



DERİN ÖĞRENME KULLANILARAK OPTİMUM JPEG KALİTE FAKTÖRÜNÜN BELİRLENMESİ

Emir ÖZTÜRK*, Altan MESUT

Trakya Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Edirne, Türkiye

Anahtar Kelimeler	Öz
<i>Veri Sıkıştırma, Derin Öğrenme, CNN, JPEG.</i>	Görüntü verisinde sıkıştırma algoritmalarının kullanılması bir gereklilik haline gelmiştir. En çok kullanılan görüntü sıkıştırma algoritmalarından biri olan JPEG, görüntü üzerinde kayıplı bir sıkıştırma gerçekleştirmekte ve verilen kalite faktörüne göre bu kayıp değişmektedir. Düşük kalite faktörlerinde dosya boyutu küçülmekte fakat bozulma gözle görülür hale gelmektedir. Yüksek kalite faktörlerinde ise kalite artmakta fakat dosya boyutundan edilen sıkıştırma karı azalmaktadır. Bu sebeple hem görüntü kalitesini korumak hem de yer kazancı sağlamak için dosya boyutu ve görüntü kalitesi arasındaki dengenin sağlanması faydalı olacaktır. Bu çalışmanın amacı, dosya boyutu ve görüntüdeki bozulmanın arasındaki oranın en iyi (optimum) olduğu kalite faktörünü derin öğrenme yöntemleri kullanarak belirlemektir. Yapılan çalışmada önerilen bir veri çıkarma yöntemi yoğun sinir ağları (dense neural networks) ile eğitilmiş ve yöntemin başarısı evrimsel sinir ağları ile yapılan denemelerle karşılaştırılmıştır. Görüntüdeki bozulmanın hesaplanmasında SSIM (Structural Similarity Index) kullanılmıştır. Elde edilen sonuçlarda önerilen yöntem kalite faktörünün belirlenmesinde CNN kullanılmasına göre %9.36 daha fazla doğruluk oranına sahip olmuştur.

FINDING THE OPTIMUM JPEG QUALITY FACTOR USING DEEP LEARNING

Keywords	Abstract
<i>Data Compression, Deep Learning, CNN, JPEG.</i>	It has become a necessity to use compression algorithms in image data. One of the most used image compression algorithms, JPEG performs lossy compression on an image and the loss varies according to the given quality factor. At low quality factor values, the file size gets smaller and artifacts become noticeable. In high quality factors, quality of image increases but savings from storage space decreases. For this reason, it will be beneficial to balance the file size and image quality in order to both maintain image quality and save space. The aim of this study is to determine the quality factor value, where the ratio between file size and image distortion is optimum, by using deep learning methods. In the study, a proposed data extraction method was trained with dense neural networks and the success of the method is compared with experimental results obtained with convolutional neural networks. Calculation of image distortion was carried out using SSIM (Structural Similarity Index). In the results obtained, the proposed method has 9.36% more accuracy than CNN in determining the quality factor.

Alıntı / Cite

Öztürk, E., Mesut, A., (2020). Derin Öğrenme Kullanılarak Optimum JPEG Kalite Faktörünün Belirlenmesi, Mühendislik Bilimleri ve Tasarım Dergisi, 8(4), 1010-1018.

Yazar Kimliği / Author ID (ORCID Number)

E. Öztürk, 0000-0002-3734-5171
A. Mesut, 0000-0002-1477-3093

Makale Süreci / Article Process

Başvuru Tarihi / Submission Date	04.03.2020
Revizyon Tarihi / Revision Date	16.10.2020
Kabul Tarihi / Accepted Date	10.11.2020
Yayın Tarihi / Published Date	25.12.2020

* İlgili yazar / Corresponding author: emirozturk@trakya.edu.tr, +90-284-226-1218-1101

1. Giriş (Introduction)

Yüksek çözünürlüklü kameraların ve cep telefonlarının yaygınlaşması ile üretilen ve internet üzerinde saklanan görüntü verisinin boyutu ve çözünürlüğü her geçen gün artmaktadır. Çözünürlüğün ve boyutun artması ile ihtiyaç duyulan kapasite de artış göstermektedir. Bu nedenle mevcut veriyi daha az yere ihtiyaç duyarak saklamak amacıyla sıkıştırma algoritmaları kullanılmaktadır. Görüntü verisinin sıkıştırılmasında JPEG (Wallace, 1992), JPEG2000 (Skodras vd., 2001), JPEG XR (Dufaux ve Sullivan, 2009), JPEG XT (Artusi vd., 2016) ve PNG (Randers-Pehrson ve Boutell, 1999) gibi kayıplı ve kayıpsız algoritmalar sıklıkla kullanılmaktadır.

Kayıpsız sıkıştırma algoritmaları verideki tekrarı azaltmak gibi yöntemler ile yer kazancı sağlarken, kayıplı sıkıştırma algoritmaları ise verinin belirli bir kısmını geri dönülmez bir şekilde eleyerek dosya boyutunu küçültmektedir. Bu sebeple kayıplı algoritmalar, kayıpsız sıkıştırma algoritmalarına göre daha fazla yer kazancı sağlamaktadır. Buna karşılık olarak sıkıştırılan görüntü üzerinde bozulmalar görülmektedir.

Kayıplı sıkıştırma algoritmalarının görüntü üzerinde sebep olduğu bozulmaların ölçülmesinde MSE (mean square error), PSNR (peak signal to noise ratio), SSIM (structural similarity index) (Z. Wang vd., 2004) gibi çeşitli yöntemler kullanılabilir.

