

**TÜRK SİVİL PLAKA STANDARTLARI İÇİN
ARAÇ PLAKA TANIMA SİSTEMİ**


Beytullah YALIM

**YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİMİ ANABİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

**MAYIS 2008
ANKARA**

Beytullah YALIM tarafından hazırlanan TÜRK SİVİL PLAKA STANDARTLARI İÇİN ARAÇ PLAKA TANIMA SİSTEMİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.


Yrd. Doç. Dr. Nurettin DOĞAN
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Elektronik-Bilgisayar Eğitimi Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan: : Prof. Dr. İnan GÜLER



Üye : Doç. Dr. M. Ali AKÇAYOL



Üye : Yrd. Doç. Dr. Nurettin DOĞAN



Üye : _____

Üye : _____

Tarih : 14 / 07 / 2008

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Beytullah YALIM

TÜRK SİVİL PLAKA STANDARTLARI İÇİN ARAÇ PLAKA TANIMA SİSTEMİ

(Yüksek Lisans Tezi)

Beytullah YALIM

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Mayıs 2008

ÖZET

Küreselleşen dünyada teknoloji alanında yaşanan hızlı gelişmeler, bu gelişmelerle birlikte teknolojik ürünlerde ve sanayideki ucuzlama, yerleşim bölgelerinin nüfusla birlikte hızla büyümesi trafiğe çıkan araç sayısını her geçen yıl misliyle katlanarak arttırmıştır. Bu artışla birlikte araç trafiğinde değişen ve gelişen ihtiyaçlar otomatik araç tanımlama sistemlerine duyulan gereksinimi tüm dünyada arttırmıştır. Bu gereksinimi karşılamak amacıyla gerçekleştirilen bu tez çalışmasında öncelikle, taşıt plakalarının resim içerisindeki yerinin bulunmasına ve bulunan plaka bölgelerinin okunmasına yönelik yöntemler tanıtılmıştır. Bununla birlikte Türk Plaka Standartları'na uygun sivil taşıt plakalarının okunması amacıyla, giriş olarak verilen uygun bir mesafeden çekilmiş taşıt görüntüsünü işleyerek plaka bölgesini koparan ve bu bölgedeki karakterleri tanıyarak taşıt plakasının bilgisayar tarafından okunmasını sağlayan bir sistem geliştirilmiştir. Geliştirilen bu sistemde, sık kullanılan resim işleme algoritmaları, tümevarımsal öğrenme ve şablon eşleştirme yöntemleri kullanılmıştır. Taşıt plakalarının yerlerinin bulunabilmesi için yeniden boyutlandırma, gri seviyeye indirgeme, Histogram eşitlemesi, Thresholding, Smearing algoritmalarından oluşan karma bir sistem tasarlanmış ve bu

algoritmalar morfolojik filtreleme yöntemleri ile birbirlerini tamamlayacak şekilde entegre edilmişlerdir. Plakanın okunması işleminde ise morfolojik filtrelemeler, şablon eşleştirme ve tümevarımsal öğrenme yöntemleri kullanılmıştır. Yapılan denemeler sonucu farklı ortamlarda önden ve arkadan çekilmiş taşıt resimlerinin plaka bölgelerinin bulunması ve plakaların okunmasında önerilen sistemin oldukça başarılı sonuçlar verdiği gözlenmiştir.

Bilim Kodu : 702.3.006
Anahtar Kelime : plaka tanıma, karakter tanıma, şablon eşleştirme,
morfolojik filtreleme
Sayfa Adedi : 94
Tez Yöneticisi : Yrd. Doç. Dr. Nurettin DOĞAN

**VEHICLE IDENTIFICATION SYSTEM FOR TURKISH PRIVATE
LICENCE PLATE STANDARTS**

(M.Sc. Thesis)

Beytullah YALIM

**GAZİ UNIVERSITY
INFORMATICS INSTITUTE**

May 2008

ABSTRACT

In the global era, experienced revolution in production processes thanks to the development of new technologies, population growth, and intensified urbanization results in increase in the number of vehicles. This situation has generated world wide interest in and demand for Automatic Vehicle Identification Systems (AVIs). This thesis introduces a new system designed to read the license plates of private vehicles based on the Turkish Licence Plate Numbering System. In the literature review part, historical development of the methods has been utilized for reading the licence plates on vehicles were examined. The main part of the study explains the procedures and techniques used at the image acquisition and processing and character recognition steps to design the new system. Image processing algorithms, inductive learning and template matching techniques were utilized in this study. More specifically, for license plate recognition, following techniques were integrated through morphological filtering: resizing the captured image, gray scale digital image, histogram equalization, Thresholding and Smearing algorithms. In character recognition and license plate identification step, morphological filterings, template matching and inductive learning were used. Experiments were conducted in different conditions and with the images from the front and back

sides of the vehicles and promising results were attained in terms of the performance and the accuracy of the system.

Science Code : 702.3.006
**Keywords : License plate recognition, character recognition,
template matching, morphologic filtering**
Page number : 94
Advisor : Assist. Prof. Dr. Nurettin DOĞAN

TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren Hocam Yrd. Doç. Dr. Nurettin DOĐAN' a, verdięi sonsuz destek ve gösterdięi sabır için Eőim Gözde Özel YALIM' a, moral kaynaęım oęlum Egemen YALIM' a, desteklerini hiç esirgemeyen bölümdeki öęretmen arkadaşlarıma teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ	xi
ŞEKİLLERİN LİSTESİ	xii
RESİMLERİN LİSTESİ	xiv
1. GİRİŞ	1
2. PLAKA TANIMA SİSTEMİ YÖNTEMLERİ	6
2.1. Önceki Çalışmalar	6
2.2. Plaka Bölgesinin Bulunması İle İlgili Yöntemler	8
2.2.1. Renkli resimden gri ölçekli resme dönüştürme	8
2.2.2. Histogram eşitleme	10
2.2.3. Siyah-Beyaz forma dönüştürme	11
2.2.4. Kapalı dörtgenler, yatay/dikey çizgiler	12
2.2.5. Süzgeç algoritmaları	17
2.2.6. Radon dönüşümü	18
2.2.7. Genişletme ve aşınma algoritması	20
2.2.8. Plaka bölgesinin ölçeklendirilmesi	23
2.3. Karakter Ayırıştırma Yöntemleri	26
2.3.1. Damla renklendirme yöntemi	27
2.3.2. Karakterler arası boşluk takibi	28
2.3.3. Dikey izdüşüm yöntemi	28
2.3.4. Maskeleyme yöntemi	29
2.3.5. Parça etiketleme	35
2.3.6. Hough Transformu (Geçişi)	36
2.4. Karakter Tanıma İle İlgili Yöntemler	38
2.4.1. Şablon eşleştirme	40

