Bilgisayar Mühendisliği / *Computer Engineering* **DOI: 10.21597/jist.714810**
Araştırma Makalesi / *Research Article*
Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 10(4): 2328-2340, 2020
Journal of the Institute of Science and Technology, 10(4): 2328-2340, 2020
ISSN: 2146-0574, eISSN: 2536-4618

Optik Karakter Tanımada Hata Yayılım Algoritmalarının Performans Kıyaslaması

Ahmet ÇELİK[1*]

ÖZET: Görüntü formatındaki belgelerin içinden karakterlerin veya verilerinin tekrar metin biçimine dönüştürülmesi büyük zaman ve iş gücü kaybı demektir. Günümüzde doküman işlemlerinde, işlem maliyetlerini düşürmek ve verimlilik oranlarını arttırmak istenilmektedir. Okutulacak belgeler üzerinde farklı yazı stilleri, yazı boyutları ve yazı biçimleri olabilmektedir. Ayrıca el yazısı notları da olabilmektedir. Bilgisayar ortamında hazırlanan ve bilinen yazı stilleriyle oluşturulan karakter değerlerinin tekrar düzenlenebilir metin formatına dönüştürme başarısı daha yüksektir ancak el yazısı karakterlerinin dönüştürme başarısı daha düşüktür. Tesseract kütüphanesinin eğitim verilerinin yeterli olmaması sebebiyle bazı yazı biçimlerinde başarı oranı düşük olabilmektedir. Bu çalışmada; OCR teknolojisi için kullanılan Tesseract kütüphanesi yardımıyla farklı yazı stilleri üzerinde, farklı yazı biçimleri uygulanarak, alfabetik karakter ve rakam okutulması gerçekleştirilmiş ve okuma başarı kıyaslaması yapılmıştır. Times New Roman, Calibri ve Arial yazı stilleri üzerinde normal, kalın ve eğik yazı biçimleri uygulanan örnekler kullanılmıştır. Ayrıca Tesseract kütüphanesi kullanımı öncesi, görüntü üzerinde Error Diffusion (Hata Yayılımı) algoritmaları ile iyileştirmeler yapılarak okuma oranları karşılaştırılmıştır. Böylece OCR tanıma yönteminin başarısını arttıran, ön işlem algoritmasının bulunması amaçlanmıştır. Elde edilen değerlere göre; belge üzerinde ön işlem olarak, Floyd Steinberg hata dağılım algoritması kullanımından sonra Tesseract kütüphanesinin daha doğru okuma yaptığı görülmüştür.

Anahtar Kelimeler: Optik karakter tanıma, Tesseract, hata yayılımı, görüntü işleme, makine öğrenmesi

Performance Comparison of Error Diffusion Algorithms in Optical Character Recognition

ABSTRACT: Converting characters or data to text through image formats means loss of time and labor. Today, It is desired to reduce transaction costs and increase efficiency rates in document transactions. For reading have been different writing styles, font sizes and writing formats on the documents. Computer-generated prepared character conversion and known writing style with success back into editable text format, the success of the conversion value higher then handwritten characters. The biggest step for reading in character, separating of characters from background. Due to the lack training data of the Tesseract library, the success rate, lows in some writing formats. In this study, reading alphabetical character and numbers was performed with the help of Tesseract library used for OCR technology and was made on different writing styles by applying different writing styles reading success comparison. Samples using normal, bold and italic writing formats were used on Times New Roman, Calibri and Arial font styles. Also on the image before using Tesseract library, Error Diffusion algorithms were compared with read rates by making improvements. Thus, it is aimed to find a pre-processing algorithm that increases the success of the OCR recognition method. According to the obtained values; As a pretreatment on the document, it was observed that the Tesseract library made a more accurate reading after using the Floyd Steinberg error distribution algorithm.

Keywords: Optical character recognition, Tesseract, error diffusion, image processing, machine learning

[1] Ahmet ÇELİK (Orcid ID: 0000-0002-6288-3182), Kütahya Dumlupınar Üniversitesi, Tavşanlı Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Kütahya, Türkiye

*Sorumlu Yazar/Corresponding Author: Ahmet ÇELİK, e-mail: ahmet.celik@dpu.edu.tr

Geliş tarihi / *Received:* 05-04-2020
Kabul tarihi / *Accepted:* 27-05-2020

GİRİŞ

Optik karakter tanımlama(Optical Character Recognition:OCR) yöntemi, 1955 yılında ilk olarak satış raporlarının, bilgisayar sistemine aktarılmasında kullanılmaya başlanmıştır (Patel ve ark., 2012). Günümüzde birçok uygulamada kullanılan popüler bir yöntemdir. OCR karakter tanımadaki başarısı ön işlemlere ve sınıflama algoritmalarına bağlıdır. Bazı durumlarda görüntü içinden karakterleri almak zor olabilmektedir çünkü karakter farklı boyutlarda, stillerde, konumlarda ve karışık arka planı olan görüntülerde olabilmektedir.

El yazısı ile oluşturulan evrakların OCR yöntemi kullanılarak düzenlenebilir sayısal ortama dönüştürülmesi ileride daha da geliştirilmesi gerekir. OCR yöntemleri makinelerin yani mobil ya da bilgisayar sistemli cihazların metinleri otomatik tanımasını sağlamaktadır. Bu yöntemin çalışması insanın göz ve beyninin beraber işleyişine benzemektedir. Göz görüntü üzerindeki metni görmektedir aslında beyin de göz tarafından okunan metinleri yorumlar. Bilgisayarlı OCR sistemlerinde birkaç problemle karşılaşılmaktadır. Birinci olarak harf ve sayı değerleri arasında çok küçük farklılıklar olduğunda örneğin sıfır (0) ile O harfi çoğunlukla karıştırılmaktadır. İkinci olarak; koyu arka planı olan veya altında başka bir görüntü olan karakterlerin okunması zor olabilmektedir (Patel ve ark., 2012).

OCR yöntemleri, taranmış ya da bir kamera yardımıyla görüntüsü alınmış belgelerin elektronik ortama aktarılmasında ya da istenildiğinde bu kayıtlar içindeki metinlerin bulunmasında kullanılabilir. Bu yöntem makine öğrenmesi yöntemlerini kullanarak kendi eğitim setini geliştirmektedir. Yapay sinir ağları, OCR yöntemlerinde kullanılabilmektedir. El yazısını tanımak zor olduğundan Akıllı Karakter Tanıma(Intelligent Character Recognition:ICR) yöntemi üzerinde çalışmalar devam etmektedir(Koyun ve ark., 2017).

OCR yöntemi kullanılırken eski kitaplar, kalitesiz fotokopi kağıtları ve faks gibi kaynaklardan okuma yapıldığında, bazı tanıma hatalarıyla karşılaşılmaktadır. Ofis uygulamalarında gerçekten faydalı olan OCR yöntemlerinin geliştirilmesi ve okunan bilgilerinin tekrar kontrol edilmesi güvenilirliği sağlayacaktır. Otomatik doğrulamanın OCR hataları üzerinde yapılması faydalı olacaktır. Kelimeler üzerinde aslında iki tip hatalarla karşılaşılmaktadır. Birinci kelime olmayan hatalı okumaların yapılması, ikinci olarak gerçek kelime hatalarıdır. "Bu" kelimesinin "8u"gibi tanınması; kelime olmayan hatalı okumadır. "Bir" kelimesinin "Bin" gibi tanınması; gerçek kelime hatasıdır(Patel ve ark., 2012; Tong ve ark.,1996).

