Optik Karakter Tanıma Metinlerini Kullanarak Görüntülerden Tablo Verilerini Ayıklama
Extracting Table Data from Images Using Optical Character Recognition Text

Mehmet Yasin AKPINAR, Erdem EMEKLİGİL, Seçil ARSLAN
AR-GE ve Özel Projeler
Yapı Kredi Teknoloji A.Ş.
İstanbul, TÜRKİYE
{mehmetyasin.akpinar, erdem.emekligil, secil.arslan}@ykteknoloji.com.tr

***Özetçe* —Görüntü halindeki dokümanların dijital ve işlenebilir formlara çevrilmesi günümüzde optik karakter tanıma (OCR) araçlarıyla oldukça başarılı bir şekilde yapılabilmektedir. Ancak, orijinal belge üzerindeki biçimin korunması konusunda hala problemler mevcuttur. Bu problemlerden önemli bir tanesi ise tablo halindeki verinin okunmasıdır. Bu bildiride baskılı formundan taranarak dijital ortama aktarılmış belgeler üzerindeki tablo içeriklerinin, bir OCR aracı ile okunarak karakter pozisyonlarının da yardımıyla tekrar tablo formuna getirilerek saklanmasını sağlayan bir yöntem önerilmektedir. Yöntemin başarımı tespit edilen satır ve sütun sayılarıyla ölçülmüş olup, ticari olarak satılmakta olan başka ürünlerle kıyaslanarak sunulmuştur.**

Anahtar Kelimeler*—*Tablo Tanıma, Optik Karakter Tanıma, Metin İşleme.

***Abstract*—The conversion of image-based documents into digital and processible forms can be accomplished quite successfully with optical character recognition (OCR) tools. However, there are still problems with preserving the format on the original document. An important one of these problems is the reading of the tabular data. In this paper, a method is proposed in which the tabular data contents of hard-copy documents is extracted from the text and character positions which are obtained from an OCR tool and transferred to digital forms. The performance of the method is measured by the number of detected rows and columns and presented with the results of other commercial products.**

Keywords*—*Table Recognition, Optical Character Recognition, Text Processing.

I. GİRİŞ

Basılı dokümanlardan bilgi çıkarma zaman zaman herkesin ihtiyacı olan bir durumdur. Bu sebeple optik karakter tanıma (OCR) araçları oldukça faydalı kullanımlar sunmaktadır. Ancak, bazı durumlarda bu araçlar yetersiz kalabilmektedir. Basılı bir belgede yer alan bir tablonun dijital ortama formu korunarak aktarılması da bunun en güzel örneklerinden birisidir. Ticari ve açık-kaynak ürünlerde bu probleme çözüm üretilmeye çalışılsa da yapılan testlerde istenilen başarımın sağlanamadığı görülmüştür.

978-1-5386-1501-0/18/$31.00

Bu bildiride, bahsedilen problemi çözmek için bir yöntem önerilmektedir. Yöntem gerçeklenirken belgelerdeki tabloların bütün sayfa içeriğini kapladığı varsayılmıştır. Bir başka deyişle önerilen yöntem bütün sayfa metnini tek bir tabloya yerleştirmeye çalışmaktadır.

Bildirinin akışı ise şu şekildedir. II. bölümde tekniğin bilinen durumu kaynaklar ile birlikte özetlenmektedir. Ardından III. bölümde önerilen yöntem detaylarıyla beraber verilmektedir. IV. bölümde ise çalışmadan elde edilen sonuçlar hazır ticari ürünlerle kıyaslanarak sunulmaktadır. Bu kıyaslama aşamasında tespit edilen satır ve sütun sayıları dikkate alınmıştır. V. bölüm olan kapanış bölümünde ise bildirinin özeti yapılıp gelecek çalışmalardan bahsedilmiştir.

