



Evrişimsel Sinir Ağları İle Rekabetçi Derin Öğrenme

Program Kodu:1001

Proje No:118E293

Proje Yürütücüsü:

Doç. Dr. MUSTAFA TANER ESKİL

EYLÜL 2021

ANKARA

Önsöz

Bu projede sezgiye dayalı; insan görme korteksinin öğrenmesinden esinlenen; literatürdeki yöntemlerin aksine sığ katmanlardan derin katmanlara, basitten karmaşığa kavramsal olarak anlamlı öznitelikler çıkarabilen; yine bu yöntemlere kıyasla çok daha basit bir mimari ve az sayıda sinir ile sınıflandırma ve kusur tespiti yapabilen; dolayısı ile eğitim ve test süreçlerinde çok hızlı; evrimsel katmanları tamamen gözetimsiz eğitilen bir derin öğrenme yöntemi önerilmiştir. Yöntemimiz, sınıflandırma ve kusur tespiti problemlerinde denenmiş ve sonuçlar raporlanmıştır. Projemiz TÜBİTAK ARDEB tarafından desteklenmiştir.

İçindekiler

1. Giriş.....	1
2. Literatür Özeti	3
3. Gereç ve Yöntem	6
3.1. Rekabetçi Derin Öğrenme ile Gözetimsiz Öğrenme ve Sınıflandırma.....	6
3.1.1. Neocognitron.....	7
3.1.2. Evrişimsel Özniteliklerin Keşfi ve Sinir Oluşturma	7
3.1.3. Süzgeç Adaylarının Çıkarımı	8
3.1.4. Gözetimsiz Eğitim	8
3.2. Rekabetçi Derin Öğrenme ile Gözetimsiz Kusur Tespiti	12
3.2.1. Ön İşleme.....	13
3.2.2. Süzgeç Adaylarının Çıkarılması	13
3.2.3. Süzgeç Seçimi	14
3.2.4. Kusur Tespiti	14
3.2.5. Kusur Olasılık Matrisinin Bölütlenmesi	16
4. Bulgular	17
4.1. Rekabetçi Derin Öğrenme ile Sınıflandırma.....	17
4.1.1. Deneysel Kurgu	17
4.1.2. MNIST	17
4.1.3. CIFAR-10	19
4.1.4. Tiny ImageNet.....	21
4.2. Rekabetçi Derin Öğrenme ile Kusur Denetimi	22
4.2.1. BRATS Veri Kümesi.....	24
5. İş Paketleri.....	28
5.1. Öznitelik Öğrenme İş Paketi.....	28
5.2. Sınıflandırma İş Paketi	34
5.3. Topaklama ve Sıra Dışı Örnek Tespiti İş Paketi	35
6. Tartışma.....	36
7. Sonuç ve Öneriler.....	37

Tablolar

Tablo 1. MNIST deneylerimizde kullandığımız RDÖ modelleri	18
Tablo 2. MNIST veri kümesi üzerinde her model için katmanlarda kullanılan benzerlik eşikleri, çıkarılan 5x5 süzgeç sayısı ve sınıflandırma doğrulukları	19
Tablo 3. MNIST veri kümesi üzerinde B Modeli için karıştırma matrisi	19
Tablo 4. CIFAR-10 veri kümesi üzerinde her model için katmanlarda kullanılan benzerlik eşikleri, çıkarılan 3x3 süzgeç sayısı ve sınıflandırma doğrulukları	20
Tablo 5. CIFAR-10 veri kümesi üzerinde B Modeli için karıştırma matrisi.	21
Tablo 6. Tiny ImageNet deneylerimizde kullandığımız RDÖ modelleri	22
Tablo 7. Tiny ImageNet veri kümesi üzerinde her model için katmanlarda kullanılan benzerlik eşikleri, çıkarılan 3x3 süzgeç sayısı ve sınıflandırma doğrulukları	23
Tablo 8. RDÖ ile kusur tespitinin literatürdeki farklı yöntemler ile karşılaştırması	28
Tablo 9. MNIST veri kümesinde farklı benzeme eşikleri ile elde edilen sinir sayısı	29
Tablo 10. MNIST veri kümesinde evrişim aşamasında boyut indirgenmesi ile elde edilen öznetelik sayıları ve ortalama başarımlar	29
Tablo 11. MNIST veri seti üzerinde rastlantısal sinir ilklendirmesi kullanılarak elde edilen öznetelik sayıları ve başarımlar	31

Şekiller

Şekil 1. Önerilen RDÖ gözetimsiz eğitim algoritması.....	11
Şekil 2. RDÖ ile kusur tespitinin şema halinde gösterimi.....	12
Şekil 3. BRATS veri kümesinde farklı kiplerde veri örnekleri (T1, T1c, T2, FLAIR).....	25
Şekil 4. Bir girdi örneği ve ilk katmanda elde edilen öznitelikler (süzgeç ağırlıkları).....	26
Şekil 5. RDÖ ile tümör tespiti	27
Şekil 6. MNIST veri seti ile 0,5 benzeme eşiği ile eğitilen RDÖ mimarisinin 4. ve son katmanından elde edilen sınırları 0,95 eşik değerinin üzerinde ve en çok uyaran girdiler	31
Şekil 7. Birinci katmanda seçilen 6 sinir için destekçilerin (en yüksek uyaranların) sınıf dağılımları	33

Özet

İmge üzerinde nesne tanımadada, nesnenin imge üzerindeki yeri, açısız yönelimi ve ölçęęi algoritmik olarak çözümlenmesi gereken önemli zorluklardır. Bu zorlukları aşmak için son yıllarda en sık kullanılan ve en başarılı sonuçları veren yaklaşım derin öğrenme ve evrişimsel yapay sinir ağlarıdır (EYSA). EYSA yaklaşımında katmanlar halinde evrişime dayalı süzgeçleme yapılır. Her katman, bir önceki katmandan gelen öznitelik haritasını (feature map) kullanarak daha karmaşık öznitelikler elde eder. Çoğunlukla son katmandan elde edilen en soyut öznitelik haritası tam baęlı (fully connected) bir yapay sinir ağının girdisini oluşturur.

EYSA yaklaşımının en kritik aşaması süzgeçlerin algoritmik öğrenilmesidir. Bu görev için aęırlıklı olarak gradyan temelli geri yayılım (backpropagation) öğrenme algoritması kullanılmaktadır. Gradyan temelli yaklaşımlarda genellikle katman sayısı ile katmanlardaki sinir sayıları önceden tespit edilir ve bu sinirlerin girdi aęırlıkları ilklendirilerek arama uzayında başlangıç noktası sabitlenir. Eğitim, sınıflandırmanın yapıldığı en son katmandan ilk katmana doğru, sinirler arası baęlantılar düzenlenerek yapılır.

Elle oluşturulan EYSA mimarisinin detayları hem veri kümesinin ne kadar zorlu olduğuna, hem de veri kümesindeki örneklerin boyut sayısına baęlıdır. Gradyana dayalı eğitimde arama uzayının büyüklüğü, verinin boyut sayısı ile üstel ilintilidir. Bu durum, pek çok problemde doğru sayıda katman ve sinir oluşturulmasını ve bu sınırlara uygun ilk deęerler atanmasını zorlaştırır. Eğitim sürecinde sınıflandırma hatasının girdi saęlayan sıę katman sinirlerine doğru paylaştırılması zordur. Bunlara ek olarak, hatanın eniyilenmesi gradyan üzerinde gerçekleştirildiği için sinir aęırlıkları ancak küçük adımlarla güncellenebilir. Bu durum imge örnekleri üzerinde çok sayıda epok yapılmasını zorunlu kılar. Dolayısı ile gradyan temelli yaklaşımlarda (1) başarı rastgele ilklendirmeye baęlıdır, (2) aşamalar kredi atama problemine (credit assignment problem) açıktır ve (3) eğitim, örnekler üzerinde çok sayıda tekrar gerektirmesi nedeniyle yavaştır.

Bu projede, EYSA mimarisi için gradyan temelli olmayan, gözetimsiz bir makine öğrenmesi yöntemi önerilmiştir. Önerilen rekabetçi derin öğrenme (RDÖ) yaklaşımı her katmanda sinirler arası rekabete dayalıdır. Yöntemimizde eğitim, katmanlar halinde yapılır. Her katman boş olarak başlar. İlk eğitim örneęi özgün bir öznitelik kabul edilerek süzgeç haline getirilir ve katmanın tek siniri olarak ilklendirilir. Yeni eğitim örneęi için katmana ait sinirlerin çıktıları hesaplanır. Belli bir eşiğin üzerinde etkinleşen sinirler arasında en yüksek çıktıya sahip sinir kazanan olarak seçilir ve öğrenme sadece bu sinir üzerinde, eğitim örneęine doğru evrilerek olur. Böyle bir sinir mevcut deęilse yeni bir sinir oluşturulur. Tüm eğitim örnekleri işlendiğinde,

katmanda eğitim kümesindeki sınıf içi ve sınıflar arası değişiminin temsili için yeterli sayıda sinir oluşturulmuş olur.

Sinirlerin, eğitim sırasında gerekikçe ve eğitim örneklerine yakınsayacak şekilde oluşturulması yoluyla EYSA'dan iki önemli hiperparametreyi; sinir sayısı ve sinir ağırlıklarının ilklendirilmesini kaldırdık. Sınıflandırma aşaması için en derin katmanda öğrenilen karmaşık özniteliklerin çıktılarını tam bağlı bir sinir ağına girdi olarak kullanarak gözetimli öğrenme gerçekleştirdik. Bu sayede, sığ evrışimsel katmanlarda kredi (hata) atama ve istikrarsız gradyanlar problemlerini ortadan kaldırmış olduk. Sinirlerin rastgele değil, eğitim örneklerine dayalı olarak oluşturulması ile evrışimsel katmanlarda öznitelik çıkarımını her bir katman için bir epokta tamamladık. Katmanlarda rastgele ve gereğinden fazla sinir oluşturulmaması; sinirlerin eğitim örneklerine dayalı olarak oluşturulması sayesinde az sayıda tekrar yapılması; bunlara ek olarak öznitelik çıkarımında geri yayılım aşamasının olmaması eğitim sürecimizi kayda değer oranda hızlandırdı. Önerdiğimiz RDÖ yöntemini, literatürde sık kullanılan MNIST, CIFAR-10 ve Tiny ImageNet veri kümeleri üzerinde sınıflandırma deneyleri ve BRATS veri kümesi üzerinde tümör bulma problemi üzerinde denedik ve sonuçlarımızı raporladık.

Abstract

Varying location, orientation and scale are the biggest challenges in object recognition with image processing. The most popular approaches to tackle these challenges are deep learning and Convolutional Neural Networks (CNN) where filtering is done in convolutional layers that scan the entire image. Each layer of a CNN processes the outputs of the previous layer to extract more and more abstract and descriptive features. Classification is performed on the outputs of the last feature extraction layer through conventional supervised learning.

The most critical stage of the CNN approach is algorithmic training of the filters in the hidden layers. One way to learn the filters is the gradient based backpropagation (BP) algorithm. Gradient based algorithms typically require the number of layers to be fixed and initialize a preset number of neurons with random weights, fixing the starting point in the search space. BP starts with the outputs of the last layer and progresses backwards as it distributes the credit (or error) towards the first layer, updating weights layer-wise.

Both the number of layers and neurons to be initialized in layers depend on the complexity of the data set and the dimensionality of the samples. This dependency makes it unpractical to initialize a correct number of hidden neurons with suitable weights for most problems. Distribution of classification error to neurons is a challenge as each neuron has multiple inputs. Error minimization is performed on a gradient; therefore weight updates can only be made with small steps. As a result, gradient based methods require repetitive exposure of training samples until convergence. Consequently in gradient based methods; (1) the success depends on random initialization, (2) learning stages are vulnerable to the credit assignment problem and (3) training is slow as it requires numerous epochs on samples.

In this project we proposed an unsupervised learning algorithm to be used with the CNN architecture. In the proposed competitive deep learning (CDL) algorithm, neurons of a layer compete with each other to become a winner. The training of the network is performed layerwise and each layer starts out empty. The first training sample is considered a novel feature, which is converted into a filter and initialized as the only neuron of the layer. In case the layer is not empty, we calculate the outputs of the neurons for the input. The neuron that is stimulated the most and above a given similarity threshold is considered the winner and its weights are updated to evolve towards the input. When no such neuron exists, a new neuron will be initialized for this novel input. Therefore at the end of the training, a required and enough number of neurons are generated and initialized for representing the variance in the training set.

A new neuron is generated only when the layer does not embody a neuron that is a representative of the given sample. This eliminates two important hyperparameters from CNN; the number of neurons and initialization of neuron weights. We performed classification by feeding the outputs of the deepest, most abstract and representative neurons as input to a fully connected network, which is to be trained in a supervised manner. Consequently, we eliminated the credit assignment and vanishing or exploding gradients problem in the shallow convolutional layers. Generation of new neurons based on the training examples significantly speeds up the convergence rate of the neuron weights and training is usually completed in a single epoch. Creation of exactly enough number of neurons and their targeted initialization together with the lack of back propagation significantly sped up our training process. We experimented on the popular MNIST, CIFAR-10 and Tiny ImageNet datasets to assess the classification performance; and on BRATS dataset to assess the tumor detection performance of the proposed algorithm.

1. Giriş

İmge üzerinde nesne tanımadada, nesnenin imge üzerindeki yeri, açısal yönelimi ve ölçeği algoritmik olarak çözülmesi gereken önemli zorluklardır. Bu zorlukları aşmak için son yıllarda en sık kullanılan ve en başarılı sonuçları veren yaklaşım derin öğrenme ve evrimsel yapay sinir ağlarıdır (EYSA). EYSA yaklaşımında katmanlar halinde evrime dayalı süzgeçleme yapılır. Her katman, bir önceki katmandan gelen öznitelik haritasını (feature map) kullanarak daha karmaşık öznitelikler elde eder. Çoğunlukla son katmandan elde edilen en soyut öznitelik haritası tam bağlı (fully connected) bir yapay sinir ağının girdisini oluşturur.

EYSA yaklaşımının en kritik aşaması süzgeçlerin algoritmik öğrenilmesidir. Bu görev için ağırlıklı olarak gradyan temelli geri yayılım (backpropagation) öğrenme algoritması kullanılmaktadır. Gradyan temelli yaklaşımlarda genellikle katman sayısı ile katmanlardaki sinir sayıları önceden tespit edilir ve bu sinirlerin girdi ağırlıkları ilklendirilerek arama uzayında başlangıç noktası sabitlenir. Eğitim, sınıflandırmanın yapıldığı en son katmandan ilk katmana doğru, sinirler arası bağlantılar düzenlenerek yapılır.

Elle oluşturulan EYSA mimarisinin detayları hem veri kümesinin ne kadar zorlu olduğuna, hem de veri kümesindeki örneklerin boyut sayısına bağlıdır. Gradyana dayalı eğitimde arama uzayının büyüklüğü, verinin boyut sayısı ile üstel ilintilidir. Bu durum, pek çok problemde doğru sayıda katman ve sinir oluşturulmasını ve bu sinirlere uygun ilk değerler atanmasını zorlaştırır. Eğitim sürecinde sınıflandırma hatasının girdi sağlayan sığ katman sinirlerine doğru paylaştırılması zordur. Bunlara ek olarak, hatanın eniyilenmesi gradyan üzerinde gerçekleştirildiği için sinir ağırlıkları ancak küçük adımlarla güncellenebilir. Bu durum imge örnekleri üzerinde çok sayıda epok yapılmasını zorunlu kılar. Dolayısı ile gradyan temelli yaklaşımlarda (1) başarı rastgele ilklendirmeye bağlıdır, (2) aşamalar kredi atama problemine (credit assignment problem) açıktır ve (3) eğitim, örnekler üzerinde çok sayıda tekrar gerektirmesi nedeniyle yavaştır.

Bu projede, EYSA mimarisi için gradyan temelli olmayan, gözetimsiz bir makine öğrenmesi yöntemi önerilmiştir. Önerilen rekabetçi derin öğrenme (RDÖ) yaklaşımı her katmanda sinirler arası rekabete dayalıdır. Yöntemimizde eğitim, katmanlar halinde yapılır. Her katman boş olarak başlar. İlk eğitim örneği özgün bir öznitelik kabul edilerek süzgeç haline getirilir ve katmanın tek siniri olarak ilklendirilir. Yeni eğitim örneği için katmana ait sinirlerin çıktıları hesaplanır. Belli bir eşğin üzerinde etkinleşen sinirler arasında en yüksek çıktıya sahip sinir kazanan olarak seçilir ve öğrenme sadece bu sinir üzerinde, eğitim örneğine doğru evrilerek olur. Böyle bir sinir mevcut değilse yeni bir sinir oluşturulur. Tüm eğitim örnekleri

işlendiğinde, katmanda eğitim kümesindeki sınıf içi ve sınıflar arası değişiminin temsili için yeterli sayıda sınır oluşturulmuş olur.

