



HARRAN ÜNİVERSİTESİ MÜHENDİSLİK DERGİSİ

HARRAN UNIVERSITY JOURNAL of ENGINEERING

e-ISSN: 2528-8733 (ONLINE)

URL: <http://dergipark.gov.tr/humder>

Eğik Karakter Tanıma Başarısını Arttırmak için Yeni Bir Yöntemin Kullanılması

Using a New Method to Improve Italic Character Recognition Success

Yazar(lar) (Author(s)): Ahmet ÇELİK¹

¹ ORCID ID: 0000-0002-6288-3182

Bu makaleye şu şekilde atıfta bulunabilirsiniz (To cite to this article): Çelik A., “Eğik Karakter Tanıma Başarısını Arttırmak için Yeni Bir Yöntemin Kullanılması”, *Harran Üniversitesi Mühendislik Dergisi*, 6(1): 01-11, (2021).

Erişim linki (To link to this article): <http://dergipark.gov.tr/humder/archive>



Eğik Karakter Tanıma Başarısını Arttırmak için Yeni Bir Yöntemin Kullanılması

Ahmet ÇELİK^{1,*}

¹Kütahya Dumlupınar Üniversitesi, Tavşanlı Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, 43300, Tavşanlı/KÜTAHYA

Öz

Basılı belge ya da görüntülerdeki karakterlerin okunarak, düzenlenebilir metinler haline getirilmesi Optik Karakter Tanıma (OKT) yöntemleriyle gerçekleştirilmektedir. Karakter tanıma yöntemlerinin temelinde görüntü işleme basamakları vardır. İlk olarak basılı belgelerin veya görüntülerin bir kamera ya da tarayıcı yardımıyla sayısal ortama aktarılması gerekmektedir. Ancak belge görüntülerinin elde edilirken ışık, gölge ve arka plan gibi çevresel parametrelere dikkat edilmediğinde karakter tanıma zor olmaktadır. Ayrıca yine belge ya da görüntülerin, eğik olarak sayısal ortama aktarılmasıyla, karakterlerde meydana gelen eğiklikler, el yazısıyla karakterlerde oluşan eğiklikler ya da italik (eğik) biçimli karakterlerdeki eğiklikler OKT tanıma başarısını olumsuz etkilemektedir. Günümüzde birçok açık ya da kapalı kaynak kodlu OKT yazılım uygulaması vardır. Tesseract en yaygın kullanılan ve karakter tanıma başarısı yüksek olan açık kaynak kodlu yazılımdır. Bu çalışmada, açık kaynak kodlu Tesseract karakter tanıma yazılımının eğik karakter tanıma başarısını arttırmaya yönelik, bir görüntü işleme yöntemi geliştirilmiştir. Geliştirilen yöntem, Dithering (Titreşim) görüntü işleme algoritmasının 1x4 boyutlu yatay biçime dönüştürülmesiyle elde edilmiştir. Yapılan testlerde, Arial ve Times New Roman yazı stillerinin eğik biçimli karakterleri kullanılmıştır. Birbirine benzeyen 429 karakter üzerinde yapılan test sonucunda, geliştirilen yöntemin %33'e varan oranlarda, Tesseract yazılımının başarısını arttırdığı görülmüştür.

Using a New Method to Improve Italic Character Recognition Success

Abstract

Reading the characters on the printed document or image, making into editable texts are realized by Optical Character Recognition (OCR) methods. Image processing steps are at the basis of character recognition methods. First step of printed documents or images should be transferred to digital media using camera or scanner. However, character recognition is difficult when environmental parameters such as light, shadow and background are not taken into consideration while obtaining document images. In addition, by transferring oblique documents or images to the digital environment, making characters italic, the characters of handwriting or the italic characters negatively affect the success of OCR recognition. Today, there are many open or closed source OCR software applications. Tesseract is the most widely used open source software with high character recognition success. In this study, a preprocessing image processing method was developed to increase the success of the open source Tesseract character recognition software, italic character recognition. The developed method was obtained by converting the Dithering image processing algorithm into a one-dimensional horizontal format. In the tests carried out, italic characters of Arial and Times New Roman writing styles were used. As a result of the test carried out, 429 characters that similar to each other, it was observed that developed method increased, success of Tesseract software rates up to 33%.

Makale Bilgisi

Başvuru: 28/11/2020

Yayın: 30/04/2021

Anahtar Kelimeler

Karakter tanıma

Tesseract

Eğik karakter

Görüntü işleme

Titreşim algoritması

Keywords

Character recognition

Tesseract

Italic character

Image processing

Dithering algorithm

1. GİRİŞ

Görüntü içindeki karakterleri düzenlenebilir karakterlere dönüştürmek için Optik Karakter Tanıma (OKT) yöntemleri kullanılmaktadır. OKT yönteminin başarısı, metin belgesi üzerindeki önışlemine bağlıdır. Farklı stil, boyut, biçim ve karmaşık arka plan nedeniyle OKT başarısı değişebilmektedir [1].

Sayısal belgeler elektronik platformlarda yaygın olarak kullanılmaktadır. Dolayısıyla kitaplar, gazeteler, formlar, faturalar gibi birçok yazılı belgenin sayısal düzenlenebilir belgelere dönüştürülmesi kaçınılmazdır. Bunu yapmanın en uygun yöntemi Optik Karakter Tanımadır. Bu yöntemde genellikle belgeler taranmakta ve ardından taranan görüntüler OKT yazılımlarına gönderilmektedir. Taranan görüntülerin üzerinde OKT yazılımlarındaki başarısı için bir miktar önışleme gerekmektedir [2].

OKT yöntemlerinde, bir belge görüntüsünden metin karakterlerini tespit edilirken, bazı belgeler, genellikle karakter algılama algoritmalarını yanlış yönlendiren arka plan görüntüleri ile birlikte gelebilir. Örneğin, arka plan resmindeki küçük noktalar veya keskin kenarlar genellikle karakter olarak algılanabilmektedir. Ancak bir görüntü ön işlem yöntemi kullanılarak, OKT sisteminden önce belge görüntülerini geliştirilmektedir. Sonra renkli görüntüleri griye dönüştürülerek, arka plan görüntüleri metin karakterlerinin kalitesini kaybetmeden etkili bir şekilde kaldırılabilir [3].