Kayıplı görüntü sıkıştırma algoritmalarından olan JPEG, DCT kullanarak görüntüdeki yüksek frekans değerlerini çıkartır ve bu sayede görüntü üzerinde sıkıştırma sağlar. Elenecek yüksek frekans değerlerinin oranı ise kalite faktörü (QF) değeri ile belirlenmektedir. JPEG için bu değer 0 ile 100 arasında bir tamsayıyı ifade etmektedir. Yüksek kalite faktörlerinde elenen veri miktarı daha az olduğundan görüntü kalitesi daha yüksek ve sıkıştırmadan elde edilen kazanç daha düşük olmaktadır. Düşük kalite faktörlerinde ise dosya boyutu oldukça küçülmemekte fakat bozulma gözle görülür bir şekilde artmaktadır. Farklı karmaşıklıkta görüntü dosyalarında verilen kalite faktörüne göre elde edilen bozulma ve dosya boyutu da farklılık göstermektedir.

Farklı boyutta ve karmaşıklıkta görüntü dosyaları için bozulma ve dosya boyutu arasındaki denge farklı kalite faktörlerinde sağlanmaktadır. Düşük karmaşıklığa sahip görüntülerde kalite faktörünün düşürülmesi görüntü üzerinde daha az bozulmaya sebep olmakta ve dosya boyutunu daha fazla küçültebilmektedir. Bu sebeple bu tarz görüntüler için düşük kalite faktörleri daha uygun olmaktadır. Yüksek karmaşıklığa sahip görüntülerde ise belli bir değerden sonra kalite faktörünün artması ile dosya boyutu artarken bozulma oranında büyük değişimler görülmemektedir. Uygun kalite faktörünün seçilmesi, dosyanın bozulma ve boyut arasındaki optimum dengesini sağlamaktadır. Bu sayede görüntü kalitesinden fazla ödün vermeden elde edilebilecek en iyi sıkıştırma sağlanabilmektedir.

Elde edilen verilerin analiz edilebilmesi ve bir sonraki aşamada bu verilerden çıkarım ve tahmin yapılabilmesi için makine öğrenmesi algoritmaları kullanılmaktadır. Makine öğrenmesi algoritmalarının başarılı olduğu birçok farklı alan bulunmaktadır. Örneğin sosyal medya üzerinde gerçekleştirilen duygu analizi işleminde SVM (Support vector machine), KNN (K Nearest Neighbours), Random Forest, lojistik regresyon gibi algoritmaların başarısı ölçülmüş ve lojistik regresyon, %70 oranında başarı göstermiştir (Çelik vd., 2020). Bir diğer çalışmada ise kullanıcı taleplerine verilen cevap süresi tahmini SVM ile gerçekleştirilmiş ve %99 eğitim başarısı elde edilmiştir (Çevik ve Kayakuş, 2020).

Büyük veri setlerinin farklı uygulamalar için eğitilmesinde ise yapay sinir ağları ve derin öğrenme algoritmaları kullanılabilir. İnsan beyninin çalışma mekanizmasından esinlenerek ortaya atılmış bir yöntem olan yapay sinir ağlarının bir türü olan derin öğrenme algoritmalarının kullanılması görüntü, ses veya doğal dil işleme gibi alanlarda büyük başarı sağlamıştır (Masters ve Luschi, 2018). Özellikle 2000'li yıllardan sonra donanımdaki gelişmeler ve grafik işlemcilerinin kullanılabilmesi ile derin öğrenme algoritmalarının kullanımı artmış ve bugünkü seviyeye gelmiştir. Makine öğrenmesinde olduğu gibi derin öğrenme algoritmalarında da oldukça yüksek başarı sonuçları elde edilebilmektedir. Örneğin, (Baştürk vd., 2018) yaptıkları çalışmada derin öğrenme algoritmalarının parmak izi tanıma konusunda başarısının SVM, KNN veya Naive Bayes gibi yöntemlere göre daha yüksek olduğunu göstermiştir. Yine yapılan bir diğer çalışmada yapay sinir ağları kullanılarak gerçekleştirilen sistem tasarımında %85 başarı elde edilmiştir (Işık ve Işık, 2020).

Derin öğrenme algoritmalarının bir çeşidi olan evrimsel sinir ağları (CNN) görüntü verisi üzerinde oldukça başarılı sonuçlar elde etmektedir (Cengil ve Çınar, 2016). (Rende vd., 2016) yaptıkları çalışmada derin öğrenme algoritmalarında model seçimi için bir test otomasyonu geliştirmiş ve CNN kullanarak görüntüleri sınıflandırmıştır. Yapılan çalışmada %99'a varan başarı sonuçları elde edilmiştir.

Görüntülerin mevcut özelliklerine göre kalite faktörünün etkili bir şekilde belirlenmesi için veri seti boyutunun önemi büyüktür. Veri setinin boyutunun eğitim başarısına olumsuz etkileri yapılan çalışmalarda irdelenmiştir

(Pinto vd., 2008; Gonzalez, 2007). Çalışmalarda sunulan sonuçlarda veri setinin iki katına çıkarılması ile incelenen eğitim modellerinin birçoğunda başarı oranları %20'ye kadar artış göstermiştir.

Bu çalışmada görüntü kalitesinin veya sıkıştırma oranının önemine göre bir kalite faktörü seçimi amaçlanmıştır. Bu sayede kullanıcıların önem verdiği duruma göre optimum bir kalite faktörü belirlenmesi işlemi gerçekleştirilecektir. Kullanıcının kaliteye ağırlık vermesi durumunda dosya boyutu daha büyük olacak şekilde, daha kaliteli görüntülerin elde edilmesini sağlayacak yüksek kalite faktörlerinin seçimi gerçekleştirilecektir. Aksi takdirde görüntüdeki bozulmanın kabul edilebilir düzeyde olduğu seçilebilecek en düşük kalite faktörünün seçilmesi ve dosya boyutunun olabildiğinde küçük tutulması amaçlanmaktadır.

