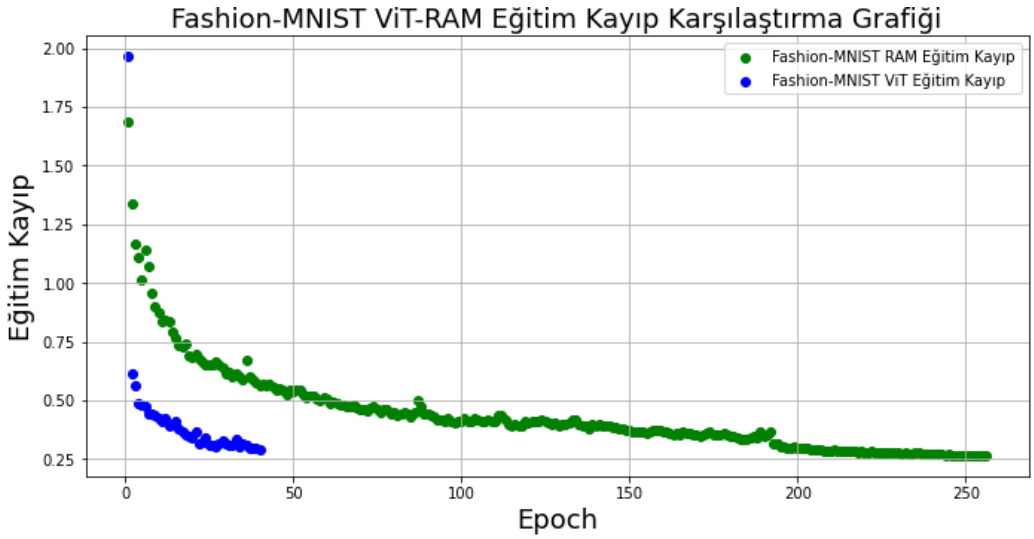
Dataset	Eğitim Kayıp	Eğitim Doğruluk	Test Kayıp	Test Doğruluk
CIFAR-10	0,184	93,500	0,834	79,130

5.5. RAM-ViT Sonuçlarının Karşılaştırması

5.5.1. Fashion-MNIST dataseti RAM-ViT sonuçlarının karşılaştırması

ViT, RAM'e göre daha yeni bir modeldir. Her iki modelde dikkat tabanlıdır. Dönüştürücü sinir ağları, RNN'lerde görülen eğitim süresinin uzunluğu ve kaybolan gradyan probleminin çözümüne yönelik bir fikir olarak ortaya atılmıştır. RNN'lerin bu sorununa LSTM ile çözüm aranırken belli bir müddetten sonra LSTM'lerde çok verimli olmamaktadır. ViT'in ortaya çıkışında bu gibi etmenler ana sebeptir. Dönüştürücü sinir ağları öncelikli olarak NLP ve çeviri çalışmalarında kullanılsada, şu an görüntü sınıflandırma görevlerinde de çeşitli uygulamalar gerçekleştirilmektedir.

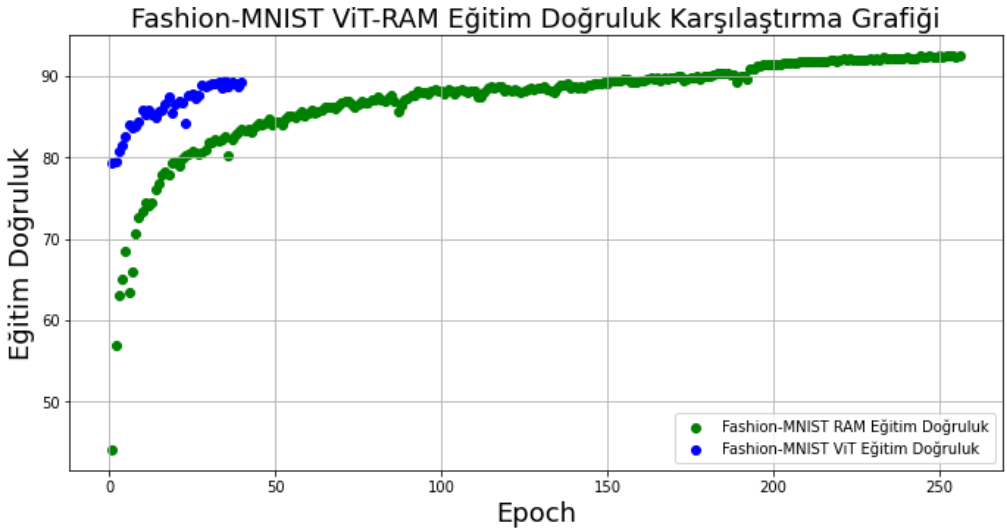
Bu bölümde elde edilen sonuçlar karşılaştırmalı olarak verilmekte ve yorumlanmaktadır. Şekil 5.19'da Fashion-MNIST datasetinin RAM-ViT eğitim kaybı karşılaştırmalı grafiği gösterilmektedir.



Şekil 5.19. Fashion-MNIST RAM-ViT eğitim kayıp karşılaştırma grafiği.