Sinirlerin, eğitim sırasında gerektiğinde ve eğitim örneklerine yakınsayacak şekilde oluşturulması yoluyla EYSA'dan iki önemli hiperparametreyi; sınır sayısı ve sınır ağırlıklarının ilklendirilmesini kaldırdık. Sınıflandırma aşaması için en derin katmanda öğrenilen karmaşık özniteliklerin çıktılarını tam bağlı bir sınır ağına girdi olarak kullanarak gözetimli öğrenme gerçekleştirdik. Bu sayede, sığ evrimsel katmanlarda kredi (hata) atama ve istikrarsız gradyanlar problemlerini ortadan kaldırmış olduk. Sınirlerin rastgele değil, eğitim örneklerine dayalı olarak oluşturulması ile evrimsel katmanlarda öznitelik çıkarımını her bir katman için bir epokta tamamladık. Katmanlarda rastgele ve gereğinden fazla sınır oluşturulmaması; sinirlerin eğitim örneklerine dayalı olarak oluşturulması sayesinde az sayıda tekrar yapılması; bunlara ek olarak öznitelik çıkarımında geri yayılım aşamasının olmaması eğitim sürecimizi kayda değer oranda hızlandırdı.

RDÖ yönteminde katmanlarda eğitim kümesinde karşılaşılan örüntüleri temsil eden ve öznitelik olarak kullanılabilen sinirler oluşur. Sığ katmanlarda kenar, köşe gibi basit, derin katmanlarda nesne parçaları, dokular gibi karmaşık öznitelikler elde edilir. Bir sinirin temsil ettiği topağa (cluster) eğitim sırasında girdi veren örnek sayısı sinire ait destekçi sayısı olarak adlandırılır. Düşük destekçi sayısı, ilgili örüntünün eğitim kümesinde nadiren görüldüğü, diğer bir deyişle sıra dışı olduğu anlamına gelir. Eğitimin tamamlanmasından sonra verilen bir sınama örneğinin, bir katmanda hiçbir bir siniri etkinleştirememesi veya destekçi sayısı düşük bir siniri etkinleştirmesi olağan dışı kabul edilerek uyarı oluşturulabilir. Bu yaklaşımın biyomedikal görüntülerde tümör tespiti, tekstilde kusur bulma ve eğitim kümesinde hata bulma (doğru kesilmemiş imgeler gibi) problemlerinde etkin olarak kullanılabileceğini gösterdik.

Bu rapor, EYSA odaklı olarak yapay sinir ağlarının (YSA) 1940'lerden başlayan gelişimini ve bu alandaki zorlukları tartışan Literatür Özeti ile başlayacaktır. Gereç ve Yöntem'de RDÖ ile gözetimsiz öğrenme, sınıflandırma ve kusur bulma yöntemimiz detaylandırılacak, Bulgular'da deney kurgumuz ve deneysel sonuçlarımız paylaşılacaktır. Tartışma bölümünde başarılı ve başarısız olduğumuz alanlar ele alınacak, nedenleri üzerine analiz yapılacak ve çözüm önerileri üzerinde durulacaktır. Son bölüm olan Sonuç ve Öneriler'de projenin çıktıları ve kazanımlarımız paylaşılacaktır. Kaynakça, ayrı bir doküman olarak sunulmuştur.

2. Literatür özeti

YSA'ların tarihi 1943 yılına kadar uzanır. McCulloch ve Pitts (1943) tarafından önerilen ilk YSA değişmez ağırlıklara sahipti. Dolayısı ile eğitilebilir değildi. Gözetimsiz eğitilebilen ilk YSA Hebb (1949) tarafından önerilmiştir. Biyolojik süreçlerden ilham alan Hebb'in öğrenme algoritması aynı anda aktif olan iki sinirin arasındaki bağlantının güçlendirilmesine dayalıydı. Sonraki yıllarda basit sinir ağları için farklı gözetimsiz (Rosenblatt 1958; Rosenblatt 1962; Widrow ve Hoff 1962; Palm 1980; Hopfield 1988) ve gözetimli (Kohonen 1972; Malsburg 1973; Willshaw ve Malsburg 1976) öğrenme algoritmaları önerildi. Minsky ve Papert (1969), doğrusal olarak ayrıştırılması mümkün olmayan sınıfların tek katmanlı perseptron ile sınıflandırılmayacağını gösterdi. ANN araştırmalarına hız kazandıran Linnainmaa'nın (1970) bir yüksek lisans tezi ile ortaya koyduğu geri yayılım öğrenme yöntemi oldu. Geri yayılımın önerilmesi birden fazla katmanlı YSA'ların eğitimini, dolayısı ile çok katmanlı mimariyi mümkün kıldı.

Geriyayılım, bir gözetimli öğrenme algoritmasıdır. Bu yöntemde her eğitim örneği sinir ağına işlenerek ağına çıktıları ve sonuç olarak sınıflandırma hatası elde edilir. Öğrenme en son katmandan başlar. Hata geriyayılımla önceki katmanlara dağıtılarak sinirler arası bağlantılar düzenlenir. Bu öğrenme yöntemi 1980'li yıllarda çok sayıda veri kümesi üzerinde başarı ile uygulandı. Ancak 1980'li yılların sonunda geriyayılım yönteminin iki gizli katmandan daha derin mimarilerde beklenen oranda başarılı olamadığı gözlemlendi. Bu başarısızlığın temel nedeni geriyayılım yönteminde istikrarsız gradyanların (vanishing and exploding gradients) oluşabilmesidir (Hochreiter 1991). Teorik olarak yeterli sayıda elemana sahip tek gizli katmanlı bir YSA tüm çok terimli sürekli fonksiyonları kestirebilir (Kolmogorov 1957). Diğer taraftan, girdi boyutu arttıkça doğru sayıda sinirin belirlenmesi ve uygun değerlerle ilklendirilmesi zorlaşır (Haykin 2008).

YSA araştırmalarındaki bu duraklama, artan işlemci (CPU ve GPU) hızları, genişleyen hafıza ve EYSA (Fukushima 1980; LeCun vd. 1998; Fukushima 2013a) ile aşıldı. Bu yaklaşımda sinir ağı evrimsel birimler olan süzgeçlerden oluşur. Tipik olarak süzgeçler küçük ve dikdörtgen pencerelerdir. Bir evrimsel katmanda öğrenilen süzgeçler bir önceki katmanın çıktıları üzerinde gezdirilir. Bu nedenle evrimsel katman, bir süzgeç için öğrenilen parametrelerin paylaşımını ve girdi üzerinde kaydırılan pencere içinde uygulanmasını olanaklı kılar. Öğrenilmesi gereken süzgeçlerin az sayıda bağlantılarının olması (5x5 bir süzgeç için 25) parametre uzayını önemli ölçüde daraltır ve sinir ağını derin öğrenmeye uygun hale getirir.

Evrişimsel sinir ağlarında önde gelen araştırmacılardan biri Fukushima'dır. Fukushima, çalışmalarında (1980, 2003, 2013a) gözetimsiz öğrenme üzerinde durmuştur. Hinton vd. (2006), derin ağları eğitmeyi mümkün kılan, ağırlıkların gözetimsiz ön eğitime dayanan bir yöntem önermiştir. Ancak literatürdeki EYSA uygulamalarında baskın olarak geri yayılım algoritması kullanılmıştır (LeCun vd. 1989; LeCun vd. 1998; Baldi ve Chauvin 1993). Kıstas (benchmark) veri kümesi MNIST (LeCun vd. 1990; Wan vd. 2013; Lee vd. 2016) ve CIFAR-10 (Krizhevsky ve Hinton 2009) ile ImageNet (Le vd. 2012; Krizhevsky vd. 2012; Zeiler ve Fergus 2013) üzerinde elde edilen en yüksek başarı oranları da yine EYSA üzerinde geri yayılım ile elde edilmiştir.

Krizhevsky vd. (2017), ağa derin katmanlar eklenirken kaybolan gradyan probleminden kaçınan ReLU adlı yeni bir aktivasyon fonksiyonu önermiştir. ReLU'nun performansı, girdi negatif olduğunda kaybolan gradyan sorunu devam ettiğinden, ağırlıkların doğru şekilde ilklendirilmesine sıkı sıkıya bağlıdır. ReLU'nun kaybolan gradyan problemini çözmek için, negatif değerlerde küçük gradyanlara sahip farklı ReLU varyantları (He vd. 2015; Maas vd. 2013) önerilmiştir. Geri yayımlı öğrenmenin başarısı, sadece kaybolan gradyanı önlemek için doğru aktivasyon fonksiyonunun kullanılmasına değil, aynı zamanda ağırlıkların iyi bir şekilde ilklendirilmesine de bağlıdır. Basit Gauss dağılımın ağırlıkların ilklendirilmesi için iyi bir seçim olmadığı gösterilmiştir (Glorot ve Bengio 2010). İstikrarsız gradyanlardan kaçınmak için farklı ağırlık ilklendirme teknikleri (He vd. 2015; Glorot ve Bengio 2010; Mishkin ve Matas 2016; Sussillo 2014; Krähenbühl vd. 2016) önerilmiştir.

İstikrarsız gradyanları önlemek için farklı mimari yapılar da önerilmiştir. Ioffe vd. (2015), geri yayılım ile yapılan ağırlık güncellemelerinin, aktivasyon fonksiyonlarının çıktılarının doymuş (saturated) bölgelere taşınmasıyla sonuçlanan iç değişinti kaymasına (internal covariate shift) yol açtığını savundular. Bunu önlemek için, aktivasyon fonksiyonuna girişler için toplu normalizasyon yapılmasını önerdiler. Toplu normalizasyon katmanı, Santurkar vd. (2018) tarafından gösterildiği gibi gradyan yönünü yumuşatır ve doymuş bölgelerden kaçınarak ağın eğitilmesini mümkün kılar. He vd. (2016) tarafından önerilen bir diğer mimari değişiklik, atlamalı bağlantılardır (skip connections). Ağ derinleştikçe ilerleyen sinyal zayıflar ve eğitim hatası artar. Sinyalin zayıflamasını önlemek için, daha derin katmanlarda çıkış sinyalini güçlendiren atlamalı bağlantılar önerilmiştir. Veit vd. tarafından (2016) belirtildiği gibi, atlama bağlantıları, eğimin içinden akabileceği sıç ağlar oluşturarak kaybolan gradyanları önler. Atlama bağlantısında küçük bir farklılığa sahip benzer bir mimari Srivastava vd. (2015a,b) tarafından da önerilmiştir.

Derin ağlar üzerine yapılan arařtırmalara gözetimli öğrenme hakim olmasına rağmen bu alanda gözetimsiz yöntemler de önerilmiştir. Gözetimsiz eğitim üzerine yapılan arařtırmaların çoęu, Kısıtlanmış Boltzmann Makineleri (Restricted Boltzmann Machines) ve Otomatik Kodlayıcılara (Autoencoders) dayanmaktadır (Vincent vd. 2008; Le vd. 2012; Bengio 2014; Carreira-Perpiñán ve Wang 2014; Lee vd. 2015). Evriřimsel otomatik kodlayıcılar (Masci vd. 2011; Hou ve Yan 2018) EYSA'larda aęırlık ilklendirme olarak kullanılacak öznitelik temsillerini öğrenmek için önerilmiştir. EYSA'lara odaklanan dięer alıřmalar hibrid bir öğrenme yöntemi izlemiřlerdir. Bu alıřmalar önceden gözetimli řekilde eğitilmiş ağları kullanır (Liang vd. 2015; Liu ve Mukhopadhyay 2018; Caron vd. 2018) veya temsili sınıflar üzerinde kendi kendine gözetimli öğrenmeye dayanır (Dosovitskiy vd. 2014).

Donanımdaki teknolojik ilerleme ve EYSA mimarisıyla, geri yayılım algoritması geniş veri kümelerine başarıyla uygulanır hale gelmiştir. Ancak, geri yayılım yaklaşımı kullanıldığında başarı rastgele ilklendirme aşamasına fazlasıyla baęlıdır. İstikrarsız gradyanlar nedeni ile sınıflandırma başarısı düşebilir. Dięer taraftan geri yayılım ile eğitim, ıktı katmanından girdi katmanına doğru, her katmandaki her sinire atfedilen hatanın önceki tüm katmanlardaki tüm sinirlere geri yayılımını gerektirdięi için yüksek karmařıklığa sahiptir. Gradyan temelli yöntemlerde arama küçük adımlar halinde yapılabildięi için geri yayılım, eğitim kümesi üzerinde çok sayıda tekrar yapılmasını gerekli kılar.

Rekabetçi öğrenme (competitive learning) (Rumelhart ve Zipser 1985) gözetimsiz bir öğrenme yöntemidir ve yapay sinir ağları için öznitelikler keřfetmek için kullanılmıştır (Maclin ve Shavlik 1995). Rekabetçi öğrenme algoritmasının en önemli özellięi her katmanda tek bir kazanan sinir seilmesi ve bu sinirin baęlantılarının girdiye benzeyecek řekilde güncellenmesidir. Bu řekilde bir katmanda bulunan sinirler, veri kümesindeki tekrarlayan belli karmařıklıktaki örüntüleri ifade eden birer öznitelik algılayıcılarına dönüşürler. Rekabetçi öğrenmede başarı, geri yayılımda olduęu gibi rastgele ilklendirme aşamasına baęlıdır. Dięer taraftan rekabetçi öğrenme, her girdi için bir katmanda tek bir sinirin eğitilmesi sayesinde geri yayılımın kredi (veya hata) atama problemine açık deęildir.

Fukushima, rekabetçi öğrenmeye dayalı Winner-Take-All (2003) ve Add-if-Silent (2013b) yöntemlerini önermiştir. Add-If-Silent yöntemini ETL-1 veri kümesi üzerinde denemiř ve %0,8'in altında hata oranı elde etmiştir. Schmidhuber (2012) ise Winner-Take-All yöntemini MNIST veri kümesine uygulayarak %0,23 ile oldukça düşük bir hata oranına ulaşmıştır.

Bu proje ile önerdiğimiz yöntem, Winner-Take-All (WTA) ve Add-if-Silent (AiS) kurallarından esinlenerek evrişimsel katmanların tamamen gözetimsiz eğitimidir. Yöntemimizde evrişimsel bir katmanın eğitimi, keşfedilen öznitelikler ile aday arasındaki benzerlik ölçüsüne dayalı olarak yeni özniteliklerin keşfedilmesine bağlıdır. Bir girdi bir özniteliğe ne kadar benzerse, benzerlik puanı o kadar yüksek olur. Bu, Siyam Ağlarının (Siamese Networks) (Wang ve Gupta 2015) görüntü üçlülere arasındaki benzerlik işlevini öğrenmesine benzer gelebilir. Siyam ağlarından farklı olarak, bir aday örüntünün keşfedilmiş özniteliklere benzerliğini basit bir iç çarpım (dot product) ile hesaplayan bir EYSA modeliyle çalışıyoruz. Bu benzerlik puanına bağlı olarak, yöntemimiz bu adayı yeni bir öznitelik veya mevcut bir özniteliğin destekçisi olarak ilan eder. Adayın benzerlik puanı, önceden belirlenmiş bir benzerlik eşiğinden düşükse, yeni bir öznitelik keşfedilmiş olur. Aksi takdirde en benzer (yani kazanan) özniteliğin ağırlıkları gözetimsiz bir şekilde güncellenir. Bu keşif yöntemi, EYSA mimarisinde hem katmanlardaki sinir sayılarının hem de bu sinirlerin ağırlıklarının ilklendirilmesini ortadan kaldırarak hiperparametre sayısını önemli oranda azaltmamıza izin verir.

Bu projede önerdiğimiz RDÖ algoritması, EYSA mimarisine rekabetçi öğrenmeyi bir araya getirir. RDÖ'nün EYSA'ya uyarlanabilmesi için katmanlarda süzgeçlerin birer öznitelik olarak oluşturulması ve güncellenmesi araştırmalarını gerçekleştirdik. Proje sonucunda RDÖ ile derin ağlarda gözetimsiz öğrenmenin gürbüz, etkili ve verimli bir yöntemi önerilmiştir. Gözetimsiz eğitilen son katmanın önüne konulan gözetimli katman(lar) ile sınıflandırma yapılmış ve yöntem bir bütün olarak denenmiştir. Bu denemelerde literatürde kıstas kabul edilen MNIST, CIFAR-10 ve Tiny ImageNet veri kümeleri kullanılmıştır.