İstenilen modelin oluşturulması için çalışmada "The Open Images Dataset v4" (Kuznetsova vd., 2018) derleminin 110 GB boyutunda bir kısmı üzerinde 10-100 arasında olmak üzere 10 farklı kalite faktöründe dosya boyutları ve bozulma oranları elde edilmiştir. Kalite faktöründeki küçük farklar sonuçta değişiklik göstermemektedir. Bu sebeple kalite faktörleri arasındaki aralık 10 olarak belirlenmiştir. Daha sonra görüntüler bu 10 farklı sınıfta etiketlenmiştir. Elde edilen bu etiketli veri CNN ve makalede önerilen bir yöntem kullanılarak eğitilmiş ve kalite faktörü tahminindeki başarıları karşılaştırılmıştır

Çalışmanın ikinci bölümünde CNN mimarisi açıklanmış ve kullanılan model verilmiş, üçüncü bölümde önerilen yöntem sunulmuştur. Dördüncü bölümde sınıflandırma verisinin elde edilmesinde kullanılan yöntem açıklanmıştır. Beşinci bölümde elde edilen test sonuçları incelenmiş ve son bölümde değerlendirme gerçekleştirilmiştir.

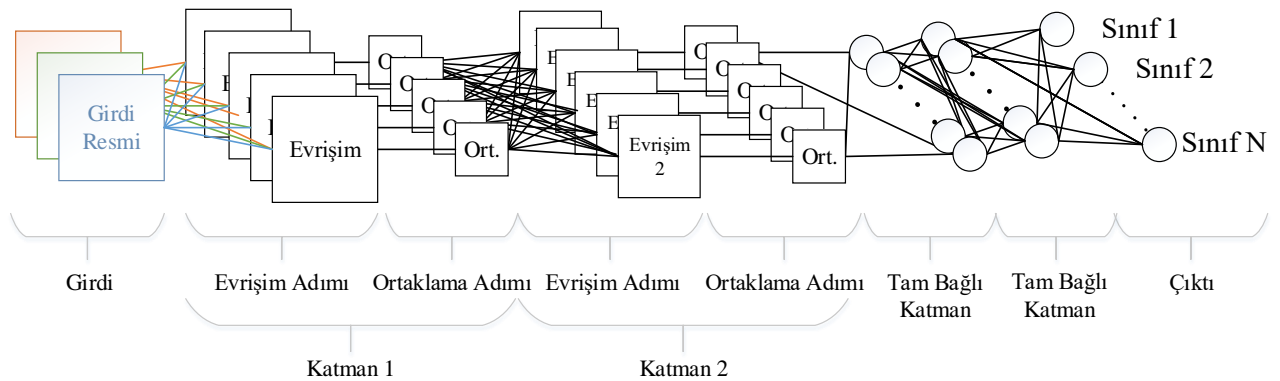
2. Evrişimsel Sinir Ağları (Convolutional Neural Networks)

Evrişimsel sinir ağları (Convolutional neural networks - CNN) çok boyutlu büyük verileri işlemek için kullanılan bir yapay sinir ağı türüdür (Ozcan ve Basturk, 2019). CNN'ler en az bir katmanda evrişim işlemi gerçekleştirmektedirler ve evrişim, havuzlama (pooling) ve tam bağlı (fully-connected) olmak üzere üç temel katman içerirler (Zhang vd., 2019).

Evrişim katmanı birden fazla farklı evrişim çekirdeğinden oluşur. Bu çekirdekler görüntü üzerinde farklı özelliklerin çıkarılması için kullanılırlar. Daha sonra aktivasyon fonksiyonu kullanılarak non-lineerlik sağlanır. Bilinen aktivasyon fonksiyonlarına sigmoid, tanh (LeCun vd., 2012), ve RELU (Nair ve Hinton, 2010) örnek verilebilir. Aktivasyon fonksiyonu olarak RELU kullanılması yapay sinir ağlarının daha hızlı öğrenmesini ve daha yüksek sınıflandırma başarıları elde etmesini sağlamaktadır.

Havuzlama katmanı ile katmana girdi olarak verilen verinin çözünürlüğü düşürülerek model maliyeti azaltılmaktadır. Bilinen havuzlama işlemlerine örnek olarak ortalama (average pooling) (T. Wang vd., 2012) ve maksimum havuzlama (max pooling) (Boureau vd., 2010) verilebilir.

Evrişim ve havuzlama katmanlarından sonra bir veya daha fazla tam bağlı katman bulunur. Tam bağlı katmanlarda önceki katmandaki tüm nöronlar mevcut katmandaki tüm nöronlara bağlıdır. Bu sayede mevcut katmanda tüm nöronların bilgisi bulunmaktadır. Tam bağlı katmandan sonra ise son katman olan çıkış katmanı bulunmaktadır. Çıkış katmanında ikiden fazla sınıf etiketi olduğunda sınıflandırma yapmak amacıyla softmax operatörü sıklıkla kullanılmaktadır (Russakovsky vd., 2015). Çıkış katmanının boyutu sınıf sayısı ile aynıdır. Örnek bir CNN mimarisi Şekil 1'de gösterilmiştir.



Şekil 1. Örnek bir evrişimsel sinir ağı mimarisi (An example CNN architecture)

Şekil 1’de ortalama havuzlama kullanılan bir evrişimsel sinir ağı mimarisi görülmektedir. Girdi resmi evrişim adımına verildikten sonra çıkartılan özellikler ortalama havuzlama işlemine tabi tutularak veri boyutunun küçültülmesi sağlanmaktadır. Daha sonra elde edilen sonuç verisi ikinci katmandaki evrişim işleminin girdisi olarak kullanılmaktadır. Evrişim ve havuzlama katmanları tamamlandığında son olarak tam bağlı katman ve çıktı katmanları yer almaktadır.

Bu çalışmada deneysel sonuçların elde edilmesinde kullanılan evrişimsel mimari Tablo 1’de verilmiştir. Tablo 1’deki mimariyi kullanabilmek adına görüntüler öncelikle 512×512 piksel olarak yeniden boyutlandırılmış ve daha sonra gri formata çevrilmiştir.