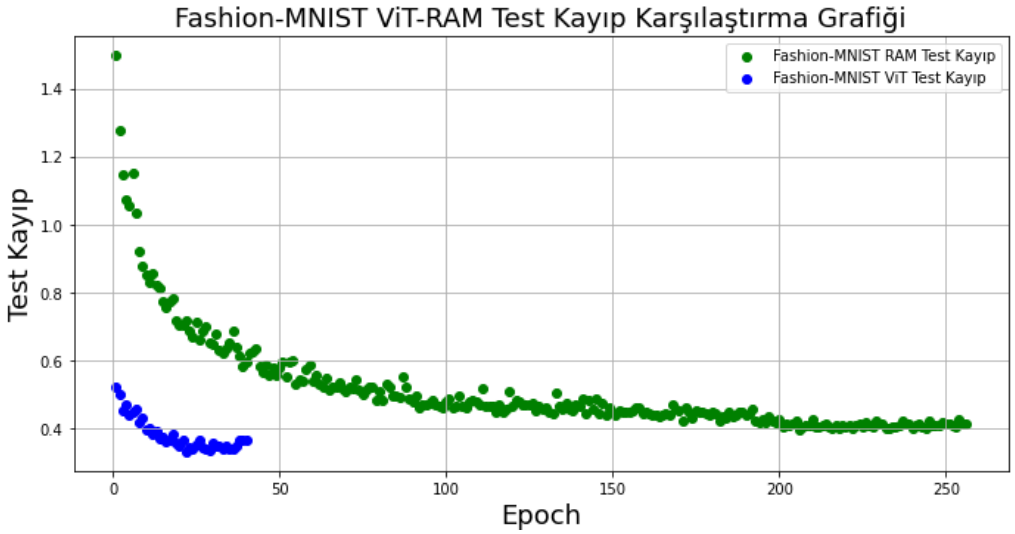
Görüldüğü üzere ViT ile 40 iterasyonda, RAM'e yakın sonuçlar elde edilmektedir. ViT, RAM'e göre eğitim süresi uzunluğu bakımından avantaj sağlamaktadır.

Şekil 5.20'de Fashion-MNIST datasetinin RAM-ViT eğitim doğruluğu karşılaştırmalı grafiği gösterilmektedir.



Şekil 5.20. Fashion-MNIST RAM-ViT eğitim doğruluğu karşılaştırma grafiği.

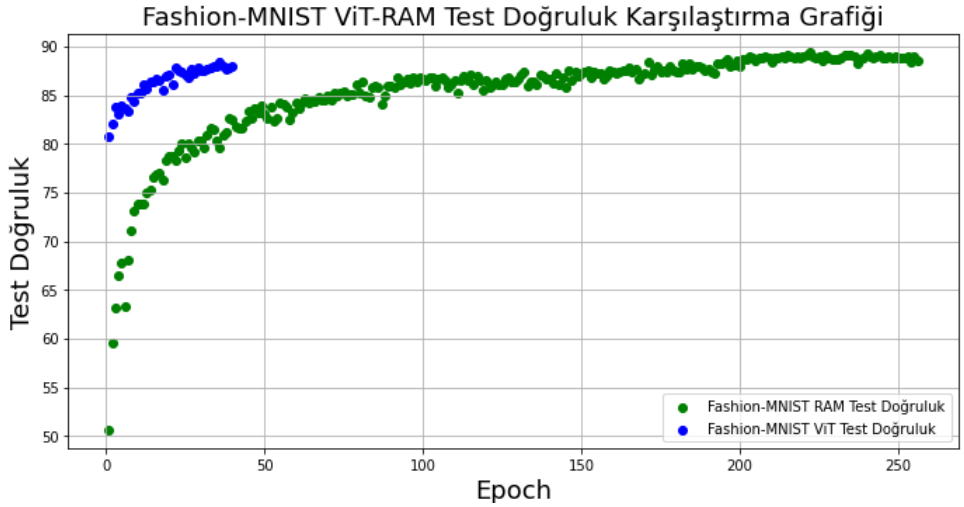
Görüldüğü üzere ViT ile 40 iterasyonda, RAM'e yakın doğruluk sonuçları elde edilmektedir. ViT, RAM'e göre eğitim süresi uzunluğu bakımından avantaj sağlamaktadır. Şekil 5.21'de Fashion-MNIST datasetinin RAM-ViT test kaybı karşılaştırmalı grafiği gösterilmektedir.



Şekil 5.21. Fashion-MNIST RAM-ViT test kayıp karşılaştırma grafiği.

Görüldüğü üzere ViT ile 40 iterasyonda, RAM'e yakın sonuçlar elde edilmektedir. ViT, RAM'e göre eğitim süresi uzunluğu bakımından avantaj sağlamaktadır.

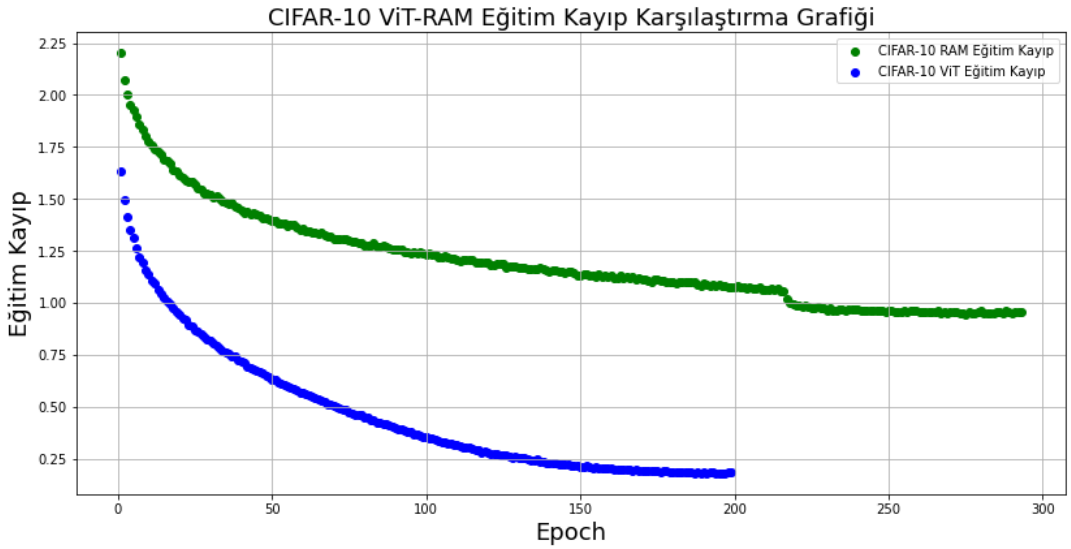
İterasyon sayısı arttırıldığında, test kayıp oranın yükselerek başlangıç değerine yaklaştığı gözlemlenmektedir. Bu nedenle 40 iterasyon yeterli görülmüştür. Şekil 5.22'de Fashion-MNIST datasetinin RAM-ViT test doğruluğu karşılaştırmalı grafiği gösterilmektedir.