OCR en çok PDF dosyaları üzerinden metin okuma amacıyla kullanılmaktadır. Kamera ya da tarayıcı yardımıyla sayısal ortama alınan görüntü formatındaki dosyaların çarpık/eğik olması ya da bulanık olması, OCR yönteminin kullanılmasının en zor olduğu durumlardır. OCR de ilk adım sınıflandırma ikinci adım ise özellik çıkarmadır. OCR hem mobil hem de bilgisayar ortamlarında kullanılabilmektedir(Mithe ve ark.,2013).

Bilgisayar donanımların iyileşmesi ile görüntü alma birimlerin gelişmesiyle görüntü işlemleri teknikleri de büyük oranda gelişme göstermiştir. Optik karakter tanıma teknolojisi de bu tekniklerden biridir. Optik karakter tanımlama(OCR) görüntü üzerindeki karakterleri okuyarak üzerinde değişiklik yapılabilen metin verilerine dönüştürmektedir. Optik karakter tanıma işlemleri genelde iki bölüme ayrılmaktadır. Bunlar; Otomatik Karakter Tanıma (Automatic Character Recognition: ADC) ve de Metin Tanıma (TR: Text Recognition) bölümleridir. Otomatik karakter tanımada sisteminde tanınan her karakter doğru kabul edilmektedir ve karşılık olarak bir Amerikan Standart Kodlama Sistemi (American Standard Code for Information Interchange: ASCII) kodu oluşturulmaktadır. Metin tanıma ise karakterlerin oluşturduğu kelimelerin tanınmasında kullanılmaktadır(Saray ve ark.,2017).

OCR teknolojisi birçok alanlarda kullanılmaktadır; Özellikle; üretim sektöründe sipariş, arıza, kayıt formlarının okunmasında, personel takip sistemlerinin digital(sayısal) ortama aktarılmasında, banka dekont, çek belgelerinin kayıt edilmesinde, faks evraklarının sayısal ortamına aktarılmasında, mahkeme, proje çıktı evraklarının tekrar bilgisayar ortamına aktarılmasında, basılmış gazete yada dergilerin kayıtlarının oluşturularak kütüphane ortamlarında sunulmasında, vergi, ceza yada bilgilendirme belgelerinin kayıt altına alınmasında çok kolaylık sağlamaktadır. Günümüzde yaygın kullanılan OCR uygulamaları ise Tesseract(GitHub,2020), Abbyy Fine Reader(ABBYY, 2020) ve GOCR(GOCR, 2020) yazılımlarıdır. Ayrıca genelde yüksek hız ve doğruluk oranına sahip, makine öğrenmesi yöntemlerinden biri olan, derin öğrenme tabanlı OCR tanıma uygulamaları da kullanılmaktadır. Bu uygulamaların bazılarında Keras ve Tensorflow kütüphaneleri kullanılarak Evrişimsel sinir ağı (Convolutional Neural Network :CNN) yapısında bir yapay sinir ağı kullanılmıştır (Gider ve ark., 2018; Koyun ve ark., 2017).

Bu çalışmada, OCR için yaygın kullanılan Tesseract kütüphanesinden yararlanılmıştır. Karakterlerin verilerin okutulması gerçekleştirilmektedir. Tesseract kütüphanesi eğitim verilerinin yeterli olmaması sebebiyle Error Diffusion (Hata Yayılımı) algoritmaları ile iyileştirmeler yapılmıştır ve algoritmaların okuma oranları karşılaştırılmıştır.

MATERYAL ve YÖNTEM

Yapılan çalışma, Microsoft Visual Studio 2012 yazılım paketi geliştirme ortamı kullanılarak, Visual C# program dili ile geliştirilmiştir. Karakter tanımlama için Tesseract kütüphanesinden yararlanılmıştır. Kütüphane eğitim verileri tek başına okuma gerçekleştirmek için yeterli değildir. Bu tür kütüphaneleri başarısını attırmak için ya eğitim veri sayını çoğaltmak ya da algoritma kullanım öncesi ön işlemler uygulamak gerekmektedir. Tesseract kütüphanesinde de başarı oranı yüksek bir okuma yapabilmek için karakter içeren gerekli ön işlemlerden geçmesi gerekmektedir bu sebeple hata yayılım (Error diffusion) algoritmalarından yararlanılmıştır. Hata yayılım algoritmaları komşu piksel değerleri ile hesaplamalar yapıp, piksel gürültülerini diğer piksellere aktarıp, belirgin hataları yumuşatmaktır. Bu algoritmalarda kullanılan matrisler resmin üzerinde yukarıdan aşağıya, soldan sağa doğru gezdirilmektedir

Hata yayılım matrislerde hedef Piksel olarak gösterilen piksel, o anda taranmakta olan pikseli belirtir. Algoritmalar görüntüyü soldan sağa, yukarıdan aşağıya tarar, hedef piksel değerlerini komşu pikseller üzerinde belirtilen katsayıları uygular ve hedef piksel değerleri birer birer hesaplanmaktadır.

Yapılan çalışma İngilizce alfabe ve rakam içeren görüntüler üzerinde uygulanmıştır. Yazı boyutu(punto) 24 seçilmiştir. Elde edilen görüntüler 600x100 piksel çözünürlüğe sahip PNG dosya formatındadır. Bu karakter değerleri Arial, Calibri ve Times New Roman yazı stilleriyle oluşturularak ve bu yazı stilleri üzerinde kalın(bold), eğikitalic) biçimlendirme uygulayarak sonuçları test edilmiştir. Şekil 1 üzerinde Times New Roman(kalın) örneği görülmektedir.

Şekil 1. Örnek metin görüntüsü(600x100 çözünürlük-Kalın-Times New Roman yazı stili.

Tesseract Kütüphanesi

Tesseract kütüphanesi, çeşitli işletim sistemleri için geliştirilen optik karakter tanıma yazılımıdır(GitHub,2020). İlk olarak 1985 - 1995 yılları arasında Hewlett-Packard şirketi tarafından

kapalı kaynak bir yazılım olarak geliştirilmiştir. 2005 yılındaysa Hewlett Packard şirketi ve University of Nevada Las Vegas (UNLV)tarafından özgür yazılım olarak yayınlanmıştır. Birçok programlama ara yüz dilini desteklemektedir. Tesseract piyasada mevcut en doğru ve açık kaynak kodlu OCR motoru olarak kabul edilmektedir.Tesseract kütüphanesinde ilk olarak Otsu eşik yöntemiyle karakterler ayrılır, sonra metin satırı tespit edilir daha sonra ise her karakter birbirinden ayrılır. Son adımda ise kelimetahmini gerçekleştirilir. Bu kütüphanede örnek sayısı eğitim amaçla artarsa başarı oranının artacağı görülmektedir. Tesseract kütüphanesinin web platform desteğinin olmaması, emsallerine göre daha yavaş gelişmesi ve dezavantajları arasında gösterilmektedir(Kutlu ve ark., 2014; Smith ve ark. 2007).