II. İLGİLİ ÇALIŞMALAR

Literatür taramasında karşılaşılan çalışmalar ana 2 gruba ayrılmaktadır. Birincisi, görüntü üzerinden metin bilgisi çıkarmaya yarayan optik karakter tanıma teknolojileri, ikincisi ise metin formatındaki dokümanlardan tablo oluşturma çalışmalarıdır.

Birinci gruba giren çalışmalara uzun yıllardır yoğun bir şekilde ulaşılmaktadır. Bu çalışmaların bir özetine [1] numaralı bildiriden ulaşılabilir. Ancak, son zamanlarda bu grubu giren çalışmalar form değiştirerek doğal ortamlarda metin tanıma [2] [3], el yazısı tanıma [4] [5] [6] ve gerçek zamanlı metin tanıma [7] [8] gibi alanlara yönelmiştir.

İkinci gruptaki çalışmalar ise yine aynı zamanlara denk gelmektedir. Ancak, bu konudaki çalışmalara daha seyrek rastlanmaktadır. 2003 yılında yayınlanan bir inceleme bildirisinde (survey) tekniğin o zamana kadarki durumu özetlenmektedir [9]. Son zamanlardaki çalışmalara ise [10] ve [11] verilebilir. Ayrıca, bu çalışmalarda da daha spesifik alanlara yönelmiş olanlar mevcuttur [12].

Bu iki yaklaşımı birleştirip görüntü üzerinden tablo verisi ayıklama çalışmaları ile pek sık olarak karşılaşılmamaktadır. [11]'de bu yönde bir yaklaşım da mevcuttur. Ayrıca, bu alandaki çalışmaların son örnekleri [13] ve [14]'te verilmiştir.

Birbirine benzer çalışmaların tekrar tekrar yapılmasının ana sebebi herkes için uygun bir yöntemin bulunamamasından kaynaklanmaktadır. Bu bildirideki çalışma da denenmiş olan açık kaynak kodlu ve ticari ürünlerden yeterli performans sağlanamadığı için gerekli hale gelmiştir.

III. YÖNTEM

Giriş bölümünde bahsedildiği gibi bu çalışmada bir dizi sıralama ve gruplama basamakları arka arkaya kullanılarak görüntü içerisinden metnin uygun bir formatta çekilmesi sağlanmıştır. Sistem girdi olarak görüntü formatından bir OCR uygulaması (ABBYY FineReader) ile okunmuş kelimeleri ve bu kelimelere ait pozisyon (sol, sağ, üst ve alt piksel değerleri) ve boyut (en ve boy) bilgilerini kullanmaktadır. Pozisyon ve boyut bilgileri hesaplanırken kelimelere ait karakterlerin OCR aracından elde edilen pozisyon bilgileri birleştirilmiştir. Bütün adımlardaki piksel değerleri için sayfanın sol-üst noktası başlangıç noktası kabul edilmektedir. Bu adımdan sonra problem bir akım (stream) işleme problemi haline gelmiştir. Uygulanan basamaklar kabaca aşağıdaki gibi listelenebilir:

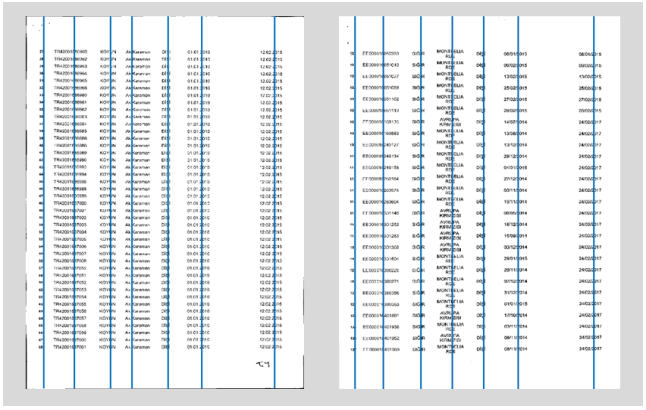
1) Sütunları tespit etme
2) Kelimeleri yukarıdan aşağıya doğru sıralama ve satırlara ayırma
3) Her satırdaki kelimeleri soldan sağa doğru sıralama
4) Kelimeleri sütunlara göre gruplama ve tablo yapısını oluşturma
5) Birden fazla satırdan oluşan hücreleri birleştirme (isteğe bağlı)