3. Gereç ve Yöntem

3.1. Rekabetçi Derin Öğrenme ile Gözetimsiz Öğrenme ve Sınıflandırma

Geleneksel EYSA yöntemlerinin aksine projede sunduğumuz yaklaşım, katmanlarda süzgeç sayısı ve süzgeçlerin ilk ağırlıkları hiperparametrelerini devre dışı bırakır. Süzgeçler, eğitim setinden tek bir epokta otomatik olarak oluşturularak ve evrilerek keşfedilir. Yaklaşımımız, süzgeçlerin gözetimsiz ve kendi kendini organize eden bir yöntem ile başlatılmasını ve eğitilmesini sağlar. Bu yöntem, Literatür Özeti'nde ifade edilen sinir ilklendirme tekniklerini göz ardı etmemize izin verdiği gibi sinirlerin rastgele ilklendirmeye dayalı olarak eğitim süresince olası kararsız davranışını ortadan kaldırır. Çalışmalarımız, Rekabetçi Derin Öğrenme (RDÖ)'nün genel EYSA yöntemlerine göre çok daha az, hatta çoğunlukla tek bir epokla eğitilebildiğini göstermiştir.

3.1.1. Neocognitron

Fukushima (1980), Hubel ve Weisel (1959)'in bulgularına dayanan ve son EYSA uygulamalarının aksine gradyan kullanılmadan eğitilebilen Neocognitron yaklaşımını önerdi. Neocognitron yaklaşımı, S ve C hücreler adı verilen, basit ve karmaşık katmanlardan oluşan bir EYSA mimarisidir. S ve C hücre düzlemleri, evrişim ve havuz katmanlarına benzer, Neocognitron yapısında hücre düzlemleri oluşturur. Eğitimi tamamlanmış bir C düzlemi yalnızca tek bir karmaşık özniteliğe tepki verir.

İlk Neocognitron mimarileri WTA yaklaşımı etrafında şekillendi. WTA yaklaşımına göre, sisteme bir girdi deseni tanıtıldığında, basit hücreler birbirleriyle rekabet ederek kazanan olmaya çalışır. Girdiyi en yüksek yanıtı veren basit hücre, ait olduğu hücre düzleminin temsilcisi (kazanan) olur. Basit hücre düzlemleri, WTA algoritması tarafından sağlanan kendi kendine organizasyon nedeniyle belirli özelliklere seçici olarak duyarlı hale gelir.

Fukushima, daha sonraki yıllarda (2013b) Neocognitron için AiS olarak adlandırılan başka bir eğitim kuralı önerdi. AiS kuralı, bir katmandaki tüm basit hücreler sessizse (önceden tanımlanmış bir oranda uyarılmamışsa), yeni bir hücrenin üretilmesini ve katmana eklenmesini önerir. Yeni hücreyi oluşturan uyarıcı vektörün değerleri, yeni hücrenin ağırlıkları olarak kullanılır. Bu hücrenin ağırlıkları artık değişmez. AiS kuralı altında, girdi uzayı öznitelik vektörleri tarafından tamamen kapsanana kadar hücrelerin sayısı monoton olarak artar.

3.1.2. Evrişimsel özniteliklerin keşfi ve sinir oluşturma

Yaklaşımımızda, girdi veri kümesindeki her örnek, süzgeç keşfi için gözetimsiz bir şekilde değerlendirilir. Bu değerlendirme sürecinin iki adımı vardır. İlk adım, proje kapsamında önerdiğimiz bir yöntem ile veri kümesinden süzgeç adaylarını elde etmektir. İkinci adım, bir benzerlik eşliğine dayalı olarak katmandaki süzgeçleri oluşturur ve eğitir. İkinci adım her bir katman için yalnızca tek bir epokta tamamlanır. Epok sonunda, mevcut evrişim katmanının eğitimi tamamlanmış olur ve artık bu katmanda güncelleme yapılmaz. İlk adımın uygulanması CPU üzerinde yürütülürken, evrişim ve seyrek örnekleme (subsampling) işlemleri GPU üzerinde yürütülür. Evrişim katmanlarının önüne konulan tam bağlantılı YSA'nın eğitimi de GPU üzerinde yapılmaktadır.

3.1.3. Süzgeç adaylarının çıkarımı

Evrişimsel süzgeçlerin alıcı alan (receptive field) boyutundaki pencerelere sahip bir imge yamasındaki öznitelikleri tespit etmeye çalıştığı gerçeğinden yola çıkarak, alıcı alan boyutundaki pencereleri 1 piksel adımlarla keserek girdilerden süzgeç adaylarını çıkardık. İmge yamalarını elde ettikten sonra, alana ait ayraç (discriminant) bilgilerini saklarken süzgeç aday sayısını kabul edilebilir sınırlarda tutmak için bir eleme işlemi uyguladık. Aşağıdaki paragraflarda, girdi imgelerden aday süzgeçleri çıkarma sürecimizi detaylandıracağız.

İlk adım olarak, tüm olası imge yamaları, girdi imgelerden bir piksel adımlar ile kesilir. Ancak bazı yamalar, arka plan görüntüleri gibi çok düşük değışintiye sahip olabilir. Değerli bilgi taşımadığı düşünölen bu yamalar ıskartaya çıkarılmıştır.

İkinci eleme adımımız, imge yamasının piksel değerlerinin ağırlık merkezine dayanmaktadır. İmgeleri birer piksel atlamalarla keserken, aynı özniteliği çerçeve içinde farklı konumlarda içeren imge yamaları elde edilir. Algoritmamız, gereksiz imge yamalarını ıskartaya çıkarmak için yamaların piksel değerlerinin ağırlık merkezini hesaplar. Hesaplanan ağırlık merkezi, çerçevenin merkezinden hem x hem de y eksenlerinde ± 0.5 piksel uzaklığında değılse, yama ıskartaya çıkarılır. Kalan görüntü yamaları, süzgeç adayları kümemizi oluşturur.

İmge yamalarına uyguladığımız bu iki eleme, eğitim sırasında temsil ve ayırıcı gücü yüksek değerli örüntülere, öznitelik veya süzgeç adayı olarak odaklanmamızı sağlar. Aynı süzgeç adayı çıkarma işlemi, derin katmanlardaki öznitelik haritalarına (feature maps) da uygulanır.

3.1.4. Gözetimsiz eğitim

Girdi imgelerden (veya gizli katmanlardaki öznitelik haritalarından) süzgeç adaylarını elde ettikten sonraki adım, bu kümeden süzgeçlerin keşfidir. Eğitim algoritmamız Algoritma 1'de sunulmuştur.

1: R: süzgeç_boyutu x süzgeç_boyutu boyutlarında imge yamaları, W: Ağırlık matrixi, V: Destekleyici sayıları, S: Skorlar, L: Katmanlar, I: İmgeler, eşik_değer: Benzetim eşiği

```
2: function KatmanEğitim(L, I)
3:   ilk_örnek ← True
4:   if L ≠ 0 then
5:     I ← GPUEvrişimiVeÖrneklemeUygula(L, I)
6:     for each imge Im, m = 1, ..., t do
7:       R ← ImgelerdenYamalara(Im, süzgeç_boyutu)
8:       R matrisinin satırlarını karıştır
9:       if ilk_örnek then
10:        WL ← R0 + G(μ, σ)
11:        VL ← 1
12:        ilk_örnek ← False
13:        continue
14:       for each yama Ri, i = 1, ..., n do
15:        S ← WL.RiT
16:        j ← argmax(S)
17:        if Sj > eşik_değer then
18:          Δ ← (Ri - WjL) / (VjL + 1)
19:          WjL ← WjL + Δ
20:          VjL ← VjL + 1
21:        else
22:          f ← Ri + G(μ, σ)
23:          f WL sonuna eklenir
24:          1 VL sonuna eklenir
25:   return W
```

Algoritma 1. RDÖ ile gözetimsiz eğitim

Önerdiğimiz eğitim yönteminde, evrişim katmanları her zaman boş olarak başlar ve süzgeçler dinamik olarak oluşturulur. Gözetimsiz öğrenme algoritmamızda katmanları ayrı ayrı eğitiriz. İlk evrişim katmanı L giriş katmanı olduğundan, Satır 5'te görüldüğü gibi R adaylarını elde etmek için ham eğitim görüntüleri I kullanılır. Satır 15'te görüldüğü gibi eğitim algoritması, yeni süzgeçler keşfetmek için süzgeç adayı ve katmanın mevcut süzgeçleri arasındaki benzerliği, iç çarpım kullanarak değerlendirir. Satır 9'da görüldüğü gibi L başlangıçta boş olduğundan, ilk süzgeç adayı keşfedilen ilk süzgeç olur. Satır 10'da görüldüğü ve Denklem 1'de sunulduğu gibi, ilk ağırlıkları elde etmek için ilk adayın piksel değerlerine bir Gauss gürültüsü eklenir. Gauss gürültüsünün eklenmesi, bir özneteliğin süzgeç olarak doğrudan eşlenmesini önlemek içindir. Her süzgecin bir V_j destekçi sayısı vardır. Destekçi sayısı, ağırlık güncellemeleri için çok önemli bir veridir; ilişkili süzgecin kaç kez başka bir süzgeç adayının temsilcisi haline geldiğini gösterir. Satır 11 ve 24'te görüldüğü gibi bir süzgeç adayı ilgili katmanın bir süzgeci haline geldiğinde destekçi sayısı 1 ile ilklendirilir.

$$\mathbf{w} \leftarrow \mathbf{R}_i + G(\mu, \sigma)$$

Denklem 1

İlk süzgeç keşfedildikten sonra, algoritma, 17. satırda görüldüğü gibi kalan adayları yeni bir süzgeç olmak için yeterince özgün olmadıklarına karar vermek için daha önce keşfedilmiş süzgeçler ile karşılaştırmaya başlar. Bunun için, satır 15'te görüldüğü gibi mevcut \mathbf{R}_i adayı ile mevcut \mathbf{W}^L katmanında önceden keşfedilen tüm süzgeçler arasında bir benzerlik puanı hesaplanır.

Keşfedilen süzgeçler, kalan adayları temsil etmek ister. Bir adayın temsilcisi olabilmek için, süzgecin adaya belli bir eşiğin üzerinde ve en çok benzemesi gerekir. \mathbf{S}_j benzerlik puanı en yüksek olan süzgeç, \mathbf{R}_i adayı için kazanan olur. Ancak en benzer süzgeç olmak adayı temsil etmek için yeterli değildir. Temsilci olmak için \mathbf{S}_j benzerlik puanının da bir benzerlik eşiğinin üzerinde olması gerekir. \mathbf{S}_j benzerlik puanı eşikten büyükse, bu, \mathbf{R}_i adayının geçmişte gözlemlenen bir örüntüyü taşıdığı anlamına gelir. Dolayısıyla bu aday yeni bir süzgeç değil, mevcut bir süzgecin destekçisi olur. Algoritma, kazanan süzgecin ağırlıklarını Denklem 2'ye göre günceller ve ilgili süzgecin destekçi sayısı bir artırılır. Bu işlemler Algoritma 1'de 17-20. satırlar arasında gösterilmiştir.

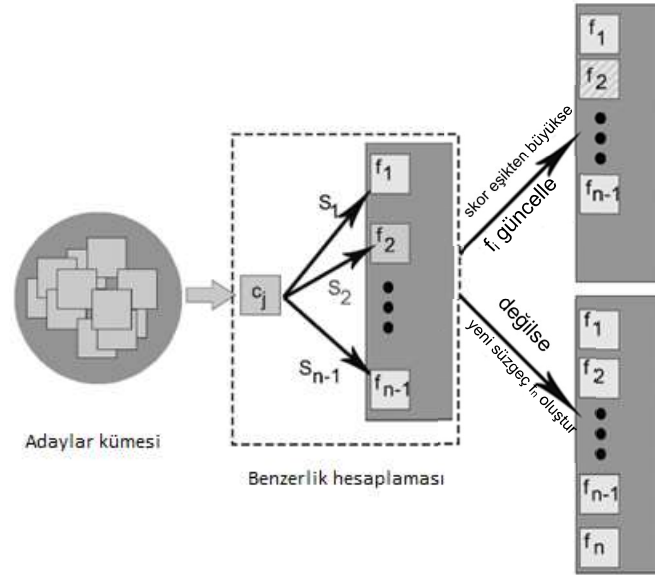
$$\mathbf{W}_i \leftarrow \mathbf{W}_i + \frac{\text{aday}_j - \mathbf{W}_i}{\text{destekçi}_{i+1}}$$

Denklem 2

\mathbf{S}_j benzerlik puanı, benzerlik eşik değerinden küçükse, algoritma \mathbf{R}_i adayının özgün bir örüntü taşıdığına karar verir. Algoritma 1'de Satır 21-24'te gösterildiği gibi bu aday, \mathbf{L} katmanı için yeni bir süzgeç olacaktır. Yeni süzgeç ağırlıkları Denklem 1'e göre ayarlanır ve destekçi sayısı 1 olur. Bu süreç Şekil 1'de temsil edilmiştir.

Gözetimsiz eğitim, \mathbf{R} içindeki tüm adaylar işlenene kadar tekrarlanır. Bir epok tamamlandığında, \mathbf{L} evrişimsel katmanının eğitimi sona erer ve sinir ağırlıkları sabitlenir.

İlk evrişim katmanı eğitildikten sonra, girdiyi derin katmanlara taşımak için evrişim ve seyrek örnekleme yapılır. Her derin katman için 3 boyutlu girdi imgeler, önceki evrişim ve seyrek örnekleme katmanlarının çıktılarıdır. Proje kapsamında geliştirdiğimiz algoritma, GPU yardımıyla girdi görüntülerine evrişim ve seyrek örnekleme işlemlerini çok hızlı şekilde gerçekleştirir. Aday süzgeç kümesi, Algoritma 2'nin çıktısı kullanılarak oluşturulur.



Şekil 1. Önerilen RDÖ gözetimsiz eğitim algoritması.

1: B: Evrişim uygulanacak sabit sayıda imge, W: Süzgeç ağırlıkları matrisi, L: katman

```

2: function GPUEvrişimiVeÖrneklemeUygula(L, B)
3:   Artışık model (sequential model) oluştur
4:   for each katman  $L_i$  do
5:     Katman  $L_i$ 'ye Conv2D uygula
6:     W dosyadan oku
7:      $W \leftarrow W/\max(W)$ 
8:      $W^+ \leftarrow W^+ / \sum W^+$ 
9:      $W^- \leftarrow -1/\text{negatif\_değerler\_sayısı}$ 
10:     $W^L \leftarrow W$ 
11:     $W^L.trainable \leftarrow \text{False}$ 
12:    if  $i \neq L-1$  then
13:      MaxPool2D katmanı ekle
14:    Modeli derle
15:    for each batch  $B_i$  do
16:      Feature_Maps $_i \leftarrow$  Katman çıktısı
17:      Feature_Maps $_i$  h5py dosyasına yaz

```

Algoritma 2. Evrişim ve seyrek örnekleme adımları

Mevcut evrişim katmanı için süzgeç bulma işlemi tamamlandıktan sonra, süzgeç sayısı ve bu süzgeçlerin ağırlıkları ile bir evrişim katmanı oluşturulmuş olur. Algoritmamız süzgeçleri keşfedip eğittiğinden, bu aşamada ağırlıklar güncellenmez. Algoritma 2'de satır 7 – 11'de gösterildiği gibi keşfedilen süzgeçlere bir normalleştirme ve ölçeklendirme uygulanır. Bu aşamadan sonra, önceden

tanımlanmış ϵ değerinden (çoğunlukla 0) daha küçük olan piksel değerleri de ölçeklenir. Burada amaç, süzgecin loş bölgesinin görüntünün parlak bir bölgesine uygulandığı durumlarda, o bölgeden gelen yanıtı sıfırlamak yerine ceza olarak değerlendirmektir.