Abbyy Fine Reader

189 dili tanıma özelliği bulunan ABBYY Fine Reader uygulaması, görüntü formatındaki dosyaları oldukça doğru bir şekilde okuyarak Microsoft dosya biçimlerine (Word, Excel, PowerPoint) dönüştürebilmekte veya arama yapılabilmektedir. Firmaların günlük olarak alınan ve işlemden geçirilen raporlarını, mektuplarını, sözleşmelerini, faturalarını yani kağıt belge üzerinde çalışma imkanı sağlar(Oki Electric Industry,2020).

ABBYY Fine Reader uygulaması, en son yapay zeka tabanlı OCR teknolojisine sahip olarak her tür belgeyi elektronik ortama alarak dijitalleştirmeyi, veriyi geri almayı, düzenlemeyi, korumayı, paylaşmayı ve ortak çalışmayı kolaylaştırmaktadır(ABBYY, 2020).

GOCR

GOCR bir Optik Karakter Okuma uygulamasıdır. Genel Kamu lisansına sahiptir(GNU) yani açık kaynak kodludur. C programlama diliyle yazılmış ve Linux tabanlı platformlarda çalıştığı gibi Windows platformunda çalışabilmektedir. 2001 yılında Joerg Schulenburg tarafından yazılmış 2010 yılından sonra proje takımı oluşturulmuş ve yazılım daha geliştirilerek çok kısa sürede karakter tanıma görevini yapabilmektedir. PNG, TIFF, JPG gibi birçok görüntü dosyası içinde tarama yapa bilmektedir(GOCR, 2020).

Error Diffussion (Hata Yayılım) Algoritmaları

Son yirmi yılda birçok dağılım tekniği geliştirilmiştir ancak Hata Yayılım (Error-Diffusion Algorithm) basitliği ve kalitesiyle diğerlerinden ayrılmış bir algoritmadır. Hata difüzyon renk taklidi, yazıcı, bilgisayar monitörü veya diğer iki seviyeli ekranlarda gri tonlamalı bir görüntüyü temsil etmek için kullanılan bir tekniktir. Ayrıca özel donanıma kolayca uygulanabilir. Algoritmamızın avantajlarından biri, düşük uygulama maliyeti, ölçeklenebilirliği ve standart hataya dayanıklılık yeteneğidir. Hata yayılım algoritması gri tonlamalı görüntüyü temsil etmektedir(Misra ve ark., 2011). Ancak hata dağılımı algoritması renkli RGB(Red, Green, Blue) görüntüler üzerinde de gürültüleri(istenmeyen pikselleri) yok etmek içinde kullanılabilmektedir(Fung ve ark.,2009). Hata yayılım algoritmaları ile ayrıca orijinal görüntülerde bulunan ince ayrıntıları ve görsel olarak tanımlanabilir yapıları korurken görsel olarak güzel, hoş yarı ton görüntüler elde edilebilir(Bakshi ve ark.,2018).

Hata dağılım algoritmaları gri ve bitmap (binary:iki renkli) görüntüler üzerinde de uygulanabilir. İki renkli(siyah-beyaz) görüntüler üzerinde sonuçlar orijinal görüntüyle daha kolay kıyaslanabilir. Bir pikselin sekiz tane komşu pikseli vardır. Hata dağılım algoritmasında temeli komşu piksellere hatanın dağıtılmasıdır. Ancak bu dağılım belirli katsayı oranında dağıtılmasıyla gerçekleşir. Sonuçta hedef pikselin renginde değişim olur(Gupta ve ark., 2009; Panda ve ark., 2015). Gerçekleştirdiğimiz bu çalışmada görüntü üzerinde bulunan karakterlerin daha belirgin olması amaçlanmıştır. Bu algoritmalar ilk olarak Floyd ve Steinberg tarafından önerilmiştir.

Floyd Steinberg hata yayılım algoritması

En yaygın kullanılan hata dağılım yöntemi Floyd ve Steinberg algoritmasıdır. Sağ komşu 1 piksel ve alttaki 3 piksel hedef pikselin değeri oluşturur. Şekil 1 üzerinde hata dağılım matris ve görüntü üzerindeki etkisi gösterilmiştir(Floyd ve ark., 1976; Caca Labs,2020).

	Hedef Piksel	7/16
3/16	5/16	1/16


a) b) c)

Şekil 1. a-)Yayılım Matrisi, b) Yalın Görüntü ve c)İşlenmiş Görüntü Sonuç

Jarvis Judice ve Ninke (JaJuNi) hata yayılım algoritması

Bu algoritma, Floyd ve Steinberg ile aynı zamanda yayınlanmıştır. Çok daha karmaşık bir hata dağılım matrisi kullanmaktadır. Hedef pikselin sağ tarafından 2 piksel ve altındaki toplam 10 piksel, hedef pikselin değeri oluşturur. Şekil 2 üzerinde hata dağılım matrisi ve görüntü üzerindeki etkisi görülmektedir(Caca Labs,2020;Jarvis ve ark.,1976).

		Hedef Piksel	7/48	5/48
1/16	5/48	7/48	5/48	1/16
1/48	1/16	5/48	1/16	1/48


a) b) c)

Şekil 2. a-)Yayılım Matrisi, b) Yalın Görüntü ve c)İşlenmiş Görüntü Sonuç

Atkinson hata yayılım algoritması

Atkinson algoritması, hatanın sadece% 75'ini yayabilir, bu da çok karanlık ve çok açık alanlarda, kontrast(karşıtlık, zıtlık) kaybına neden olur. Ancak bu algoritma orta tonlarda daha iyi karşıtlık oluşturur. Hedef pikselin sağ tarafından 2 piksel ve alttaki toplam 4 piksel, hedef pikselin değeri oluşturur. Şekil 3 üzerinde hata dağılım matrisi ve görüntü üzerindeki etkisi görülmektedir(Caca Labs,2020).

	Hedef Piksel	1/8	1/8
1/8	1/8	1/8	
	1/8		


a) b) c)

Şekil 3. a-)Yayılım Matrisi, b) Yalın Görüntü ve c)İşlenmiş Görüntü Sonuç

Shaiu-Fan hata yayılım algoritması

Shaiu-Fan algoritması, Floyd-Steinberg'de işlenen görüntülerden daha az yayılım gösteren bir algoritmadır. Sağ taraftan 1 piksel, sol ve alttan 3 hedef pikselin değeri oluşturur. Şekil 4 üzerinde hata dağılım matrisi ve görüntü üzerindeki etkisi gösterilmektedir(Caca Labs,2020; Shiau ve ark.,1994).

