

**DERİN ÖĞRENME YAKLAŞIMI İLE OTOMATİK KUMAŞ HATA
TESPİTİ SİSTEMİ İÇİN PERFORMANSI VE VERİMİ YÜKSEK
ALGORİTMA GELİŞTİRME PROJESİ SONUÇ RAPORU**

Proje No : M-609
Tür : B-Tipi Bireysel Araştırma Projesi
Proje Yürütücüsü : Yrd. Doç. Dr. Ahmet Gürkan Yüksek
Proje Araştırmacısı : Arş. Gör. Abdulkadir Şeker

ÖZET

Kumaş hatası tespiti tekstil endüstrisi alanında kalite aşamalarında kritik önem arz etmektedir. Hataların tespitinde, gelişen pazar hacmi ve üretim kapasitelerinin büyüklüğü sebebiyle insan görüşü ile tespit, büyük oranda zaman kaybına ve hata tespit oranının %60 seviyelerine kadar düşmesine sebep olmaktadır. Bundan dolayı hata tespitinde otomatik sistemler geliştirilmeye başlanmıştır. Bu bağlamda daha yüksek başarımlar elde edebilmek için görüntü işleme alanında birçok yöntem denenmiştir.

Kumaşın kendine has bir dokusunun olması sebebiyle, öz nitelikleri çıkarılırken diğer görüntü türlerinden ayrı olarak incelenmesi gerektirmektedir. Öz nitelikler bilgisayarlı görmede özellikle sınıflandırma problemlerinde ham madde olmaktadır. Bu yüzden doğru öz nitelikleri çıkarmak, hata tespitinde en önemli aşamadır. Bu amaç doğrultusunda, çoklu-katman mimarisi ve kendi öz niteliklerini çıkararak son yıllarda görüntü ve ses işleme alanında büyük başarılar getirmesi ile öne çıkan derin öğrenme kumaş hatası tespitine uygulanmıştır. Giriş verisini sıkıştırma ya da genişletme ile temsil eden (stacked autoencoder) yığınlı oto kodlayıcılar -bir derin öğrenme yöntemi- kumaş hatası tespitinde denenmiş ve kabul edilebilir başarımlar elde edilmiştir. Çalışmanın asıl amacı oto kodlayıcının hiper parametreleri ve giriş değeri ile oynamalar yaparak öz nitelik çıkarımı başarısını artırmaktır.

1. GİRİŞ

Makine öğrenmesi, bilgisayarlı görme ve yapay sinir ağları algoritmalarıyla kurulan otomasyonlar, endüstriyel sektörlerde birçok iş aşamasında insan gücü yerine kullanılmaya başlanmıştır. Özellikle bazı tehlikeli veya aşırı hassas olunmasını gerektiren işlerde bu tarz sistemler kullanılmıştır. Seri üretim yapılan sistemlerde sürekli ve hızlı bir akış olduğunda, bu sürecin insan görüşüyle takip edilmesi zorlaşmaktadır.

Kumaş hatası tespiti işlemi de bu tarzda süreklilik olan bir aşamadır. Geleneksel tespitite, kontrol işçisi aydınlatılmış bir inceleme masasından dakikada 8-10 metre hızla geçen 2-3 metrelik kumaş toplarında hata tespiti yapmaktadır. Hata gördüğü durumlarda sistemi durdurup hata olan bölgeyi işaretleyip kaydeder. Bu işlemin sonunda kumaş topundaki hata oranına göre kabul/ret işlemi yapılır. Tespit işlemi uzun ve yorucu bir işlemdir. Sürekli aynı noktaya bakan gözlemciler göz yorgunluğu sebebiyle en fazla 1 saat denetim yapabilmektedir. Geleneksel sistemlerde hata oranı bunlara rağmen %60'larda çıkmaktadır. Ayrıca bu türden işler yapan işçilerde Alzheimer hastalığı görülme riski de artmıştır. Bu sebeplerden dolayı kumaş hata tespitinde bilgisayarlı sistemler denenmeye başlanmıştır.

Yapay sinir ağları ile kumaş hatası tespiti daha önce çalışılmış bir konudur. Donanımsal gelişmeler ve GPU tabanlı algoritmalar sayesinde yapay sinir ağlarında gizli katman sayısının artırılmasına imkan sağlamıştır. Bu tür mimarilere genel manada derin ağlar denilmektedir. Derin öğrenme başlığı altında çalışılan bir çok konu ve algoritma bulunmaktadır. Bunların hepsinin ortak yönü ise, farklı bilgi seviyelerinde oluşturulmuş derin ağlar içermesidir. Bu sayede en iyi özneteliklerin ortaya çıkarılacak ve daha başarılı ağlar oluşturulacaktır. Derin öğrenme algoritmaları önce sinyal ve ses işleme alanlarında, daha sonra ise görüntü işleme alanına uygulanmıştır. Elde edilen başarılarından yola çıkarak, bu projede de kumaş hatası tespitinde derin öğrenme algoritmaları kullanmak hedeflenmiştir.

2. KUMAŞ HATALARI TESPİTİ

2.1. Kumaş Hata Türleri

Hata, bir şeyin mükemmelliğini veya bütünlüğünü bozan herhangi bir eksiklik/fazlalık olarak tanımlanabilir.

Kumaş yün, pamuk, polyester veya bunların karışımlarından oluşan tekstil ipliklerinde üretilir. Kumaş hatası, kumaşta tüketici tarafından kabul edilemeyecek büyüklükte herhangi bir ayıp veya düzensizlik anlamına gelir.

Kumaş hatası türleri de kumaşın tipine göre ayrılmaktadır. Kumaş tipleri ise, üretildiği ham maddeye veya üretim şekline göre sınıflandırılır.

Dokuma kumaşlardaki hatalar:

Dokuma kumaşlar çözgü ve atkılardan oluşur. Hatalara genel olarak çevresel etkenler veya iplikteki bozukluklar sebep olur. İplikte renkli yabancı maddeler bulunduğunda renkli benekler (coloured flecks), dokuma aşamasında iplik kopmasıyla meydana gelen kaçıklar(knots), bir yerine iki ya da daha fazla uç dokunduğunda oluşan çift uç (double ends) hataları başlıca hatalardır. Aşağıda Şekil 1.1’de hatalardan bazıları gösterilmiştir.



Şekil 2.1 Dokuma kumaşlardaki bazı hatalar

Kadife ve havlu kumaşlardaki hatalar:

Bu tür kumaşlar yine dokuma yoluyla üretilmektedir. Dokuma esnasında, dokuma yüzeyinde ham lifler belli uzunlukta bırakılarak yumuşak bir yüzey elde edilir. Genelde bu yönde hatalar meydana gelir. Yüzey üzerinde bırakılan ipliklerin uzunluğu hataya sebep olur.

Örgü kumaşlardaki hatalar:

Örgü kumaşlar iplik tipi ve ön işlemleriyle dokuma kumaşlardan ayrışır. Kendine özgü hataları mevcuttur. Yırtık/ delikler, yatay şeritler karşılaşılan bazı hatalardır.

2.2. Kumaş Hatası Ölçülendirme Sistemleri

Kumaş hataları hem kumaş üreticileri hem de satıcıları için geçerli olan bir sistem ile değerlendirilmelidir. Bunun için yapılan çalışmalarla iki tarafından kabul ettiği derecelendirme sistemleri oluşturulmuştur. 4-puanlı ve 10-puanlı hata derecelendirme sistemi bunlardan en çok kabul görenleridir. 4-puanlı sistemde hatanın boyutuna göre 1-4 arası bir ceza puanı verilmektedir. Bu sistem genişlik sınırlaması olmaması ve kolay uygulanabilmesi sebebiyle çok tercih edilmektedir. 10-puanlı sistemde ise değerlendirme uzunluk ve genişlik olarak yapılmaktadır.

DEFECT SIZE	PENALTY
3 inches or less	1 Point
Over 3 inches, but less than 6	2 Point
Over 6 inches, but less than 9	3 Point
Over 9 inches	4 Point

Şekil 2.2 Kumaş hatasında 4-puanlı derecelendirme sistemi

2.3. Kumaş Hatası Tespit Teknikleri

Kumaş hatası tespiti için iki seçenek mevcuttur. İlki dokuma esnasında hatanın mevcudiyetini kontrol etmek, ikincisi ise üretim aşamasında yapılan tespittir. İlk aşama daha karmaşık olduğundan genellikle üreticiler ikinci seçenek üzerine yoğunlaşmıştır.

Manuel kontrol yapılırken, kumaş topu ışıklandırılmış bir masa üzerinden dakikada 8-10 metre hızla geçirilir, kontrol yapan kişi hata ile karşılaştığında kumaşın akışını durdurur ve hatayı işaretleyip kaydeder.