Tablo 1. Çalışmada kullanılan evrişimsel sinir ağı mimarisi (CNN architecture used in experiments)

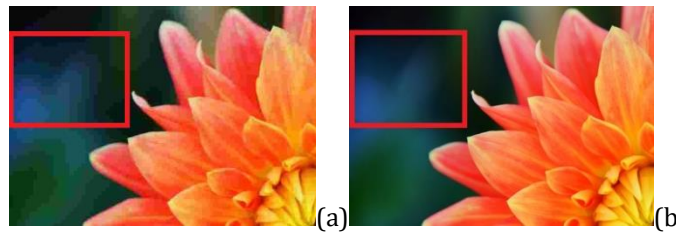
Katman	Çıktı boyutu
Girdi	(512×512×1)
Evrişim	(510×510×32)
Havuzlama	(170×170×32)
Evrişim	(168×168×64)
Havuzlama	(56×56×64)
İndirgeme	200704
Yoğun	32
Seyreltme (0.4)	-
Yoğun	32
Seyreltme (0.3)	-
Yoğun (Çıktı)	10

Tablo 1’de görüldüğü gibi evrişim işleminde seçilen 3×3 boyutundaki kayan pencere sebebiyle bu boyut 510×510’a düşmüştür. İlk katmanda 32 çekirdek kullanılmıştır. Havuzlama işleminde 3×3’lük boyuttaki bir blok tek piksele indirgendigi için boyut 170×170×32 olmuştur.

İndirgeme (flatten), eldeki verinin tek boyutlu bir veri haline getirilmesi işlemi üstlenmektedir. Yoğun katmanlar ise indirgeme katmanına doğrudan bağlıdır. Seyreltme (dropout) düğümleri verilerin eğitilmesinde aşırı öğrenmeyi (overfit) engellemek için verilen oranda nöronları görmezden gelme işlemi gerçekleştirmektedir.

3. Önerilen Yöntem (Proposed Method)

JPEG, sıkıştırma aşamasında resmi 8×8 bloklara bölmekte ve bu blokların üzerinde belirli bir kayıp oluşturan niceleme (Quantization) işlemini gerçekleştirmektedir. Çok yüksek kayıplı sıkıştırma yapıldığında bu bloklardaki veri kaybı sebebiyle görüntülerde bloklama etkisi (blocking effect) gözükmemektedir (Şekil 2). Bloklama etkisi renk geçişleri fazla olan ve detaylı görüntülerde düz bir alana sahip ya da tek renkten oluşan bir resme göre daha fazla gözükmemektedir. Bu sebeple daha az görsel karmaşıklığa sahip görüntülerde düşük kalite faktörlerinde bile fazla bozulma görülmezken yüksek karmaşıklıkta görüntülerde kalite faktörünün daha yüksek tutulması gerekmektedir.



Şekil 2. Kf 10 (a) ve 100 (b) için bloklama etkisi (Blocking effect for QF10 (a) and QF100 (b))

Evrişimsel sinir ağları, kullandığı çekirdekler ile özellik çıkarımı işlemi gerçekleştirirken, önerilen yöntem bu karmaşıklığın görüntüden bir özellik vektörünün çıkartılması ile tespit edilebilmesini amaçlamaktadır. Bu sayede eğitim sırasında evrişimsel mimariye göre işlenmesi gereken veri miktarı ve eğitim sırasında yapılan işlem sayısı azaltılmış olacaktır.

Birbirine yakın renk değerlerine sahip bloklarda bloklama etkisi renk benzerliği sebebiyle daha az gözükcektir. Bloklar arası renk değerleri birbirlerinden farklıysa bloklar daha fazla görünür olacaktır. Bu sebeple önerilen yöntemde bloklama etkisinin görünürlüğünün tespiti için blokların kendi içlerindeki renk çeşitliliğinin ve bloklar arası renk farkının belirlenmesi amaçlanmıştır. Blokların kendi içindeki renk çeşitliliğinin tespiti için blok içi

varyans hesaplanmıştır. Bloklar arası renk farkının oranının tespiti için ise blokların ortalama renk değerleri elde edilmiştir. Elde edilen bu vektörler birleştirilerek önerilen yöntemin eğitim vektörü olarak kullanılmıştır.

Vektörün elde edilmesinde görüntü öncelikle gri tabanlı (grayscale) formata dönüştürülüp $n \times n$ boyutunda bloklara bölündükten sonra B blok sayısı, $p_{b_{ij}}$ b bloğunun (i, j) konumundaki piksel ve μ_b blok için piksel değerlerinin ortalaması olmak üzere görüntü üzerindeki her blok için varyans (V) değeri hesaplanmaktadır.

$$V_b = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (p_{b_{ij}} - \mu_b)^2, b = 1, 2, \dots, B \quad (1)$$

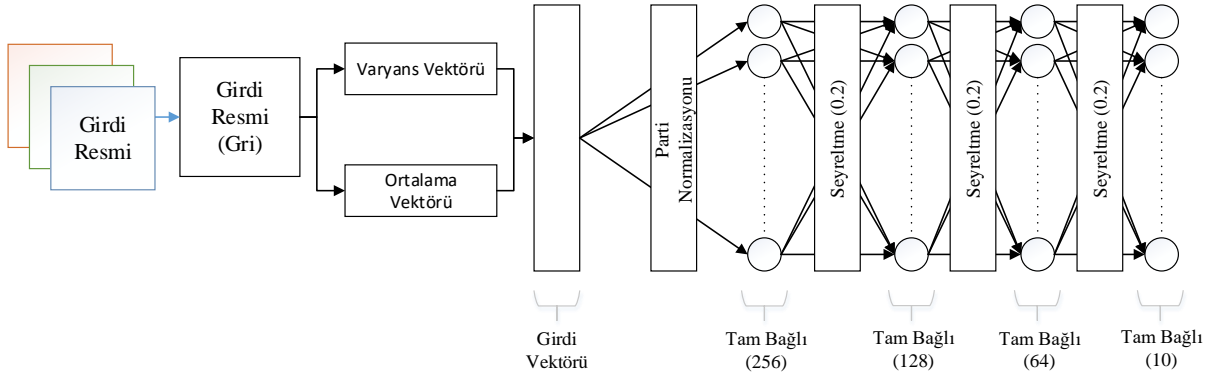
Bu işlemin amacı resmin bölünmüş her bir blok içerisindeki renk karmaşıklığını tespit etmektir. Daha sonra bloklar arası geçişte renk değişimlerinin değerini takip etmek amacı ile öncelikle her bloğun ortalama renk değeri (μ_{b-1}) hesaplanıp kendisinden sonra gelen bloğun ortalama renk değeri (μ_b) ile farkı (F) elde edilmektedir.