		Hedef Piksel	1/2
1/8	1/8	1/4	

a) b) c)

Şekil 4. a-)Yayılım Matrisi, b) Yalın Görüntü ve c)İşlenmiş Görüntü Sonuç

Stucki hata yayılım algoritması

Stucki algoritması, Jarvis-Judice-Ninke(JaJuNi) hata dağılım algoritmasının daha hafif bir varyasyonudur. Bu algoritmada da sağ taraftan 2 piksel, alt taraftan 10 piksel hedef pikselin değerini oluşturur. Şekil 5 üzerinde hata dağılım matrisi ve görüntü üzerindeki görülmektedir(Caca Labs,2020; Stucki,2012).

		Hedef Piksel	4/21	1/21
1/21	2/21	4/21	2/21	1/21
1/42	1/21	2/21	1/21	1/42

a) b) c)

Şekil 5. a-)Yayılım Matrisi, b) Yalın Görüntü ve c)İşlenmiş Görüntü Sonuç

Frankie Sierra hata yayılım algoritması

Frankie Sierra hata dağılım algoritması, daha az piksele yayıldığı için biraz daha hızlı varyasyondur. Hedef pikselin değeri sağ taraftaki piksel, alttaki ve sol alt köşedeki pikseller kullanılarak hesaplanır. Şekil 6 üzerinde dağılım matrisi ve görüntü üzerindeki etkisi görülmektedir(Caca Labs,2020).

	Hedef Piksel	1/2
1/4	1/4	

a) b) c)

Şekil 6. a-)Yayılım Matrisi, b) Yalın Görüntü ve c)İşlenmiş Görüntü Sonuç

BULGULAR ve TARTIŞMA

Yapılan çalışmada gerçekleştirilen yazılım içerisinde karakter barındıran görüntüler üzerinde ayrı ayrı uygulanmış ve çizelge 1 değerleri elde edilmiştir. Çizelge 1 görüldüğü üzere kullanılan algoritmalar, yazı stilleri(Times New Roman, Arial, Calibri), yazı biçimleri(Normal, Kalın, Eğik) ve karakterlerin özellikleri(Büyük Harf, Küçük Harf ve Rakamlar) alanları mevcuttur. Ayrıca algoritmalar uygulandığında tespit performanslarını gösteren başarı(%) alanı mevcuttur. Başarı alanı yazı biçimlerinin sağ tarafında olup sonuçları göstermektedir. Test için 26 tane büyük harfli İngilizce alfabe karakteri, 26 tane küçük harfli İngilizce alfabe karakteri ve 10 tane rakam(nümerik) değerin okunması başarı oranı tespit edilmiştir.

Elde edilen sonuçlarda en çok "Q" karakteri "O" karakteriyle, "B" karakteri "8" rakamıyla, "0" rakamı "O" karakteriyle ve "1" rakamı "I" karakteriyle bazı durumlarda da "u" karakteri "v" karakteriyle karıştırıldığı ve yanlış okuma yapıldığı görülmüştür.

Kullanılan görüntü her yazı stili ve biçimi için ayrı ayrı oluşturulmuştur. Yani Times New Roman yazı stili için Normal, Kalın ve Eğik olarak, Calibri yazı stili için Normal, Kalın ve Eğik olarak ayrıca Arial yazı stili için Normal, Kalın ve Eğik örnekleri üzerinde belirtilen bütün algoritmalar uygulanmış ve sonuçları çizelgeye işlenmiştir.

Çizelgeden görüldüğü üzere görüntü üzerinde önce hata yayılım algoritması içermeyen Tesseract kütüphanesi hiçbir ön işlem yapılmadan kullanılmış elde kıyaslama amaçlı değerler kaydedilmiştir. Sonra diğer 6 hata yayılım algoritması, ön işlem olarak kullanılmış ve Tesseract kütüphanesi bu ön işlemden sonra uygulanmıştır.

Elde edilen değerlere göre bazı yazı biçimlerinde hata yayılım algoritmalarının yüksek oranda başarı sağladığı görülmüştür. Ayrıca en çok Times New Roman yazı stilinde başarı oranı düşük değer elde edilmiş ancak Calibri ile Arial yazı tiplerinde başarı oranı yüksek değerler elde edilmiştir.

Ön işlem uygulanmayan Tesseract kütüphanesinin Times New Roman yazı stilinin Kalın ve Eğik yazı biçimleri için %11,54 değeri gibi düşük başarı oranları elde edilmiştir. Ancak bu yazı biçimleri için Floyd Steinberg hata dağılım algoritması kullanıldığında %90 gibi yüksek başarı oranlarına ulaşılabilmiştir. Ayrıca Times New Roman yazı stili ve Kalın biçimleri için Frankie Sierra Matrisi ve Stucki hata yayılım algoritmaları daha yüksek doğrulukta tespit gerçekleştirmiştir.

Ancak bu algoritmaların aynı yazı stili için eğik yazı biçimi üzerindeki doğru tespit oranları daha düşüktür. Calibri yazı stili ve eğik yazı biçimli örnek üzerinde de Floyd Steinberg hata yayılım algoritmasının(ortalama 98,6 başarı oranıyla), ön işlem uygulanmayan Tesseract algoritmasından(ortalama %92,6 başarı oranı) daha iyi sonuç verdiği görülmektedir.

Rakamlar üzerindeki yüksek doğruluk tespit oranı, bazı hata yayılım algoritmalarında, özellikle Calibri yazı stili ve eğikyazı biçimli testler üzerinde görülmektedir. Rakamla üzerinde en çok karşılaşılan hatalar "1" rakamıyla "0" rakamının yanlış okunması durumlarıdır. Özellikle Times New Roman Yazı stilinde en çok bu hatalarla karşılaşılmıştır.

Shaiu Fan Matrisi algoritması hata yayılım algoritmaları içinde değerlendirme yapıldığında en başarısız algoritma sonucuna ulaşılabilir. Çünkü özellikle Normal yani özerinde hiçbir biçimlendirme kullanılmayan örneklerinde ve eğik biçimlendirme kullanılan örneklerde okuma yapılamamıştır.