A. Sütun Tespiti

Önerilen yöntem gerçeklenirken öncelikle sütun tespiti üzerine çalışılmıştır. Bunun en önemli sebebi sütun sayısının satır sayısından daha az olması ve bu sebeple sayfa eğimlerinden daha az etkilenmesidir. sütunların tespit edilebilmesi için her kelimenin yatay eksendeki orta noktalarının piksel değerleri hesaplanarak bir histogram oluşturulmuştur. Bu histogramda kullanılan kutu (bin) sayısı empirik olarak sayfadaki sütun sayısı * 10 şeklinde belirlenmiştir.

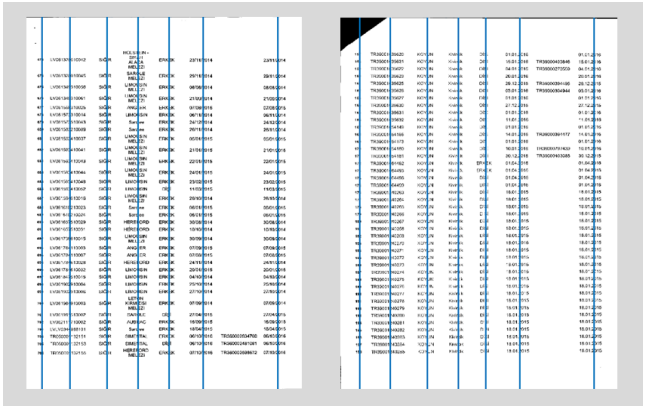
Sayfa kirliliği ya da format bozukluğundan meydana gelebilecek yanlış sütun tespitlerinin engellenebilmesi için belirli bir eşik değerin altındaki kutuların hiç dikkate alınmaması gerekmektedir. Bunun için de histogram içerisinde en yüksek değerin 1/4'ünden daha az değere sahip kutuların 0 olarak kabul edilmesi sağlanmıştır.

Histogram bilgisi bu şekilde elde edildikten sonra kelimelerin yoğunlaştığı kutuların bulunabilmesi için bir kayan pencere (sliding window) yapısından yararlanılmıştır. Bu adımda pencere sayısı, sayfadaki sütunların kutu sayısı bazında ortalama uzunlukları göz önünde bulundurularak belirlenmiştir. Test kümesinde yer alan sayfalarda kutu sayısı yukarıdaki formülle belirlendiği takdirde yaklaşık 5 kutuya denk gelmektedir ve bu sebeple pencere sayısı 5 olarak kararlaştırılmıştır. Hazırlanan pencere histogram datası üzerinde gezdirilerek pencere içerisinde kalan kutulardan en yüksek değere sahip olan not edilerek yeni bir histogram oluşturulmuştur. Bu uygulamanın amacı lokal olarak en yüksek değere sahip olan kutuların tespitini kolaylaştırmaktır. Böylece yeni elde edilen histogramda pencere sayısına eşit değerdeki kutular lokal maksimumları belirtmektedir ve sütunların orta noktaları bu kutular olarak kabul edilmiştir.



Şekil 1: Doğru Tespit Edilmiş Sütun Örnekleri

Şekil 1'de, yukarıda anlatılan yöntemle doğru olarak tespit edilmiş sütunlar gösterilmektedir. Tablo yapısını belirten herhangi bir çizgi ya da işaret bulunmadığı halde tüm sütunlar ufak hata paylarıyla (bir kutu genişliğinden daha az) doğru olacak şekilde tepit edilebilmiştir. Ancak, Şekil 2'deki gibi bir sütununda az bilgi içeren tablolarda önerilen yöntem bu sütunları tespit edemeyebilmektedir. Bu durumun en önemli sebebi sayfa kirliliği ya da format bozuklukları için alınan 1/4'lük önlemdir. Bu örneklerin doğru sonuç vermesi bu kontrolün kaldırılmasıyla mümkün olmasına rağmen, başka belgelerde daha kötü sonuçlara sebep olabilmektedir. Bu sebeple kontrolün bu şekilde kalması daha uygun görülmüştür.