	Sayfa
2.4.2. Momentler yardımıyla optik karakter tanıma.....	41
3. PLAKA TANIMA SİSTEMİ	46
3.1. Plaka Bölgesinin Bulunması	47
3.1.1. Yeniden boyutlandırma	48
3.1.2. Gri seviyeye indirgeme	48
3.1.3. Histogram Eşitleme	51
3.1.4. Siyah-beyaz forma dönüştürme (Thresholding).....	55
3.1.5. Morfolojik filtrelemeler	58
3.1.6. Smearing algoritması.....	60
3.2. Plakadaki Karakterlerin Ayrıştırılması.....	66
3.2.1. Önışleme algoritmaları	66
3.2.2. Karakter en/boy oranısı ve ölçülendirme.....	72
3.3. Plakanın Okunması (Karakter Tanıma).....	80
3.3.1. Karakterlerin boyutlandırılması	81
3.3.2. Projeksiyon işaretleme.....	84
3.3.3. Tümevarımsal öğrenme ve şablon eşleştirme	85
4. SONUÇ VE ÖNERİLER	88
KAYNAKLAR.....	92
ÖZGEÇMİŞ.....	94

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Histogram eşitlemesi öncesi renklerin örnek tekrar sayıları.....	11
Çizelge 2.2. Nokta belirlemesi için 3x3 maskelemesi	30
Çizelge 2.3. Yatay Çizgiler için örnek Maskeleme	30
Çizelge 2.4. Değişik Yönlerdeki Çizgi maskelemeleri.....	31
Çizelge 2.5. a. Bir görüntünün 3x3 alanı b. Roberts maskelemesi.....	32
Çizelge 2.6. Kenar belirlemesi için Prewitt maskelemesi.....	33
Çizelge 2.7. Köşe belirlemesi için Sobel maskelemesi.....	33
Çizelge 2.8. Köşe belirlemek için Gaussian fonksiyonunun Laplacian' ını kullanan bir 5x5 maskelemesi	34
Çizelge 3.1. Şablon metodunun uygulanmasıyla elde edilen arşivlenmiş yanlış sonuçlar ve tekrarlı öğrenme sonucu doğru okuma	86
Çizelge 4.1. Yapılan testlerde sistemin öğretilmesi için kullanılan karakter sayıları.	89
Çizelge 4.2. Deneysel sonuçlar	90

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. RGB değerlerinin kartezyen koordinatlarda gösterimi	8
Şekil 2.2. Histogram eşitleme örneği ve diyagramı, a.Histogram eşitlemeden önce resim ve diyagram, b. Histogram eşitledikten sonra resim ve diyagram. .11	11
Şekil 2.3. Çekirdek ile matrisin korelasyonu	18
Şekil 2.4. Koordinat düzleminden, Radon bölgesine geçiş.....	19
Şekil 2.5. Genişletme ve aşınmanın gösterimi a. Genişletme b. Aşınma	21
Şekil 2.6. Ara değerleri bulunacak noktalar (Kinder <i>v.d.</i> 1999).....	24
Şekil 2.7. Doğrusal ara değerlendirme sonuçları (Kinder <i>v.d.</i> 1999)	24
Şekil 2.8. Polinomal ara değerlendirme sonuçları (Kinder <i>v.d.</i> 1999).....	25
Şekil 2.9. Spline ara değerlendirme (Kinder <i>v.d.</i> 1999).....	26
Şekil 2.10. İki ayrı plakanın izdüşümlerinin çıkartılması (Cheokman W.U <i>et al.</i> 2005).....	29
Şekil 2.11. Kontur Takibi.....	35
Şekil 2.12. Koordinat noktaları, b., ve c. olası düz çizgi ayarlamaları.	37
Şekil 2.13. Düz satırın parametrik tanımı	37
Şekil 2.14. Şablon eşleştirme yöntemi a. Plaka üzerinden ayrıştırılan karakter resmi . b.Veritabanında karşılaştırma sonucu bulunan karakter (en yakın karakter)	41
Şekil 3.1. Histogram eşitleme işlemi a. Düşük kontrasta sahip resmin histogramı , b. Histogram eşitleme işleminden sonra resmin histogramı.....	52
Şekil 3.2. Plaka üzerindeki karakterlerin maximum ve minimum ölçüleri.....	58
Şekil 3.3. Yatay Smearing işlemi	61
Şekil 3.4. Plaka bölgesinin koordinatlarının belirlenmesi	64
Şekil 3.5. Plaka üzerindeki boşlukların yakalanması	73
Şekil 3.6. Plakanın dss() dizisine dikey değerleriyle sayısal olarak aktarılması.....	73

Şekil	Sayfa
Şekil 3.7. Plaka üzerindeki kısım başlangıç bitiş noktalarının yakalanması.....	74
Şekil 3.8. Plakanın diziye 1 ve 0 olarak aktarılması.....	76
Şekil 3.9. Plakanın karakterlerinin yatayda başlangıç ve bitiş noktalarının diziye yüklenmesi.....	77
Şekil 3.10. Karakter tanıma modülünün çalışması	81
Şekil 3.11. Plakadan kopartılan karakterlerin boyutlarının eşitlenmesi (16x32 piksel)	82
Şekil 3.12. Karakterin yatay ve dikey projeksiyonlarının çıkartılması.....	85

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 1.1. Türk sivil plaka çeşitleri, a. Tek harfli, b-c. İki harfli , d. Üç harfli.....	3
Resim 2.1. Kapalı dörtgenler yatay/dikey çizgiler yönteminin uygulaması, a. Siyah – beyaz forma dönüştürülmüş resim, b. Kapalı dörtgenler yatay/dikey çizgiler yöntemi uygulanmış resim	13
Resim 2.2. Açılış ve Kapanış işlemlerinin araç resmine uygulanması a. Orijinal gri ölçekli girdi görüntüsü, b. Açılıştan sonraki işlem görmüş görüntü, c. Çıkarımdan (a' dan b) sonraki işlem görmüş görüntü.....	22
Resim 2.3. Plaka üzerindeki harfler arasında boşluk takibi	28
Resim 3.1. Üretilen plaka tanıma yazılımının arayüzü.....	47
Resim 3.2. Orijinal 320x240 resim ve gri seviyeye indirgenmiş hali.....	49
Resim 3.3. Araç resmi üzerinde histogram eşitleme işleminin uygulanması, a. Gri seviyeye indirgenmiş araç resmi b. Histogram eşitleme işlemine tabi tutulmuş araç resmi.....	54
Resim 3.4. Histogram eşitleme işleminden sonra siyah-beyaz forma dönüştürülmüş resim	56
Resim 3.5. a. Siyah-beyaz resim b. Morfolojik filtreleme işleminden geçmiş resim	59
Resim 3.6. Smearing ve filtreleme işlemlerinden geçirilerek resim üzerinde plaka bölgesinin bulunması.....	62
Resim 3.7. Plaka bölgesinin kopartılması	64
Resim 3.8. Birden fazla aday plaka bölgesi işlemi, a. Aracın resmi, b. Smearing algoritması öncesi son durum, c. İki aday bölge derecelendirme işlemi ..	65
Resim 3.9. Plaka resmini gerçek renkten gri seviyeye indirgenmesi	67
Resim 3.10. Histogram eşitlemesi işleminden geçirilmiş plaka resmi	67
Resim 3.11. a. Gri seviye ve siyah-beyaz forma dönüştürülmüş plaka resimleri, b. Histogram eşitlenmiş ve siyah-beyaz forma dönüştürülmüş plaka resimleri.	69
Resim 3.12. Morfolojik filtreleme işlemi.....	70