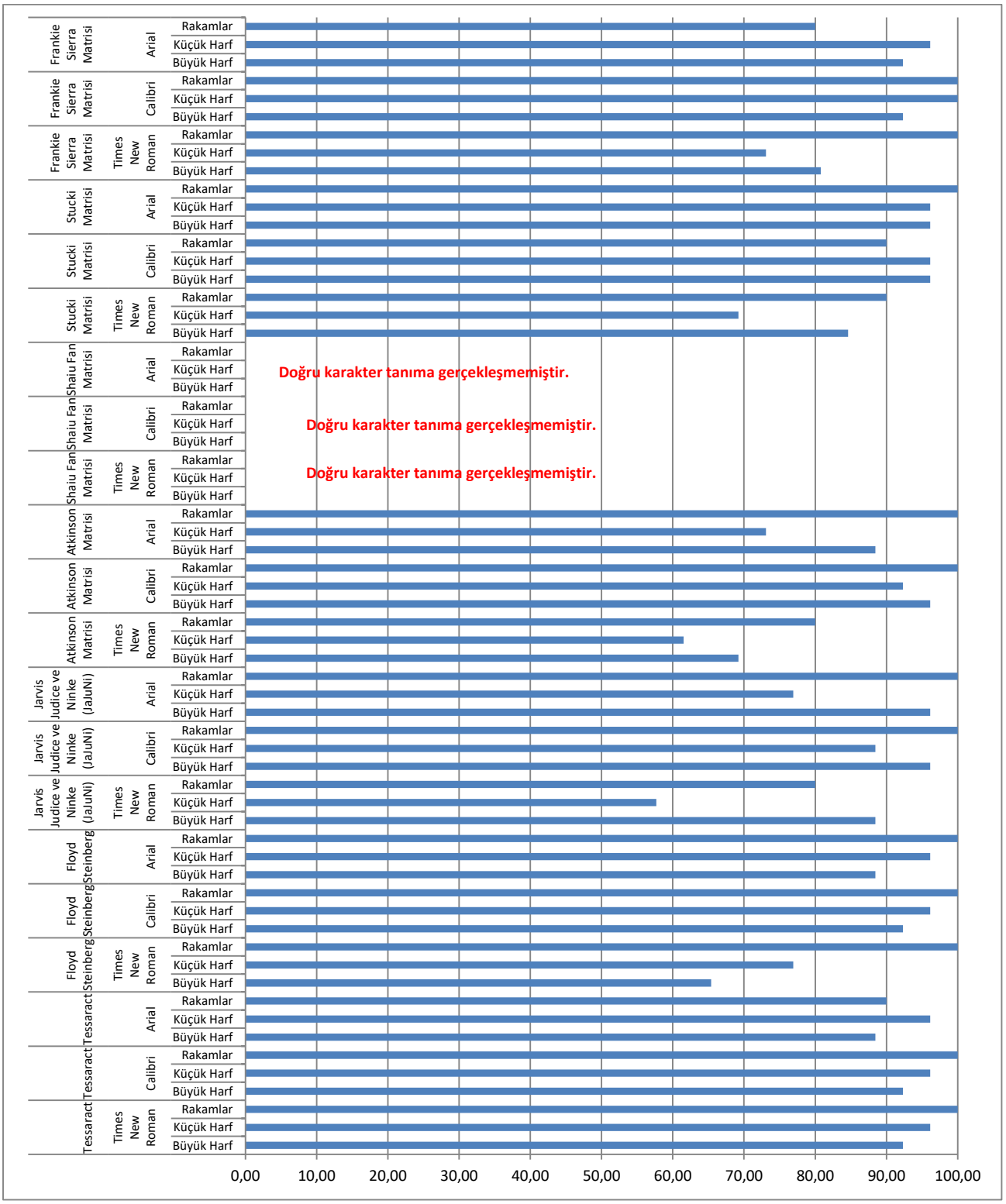
Çizelgeden elde edilen değerlere göre; Floyd Steinberg hata yayılım algoritması, Times New Roman, Calibri ve Arial yazı stillerinin bütün biçimleriyle(Normal, Kalın, Eğik), büyük harfli karakter, küçük harfli karakter ve rakam okumada iyi sonuç vermiştir. Özellikle Times New Roman stilinde, ortalama %74 başarı oranı ile yalın Tesseract algoritmasından (ortalama %64 başarı oranı) daha yüksek oranda başarılı okuma gerçekleştirilmiştir.

Çizelge 1. Karakter ve rakamlar üzerinde hata yayılım algoritmalarının Tesseract Kütüphanesi üzerindeki performans kıyaslaması

Algoritma Adı	Örnek Sayısı	OCR Okuma Performans Değerleri						Karakter Türleri	Yazı Stilleri
		Normal	Başarı(%)	Kalın	Başarı(%)	Eğik	Başarı(%)		
Tessaract	26	24	92,31	14	53,85	21	80,77	Büyük Harf	Times New Roman
	26	25	96,15	18	69,23	3	11,54	Küçük Harf	
	10	10	100,00	5	50,00	3	30,00	Rakamlar	
Tessaract	26	24	92,31	24	92,31	25	96,15	Büyük Harf	Calibri
	26	25	96,15	24	92,31	24	92,31	Küçük Harf	
	10	10	100,00	10	100,00	9	90,00	Rakamlar	
Tessaract	26	23	88,46	25	96,15	25	96,15	Büyük Harf	Arial
	26	25	96,15	26	100,00	26	100,00	Küçük Harf	
	10	9	90,00	9	90,00	10	100,00	Rakamlar	
Floyd Steinberg	26	17	65,38	18	69,23	17	65,38	Büyük Harf	Times New Roman
	26	20	76,92	19	73,08	17	65,38	Küçük Harf	
	10	10	100,00	9	90,00	6	60,00	Rakamlar	
Floyd Steinberg	26	24	92,31	16	61,54	25	96,15	Büyük Harf	Calibri
	26	25	96,15	23	88,46	26	100,00	Küçük Harf	
	10	10	100,00	10	100,00	10	100,00	Rakamlar	
Floyd Steinberg	26	23	88,46	25	96,15	25	96,15	Büyük Harf	Arial
	26	25	96,15	25	96,15	24	92,31	Küçük Harf	
	10	10	100,00	8	80,00	10	100,00	Rakamlar	
Jarvis Judice ve Ninke (JaJuNi)	26	23	88,46	16	61,54	0	0,00	Büyük Harf	Times New Roman
	26	15	57,69	2	7,69	0	0,00	Küçük Harf	
	10	8	80,00	0	0,00	0	0,00	Rakamlar	
Jarvis Judice ve Ninke (JaJuNi)	26	25	96,15	6	23,08	23	88,46	Büyük Harf	Calibri
	26	23	88,46	22	84,62	18	69,23	Küçük Harf	
	10	10	100,00	10	100,00	6	60,00	Rakamlar	
Jarvis Judice ve Ninke (JaJuNi)	26	25	96,15	25	96,15	25	96,15	Büyük Harf	Arial
	26	20	76,92	25	96,15	20	76,92	Küçük Harf	
	10	10	100,00	10	100,00	10	100,00	Rakamlar	
Atkinson	26	18	69,23	16	61,54	0	0,00	Büyük Harf	Times New Roman
	26	16	61,54	13	50,00	0	0,00	Küçük Harf	
	10	8	80,00	0	0,00	0	0,00	Rakamlar	
Atkinson	26	25	96,15	25	96,15	23	88,46	Büyük Harf	Calibri
	26	24	92,31	24	92,31	22	84,62	Küçük Harf	
	10	10	100,00	10	100,00	10	100,00	Rakamlar	
Atkinson	26	23	88,46	24	92,31	25	96,15	Büyük Harf	Arial
	26	19	73,08	25	96,15	24	92,31	Küçük Harf	
	10	10	100,00	10	100,00	10	100,00	Rakamlar	
Shaiu Fan Matrisi	26	0	0,00	5	19,23	0	0,00	Büyük Harf	Times New Roman
	26	0	0,00	7	26,92	0	0,00	Küçük Harf	
	10	0	0,00	1	10,00	0	0,00	Rakamlar	
Shaiu Fan Matrisi	26	0	0,00	25	96,15	0	0,00	Büyük Harf	Calibri
	26	0	0,00	25	96,15	0	0,00	Küçük Harf	
	10	0	0,00	10	100,00	0	0,00	Rakamlar	
Shaiu Fan Matrisi	26	0	0,00	25	96,15	0	0,00	Büyük Harf	Arial
	26	0	0,00	26	100,00	0	0,00	Küçük Harf	
	10	0	0,00	10	100,00	0	0,00	Rakamlar	
Stucki	26	22	84,62	23	88,46	0	0,00	Büyük Harf	Times New Roman
	26	18	69,23	18	69,23	0	0,00	Küçük Harf	
	10	9	90,00	8	80,00	0	0,00	Rakamlar	
Stucki	26	25	96,15	23	88,46	25	96,15	Büyük Harf	Calibri
	26	25	96,15	23	88,46	22	84,62	Küçük Harf	
	10	9	90,00	10	100,00	9	90,00	Rakamlar	
Stucki	26	25	96,15	25	96,15	24	92,31	Büyük Harf	Arial
	26	25	96,15	25	96,15	22	84,62	Küçük Harf	
	10	10	100,00	9	90,00	10	100,00	Rakamlar	
Frankie Sierra Matrisi	26	21	80,77	22	84,62	14	53,85	Büyük Harf	Times New Roman
	26	19	73,08	16	61,54	0	0,00	Küçük Harf	
	10	10	100,00	8	80,00	4	40,00	Rakamlar	
Frankie Sierra Matrisi	26	24	92,31	16	61,54	22	84,62	Büyük Harf	Calibri
	26	26	100,00	24	92,31	23	88,46	Küçük Harf	
	10	10	100,00	10	100,00	8	80,00	Rakamlar	
Frankie Sierra Matrisi	26	24	92,31	25	96,15	25	96,15	Büyük Harf	Arial
	26	25	96,15	22	84,62	24	92,31	Küçük Harf	
	10	8	80,00	10	100,00	10	100,00	Rakamlar	