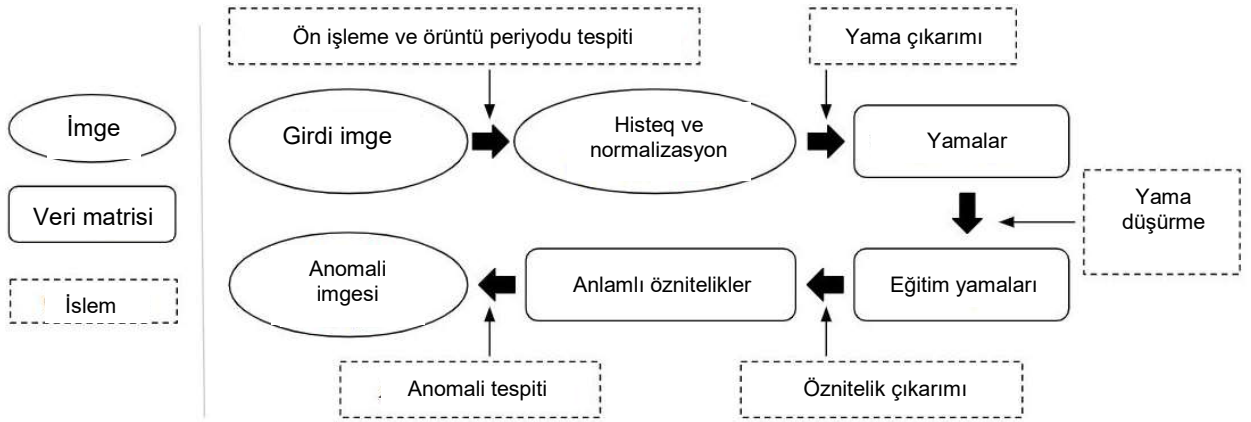
Normalleştirilmiş ve ölçeklendirilmiş süzgeç ağırlıkları ayarlandığında, evrişim katmanının eğitimini önlemek için katmanın 'eğitilebilir' parametresi False olarak ayarlanır. Model, evrişim katmanından sonra bir seyrek örnekleme katmanı gerektiriyorsa, modele bir örnekleme katmanı eklenir.

3.2. Rekabetçi Derin Öğrenme ile Gözetimsiz Kusur Tespiti

Bölüm 3.1'de detaylandırdığımız RDÖ yaklaşımı, kusur tespitinde aşağıdaki adımlar ile kullanılır:

- İlk aşama, eğitimden önce gerçekleştirilen adımlara karşılık gelir; bunlar ön işleme ve yama çıkarmadır.
- İkinci aşama eğitime karşılık gelir. Yama çıkarma işleminden sonra, eğitim Algoritma 1'de gösterildiği gibi gözetimsiz bir şekilde yapılır.
- Test olan son aşama, kusur algılama adımına karşılık gelir.

Şekil 2, yaklaşımımızın genel şemasını göstermektedir.



Şekil 2. RDÖ ile kusur tespitinin şema halinde gösterimi.

3.2.1. Ön işleme

Kusur tespitinde kullanılan imgeler, dijital kamera, MRI veya başka kiplerde algılamayı etkileyen zayıf kontrasta sahip olabilirler. Kusur tespitinin sağlıklı yapılabilmesi için girdi görüntülerinin kontrastının ayarlanması gerekir.

Gama düzeltme, doğrusal ölçekleme, uyarlamalı histogram eşitleme (adaptive histogram equalization) ve histogram eşitleme gibi çeşitli kontrast düzeltme yaklaşımlarını test ettik. Basit histogram eşitleme, kusur tespitinde en iyi sonuçları verdi. Bu nedenle hem eğitim hem de test görüntüleri için histogram eşitleme kullanmayı tercih ettik.

Histogram eşitleme tamamlandıktan sonra, piksel değerlerini $[0, 1]$ arasında normalleştirmek için 255'e böldük ve Denklem 3 ve 4'te gösterildiği gibi veri normalizasyonu yaptık.

$$I \leftarrow \frac{(I-\mu)}{\sigma} \quad \text{Denklem 3}$$

3.2.2. Süzgeç adaylarının çıkarılması

$G \times Y$ boyutlarında bir girdi görüntüsü I verildiğinde, bu imgeden $p \times p$ boyutlarındaki yamalar kırılır. Yama boyutu pencere veya süzgeç boyutu olarak isimlendirilir ve sınır ağının algılayıcı alanına (kernel) karşılık gelir. Bu yamalar üst üste binen pencerelerin imge satır ve sütunları üzerinde gezdirilmesi ile çıkarılır. Yamalar böylece kısmi bölgelerini komşularıyla satır ve sütun yönünde paylaşır. o örtüşmeyi kontrol eden adım parametresi iken, I görüntüsündeki toplam yama sayısı aşağıdaki denklem kullanılarak hesaplanabilir.

$$N_{patches} = \left[\frac{W - p}{o} \right] \cdot \left[\frac{H - p}{o} \right] \quad \text{Denklem 4}$$

Sonuç olarak, her girdi imge için $N_{patches}$ adet $p \times p$ boyutlarında yama elde edilir. Bu yamalar vektör haline getirilerek kullanılacaktır. Eğitimde, eğitim imge(ler)inden çıkarılan yamaların her biri, Algoritma 1'de ayrıntılı olarak açıklanan süzgeç seçimi ve eğitimi için bir adaydır. Test aşamasında, test imgesindeki yamalar aynı şekilde çıkarılır ve kusurlu/kusursuz imge bölütleme için kullanılır.

Bir pikseli yerel komşuluk bilgilerini dikkate alarak sınıflandırmanın, tek piksel üzerinde karar vermekten daha etkili olduğu gösterilmiştir. Komşuluk alanının büyüklüğü süzgeç penceresinin büyüklüğüne bağlıdır. Yaklaşımımızda evrimsel katmanlardan sonra seyrek örnekleme kullandığımız için süzgeç penceresi boyutuna karşılık gelen komşuluk boyutu, derin katmanlara doğru ilerledikçe büyür. Bu da farklı ölçeklerdeki örüntüleri yakalamamıza ve böylece farklı büyüklüklerdeki kusurları tespit etmemize olanak tanır. RDÖ içindeki en derin katman, girdi imge üzerinde en geniş alanda kusur arayan katmandır.

Bazı durumlarda, çıkarılan yamaların bazıları anlamlı bilgiler içermez. Bu nedenle, eğitim ve öznitelik seçiminden önce bilgi içermeyen yamaları ıskartaya çıkarmak için deęişintiye dayalı bir eleme gerçekleştiririz. Bölüm 3.1.3'te anlatıldığı gibi bir yamadaki piksellerin deęişintisi, tanımlanmış bir eşiğin altında ise bu yama ıskartaya çıkarılır. Aksi takdirde, bir sonraki adım olan süzgeç seçimi için aday olarak tutulur.

3.2.3. Süzgeç seçimi

Kusur tespitinde tüm ağ evrimsel katmanlardan oluşmaktadır. Diğer bir deyişle derin ağın önüne tam bağlantılı bir YSA konarak gözetimli eğitim yapılmaz. Süzgeç seçimi ve evrimi ile eğitim, Algoritma 1 ve 2'de gösterildiği gibi yapılmaktadır. Aynı şekilde, tüm evrimsel katmanlar için eğitim tamamlandıktan sonra süzgeç ağırlıkları sabitlenir ve kusurların tespitinde gözetimsiz olarak kullanılır.

Bu noktada hatırlatmak gerekir ki, evrimsel katmanların eğitiminde herhangi bir aşamada sinir sayısı bir hiperparametre olarak belirlenmez; süzgeçleri temsil eden sinirler eğitim sırasında katman katman ve dinamik olarak oluşturulur. Sonuç olarak, ayarlanması gereken en önemli hiperparametre bir benzerlik eşiğidir.

3.2.4. Kusur tespiti

Başlangıçta yalnızca kusur içermeyen imgeler ile eğitim yaptık. Bu ön kabul altında, öznitelik haline gelen tüm yamalar kesinlikle hatasız örnekler içerir. Sezgisel olarak, yalnızca kusursuz örüntülerden oluşan bu derin ağdan geçirilen bir test imgesinin çıktısı, imge üzerinde kusur aramak için gösterge olarak kullanılabilir. Kusur tespitinde evrimsel ağlarda süzgeçlerin imge üzerindeki yamalar ile arasındaki Manhattan uzaklıkları kullanılmıştır. A bir yama iken B bir süzgeçken Manhattan uzaklığı,

Denklem 5'te gösterildiği gibi bulunur. Bu durumda elde edilen mesafe değeri, eğitimde kullanılan benzerlik değeri ile negatif korelasyona sahiptir. Bu değer bir farklılık puanı olarak kabul edilebilir.

$$d(A, B) = \sum_{i=1}^P \sum_{j=1}^P |a_{ij} - b_{ij}|$$

Denklem 5

Katmanlarda her bir süzgecin girdinin tamamı üzerinde gezdirilmesinden sonra elde edilen mesafe değerleri, bir kusur arama eşiği ile kıyaslanır ve imgenin ilgili alanının kusurlu olup olmadığına dair etiketleme yapılır. Burada yine, verilen bir imge yamasına tüm süzgeçler uygulandığında aralarındaki en düşük mesafe değerlerini alır ve 2 boyutlu bir imge-süzgeç mesafe matrisi oluştururuz. Bu imge üzerinde bir yamanın kusurlu olup olmadığına karar vermek için bu aşamada bir kusur eşik değeri belirlenmiştir.

Mesafe puanı kusur eşiğinden düşükse, yamanın eğitim sırasında keşfettiğimiz özneliklerden birine yeterince benzediği ve bir kusur taşımadığı çıkarımını yapıyoruz. Burada not edilmesi gerekir ki yamanın girdi imgede ne büyüklükte bir alana geldiği, kusurun tespit edildiği katmanın derecesine bağlıdır. İlk katman, sadece p x p alanında kusur ararken, sonraki katmanlar seyrek örnekleme ölçeği (stride) 2 olduğunda bir önceki katmanın 2 katı büyüklükte bir alanda kusur arıyor olurlar.

İmge üzerindeki kusur olasılıkları, merkezi piksel için en yüksek ve yama sınırlarında en düşük olacak şekilde kusur bulunan pencere içine dağıtılır. Olasılık bilgisi, merkezi piksel için en yüksek ve yama sınırlarında en düşük olacak şekilde dağıtılır. Yamadaki her pikselin ağırlığı, aşağıdaki gibi iki boyutlu simetrik Gauss çekirdeği (kernel) tarafından belirlenir:

$$G(i, j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right)$$

Denklem 6

Yamaların her katmanda girdi imgesi üzerinde birbirleri ile örtüşüyor olması ve derin katmanlardaki süzgeçlerin daha geniş bölgeleri inceleyen algılayıcılar olması nedeni ile her bir piksel birçok aşamada farklı komşuluklarla kusur olasılıkları hesabına dahil edilir. Bu olasılıklar, girdi imge ile aynı sayıda satır ve sütuna sahip (padding yapılarak) 2 boyutlu bir kusur olasılığı matrisi oluşturulacak şekilde toplanır. Bu işlemlerin sonunda, her pikselin değerinin o pikselin kusurluluk ihtimalini gösterdiği nihai bir kusur olasılığı görüntüsü elde ederiz.

3.2.5. Kusur olasılık matrisinin bölütlenmesi

Kusur ihtimalini içeren bir görüntü elde ettikten sonra, kusura dayalı imge bölütlemeyi gerçekleştirmek için bu görüntünün ikili hale getirilmesi gerekir. Böyle bir ikileştirme için en iyi eşik değerini bulmak çok önemlidir.

Bu adım için maksimum iki boyutlu entropi (Abutaleb 1989) yöntemini seçtik. Bu yöntem, global eşikleme yöntemlerinin aksine komşuluk bilgilerini dikkate alır. Yerel bağıl entropi, bir pikselin parlaklığının, komşusunun ortalama parlaklığından farkını ölçer:

- Bir pikselin parlaklığı komşularına benziyor ise yerel görelî entropi değeri küçüktür. Bu, genellikle bir piksel ve komşusunun aynı sınıfa (arka plan veya kusur sınıfına) ait olduğu durumdur.
- Öte yandan, bir pikselin parlaklığı komşularınıninkinden çok farklıysa bu pikselin yerel entropi değeri yüksektir. Bu, pikselin görüntü veya bir kenar piksel olduğu durumdur.

L farklı gri tonlama yoğunluk seviyelerine sahip bir n x n komşuluğu verildiğinde, bu komşuluğun ortalama gri tonlama yoğunluk seviyesi değeri Denklem 7'deki gibi hesaplanır.

$$\bar{I}(x, y) = \frac{1}{n^2} \sum_{i=-(n-1)/2}^{(n-1)/2} \sum_{j=-(n-1)/2}^{(n-1)/2} I(x+i, y+j)$$

Denklem 7

Yoğunluğu (s, t)'den az olan arka plan bölgesi için H_B entropisi ve yoğunluğu (s, t) değerinden yüksek olan kusur için entropi H_A Denklem 8 ve 9'da gösterildiği gibi belirlenir.

$$H_B(s, t) = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} \left(\frac{p_{ij}}{P_1} \right) \log \left(\frac{p_{ij}}{P_1} \right)$$

Denklem 8

$$P_1 = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij}$$

$$H_A(s, t) = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} \left(\frac{p_{ij}}{P_2} \right) \log \left(\frac{p_{ij}}{P_2} \right)$$

Denklem 9

$$P_2 = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} p_{ij}$$

Bu durumda, imgenin toplam entropisi H_I Denklem 10'daki gibi olur:

$$H_I = H_B + H_A \quad \text{Denklem 10}$$

İdeal eşik vektörü (s^*, t^*) , Denklem 11'i sağlamalıdır:

$$H(s^*, t^*) = \max \{H_I(s, t)\} \quad \text{Denklem 11}$$

Son olarak, kusur sonucunun kenarları yumuşatılırken, varsa görüntüdeki gürültüyü gidermek için görüntüye açma (opening) uygulanır.

4. Bulgular

4.1. Rekabetçi Derin Öğrenme ile Sınıflandırma

Projede önerdiğimiz yöntem, öznitelik çıkarma ve sınıflandırma performanslarını değerlendirmek için farklı RDÖ modellerine uygulanmıştır. Deneyler, MNIST, CIFAR-10 ve Tiny ImageNet veri kümeleri üzerinde gerçekleştirilmiştir.

4.1.1. Deneysel kurgu

Deneylerimizdeki tüm modeller, Theano üzerinde geliştirilmiş Keras kütüphanesini kullanmaktadır. Evrişim katmanlarının yapılandırılmasının yanı sıra Keras, GPU üzerinde evrişim ve seyrek örnekleme işlemlerini çok hızlı şekilde gerçekleştirmemizi sağlar. Tüm modellerimizde tam bağlantılı katmanlar yine Keras tarafından eğitildi. Aktivasyon fonksiyonu olarak çıktı katmanı dışındaki tüm katmanlarda ReLU kullandık. Eğitimde kayıp fonksiyonu (loss function) için kategorik çapraz entropi (categorical cross-entropy) ve tam bağlantılı katmanlardaki ağırlık güncellemeleri için Adadelta (Zeiler 2012) yöntemi seçildi. Öznitelik çıkarımı sağlayan evrişimsel katmanlar 1, geri yayımlı tam bağlantılı katmanlar ise 50 epokta eğitildi.

4.1.2. MNIST

MNIST (LeCun vd. 1990; Wan vd. 2013; Lee vd. 2016), elle yazılmış bir rakam veri kümesidir. 60.000 eğitim ve 10.000 test görüntüsünden oluşmaktadır. Deneylerimiz için veri kümesi 50.000 adet eğitim, 10.000 adet doğrulama ve 10.000 adet test imgesine ayrıştırıldı. Doğrulama kümesi, orijinal eğitim setinden rastgele seçildi. Hem eğitim hem de doğrulama kümelerinde sınıfların dağılımında denge gözetmedik.

Deneylerimiz, Tablo 1'de görüldüğü gibi farklı topolojilere sahip modeller üzerinde gerçekleştirilmiştir. RDÖ ile katmanlarda sinir sayısı ve sinir ağırlıkları belirleme aşamalarını kaldırdığımız için, deneylerimizi farklı benzerlik eşikleri kullanarak gerçekleştirdik. 0,5'in altındaki benzerlik eşik değeri, eğitim veri kümesinden çıkarılan süzgeç sayısı büyük ölçüde azaldığından iyi performans göstermedi. RDÖ'nün performansını bağımsız değerlendirebilmek için tüm ön işleme ve veri artırma (augmentation) işlemlerinden kaçındık.

Tablo 1. MNIST deneylerimizde kullandığımız RDÖ modelleri. Evrişim katmanlarında süzgeç boyutları, MaxPool boyutları ve adım (stride) gösterilmiştir.