Otomasyon sistemleriyle kontrol yapılırken ise, 180 derece konumlandırılmış kameralar, görüntü yakalayıcılar, bilgisayarlar ve ışıklandırma sistemleri ile elektriksel ve donanımsal araçlardan oluşan bir sistem kullanılır. Burada en önemli faktör doğru ışıklandırmadır, çünkü görüntünün kalitesini belirleyen ışık seviyesidir ve türüdür. Kumaşın akış hızı, doğru tip kameralar seçilmesi de hata tespitini etkileyen diğer araçlardır.

2.4. Hata Tespit Etme Algoritmaları

Kumaş hatası tespitinde kullanılan çok çeşitli yaklaşım ve bunların içinde farklı algoritmalar bulunmaktadır. Öncelikle hata türleri sınıflandırılmış, kritik olan hatalar üzerinde çalışmalar yapılmıştır. Hata türleri incelenirken önce motif tabanlı olanlar ve sade kumaşlar birbirinden ayrılmıştır. Desenli kumaşlarda tespit işlemi çok daha zorlayıcı ve doğruluk oranı bir hayli düşüktür. Desensiz kumaşlar diğerine göre daha kolay tespit edilebilmektedir. Bu türdeki kumaşların hata tespitinde ise p1 olarak adlandırılan kumaş türleri üzerinde çalışmalar yoğunlaşmıştır. İstatiksel, izgesel (spectral), model tabanlı, öğrenme, yapısal yaklaşımlar çeşitli algoritmalar ile denenmiştir. Bunlar içinde en başarılı olanlar izgesel yaklaşımlar içinde fourier dönüşümü ve dalgacık dönüşümü teknikleridir. Bunların dışında destek vektör makinaları, yapay sinir ağları da hata tespitinde kullanılmıştır.

Bu projede ise derin öğrenme algoritmaları hata tespitinde kullanılmıştır.

3. DERİN ÖĞRENME

3.1. Derin Öğrenme Nedir?

Derin öğrenme, veride çoklu işlem katmanları tarafından yüksek seviyeli soyutlamalar modellemeye çalışan bir takım algoritmalara dayanan bir makine öğrenme yöntemidir. Kabaca ifade etmek gerekirse, ağdaki her gizli katmanın farklı seviyede öznitelikler çıkarmaya çalıştığı bir yöntemdir. Derin öğrenmenin gücü çok sayıda giriş ile ağı eğiterek kendi özniteliklerini kendisi çıkarabilmesinden gelmektedir. Bu sayede doğru öznitelik çıkarma işlemi en iyi şekilde çözülmüş olacaktır. Doğru öznitelikle yapılan eğitimlerin sonucunda ise, hata payı çok düşük olan modeller ortaya çıkacaktır.

Derin öğrenme fikrinin temeli 1960'lara dayansa da, o zamanın donanımsal sıkıntıları sebebiyle başarılı olamamıştır. Çoklu gizli katmana sahip modeller GPU tabanlı algoritmalar sayesinde tekrar ön plana çıkmış, 2006 yılında Toronto Üniversitesi'nde Prof. Geoffrey Hinton ortaya attığı derin öğrenme ağı, bu alan için bir milat olmuştur. Bu tarihten sonra birçok alanda derin öğrenme yöntemleri kullanılmıştır.

3.2. Derin Öğrenme Algoritmaları

- I. Derin Boltzmann Makinaları: (DBM)
- II. Oto-kodlayıcılar (AE)
- III. Derin İnanç Ağları (DBN)
- IV. Konvolüsyonel Sinir Ağları (CNN)

Bu projede öncelikle CNN uygulamaları yapılmış, kumaş üzerinde uygun olmadığı için AE algoritmasıyla proje gerçekleştirilmiştir.

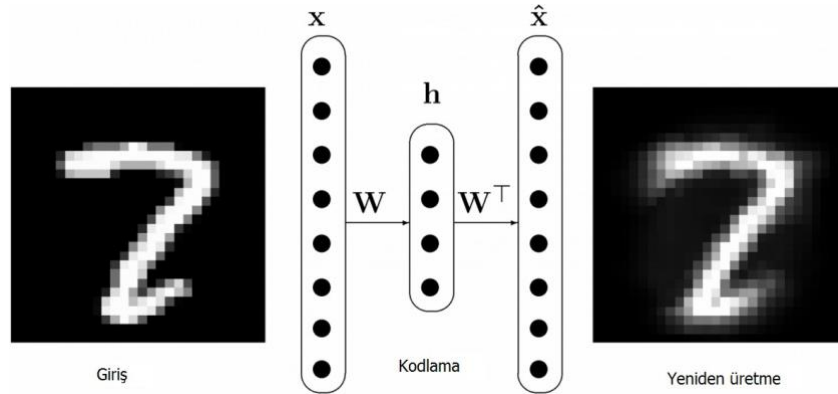
4. PROJE ÖN HAZIRLIK AŞAMASI

4.1. CNN Algoritması

Konvolüsyonel Sinir ağları ile yapılan çalışmada önceden eğitilmiş bir model ile kedi-köpek resimlerinden oluşturulan bir veriseti üzerinde sınıflandırma problemi çözülmüştür. Aynı model ile kumaş hatası tespiti uygulanılmak istenmiş ancak başarılı olmamıştır. Çünkü modelin eğitimi için kullanılan resimler daha çok insan ve nesne tabanlı resimler içermektedir, bu veriseti ise kumaş dokusunu ve özelliklerini kapsamamaktadır. CNN algoritması kullanılarak yapılan çalışma 1.Raporda detaylı bir şekilde anlatılmıştır.

4.2. AE Uygulamaları

Oto-kodlayıcılar kendi girişini çıkışında tekrarlamayı hedefleyen özel bir sinir ağı modelidir (Şekil 4.1). Bunu yaparken ara katmandaki (h) ağırlıklar sürekli güncellenir ve sonunda girişi en iyi temsil eden (özet veri) öznitelikler elde edilmiş olur.

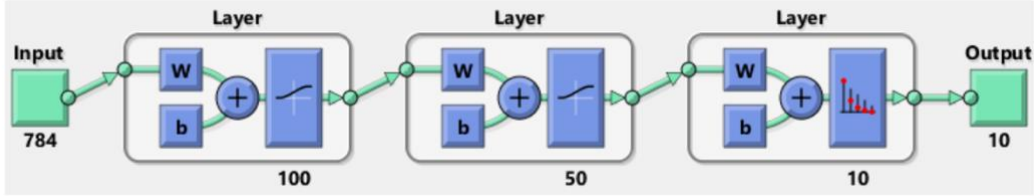


Şekil 4.1. Oto-kodlayıcı Modeli

AE modellerinde sistemin ihtiyacına göre sıkıştırma veya genişletme yapılabilmektedir. Gizli katmandan üretilen veri giriş olarak kullanılarak, başka bir AE daha eğitilir. Bu şekilde AE modellerin birleştirilmesiyle oluşan derin ağlara ise yığınlı (stacked) oto-kodlayıcı veya derin oto-kodlayıcı (Deep Autoencoder) denilmektedir.

Kumaş verisinden önce AE modelleri daha iyi anlamak adına sentetik el yazısı veriseti (MNIST) ile bir demo yapılmıştır. Projede yapılan bütün testler matlab ortamında gerçekleştirilmiştir. Giriş verisi 28 x 28 çözünürlükteki 5000 rakam görüntüsünden oluşmaktadır. Her rakam için 500 görüntü elde edilmiştir.

Yığınlı modelde öncelikle 784 düğümden oluşan giriş verisi 100 düğüme indirilmiş, ardından 100 giriş 50 düğümlle temsil edilmiş, buradan elde edilen öznitelikler ile bir softmax katmanı eğitilmiş ve 10 çıkış alınmıştır (Şekil 4.2). Softmax katmanında verilen bir giriş değeri için, her çıkışa 0-1 arasında bir değer verir, bunlardan en yüksek olanı tahmin olarak alınır.



Şekil 4.2. Derin Oto-kodlayıcı Modeli

Yapılan eğitim ve test sonucunda bir karışıklık matrisi (confusion matrix – CM) üretilmiştir. Bu matris test sonucunda yapılan tahminleri ve gerçek değerlere göre bir doğruluk (accuracy) ölçütü ortaya koymaktadır.