Elde edilen değerlere göre, normal yazı biçimi kriteri dikkate alınarak, ön işlem olmayan Tesseract ve diğer hata yayılım algoritmalarının bütün yazı stilleri üzerindeki kıyaslama grafiği şekil 7

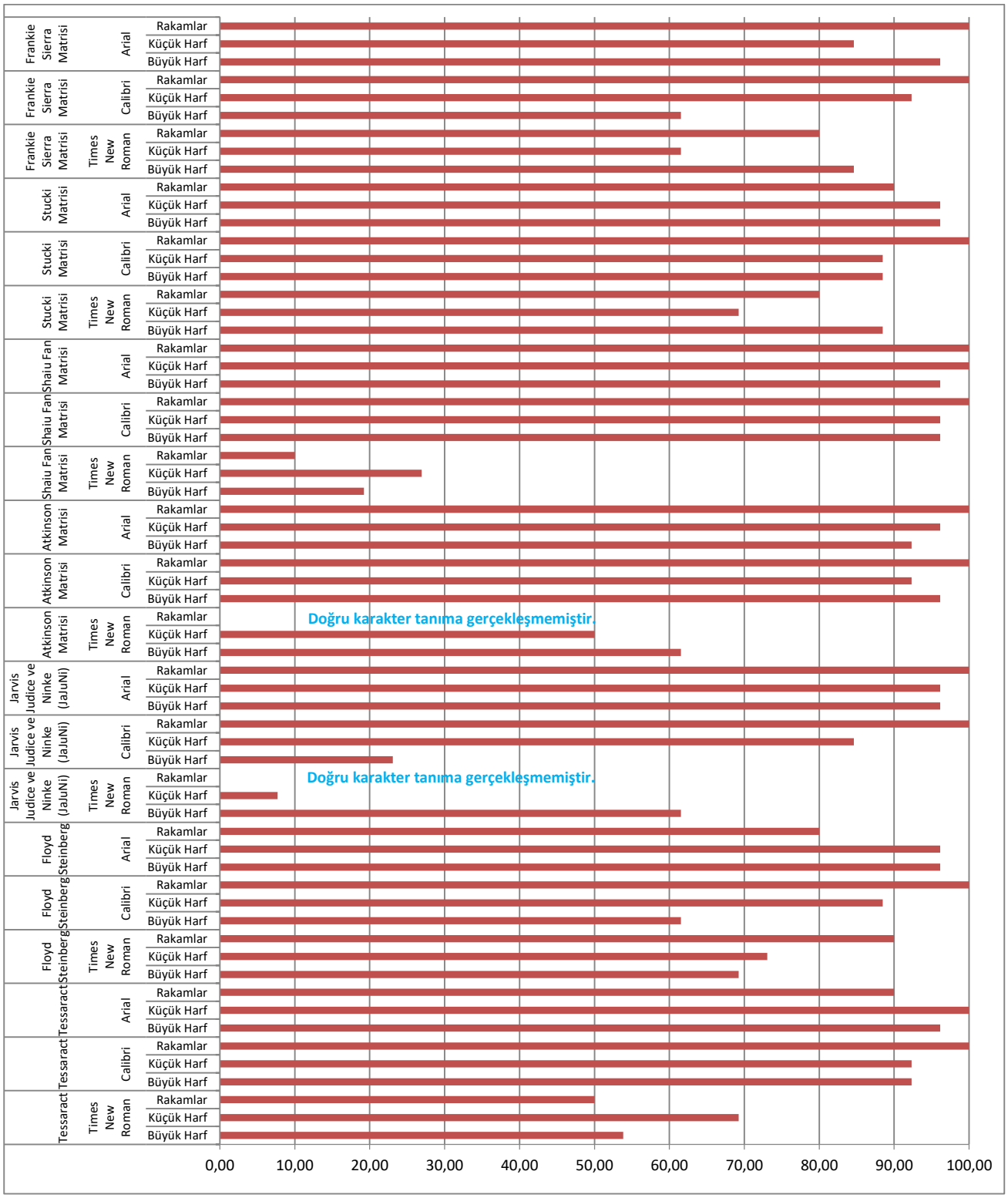
üzerinde verilmiştir. Normal yazı biçiminde Shaiu- Fan hata yayılım algoritması ön işlemiyle hiçbir doğru değer okuması gerçekleştirilememiştir. Ancak diğer algoritmaların başarı oranları birbirlerine yakındır. Ancak Arial yazı stili örnekte hata algoritmalarıyla daha yüksek oranları elde edilmiştir.