Model	Katman 1	Katman 2	Katman 3	Katman 4	Katman 5	Sınıflandırıcı			
A	Evrişim 5 x 5	MaxPool 2 x 2, 2	Evrişim 5 x 5	-	-	Tam Bağlantılı (1000)	Dropout (%50)	Tam Bağlantılı (500)	Çıktı (10)
B	Evrişim 5 x 5	Evrişim 5 x 5	MaxPool 2 x 2, 2	-	-				
C	Evrişim 5 x 5	MaxPool 2 x 2, 2	Evrişim 5 x 5	MaxPool 2 x 2, 2	Evrişim 5 x 5				
D	Evrişim 5 x 5	Evrişim 5 x 5	MaxPool 2 x 2, 2	Evrişim 5 x 5	Evrişim 5 x 5				

Tablo 1'de gösterilen her modele farklı benzerlik eşikleri bileşimleri uyguladık. Deneyimizde 5 x 5 süzgecin, daha küçük 3 x 3 süzgece kıyasla daha yüksek doğruluk değerleri sağladığını gözlemledik. Model B, Tablo 2'de gösterildiği gibi MNIST veri kümesi üzerinde %99.03 doğrulukla en iyi sınıflandırma performansını elde etti. Modelimiz 10.000 test imgesinde sadece 97 yanlış tahmin yapmıştır. Modelin tahminlerinin karıştırma (confusion) matrisi Tablo 3'te gösterilmiştir. En iyi tahmin edilen basamak sınıfının 1, en yanıltıcı sınıfın ise 9 olduğunu gözlemledik. Her bir modelimizin eğitim ve testi 30 dakikanın altında tamamlandı.

Tablo 2. MNIST veri kümesi üzerinde her model için katmanlarda kullanılan benzerlik eşikleri (K1-K5), çıkarılan 5x5 süzgeç sayısı (SS1-SS5) ve sınıflandırma doğrulukları (Per).

Model	K1	K2	K3	K4	K5	SS1	SS2	SS3	SS4	SS5	Per (%)
A	0,5	-	0,7	-	-	50	-	70	-	-	98,85
B	0,5	0,7	-	-	-	50	107	-	-	-	99,03
C	0,5	-	0,6	-	0,7	50	-	35	-	450	98,28
D	0,5	0,5	-	0,5	0,6	50	24	-	88	62	97,01

Tablo 3. MNIST veri kümesi üzerinde B Modeli için karıştırma matrisi.

	0	1	2	3	4	5	6	7	8	9	Per (%)
0	975	0	0	0	0	0	3	1	1	0	99,49
1	0	1132	2	0	0	0	1	0	0	0	99,74
2	2	0	1027	0	1	0	0	2	0	0	99,52
3	0	0	0	1004	0	1	0	2	2	1	99,41
4	0	0	0	0	974	0	5	0	0	3	99,19
5	2	0	0	8	0	877	4	1	0	0	98,32
6	1	2	0	0	1	2	952	0	0	0	99,37
7	0	3	6	1	0	0	0	1015	1	2	98,74
8	1	0	3	4	0	3	1	2	957	3	98,25
9	0	2	0	3	7	1	1	5	0	990	98,12

4.1.3. CIFAR-10

CIFAR-10 (Krizhevsky ve Hinton 2009) veri kümesi, literatürde önerilen derin öğrenme yöntemlerini karşılaştırma amacı ile yaygın kullanılan veri kümelerinden biridir. Bu veri kümesi, 10 farklı sınıfta 60.000 adet 32x32 renkli imge içerir. Kümedeki imgeler, uçak, araba, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyonlardan oluşan 10 farklı sınıfı temsil eder. Veri kümesindeki sınıf dağılımı dengelidir; her sınıftan 6.000 imge bulunmaktadır.

Bu veri kümesi üzerinde yaptığımız deneylerde, veri kümesini 40.000 adet eğitim, 10.000 adet doğrulama ve 10.000 adet test imgesine böldük. Doğrulama kümesini, orijinal eğitim setinden rastgele seçtik. Hem eğitim hem de doğrulama kümelerinde sınıfların dağılımında denge gözetmedik.

Deneylemimizi, MNIST deneylemimizde belirlediğimiz modeller üzerinde gerçekleştirdik. Bu modeller Tablo 1’de verilmiştir. MNIST’ten farklı olarak, veri kümesi renkli imgelerden oluşmaktadır. Renkli imgeler, ilk katmanda 3 boyutlu ($p \times p \times 3$) süzgeçler oluşturur. Derin katmanlar halihazırda 3 boyutlu girdiler ($p \times p \times$ süzgeç sayısı) üzerinde süzgeç keşfettikleri için bu durum çalışmalarımızda zorluk oluşturmadı.

CIFAR-10 üzerinde yaptığımız deneylerde, MNIST deneylemimizden farklı olarak, daha küçük 3×3 boyutlarındaki süzgeçlerin, 5×5 süzgeçlere kıyasla daha yüksek doğruluk değerleri sağladığını gözlemledik. MNIST’te olduğu gibi Model B, CIFAR-10 veri kümesi üzerinde %62,93 doğrulukla en iyi sınıflandırma performansını elde etti (Tablo 4). Modelin tahminlerinin karıştırma matrisi Tablo 5’te gösterilmiştir. En iyi tahmin edilen sınıfın Gemi, en yanıltıcı sınıfın ise Köpek olduğunu gözlemledik. Modellerimizin eğitim ve testi 60 dakikanın altında tamamlandı.

Tablo 4. CIFAR-10 veri kümesi üzerinde her model için katmanlarda kullanılan benzerlik eşikleri (K1-K5), çıkarılan 3×3 süzgeç sayısı (SS1-SS5) ve sınıflandırma doğrulukları (Per).

Model	K1	K2	K3	K4	K5	SS1	SS2	SS3	SS4	SS5	Per (%)
A	0,3	-	0,4	-	-	54	-	14	-	-	59,89
B	0,3	0,4	-	-	-	54	34	-	-	-	62,93
C	0,3	-	0,4	-	0,5	54	-	34	-	44	59,50
D	0,2	0,5	-	0,5	0,5	30	157	-	39	59	52,21

Tablo 5. CIFAR-10 veri kümesi üzerinde B Modeli için karıştırma matrisi.

	Uçak	Araba	Kuş	Kedi	Geyik	Köpek	Kurbağa	At	Gemi	Kamyon	Per (%)
Uçak	697	11	71	30	35	11	21	12	800	32	69,70
Araba	58	679	18	27	14	10	23	10	53	108	67,90
Kuş	58	4	498	98	147	50	83	36	16	10	49,80
Kedi	12	8	55	546	96	120	100	37	9	17	54,60
Geyik	23	4	78	90	635	25	59	69	13	4	63,50
Köpek	13	3	64	267	77	448	60	57	5	6	44,80
Kurbağa	6	4	45	83	113	20	706	13	6	4	70,60
At	25	0	32	81	99	46	16	678	3	20	67,80
Gemi	91	22	25	21	17	14	14	4	774	18	77,40
Kamyon	76	91	17	45	10	19	36	27	47	632	63,20

4.1.4. Tiny ImageNet

ImageNet, makine öğrenmesi problemlerinde kullanılmak üzere toplanmış en büyük imge veri tabanıdır. Veri kümesindeki 14 milyondan fazla imge manuel olarak 21 binden fazla grup ve sınıfa ait olarak işaretlenmiştir. 2010 yılından itibaren düzenlenen ImageNet Large Scale Visual Recognition Challenge (ILSVRC), hem fotoğraftaki ana nesneye dayalı olarak imgelere sınıf etiketi atmayı hem de imge içindeki nesnelerin yerini belirlemeyi konu alır.

ImageNet, boyutları dolayısı ile eğitim yönteminin eniylenmesinde bütün veri kümesinin kullanımına olanak vermez. Bu nedenle çalışmalarımızı ImageNet'in hem sayı ve kategori, hem de çözünürlük olarak küçültülmüş hali olan Tiny ImageNet (Le ve Yang 2015) üzerinde gerçekleştirdik.

ILSVRC2014'ün bir alt kümesi olan Tiny ImageNet veri kümesi, 200 sınıftan gelen imgelere sahiptir. Her sınıfta 500 eğitim imgesi, 50 doğrulama imgesi ve 50 test imgesi bulunmaktadır. Toplam olarak veri kümesi 10.000 adet test imgesine sahiptir. Veri kümesi tüm araştırmacıların kullanımına açıktır. Tüm imgeler 64x64 çözünürlüğe sahiptir. Sınıf sayısının azlığı eğitimi kolaylaştırırken, çözünürlüğü

düşürülmüş imgeler zorlaştırır. Tiny ImageNet üzerinde yapılan araştırmalarda çoğunlukla veri büyütme veya artırma yöntemleri kullanılmaktadır.

Tiny ImageNet üzerinde yaptığımız çalışmalarda Tablo 1’de sunduğumuz RDÖ mimarilerinin yeterli gelmediğini gözlemledik. Bunun en açık sebebi sınıf sayısının MNIST ve CIFAR-10’e göre çok daha yüksek olması ve modellerin alanı temsil gücünün yetersiz kalmasıdır. Bu nedenle yeni ve daha derin mimariler denedik ve en başarılı mimariyi sonuçlarımıza dahil ettik. RDÖ mimarilerinin detaylarını Tablo 6’da ve bu mimarilerin başarımını Tablo 7’de sunduk.

Tablo 6. Tiny ImageNet deneylerimizde kullandığımız RDÖ modelleri. Evrişim katmanlarında süzgeç boyutları, MaxPool boyutları ve adım (stride) gösterilmiştir.

Model	K1	K2	K3	K4	K5	K6	K7	K8	Sınıflandırıcı
A	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	Evrişim 3 x 3	-	-				
B	Evrişim 3 x 3	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	-	-				
C	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	Evrişim 3 x 3				
D	Evrişim 3 x 3	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	Evrişim 3 x 3	Evrişim 3 x 3				
E	Evrişim 3 x 3	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	Evrişim 3 x 3	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	Evrişim 3 x 3	MaxPc 2 x 2, \downarrow	

Tam Bağlantılı (5000)

Dropout (%50)

Tam Bağlantılı (1000)

Çıktı (200)

4.2. Rekabetçi Derin Öğrenme ile Kusur Denetimi

Proje ile ortaya koyduğumuz RDÖ yaklaşımı, tamamen gözetimsiz oluşturulan evrişimsel katmanlar ile girdi uzayında gözlemlenen örüntüleri evrişim katmanlarının süzgeçleri olarak yakalar. Eğitim veri kümesi ne kadar büyük olursa, eğitim tamamlandığında ortaya çıkan ağ, alanı o kadar iyi tanımış olacaktır. Bu ağa verilen kusursuz yeni bir örnek, örnekteki örüntüleri temsil eden sınırları uyarır ve son katmanda örneğin topağını (cluster) temsil eden sinir yüksek bir sinyal üretir.

Tablo 7. Tiny ImageNet veri kümesi üzerinde her model için katmanlarda kullanılan benzerlik eşikleri (K1-K8), çıkarılan 5x5 süzgeç sayısı (SS1-SS8) ve sınıflandırma doğrulukları (Per).

Model	K1	K2	K3	K4	K5	K6	K7	K8	Per (%)
A	0,4	-	0,6	-	-	-	-	-	38,34
B	0,4	0,5	-	-	-	-	-	-	41,72
C	0,4	-	0,6	-	0,6	-	-	-	40,69
D	0,4	0,5	-	0,5	0,6	-	-	-	49,85
E	0,4	0,5	-	0,5	0,6	-	0,6	-	48,76

Diğer taraftan, bu ağa eğitim kümesinde gözlemlenmeyen bir örüntü verildiğinde bu durum ağda kararsızlığa ve farklı katmanlarda sinir çıktılarının düşük değerlere sahip olmasına neden olacaktır. Proje kapsamında çalışmalarımız, tüm sinirlerin düşük çıktı verdiği katmanların belirlenmesi ve girdi imgedeki ilgili bölgenin kusur olarak işaretlenmesi ile başladı. Bu çalışmaların sonuçlarını önceki ilerleme raporlarımızda sunmuştuk.

Ancak kusurlu bölge için bir katmandaki tüm sinirlerin düşük uyarıma sahip olacağı varsayımı, eğitim setinde hiç benzer kusur bulunmamasını gerektirdiği için yaklaşımımız tümüyle gözetimsiz değildi. Diğer bir deyişle, algoritmamızı her zaman önceden seçilmiş kusursuz örnekler ile eğitiyor ve kusurlu ya da kusursuz olduğunu bildiğimiz örnekler ile test ediyorduk.

Projemizin ek süresi içinde algoritmamızı tamamıyla gözetimsiz hale getirecek bir çalışma tamamladık. Bu çalışma sinirlerin destekçi sayısına dayalıdır. Bir sinirin destekçi sayısı, temsil ettiği topağa ait örnek sayısını gösterir. Destekçi sayısının çok az olması, ilgili örüntünün girdi uzayında nadiren gözlemlendiğini ifade eder. Eşik destekçi sayısını deneysel olarak *doğrulama veri kümesi üzerinde* ortaya koyduğumuz bir formüle dayandırarak bu nadir örüntüleri temsil eden sinirlerin uyarılmasını da kusur olarak belirledik ve yöntemimizi tamamen gözetimsiz hale getirdik. Bu sayede yöntemimiz, sadece projede incelediğimiz BRATS veri tabanı için değil, sinyallere dayalı tüm girdi uzayları için kullanılabilir bir yöntem haline gelmiştir.

Gözetimsiz eğitimi sağlayabilmek için yaptığımız bu çalışma haricinde eğitim kümemizi büyüterek çalışmamızı daha kapsamlı hale getirdik ve kusur eşliğimizi uyarlanabilir bir formüle dayandırdık. Bu sayede sisteme kusur tespiti için yeni bir hiperparametre tanımlamaktan kaçınmış olduk.

Aşağıdaki paragraflarda BRATS veri kümesi üzerindeki sonuçlarımızı paylaşacağız.

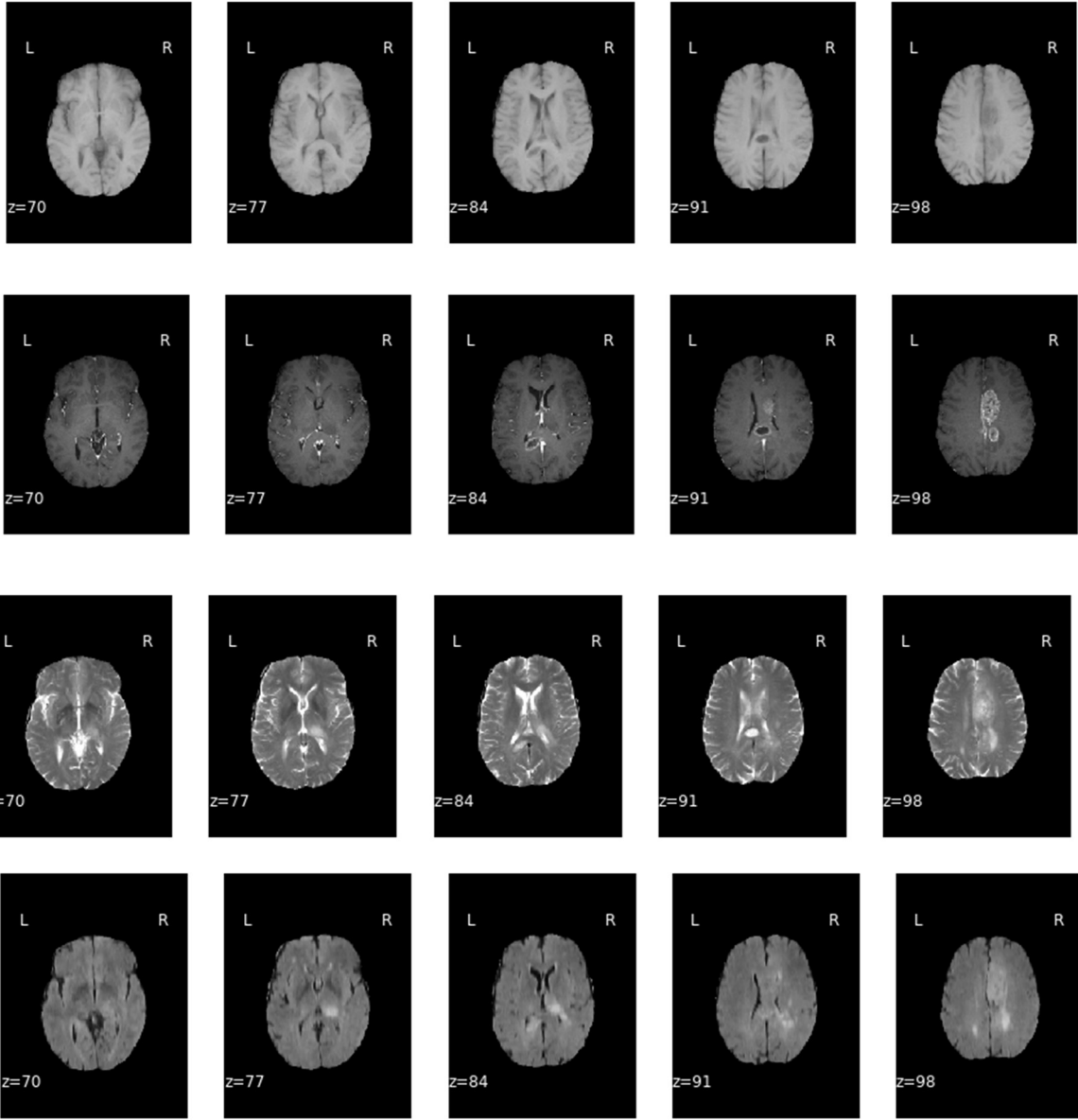
4.2.1. BRATS veri kümesi

Deneylerimizde BRATS veri kümesinin Glioblastoma (GBM / HGG) ve düşük dereceli glioma (LGG) için klinik olarak edinilmiş preoperatif çok kipli (multimodal) MRI verilerini kullandık. Örnekler, farklı kurumlardan elde edilmiştir; farklı protokoller ile farklı tarayıcılardan alındıkları için BRATS zorlu ve gerçekçi bir veri kümesidir. BRATS verisi 4 farklı kipte taranmış imgelerden oluşur. Şekil 3, farklı kiplerde elde edilen kesitleri göstermektedir.