Şekil 2: Hatalı Tespit Edilmiş Sütun Örnekleri

B. Kelimelerin Sıralanması ve Satırlara Ayrılması

Sayfa üzerindeki sütun pozisyonları tespit edildikten sonra kelimeler üst piksel değerleri göz önüne alınarak sıralanmıştır. Sonrasında kelimelerin üst ve alt piksel değerleri kullanılarak satır ayrımları yapılmıştır. Bu ayrım yapılırken sayfa eğikliğinin bir miktar tolere edilebilmesi için hep bir kelimenin alt piksel değeriyle kendisinden sonra gelen kelimenin üst piksel değeri kıyaslanmıştır. Bu değerler örtüşmediğinde, yani bir kelimenin alt piksel değeri kendisinden sonra gelen kelimenin üst piksel değerinden daha küçük olduğunda bu iki kelimenin

aynı satırda olmadığı kabul edilerek sonraki kelimenin yeni bir satıra yerleştirilmesi sağlanmıştır.

C. Satırlardaki Kelimelerin Sıralanması

Bu aşamada kelimelerin ait oldukları satırlar içerisinde soldan sağa sıralanmaları sağlanmıştır. Ancak, birden fazla satırdan oluşan hücrelerdeki sıralamanın bozulmaması için bu kontrol üst piksel değerlerini de içerecek şekilde düzenlenmiştir. Böylece alt alta olan kelimelerin sol piksel değerlerine bakılmaksızın üstte olanı daha önde yer alacak hale getirilmiştir.

D. Kelimelerin Sütunlara Ayrılması

Kelimeler sıralandıktan sonra sıralamalarına göre en yakın oldukları sütuna atanmıştır. Bu atamalar yapılırken de her kelime arasında bir boşluk olacak şekilde bağlama (concatenate) işlemi yapılmıştır. Bu işlem sonucunda elde edilen yapı, ulaşılmak istenen tablo yapısının ilk halidir ve çok düzgün sayfalarda, örneğin hiç bir hücresinde birden fazla satırlık bilgi bulunmayan tablolarda, yeterli seviyede başarım gösterebilmektedir. Ancak, çalışılan belgelerde sıklıkla bir hücre birden fazla satırlık değer alabildiği için tablo içeriğinde yalnızca bir sütunu dolu olan satırlar oluşmaktadır. Şekil 3'te bu duruma bir örnek gösterilmektedir. Tüm örnek tek satırlık bir bilgi içermesine rağmen yukarıdaki adımlar sonucunda 4.sütunda bulunan bilgi 3 satırlık yer kapladığı için bu bilgiler fazlalık satırlar olarak sonuçlanmaktadır. Bu örnekler için de opsiyonel olan E adımı uygulanarak birleşim sağlanabilmektedir.

Şekil 3: Hatalı Ayrılmış Satır Örneği

E. Satırların Birleştirilmesi

Bahsedilen durumun çözülebilmesi için satır birleştirici bir adım daha eklenmek durumunda kalınmıştır. Bu birleştirici, öncelikle tabloyu yukarıdan aşağıya doğru tarayarak dolu olan hücre sayılarına göre birleştirme yapılıp yapılmayacağına karar verir. Daha sonra ise tespit edilen az içeriğe sahip sütunların üst satıra mı yoksa alt satıra mı birleştirileceğini yakınlık durumuna göre hesaplar. Bunun sonucuna göre de ilgili hücreleri sırasına göre aralarda birer boşluk bırakacak şekilde bağlayarak (concatenate) birleştirme işlemini gerçekleştirir.