OKT yöntemlerinde, metni parazit ve arka plandan ayırmak için çeşitli işlemler gerçekleştirilir ve metin rengi hakkında renk bilgileri sağlanır. Bu sistemlerde karakterleri böldükten ve karakterleri grafiklerden ayırdıktan sonra karakterler bozulabilmektedir. OKT sistemi, bozulan karakterleri tanıyamaz ve böylece verimlilik ciddi şekilde etkilenebilmektedir [4].

Günümüzde sayısal görüntüleri elde etmek için tarayıcı veya kamera gibi büyük cihazlara ihtiyaç kalmamıştır. Artık küçük taşınabilen akıllı telefonlarda kameralar yardımıyla kolayca karakter içeren görüntüler sayısal ortama aktarılmaktadır. Ancak bir telefon kamerasından belge görüntüsü yakalamak basit görünse de mükemmel bir resim elde etme şansı azdır. Resim kısıtsız bir ortamda yakalanırken olduğu gibi, belgenin okunabilirliğini daha fazla etkileyen, görsel kalitesini engelleyecek şekilde kayma ihtimali vardır. Düşük kaliteli belgeler kötü sonuç verir. Belge görüntüleri, bulanıklık, düzensiz aydınlatma, perspektif bozulması, düşük çözünürlük ve lekeler olduğunda daha zor okunabilmektedir. Bu yüzden görüntüler üzerinde temizlik yapılması (istenmeyen piksellerin silinmesi) karakter tanıma başarısını arttırmaktadır [5].

Kessi vd. [4], PDE (Kısmi Diferansiyel Denklem) tabanlı yaklaşımla ilk Karakter Yeniden Yapılandırma Sistemini sunmuşlardır. Bu sistem, Metin ve grafikler arasındaki ayrımın bıraktığı eksik parçaları tamamlamıştır. ABBY Fine Reader ile OCR motorunun uygulanması, OCR hatalarında önemli azalma olduğunu göstermektedir. Deneyde hiçbir eğitim adımı olamamasına rağmen daha iyi performans göstermektedir.

Chan vd. [6], Termal görüntülemeyle elde edilen görüntüler üzerindeki karakterlerin tespitini gerçekleştirmişlerdir. Ayrıca sıcaklık ölçeği bilgisine sahip olmayan termal görüntülerin işlenerek içeriklerindeki değerler optik karakter tanıma yöntemiyle tanımlanmıştır. Uygulama önceden tanımlanan kütüphane içindeki karakterlerin termal görüntüden elde edilen karakterlerin karşılaştırılması temeline dayanmaktadır.

Priambada vd. [7], Endonozya'da yol işaretlerini tanımlamak bir çalışma yapmışlardır. Yapılan çalışmada optik karakter tanıma yöntemi olarak Tesseract kütüphanesi kullanılmıştır. Uygulamada video görüntüsü içinden içinde karakter bulunan görüntüleri alarak görüntüleri önce RGB modundan HSV ye dönüştürülmüştür. Daha sonra bu görüntüler üzerinde OTSU eşikleme kullanarak Canny kenarlık tespit gerçekleştirilmiştir. Video görüntülerinden karakter okuma zor olduğunda ROI (Region of Interest: İlgi Alanı) yöntemi kullanılmıştır.

Saray vd. [8], yabancı ülkede seyahat esnasında yol tabelalarının okunması ve istenilen dile çevrilmesi amacıyla bir uygulama geliştirmişlerdir. Kamera sistemi için Arduino kart kullanarak bir donanım tasarlamışlar ve donanım üzerinde Optik karakter tanımlamak için EMGU.CV kütüphanesi kullanmışlardır.

Şenyıldız vd. [9], yaptıkları çalışmada, akıllı telefon yardımıyla T.C. nüfus cüzdanı üzerindeki bilgileri, karakter tanıma metodu kullanılarak elde etmişlerdir. Bu çalışmada görüntü işleme teknikleri kullanılarak karakterlerin olduğu bölgeler ayrılmıştır. Belirli sınırlar içinde %100'e yakın başarı elde edilmiştir.

Shahreza vd. [2], yaptıkları çalışmada, optik karakter tanıma yöntemi kullanmak amacıyla taranan gazete sayfalarında Farsça / Arapça metinleri ayırt etmek için bir yöntem sunmuşlardır. Kullanılan yöntemde nokta / bağlantı karakterleri gibi Farsça / Arapça metinlerin özellikleri dikkate alınmıştır.

Shen vd. [3], yaptıkları çalışmada, bir belgenin arkaplan görüntüsünü çıkarıp, üzerindeki karakterleri tanımlarken, Tesseract (açık kaynaklı bir OKT yazılımı) ve iki ticari OKT yazılımı ABBYY Finereader ve HANWANG (Çince karakterler için OKT yazılımı) karşılaştırmışlardır. Tesseract yazılımının HANWANG ve ABBYY Finereader yazılımlarından daha iyi sonuç verdiğini gösterilmiştir.

Karasu vd. [10], yaptıkları çalışmada, yaygın olarak kullanılan, açık kaynak kodlu Tesseract, CuneiForm ve GOCR optik karakter tanıma sistemlerini oluşturdukları veri setleri üzerinde karşılaştırmışlar. Karşılaştırma sonucunda Tesseract optik karakter sisteminin en iyi performans sağladığını gösterilmiştir.

Özalp vd. [11], yaptıkları çalışmada, araç plaka karakterlerini tanımak için, GOCR açık kaynak kodlu optik karakter tanıma kütüphanesi kullanmışlardır. Yapay Sinir Ağları da görüntü işleme adımlarında kullanılmaktadır. Yapılan çalışmada Yapay Sinir Ağı algoritmalarından biri olan Rumelhart yöntemini kullanarak deneysel bilgileri eğitilmiştir. Yapılan çalışmalar sonucunda %80-%90 değer aralığında başarı elde edilmiştir.