$$F_{b-1} = \mu_b - \mu_{b-1}, b = 2, 3, \dots, B \quad (2)$$

Elde edilen bu varyans (V) ve fark (F) değerleri özellik vektörü (O) olarak kullanılmaktadır.

$$O_{Resim} = [V_1 V_2 \dots V_B F_1 F_2 \dots F_{B-1}] \quad (3)$$

Bu özellik vektörlerinin eğitilmesi için tam bağlı katmanlardan oluşan bir sinir ağı kullanılmaktadır. Sinir ağının çıktı katmanı sınıf sayısı kadar düğüm içermektedir. Kullanılan yapı Şekil 3'te gösterilmiştir.



Şekil 3. Önerilen yöntem (Proposed method)

Girdi vektörü elde edildikten sonra parti normalizasyonu (batch normalization) işlemi gerçekleştirilmektedir. Bu işlemin çıktısı 256 düğümlü bir tam bağlı katmana verilmektedir. Tam bağlı toplam 3 katmandan sonra 10 sınıf için çıkış katmanı bulunmaktadır.

4. Sınıflandırma Verisinin Elde Edilmesi (Obtaining Classification Data)

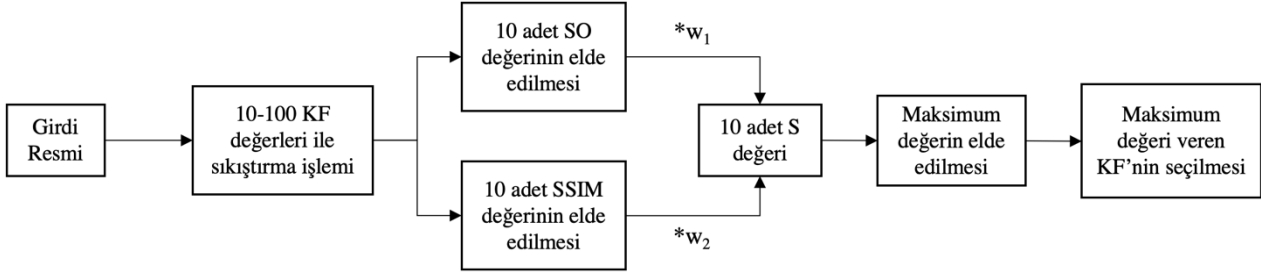
Sinir ağlarının eğitilmesi için görüntü verisinden elde edilmiş etiketli verilere ihtiyaç duyulmaktadır. Çalışmada görüntü verisinden kalite faktörü belirlenmesi işlemi gerçekleştirileceğinden etiket olarak dosya boyutu ve SSIM değeri parametrelerinin optimum değerine karşılık gelen kalite faktörünün elde edilmesi gerekmektedir. Mevcut görüntü derleminin içindeki her görüntüde 10 ile 100 arasındaki 10'un katındaki tüm kalite faktörü değerleri için sıkıştırılmış dosya boyutu ve SSIM değerleri elde edilmiştir. SSIM 0-1 arası değerler almaktadır. SSIM değerinin yüksek olması bozulmanın daha az olması anlamına gelmektedir. Daha sonra sıkıştırılmış dosya boyutu ve orijinal dosya boyutu değerleri kullanılarak Eşitlik 4'teki işlem gerçekleştirilmiştir ve 0-1 arasında bir Sıkıştırma oranı (SO) elde edilmiştir.

$$SO = \frac{\text{Sıkıştırılmış dosya boyutu}}{\text{Orijinal dosya boyutu}} \quad (4)$$

SSIM ve Eşitlik 4'ten elde edilen değerler dosya boyutu ve bozulma için 0-1 aralığında değerler vermektedir. Dosya boyutunun minimum ve SSIM değerinin maksimum olması istendiği için bu değerlerin ağırlıklı farkları alınarak bir skor (S) değeri hesaplanmaktadır.

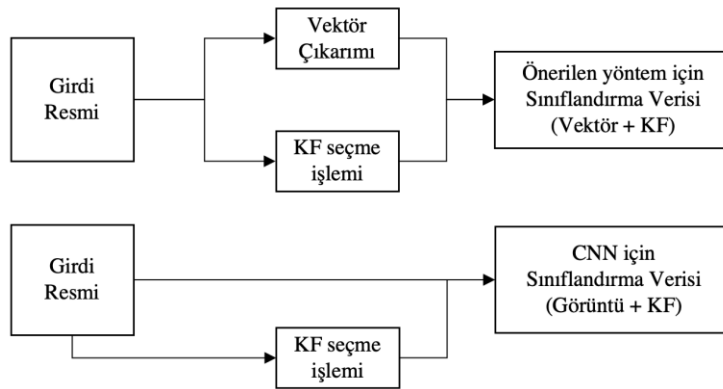
$$S = w_1 * SSIM - w_2 * SO \quad (5)$$

Burada ağırlık değerlerinin istenilen şekilde seçilmesi ile dosya boyutunun veya görüntü kalitesinin önemi belirlenebilmektedir. Görüntü üzerindeki kalitenin korunması istendiğinde w_1 değeri w_2 değerine göre daha yüksek belirlenebilmektedir. Aynı şekilde sıkıştırma miktarı önemli fakat kalite önemsiz ise w_2 değeri w_1 değerine göre yüksek seçilebilecektir. İki ağırlığın eşit seçilmesi durumunda dengeli bir kalite faktörü belirlenecektir.