Confusion Matrix											
Output Class	1	2	3	4	5	6	7	8	9	10	
	458 9.2%	2 0.0%	1 0.0%	2 0.0%	1 0.0%	18 0.4%	11 0.2%	9 0.2%	0 0.0%	4 0.1%	90.5% 9.5%
	6 0.1%	473 9.5%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	10 0.2%	4 0.1%	0 0.0%	6 0.1%	94.4% 5.6%
	4 0.1%	1 0.0%	436 8.7%	3 0.1%	23 0.5%	5 0.1%	5 0.1%	12 0.2%	5 0.1%	4 0.1%	87.6% 12.4%
	0 0.0%	2 0.0%	0 0.0%	486 9.7%	0 0.0%	4 0.1%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	98.4% 1.6%
	3 0.1%	0 0.0%	37 0.7%	0 0.0%	462 9.2%	2 0.0%	0 0.0%	12 0.2%	1 0.0%	0 0.0%	89.4% 10.6%
	5 0.1%	1 0.0%	9 0.2%	1 0.0%	5 0.1%	461 9.2%	0 0.0%	9 0.2%	8 0.2%	2 0.0%	92.0% 8.0%
	23 0.5%	12 0.2%	5 0.1%	0 0.0%	0 0.0%	0 0.0%	473 9.5%	2 0.0%	0 0.0%	1 0.0%	91.7% 8.3%
	0 0.0%	2 0.0%	10 0.2%	3 0.1%	6 0.1%	7 0.1%	0 0.0%	451 9.0%	1 0.0%	6 0.1%	92.8% 7.2%
	1 0.0%	0 0.0%	1 0.0%	3 0.1%	3 0.1%	1 0.0%	0 0.0%	1 0.0%	480 9.6%	5 0.1%	97.0% 3.0%
	0 0.0%	7 0.1%	0 0.0%	1 0.0%	0 0.0%	2 0.0%	1 0.0%	0 0.0%	3 0.1%	472 9.4%	97.1% 2.9%
Target Class											93.0% 7.0%
	1	2	3	4	5	6	7	8	9	10	

Şekil 4.3 Rakamlar için üretilmiş CM

CM öğrenmenin kalitesini göstermektedir. Satırlar tahmin edilen sınıfları, kolonlar ise örneklerin aslında hangi sınıfa ait olduğunu gösterir. Köşegendeki hücreler doğru tahmin edilen örneklerin sayısını ve yüzdesini verir, diğer hücrelerde ise sınıflandırıcının yaptığı hatalar gösterilir. En sağdaki sütun tahmin edilen sınıfların doğruluğunu verir. En alttaki satır ise, gerçek sınıf için doğruluğu verir. Sağ en alt hücre ise modelin toplam doğruluk değerini ifade eder. Şekil 4.3'te gösterilen matriste 10. Kolon sıfır(0) rakamını temsil etmektedir. Derin ağ modeli, en yüksek doğrulukta dokuz (9) rakamını tanımıştır. Toplam doğruluk %93 olarak görülmektedir.

4.3. Kumaş Veriseti

Kumaş veriseti araştırmaları sonucunda Almanya Freiburg Üniversitesi'nde bir çalışma grubunun oluşturduğu TILDA kumaş veriseti bulunmuştur. Bu verisetinde biri hatasız, diğerleri çeşitli hataları barındıran 8 farklı sınıfta kumaş fotoğrafları bulunmaktadır. Ancak veriseti online erişime açık değildir. Bu yüzden üniversite ile olan görüşmeler ve veriseti elde etme süreci uzama ihtimaline karşın, kendi verisetinin üretilmesi fikri daha mantıklı bulunmuştur.

Bu amaç doğrultusunda bir tekstil işleme fabrikasından alınan hatalı kumaş örnekleri ile veriseti oluşturulmuştur. Veri seti için kaynak fotoğraflar bir DSLR fotoğraf makinesiyle farklı açılardan çekilmiştir. Kumaş parçalardan kırıksıklık, gölge vs etkenlerden dolayı kaliteli görüntüler alınamadığından, parçalar beyaz mukavvalar üzerine gerdirilerek zımbalanmıştır. Makinede diyafram, perde açıklığı, ISO ayarları manuel olarak yapılmıştır. Bu sayede daha kaliteli görüntüler elde edilmiştir.

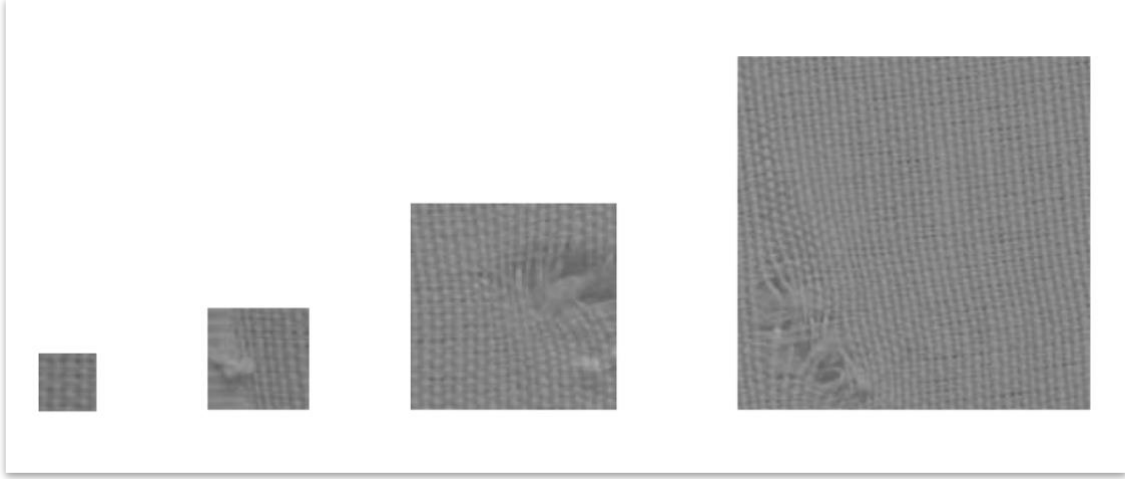


Şekil 4.4 Kumaşların fotoğrafların çekilmesi

Kaynak fotoğraflar kaliteye göre elenmiş, 10 farklı hatalı/hatasız fotoğraf ile veriseti oluşturulmaya başlanmıştır. Derin öğrenme algoritmalarının verimliliğinde en önemli faktör örnek sayının fazla olmasıdır. Bu sebeple elimizde 10 kaynak fotoğraftan matlab ile yazılmış aşağıdaki script ile rastgele pencereler alınmış, 5000 örnekli veriseti hazırlanmıştır.

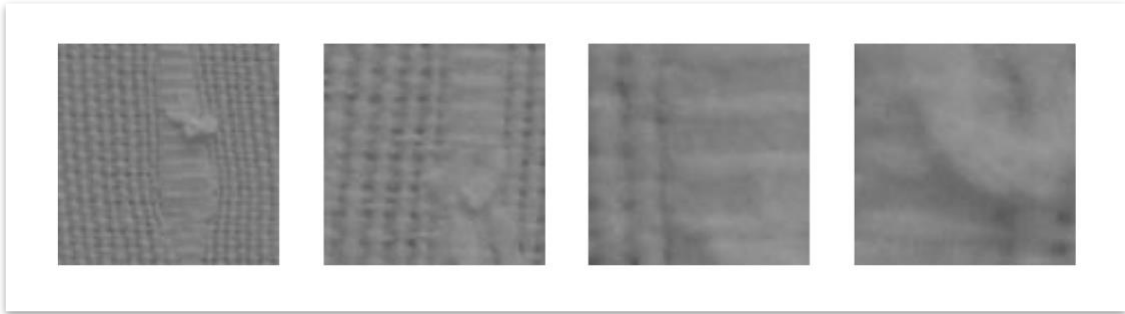
```
%Görüntüden rastgele pencereler alan script.
clear;
clc;
frame = -1; %Alınacak pencere büyüklüğü. (100x100)
NumberOfSamples = 500; %bir görüntüden alınan örnek sayısı
for s = 1:1
    resize_param=0; %Kaynak resmin yeniden boyutlandırma parametresi
    frame = frame+100;
    for r = 1:5
        resize_param=resize_param+0.2;
        dir = strcat('Sample',num2str(frame+1),'_r0'
, num2str(resize_param*10));
        mkdir(dir);
        index = 1;
        for k = 1:10
            %görüntüyü yükle
            stringImage = 'FABRIC/DSC_000';
            stringImage = strcat(stringImage,int2str(k),'.JPG');
            im = imread(stringImage);
            img = rgb2gray(im);
            img = imresize(img,resize_param);
            %görüntünün boyutlarını al.
            [x,y] = size(img);
            %rastgele pixel değerleri üret.
            randx = randi([1 (x-frame)],1,NumberOfSamples) ;
            randy = randi([1 (y-frame)],1,NumberOfSamples) ;
            patches = cell(NumberOfSamples);
            for i=1:NumberOfSamples
                startx = randx(i);
                starty = randy(i);
                patches{i} =img(startx:(startx+frame),
                starty:(starty+frame));
                imwrite(patches{i},strcat(dir,'/s',num2str(frame+1)
, '_r0',num2str(resize_param*10),'_',
                num2str(k),'_',num2str(index),'.jpg'),'jpg');
                index = index+1;
            end
        end
    end
end
end
```

Alınan örnekler çeşitli pencere boyutlarında gruplandırılmış, en uygun boyutlara sahip olan grup uygulamada kullanılmıştır. 25x25 lik resimlerden 400x400 çözünürlüğe sahip resimlere kadar testler yapılmıştır.