Saber vd. [12], yaptıkları çalışmada, iki Arap Optik Karakter Tanıma sistemi incelemişlerdir. Bunlar biri, Automatic Reader 11.0 Gold (Sakhr) ve diğeri Readiris Pro 11 Underground Edition (Readiris) tanıma sistemidir. Gazete, kitap ve dergi gibi farklı gerçek taranmış görüntü türleri üzerinde, iki Arap OKT sistemini farklı yazı stilleri (Arabic, Arial ve Times New Roman) ve farklı yazı boyutlarında (12, 14, 16 ve 18) kullanılarak karşılaştırılmıştır.

Patel vd. [1], yaptıkları çalışmada, Araç plakasındaki karakterleri okumak için açık kaynak kodlu Tesseract ile ticari bir yazılım olan Transym yazılımını karşılaştırmışlardır. Yapılan testler sonucunda Tesseract ile %70 oranında, Transym ile %47 oranında başarılı tespit gerçekleştirilmiştir.

Hadar vd. [13], yaptıkları çalışmada, OKT tanıma başarısını arttırmak için tek katmanlı ve çok katmanlı sinir ağları oluşturularak eğitim süreci gerçekleştirmişlerdir. Katman parametreleri olarak yazı stilleri ve yazı boyutlarını kullanmışlardır. Karakter örnekleri olarak, en sık karşılaşılan “e”, “t”, “a” ve “o” karakterleri üzerinde testler gerçekleştirmişlerdir. Karakterlerin merkez noktalarındaki değişimler üzerinde yoğunlaşmışlardır.

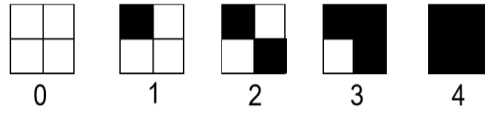
2. DITHERING ALGORİTMASI

Dithering (Titreşim) algoritması görüntü işleme yöntemlerinden biridir. Bu algoritma hem görüntüler hem de videolar üzerinde uygulanabilmektir. Ayrıca ses sinyal görüntüleri üzerinde de kullanılmaktadır. Bu algoritmanın genelde kullanım amacı düzeltme ya da keskinleştirme yapmaktır. Dithering işlemi genel iki boyutlu ya da tek boyutlu filtrelerle yapılabilmektedir.

Video görüntüleri üzerinde belirli bölgenin kalitesini arttırmak istendiğinde tersine titreşim yöntemi kullanılabilir. Ancak bir görüntü üzerinde hızlı yapılmak istendiğinde en basit yöntem olan sıralı titreşim yöntemi tercih edilmektedir. Bu da Bayer Dithering olarak bilinmektedir [14].

Titreşim işleminde, bir görüntünün ayrıntıları ortaya çıkarılarak daha doğru yorumlanmasını sağlayabilmektedir. Bunu görüntü üzerinde bazı noktalara pikseller ekleyerek veya çıkararak yapılmaktadır [15].

Görüntü üzerinde titreşim işlemi sıralı ya da sırasız olarak gerçekleştirilebilmektedir. Sıralı titreşimde verilen matris verilerine göre her piksellin gri değeri kıyaslanarak beyaz ya da siyah renkli biçime dönüştürülmektedir. Sırasız titreşimde ise düzensiz olarak görüntünün belirli bölgelerine beyaz pikseller eklenmektedir. Rastgele eşik değerleri seçildiğinden farklı zamanlarda değişik sonuçlarla karşılaşılabilir. Şekil 1 üzerinde 2x2 sıralı titreşim matrisinin eşik değerine göre sonuçlarını göstermektedir [16].



Şekil 1. 2x2 boyutlu Sıralı Titreşim Matrisi [16].

Şekilde dört pikselin titreşim matrisi kıyaslamalarına göre oluşan siyah beyaz pikseller görülmektedir. 2x2 sıralım titreşim matrisinden yatay ve dikey iki pikselin kıyaslamaları yapılmaktadır. Görüntü pikselinde değişiklikler hem dikey hem de yatay olarak gerçekleşebilmektedir. Bu yöntemde göre kullanılan bu dört pikselin birlikte beş farklı durumu ortaya çıkmaktadır [16].

Christou vd. [17], yaptıkları çalışmada, görüntüler üzerinde en uygun titreşim ve gürültü biçimlendirme yöntemini bulmayı amaçlamışlardır. Sinyal işleme temeline dayanarak titreşim yönteminin sayısal ses dosyaları üzerinde de uygulanabileceğini gösterilmiştir.

Cornel vd. [18], yaptıkları çalışmada, Zoraki Rastgele Örnekleme (Forced Random Sampling) adlı uygulamayı Grafik İşlemci Birimleri (GPU) üzerinde verimli bir şekilde çalıştırmışlardır. Bu çalışmada mavi renkli gürültüler rastgele titreşim yöntemiyle kaldırılmıştır. 213 örnek görüntü GPU ve CPU üzerinde işlenerek testler gerçekleştirilmiştir. Ayrıca sıralı titreşim matrisleriyle, mavi gürültü özelliklerine sahip noktaların, gri tonlamalı bir görüntüye dağıtılmasını sağlanmıştır. Böylece düzenli görüntü işlem basamağının, kolayca paralel olarak işlenebileceğini gösterilmiştir.

3. TESSERACT KARAKTER TANIMA YÖNTEMİ

Tesseract optik karakter tanıma sistemi, 1984 ve 1994 yıllarında HP firması tarafından geliştirilmiştir [19]. 2005 yılından itibaren açık kaynak olarak dağıtılmaya başlanmıştır [20]. Tesseract yapısında Adaptive Thresholding (Adaptif eşikleme) yöntemi kullanılmaktadır [21]. Şuanda Google tarafından geliştirilmekte ve yönetilmektedir. Birçok yabancı dili desteklemektedir.