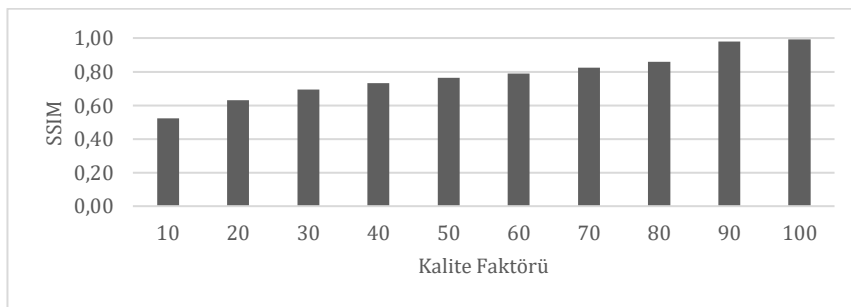
Şekil 4. Kalite faktörünün belirlenmesi (Selection of quality factor)

Şekil 4'te sınıflandırma verisinin elde edilmesi için ihtiyaç duyulan kalite faktörü değerinin belirlenmesi işlemi gösterilmiştir. Girdi resmi sıkıştırılıp aynı anda dosya boyutu eldesi ve kalite ölçümü gerçekleştirilebilmektedir. Daha sonra bu değerler istenilen ağırlık değerleri ile çarpılarak 10 farklı kalite faktörü için S değerleri elde edilmektedir. Son aşamada ise maksimum S değerine sahip kalite faktörü o görüntü için etiket verisi olarak seçilmektedir.



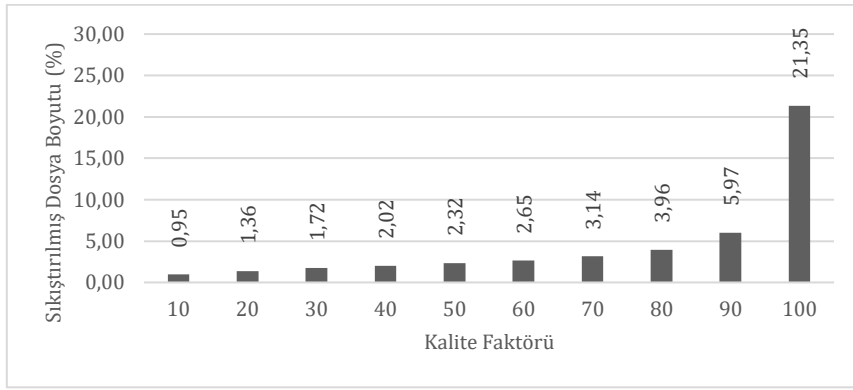
Şekil 5. Sınıflandırma verisinin elde edilme aşamaları (Stages of obtaining classification data)

CNN ve önerilen yöntem için veri seti ayrı ayrı hazırlanmaktadır (Şekil 5). Tüm görüntü dosyaları için kalite faktörleri belirlendikten sonra CNN veri seti girdi görüntüsü ve kalite faktörü ikilisi olarak oluşturulmaktadır. Önerilen yöntemde ise görüntüden üçüncü bölümde belirtildiği gibi bir vektör çıkarımı gerçekleştirilmektedir. Bunun ardından CNN'de olduğu gibi elde edilen vektör ve kalite faktörü ikilileri eğitim verisi olarak belirlenmektedir.



Şekil 6. Kalite faktörüne göre SSIM değerinin değişimi (Change of SSIM value according to the quality factor)

Kalite faktörünün 10-100 arasındaki değişimindeki bozulma lineer değildir. Bu sebeple belirli bir değer altında bozulmadaki artış kalite faktöründeki azalmadan daha fazla olmaktadır (Şekil 6).



Şekil 7. Kalite faktörüne göre sıkıştırılmış dosya boyutunun değişimi (Change of compressed file size according to quality factor)

Şekil 7'deki grafikte de görüldüğü gibi bir dosya için kalite faktörü ile sıkıştırma oranı da doğru orantıya sahip değildir. Değişimdeki bu non-lineerlik sebebiyle optimum kalite faktörü farklı ağırlıklar için farklı aralıklarda elde edilmektedir. Bu sebeple deneysel sonuçlarda farklı ağırlıklara göre yapılan sınıflandırmada farklı kalite faktörü değerleri bulunmaktadır.

5. Deneysel Sonuçlar (Experimental Results)

Yapılan çalışmada "The Open Images Dataset v4" derleminden 389655 adet görüntü üzerinde bir önceki bölümde kullanılan sınıflandırma yöntemi ile etiketleme işlemi gerçekleştirilmiştir. Daha sonra etiketlenen bu veriler üzerinde Tablo 1'deki evrimsel sinir ağı ve Şekil 3'te verilen önerilen sinir ağı eğitilmiş ve başarımları elde edilmiştir. Tüm ağların eğitiminde görüntüler 512×512 boyutuna indirgenmiş, gri formatta kullanılmıştır. Eğitim 20 adım (epoch) ve 32 parti (batch) büyüklüğü parametresi seçilerek gerçekleştirilmiştir. Ayrıca eğitimde aşırı öğrenmenin tespiti amacıyla veri seti %80-%20 oranında bölünmüş ve %20'lik kısmı geçerleme (validation) için kullanılmıştır. Sınıflandırma için w_1 (SSIM) ve w_2 (sıkıştırma oranı) ağırlıkları sırasıyla %30-%70, %50-%50 ve %70-%30 olarak belirlenmiştir. Bu oranların seçiminde veri setinin dağılımı göz önüne alınmıştır. Herhangi bir ağırlığın %30'un altına düşürülmesi durumunda veri setinde en iyi sonucu veren kalite faktörü değeri tek bir noktada toplanmaktadır. Örneğin w_1 ve w_2 değerleri sırasıyla %20 ve %80 olarak belirlendiğinde örneklerin 387546 tanesi (%99'u) 100 kalite faktörü sınıfına dahil olmaktadır. Benzer şekilde w_1 ve w_2 değerlerinin %80 ve %20 seçilmesi durumunda örneklerin tamamı 10 kalite faktöründe toplanmaktadır. Bu durumda elde edilen veri seti eğitim için uygun olmamaktadır. Eğitim verisi dağılımının en iyi olduğu oranlar seçilmiş ve bu oranlar için her sınıfa düşen eğitim verisi miktarı Tablo 2'de verilmiştir.