Şekil 7.Normal yazı biçimi üzerinde OCR okuma performans grafiği

Elde edilen değerlere göre, kalın yazı biçimi kriteri dikkate alınarak, ön işlem olmayan Tesseract ve diğer hata yayılım algoritmalarının bütün yazı stilleri üzerindeki kıyaslama grafiği şekil 8 üzerinde verilmiştir. Kalın yazı biçiminde Atkinson ve JaJuNi hata yayılım algoritmalarıyla ön işlem uygulanarak

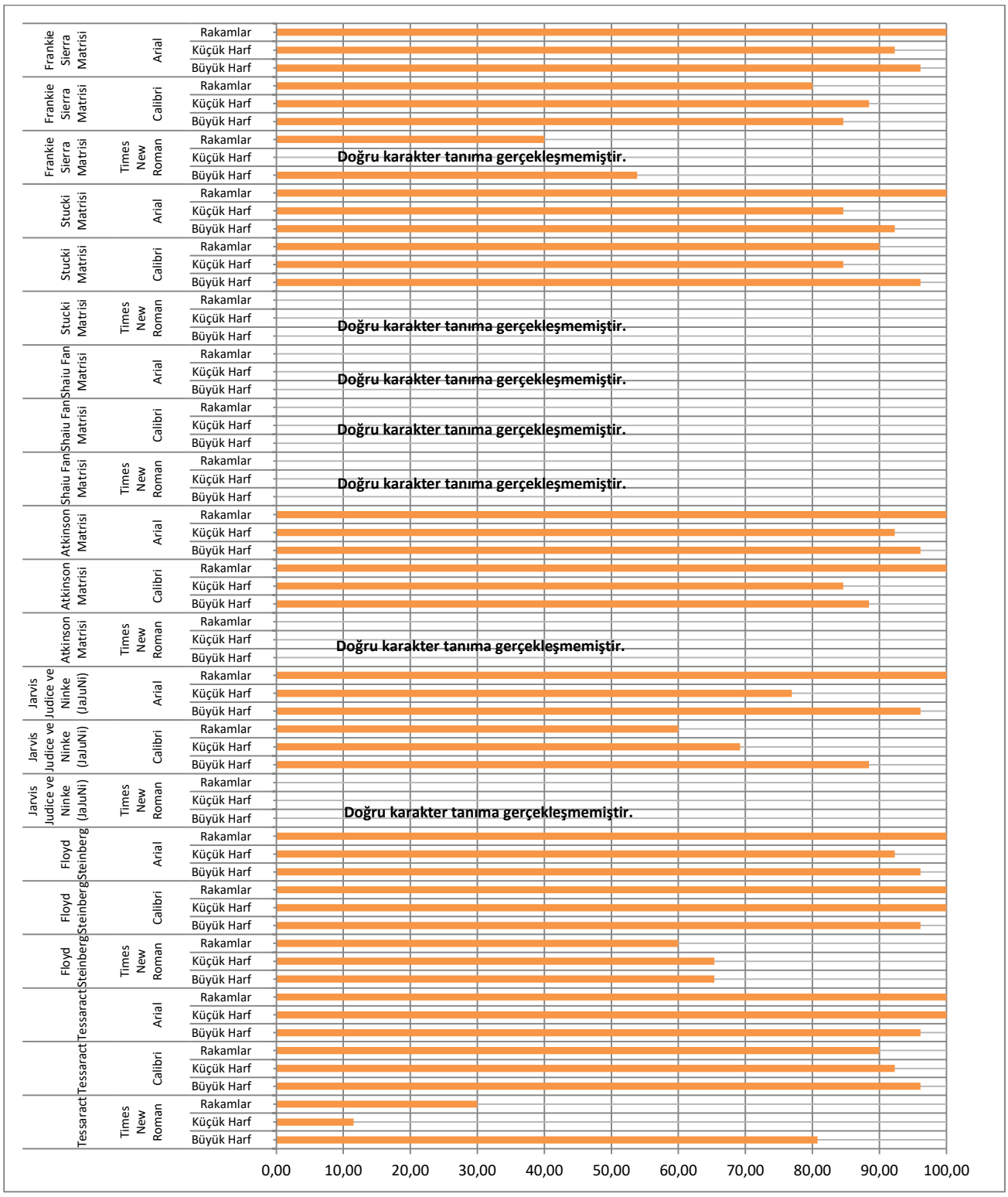
Rakamlar üzerinde hiçbir doğru değer okuması gerçekleştirilememiştir. Ön işlem uygulanmayan Tesseract algoritmasının özellikle Times New Roman ve Calibri yazı stilinde başarı oranın diğer algoritmalardan çok düşüktür.



Şekil 8. Kalın Yazı Biçimi Üzerinde OCR Okuma Performans Grafiği

Ayrıca elde edilen değerlere göre, eğik yazı biçimi kriteri dikkate alınarak da algoritmalar bütün yazı stilleri üzerindeki değerlendirmesi yapılmış kıyaslama grafiği şekil 9 üzerinde verilmiştir. Genel

olarak başarı oranı düşüktür ancak Floyd Steinberg yüksek başarı oranına sahiptir. Eğik yazı biçiminde Shaiu-Fan hata yayılım algoritmasının, bütün yazı stillerinde hiçbir tespit yapamadığı görülmüştür.



Şekil 9. Eğik Yazı Biçimi Üzerinde OCR Okuma Performans Grafiği

SONUÇ

Optik karakter okuma (OCR) ihtiyacı günümüzde; birçok alanda kağıt ya da görüntü üzerindeki karakterlerin elektronik ortama yani tekrar düzenlenebilir ortama dönüştürmek için artmaktadır. Bu tür

belge üzerindeki verileri tekrar elektronik ortama aktarabilmek eğer insan eli ile yapılırsa hem zaman almakta hem maliyeti yükseltmektedir. Ayrıca insan faktörü olduğundan hata oranı da yüksek olmaktadır. Bu durumu kolaylaştırmak için birçok uygulamalar vardır. Bu uygulamaların bazıları açık kaynak kodlu bazıları ise kapalı kaynak kodludur. Açık kaynak kodlu uygulamalarda, dünyanın her tarafından gönüllü yazılımcı ve kullanıcılar kaynak kodlara erişebildiklerinden, uygulamanın daha iyi çalışmasına katkı sağlayacak güncelleme yaparak geliştirebilmektedirler. Ancak kapalı kaynak kodlu uygulamalarda, kaynak kodlara hiçbir şekilde erişim olmadığından, sadece uygulamanın sahibi olan firmanın çalışanları, yazılım güncelleme yaparak geliştirmektedir.

Günümüzde bilgisayar cihazlarının yanı sıra mobil cihazlar üzerinde uyumlu yazılımlara ihtiyaç vardır. Yapılan çalışmalarda bu durumun dikkate alınması yaygın kullanım oranı oluşturacaktır. İşlenecek dökümanın görüntüsünün sayısal ortama kaydedilmesi sonra buradan karakter tanıma yazılımlarının uygulanması gerekir. Bu işlem kamera veya tarayıcı makineleri yardımıyla gerçekleşir. Sadece belge değil aynı zamanda kimlik kartları, pasaport veya kartvizitler üzerindeki bilgilerin okuma ihtiyacı olabilmektedir. Özellikle güvenlik noktalarında kimlik kartlarındaki karakteri okumaya ihtiyaç olabilmektedir.

Bilgisayar ile oluşturulmuş dokümanlardan okuma oranı el yazısıyla oluşturulmuş dokümanlardan daha yüksektir. Ancak yazı stillerine göre veya yazı biçimlerine göre değişebilmektedir. Ancak %100 doğruluk oranına ulaşmak zor olabilmektedir. Bunu geliştirmek için yazılımcılar makine öğrenmesi yöntemleriyle eğitim setleri kullanarak daha çok örnek kullanarak çaba sarf etmektedir.