Satırların birleşip birleşmeyeceğinin kararının verilebilmesi için ise ardışık ikili satırların içerikleri incelenmiştir. İncelenen iki satırın aynı anda bilgi içeren sütun sayısı 2 veya daha az ise bu iki satırın aslında tek bir satırlık bilgi içerdiği varsayımı yapılmıştır. Yani Şekil 3'teki örneği ele alacak olursak, yalnızca 4.sütunda ardışık iki satırda birden bilgi bulunmaktadır. Diğer sütunlarda ya ilk satır ya da ikinci satır bilgi içermemektedir. Dolayısıyla bu üç satırın 2 adımda birleştirilmesi uygundur. Buradaki kontrolün 2 olarak belirlenmesinin sebebi belge kirliliğinin yol açabileceği karakter okumalarıdır. Eğer belge içeriğinde birden fazla satırlık bilgi içeren hücreler birden daha fazla sütunda bulunuyorsa, bu sayının bu tip sütun sayısı + 1 şeklinde belirlenmesi yerinde olacaktır.

IV. Sonuçlar ve Karşılaştırma

Bu bölümde önerilen sistemin başarımları ticari olarak satılmakta olan ürünlerle karşılaştırarak sunulmuştur. 58 belgelik test kümesi üzerinde doğruluk bilgilerinin çıkarılması için bir yorumcu (annotator) ile çalışılmıştır. Yorumcunun görevi test kümesi içerisindeki her belgeyi inceleyerek satır ve sütun sayılarının not edilmesi ile referans değerlerinin belirlemek olmuştur.

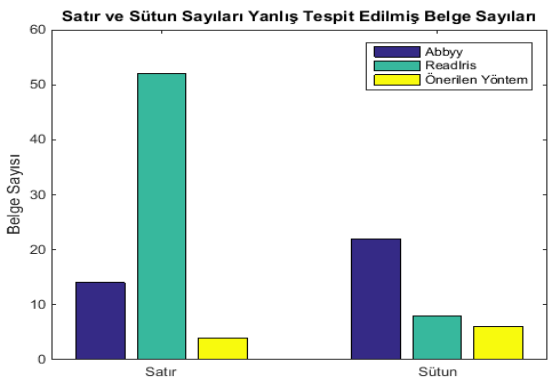
Ticari ürünlerden ilki, çıktıları önerilen yöntemde de kullanılanılan ABBYY FineReader 11 Release 8 versiyonudur. Bu ürün çalışmadan önce de kullanılmakta ve tablo okuma performansındaki problemler sebebiyle önerilen yöntemin ortaya çıkmasında önemli bir rol oynamaktadır. Bu ürünün en önemli problemleri sayfayı bloklar halinde okurken tabloyu da ikiye veya daha fazla sayıya bölerek okuması, bazen de hiç tablo tespit edememesidir.

Bu ürünün başarımı ölçülürken okunması istenen belgede tespit edebildiği en büyük tablo yapısı göz önüne alınmıştır. Örneğin bir sayfada 27x5 ve 26x2 boyutlarında tablo tespit ediyorsa 27 satır ve 5 sütun tespit edebildiği kabul edilmiştir.

Bir diğer ticari ürün de Readiris Pro 16 versiyonudur. Bu ürün görüntü formatındaki dosyaları okuyarak docx, pdf, xlsx ve bir kaç farklı formda daha kaydetme imkanı sağlamaktadır. Ürünün en büyük problemleri tablo yapısını oluştururken boş sütun ve satırlar bulması, biçim özelliklerini yeterince düzgün kopyalayamamasıdır. Ayrıca, bilgileri çok fazla ayırmaya giderek birden fazla satırdan oluşan hücreleri iyi tespit edememektedir.

Başarım ölçümleri için okunan belgeler xlsx formatında kaydedilmiş, sonrasında ise bu dosyalar açılarak tespit edilen satır ve sütun sayıları not edilmiştir. Bu aşamada boş satırlar dikkate alınmamış ve hesaplama dışında tutulmuştur.