Şekil 4.5 (25x25 – 200x200) arasında çözünürlüklere sahip örnekler

Verisetine karar verirken iki ana parametre göz önüne alınmıştır. Birincisi kaynak resmin yeniden boyutlandırılma parametresi, ikincisi alınan pencerelerin büyüklüğünün ne olacağıdır. Yeniden boyutlandırma parametresi ile hem pencerenin ne kadar bilgi/ayrıntı içerdiği hem de pencerenin ayırt ediciliği belirlenmektedir. Aşağıdaki Şekil 4.6’da aynı bölgenin farklı parametre ile yeniden boyutlandırılmış kaynak resimlerden alınmış halleri verilmiştir.



Şekil 4.6 Yeniden boyutlandırma ile aynı bölgenin farklı görüntüleri

Alınan pencereler sırasıyla 0.2, 0.4, 0.8 ve 1 ile yeniden boyutlandırılmış kaynak resimlerden alınmıştır. Pencereler aynı bölgeyi gösterse de, daha yakın/uzak mesafeden görüntü alındığı için farklı miktarlarda bilgi içermektedir. Burada bilgiden kasıt, düzenli kumaş dokusu demektir.

4.4. Veriseti Hiper parametreleri

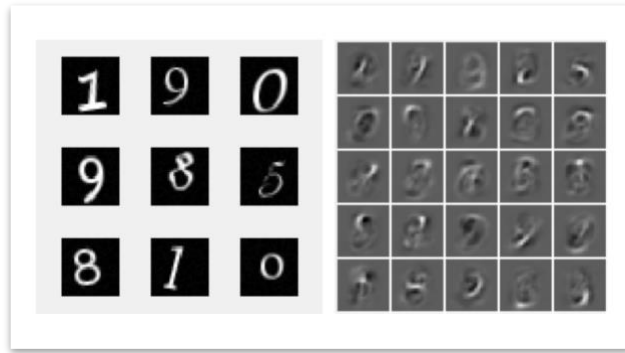
Makine öğrenmesi konularında hiper parametre adı verilen özel parametreler bulunmaktadır. Hiper parametre optimizasyon için kullanıcı tarafından ayarlanan değerlerdir. Deneme yanılma yöntemi veya önceden yapılmış çalışmalara baz alınarak ayarlanırlar. Parametre dediğimiz şey ise modelin zamanla öğrendiği değerlerdir. İki terim arasındaki fark da budur.

Bu çalışmanın asıl amacı bir derin öğrenme algoritması olan oto kodlayıcıların kumaş verisi üzerinde çalışmasını test etmektir. Bu doğrultuda oto kodlayıcıların çalışmasını sağlayacak hiper parametrelere karar vermek, yapılacak başka çalışmalara yol göstermek amaçlanmıştır. Çalışmada ayarlanan parametreler şunlardır.

- a) İterasyon sayısı** : Ağın kaç kez çalışacağını belirten değerdir. Öğrenme için gerekli olan optimum değer tespit edilecektir.
- b) Giriş boyutu** : Giriş verisinin kaç düğümden oluştuğunu söyleyen değer. 25x25 çözünürlükte bir resim için bu değer 625 olmaktadır.
- c) Giriş tipi** : Resmin ne kadar bilgi içerdiği ile ilgilidir. Yeniden boyutlandırma parametresi ne kadar küçük olursa, o kadar daha detaylı bilgi içeren görüntüler elde edilir.
- d) Gizli katman** : Gizli katman sayısı ve bu katmanlardaki düğüm sayısını verir. Giriş değerinin ne kadar sıkıştırılacağı önemlidir.

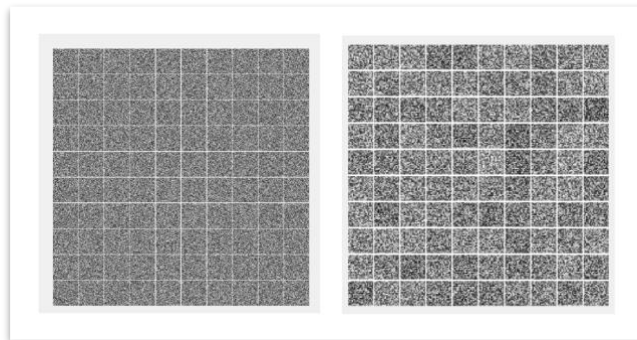
5. UYGULAMA

Derin Oto kodlayıcıların eğitimi yapılırken aynı anda bir katman eğitilir. Giriş kısmında da anlatıldığı üzere oto kodlayıcının giriş değeri sıkıştırılıp tekrar genişletilerek aynı değer çıkışta görmek hedeflenmiştir. Kodlama ve kod çözme aşamalarında aradaki gizli katman değerleri eğitimin gerçekleşip gerçekleşmediğini gösterecektir. Aşağıdaki Şekil 5.1’de MNIST veriseti ile oluşturulan modelin ilk oto kodlayıcısındaki gizli katman düğümlerinin değerleri görselleştirilerek oluşturulmuş bir görüntüler verilmiştir. Görüldüğü gibi rakamların ayırt edici özelliklerini barındıran birbirinden farklı öznitelikler ortaya çıkmıştır. Bu öznitelikler ile eğitim gerçekleştirilmiş, çok başarı sonuçlar elde edilmiştir (Bkz. Bl 4).



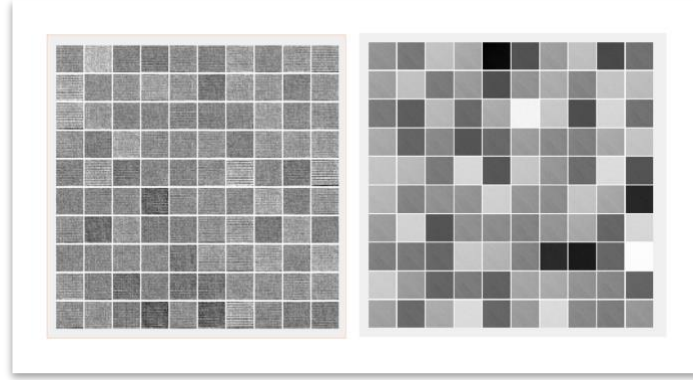
Şekil 5.1 Sentetik rakam örnekleri (solda), elde edilen öznitelikler (sağda)

Kumaş veriseti için de benzer sonuçlar çıkması beklenmektedir. Aksi takdirde eğitim gerçekleşmediği ortaya çıkacaktır. 100x100 pencerelerle yapılan ilk testte gizli katmandaki düğüm sayısı 100 seçildiğinde; 10000 giriş değerinin 100 düğüm ile temsil edileceği anlamına gelir. Bu sıkıştırmada gizli katman değerleri görselleştirildiğinde gürültü benzeri sonuçlar ortaya çıkmış ve eğitim gerçekleşmemiştir (Şekil 5.2).



Şekil 5.2 Farklı iterasyon sayıları ile 100 düğümlü gizli katman çıktıları

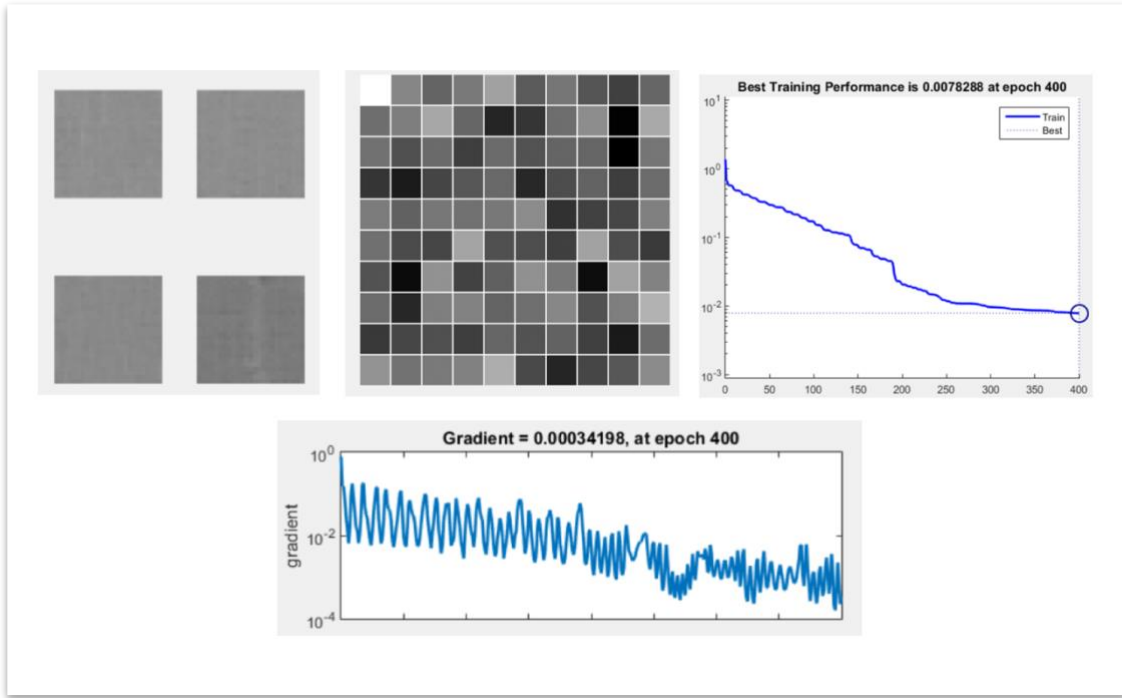
Pencere boyutu küçüldükçe daha anlamlı sonuçlar çıkmaya başlamıştır. Şekil 5.3'te 50x50 ve 25x25 için sonuçlar görselleştirilmiş, en ayırt edici öznelitliklerin 25x25'lik görüntülerden çıktığı açıkça görülmüştür. Buradan yola çıkarak 625 ten 100 e sıkıştırma yapmak, yani giriş ile gizli katman arasında bu veriseti için 1/5'lik bir oran olması iyi sonuçlar çıkardığı gözlemlenmiştir.