Şekil 5.22. Fashion-MNIST RAM-ViT test doğruluğu karşılaştırma grafiği.

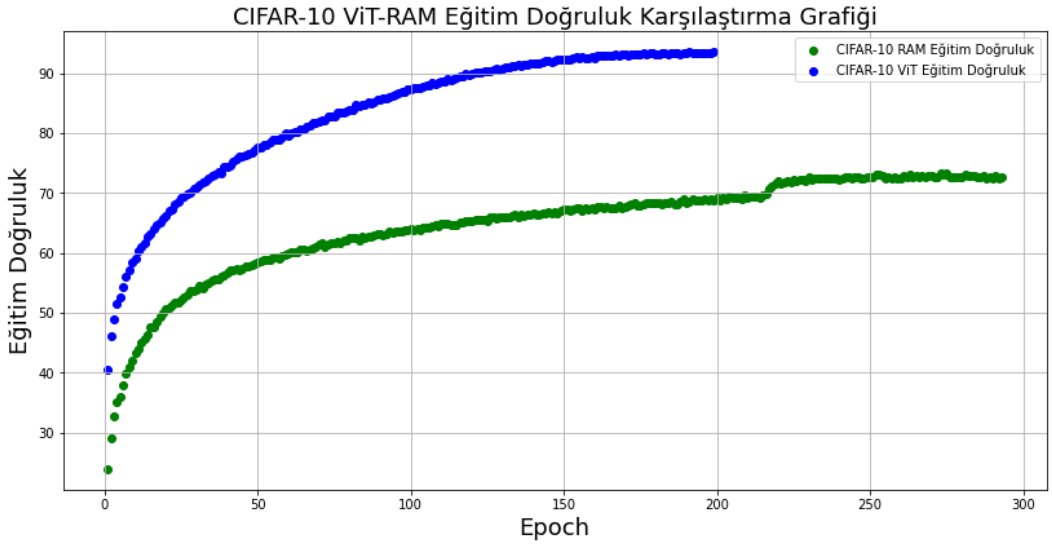
5.5.2. CIFAR-10 dataseti RAM-ViT sonuçlarının karşılaştırması

CIFAR-10 dataseti yapısı gereği (renkli) Fashion-MNIST datasete göre daha karmaşık bir settir. RAM-ViT sonuçlarının ayrımı bu set üzerinde daha açık bir şekilde gözükmektedir. ViT, daha düşük eğitim ve test kaybı, daha yüksek eğitim ve test doğruluğu sağlamaktadır. Şekil 5.23'te CIFAR-10 datasetinin RAM-ViT eğitim kaybı karşılaştırmalı grafiği gösterilmektedir.



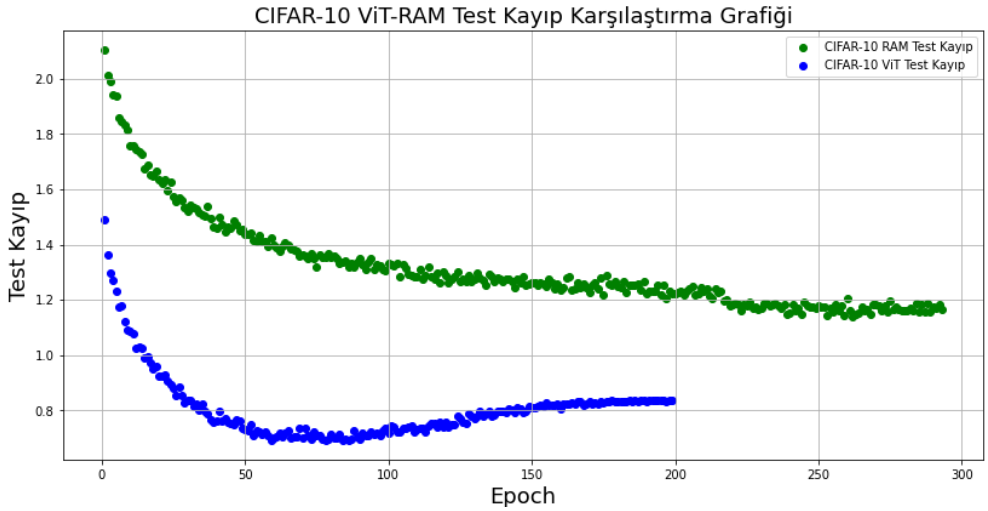
Şekil 5.23. CIFAR-10 RAM-ViT eğitim kayıp karşılaştırma grafiği.

Görüldüğü üzere ViT ile RAM'e göre daha hızlı bir biçimde, daha az kayıp yaşanmıştır. Şekil 5.24'te CIFAR-10 datasetinin RAM-ViT eğitim doğruluğu karşılaştırmalı grafiği gösterilmektedir.