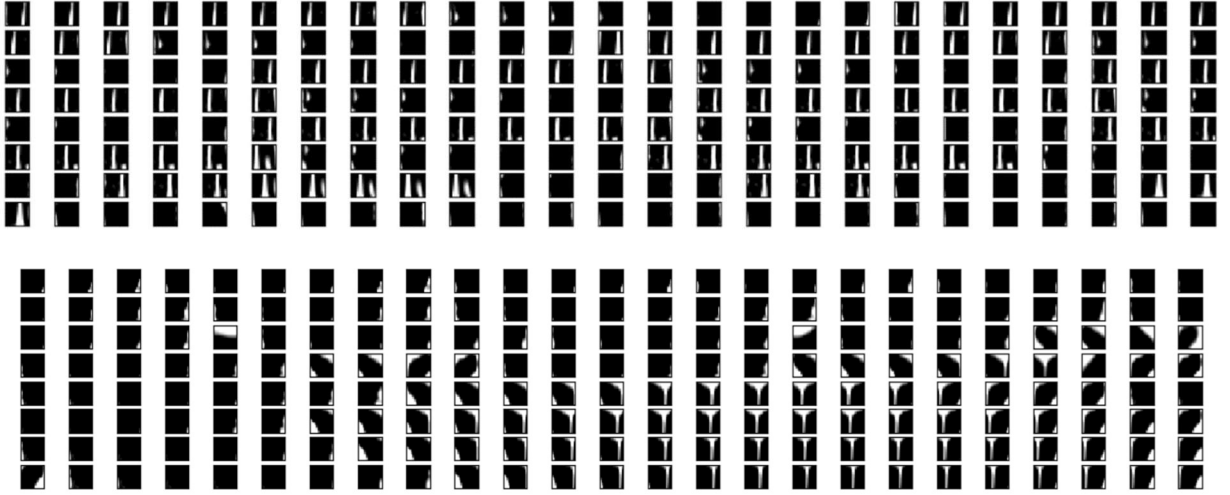
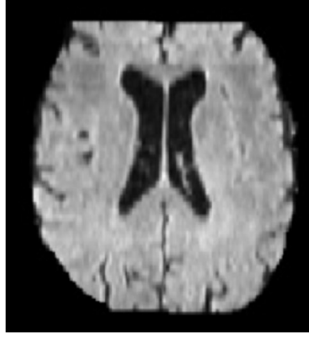
- T1: T1-ağırlıklı, gerçek imge, sagittal veya eksensel 2B veri edinimi, 1-6 mm kesit kalınlığı
- T1c: T1-ağırlıklı, netliği artırılmış (Gadolinium) imge, 3B veri edinimi ve 1 mm hacimsel boyut
- T2: T2-ağırlıklı imge, eksensel 2B veri edinimi, 2–6 mm kesit kalınlığı
- FLAIR: T2-ağırlıklı FLAIR ime, sagittal veya eksensel 2B veri edinimi, 2–6 mm kesit kalınlığı

Eğitimimizi 2019 yılında araştırmacılar ile paylaşılan BRATS veri kümesinin HGG sınıfı üzerindeki tüm örnekler üzerinde yaptık. T1, T1c, T2 ve FLAIR kiplerinden aldığımız kesitleri 3 boyutlu girdiler olarak hazırladık. Girdi 3 boyutlu olduğu için ilk katmanda elde ettiğimiz süzgeçler de 3 boyutludur (p x p x 4). İlk katmanda elde ettiğimiz süzgeçlere ait örnekler Şekil 4'te sunulmuştur.

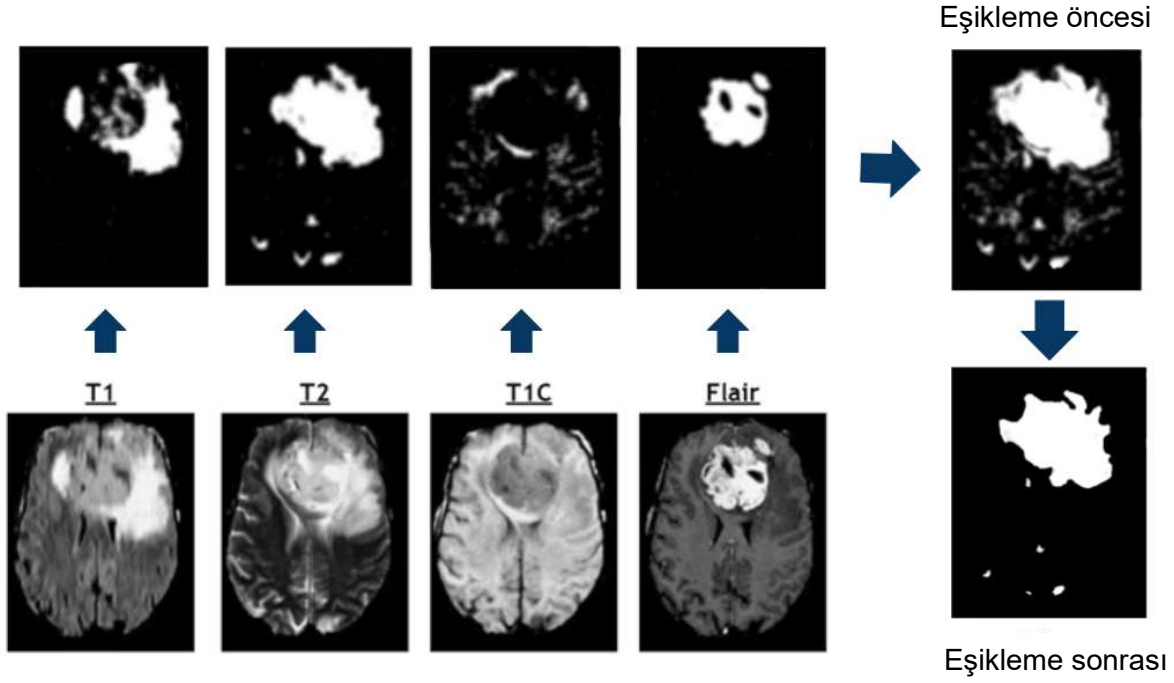
Tüm katmanlarda girdiler ile süzgeçler arasındaki Manhattan uzaklıklarını tespit ettik ve Bölüm 3.2.5'de detaylandırdığımız 2 boyutlu en yüksek entropi eşikleme ile bölütledik. Bölütleme sonucu elde ettiğimiz hastalıklı ve sağlıklı bölgelere bir örnek Şekil 5'te sunulmuştur.



Şekil 3. BRATS veri kümesinde farklı kiplerde veri örnekleri (T1, T1c, T2, FLAIR)



Şekil 4. Bir girdi örneği ve ilk katmanda elde edilen öznelikler (süzgeç ağırlıkları). Öznelikler gri seviyeye dönüştürülmüştür.



Şekil 5. RDÖ ile tümör tespiti. Dört farklı kipe ait imgeler için Manhattan uzaklıkları ve eşikleme sonrası tespit edilen tümör.

Sonuçlarımızı, bölütlemelerde sık kullanılan Dice Score (DSC) metriği ile numerik hale getirdik. Bu metrikte X gerçek referans bölütleme ve Y de tespit edilen bölütleme alanı iken DSC Denklem 12'deki gibi bulunur. DSC böylelikle sadece bulunan gerçek pozitifleri değil, hatalı pozitifleri de hesaba katarak bir performans metriği sunar.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Denklem 12

Önerdiğimiz yeni yöntemler ve eğitim kümemizi genişletmemiz, kusur tespitinde performansımızı kayda değer oranda yükseltmiştir. Yaklaşımımızın gözetimsiz öğrenme paradigmasını kullanan diğer yaklaşımlarla karşılaştırması Tablo 8'de görülebilir.

Tablo 8. RDÖ ile kusur tespitinin literatürdeki farklı yöntemler ile karşılaştırması

Yöntem	Veri Kümesi	DSC
K-Means	BRATS	0,69
GMM	BRATS	0,67
FCM	BRATS	0,75
FCM cascade	BRATS	0,77
SOM	Harvard Database	0,37
FKM	Harvard Database	0,36
SOM-FKM	Harvard Database	0,47
RDÖ	BRATS	0,68

5. İş Paketleri

Bu bölümde proje süresince kaydettiğimiz ilerleme iş paketleri bazında paylaşılacaktır. Her bir iş paketinin sonuç ve çıktıları, ilerleme raporlarında sunulduğu şekli ile başlıklar altında sunulmuştur.

5.1. Öznitelik Öğrenme İş Paketi

Öznitelik oluşturma: Yaklaşımımızda eğitim örnekleri evrimsel katmanlardan geçirilerek ilerletilir. Eğitim örneği için bir katmanda oluşan sinir çıktıları belli bir eşik altında kalıyorsa ilgili katmanda bu örneği temsil edecek yeni bir sinir ilklendirilir. Proje önerisinde bu eşik ne olacağına topaklanma başarımına göre karar verileceği ifade edilmişti. Bu çalışmayı ilk proje döneminde tamamladık.

Benzetim eşik değeri yükseltilmesinin katmanlarda çok hızlı sinir sayısı artışına neden olduğunu, ancak oluşturulan bu sinirlerin MNIST veri kümesindeki ortalama sınıflandırma başarımına kayda değer etkisi olmadığını gözlemledik. Elli bin eğitim örneği kullanılarak eğitilen 2 katmanlı RDÖ mimarisinin 1000, 500 ve 10 düğümden oluşan tam bağlı sinir ağı ile 10 rastgele sınıflandırma deneyinden elde ettiğimiz sonuçlar Tablo 9'da sunulmuştur.

Tablo 9. MNIST veri kümesinde farklı benzeme eşikleri ile elde edilen sınır sayısı

Benzeme Eşiği (Katman 1)	Benzeme Eşiği (Katman 2)	Elde Edilen Sınır Sayısı (Katman 1)	Elde Edilen Sınır Sayısı (Katman 2)	Ortalama Sınıflandırma Başarımı (%)
0,5	0,5	53	57	98,82
0,6	0,5	99	55	98,87
0,6	0,6	99	122	98,96
0,7	0,5	187	58	98,89
0,7	0,6	187	129	99,05

Aynı gözlemi daha ileriki proje dönemlerinde yaptığımız Fashion MNIST, BRATS, Patterned Fabric Dataset, CIFAR-10, Tiny Imagenet gibi veri setlerinde de yaptık. Sonuç olarak RDÖ algoritmasında 0,5 benzetim eşliğinin hemen her imge veri setinde en yükseğe oldukça yakın sonuçlar verdiğini gördük ve deneylerimizi bu eşikten başlayarak yaptık.

Bu iş paketi görevinde yaptığımız bir diğer çalışma da katmanlar arasında uygulanan örnekleme (pooling) evrişim aşamasında yapılması idi. Maksimum pooling türü örnekleme derin katmanlarda bir sınırın resimde hangi bölgeyi işlediğinin kaybedilmesine neden olur, bu da hata tespitinde ilgili bölgenin bulunmasına ve sınıflandırma performansına olumsuz etki eder. Bu amaçla evrişimi birden fazla piksellik kaydırmalar ile yaptık ve Tablo 10'daki başarı oranlarını elde ettik. Birinci ilerleme raporumuzda sunduğumuz bu çalışmanın ardından evrişim aşamasında örnekleme yapmayı benimsedik.

Tablo 10. MNIST veri kümesinde evrişim aşamasında boyut indirgenmesi ile elde edilen öznitelik sayıları ve ortalama başarımları. K0-K3 katman sayılarını göstermektedir.

Deney	Benzeme Eşiği				Öznitelik Sayısı				Ortalama Sınıflandırma Başarımı (%)
	K1	K2	K3	K4	K1	K2	K3	K4	
1	0,5	0,6	0,7		50	304	3712		99,05
2	0,5	0,5	0,5		50	107	110		99,17
3	0,5	0,5	0,5	0,5	50	107	110	77	99,16

Öznitelik gösterimi:

Evrişimsel sınır ağlarının ilk katmanındaki sınırlar, ağırlıkları temsil ettikleri özniteliğe bire bir denk geldiğinden kolayca gözlemlenebilirler. Ancak derin katmanlarda aynı durum geçerli değildir. Derin sınırlar, önceki katmanlardaki özniteliklerin sonuçlarını birleştirdikleri için birleşiminden bu sınırların temsil ettikleri öznitelikleri gözlemlemek için özel yöntemler geliştirilmesi gerekir.

Evrişim katmanı sonrasındaki örneklemenin maksimum pooling türü olması görüntülemeyi zorlaştıran etmenlerden biridir. Zira bu durum, derin sinirin aktivasyonunun geri yayılımını güçleştirir. Öznitelik Oluşturma görevi altında uyguladığımız evrişim zamanında örnekleme yöntemi ile geri yayılım ile gösterim mümkün hale gelmektedir. Ancak sinir uyarımı ve geriye yayılım ile görüntüleme tek çözüme sahip olmadığından bu yöntem de olasılıksal öznitelik imgeleri üretmektedir.

Bu iş paketini ilk proje döneminde tamamladık ve sonuçlarımızı izleme raporunda sunduk. Çalışmamızda derin katmanlardaki özniteliklerin gözlemlenmesini sına setindeki örnekleri kullanarak yaptık. Gözetimsiz eğitim tamamlandıktan ve evrişimsel sinir ağı son halini aldıktan sonra sinir ağını sına seti ile besledik ve derin katmanlardaki her bir siniri 0.95 eşik değerinin üzerinde ve en çok uyarıcı, yani en yüksek çıktıyı veren girdi örneklerini, sinirin temsil ettiği topaklardan gelen örnekler olarak sakladık. Şekil 6'da, MNIST veri seti üzerinde eğittiğimiz bir evrişimsel sinir ağının 4. ve son katmanında elde edilen sinirlerin temsil ettikleri topaklardan örnekler gösterilmiştir.

Öznitelik gösterimi çalışmamızda derin katmanlardaki sinirlerin sadece sınıflara değil, sınıf içi deęiřintiyi temsil eden sayı tiplerine yakınsadığını tespit ettik. Şekil 6'da görüldüğü gibi sayıların farklı yazım şekilleri farklı sinirlerde öznitelik olarak ortaya çıkmaktadır. Diğer bir deyişle, gözetimsiz olarak oluşturulan derin katmanlar sınıfları ve sınıf içi farklılaşmayı temsil etme yeteneğine sahiptirler.

Rastgele bozulma:

Proje önerisinde sunduğumuz yöntemde bir örnek bir katmandaki hiçbir siniri eşik seviyesinin üzerinde uyaramazsa, bu örnek bir Gauss dağılımından üretilen gürültü ile rastgele bozularak ilgili katmanda yeni bir sinir oluşturulmaktadır. Birinci iş paketimizdeki Rastgele Bozulma görevi, bu gürültünün nasıl bir dağılımdan elde edilmesi gerektiğini konu almaktaydı.

Projemizin ilk döneminde ilgili çalışmayı tamamladık ve proje ilerleme raporunda sunduk. Bu amaçla yaptığımız deneyler Tablo 4'te sunulmuştur.