Şekil 4'te her bir ürün ve önerilen yöntem için satır ve sütun sayısı yanlış tespit edilmiş belge sayıları sunulmaktadır. Bu sayılar hesaplanırken ilgili ürünün tespit ettiği satır ve sütun sayıları daha önce yorumcu tarafından belirlenmiş satır ve sütun sayılarıyla kıyaşlanmıştır. Bu sayılar eşit olmadığı durumda ilgili ürün için hatalı belge sayısı 1 artırılmıştır. Hem satır bazlı hem de sütun bazlı sonuçlar çıkarılarak görselleştirilmiştir.



Şekil 4: Satır Ve Sütun Sayıları Yanlış Tespit Edilen Belge Sayıları

Satır bazlı sonuçlarda Readiris en kötü performansı gösterirken (52/58 hatalı tespit), önerilen yöntem en iyi performansa sahiptir (4/58 hatalı tespit). Sütun bazlı sonuçlarda ise ABBYY ürünü bazı belgelerde birden fazla tablo tespit ettiği ve bu tablolar dikey olarak bölündüğü için en kötü performansı sergilemiştir (22/58 hatalı tespit). Bu hesaplamada da yine en iyi başarım önerilen yönteme aittir (6/58 hatalı tespit). Ayrıca, ABBYY ürünü 3 belgede hiç tablo tespit edememiştir.

Tablo I'de ise her ürün ve önerilen yöntem için test kümesi üzerinde tespit edilen satır ve sütun sayıları kümülatif olarak toplanarak, referans değerleriyle kıyaslanmıştır. Bu sonuçlar da yine satır ve sütun bazlı olarak ikiye ayrılmıştır.

Tablo I: Toplam Satır ve Sütun Tespitleri

	Satır			Sütun		
	Sayı	**Fark**	**Fark (%)**	**Sayı**	**Fark**	**Fark (%)**
Referans	1686	-	-	418	-	-
Abbyy	1554	-132	-7,83	365	-53	-12,68
ReadIris	2871	+1185	+70,28	423	+5	+1,20
Önerilen Yöntem	1677	-9	-0,53	411	-7	-1,67

Satır bazlı sonuçlarda Readiris ürünü satır birleştirme yapmadığı için oldukça kötü bir sonuç vermiştir ve yaklaşık %70 fazladan satır tespitinde bulunmuştur. ABBYY ürünü ise referans değerlerine göre %7,83'lük eksik tespit ortaya çıkarmıştır. Buna karşılık önerilen yöntem %0,53 fark ile önde kalmayı başarmıştır. Bu fark, tüm test kümesi üzerinde yalnızca 9 eksik satır tespitine tekabül etmektedir.

Sütun bazlı sonuçlarda ise daha önce bahsedilen problemlerden dolayı ABBYY ürünü en kötü sonucu vermiştir (%12,68 eksik tespit). Ancak, Readiris ürünü belge bazında önerilen yöntemden geride olmasına rağmen bu hesaplamada %1,20 fazla tespit ile en yüksek performansı göstermiştir. Önerilen yöntem ise %1,67 eksik tespit ile hemen arkasında yer almıştır.

Sonuç olarak hem satır hem de sütun bazlı kıyaslamalar birlikte göz önüne alındığında önerilen yöntem bu iki ticari ürüne göre bariz bir üstünlük sağlayabilmektedir. Bu durumun önemli sebeplerinden bir tanesi, önerilen yöntemin hedef odaklı olup bütün OCR metnini bir tabloya çevirmeye çalışmasıdır. Diğer ürünlerde böyle bir durum söz konusu değildir.