Şekil 5.24. CIFAR-10 RAM-ViT eğitim doğruluğu karşılaştırma grafiği.

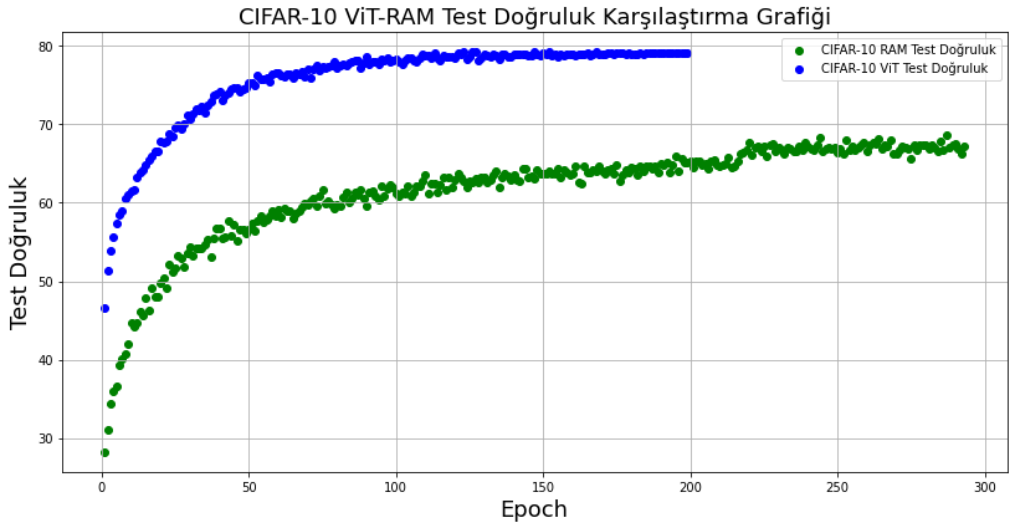
Görüldüğü üzere ViT ile RAM'e göre daha hızlı bir biçimde, daha yüksek doğruluk değeri elde edilmektedir. Şekil 5.25'te CIFAR-10 datasetinin RAM-ViT test kaybı karşılaştırmalı grafiği gösterilmektedir.



Şekil 5.25. CIFAR-10 RAM-ViT test kayıp karşılaştırma grafiği.

Görüldüğü üzere ViT ile 200 iterasyonda, RAM'den çok daha düşük kayıp değeri elde edilmektedir. ViT, RAM'e göre eğitim süresi uzunluğu bakımından avantaj sağlamaktadır.

İterasyon sayısı arttırıldığında, test kayıp oranın yükselerek başlangıç değerine yaklaştığı gözlemlenmektedir. Bu nedenle 200 iterasyon yeterli görülmüştür. Şekil 5.26'da CIFAR-10 datasetinin RAM-ViT test doğruluğu karşılaştırmalı grafiği gösterilmektedir.



Şekil 5.26. CIFAR-10 RAM-ViT test doğruluğu karşılaştırma grafiği.

Görüldüğü üzere ViT ile 200 iterasyonda, RAM'den çok daha yüksek doğruluk değeri elde edilmektedir. Hem eğitim süresi hemde doğruluk ve kayıp açısından ViT'in avantajı görülmektedir.

5.6. Eğitim Süreleri

ViT eğitimlerinin süresi RAM eğitimlerine göre daha uzun sürmektedir. Bu uzunluk RAM modelin RNN yapısından kaynaklanmaktadır. Yine CIFAR-10 datasetinin eğitimi de Fashion-MNIST datasetinin eğitimine göre hem epoch sayına hem de karmaşıklık düzeyine (gri-renkli görüntü) bağlı olarak daha uzun sürmüştür. Tablo 5.15'te gerçekleştirilen eğitimlerin süresi verilmektedir.

Tablo 5.15. Eğitim süreleri.

Dataset	Model	Süre
Fashion-MNIST	RAM	3.33 h
Fashion-MNIST	ViT	1,15 h
CIFAR-10	RAM	3,41 h
CIFAR-10	ViT	2,16 h

6. SONUÇ VE ÖNERİLER

Çalışmanın omurgasını yineleyen derin ağlar ve görsel dikkat modeli oluşturmaktadır. Yineleyen derin ağların önemli iki problemi göz önünde bulundurulmuş, dönüştürücü sinir ağlarının kullanıldığı bir başka dikkat tabanlı model olan görü dönüştürücünün, yineleyen derin ağlarla bir kıyaslaması yapılarak, nasıl bir tepki verdiğini gözlemleme çalışmanın odak noktası olarak seçilmiştir. ViT modeli birçok NLP çalışmasında sıklıkla kullanılmakla beraber, son zamanlarda görüntü sınıflandırma uygulamalarında da çeşitli çalışmalar gerçekleştirilmiştir. Dikkat tabanlı modeller olan RAM ve ViT modelleri incelenmiş, Fashion-MNIST ve CIFAR-10 datasetleri üzerinde eğitimler gerçekleştirilerek, sonuçları karşılaştırmalı grafikler ve tablolar halinde sunulmuştur. Bu datasetler PyTorch aracılığıyla direkt etkileşim sağlanabilen, her biri on ayrı görüntü sınıfından oluşan görüntü kümeleridir. RAM, giriş görüntüsünün tamamıyla uğraşmak yerine sadece ilgilenilen alana odaklanan limitli bir sensör ile bir dizi işlemden geçen bir pekiştirmeli öğrenme ajanının karar verme prosesi ve bu süreçlerden sonra davranışları sonucu aldığı ödülleri maksimize etme gayesine dayanırken; ViT ise giriş görüntülerinin parçalara ayırarak işleme sokan ve RNN'lerdeki eğitim süresinin uzunluğu, kaybolan gradyan problemleri gibi sorunlara alternatif bir çözüm önerisi olarak öne çıkan dönüştürücü ağları temel alan modellerdir. ViT, RAM'e göre daha yeni geliştirilmiş bir tasarımdır. ViT, yumuşak dikkatin bir alt sınıfı olan öz dikkat mekanizmasını kullanan bir dikkat modeliyken; yineleyen dikkat modelinde ise sert dikkat kullanılmaktadır. Yumuşak dikkat görüntünün farklı noktalarına odaklanarak işlem yaparken; sert dikkat ise bir bölüme dayalı olarak yapmaktadır. ViT, uygulamacılara paralel olarak girdileri işleme sokması ve girdilere bir konum numarası ekleyerek bilginin unutulma problemini aşması ile büyük bir avantaj sağlamaktadır. Deney sonucu elde edilen sonuçlara göre, ViT modelin hem eğitim uzunluğu hem de doğruluk ve kayıp oranları açısından RAM'e göre üstünlük gösterdiği anlaşılmaktadır. Fashion-MNIST datasetin gri görüntülerden oluşması nedeniyle gerçekleştirilen eğitim sonuçlarında büyük değişkenlikler söz konusu olmamıştır. Ancak eğitim süresi bakımından ViT modelin RAM modele üstünlük sağladığı gözlemlenmiştir.