Resim	Sayfa
Resim 3.13. Plaka tanıma sisteminde Karakter ayrıştırma modülünün tespit edemediği plaka 3. kısım 4. karakter gürültüsünün karakter tanıma modülünde elenmesi	78
Resim 3.14. Plakadaki tüm karakterlerin yatay ekseninde başlangıç ve bitiş noktalarının tespit edilmesi	78
Resim 3.15. Karakterlerin yatay ekseninde taranarak dikey sınırlarının belirlenmesi ..	78
Resim 3.16. Deforme olmuş karakter ve sistem tek karakter olarak kabul edilmesi..	80
Resim 3.17. Plakanın sisteme öğretilmesi.....	86

1. GİRİŞ

Teknolojinin küreselleşen dünyada olağanüstü hızlı ilerlemesi ve bu ilerlemenin son yıllarda daha büyük bir hızda gerçekleşmesi onunla uğraşanları dahi hayrete düşürecek bir hale gelmiştir. Hızla gelişen teknoloji ile birlikte günlük yaşamda karşılaşılan problemlerin giderilmesi veya rutin olarak tekrarlanan işlemlerin yapılması sırasında, işleri hızlandırmak ve hata yapmaya çok müsait bir varlık olan insan faktörünü en aza indirmek için, her alanda bilgisayar kontrollü sistemlere, bir başka deyişle otomasyon sistemlerine geçiş yaşanmaktadır. Tabi ki teknolojinin hızlı ilerlemesi ve otomasyon sistemlerine yönelme eğilimi, otomotiv sektöründeki yerini de almıştır. Bu durum kısa sürede ve yüksek kalitede daha çok otomobilin üretilmesini ve buna bağlı olarak fiyatlardaki düşmeyle beraber tüm dünyada olduğu gibi Türkiye’de de trafiğe çıkan motorlu araç sayısını katlanarak arttırmıştır. Bu artışa bağlı olarak günlük yaşamı oldukça etkileyen araç trafiği problemleri trafik akışını kontrol eden otomatik araç tanıma sistemlerine duyulan ihtiyacı oldukça arttırmıştır. Bu ihtiyaca cevap vermeye yönelik olarak yapılan çalışmaların amacı; araçları özel bir noktadan geçerken tanımlamak, konumunu belirlemek, hareketlerini gözlemlemek ve bu verileri kullanarak trafik denetimini kolaylaştırmaktır.

Tez süresince yapılan çalışmalarda da araçları tanımadaki insan görme ve algılama hatalarından kaynaklanacak problemleri en aza indirmek ve bilgisayar tabanlı sistemleri kullanarak insandaki görme ve algılama mekanizmalarını bilgisayar sistemine kazandırmaya çalışmak olacaktır.

Sayılan bu trafik problemlerinin çözümüne yönelik geliştirilecek Görme/Algılama Sisteminin amacı ; trafikte seyir halinde veya duran araçların, çekilen resmi üzerindeki plaka bilgisinin resim formatından sayısal forma dönüştürülmesi olarak tanımlanan “Araç Plaka Tanıma” (“Licence Plate Recognition”, LPR) işlemini gerçekleştirmektir. Bu tanıma işleminde üzerinde durulan nokta, insan faktörü olmaksızın ya da en aza indirgenmiş hali ile problemin sonuçlandırılabilmesidir.

Günümüzde trafik denetimini sağlamaya yönelik olarak, detektörler ve radyo frekanslarını kullanan radarlar, mikrodalga detektörleri, yolun altına yerleştirilen tüpler ve loop detektörleri bulunmaktadır. Bu sistemlerde, denetimin sağlanacağı yolun giriş ve çıkışlarına detektörlerin, radyo frekanslarını algılayan aygıtların yerleştirilmesi gerekmektedir. Ancak bu donanımların kurulumu ve algılayıcıların fazlalığı bu sistemleri pahalı hale getirmektedir. Bununla beraber, bu sistemlerin işletimi de zordur [1].

Pahalı, kurulumu ve işletimi zor sistemlerin yerine geçebilecek bilgisayar tabanlı sistemlere rağbetin günümüzde daha da arttığı gözlenmektedir. Bilgisayar tabanlı sistemlerin kurulumu kolaydır ve çok daha ucuza araç tanıma problemlerini çözebilmektedirler. Esas olan aracı tanımdır ve her aracın kendisine özel bir barkodu yani plakası mevcuttur. Bilgisayar tabanlı otomatik araç tanıma işlemi plaka bilgisi kullanılarak gerçekleştirilmektedir. Bu yüzden araç plaka tanıma problemi, ekstra bir donanım gerektirmeden, sadece görüntü işleme teknikleri kullanılarak ta çözülebilir. Bu yöntemde kamera yardımı ile aracın görüntüsü alınır ve alınan görüntüye çeşitli resim işleme ve görüntü tanıma yöntemleri uygulanarak aracın üzerindeki plaka bilgisi okunur. Bu sebeple bilgisayar tabanlı otomatik araç tanıma sistemi tez süresince Plaka Tanıma Sistemi (PTS) olarak adlandırılacaktır.

Plaka Tanıma Sistemleri günümüzde otomatik park sistemleri, trafik denetimi, araç takibi, köprü ve otoyol otomatik geçiş sistemleri, kurumların ve toplu yaşam sitelerinin giriş-çıkış kontrol noktaları gibi pek çok alanda kullanılabilir. Sıralanan nedenlerden dolayı Plaka Tanıma Sisteminin (PTS) günlük yaşamda oldukça yararlı sonuçlar getirdiğini söylemek yanlış bir kabullenme olmayacaktır.