Tablo 2. Farklı oranlar için her sınıfa karşılık gelen eğitim örneği sayısı (Training sample count for each class for different ratios)

Oran(w_1 - w_2)	Sınıflar									
	10	20	30	40	50	60	70	80	90	100
30-70	427	283	1123	3494	7934	16192	55219	94560	191732	18691
50-50	226	127	170	140	826	1273	16180	41487	208547	120679
70-30	190	103	128	80	596	596	6991	20202	116489	244280

Beklendiği üzere 10-60 arası kalite faktöründe olmasına rağmen S değeri yüksek olan görüntü sayısı diğer sınıflara göre çok daha azdır. Genellikle görüntüler 80, 90 ve 100 kalite faktörlerinde optimum oranı elde etmiştir.

Önerilen yöntemde blok boyutunun 64×64 seçildiği durumda farklı oranlardaki başarımları Tablo 3'te verilmiştir.

Tablo 3. Önerilen yöntemin farklı oranlar için sonuçları (Results of the proposed method for different ratios)

Oran(w_1 - w_2)	Eğitim		Geçerleme	
	Kayıp	Doğruluk	Kayıp	Doğruluk
30-70	1.1171	0.5461	1.1076	0.5466
50-50	0.8433	0.6451	0.8457	0.6463
70-30	0.6379	0.7185	0.6533	0.7105

Tablo 3'ten görüldüğü üzere en yüksek doğruluk SSIM değerinin yüzdeliğinin %70 oranında seçilmesi (70-30) ile elde edilmiştir. Önerilen yöntemde blok büyüklüğünün etkisinin incelenmesi amacı ile en iyi sonuçların elde

edildiği 70-30 oranındaki sınıflandırma üzerinde 64×64, 32×32 ve JPEG'in blok boyutu olan 8×8 boyutları seçilmiş ve başarı sonuçları Tablo 4'te verilmiştir.

Tablo 4. 70-30 oranı için önerilen yöntemin farklı blok boyutlarında elde edilen sonuçlar (Results obtained for different block sizes of the proposed method for 70-30 ratio)

Blok Boyutu	Eğitim		Geçerleme	
	Kayıp	Doğruluk	Kayıp	Doğruluk
64×64	0.6379	0.7185	1.2381	0.7105
32×32	0.6019	0.7440	0.6533	0.7284
8×8	0.5034	0.8024	0.6298	0.7372

Tablo 4'te görüldüğü gibi en yüksek doğruluk 8×8 bloklardan elde edilmektedir. Bunun sebebi blok boyutunun küçülmesi ile resmin karmaşıklığı ile ilgili elde edilen bilginin artması ve sinir ağının daha fazla bilgi sahibi olabilmesidir. Önerilen yöntemin en iyi sonucunun elde edildiği 8×8 blokların evrişimsel sinir ağı ile elde edilen sonuçlar ile karşılaştırılması Tablo 5'te verilmiştir.

Tablo 5. Evrişimsel sinir ağı ile önerilen yöntemin karşılaştırılması (Comparison of the convolutional neural network and the proposed method)

Yöntem	Oran	Eğitim		Geçerleme	
		Kayıp	Doğruluk	Kayıp	Doğruluk
Evrişimsel	30-70	1.1888	0.5210	1.2921	0.5497
	50-50	0.9845	0.6064	0.9707	0.5491
	70-30	0.7858	0.7088	0.6906	0.7146
Önerilen Yöntem (8×8)	70-30	0.5034	0.8024	0.6298	0.7372

Tablo 5'ten görüldüğü üzere önerilen yöntem 8×8 blok boyutunda SSIM değerine %70 ağırlık verildiğinde evrişimsel sinir ağının tüm sonuçlarından daha iyi sonuçlar elde etmiştir. Ayrıca evrişimsel sinir ağında da en yüksek sınıflandırma başarısı SSIM değerinin %70 ağırlıklı olduğu durumda elde edilmiştir.

CNN'de görüntülerin tümünün kullanılmasına karşılık önerilen yöntemde vektörlerin çıkartılması eğitim uzayını küçültmektedir. 512×512 bir görüntü dosyası 786432 bayt (512×512×3) yer kaplarken, bu görüntüden çıkartılan bir vektör en küçük blok boyutu olan 8×8 boyutunda seçildiğinde 8191 bayt (4096 + 4095) yer kaplamaktadır. Böylece önerilen yöntemde eğitim işlemi daha az kaynak ihtiyacı duymaktadır.

Yöntemlerin başarısının ölçülmesinde sınıflandırma nominal olarak gerçekleştirildiği için yöntemin 80 kalite faktörü yerine yakın olan 70 veya 90 gibi bir kalite faktörüne göre karar vermesi durumunda bile yanlış sınıflandırma olarak kabul edilecek ve başarıyı düşürecektir. Buna rağmen önerilen algoritma %80 oranında bir eğitim başarısı ve %73 oranında bir geçerleme başarısı elde etmiştir.

5. Sonuç ve Tartışma (Result and Discussion)

Görüntü verilerinin sıkıştırılması kadar kaliteli saklanması da önemlidir. Sıkıştırmayı kaliteyi bozmadan sağlayabilmek adına optimum kalite faktörünün seçilmesi hem görüntüdeki bozulmayı minimize edecek hem de kaliteden belirli bir seviyeye kadar ödün vermeden yer kazancı sağlanacaktır.