Renkli görüntü yapısına sahip CIFAR-10 datasetle gerçekleştirilen eğitim sonuçlarında ise açık bir şekilde ViT modelin hem eğitim hem de doğruluk ve kayıp oranları bakımından RAM modele üstünlük sağladığı gözlemlenmiştir. ViT modelinde, büyük veri sayısına sahip datasetlerde önceden eğitim uygulanarak yapılan çalışmalarda başarı oranın yükseldiği bilinmektedir. Önceden eğitilmiş modeller kullanıldığında; doğruluk oranlarının artıp, kayıp oranlarının azalacağı öngörülmektedir. Yineleyen dikkat modeli, ilk çıktığı yıllarda elde ettiği sonuçlarla kullanıcılarının sonuçlardan memnun olduğu bir modeldir. Ancak verileri sıralı bir şekilde işlemesi nedeniyle eğitim süresinin uzun sürmesi en büyük dezavantajıdır. ViT mimarisi ise son üç yılın derin öğrenme uygulamalarında en popüler konularından birisi haline gelmiştir. Görüntü işleme, doğal dil işleme, biyomedikal veri işleme ve tıp alanlarında uygulanabilmesi, çok sınıflı çalışmalara uygunluğu, paralel işlem yapmasının eğitim süresi üzerindeki olumlu etkisi ve yüksek kapasitesi gibi avantajlı pek çok özelliğiyle transformer ağ mimarisi ön plana çıkmaktadır.

KAYNAKLAR

[1] Güneş, Ö. (2010). *Nitelik Tabanlı Nesne Sınıflandırması* [Yüksek Lisans tezi]. Deniz Harp Okulu.

[2] Balbozan, F. İ. (2011). *Kameralı Lazer Tarama Sistemi İle Nesne Sınıflandırması Ve Uygulamaları* [Yüksek Lisans tezi]. Dokuz Eylül Üniversitesi.

[3] Mundy, J. L., Zisserman, A. (1992). *Geometric invariance in computer vision*. MIT Press.

[4] Felzenszwalb, P., McAllester, D., Ramanan, D. (2008). A Discriminatively Trained, Multiscale, Deformable Part Model. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, USA, 1–8. doi: 10.1109/CVPR.2008.4587597.

[5] Codedamn (2022, 28 Ocak). What is Soft and Hard Attention Model in Computer Vision? https://codedamn.com/news/machine-learning/soft-vs-hard-attention-model-in-computer-vision adresinden 6 Mayıs 2022 tarihinde alınmıştır.

[6] Xu, B., Liu, J., Hou, X., Liu, B., Garibaldi, J., Ellis, I., Green, A., Shen, L., Qiu, G. (2019). Look, Investigate, and Classify: A Deep Hybrid Attention Method for Breast Cancer Classification. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019),* 914–918. doi: 10.1109/ISBI.2019.8759454.

[7] Savaş, S. (2020, 20 Mayıs). Sınıflandırma, Regresyon, Kümeleme ve Birliktelik Kuralları.https://medium.com/verimadencili%C4%9Fi/s%C4%B1n%C4%B1and%C4%B1rma-regresyonk%C3%BCmeleme-ve-birliktelikkurallar%C4%B1-e8ee1e47aeed adresinden 1 Kasım 2022 tarihinde alınmıştır.

[8] Akbulut, S. (2006). *Veri Madenciliği Teknikleri ile Bir Kozmetik Markanın Ayrılan Müşteri Analizi ve Müşteri Segmentasyonu* [Yüksek Lisans tezi]. Gazi Üniversitesi.

[9] Onar, A. (2020, 21 Eylül). Derin Öğrenme Yöntemleri ve Uygulamaları. https://alitunacanonar.medium.com/derin-%C3%B6%C4%9Frenme-y%C3%B6ntemleri-ve-uygulamalar%C4%B1-1ce215de40e8 adresinden 1 Kasım 2022 tarihinde alınmıştır.

[10] Yang, C. (2020). An Overview of the Attention Mechanisms in Computer Vision. *3rd International Conference on Computer Information Science and Artificial Intelligence,* 1693(1), 1-7. doi: 10.1088/1742-6596/1693/1/012173.