Daha önce bahsedildiği üzere aracı tanımanın yolu Plaka Tanıma Sistemleri için plaka tanımadan geçmektedir. Bu tez çalışmasında Türk plaka standartlarına uyan sivil araçların plakalarının tanınması hedeflenmektedir. Türk plaka standartlarındaki dikdörtgen sivil plakalar 7 veya 8 karakterden oluşmaktadır. Bu karakterin ilk ikisi plakanın ait olduğu il kodunu göstermektedir. Ortada 1, 2 veya 3 harf bulunmakta ve ortadaki harf sayısına göre sonda 4, 3-4, 2 rakam bulunmaktadır. Daha açık olarak

belirtmek gerekirse ortada 1 harf olursa sonda 4 rakam, 2 harf olursa sonda 3 veya 4 rakam, 3 harf olursa sonda 2 rakam bulunmaktadır. Aşağıda sivil plakalara örnekler verilmiştir.



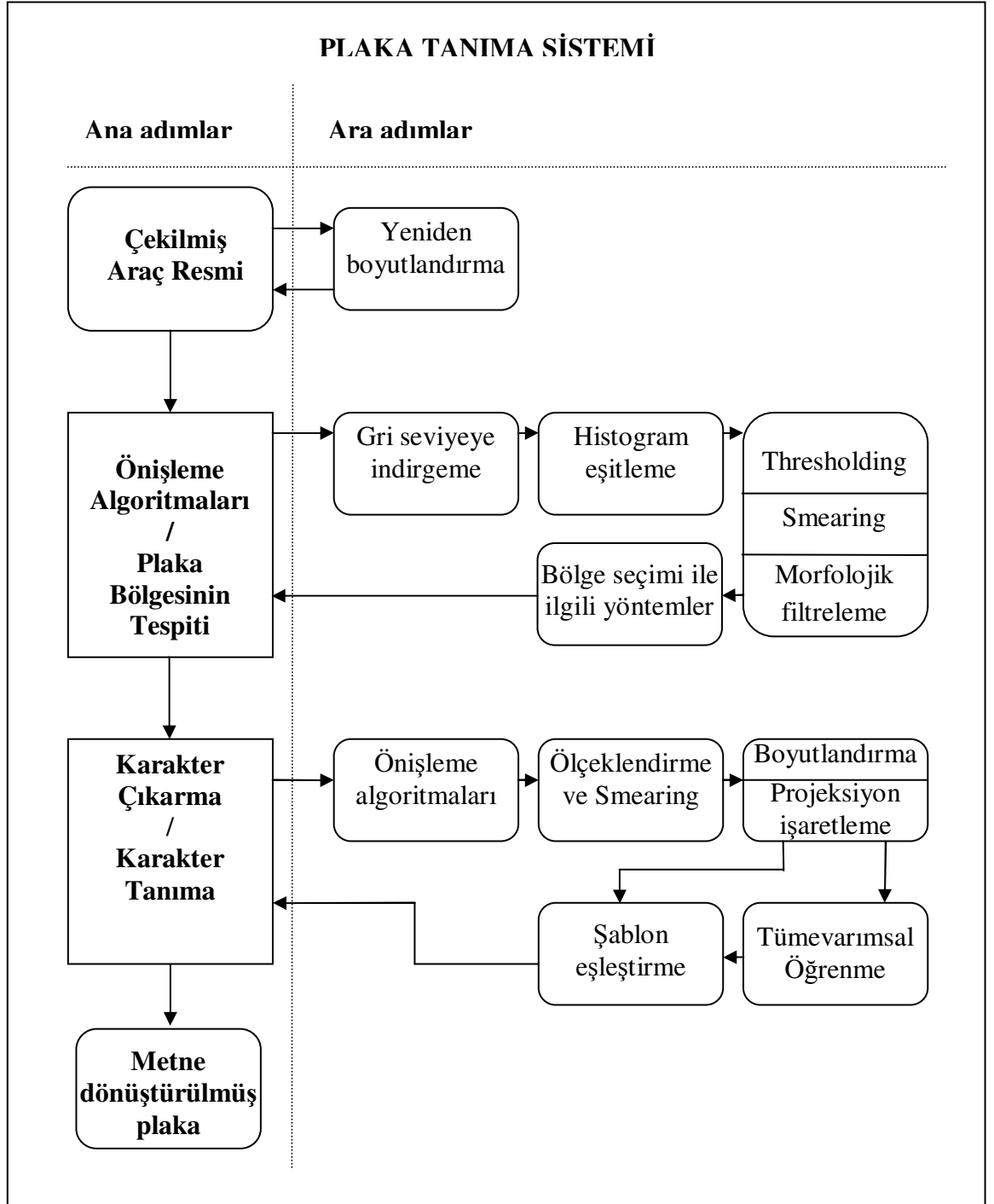
Resim 1.1. Türk sivil plaka çeşitleri, a. Tek harfli, b-c. İki harfli , d. Üç harfli

Bu tez kapsamında tasarlanan plaka tanıma sisteminin adımlarını inceleyecek olursak, bu adımlar; birtakım resim işleme algoritmalarıyla aday plaka bölgelerinin resimden kopartılması, plaka üzerindeki karakterin yerlerinin tespiti ve son olarak resim formatındaki karakterlerin metin formatına dönüştürülmesi şeklinde tanımlanabilir.

Aday plaka bölgelerinin tespit edilmesi işlem basamağında, resmin gri seviyeye indirgenmesi, Histogram eşitlemesi , Eşikleme (Thresholding) algoritması ile siyah-beyaz forma dönüştürme, Smearing algoritması ve birtakım morfolojik resim işleme yöntemlerinden yararlanılmıştır.

Plaka üzerindeki karakterin yerlerinin bulunması işlem basamağında yine morfolojik resim işleme algoritmalarından, genişletme ve daraltma yönteminden, Smearing algoritmalarından yararlanılmaktadır. Ayrıca karakter sayısı ve sınır noktaları belirlenirken plakanın harf sayısını tespit eden bir algoritma geliştirilmiştir.

Karakterlerin resim formatından text formata dönüştürülmesi işleminde yatay-dikey projeksiyona dayalı şablon eşleştirme yöntemi ve tümevarımsal öğrenme yöntemleri kullanılmıştır.



Şekil 1.1. Tez kapsamında ele alınan Araç Plaka Tanıma Sistemi genel tasarımı

Plaka Tanıma sistemleri ilgili çeşitli metotların ele alındığı resim işleme yöntemleri bölüm 2’ de “İlgili Yöntemler” başlığı altında anlatılmıştır. Bölüm 3’ te tez kapsamında geliştirilen plaka tanıma sistemi anlatılmıştır. Son olarak sonuç ve tartışma bölümünde geliştirilen sistemin performansı üzerinde durulmuştur.

2. PLAKA TANIMA SİSTEMİ YÖNTEMLERİ

Plaka tanıma sistemleri genel olarak üç aşamadan oluşmaktadır. Bu aşamalardan birincisi plaka bölgesinin araç resmi üzerindeki yerinin belirlenmesi, ikincisi plaka bölgesindeki karakterlerin ayrıştırılması ve son olarak çıkartılan karakterin resim formatından metin formatına dönüştürülmesi yani karakterlerin tanınması işlemleridir. Bu işlemleri gerçekleştirilirken sayısal resim işleme tekniklerinden faydalanılmaktadır. O halde önce sayısal resim işleme yöntemlerini incelemek gerekir. Bu bölümde plaka tanıma problemi ile ilgili dünyadaki önceki bazı çalışmalardan bahsedilecek, plaka bölgesinin çıkartılması ve karakter tanıma ile ilgili literatürde kullanılan yöntemler anlatılacaktır.