Şekil 5.3 Gizli katmandan çıkan öznelitlikler (solda 50, sağda 25 çözünürlük için)

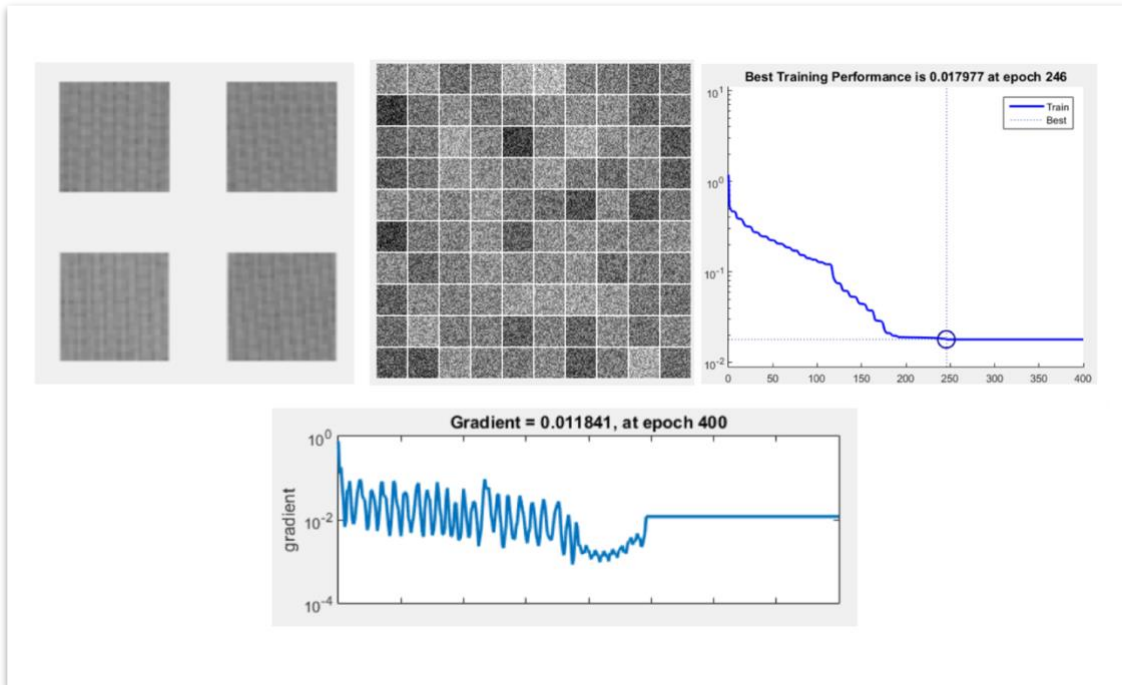
Bu eğitimin ilk aşamasıdır, ikinci aşamada oto-kodlayıcının çıktısını giriş kabul eden bir başka oto kodlayıcı kullanılacak ve sınıflandırma için işe yarar öznelitlikler buradan elde edilecektir. Bu öznelitlikler ile sınıflandırma yapabilmek için, ağız denetimli bir şekilde eğitilmesi, yani verisetinin etiketlenmiş olması gerekmektedir. Önceki katmanlarda yapılan eğitim denetimsiz olduğu için etiket verisine ihtiyaç duyulmamıştır. Ancak bu adım bir etiketleme problemini de birlikte getirmiştir. 25x25 çözünürlüklü örnekleri manuel olarak etiketlemek mümkün değildir. Bu sebeple görüntüler daha büyük boyutta alınacak ve sisteme giriş yaparken boyut küçültülecektir. Burada karar verilmesi gereken ise görüntünün hangi değerden 25x25'e düşürüleceğidir. Buna karar vermek için, yukarıda yazılan script ile farklı yeniden boyutlandırma değerleri ve pencere boyutları ile 20 grup veriseti oluşturulmuştur. Veriseti 100-r, 200-r, 300-r 400-r boyutunda ve 1, 0.8, 0.6, 0.4, 0.2 yeniden boyutlandırma parametreleriyle 4gün süren bir çalışma süresinde oluşturulmuştur. Aşağıdaki şekillerde her bir veriseti grubuna ait kodlayıcının çıktılarının görselleştirilmesi, hata değerleri, ve türev eğrisi verilmiştir. Bunlardan çıkarılan yorumlar ile doğru giriş değeri bulunacaktır.

En ayırt edici grup 1. Olandır. Türev eğrisi de aşağı yönde eğilim göstermiştir



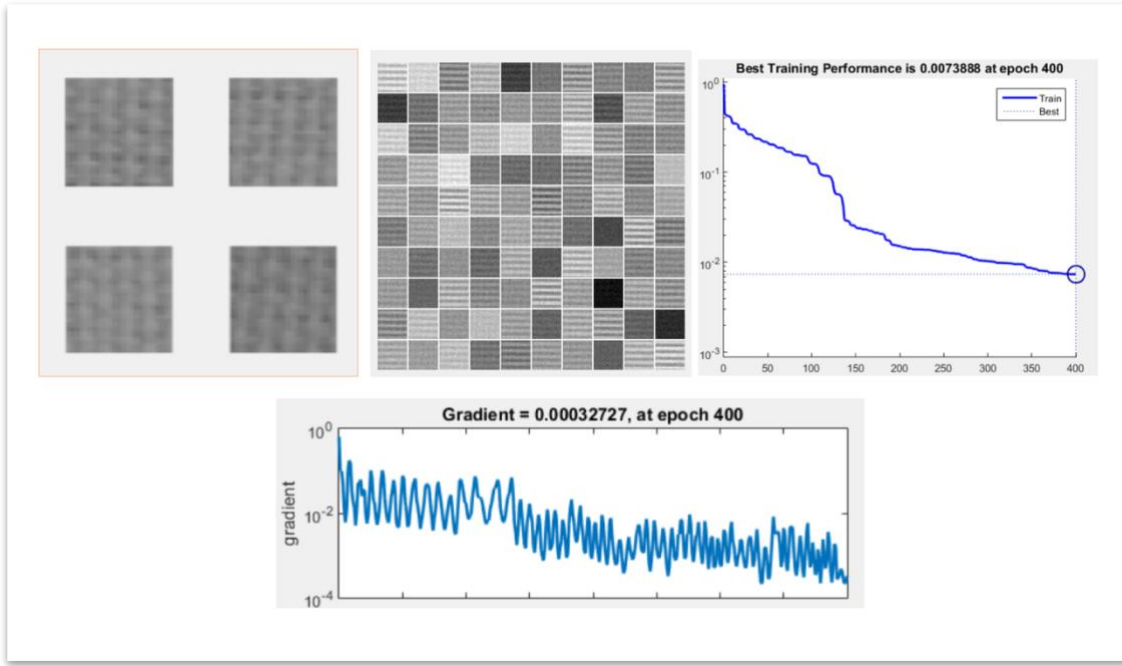
Şekil 5.4 Grup-1 “Çözünürlüğü 100- r ve yeniden boyutlandırma parametresi 0.2 olanlar”

Grup-2 hata çok yüksek değerde kalmış, türevde bir azalma görülmemiştir.