Bu çalışmada önerilen yöntem ile resmin karmaşıklığının tespitinin yapılması amaçlanmıştır. Karmaşıklığın doğru bir şekilde tahmin edilmesi sınıflandırma başarısı açısından önem arz etmektedir. Yöntem blokların piksel çeşitliliği ve bloklar arası farkı tespit etmeyi amaçlamaktadır.

Görüntü verilerinde özellik çıkarımı için oldukça fazla kullanılan evrişimsel sinir ağları görüntüdeki özellikleri elde edip buna göre sınıflandırma gerçekleştirebilmektedir. Görüntü üzerinde çeşitli maskelerin gezdirilmesi ile elde edilen özellikler üzerinden eğitim gerçekleştiren bu ağlar sıkıştırma için yapılan sınıflandırmada karmaşıklığı bulma konusunda önerilen yöntem kadar başarılı sonuçlar göstermemiştir.

Çalışmada önerdiğimiz yöntem ile çıkarılan vektörler sayesinde görüntü verisinin evrişimsel ağlara verilmesine göre çok daha küçük bir veri uzayında çalışabilmekte ve bu sayede yöntemin karmaşıklığı evrişimsel sinir ağlarına göre daha düşük olmaktadır. Böylece modelin kullanılmasında ihtiyaç duyulan kaynaklar da azaltılabilmektedir.

Önerilen yöntem sayesinde yöntemin kullanılacağı ortamın ihtiyacına göre belirlenecek ağırlıklarla kalite faktörü seçimi yapılmaktadır. Veri boyutunun önemli olduğu durumlarda yöntem düşük kalite ve yüksek sıkıştırma

oranları için düşük kalite faktörlerini seçmektedir. Aksi takdirde algoritma görüntünün kalitesini belirli bir seviyenin üstünde tutabilecek optimum kalite faktörünü seçiyor olacaktır.

İleride yapılacak çalışmalarda blok içi ve bloklar arası farklı metrikler denenip başarı oranına etkisinin elde edilmesi planlanmaktadır. Ayrıca veri çeşitliliğinin artırılması ve veriyi açıklayabilecek daha iyi metriklerin bulunması ile mevcut modelin daha fazla genelleştirilmesi ve başarı oranının artırılması amaçlanmaktadır.

Çıkar Çatışması (Conflict of Interest)

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir. No conflict of interest was declared by the authors.

Kaynaklar (References)

- Artusi, A., Mantiuk, R.K., Richter, T., Korshunov, P., Hanhart, P., Ebrahimi, T., Agostinelli, M., 2016. JPEG XT: A Compression Standard for HDR and WCG Images. *IEEE Signal Processing Magazine*, 33 (2), 118–24.
- Baştürk, A., Baştürk, N.B., Qurbanov, O., 2018. Parmak İzi Tanıma İçin Farklı Sınıflandırıcıların Karşılaştırmalı Başarım Analizi. *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 7 (2), 504–13.
- Boureau, Y.L., Ponce, J., Lecun, Y., 2010. A Theoretical Analysis of Feature Pooling in Visual Recognition. *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*.
- Çelik, Ö., Osmanoğlu, U. Ö., & Çanakçı, B., 2020. Sosyal Medya Yorumlarından Duygu Analizi. *Mühendislik Bilimleri ve Tasarım Dergisi*, 8(2), 366–374.
- Cengil, E., Çınar, A., 2016. A New Approach for Image Classification: Convolutional Neural Network. *European Journal of Technic EJT*, 6 (2), 96–103.
- Çevik, K. K., Kayakuş, M., 2020. Bilişim Teknolojileri Departmanında Kullanıcıların Taleplerine Cevap Verme Süresinin Makine Öğrenmesi İle Tahmin Edilmesi. *Mühendislik Bilimleri ve Tasarım Dergisi*, 8(3), 728–739.
- Dufaux, F., Sullivan, G.J., 2009. The JPEG XR Image Coding Standard. *IEEE Signal Processing Magazine*.
- Gonzalez, T.F., 2007. *Handbook of Approximation Algorithms and Metaheuristics*. *Handbook of Approximation Algorithms and Metaheuristics*, 1–1432.
- Işık, N., Işık, A. H., 2020. Yapay Sinir Ağları İle Elektrostatik Lens Sistem Tasarımı. *Mühendislik Bilimleri ve Tasarım Dergisi*, 8(2), 388–396.
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., et al., 2018. The Open Images Dataset v4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *ArXiv Preprint ArXiv:1811.00982*.
- LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.R., 2012. *Efficient Backprop*. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- Masters, D., Luschi, C., 2018. Revisiting Small Batch Training for Deep Neural Networks, 1–18.
- Nair, V., Hinton, G.E., 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*.
- Özcan, T., Baştürk, A., 2019. Lip Reading Using Convolutional Neural Networks with and without Pre-Trained Models. *Balkan Journal of Electrical and Computer Engineering*, 7 (2), 195–201.
- Pinto, N., Cox, D.D., DiCarlo, J.J., 2008. Why Is Real-World Visual Object Recognition Hard? *PLoS Computational Biology*, 4 (1), 0151–56.
- Randers-Pehrson, G., Boutell, T., 1999. PNG (Portable Network Graphics) Specification, Version 1.2. PNG Development Group.
- Rende, F.Ş., Bütün, G., Karahan, Ş., 2016. Derin Öğrenme Algoritmalarında Model Testleri: Derin Testler. *CEUR Workshop Proceedings*, 172154–59.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., et al., 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*.
- Skodras, A., Christopoulos, C., Ebrahimi, T., 2001. The JPEG 2000 Still Image Compression Standard. *IEEE Signal Processing Magazine*.
- Wallace, G.K., 1992. The JPEG Still Picture Compression Standard. *IEEE Transactions on Consumer Electronics*.
- Wang, T., Wu, D.J., Coates, A., Ng, A.Y., 2012. End-to-End Text Recognition with Convolutional Neural Networks. In *Proceedings - International Conference on Pattern Recognition*.
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*.
- Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., Yu, B., 2019. Recent Advances in Convolutional Neural Network Acceleration. *Neurocomputing*, 32337–51.