4. MATERYAL ve METOT

Bu çalışmada, 1X4 boyutunda matris kullanılarak, uygulama gerçekleştirilmiştir. Ancak karakter içeren belgenin görüntüsü önce renkli formattan gri renk formatına dönüştürülmüştür. Görüntü içindeki her pikselin değeri renkli RGB formatından (Red, Green, Blue: Kırmızı, Yeşil, Mavi) griye dönüştürülmesi RGB değerlerinin ortalaması alınarak gerçekleşir. Denklem 1 de gri formata dönüştürme işlemi gösterilmektedir.

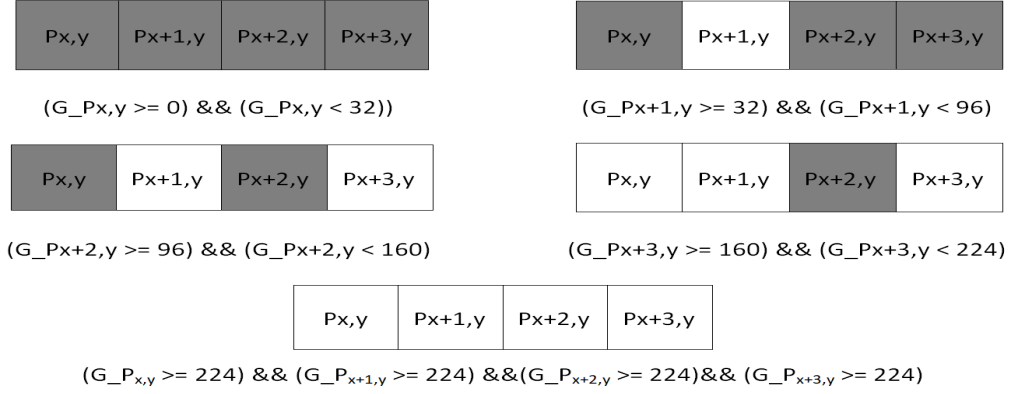
$$G_{P_{x,y}} = \frac{(P_{Red(x,y)} + P_{Green(x,y)} + P_{Blue(x,y)})}{3} \quad 1)$$

Denklemdaki $G_{P_{x,y}}$ kavramı x,y konumundaki pikselin gri değerini, $P_{Red(x,y)}$, x,y konumundaki pikselin kırmızı renk değerini, $P_{Green(x,y)}$, x,y konumundaki pikselin yeşil renk değerini ve $P_{Blue(x,y)}$, x,y konumundaki pikselin mavi renk değerini göstermektedir.

Elde edilen gri değerden ikilik Bitmap biçime dönüştürme ise tüm dört piksellik parçalar halinde yatay olarak titreşim matrisindeki eşik renk değerleriyle kıyaslanarak elde edilmiştir. Sonuç olarak ikilik görüntüde pikselin renk değeri beyaz ya da siyah olarak gösterilmektedir. Bu çalışmada geliştirilen Yatay Titreşim algoritmasının matematiksel ifadesi Denklem 2 de gösterilmektedir.

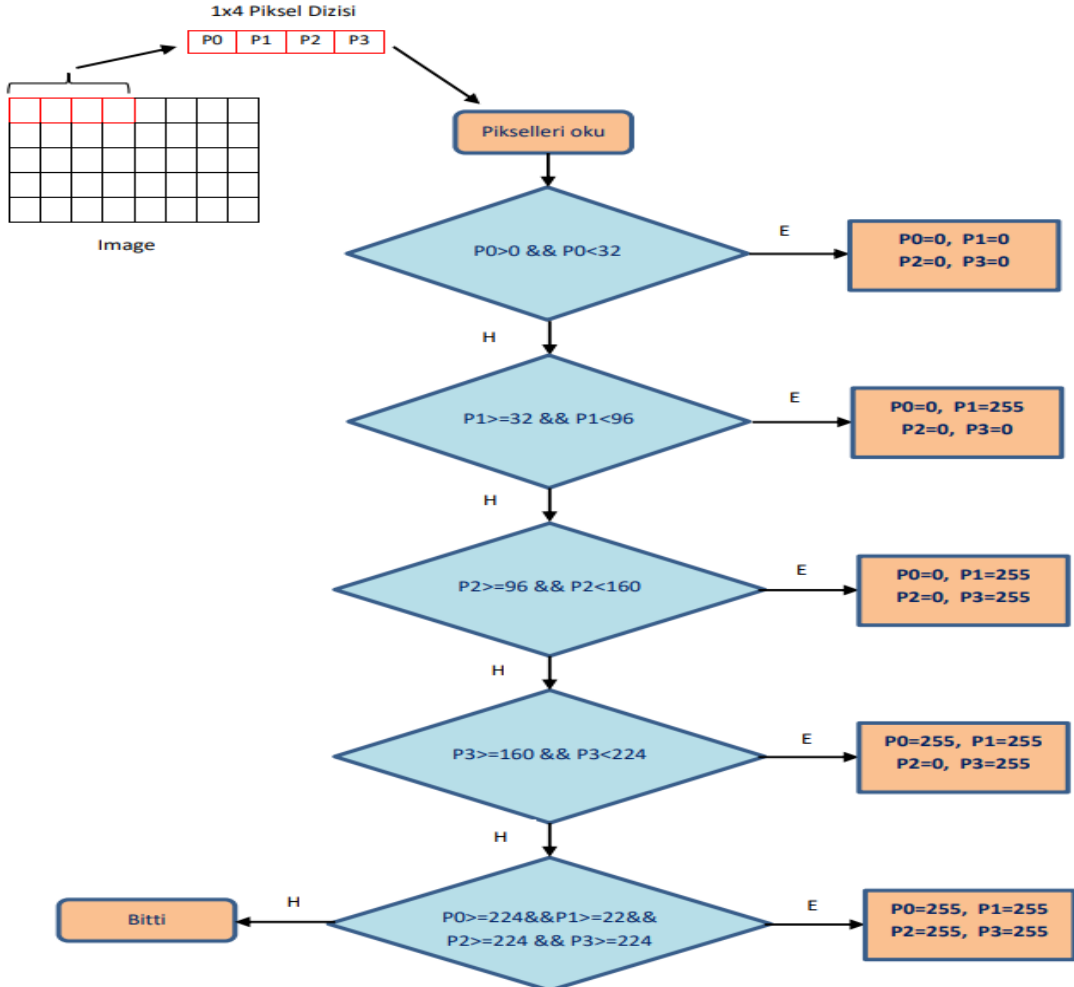
$$\begin{aligned} & \text{Eğer } (G_{P_{x,y}} \geq 0) \ \&\& (G_{P_{x,y}} < 32) \rightarrow P_{x,y} = 0, P_{x+1,y} = 0, P_{x+2,y} = 0, P_{x+3,y} = 0; \\ & \text{Eğer } (G_{P_{x+1,y}} \geq 32) \ \&\& (G_{P_{x+1,y}} < 96) \rightarrow P_{x,y} = 0, P_{x+1,y} = 255, P_{x+2,y} = 0, P_{x+3,y} = 0; \\ & \text{Eğer } (G_{P_{x+2,y}} \geq 96) \ \&\& (G_{P_{x+2,y}} < 160) \rightarrow P_{x,y} = 0, P_{x+1,y} = 255, P_{x+2,y} = 0, P_{x+3,y} = 255; \\ & \text{Eğer } (G_{P_{x+3,y}} \geq 160) \ \&\& (G_{P_{x+3,y}} < 224) \rightarrow P_{x,y} = 255, P_{x+1,y} = 255, P_{x+2,y} = 0, P_{x+3,y} = 255; \\ & \text{Eğer } (G_{P_{x,y}} \geq 224) \ \&\& (G_{P_{x+1,y}} \geq 224) \ \&\& (G_{P_{x+2,y}} \geq 224) \ \&\& (G_{P_{x+3,y}} \geq 224) \\ & \rightarrow P_{x,y} = 255, P_{x+1,y} = 255, P_{x+2,y} = 255, P_{x+3,y} = 255; \end{aligned} \quad 2)$$