V. Kapanış

Bu bildiride tablo yapısında bilgi içeren basılı örneklerden bilgilerin formu korunarak çıkarılmasını sağlayan bir yöntem önerilmiştir. Giriş bölümünde yapılan çalışmanın genel amacı açıklanıp, bildiri akışından bahsedilmiştir. Ardından İlgili Çalışmalar bölümünde tekniğin bilinen durumuna değinilmiştir. 3.bölüm olan Yöntem bölümünde yapılan çalışma detaylarıyla açıklanmıştır. Bu yöntem ve mevcutta bulunan ticari ürünler kullanılarak Sonuçlar ve Karşılaştırma bölümünde kıyaslamalı bir şekilde performans ölçümleri yapılmıştır.

Mevcut proje bir başlangıç çalışması olmakla birlikte geliştirilebilir yanları fazladır. Örneğin, sayfa içerikleri eğik gelen tarama örneklerinde bu durumun tolere edilebilmesi için satır ve sütunların yatay ve dikey olarak değil, eğimli olarak tespit edilmesi gerekmektedir. Ayrıca, daha ileri teknikler kullanılarak sayfa üzerindeki biçim özelliklerinin de kopyalanabilmesi mümkündür. Bu çalışmaların da ilerleyen zamanlarda yapılması planlanmaktadır.

Teşekkür

Bu çalışmamız TÜBİTAK TEYDEB tarafından 3160184 no'lu proje kapsamında desteklenmiştir.

Kaynakça

[1] Islam, N., Islam, Z., & Noor, N. (2016). A Survey on Optical Character Recognition System. Journal of Information & Communication Technology-JICT Vol. 10 Issue. 2.

[2] Baran, R., Partila, P., & Wilk, R. (2018, January). Automated Text Detection and Character Recognition in Natural Scenes Based on Local Image Features and Contour Processing Techniques. In International Conference on Intelligent Human Systems Integration (pp. 42-48). Springer, Cham.

[3] Shabana, M. A., Jose, A., & Sunny, A. (2018). TEXT DETECTION AND RECOGNITION IN NATURAL IMAGES.

[4] Kumar, P., Saini, R., Roy, P. P., & Pal, U. (2018). A lexicon-free approach for 3D handwriting recognition using classifier combination. Pattern Recognition Letters, 103, 1-7.

[5] Samanta, O., Roy, A., Parui, S. K., & Bhattacharya, U. (2018). An HMM Framework based on Spherical-Linear Features for Online Cursive Handwriting Recognition. Information Sciences.

[6] Sueiras, J., Ruiz, V., Sanchez, A., & Velez, J. F. (2018). Offline Continuous Handwriting Recognition Using Sequence to Sequence Neural Networks. Neurocomputing.

[7] Chauhan, R., & Pipalia, D. (2018). Smart Electronic Real Time Text Recognition Application. Journal of Electronic Design Technology, 8(3), 1-7.

[8] Liu, Z., Li, Y., Ren, F., Yu, H., & Goh, W. (2018). SqueezedText: A Real-time Scene Text Recognition by Binary Convolutional Encoder-decoder Network.

[9] Zanibbi, R., Blostein, D., & Cordy, J. R. (2004). A survey of table recognition. Document Analysis and Recognition, 7(1), 1-16.

[10] Yildiz, B., Kaiser, K., & Miksch, S. (2005, December). pdf2table: A method to extract table information from pdf files. In IICAI (pp. 1773-1785).

[11] Coüasnon, B., & Lemaitre, A. (2014). Recognition of Tables and Forms. Handbook of Document Image Processing and Recognition, 2014.

[12] Parikh, R., & Vasant, A. (2013). Table of Content Detection using Machine Learning: Proposed System. International Journal of Artificial Intelligence & Applications, 4(3), 13.

[13] Bansal, A., Harit, G., & Roy, S. D. (2014, December). Table Extraction from Document Images using Fixed Point Model. In Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing (p. 67). ACM.

[14] Vasileiadis, M., Kaklanis, N., Votis, K., & Tzovaras, D. (2017, April). Extraction of Tabular Data from Document Images. In Proceedings of the 14th Web for All Conference on The Future of Accessible Work (p. 24). ACM.