[11] Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K. (2015). Spatial Transformer Networks. *Advances in Neural Information Processing Systems 28 (NIPS 2015).* 1-15. https://doi.org/10.48550/arXiv.1506.02025.

[12] Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E. (2020). Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8), 2011-2023. doi: 10.1109/TPAMI.2019.2913372.

[13] Woo, S., Park, J., Lee, J., Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. *European Conference on Computer Vision (ECCV)*, Munich, 3-19. https://doi.org/10.1007/978-3-030-01234-2_1.

[14] Wang, X., Girshick, R., Gupta, A., He, K. (2017). Non-local Neural Networks. *Conference on Computer Vision and Pattern Recognition*, 7794-7803. doi:10.1109/CVPR.2018.00813.

[15] Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K. (2014). Recurrent models of visual attention. *Advances Neural Information Processsing Systems*, 3, 2204–2212. https://doi.org/10.48550/arXiv.1406.6247.

[16] Vakanski, A., Xian, M., Freer, P. E. (2020). Attention-Enriched Deep Learning Model for Breast Tumor Segmentation in Ultrasound Images. *Ultrasound in Medicine and Biology*, 46(10), 2819-2833. https://doi.org/10.1016/j.ultrasmedbio.2020.06.015.

[17] Ablavatski, A., Lu, S., Cai, J. (2017). Enriched deep recurrent visual attention model for multiple object recognition. *2017 IEEE Winter Conference on Applications Computer Vision*, 971–978. doi: 10.1109/WACV.2017.113.

[18] Shaikh, M., Kollerathu, V. A., Krishnamurthi, G. (2019). Recurrent Attention Mechanism Networks For Enhanced Classification Of Biomedical Images. *2019 IEEE 16th International Symposium on Biomedical Imaging*, 1260–1264. doi: 10.1109/ISBI.2019.8759214.

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkroeit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). Attention Is All You Need. 1–15. https://doi.org/10.48550/arXiv.1706.03762.

[20] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigord, G., Gelly, S., Uszkroeit, J., Houlsby, N. (2020). An Image Is Worth 16x16 Words: Transformers For Image Recognition At Scale. 1–22. https://doi.org/10.48550/arXiv.2010.11929.

[21] Tuncel, İ., Albayrak, A., Akın, M. (2022). Öz Dikkat Mekanizması Tabanlı Görü Dönüştürücü Kullanılarak Sıtma Parazit Tespiti. *DUJE (Dicle Univ. J. Eng.*, 13(2), 271–277. doi: 10.24012/dumf.1120289.

[22] Van Yüzüncü Yıl Üniversitesi (2022). Görme Olayı. http://fenbilgisiegitimi.yyu.edu.tr/k/groac/ adresinden 4 Kasım 2022 tarihinde alınmıştır.

[23] Okatan, A. (2021, 21 Aralık). Nasıl Görürüz?. https://bilimgenc.tubitak.gov.tr/makale/nasil-goruruz adresinden 4 Kasım 2022 tarihinde alınmıştır.

[24] Çetinkaya, T. S., Sertbaş, A. (2022). Derin Öğrenme Algoritmalarının GPU ve CPU Donanım Mimarileri Üzerinde Uygulanması ve Performans Analizi: Deneysel Araştırma. *Avrupa Bilim ve Teknoloji Dergisi*, 33, 10–19. doi: 10.31590/ejosat.937936.

[25] Türkoğlu, İ. (1996). *Yapay Sinir Ağları İle Nesne Tanıma* [Yüksek Lisans tezi]. Fırat Üniversitesi.

[26] Kara, A. (2020, 26 Eylül). Bilgisayarlı Görü (Computer Vision). https://www.datasciencearth.com/bilgisayarli-goru-computer-vision/ adresinden 4 Kasım 2022 tarihinde alınmıştır.

[27] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36 (4), 193–202. doi: 10.1007/BF00344251.

[28] Viola, P., Jones, M. (2001). Robust Real-time Object Detection. *Second International Workshop On Statistical And Computational Theories Of Vision – Modeling, Learning, Computing, And Sampling*, Vancouver (CANADA), 1–25.

[29] Akbay, Y. E. (2017). İdealar Teorisi Bağlamında Platon'da Akıl İlkelerinin Analizi. *Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 33(28), 133–155.

[30] Bayık, F. (2019). Aristoteles ve Descartes Bağlamında Akıl ve Zekâ Kavramlarının Farkları. *Kaygı. Uludağ Üniversitesi Fen-Edebiyat Fakültesi Felsefe Dergisi*, 18(1), 172–187. doi: 10.20981/kaygi.529827.

[31] Pirim, H. (2006). Yapay Zeka. *Yaşar Üniversitesi E-Dergisi*, 1(1), 81–93.

[32] Gürcan, S. (2019). Satranç ve Bilgisayar (Satranç Okulu). https://www.satrancokulu.com/yazilar/makaleler/satranc-ve-bilgisayar/ adresinden 5 Kasım 2022 tarihinde alınmıştır.

[33] Murat, S. (2021). *İnsansız Hava Aracı Görüntülerinden Derin Öğrenme Yöntemleriyle Nesne Tanıma* [Yüksek Lisans tezi]. Maltepe Üniversitesi.

[34] Mutludoğan, K. (2020). *Derin Öğrenme Tabanlı Şeffaf Nesne Tanıma* [Yüksek Lisans tezi]. Uludağ Üniversitesi.

[35] Turovsky, B. (2016, 15 Kasım). Found in translation: More accurate, fluent sentences in Google Translate. https://blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate/ adresinden 6 Aralık 2022 tarihinde alınmıştır.