Denklemden x, y olarak belirtilen konumlardaki, $G_{P_{x,y}}$, $G_{P_{x+1,y}}$, $G_{P_{x+2,y}}$ ve $G_{P_{x+3,y}}$ gri değer kıyaslamalarına göre piksellerin renk değerleri siyah (0) ve beyaz (255) biçiminde renk değerleri elde edilmiştir. Şekil 2’de geliştirilen 1x4 boyutlu, Yatay Titreşim yönteminin, pikseller üzerindeki etkisi ve eşik değerleri gösterilmektedir.



Şekil 2. Yatay Titreşim yönteminin pikseller üzerinde etkisi

Şekildeki $P_{x,y}$, $P_{x+1,y}$, $P_{x+2,y}$, ve $P_{x+3,y}$ belirlenen konumlardaki pikseller üzerinde titreşim matrisi yardımıyla eşikleme uygulandığında, etkilenen ve renkleri siyah yada beyaz renge dönüşen pikselleri göstermektedir. $G_{P_{x,y}}$, $G_{P_{x+1,y}}$, $G_{P_{x+2,y}}$ ve $G_{P_{x+3,y}}$ ise belirlenen konumlardaki RGB renkli piksel değerlerinin gri renkli piksel değerlerini göstermektedir. Bu çalışmada kullanılan yöntemin akış çizelgesi şekil 3 üzerinde görülmektedir.



Şekil 3. Yatay Titreşim algoritmasının akış çizelgesi

Şekildeki ilk aşamada görüntü içinden 1x4 boyutunda piksel dizileri (yatay olarak birbirine komşu dörder piksel P0, P1, P2 ve P3) okunmaktadır. Bu piksel dizileri görüntünün sırayla tüm piksel satırlarından okunarak ve akış çizelgesi üzerindeki adımlar gerçekleştirilmektedir. Akış çizelgesinde işlenen görüntü gri renk formatında olduğundan P0, P1, P2 ve P3 piksellerinin renkleri gridir. Çizelgedeki beş karar adımı vardır ve bu adımlar sırayla uygulanmaktadır. İlk karar adımında olumlu sonuç (Evet: E) oluştuğunda, P0, P1, P2 ve P3 piksellerinin hepsinin yeni değeri siyah renk (0) olmaktadır. En son karar adımında olumlu sonuç oluştuğunda ise P0, P1, P2 ve P3 piksellerinin hepsinin yeni beyaz renk (255) olmaktadır. Diğer karar adımlarında olumlu sonuç oluştuğunda ise oluşan yeni renk değerleri tümünde aynı olmamaktadır. Karar adımlarında her olumsuz (Hayır: H) sonuç oluştuğunda diğer dallanmalara geçilmektedir.

Düz metin biçimindeki karakterler, saat yönünde eğilerek eğik metin biçimine dönüştürülmektedir. Düz metin biçimiyle, eğik metin biçimi arasında 12 derecelik eğim farkı olduğu ölçülmüştür. Şekil 4 üzerinde örnek “I” ve “B” karakterlerinin düz ve eğik biçimleri gösterilmektedir. Diğer bütün karakterlerin eğik biçimleri aynı şekilde eğik olmaktadır.



Şekil 4. “I” ve “B” karakterlerinin Düz ve Eğik biçimleri

Bu çalışmada, test verileri için toplam 429 eğik karakter kullanılmıştır. Bu karakterler rastgele 10 adet gruba ayrılmıştır. Arial ve Times New Roman yazı stilleri için aynı test grupları kullanılmıştır. Şekil 5 üzerinde birinci gruba ait karakter verileri kullanılmıştır.