Öznitelik 1	Öznitelik 2	Öznitelik 4	Öznitelik 5	Öznitelik 6	Öznitelik 10	Öznitelik 11	Öznitelik 14	Öznitelik 17	Öznitelik 18
5	9	9	3	5	4	4	5	4	3
5	9	9	3	5	4	4	5	4	3
5	9	9	3	5	4	4	5	4	3
5	9	9	3	5	4	4	5	4	3
5	9	9	3	5	4	4	5	4	3
5	9	9	3	5	4	4	5	4	3
5	9	9	3	5	4	4	5	4	3
5	9	9	3	5		4	5	4	3
5	9	9	3	5			5	4	3
5	9	9	3				5	4	3

Şekil 6 – MNIST veri seti ile 0,5 benzeme eşiği ile eğitilen RDÖ mimarisinin 4. ve son katmanından elde edilen sınırları 0,95 eşik değerinin üzerinde ve en çok uyararı girdiler.

Tablo 4 – MNIST veri seti üzerinde rastlantısal sınırları ilklendirilmesi kullanılarak elde edilen öznitelik sayıları ve başarımları. Bozulmada 0 ortalamaya ve değişen standart sapmaya sahip Gauss dağılımı kullanılmıştır.

Deney	Standart Sapma	Öznitelik Sayısı				Ortalama Sınıflandırma Başarımı (%)
		K1	K2	K3	K4	
1	0,000	50	107	110	77	99,16
2	0,001	50	108	114	77	99,14
3	0,010	54	117	126	81	99,06
4	0,100	71	148	173	102	89,21

Denelerimizde rastgele bozulmanın öznitelik sayısını gereksiz şekilde artırdığını ve ortalama sınıflandırma başarımına olumsuz etkisi olduğunu gördük. Oluşturulan yeni bir sinirin yeni bir topağı temsil ettiği düşüncesi ile bu sinirin girdi ile birebir aynı olması, harici durumda sinirin örnekten iraksayıp topaklar arası ortalama bir değere dönüşmesi ihtimalinin bulunması anlamlı görüldüğü için çalışmamız sonucunda rastgele bozulma işlemi RDÖ algoritmasından çıkarmaya karar verdik.

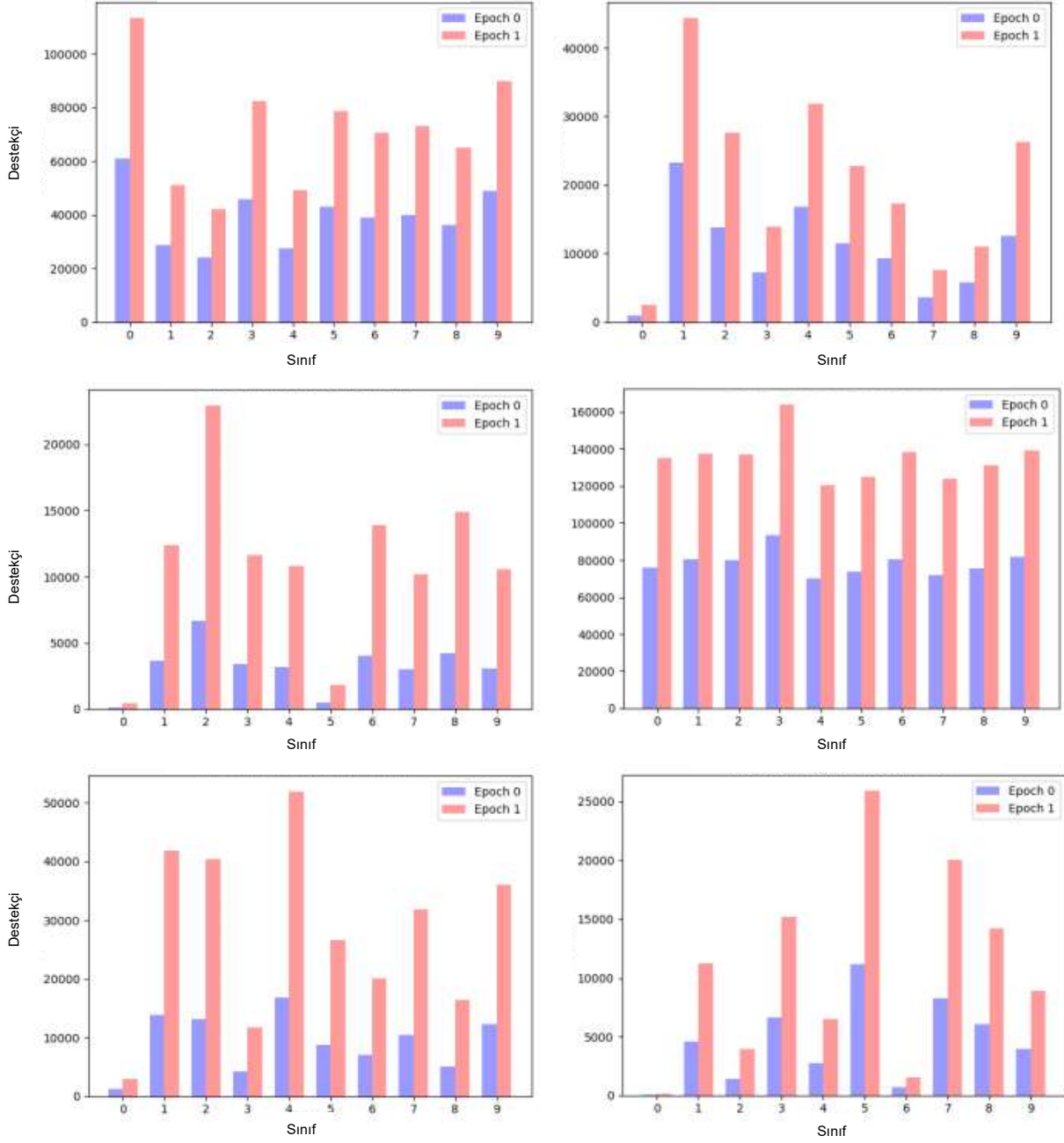
Öznitelik düşürme:

Yöntemimizde sıra dışı (kazanan sinirin olmadığı) her örüntü için yeni bir sinir oluşturulmaktadır. Oluşturulan bu sinirlerin temsil güçleri ilk proje döneminde incelenmiş ve ilerleme raporunda sunulmuştur. Sinirlerin değerlendirilmesi için öncelikli olarak bir dağınıklık (entropy) ölçütü belirlemeye çalıştık. Sinirlere ait destekçilerin sınıf dağılımlarını inceledik. Şekil 7, birinci katmanda rastgele seçilen 6 filtre için sınıf dağılımlarını gösterir. Bazı sinirler, belli sınıflardan örnekleri temsilde uzmanlaşsa da, genel olarak her sınıftan örneğe hizmet ettikleri görülüyor. Bu durum, derin katmanlarda daha da keskinleşmektedir.

Bu gözlem, derin katmanlarda elde ettiğimiz özniteliklerin sınıfları ve sınıf içi farklılaşmayı çok iyi temsil ettiğini gösterdi. Overfitting'e önlem olarak denediğimiz drop-out yönteminin sınıflandırma başarısına istatistiksel olarak anlamlı bir etkisinin olmaması da bu gözlemimizi destekleyen bir gözlemdir.

Aynı şekilde, ilerleyen proje dönemlerinde Fashion MNIST, CIFAR-10, Tiny ImageNet gibi veri kümelerinde drop-out yöntemi ile elde ettiğimiz deneysel sonuçlar, özniteliklerin sağlıklı şekilde seçildiğini ve öznitelik düşürmenin bir ihtiyaç olmadığını göstermiştir. Bu nedenlerle öznitelik düşürmenin doğru olmadığına karar verdik.

Algoritmamızın genelleme yeteneğini etkileyen tek parametre eşik değeridir. Bu parametre düşükken öbek sayısı düşük olmakta ve overfitting etkisi azalırken ağın genelleme yetkinliği artmaktadır. En yüksek eşik değeri 1 için her örnek bir sinir tarafından temsil edilmekte, örnek sayısı kadar öbek oluşumu ile overfitting gerçekleşmektedir. Doğru eşik değeri seçimiyle (önceki çalışmalarımızda bu değer [0,4 0,5] aralığında olduğunu belirlemiştik) sağlıklı öznitelik çıkarımı yapıldığı, ve bu aralıkta çalışıldığı sürece öznitelik düşürülmemesi gerektiği tespit edilmiştir.



Şekil 7 – Birinci katmanda seçilen 6 sinir için destekçilerin (en yüksek uyarıların) sınıf dağılımları.

Derin katmanlar:

Proje önerisinde katman sayısının nasıl belirleneceği ve bu katmanlarda öznitelik birleştirmenin gerekli olup olmadığı sorusuna yanıt arayacağımızı ifade etmiştik. İlk proje döneminde yaptığımız çalışmalarda katmanlarda sağlıklı özniteliklerin elde edildiğini gözlemledik. Bu öznitelikler, içinde buldukları katman derinleştikçe girdi imge üzerinde daha geniş bir bölgeyi tarıyor olurlar. En derin katmandaki öznitelikler imgeyi bir bütün olarak ele alır, kendinden önceki katmanlardaki özniteliklerin çıktılarını birleştirerek 1x1 piksellik bir çıktı oluştururlar. Bu, örneğin CIFAR-10'deki kedi sınıfı için belli kamera parametreleri ve açısı ile belli bir kedi türünün algılayıcısıdır.

Projenin ilk döneminde MNIST ve daha sonraki dönemlerde CIFAR-10, Tiny ImageNet, ve tekstil veri kümelerinde katman sayısını imgeyi tek piksele indirecek şekilde seçip ($\lceil \log_2 n \rceil$) denemeler yaptık ve önerimizin doğruluğunu sınadık. Bu çalışma bize katman sayısını hiperparametre uzayından çıkarma imkanını vermiştir.

Topaklanma başarımı:

Projeye başlamadan önce önerdiğimiz yöntemin derin katmanlarda nasıl bir topaklanma sağlayacağı konusunda endişelerimiz bulunuyordu. Bu nedenle eğitimlerimiz sırasında derin katmanlarda oluşan özniteliklerin sınıf dağılımlarını incelemeyi ve bu öznitelikleri görselleştirerek incelemeyi iş paketlerimiz arasına koymuştuk.

İlk proje dönemi içinde yaptığımız çalışmalarda derin katmanlarda oluşan özniteliklerin sınıfları ve sınıf içi değişimleri sağlıklı şekilde temsil edebildiğini gözlemledik (Şekil 6). Bu öznitelikleri destekleyen örneklerin sınıf dağılımlarına baktığımızda bazı özniteliklerin ortak sınıf özelliklerini yansıttığını, bazı özniteliklerin ise sınıfa ait özelliklere yakınsadığını gözlemledik (Şekil 7). Bu gözlemler sonucunda yöntemimizin 0.5 genel benzeme eşiği 1 piksellik evrişim atlamaları ile 4. ve son katmanda anlamlı topaklara ulaşabildiğini raporladık.

5.2. Sınıflandırma İş Paketi

CDL değişkenleri:

Projemizin ikinci döneminde, Sınıflandırma İş Paketi altında BRATS ve CIFAR-10 veri kümeleri çalışılmış ve sınıflandırma başarımını artırmak için hiperparametre eniyilemesi yapılmıştır. Daha sonraki dönemlerde çalışmamız Tiny ImageNet ile devam ettirilmiştir.

BRATS veri kümesinin HGG sınıfında her örnek 4 kipte imgeye sahip olduğu için algoritmamızı öncelikle 3 boyutlu girdi ile çalışabilecek şekilde düzenledik. Veri kümesi imgelerini 200x200 boyutlarında örnekleyerek 1 piksellik atlamalar ile eğitim örneklerini çıkardık. Benzetim eşiğini ve pencere boyutlarını deneme yanılma yöntemi ile 0,5 ve 11x11'e sabitledik. Elde ettiğimiz anomali olasılık matrisini 2 boyutlu en yüksek entropi eşiklemesi ile bölütleedik. İki katmanlı deneylerimizde en iyi Dice Score olarak 0,68 elde ettik.

CIFAR-10 veri kümesi üzerinde yaptığımız çalışmalarda en iyi sonuçları pencere boyutlarını 3x3'a indirerek, 0,3 ve 0,4 eşik değerleri ile elde ettik. Bu deneylerde başarı oranımız %62,93 oldu. Çalışmamızı ilerleterek Tiny ImageNet veri kümesinde aynı pencere boyutları ve benzer eşik değerleri ile %49,85 oranında sınıflandırma başarıyı elde ettik.

Algoritma hiperparametrelerinin eniyilenmesi için grid search yaklaşımı ile eksiksiz (exhaustive) search yapsak da literatürde raporlanan başarı oranlarına ulaşamadık. Bunun nedenleri ve çözüm önerilerimiz Tartışma konu başlığı altında sunulacaktır.

GPU dağıtımı:

Projemizin ikinci döneminde evrişim ve örnekleme (pooling) görevleri 2 GPU'ya dağıtılarak ortalama 27 saat süren 4 katlı RDÖ mimarisi eğitimini yarım saatin altına indirdik. GPU kullanımı bize 50 katın üzerinde hız kazandırdı.

5.3. Topaklama ve Sıra Dışı Örnek Tespiti İş Paketi

Arayüz hazırlanması:

Projemizin son döneminde Internet üzerinde arayüzler hazırlanarak yazılımımızın araştırmacılar tarafından denenebilmesini hedeflemiştik. Ancak bu çalışmayı yayınlarımızın gerçekleşmesinden sonraya erteledik. Tüm yazılımımız Python dilinde olduğu ve Python web deployment desteği verdiği için bu oldukça basit ve kısa sürecek bir çalışmadır.

Kavramsal öznitelikler:

Bu çalışma İlk proje döneminde tamamlanmış ve raporlanmıştır. Derin katmanlardaki öznitelikler gözlemlenerek incelenmiş ve gürbüz topaklama yapıldığı izlenmiştir.

Öznitelik birleştirme ve ayırma:

İlk iki iş paketinde özniteliklerin sınıflar arası ve sınıf içi değişintiyi temsil ettiği deneysel olarak teyit edilmiştir. Anormal doku tespitinde elde edilen öznitelikler (Şekil 4) veri kümesini doğru şekilde temsil etmektedir. Bu nedenle öznitelik birleştirilmesi veya bölünmesi yönünde bir çalışma yapılmasına gerek olmamıştır.

6. Tartışma

MNIST için literatürde raporlanmış en yüksek performans, Wan vd. (2013) tarafından raporlandığı gibi DropConnect yöntemiyle %99,79'dur. Bu çalışmada, rastgele döndürme, ölçekleme, yatay çevirme ve kırma gibi çeşitli veri artırma teknikleri uygulanır. Ayrıca bu sonuç, 5 farklı modelden oluşan bir sınıflandırıcı topluluğundan (ensemble) elde edilmiştir. Bu topluluk modelinde veri artırma olmadan en iyi sonuç, 1.000 epok eğitimden sonra %99,43 olarak rapor edilmiştir. Bu proje ile önerdiğimiz RDÖ algoritması, eğitilmesi çok daha hızlı olan basit bir mimari ile hiçbir ön işleme ve veri artırımı işlemlerine ihtiyaç duymadan, MNIST üzerinde %99,03 sınıflandırma başarısına ulaşmıştır.

CIFAR-10 ve ImageNet gibi gerçek görüntüler içeren veri kümelerinde, çok sayıda farklı benzeştirme yöntemleri, benzerlik eşikleri ve mimariler denememize rağmen literatürdeki en iyi sonuçlara ulaşamadık. Literatürdeki yüksek başarıya sahip tüm yöntemler, gözetimli eğitim ile eniyileme yaparlar. Elde edilen yapay sinir ağı, ilgili veri kümesine özgü bir çözümdür.

Gözetimsiz öğrenme ile aynı sınıflandırma başarımına ulaşamamızın temel sebebinin gözetimsiz paradigmada yeterli keşif yapamamız olduğunu düşünüyoruz. Projede başlangıç noktamız, yapay görüyü, insanın görme korteksinde olduğu gibi basitten karmaşığa örüntüler oluşturarak geliştirmektir. İnsanda Gabor süzgeçlerine benzeyen basit örüntülerin öğrenilmesinde etiket, kategori veya sınıf yoktur; öznitelik çıkarımı gözetimsiz olarak yapılır. Bu ideale ulaşmak için yaptığımız çok sayıda denemeden sonra, basit öznitelikleri çıkarmadan karmaşık veri kümelerini doğrudan öğrenmenin zor olduğunu, ancak en başarılı yaklaşımlara yakınsadığını gördük.