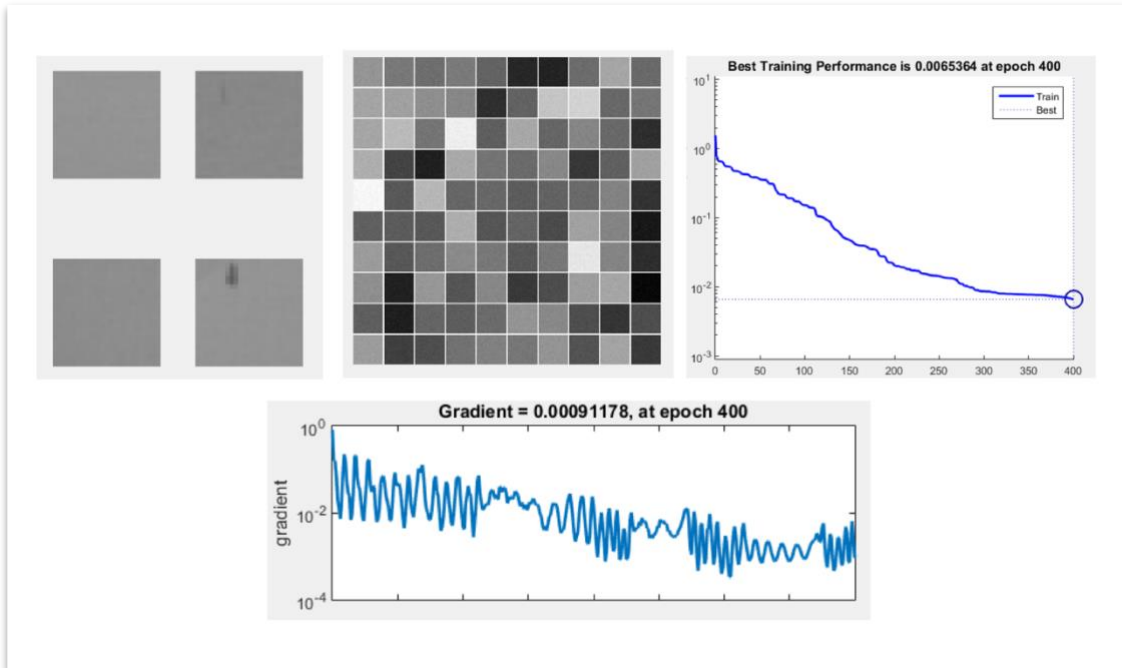
Şekil 5.5 Grup-2 “Çözünürlüğü 100- r ve yeniden boyutlandırma parametresi 0.4 olanlar”

Grup-3 hata değeri düşüktür, fakat öznitelikler yeterince ayırt edici değildir.



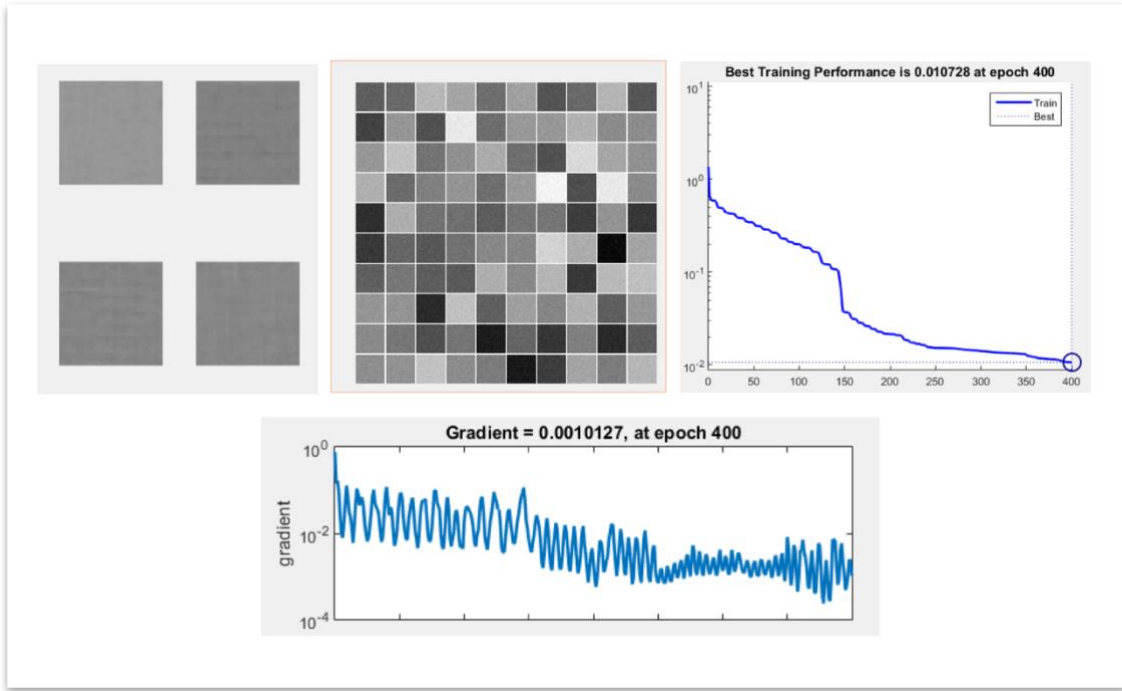
Şekil 5.6 Grup-3 “Çözünürlüğü 100- r ve yeniden boyutlandırma parametresi 0.8 olanlar”

Grup-4, grup-1 gibi güzel sonuçlar vermiştir. Ancak eğitim süresi açısından bakıldığında 1.grup daha kısa sürede eğitilmesi tercih sebebidir.



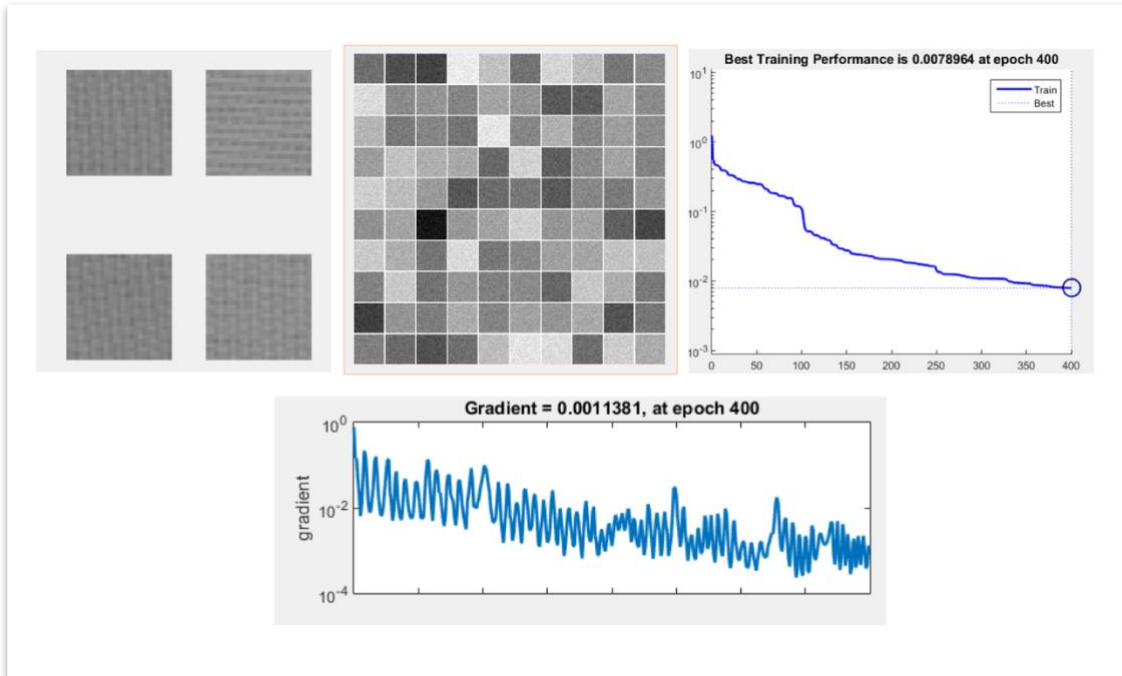
Şekil 5.7 Grup-4 “Çözünürlüğü 200- r ve yeniden boyutlandırma parametresi 0.2 olanlar”

5.grup değerleri 4.’ye kıyasla yeterli değildir.