23	BB	23	BB
18	SB	18	SB
38	BZ	38	BZ
33	SB	33	SB
73	ZZ	73	ZZ
71	SS	71	SS
77	GSB	77	GSB
78	SBC	78	SBC
67	UV	67	UV
17	VG	17	VG

a) b)

Şekil 5. Grup 1 test karakterleri: a) Eğik-Arial b) Eğik-Times New Roman

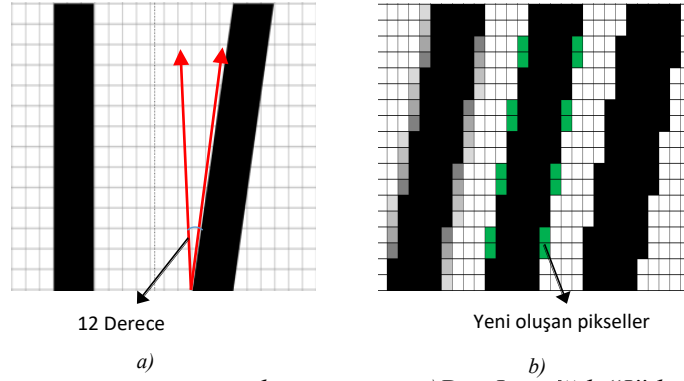
Karakter grupları “8”, “B”, “5”, “S”, “G”, “6”, “Z”, “2”, “C”, “U” ve “V” gibi birbirine en çok benzeyen karakterlerden oluşturulmuştur. Böylece tanımlama basamağında, en çok birbiriyle karıştırılan karakterlerin okunma başarısı test edilmiştir. Toplam 429 karakter, on farklı grup içinde dağıtılmış ve test veri grupları oluşturulmuştur. Test için kullanılan karakter gruplarının sayıları, tablo 1 üzerinde gösterilmektedir.

Tablo1.429 karakterlerin test veri grupları içindeki sayıları

Grup Adı	Karakter Sayısı	Grup Adı	Karakter Sayısı
Grup 1	42	Grup 6	43
Grup 2	43	Grup 7	45
Grup 3	45	Grup 8	41
Grup 4	43	Grup 9	42
Grup 5	43	Grup 10	42

5. BULGULAR ve TARTIŞMA

Bu çalışmada kullanılan yöntemle göre ile koşulu sağlayan pikseller karakterin sol tarafına ve sağ tarafına eklenmektedir. Piksellerdeki değişiklik sadece yatay olarak gerçekleşmekte fakat dikey olarak herhangi bir değişiklik olmamaktadır. Böylece eğik yazı metinleri düzleşmekte(dikey hale gelme) ve karakter tanıma işlemine fayda sağlamaktadır. Eğik “I” karakteri piksellerinin, sayısal olarak grafik üzerindeki yerleşimi ve algoritma uygulanması sonucu meydana gelen değişiklik şekil 6 üzerinde gösterilmektedir.



Şekil 6. “I” karakterinin görüntü üzerinde gösterimi: a) Düz I ve Eğik “I” karakteri b) Eğik karakter üzerinde Yatay Titreşim algoritması uygulama sonucu meydana gelen değişiklik

Şekildeki yeşil renkli pikseller titreşim algoritması uygulandıktan sonra oluşan piksellerdir. Uygulama sonucu eğiklik derecesinin azaldığı görülmektedir. Bu da karakter okuma başarısını arttırmaktadır. Geliştirilen 1x4 boyutlu yatay titreşim algoritma uygulamasının, örnek karakterler (I ve B) üzerindeki etkisi şekil 7 üzerinde gösterilmektedir. Şekilden görüldüğü üzere işlenen “I” ve “B” karakterlerinde, düz karakter haline dönüşmek için eklentiler oluşmuştur. Ancak düz I harfinde bir etki görülmemiştir.



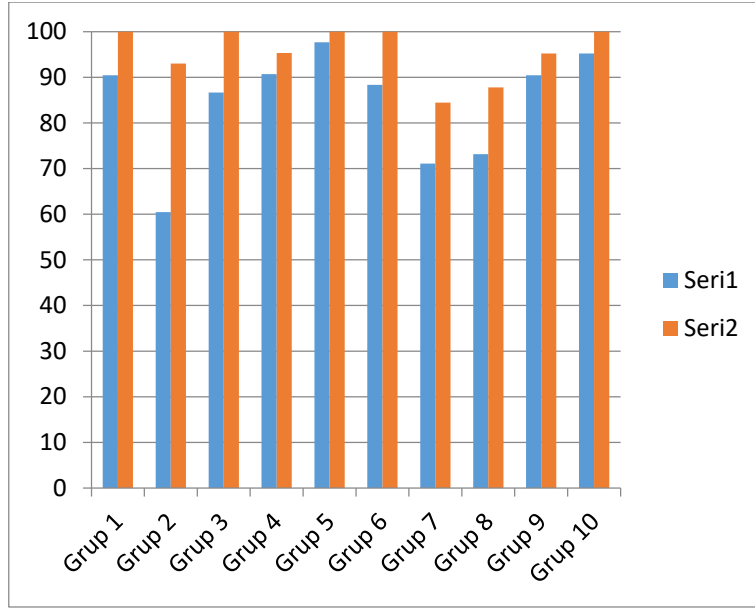
Şekil 7. Geliştirilen algoritmanın karakterlere uygulama sonucu: a) Düz I ve B b) Eğik I ve B

Eğik biçimli Arial yazı stiliyle oluşturulan karakter gruplarının tanıma başarısı, tablo 2 üzerinde gösterilmektedir. Tablodaki değerlere göre, Grup 2 karakterlerini Tesseract ile tanıma başarısı %60,465 iken, yine aynı Grup 2 verileri üzerinde Yatay Titreşim yöntemi uygulandıktan sonra Tesseract tanıma başarısı %93,023 değerine yükselmiştir. Bu yöntem, yaklaşık %33 oranında başarı sağlamıştır. Bütün eğik biçimli Arial yazı stiliyle oluşturulmuş Grup karakterlerinin tanıma başarı oranına bakıldığında, Yatay Titreşim yöntemi, ön işlem olarak kullanıldığında, Tesseract yazılımının başarısını arttırdığı görülmektedir.