2.1. Önceki Çalışmalar

Otomatik araç tanıma konusunda ilk gelişme optik tarayıcı sistemlerin kullanılmasıyla 1960'larda ABD'de gerçekleşmiştir. Daha sonraki yıllarda mikro elektronikteki gelişmeler çalışmaları yoğunlaştırmış, induktif döngü, radyo frekansları, kızılötesi ve mikrodalga sistemleri araç gözetleme ve trafik kontrolünde kullanılmıştır.

Araç tanıma teknolojisinin yol ücretlendirilmesi için denenmesi ilk olarak Hong Kong'ta 1983-1985 tarihleri arasında uygulanmıştır. Önce gönüllü bir araç grubu elektronik plakalarla donatılmıştır. Sonra bu plakalara ait araçlar yoldan geçtiğinde araçların fotoğrafları kapalı devre TV ile alınmış ve kontrol merkezine iletilmiştir [2]. Araç tanıma sistemine yönelik çalışmalar özellikle Avrupa ve Japonya'da devam etmektedir. "Prometheus" denen dokuz yüz milyon dolarlık altı ülkeyi kapsayan bir proje hayata geçirilmiştir. Bu program araçları takip elektronik ceza kesme, plaka tanıma ve araç tanımayı içermektedir. Bununla güvenliği, verimi ve konforu arttırmak, ekonomik çözümler üretmek, kirliliği azaltmak amaçlanmaktadır [3]. Japonya'da da "Prometheus" a benzer bir program yapılmıştır. Programda yapay zekaya daha çok önem verilmiş, otomatik şoför denenmiştir. Hollanda ve Norveç'te de elektronik ücret toplanmaya yönelik araç tanıma sistemleri kurulmuştur [4].

Araç plakası, tanıma ve gözetleme için kullanılan tekniklerden biridir. Bu teknikte aracın plakasının yeri bulunur sonra bu plaka bilgisi ASCII karakterlere çevrilir. Sistemin yeni geliştirilmesi ve ticari değeri olmasından dolayı yapılan çalışmaların detayları hakkında literatürde birkaç yaklaşımdan fazlası gözükmemektedir. 1980’de Elsydel Ltd. şirketi gişelerde kullanılmak üzere bir araç plakası tanıma sistemi geliştirmiştir. Optik sensör ile algılanan aracın önden CCD kamera ile plakasının resmi yakalanmış, sonra araç plakasındaki karakterler deşifre edilmiştir. Sistem Fransa’da otoyollarda 1988’de test edilmiştir [1].

“Computer Recognition Systems Ltd.” şirketi 1989’da syntax forcing algoritması kullandığı bir araç plaka tanıma sistemi geliştirmiştir. Algoritması hakkında bilgi verilmeyen sistemin başarısının %93 olduğu açıklanmıştır [5].

1990’larda Newcastle-Upon-Tyne Üniversitesi’nin geliştirdiği plaka tanıma sisteminde aracın geçtiğini algılayan tetikleme ünitesi ve bu tetiklemeyle resim yakalayan kamera ünitesi bulunmaktaydı. Plakanın yeri, plaka zemini ve karakterlerin renk farkından yararlanılarak bulunmuştur. Yeri tespit edilen plakadaki karakterler etiketleme algoritması ile ayrıştırılmış ve yapay sinir ağları kullanılarak tanınmıştır [6].

1991’ de “CSIRO ve Telstra Corporation” işbirliği ile geliştirilen trafik denetleme sistemi bir araç tanıma modülünden oluşmaktaydı. Sistem yüksek kaliteli resimlerden araçların plakasını bulup kontrol merkezine göndermekteydi. Merkez araçların hızını ölçmekteydi ve şoförlerin yolda geçirdikleri süreyi hesaplamaktaydı yapay sinir ağları sistemde hem plakanın yerini hem de karakter tanımada kullanılmaktaydı. Plakaların %90 başarı ile tanındığı rapor edilmiştir [7].

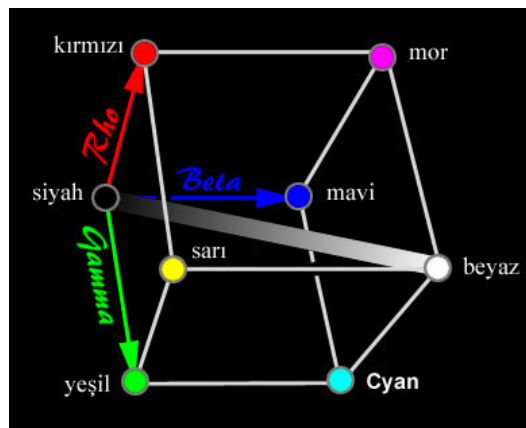
Bristol Üniversitesi bünyesinde bulunan araştırma merkezi “The Advanced Computing Research Centre” trafik denetleme, trafik gözetleme ve plaka tanıma sistemleri geliştirmiştir. Üniversiteden E.L. Dagless ve arkadaşları plaka yerini bulmak için çizgi metodunu geliştirmişlerdir. Metot da, görüntüden yatay çizgiler

alınmış ve bu çizgiler üzerindeki histogramdan yararlanmıştır. Histogramdaki değişimlerden karakter olabilecek yerler saptanmış ve karakterler aynı yöntemle ayrıştırılmıştır [8]. Aynı üniversiteden C. John Setchell trafik denetleme amacıyla yaptığı çalışmada, plaka yeri bulmada bu yöntemi kullanmış karakteri tanımayı yapay sinir ağlarıyla gerçekleştirmiştir [1].

2.2. Plaka Bölgesinin Bulunması İle İlgili Yöntemler

2.2.1. Renkli resimden gri ölçekli resme dönüştürme

Renkli resimler plaka tanıma sistemleri için çok fazla gereksiz ayrıntı içermektedir. Resmin renkli olması plaka tanıma sistemlerindeki kararlılığı da etkileyebilmektedirler. Daha önce belirtildiği üzere araç tanımanın yolu plaka bilgisinin okunmasından geçmektedir ve Türk plaka standartlarındaki sivil plakalar beyaz zemin üzerine siyah karakterlerden oluşmaktadır. Resimdeki renkli kısımlar gereksiz ayrıntılar olarak ta düşünülebilir. Bu nedenle renkli resmin gri forma dönüştürülmesi uygulanması gereken ilk ön işleme algoritması sayılabilir. Resimler piksellerden pikseller de Kırmızı-Yeşil-Mavi renklerin (RGB) farklı tonlarda birleşimlerinden oluşmaktadır.