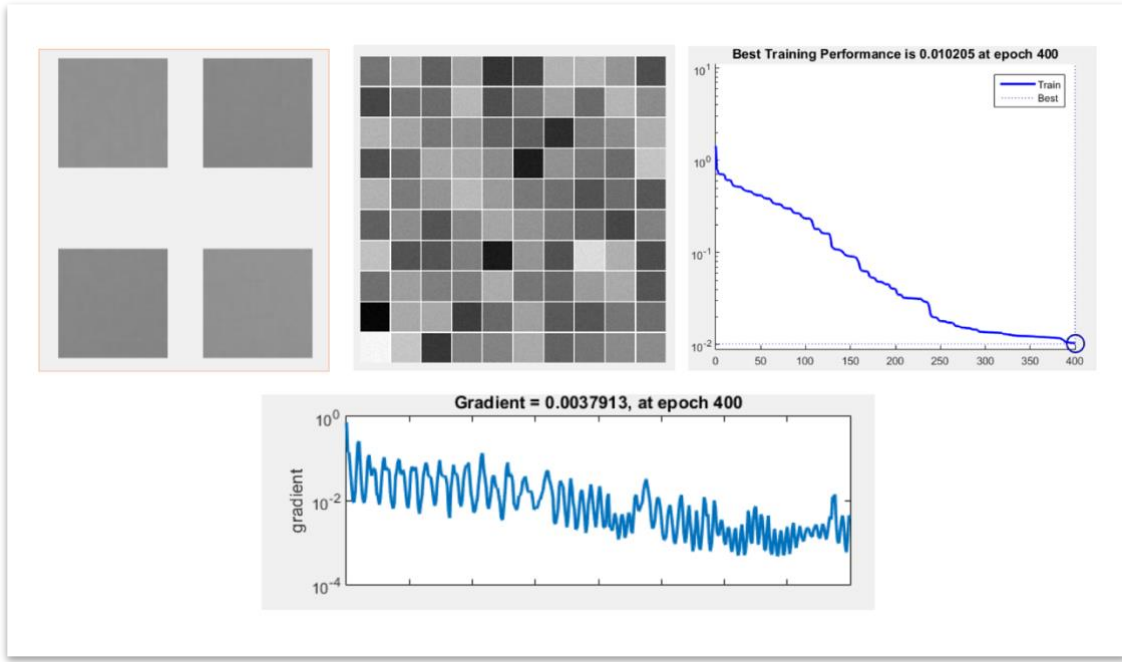


Şekil 5.8 Grup-5 “Çözünürlüğü 200- r ve yeniden boyutlandırma parametresi 0.4 olanlar”

Ayırt edici öznelilikler grup-6 sonucunda çıkmamıştır.



Şekil 5.9 Grup-6 “Çözünürlüğü 200- r ve yeniden boyutlandırma parametresi 0.8 olanlar”

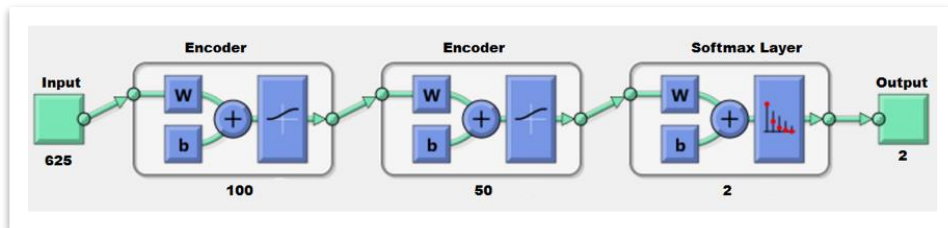


Şekil 5.10 Grup-7 “Çözünürlüğü 300-r ve yeniden boyutlandırma parametresi 0.2 olanlar”

Grup-7 ve incelenen diğer 300r ve 400r’lik gruplarda benzer sonuçlar çıktığı için sonuçlar burada gösterilmemiştir. Bu analiz sonucunda ise bazı kanılara varılmıştır.

- ✓ Optimum iterasyon sayısı 400 olabilir.
- ✓ En ayırt edici öznetelikler 0.2 ve 0.4 yeniden boyutlandırma ile çıkmıştır.
- ✓ 100x100 ve 200x200 görüntüler daha kabul edilebilir ve anlamlı sonuçlar vermiştir.
- ✓ Sonuçların içinde kabul edilebilir olanlardan en hızlı çalışanı 0.2 yeniden boyutlandırma parametresiyle 100x100 görüntülerden oluşan veriseti grubudur.
- ✓ 300 ve 400’lük görüntüler 25x25 değerlerine düşürüldüğünde çok fazla bilgi kaybına uğradığından anlamsız ve gürültülü sonuçlar üretmişlerdir.

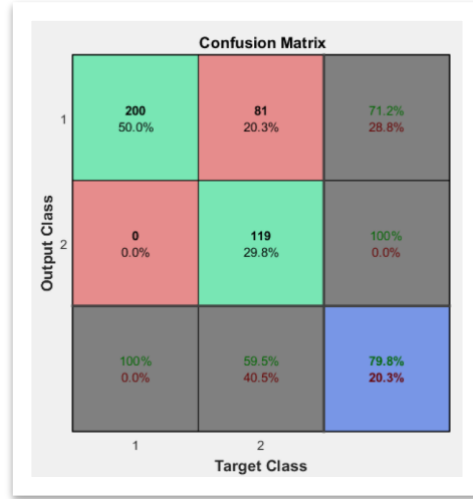
Bu bilgilerin ışığında veriseti oluşturulmuş ve etiketlenmiştir. Modele softmax katmanı da eklenmiş ve derin ağ modeli aşağıdaki son halini almıştır.



Şekil 5.11 Derin ağ modeli

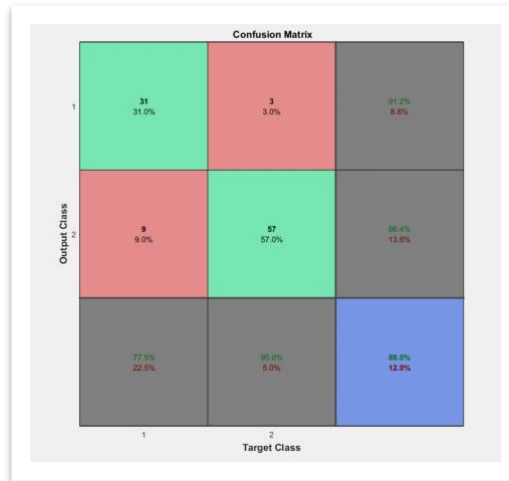
6. TEST SONUÇLARI

Yapılan testlerin sonuçları bu bölümde verilmiştir. Öncelikle ilk test sonuçları 200 hatalı 200 hatasız örnek ile yapılmıştır. Aşağıda bu sonuca ait confusion matrisi verilmiştir.



Şekil 6.1 Confusion matris

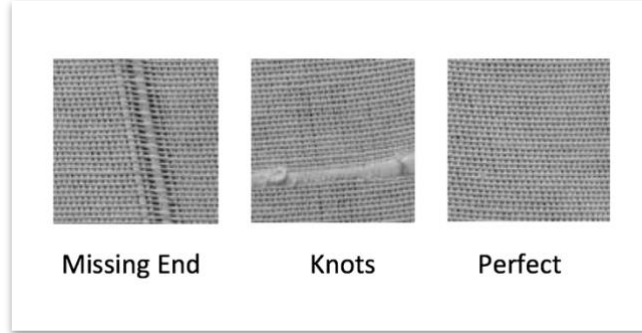
Bu sonuçlara bakıldığında çelişkili değerler olduğu görülmüştür. Hatalı kumaş örnekleri %100 bulunmuşken, hatasızlarda oran düşük çıkmıştır. Toplam doğruluk %79 olarak görülmüştür. Bu testin üzerine veriseti tekrar oluşturulmuş, eğitim için kullanılan görüntülerin sayısı 5000'e çıkarılmıştır (İlk testte eğitim için 2000 resim etiketlenerek kullanılmıştır.).



Şekil 6.2 Yeni confusion matrisi

Yeni yapılan test ardından kabul edilebilir bir seviyede %88 lik bir oran görülmüştür. Fakat yine de hata yapılmasının üzerine gidilmiş, ne tür hatalı kumaşların bulunamadığı tespit

edilmeye çalışılmıştır. Yapılan çalışmanın sonucu ise model kayıp uç (missing end) hatalarında başarısız olduğu görülmüştür. Bunun sebebi olarak ise yeniden boyutlandırılma işleminde bu hataların kaybolduğu düşünülmüştür.



Şekil 6.3 Kumaş Örnekleri

Bu tezi doğrulamak için, verisetinden bu tür hataya sahip örnekler çıkarılmış, yeniden model çalıştırılmıştır. Sonuç olarak beklenildiği gibi hata oranı oldukça düşmüştür.



Şekil 6.4 Final confusion matrisi (1 hata türü çıkarılmış test seti ile)

Bu işlemin ardından toplam doğruluk oranı %96 seviyelerine ulaşmıştır. Bu sonuç oldukça başarılı bir model üretildiğini göstermektedir.

7. KAZANIMLAR VE GELECEK ÇALIŞMALAR

Projenin kazanımlarından birisi son yıllarda sıkça kullanılan derin öğrenme konusuna giriş yapmış olmak, kumaş hatası tespitini ilk defa derin öğrenme ile uygulamak olmuştur. Bu bağlamda oto kodlayıcıların çalışmasını sağlayan optimum hiper parametreler öğrenilmiş ve başka akademik çalışmalara destek olacağı düşünülmüştür.