[36] Microsoft (2022, 29 Kasım). Derin öğrenme ve makine öğrenmesi teknikleri. https://learn.microsoft.com/tr-tr/azure/machine-learning/concept-deep-learning-vs-machine-learning adresinden 6 Aralık 2022 tarihinde alınmıştır.

[37] Kızrak, M. A., Bolat, B. (2018). Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma. *Bilişim Teknolojileri Dergisi*, 11(3), 263–286. doi: 10.17671/gazibtd.419205.

[38] Sinir Hücresi (Nöron) Nedir? (Yapısı, Özellikleri ve Çeşitleri) (2022). https://ogreniyo.com/sinir-hucresi/ adresinden 6 Aralık 2022 tarihinde alınmıştır.

[39] Gülcü, A., Kuş, Z. (2019). Konvolüsyonel Sinir Ağlarında Hiper-Parametre Optimizasyonu Yöntemlerinin İncelenmesi. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C Tasarım ve Teknoloji*, 7(2), 503–522. doi: 10.29109/gujsc.514483.

[40] Kayali, N. Z., Omurca, S. İ. (2021). Konvolüsyonel Sinir Ağları (CNN) ile Çin Sayı Örüntülerinin Sınıflandırması. *Journal of Computer Science*, 184–191. https://doi.org/10.53070/bbd.989668.

[41] Karahan, T., Nabiyev, V. (2021). Plant identification with convolutional neural networks and transfer learning. *Pamukkale University Journal Engineering Sciences*, 27(5), 638–645. doi: 10.5505/pajes.2020.84042.

[42] Kunt, T. (2021). Evrişimli Ve Yinelemeli Sinir Ağları İle Görüntülere Başlık Atama [Yüksek Lisans tezi]. Gazi Üniversitesi.

[43] Zhang, A., Lipton, Z. C., Li, M., Smola, A. J. (2022). *Derin Öğrenmeye Dalış*.

[44] Gül, M. (2021). *Taşınabilir Yürütülebilir Dosyalarda Yinelenen Sinir Ağlarını Kullanarak Statik Kötü Amaçlı Yazılım Algılama* [Yüksek Lisans tezi]. Mersin Üniversitesi.

[45] Erciyes, N. E. (2022). *Deep Learning Methods With Pre-Trained Word Embeddings And Pre-Trained Transformers For Extreme Multi Label Text Classification* [Yüksek Lisans tezi]. Çankaya Üniversitesi.

[46] Sowmya, B. J., Chetan, Srinivasa, K. G. (2016). Large scale multi-label text classification of a hierarchical dataset using Rocchio algorithm. *2016 International Conference Computer Sytsem and Information Technology for Sustainable Solutions (CSITSS)*, Bengalaru, INDIA, 291–296. doi: 10.1109/CSITSS.2016.7779373.

[47] Süzen, A. A., Yıldız, Z., Yılmaz, T. (2019). LSTM Tabanlı Derin Sinir Ağı ile Ayak Taban Basınç Verilerinden VKİ Durumlarının Sınıflandırılması. *BEÜ Fen Bilimleri Dergisi*, 8(4), 1392-1398. https://doi.org/10.17798/bitlisfen.540273.

[48] Yalman, H. İ., Tüfekçi Z. (2022). Konuşma Tanımaya Uygulanan BiRNN, BiLSTM ve BiGRU Modellerinin Performans Değerlendirmesi. *Avrupa Bilim ve Teknoloji Dergisi*, 36, 121-127. doi: 10.31590/ejosat.1111314.

[49] Çarkacı, N. (2018, 22 Ocak). Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler. https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4 adresinden 21 Aralık 2022 tarihinde alınmıştır.

[50] Agarap, A. F. M. (2019). Deep Learning using Rectified Linear Units (ReLU). 1–7. https://doi.org/10.48550/arXiv.1803.08375.

[51] Karukuş, B. A. (2018, 17 Nisan). Derin Sinir Ağları için Aktivasyon Fonksiyonları. http://buyukveri.firat.edu.tr/2018/04/17/derin-sinir-aglari-icin-aktivasyon-fonksiyonlari/ adresinden 21 Aralık 2022 tarihinde alınmıştır.

[52] Hendrycks, D., Gimpel, K. (2016). Gaussian Error Linear Units (GELUs). 1–9. https://doi.org/10.48550/arXiv.1606.08415.

[53] Poulinakis, K. (2022, 30 Ağustos). GELU, the ReLU Successor? Gaussian Error Linear Unit Explained. https://pub.towardsai.net/is-gelu-the-relu-successor-deep-learning-activations-7506cf96724f adresinden 31 Mart 2023 tarihinde alınmıştır.

[54] Gülüm, S. (2021, 16 Mart). Softmax: Bir Etkinleştirme İşlevi. https://medium.com/deeper-deep-learning-tr/softmax-bir-aktivasyon-fonksiyonu-da8382d8a281 adresinden 31 Mart 2023 tarihinde alınmıştır.

[55] Doruk, A. E. (2021, 1 Şubat). MXNet ile Derin Öğrenme 2.1: Softmax Regresyon (Teori). https://www.veribilimiokulu.com/mxnet-ile-derin-ogrenme-2-1-softmax-regresyon-teori/ adresinden 31 Mart 2023 tarihinde alınmıştır.

[56] Koech, K. E. (2020, 30 Eylül). Softmax Aktivasyon İşlevi — Aslında Nasıl Çalışır? https://towardsdatascience.com/softmax-activation-function-how-it-actually-works-d292d335bd78 adresinden 31 Mart 2023 tarihinde alınmıştır.