Hata yayılım algoritmaları görüntü işlemede önemli bir paya sahiptir. Özellikle görüntü üzerinde bütünlüğü bozan, gerçek görüntüyle ilgisi olmayan istenmeyen piksellerin temizlenmesinde kullanılmaktadır. Bu işlemlerde komşu piksellerin değerlerinden yararlanılarak hedef pikselin değeri hesaplanmaktadır. Pikseller görüntü içinde kontrol edilebilen en küçük yapı taşı olduğundan görüntüyü oluşturan bütün pikseller üzerinde bu işlemler gerçekleştirilebilir.

Bu çalışmada optik karakter tanıma uygulamalarından ve yaygın kullanım oranına sahip olan Tesseract kütüphanesinin başarı oranını arttırmaya yönelik araştırma yapılmış ve hata yayılım algoritmalarını olumlu yönde katkısı tespit edilmiştir. Kullanıcıların en çok tercih ettiği üç yazı stili ve üç yazı biçimi üzerinde testler gerçekleştirilmiştir.

Elde edilen değerlere göre kalın yazı biçiminde hata yayılım algoritmalarının yüksek başarı sağladıkları ayrıca eğik yazı tipinde de hata yayılım algoritmalarından bazılarının daha yüksek başarı sağladıkları görülmüştür. Kalın yazı biçimi kullanılan Times New Roman yazı stilinde, yalın Tesseract kütüphanesinin karakter tanıma oranının çok düşük olduğu ancak hata yayılım algoritmalarının optik karakter tanıma oranını yükselttiği gözlemlenmiştir. Bu çalışmada, yirmi dört yazı büyüklüğü(punto) seçilerek İngilizce alfabe değerleri kullanılmıştır. Çünkü günümüzde kimlik ve pasaportlarda İngilizce karakterler kullanılmaktadır. Burada uluslararası karakterler ve rakamlar(nümerik) değerler üzerinde optik karakter okuma başarı oranı tespit edilmiştir. Bundan sonra kullanıcılar, elde edilen sonuçları dikkate alarak, en çok karşılaşılan hatalı değerleri tekrar kontrol edip doğruluk testi yapabilecektir.

Elde edilen değerlere göre; Floyd Steinberg hata dağılım algoritması daha doğru okuma sonucu oluşturmuştur. Bu sonuçlar bize Tesseract kütüphanesinin kullanımından önce hata yayılım algoritmalarının kullanımının yararlı olacağını, doğru karakter tespitinde katkı sağlayacağını göstermektedir.

Bu aşamadan sonra; başka yazı stilleri ve yazı boyutları kullanılarak farklı diller üzerinde deneylerin yapılması ve başarı oranlarını tespit edilmesi gerçekleştirilebilir. Tesseract kütüphanesi birçok dili desteklediğinden örnek sayısının arttırılmasına zaten imkanı tanımaktadır. Böylece doğruluk oranı daha da kesinlik kazanabilecektir.

KAYNAKLAR

ABBYY, 2020. Abbyy Ürünleri, https://www.abbyy.com/tr-tr/finereader/ (Erişim Tarihi:30.03 2020).

Bakshi A, Patel AK, 2018. A Novel Error Diffusion Algorithm for Halftoning Greyscale Image Using Pull Based Method. International Conference on Communication and Signal Processing (ICCSP'18), 3-5 April 2018, Chennai.

Caca Labs, 2013. Error Diffussion, http://caca.zoy.org/study/part3.html (Erişim Tarihi:20.03.2020).

Floyd RW, Steinberg L, 1976. An Adaptive Algorithm for Spatial Grey Scale. Proceedings of the Society of Information Display, 17(2): 75-77.

Fung YH, Chan YH, 2009. A Multiscale Error Diffusion Algorithm for Green Noise Digital Halftoning. 17th European Signal Processing Conference (EUSIPCO 2009), August 24-28 2009, Glasgow.

Gider Ç, Albayrak SV, 2018. Identifing of Alphanumerical Codes in Promotional products by Using of Deep Neural Network. 2018 3rd International Conference on Computer Science and Engineering (UBMK 2018),20-23 Sept. 2018, Sarajevo.

GitHub Inc, 2020. Tesseract OCR,https://github.com/tesseract-ocr/tesseract (Erişim Tarihi:05.05.2020).

Gupta A, Khandelwal V, Agarwal N, Gupta A, 2009. Five Neighbor Stochastic Error Diffusion for Digital Halftoning, 2nd IEEE International Conference on Computer Science and Information Technology, 8-11 Aug. 2009, Beijing.

GOCR, 2020. Open Source Character Recognition, https://www-e.ovgu.de/jschulen/ocr/index.html (Erişim Tarihi:30.03 2020).

Jarvis JF, Judice CN, Ninke WH, 1976. A Survey of Techniques for the Display of Continuous Tone Pictures on Bi-level Displays. Computer Graphics and Image Processing, 5(1): 13–40.

Koyun A, Afşin E, 2017. Derin Öğrenme ile İki Boyutlu Optik Karakter Tanıma, Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 10(2):11-14.

Kutlu H, Yazan E, 2014. OCR API'leri ve Gerek Zamanlı OCR Uygulaması. XVI. Akademik Bilişim Konferans (AB 2014), 5-7 Şubat 2014, Mersin.

Misra I, Deshpande A, Narayanan PJ, 2011. Hybrid Implementation of Error Diffusion Dithering. 18th International Conference on High Performance Computing, 18-21 December 2011, Bangalore.

Mithe R, Indalkar S, Divekar N, 2013. Optical Character Recognition, International Journal of Recent Technology and Engineering, 2(1):72-75.

Oki Electric Industry, 2020. OKI opne up your Dreams, https://www.oki.com(Erişim Tarihi:30.03.2020).

Panda NR, Sahoo AK, Kumar S, 2015. A Negative Multiscale Error Diffusion Technique for Digital Halftoning. 2015 International Conference on Pervasive Computing (ICPC), 8-10 Jan. 2015, Pune.

Patel C, Patel A, Patel D, 2012. Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study, International Journal of Computer Applications, 55(2):50-56.

Saray T, Çetinkaya A, 2017. Okatan A. Optik Karakter Tanma Yöntemi ile Otomatik Tabela Okuyucu. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı(UBMK'17), 5-8 October 2017, Antalya.

Shiau J, Fan Z, 1994. Method for Quantization Gray Level Pixel Data with Extended Distribution Set. U.S. patent Xerox Corporation, No:5,353,127,s.1-7, Stamford, US.

Smith R, 2007. An Overview of the Tesseract OCR Engine. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September 2007, Parana.

Stucki P, 2012. MECCA- a multiple error correcting computation algorithm for bi-level image hard copy reproduction. Research report RZ1060, IBM Research Laboratory, Zurich, Switzerland.

Tong X, Evans DA, 1996. A Statistical Approach to Automatic OCR Error Correction in Context. Fourth Workshop on Very Large Corpora(WVLC-96), 5-9 August 1996, Copenhagen..