Tablo 2. Kullanılan yöntemlerin Arial yazı stili üzerindeki başarı oranları

Grup Adı	Tesseract(%)	Yatay Dithering(%) + Tesseract
Grup 1	90,476	100,000
Grup 2	60,465	93,023
Grup 3	86,667	100,000
Grup 4	90,698	95,349
Grup 5	97,674	100,000
Grup 6	88,372	100,000
Grup 7	71,111	84,444
Grup 8	73,171	87,805
Grup 9	90,476	95,238
Grup 10	95,238	100,000

Tablo 2 verilerine göre, elde edilen grafik ise şekil 8 üzerinde görülmektedir. Burada test veri grup karakterleri üzerinde, ön işlemsiz Tesseract ile hiçbir zaman %100 başarı elde edilememiştir. Ancak Yatay Titreşim algoritması kullanıldığında, beş grubun karakterlerini tanıma başarısı %100 değerine yükselmiştir. 429 karakterin ön işlemsiz, Tesseract ile karakter tanıma başarısı ortalama % 84,434 iken, Yatay Titreşim algoritması, ön işlem olarak kullanıldıktan sonra, karakter tanıma başarısı ortalama, %95,585 değerine yükselmiştir.

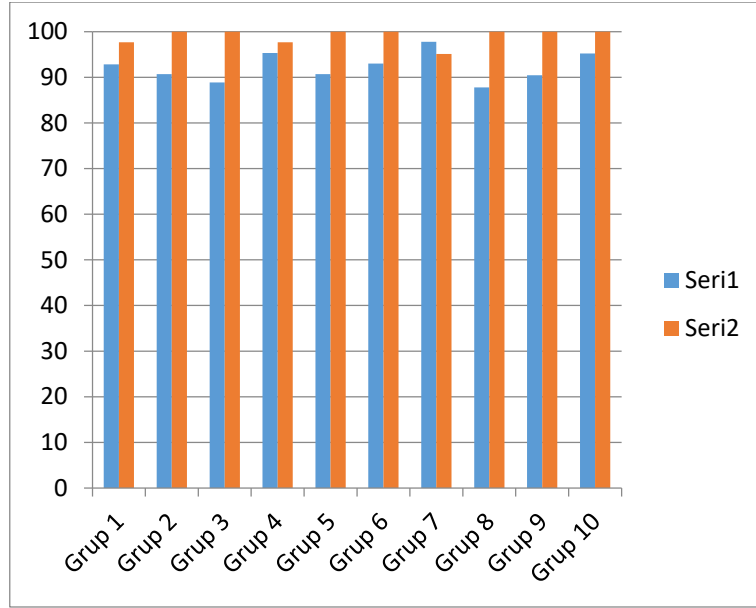
**Şekil 8.** Kullanılan yöntemin Arial yazı üzerindeki başarı etkisini gösteren grafik

Eğik biçimli Times New Roman yazı stiliyle oluşturulan karakter gruplarının tanıma başarısı tablo 3 üzerinde gösterilmektedir. Tablodaki değerlere göre, Grup 8 karakterlerini Tesseract ile tanıma başarısı 87,805 iken, yine aynı Grup 8 verileri üzerinde Yatay Titreşim işlemi uygulandıktan sonra Tesseract tanıma başarısı %100 değerine yükselmiştir. Bu yöntem, yaklaşık %13 oranında başarı sağlamıştır. Bütün eğik biçimli Times New Roman yazı stiliyle oluşturulmuş Grup karakterlerinin tanıma başarı oranına bakıldığında, Yatay Titreşim yöntemi ön işlem olarak kullanıldığında, Tesseract yazılımın başarısını arttırdığı görülmektedir.

Tablo 3. Kullanılan yöntemlerin Times New Roman yazı stili üzerindeki başarı oranları

Grup Adı	Tesseract(%)	Yatay Dithering(%) + Tesseract
Grup 1	92,857	97,674
Grup 2	90,698	100,000
Grup 3	88,889	100,000
Grup 4	95,349	97,674
Grup 5	90,698	100,000
Grup 6	93,023	100,000
Grup 7	97,778	95,122
Grup 8	87,805	100,000
Grup 9	90,476	100,000
Grup 10	95,238	100,000

Tablo 3 verilerine göre, elde edilen grafik ise şekil 9 üzerinde görülmektedir. Burada da test veri grup karakterleri üzerinde, ön işlemsiz Tesseract ile hiçbir zaman %100 başarı elde edilememiştir. Ancak Yatay Titreşim algoritması kullanıldığında, yedi grubun karakterlerini tanıma başarısı %100 değerine yükselmiştir. 429 karakterin ön işlemsiz, Tesseract karakter tanıma başarısı ortalama % 92,281 iken, Yatay Titreşim algoritması, ön işlem olarak kullanıldıktan sonra, karakter tanıma başarısı ortalama 99,047 değerine yükselmiştir.

**Şekil 9.** Kullanılan yöntemin Times New Roman yazı üzerindeki başarı etkisini gösteren grafik

5. SONUÇ ve ÖNERİLER

Bu çalışmada eğik karakterlerin tanımda karşılaşılan zorluklara çözüm bulmak için bir yöntem geliştirilmiş ve bu yöntemin başarısı gösterilmiştir. Eğik karakterlere, belge görüntü elde edilirken kamera ya da tarayıcıdan düzgün yatay-dikey yönlendirme yapılmadığında karşılaşılabilmektedir. Ayrıca el yazı biçimlerinde ya da metin editörlerini kullanırken, karakter yazımında eğik yazı biçimi seçildiğinde de karşılaşılabilmektedir.

Birçok araştırmacı OKT tanıma sistemleri üzerinde çalışmalar yapmış ve yaygın kullanılan açık kaynak kodlu Tesseract yazılımının başarısının yüksek olduğunu göstermiştir. Yapılan testlerde Tesseract

yazılımının normal biçimli karakterleri tanıma başarısının yüksek olduğu ancak eğik biçimli karakterleri tanıma başarısının düşebildiği görülmüştür. Arial stilini kullanan bazı karakter gruplarında %60 değerinde karakter tanıma başarısı görülmüştür. Bu çalışmada metin editörleri yazılan Arial ve Times New Roman yazı stillerinin kullanıldığı eğik karakterler üzerinde testler gerçekleştirilmiştir.