Hatalı / hatasız şeklinde iki sınıf üzerinde testler yapılmış, 25x25 çözünürlüğün oto kodlayıcının çalışmasında optimum olduğuna karar verilmiştir. Ayrıca yapılan denemeler sonucunda giriş / gizli katman oranının 1/5 -1/6 arasında olması daha ayırt edici öznelilikler çıkardığı görülmüştür. Hiper parametrelerden iterasyon sayısı 100-1000 arasında denenmiş, 400 iterasyonun eğitim için yeterli olduğu görülmüştür. Yeniden boyutlandırma parametresi üzerine yoğunlaştığımızda ise, bir resme yakından bakmanın daha iyi sonuçlar getireceği düşünülürken, kumaş üzerinde durum tam tersi olmuştur. Kumaşın kendine has dokusu sebebiyle resmi uzaktan ele almak, görüntünün daha düzenli bilgi içeriyor olması anlamına geldiğinden sonuçların başarısı diğer duruma kıyasla açık bir şekilde önde olduğu görülmüştür.

Bu proje boyunca katıldığımız aşağıda da belirtilen eğitim, seminer ve kongreler diğer önemli kazanımlardır.

- ✓ 28 Kasım 2015 ve 12 Aralık 2015 tarihlerinde ODTÜ’de düzenlenen derin öğrenme eğitimlerine katıldık. Türkiye’deki bu alanda ilk olan derin öğrenme grubuna dahil olarak bu doğrultuda başka akademik toplantı ve görüşmelerde bulunmaya devam ediyoruz.
- ✓ 1-3 Aralık 2015 tarihlerinde İstanbul Üniversitesi’nde düzenlenen INET-TR etkinliğinde Derin öğrenmeye giriş konferansına katıldık.
- ✓ 12 Ocak 2016 tarihinden İstanbul Teknik Üniversitesi UHEM bünyesinde düzenlenen Paralel Programlama Eğitimlerine katıldık.
- ✓ 16-19 Mayıs 2016 tarihlerinde Zonguldak’ta düzenlenen SIU 2016 kurultayına projemizin bir çıktısı olan bir bildiri ile katılıp, sunum yaptık [33].

Gelecek çalışmalar olarak ise, birincisi daha geniş bir veriseti üzerinde çalışarak, modelin doğruluğunu ispatlamak olacaktır. Bir diğer beklenti TILDA verisetiyle modeli test etmek, hatta mümkünse bu verisetinde kumaş hata türlerinin tespitlerini yapmak olacaktır. Ayrıca kazandığımız derin öğrenme deneyimi sayesinde, bu alanda çalışmalara devam edeceğiz.

Özellikle GPU programlama üzerine gidilerek, gömülü sistemler ile derin öğrenme algoritmalarını çalıştıracığımız projeler planlıyoruz.

PROJEDEN ÜRETİLEN AKADEMİK ÇALIŞMA

Abstract

Fabric defect detection have importance in terms of sectoral quality. Automatic systems are developed on the defect detection, in this regard many methods are tried to obtain high precision with image processing studies. In this study, deep learning which distinguishes with multi-layer architectures and reveals high achievement is applied to fabric defect detection. Autoencoder - a deep learning algorithm- that aimed to represent input data via compression or decompression is tried to detect defect of fabrics and it gains acceptable success. The vital goal of this study is to increase achievement of feature extraction by tuning up the autoencoder's input value and hyper parameters.

Published in:

[2016 24th Signal Processing and Communication Application Conference \(SIU\)](#)

Date of Conference : 16-19 May 2016

Page(s) :1437 - 1440 INSPEC Accession Number :16092853

Conference Location :Zonguldak DOI:[10.1109/SIU.2016.7496020](#) Publisher :IEEE

KAYNAKÇA

- [1] H. Y. T. Ngan, G. K. H. Pang and N. H. C. Yung, "Automated fabric defect detection-A review," *Image and Vision Computing*, vol. 29, pp. 442-258, 27 February 2011.
- [2] K. Kaur, N. Gupta and K. Adhikary, "An Automatic Method to Inspect Discontinuities in Textile," *IJCSET*, vol. 1, no. 8, pp. 496-498, September 2011.
- [3] H. Choudhary, "slideshare," March 2012. [Online]. Available: <http://www.slideshare.net/hiteshhobbit/fabric-defects-11884107>. [Accessed January 2016].
- [4] "Textiles Committee," [Online]. Available: <http://textilescommittee.nic.in/writereaddata/files/publication/gar5.pdf>.
- [5] V. R. Aitharaju and R. C. Averill, "Three-dimensional properties of woven-fabric composites," *Composites Science and Technology*, vol. 59, pp. 1901-1911, February 1999.
- [6] "www.cottoninc.com," Cotton Incorporated, [Online]. Available: <http://www.cottoninc.com/product/Tech-Assistance-Training/Classifications/>.

- [7] P. V. Mehta, "Inspection," in *Managing Quality in the Apparel Industry*, New Age International, 1998, pp. 18-19.
- [8] Ö. Kısaoğlu, "Kumaş Kalite Kontrol Sistemleri," *Pamukkale University Engineering College Journal of Engineering Sciences*, vol. 12, pp. 233-241, 2006.
- [9] A. Kumar, "Computer-Vision-Based Fabric Defect Detection: A Survey," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 348-362, January 2008.
- [10] H. Choudhary, "www.slideshare.net," March 2012. [Online]. Available: <http://www.slideshare.net/hiteshhobbit/fabric-defects-11884107>. [Accessed june 2015].
- [11] K. V. N. Kumar and U. S. Ragupathy, "An Intelligent Scheme for Fault Detection in Textile Web Materials," *International Journal of Computer Applications*, vol. 46, no. 10, pp. 23-29, May 2012.
- [12] J. L. Dorrity, G. Vachtsevanos and W. Jasper, "Real-Time Fabric Defect Defection&Control in Weaving Processes," *National Textile Center*, pp. 143-152, 1995.
- [13] "www.wikipedia.org," wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Confusion_matrix. [Accessed 11 July 2015].
- [14] H. Y. T. Ngan, G. K. H. Pang and N. H. C. Yung, "Motif-based defect detection for patterned fabric," *Pattern Recognition*, vol. 41, pp. 1878-1894, 2008.
- [15] Z. Kang, C. Yuan and Q. Yang, "The fabric defect detection technology based on wavelet transform and neural network convergence," in *International Conference on Information and Automation* , 2013.
- [16] Y. Bengio, *Learning Deep Architectures for AI*, Foundation and Trends in Machine Learning, 2009.
- [17] Y. Bengio, P. Vincent and A. Courville, *Representation Learning: A Review and New Perspectives*, 2012.
- [18] T. Dettmers, "Nvidia ParalelForAll Blog," Nvidia, 16 December 2015. [Online]. Available: <https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/>. [Accessed 20 March 2016].
- [19] K. Fukushima, "Neocognitron: A Self-organizing Neural Network Model For a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, vol. 36, pp. 193-202, 1980.
- [20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, *Handwritten Digit Recognition with a Back-Propagation Network*, New Jersey, 1989.
- [21] D. Crevier, *Ai: The Tumultuous History of the Search for Artificial Intelligence*, New York: NY:BasicBooks, 1993, p. 203.

- [22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504-507, 2006.
- [23] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation*, vol. 9, pp. 1735-1780, 1997.
- [24] D. Cireşan, U. Meier, J. Masci and J. Schmidhuber, "A Committee of Neural Networks for Traffic Sign Classification," in *Proceedings of International Joint Conference on Neural Networks*, California, 2011.
- [25] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. 1207.0580, 2012.
- [26] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *International conference on artificial intelligence and statistics*, 2009.
- [27] Y. Bengio, "Layer-Wise Training of Deep Belief Networks," in *Learning Deep Architectures for AI*, 2009, p. 60.
- [28] Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks," in *Advances in Neural Information Processing Systems 19 (NIPS'06)*, 2007.
- [29] "Deep Belief Networks," [Online]. Available: <http://deeplearning.net/tutorial/DBN.html>. [Accessed 27 April 2016].
- [30] "UFLDL Tutorial," [Online]. Available: <http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/>. [Accessed 28 April 2016].
- [31] G. E. Hinton, S. Osindero and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527-1554, 2006.
- [32] W. o. T. A. o. DFG, "TILDA Textile Texture-Database," University of Freiburg, [Online]. Available: <http://lmb.informatik.uni-freiburg.de/resources/datasets/tilda.en.html>. [Accessed 05 May 2016].
- [33] A. Şeker, K. A. Peker, A. G. Yüksek and E. Delibaş, "Fabric defect detection using deep learning," 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, 2016, pp. 1437-1440.