Görme korteksinin öğrenimini modelleyebilmek için öncelikle gerçek, yüksek çözünürlüklü ve yeterince genel çok sayıdaki imge örneğinde basitten karmaşığa örüntüleri süzgeçler olarak yakalayabilmek gereklidir. Ancak burada bilgisayar ve sunucularımızın hafıza ve paralel işleme kapasitesindeki yetersizlik önümüze bir engel olarak çıktı. Bu hedefe dönük çalışmalarımız sürmektedir. Nihai

hedefimiz, gözetimsiz öğrenmeyi imge 1 piksel x1 piksel x n öznitelik boyutlarına gelene kadar yeni katmanlarla sürdürmek ve bu son 'pikselin' temsil ettiği kavramı elle etiketleyerek gözetimli eğitimi tamamen kaldırabilmektir.

Bu proje ile, kusur tespitinde alandan bağımsız, benzerlik eşiği, kernel boyutları ve örnekleme adımı haricindeki tüm hiperparametreleri ortadan kaldıran bir yöntem önermiş olduk. Yöntemimiz, bir mekanizmanın ürettiği sinyalleri dinler, onun normal çalışma prensiplerini öğrenir ve bu prensiplerin dışında bir sinyal geldiğinde sıra dışılığı (veya nadirliği) işaret eder. Yöntemimizin kusur tespitindeki başarısını MNIST veri kümesinde hatalı eğitim örneklerinin çıkarımı, tekstilde kusur tespiti ve BRATS veri kümesinde kist çekirdeğinin bölütlenmesi problemlerinde gösterdik. BRATS'te daha yüksek başarımlar elde etmek için, yukarıdaki paragraflarda anlattığımız nedenle çok daha geniş bir eğitim kümesine ihtiyaç olduğunu düşünüyoruz.

7. Sonuç ve Öneriler

Literatürdeki derin öğrenme çalışmaları çoğunlukla geleneksel EYSA, SkipConnections, U-Net gibi halihazırda var olan mimarilerin hiperparametrelerinin eniyilenmesi ile belli bir alana uygulanmasını konu alırlar. Bu proje ile geleneksel EYSA'ya yeni bir öğrenme yöntemi önerdik. Önerdiğimiz yöntem, geleneksel EYSA'dan çok hızlı eğitilmekle kalmayıp katmanlardaki sinir sayısı ve bu sinirlerin ilklendirilmesi hiperparametrelerini ortadan kaldırmıştır. Sadece belli bir imge alanına değil, genel olarak imge işleme problemine uygulanabilir bir yöntemin ilk adımlarını oluşturmuştur.

Önerdiğimiz RDÖ algoritmasının gözetimsiz öznitelik çıkarımı ile MNIST veri kümesinde çok daha basit bir mimari ile literatürdeki gözetimli yaklaşımların en iyi sonuçlarına yaklaştık. Bu, gözetimsiz öznitelik çıkarımına doğru bir yaklaşım getirdiğimizi göstermektedir (Başarı Ölçütü 1.1). Evrimsel son katmanda sinir sayısının 50'nin altında olduğu yüksek başarımlara ulaşan bir model önerdik (Başarı Ölçütü 2.1) ve eğitim süresini yarım saatin altına indirdik (Başarı Ölçütü 2.2). Tartışma Bölümü'nde verdiğimiz öngörülerle ImageNet yarışmasına katılacak sonuçlara ulaşamadık. Ancak MNIST ve varyantı veri kümelerini kullanarak değerlendirme aşamasında olan bir dergi makalesi hazırladık (Başarı Ölçütü 2.5).

Kusur tespitinde BRATS veri kümesi üzerinde yaptığımız çalışmalarda %68 Dice Score'a ulaştık da, proje önerisinde hedeflediğimiz %15 hata payına ulaşamadık. Ancak tekstilde kusur denetiminde

literatürdeki en iyi yöntemlerden daha yüksek başarı sağlayan yöntemimiz Geniştirilmiş SCI tarafından taranan Applied Soft Computing dergisinde yayımlanmak üzere kabul edilmiştir. Aynı alanda bir patent başvurumuz da hazırlık aşamasındadır.

Projemizin ilk iş paketi olan Öznitelik Öğrenme altındaki Öznitelik Oluşturma, Derin Katmanlarda Öznitelik Gösterimi, Rastgele Bozulma, Öznitelik Düşürme, Katman Sayısı ve Topaklanma Başarımı aşamaları tamamlanmıştır. İkinci iş paketi Sınıflandırma altında RDÖ değişkenleri çalışılmış ve hiperparametre eniyilenmesi tamamlanmıştır. Evrişim ve seyrek örnekleme aşamaları GPU'lara dağıtılarak eğitim hızlandırılmıştır. İş paketinin son aşaması olan İnternet'te arayüz paylaşımı için yayınlarımızın kabulünü bekliyoruz. Üçüncü iş paketinde kusur tespiti için Kavramsal Öznitelikler çalışılmış ve Öznitelik Birleştirme olanağı incelenmiştir. Bu iş paketinin de İnternet arayüzü, yayınlarımızın kabulüne kadar bekletilmektedir.

Bu projede sezgiye dayalı; insan görme korteksinin öğrenmesinden esinlenen; literatürdeki yöntemlerin aksine basitten karmaşığa kavramsal olarak her biri anlamlı öznitelikler çıkarabilen; yine bu yöntemlere kıyasla çok daha basit bir mimari ve az sayıda sinir ile sınıflandırma ve kusur tespiti yapabilen; dolayısı ile eğitim ve test süreçlerinde çok hızlı; evrişimsel katmanları tamamen gözetimsiz eğitilen bir derin öğrenme yöntemi önerilmiştir. Yöntemimiz basitten karmaşığa öznitelikler çıkardığı için MNIST, EMNIST, Fashion MNIST gibi basit örüntülere sahip veri setlerinde başarılı olmuştur. Daha karmaşık veri setlerinde belli bir düzeye kadar başarı elde edilebilmiş, ancak daha yüksek başarı için daha yüksek çözünürlüklü gerçek imgelerde, çok sayıda basit öznitelikten başlayarak bunların birleşimi karmaşık özniteliklere doğru çıkarım yapmak gerekliliği ortaya çıkmıştır. Bu aşamada, çok yüksek hafıza ve paralel işlemedeki engeller, proje süresi içinde hedeflediğimiz başarı oranlarına ulaşmamıza engel olmuştur. Çıkarımlarımız, aşağıda başlıklar halinde sunulmuştur:

1. Gerçek görüntülerde gözetimsiz öğrenme, yüksek çözünürlüklü çok sayıda eğitim imgesinin en küçük örüntü çerçevesinden başlayarak işlenmesini gerektirir.
2. Gerçek görüntülerde çıkarılan bu öznitelikler, binlerce süzgeçle ifade edilebilir.
3. Süzgeç sayısının yüksek olması, evrişim aşamasında GPU ve hafıza engeline takılmaktadır.
4. Çözümün gerçekleştirilebilir olması için GPU kümelerinde dağıtık hesaplama yapılmasına ihtiyaç vardır.

Bu araştırma projesinde yaptığımız çalışmaların akademiye en önemli katkısı, araştırmacılara gözetimsiz EYSA eğitimi için yüksek donanımlı bir kod temeli sunmuş olmasıdır. Araştırma projelerinin doğasına uygun olarak ulaşamadığımız hedefler olmuş, ancak bu hedeflere neden ulaşamadığımız sorgulanmış, incelenmiş, ve geleceğe dönük hedefler önerilmiştir. Elde ettiğimiz deneyim, karşılaştığımız zorluklar, bulgulardan elde ettiğimiz çıkarımlar ve geleceğe dönük hedeflerimiz, bizim için çok önemli kazançlar olmuştur. Projemizden elde ettiğimiz bulgularımız doğrultusunda çalışmalarımız hız kesmeksizin devam edecektir.

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

Proje Yürütücüsü:	Doç. Dr. MUSTAFA TANER ESKİL
Proje No:	118E293
Proje Başlığı:	Evrişimsel Sinir Ağları İle Rekabetçi Derin Öğrenme
Proje Türü:	1001 - Araştırma
Proje Süresi:	24
Araştırmacılar:	
Danışmanlar:	
Projenin Yürütüldüğü Kuruluş ve Adresi:	IŞIK Ü. MÜHENDİSLİK F. BİLGİSAYAR MÜHENDİSLİĞİ B.
Projenin Başlangıç ve Bitiş Tarihleri:	15/10/2018 - 15/04/2021
Onaylanan Bütçe:	349223.0
Harcanan Bütçe:	176270.39

TÜBİTAK

Öz:

İmge üzerinde nesne tanımadaki, nesnenin imge üzerindeki yeri, açışal oryantasyonu ve ölçütü algoritmik olarak çözümlenmesi gereken önemli zorluklardır. Bu zorlukları aşmak için literatürde en sık kullanılan ve en başarılı sonuçları veren yaklaşım evrişimsel yapay sinir ağlarıdır (EYSA). EYSA yaklaşımında katmanlar halinde evrişimsel süzgeçleme yapılır. Her katman, bir önceki katmanın sonuçlarını kullanarak daha karmaşık öznitelikler elde eder. Son katmandan elde edilen en soyut öznitelikler ile sınıflandırma yapılır.

EYSA yaklaşımının en kritik aşaması süzgeçlerin algoritmik öğrenilmesidir. Bu görev için ağırlıklı olarak gradyan temelli geri yayılım (backpropagation) öğrenme algoritması kullanılmaktadır. Gradyan temelli algoritmalarda çoğunlukla hem katman sayısı, hem de katmanlar için belli sayıda sinir seçilir ve bu sinirlerin girdi ağırlıkları ilklendirilerek arama uzayında başlangıç noktası sabitlenir. Eğitim, sınıflandırmanın yapıldığı en son katmandan ilk katmana doğru, sinirler arası bağlantılar düzenlenerek yapılır.

Katmanlarda oluşturulması gereken sinir sayısı hem veri kümesinin ne kadar zorlu olduğuna, hem de veri kümesindeki örneklerin boyut sayısına bağlıdır. Arama uzayının büyüklüğü, verinin boyut sayısı ile üstel ilintilidir. Bu durum, pek çok problemde doğru sayıda ve uygun ilk değerlerle sinir oluşturulmasını zorlaştırır. Her sinirin çok sayıda girdisi olduğu için eğitim aşamasında sınıflandırma hatasının girdi sağlayan sinirlere doğru paylaşılması zordur. Hatanın eniyilenmesi gradyan üzerinde gerçekleştirildiği için sinir ağırlıkları ancak küçük adımlarla güncellenebilir. Bu durum imge örnekleri üzerinde çok sayıda epok yapılmasını zorunlu kılar. Bu nedenlerle, gradyan temelli yaklaşımlarda (1) başarı rastgele ilklendirmeye bağlıdır, (2) aşamalar kredi atama problemine (credit assignment problem) açıktır ve (3) eğitim, örnekler üzerinde çok sayıda tekrar gerektirmesi nedeniyle yavaştır.

Bu projede, EYSA mimarisi için gradyan temelli olmayan, gözetimsiz bir makine öğrenmesi yöntemi önerilmiştir. Önerilen rekabetçi derin öğrenme (RDÖ) yaklaşımı her katmanda sinirler arası rekabete dayalıdır. Yöntemimizde eğitim, katmanlar halinde yapılır. Her katman boş olarak başlar. İlk eğitim örneği, özgün bir öznitelik kabul edilerek süzgeç haline getirilir ve katmanın tek siniri olarak ilklendirilir. Eğitim örneği için katmanda sinir varsa, çıktılar hesaplanır. Belli bir eşğin üzerinde etkinleşen sinirler arasında en yüksek çıktıya sahip sinir kazanan olarak seçilir ve öğrenme sadece bu sinir üzerinde, eğitim örneğine doğru evrilerek olur. Böyle bir sinir mevcut değilse yeni bir sinir oluşturulur. Tüm eğitim örnekleri işlendiğinde, katmanda eğitim kümesindeki örüntülerin değişkenliğinin temsili için yeterli sayıda sinir oluşturulmuş olur.

Sinirlerin eğitim sırasında gerektikçe ve eğitim örneklerine yakınsayacak şekilde oluşturulması yoluyla EYSA'dan iki önemli hiperparametreyi; sinir sayısı ve sinir ağırlıklarının ilklendirilmesini kaldırdık. Sınıflandırma aşaması için, en derin katmanda öğrenilen karmaşık öznitelikleri tam bağlı bir sinir ağına girdi olarak kullanarak gözetimli öğrenme gerçekleştirdik. Bu sayede, sık evrişimsel katmanlarda kredi (hata) atama ve istikrarsız gradyanlar problemlerini ortadan kaldırmış olduk. Sinirlerin rastgele değil, eğitim örneklerine dayalı olarak oluşturulması ile evrişimsel katmanlarda öznitelik çıkarımını her bir katman için bir epokta tamamladık. Katmanlarda rastgele ve gereğinden fazla sinir oluşturulmaması; sinirlerin eğitim örneklerine dayalı olarak oluşturulması sayesinde az sayıda tekrar yapılması; bunlara ek olarak öznitelik çıkarımında geri yayılım aşamasının olmaması eğitim sürecimizi kayda değer oranda hızlandırdı. Önerdiğimiz RDÖ yöntemini, literatürde sık kullanılan MNIST, CIFAR-10 ve Tiny ImageNet veri kümeleri üzerinde sınıflandırma deneyleri ve BRATS veri kümesi üzerinde tümör bulma problemi üzerinde denedik ve sonuçlarımızı raporladık.

<p>Abstract:</p>	<p>Varying location, orientation and scale are the biggest challenges in object recognition with image processing. The most popular approach to tackle these challenges is the Convolutional Neural Networks (CNN) where filtering is done in convolutional layers that scan the entire image. Each layer of a CNN processes the outputs of the previous layer to extract more and more abstract and descriptive features. Classification is performed on the outputs of the last feature extraction layer through conventional supervised learning.</p> <p>The most critical stage of the CNN approach is algorithmic training of the filters in the hidden layers. One way to learn the filters is the gradient based backpropagation (BP) algorithm. Gradient based algorithms typically initialize a preset number of neurons with random weights, fixing the starting point in the search space. BP starts with the outputs of the last layer and progresses backwards as it distributes the credit (or error) towards the first layer, updating weights layer-wise.</p> <p>The number of neurons to be initialized in a layer depends on both the complexity of the data set and the dimensionality of the samples. This dependency makes it impractical to initialize a correct number of hidden neurons with suitable weights for most problems. Distribution of classification error to neurons is a challenge as each neuron has multiple inputs. Error minimization is performed on a gradient; therefore weight updates can only be made with small steps. As a result, gradient based methods require repetitive exposure of training samples until convergence. Consequently in gradient based methods; (1) the success depends on random initialization, (2) learning stages are vulnerable to the credit assignment problem and (3) training is slow as it requires numerous epochs on samples.</p> <p>In this project we proposed an unsupervised learning algorithm to be used with the CNN architecture. In the proposed competitive deep learning (CDL) algorithm, neurons of a layer compete with each other to become a winner. The training of the network is performed layerwise and each layer starts out empty. The first training sample is considered a novel feature, which is converted into a filter and initialized as the only neuron of the layer. In case the layer is not empty, we calculate the outputs of the neurons for the input. The neuron that is stimulated the most and above a given similarity threshold is considered the winner and its weights are updated to evolve towards the input. When no such neuron exists, a new neuron will be initialized for this novel input. Therefore at the end of the training, a required and enough number of neurons are generated and initialized for representing the variance in the training set.</p> <p>A new neuron is generated only when the layer does not embody a neuron that is a representative of the given sample. This eliminates two important hyperparameters from CNN; the number of neurons and initialization of neuron weights. We performed classification by feeding the outputs of the deepest, most abstract and representative neurons as input to a fully connected network, which is to be trained in a supervised manner. Consequently, we eliminated the credit assignment and vanishing or exploding gradients problem in the shallow convolutional layers. Generation of new neurons based on the training examples significantly speeds up the convergence rate of the neuron weights and training is usually completed in a single epoch. Creation of exactly enough number of neurons and their targeted initialization together with the lack of back propagation significantly sped up our training process. We experimented on the popular MNIST, CIFAR-10 and Tiny ImageNet datasets to assess the classification performance; and on BRATS dataset to assess the tumor detection performance of the proposed algorithm.</p>
<p>Anahtar Kelimeler:</p>	<p>Evrşimsel yapay sinir ağları, rekabetçi öğrenme, ilklendirme, toplama, öznitelikler</p>
<p>Fikri Ürün Bildirim Formu Sunuldu Mu?:</p>	<p>Evet</p>