[57] Tekin, B. Y. (2021, 7 Şubat). Keras Loss Fonksiyonları. https://medium.com/operations-management-t%C3%BCrkiye/keras-loss-fonksiyonlar%C4%B1-2955e86a9e07 1 Nisan 2023 tarihinde alınmıştır.

[58] Bircanoğlu, C. (2017). A Comparison of Loss Function in Deep Embedding. [Yüksek Lisans tezi]. Bahçeşehir University.

[59] Golik, P., Doetsch, P., Ney, H., (2013). Cross-entropy vs. Squared error training: A theoretical and experimental comparison. *Proceedings of the Annual Conference of 60 the International Speech Communication Association*, 2(2), 1756–1760. doi:10.21437/Interspeech.2013-436.

[60] Devreyakan (2021, 5 Ağustos). Yitim Fonksiyonları Nedir? https://devreyakan.com/yitim-fonksiyonlari-nedir/ adresinden 1 Nisan 2023 tarihinde alınmıştır.

[61] Ser, G., Bati, C. T. (2019). Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması. *YYU Journal of Agricultural Science*, 29(3), 406–417. https://doi.org/10.29133/yyutbd.505086.

[62] Atcılı, A. (2020, 6 Kasım). Yapay Sinir Aglarinda Kullanilan Optimizasyon Algoritmalari. https://medium.com/machine-learning-türkiye/yapay-sinir-ağlarında-kullanılan-optimizasyon-algoritmaları-3e87cd738cb5#:~:text=Adam Optimizasyon Algoritması adresinden 1 Nisan 2023 tarihinde alınmıştır.

[63] Deep RL Course (2020). Welcome to Deep Reinforcement Learning. https://huggingface.co/learn/deep-rl-course/unit0/introduction?fw=pt adresinden 1 Nisan 2023 tarihinde alınmıştır.

[64] Aydın, B. M. (2022). *Pekiştirmeli Öğrenme Yöntemi İle Optimal DC Motor Hız Kontrolcüsünün Tasarlanması* [Yüksek Lisans tezi]. Sakarya Üniversitesi.

[65] Sutton, R., Barto, A. (2019, 1 Ocak). Pekiştirmeli Öğrenme. https://yz-ai.github.io/blog/pekistirmeli-ogrenme/pekistirmeli-ogrenme-bolum-1 adresinden 5 Aralık 2022 tarihinde alınmıştır.

[66] Torres, J. (2020, 10 Eylül). Policy-Gradient Methods. https://towardsdatascience.com/policy-gradient-methods-104c783251e0 adresinden 11 Nisan 2023 tarihinde alınmıştır.

[67] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Appears In: Machine learning*, 8(3-4), 229–256. https://doi.org/10.1007/BF00992696.

[68] Torres, J. (2020, 22 Temmuz). Monte Carlo Yöntemleri. https://towardsdatascience.com/monte-carlo-methods-9b289f030c2e adresinden 11 Nisan 2023 tarihinde alınmıştır.

[69] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bal, J., Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. 1–12. https://doi.org/10.48550/arXiv.1912.01703.

[70] TalentGrid (2021,10 Haziran). PyTorch Nedir? PyTorch ile Derin Öğrenmeye Giriş. https://talentgrid.io/tr/pytorch-nedir/ adresinden 12 Nisan 2022 tarihinde alınmıştır.

[71] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine Learning*, 229–256.

[72] Sajil, C. K. (2022). Discount Factor in Reinforcement Learning. https://intuitivetutorial.com/2020/11/15/discount-factor/ adresinden 8 Aralık 2022 tarihinde alınmıştır.

[73] Gheflati, B., Rivaz, H. (2021). Vision Transformers For Classification Of Breast Ultrasound Images. *Computer Vision and Pattern Recognition*, CANADA, 1–5. https://doi.org/10.48550/arXiv.2110.14731

[74] Karagüler, C. (2022). *Hierarchical Image Classification With Self-Supervised Vision Transformer Features* [Yüksek Lisans tezi]. Graduate School İzmir Institute of Technology.

[75] Ankit, U. (2022, 28 Haziran). Transformer Neural Networks: A Step-by-Step Breakdown. https://builtin.com/artificial-intelligence/transformer-neural-network adresinden 19 Aralık 2022 tarihinde alınmıştır.

[76] CIFAR-10. https://www.cs.toronto.edu/~kriz/cifar.html adresinden 20 Aralık 2022 tarihinde alınmıştır.

[77] Fashion-MNIST Dataset. https://github.com/zalandoresearch/fashion-mnist adresinden 20 Aralık 2022 tarihinde alınmıştır.

[78] Fashion-MNIST. https://www.kaggle.com/datasets/zalando-research/fashionmnist adresinden 20 Aralık 2022 tarihinde alınmıştır.

[79] Amidi, A., Amidi, S. (2022). Derin Öğrenme Püf Noktaları Ve İpuçları El Kitabı. https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-deep-learning-tips-and-tricks adresinden 21 Aralık 2022 tarihinde alınmıştır.

ÖZGEÇMİŞ

Ad-Soyad :Oğuzhan Bubo

ÖĞRENİM DURUMU:

- **Lisans :** 2020, Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü
- **Yükseklisans :** Devam Ediyor, Sakarya Üniversitesi, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektrik Mühendisliği Programı

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2019-TÜBİTAK - İntern

TEZDEN TÜRETİLEN ESERLER:

- Bubo, O., Baraklı, B. (2023, 04-06, Şubat). Yineleyen Derin Ağ ve Görsel Dikkat Modeli İle Nesne Tanıma. *8th International Baskent Congress on Physical, Engineering, And Applied Sciences - BZT Academy*, Ankara, Turkey. (Bildiri)
  .