Şekil 2.1. RGB değerlerinin kartezyen koordinatlarda gösterimi

R(red), G(green), B(blue) değerleri 0-255 arası değer almaktadırlar. 0' a yaklaştıkça rengin değeri en koyu halini almakta, 255'e yaklaştıkça da en açık tonuna

yaklaşmaktadır. Bu model, kartezyen koordinat sistemi kullanılarak açıklanmaktadır.

Küpün kartezyen koordinatlardaki (Şekil 2.1.) (0,0,0) orijin noktası siyah, orijinden en uzakta bulunan (1,1,1) noktası beyaz olarak gösterilmiştir. Bu gösterimden anlaşılacağı gibi beyaz tüm ana renklerin bir araya gelmesi sonucu oluşmaktadır. Ters durumda siyah meydana gelir. Bu modelde üç boyutlu uzayın bileşenleri kırmızı, yeşil ve mavidir. Bu yüzden küpün eksenleri R G ve B yi göstermektedir. Saf mavi (0,0,1) şeklinde gösterilmektedir. Diğerleri de aynı şekilde bir uç noktayı temsil eder. Gri değerler ise siyahtan beyaza doğru çizilen doğru üzerinde gösterilmektedir. Tüm renklerin [0,1] kapalı aralığında gösteriminin sebebi renklerin bu aralığa normalize edilmiş olmalarıdır. Aslında renk değerleri [0,255] aralığında değişmektedir.

Renkli sayısal bir resmi gri seviye resme çevirebilmek için RGB renk modelinde belirtilen gri-seviye renk değerlerinin orijinal resimdeki piksellerin her bir renk kanalındaki sahip oldukları renk değerlerinin yerine, resmin özelliklerini değiştirmeden konulması ile elde edilmektedir.

RGB küpündeki (Şekil 2.1.) beyaz-siyah köşegenindeki tüm değerler gri seviyedeki renk kodlandır. Bu ise RGB renk kanallarının hepsinin ortak bir değer ile doldurulmasından elde edilir. (127,127,127) üçlüsü grinin tonu iken (127,18,63) beyaz -siyah köşegeninde bulunmadığı için gri seviye değildir. Resmi gri forma dönüştüren eşitlik aşağıda verilmiştir.

$$Y=0.299R+0.587G+0.114B \quad (2.1)$$

Eşitlik 2.2 her bir piksel için uygulanır ve ortaya çıkan Y değeri, gri ölçekli resmin yeni RGB değerleridir.

$$R=Y \quad G=Y \quad B=Y \quad (2.2)$$

2.2.2. Histogram eşitleme

Günün farklı zamanlarında çekilen resimler farklı gri parlaklık/kontrast seviyelerine sahiptir. İnsanlar olarak bizler ortamdaki küçük ışık değişimlerinden pek etkilenmeyiz. Örneğin bir arkadaşımızın yüzünü öğle güneşi altında da, florasın ışığında da, akşam güneşinde de tanıyabiliriz. Ama bilgisayarlar için ortamdaki ışık değişimleri sorun olabilmektedir. Örneğin bir ağaç görüntüsünün 0..255 aralığındaki gri seviyelerinde kaydediliyor olduğunu farz edelim. 0 tam siyahı, 255 tam beyazı, aradaki değerler de gri tonlarını gösteriyor olsun. Bu görüntüyü öğlen vakti kaydettiğimizde elimizdeki verilerin ortalama değeri 255'e yakın değerler olurken, aynı ağacın görüntüsünü akşam kaydettiğimizde ortalama değer 0'a daha yakın olacaktır ve bu bilgisayar ile tanıma açısından küçük de olsa bir sorun oluşturur. Parlaklık ve kontrast seviyelerindeki bu dengesizlik sebebiyle oluşan problemleri aşabilmek için, resmin gri ton dağılımının homojen yapılandırılması sağlanmalıdır [9].

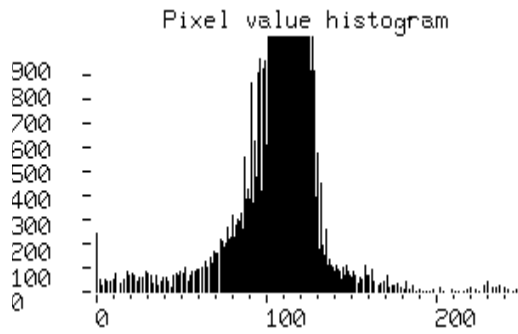
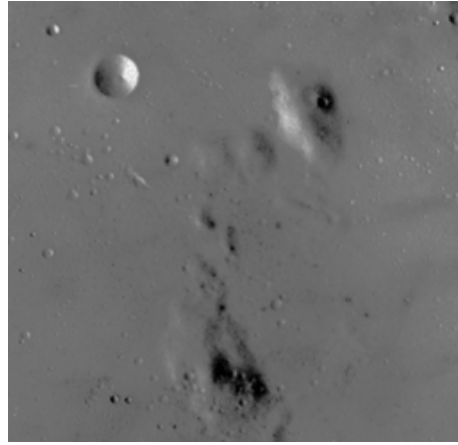
Her resimde gri ton dağılımı farklı olduğu için gri ton dağılım skalasında ("histogram") homojenliği sağlamak üzere histogram eşitleme ("histogram equalization") yöntemi kullanılmaktadır (Şekil 2.2.). Histogram: bir veri setindeki verilerin dağılımını veya başka bir deyişle kullanım sıklığını ya da frekansını gösteren bir tablodur diyebiliriz [10].

Histogram eşitleme işleminde, resmin kümülatif gri seviye dağılım skalası üzerinde normal dağılım uygulanmaktadır. Bu yeniden dağılım, gri seviye dağılımında dengeleme sağlamaktadır [11].

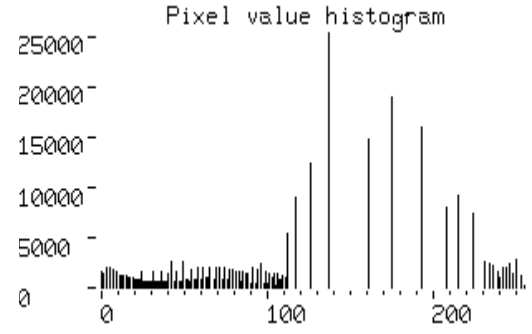
Histogramlar çoğu zaman grafik şeklinde ifade edilirler. Renk kodlarını yatay eksene dizilip tekrar sayılarını da dikey eksende işaretlenirse, histogramı grafik şeklinde ifade edilmiş olur. Örneğin 0..255 aralığında dağılmış 100 elemanlı bir veri seti için bir histogram şöyle olabilir:

Çizelge 2.1. Histogram eşitlemesi öncesi renklerin örnek tekrar sayıları

Renk Kodu	Tekrar Sayısı
0	15
1	17
2	25
3	20
...	...
255	3



a



b

Şekil 2.2. Histogram eşitleme örneği ve diyagramı, a.Histogram eşitlemeden önce resim ve diyagram, b. Histogram eşitledikten sonra resim ve diyagram.