Tesseract yazılımının başarısını arttırmak için bir ön görüntü işlem yöntemi geliştirilmiştir. Dithering (Titreşim) algoritması, 1x4 boyutlu yatay titreşim algoritmasına dönüştürülmüştür. İlk pikselin gri değerine göre komşu pikseller siyah ya da beyaz renge dönüştürülmüştür. Böylece piksel renk değerlerin de sadece yatay değişimler gerçekleşmiştir. Böylece eğik karakterlerin normal biçime dönüşme eğilimleri gözlemlenmiştir. Bu yöntem karakterler üzerinde ilk defa uygulanmıştır.

Yapılan bu çalışmada, şekilleri birbirine benzeyen 429 eğik biçimli karakterin üzerinde Arial ve Times New Roman yazı stilleri uygulanmış ve görüntü formatında kaydedilmiştir. Daha sonra görüntüler üzerinde önce sadece Tesseract yazılımıyla OKT gerçekleştirilmiş ve başarısı gözlemlenmiştir. Daha sonra bu görüntüler üzerinde önce yatay titreşim algoritması uygulanmış takiben Tesseract yazılımı ile OKT gerçekleştirilmiştir. Elde edilen sonuçlara göre Yatay Titreşim algoritması uygulanan ve Arial stili eğik karakterlerin tanıma başarısı ortalama % 95,581, Times New Roman stili kullanılan eğik karakterlerin tanıma başarısı ise ortalama % 99,047 olduğu görülmüştür. Arial ve Times New Roman yazı stilleri farklı özelliklere sahip olduğundan, elde edilen başarı oranları da aynı değildir. Bu çalışmadan elde edilen değerler, geliştirilen Yatay Titreşim yönteminin eğik yazı biçimlerini okuma başarısını arttırdığını göstermektedir. Bu yöntem el yazısı karakterlerini veya eğik görüntülerdeki karakterlerin tanınma başarısını arttıracaktır.

KAYNAKLAR

- [1] C. Patel, A. Patel, D. Patel, Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study. *International Journal of Computer Applications*, 55:10 (2012), 50-56.
- [2] S.S. Shahreza, M.T.M. Shalmani, M.H.S. Shahreza, Preparing Persian/Arabic Scanned Images for OCR. *2nd International Conference on Information & Communication Technologies*, (2006) 1332-1336.
- [3] M. Shen, H. Lei, Improving OCR Performance with Background Image Elimination. *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (2015) 1566-1570.
- [4] L. Kessi, F. Lebourgeois, C. Garcia, An Efficient New PDE-based Characters Reconstruction After Graphics Removal. *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)* (2016) 441-446.
- [5] P. Sharma, S. Sharma, Image Processing based Degraded Camera Captured Document Enhancement for Improved OCR Accuracy. *6th International Conference - Cloud System and Big Data Engineering (Confluence)* (2016) 441-444.
- [6] W. T. Chan, T. Y. Lo, C. P. Tso, K. S. Sim, A Method in Applying OCR in Processing of Thermal Images. *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)* (2011) 244-248.
- [7] P. Sharma, S. Sharma, S. Priambada, D. H. Widyantoro, Levensthein Distance as a Post-Process to Improve the Performance of OCR in Written Road Signs. *6'th International Conference-Cloud System and Big Data Engineering (Confluence)* 2016 1-6.
- [8] T. Saray, A. Çetinkaya, A. Okatan, Optik Karakter Tanıma Yöntemi ile Otomatik Tabela Okuyucu. *International Conference on Computer Science and Engineering (UBMK)* (2017) 1-6.
- [9] A. Şenyıldız, A. C. Dirican, Görüntü İşleme Teknikleri ve Akıllı Telefon Kullanarak Nüfus Cüzdanı Tespiti ve Otomatik Bilgi Çıkarılması, Düzce, Türkiye, 2016.

- [10] K. Karasu, M. Baştan, Turkish OCR on Mobile and Scanned Document Images. 23 nd Signal Processing and Communications Applications Conference (SIU) (2015) 2074-2077.
- [11] A. N. Özalp, M. Dügenci, Plaka Tanıma Sistemlerine Yeni Yaklaşımlar. Akademik Bilişim'11 - XIII. Akademik Bilişim Konferansı Bildirileri (2011) 747-751.
- [12] S. Saber, A. Ahmed, M. Hadhoud, Robust Metrics for Evaluating Arabic OCR Systems. International Image Processing, Applications and Systems Conference (2014) 1-6.
- [13] H. I. Avi-Itzhak, T. A. Diep, H. Garland, High Accuracy Optical Character Recognition Using Neural Networks with Centroid Dithering. IEEE Transactions On Pattern Analysis and Machine Intelligence, 17:3 (1995) 218-224.
- [14] R. Ulichney, One-dimensional Dithering, Cambridge Research Lab, Tecnicl Report Series, CRL 98/2, Cambridge, USA, 1998.
- [15] Ronja's Shader Tutorials, Dithering, <https://www.ronja-tutorials.com/2019/05/11/dithering.html>, [Online], 11.06.2020.
- [16] C. Wolberg, Digital Halftoning, Image Processing Course Notes, The City College of The University of NewYork Department of Computer Science, NewYork, USA, 2020.
- [17] C. N. Christou, Optimal Dither and Noise Shaping in Image Processing, University of Waterloo, Ontario, Canada, 2008.
- [18] D. Cornel, R. F. Tobler, H. Sakai, C. Luksch, M. Wimmer, Forced Random Sampling Fast Generation of Importance-Guided Blue-Noise Samples, 33 (2017) 833-843.
- [19] R. Smith, An Overview of the Tesseract OCR Engine, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) 629-633.
- [20] 2020 GitHub, Inc., Tesseract OCR, <https://github.com/tesseract-ocr/tesseract>, [Online], 20.06.2020.
- [21] F. Shafait, T. Breuel, D. Keysers, Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images. 5th Document Recognition and Retrieval Conference, part of the IS&T-SPIE Electronic Imaging Symposium (2008). 5-10.