2.2.3. Siyah-Beyaz forma dönüştürme

Plaka üzerinde yer alan karakterlerin bilgilerine daha rahat ulaşabilmek için, plakanın çevresel tüm objelerini görüntüden temizlemek gerekmektedir. Bunun için öncelikle

plaka üzerinde hangi objenin ön plan, hangi objenin arka plana ait olduğunun tespit edilmesi sağlanmalıdır. Bu amaç için yapılandırılan işleme siyah/beyaz forma dönüştürme (“binarization”) denmektedir.

Ön plan, arka plan ayrımı iyi bir eşikleme (“Thresholding”) değer saptaması ile gerçekleştirilmektedir. Gri seviye değerleri, eşik değerini geçemeyen pikseller arka plan, geçebilen pikseller ise ön plan üzerinde durmaktadır. Bu şekilde sınıflandırma yapılarak resim siyah beyaz forma dönüştürülmektedir [10].

Siyah-beyaz forma dönüştürme işlemi için kullanılacak birçok yöntem vardır. Ancak bu yöntemlerden en çok kullanılan yöntem ortalama ve standart sapma ile eşikleme (“mean&Standard deviation”) yöntemidir. Bu yöntem, ön/arka platform ayrımının yapılabilmesi için kullanılan en basit yöntemlerden birisidir. Bu yöntemde, aday plaka bölgesinin tamamı üzerinde ortalama değer saptanması yapılmaktadır. Tüm aday plaka bölgesi bu ortalama değer (μ) üzerinde sorgulanmaktadır. Eğer, plaka pikselleri, sorgulama sırasında ortalama değer artış veya azalış yönünde standart sapma aralığı içinde kalıyorsa ($[\mu \pm \sigma]$), karakterleri taşıyan ön plan, eğer bu aralığın içinde yer almıyorsa arka plan olarak sınıflandırılmaktadır. Bu metot, plakalarda net sonuçlar vermemektedir. Işık koşullarından ise doğrudan etkilenen bir yöntem olmaktadır [10].

2.2.4. Kapalı dörtgenler, yatay/dikey çizgiler

Plakanın yerini bulabilmek için kullanılan yöntemlerden birisi kapalı dörtgenlerdir. Bu metot yatay ve dikey kenar bilgileri kullanarak plakanın yerini tespit etmeye çalışmaktadır. Metodun uygulanışı sırasıyla şöyle yapılandırılmaktadır. Önce resim siyah/beyaz şekle dönüştürülmektedir (“binarization”). Ardından resimde kenar bilgisi çıkarsama (“edge detection”) uygulaması yapılmaktadır. Kenar bilgisi elde edildikten sonra, resim üzerinde “Hough transform” yöntemi ile belirlenmiş eğim toleransı ile yatay ve dikey çizgiler tespit edilmektedir. Çıkarılmış bu yatay ve dikey çizgilerden kapalı dikdörtgenler ya da dikdörtgen formuna en yakın dikey ve yatay

çizgi gurupları değerlendirilmektedir [12].

$$x \cos\Phi + y \sin\Phi = \varepsilon \quad (2.3)$$

Eşitlik (2.3)' de yer alan “ Φ ” değişkeni, kenar bilgisi çıkarılmış resimde yer alan aday çizginin, normalinin “ x ” eksenine yaptığı açığı ki bu açı belirli bir aralık tolerans değeri içermektedir, (x,y) değişkenleri koordinatları, “ ε ” ise eğim bilgisini göstermektedir [13].

Yatay ve dikey çizgiler ya da kapalı dikdörtgenler, bütün resim boyunca taranmaktadır. Amaç, belirli büyüklükteki bulunacak dikdörtgenlerin aday plakayı barındırmasıdır.

Kapalı dikdörtgenler, yatay/dikey çizgilerin çıkarılması metodu ancak plakaların çerçevelerinin net bir biçimde elde edilebildiği görüntülerde başarı ile uygulanabilmektedir [14] (Resim 2.1.). Plakaların kenar bilgilerinin bulunmadığı veya kaybolduğu resimlerde, bu metot başarısını kaybetmektedir. Ayrıca resimlerde, gri seviye kontrast problemi, akşam aydınlığı gibi ışık miktarının yetersiz olduğu hava koşullarında çekilen resimlerde bu metodun başarı oranı düşmektedir. Bu açıdan metodun zayıf yönleri uygulama alanını zayıf bırakmaktadır [12].



Resim 2.1. Kapalı dörtgenler yatay/dikey çizgiler yönteminin uygulaması, a. Siyah – beyaz forma dönüştürülmüş resim, b. Kapalı dörtgenler yatay/dikey çizgiler yöntemi uygulanmış resim

Aşağıda Kapalı dörtgenler yatay/dikey çizgiler yöntemini gerçekleştiren kod parçası verilmiştir.

```

c = 0
t = 0
For X = 0 To picwidth - 1
    c = 0
    t = 0
    For Y = 0 To picheight - 1
        s = GetPixel(h, X, Y)
        r = getred(s)
        If r = 255 Then
            t = t + 1
        Else
            t = 0
            c = 0
        End If
        If t > 30 Then
            SetPixelV h, X, Y, RGB(0, 0, 0)
        Else
            If t >= 30 And c = 0 Then
                For k = Y - 29 To Y - 1
                    SetPixelV h, X, k, RGB(0, 0, 0)
                Next k
                c = 1
            End If
        End If
    Next Y
Next X

c = 0
t = 0

```

```

For X = 0 To picwidth - 1
    c = 0
    t = 0
    For Y = 0 To picheight - 1
        s = GetPixel(h, X, Y)
        r = getred(s)
        If r = 255 Then
            t = t + 1
        Else
            t = 0
            c = 0
        End If
        If t < 2 Then
            SetPixelV h, X, Y, RGB(0, 0, 0)
        Else
            If t >= 2 And c = 0 Then
                For k = Y - 1 To Y
                    SetPixelV h, X, k, RGB(255, 255, 255)
                    c = 1
                Next k
            End If
        End If
    Next Y
Next X

c = 0
t = 0
For Y = 0 To picheight - 1
    c = 0
    t = 0
    For X = 0 To picwidth - 1
        s = GetPixel(h, X, Y)

```

```

r = getred(s)
If r = 255 Then
    t = t + 1
Else
    t = 0
    c = 0
End If
If t > 30 Then
    SetPixelV h, X, Y, RGB(0, 0, 0)
Else
    If t >= 30 And c = 0 Then
        For k = X - 29 To X - 1
            SetPixelV h, k, Y, RGB(0, 0, 0)
            c = 1
        Next k
    End If
End If
Next X
Next Y

c = 0
t = 0
For Y = 0 To picheight - 1
    c = 0
    t = 0
    For X = 0 To picwidth - 1
        s = GetPixel(h, X, Y)
        r = getred(s)
        If r = 255 Then
            t = t + 1
        Else
            t = 0

```

```

        c = 0
    End If
    If t < 2 Then
        SetPixelV h, X, Y, RGB(0, 0, 0)
    Else
        If t >= 2 And c = 0 Then
            For k = X - 1 To X
                SetPixelV h, k, Y, RGB(255, 255, 255)
                c = 1
            Next k
        End If
    End If
Next X
Next Y

```

2.2.5. Süzgeç algoritmaları

Ortalama süzgeci: belirlenen çekirdek ile resim matrisini korelasyon işlemine tabi tutar. Ortalama süzgecin çekirdeği:

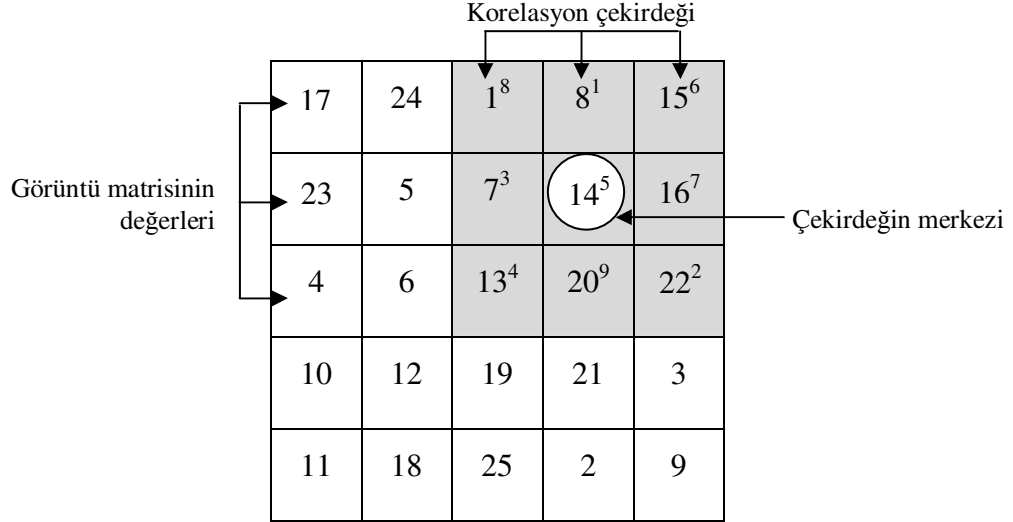
$$\begin{bmatrix} 1 & 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & 1 \end{bmatrix}_{nxn}$$

(n x n)

(2.4)

Çekirdeğin merkezi, korelasyonu hesaplanacak noktanın üzerine getirilir (Şekil 2.3.). Merkeze komşu olan noktalar, çekirdeğin içindeki değerler ile ağırlandırılarak toplamları alınır. Konvolusyona çok benzeyen bu işlemdeki fark çekirdeğin sabit kalarak hesaplanacak noktaları ağırlıklandırmasıdır [15].

$$1 \cdot 8 + 8 \cdot 1 + 15 \cdot 6 + 7 \cdot 3 + 14 \cdot 5 + 16 \cdot 7 + 13 \cdot 4 + 20 \cdot 9 + 22 \cdot 2 = 585$$



Şekil 2.3. Çekirdek ile matrisin korelasyonu

2.2.6. Radon dönüşümü

“Radon Dönüşümü”; iki boyutlu resmin özelliklerinin belirlenmesini sağlayan bir dönüşümdür. Görüntüdeki çizgileri, radon bölgesindeki noktalara dönüştürür.

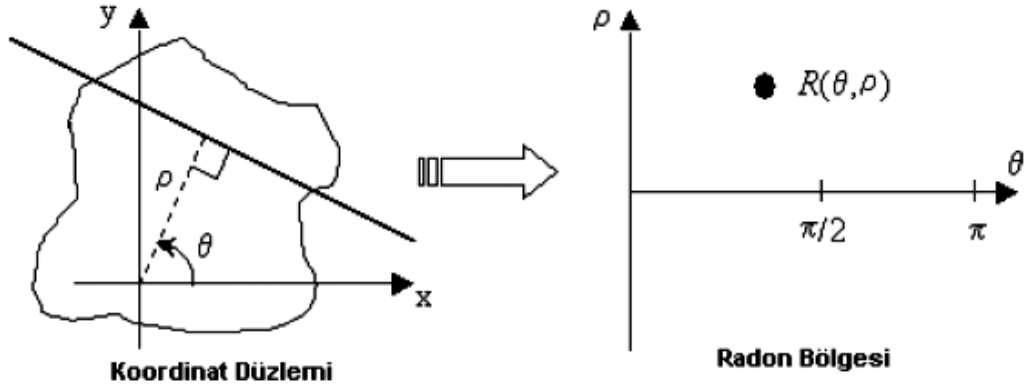
“Radon Dönüşümü”, resimdeki en belirgin çizgiyi düzlemle paralel hale getirecek şekilde çalışır. Görüntüyü içeren matrisin, belirli yöndeki izdüşümünü hesaplar. İki boyutlu $A(x,y)$ fonksiyonunun izdüşümü, bu fonksiyonun belirli bir yöndeki çizgi integralidir. İz düşümleri, belirlenen herhangi bir θ açısı kullanılarak hesaplanabilir. $A(x,y)$ fonksiyonunun “Radon Dönüşümü”, y eksenine paralel bir çizgi integralidir.

Verilen bir $A(x,y)$ fonksiyonu için, radon dönüşümü:

$$R(\theta, \rho) = \int_{-\infty}^{\infty} A(\rho \cos \theta - s \sin \theta, \rho \sin \theta + s \cos \theta) ds \quad (2.5)$$

Yukarıdaki eşitlik “s” doğrusu boyunca resmin integralini hesaplar. Formülde: ρ ,

çizginin orijine olan uzaklığı, θ çizginin yatayla arasındaki açıdır.



Şekil 2.4. Koordinat düzleminde, Radon bölgesine geçiş

Medikal görüntüleme her bir nokta için $R(\theta, \rho)$ ışın toplamı olarak adlandırılır. Sonuçta çıkan grafiğe ise “shadowgram” adı verilir.

Görüntü geri yansıtma yöntemi ile ışın toplamlarından, resmin ilk haline geri dönüş aşağıdaki ifade ile mümkündür:

$$B(x, y) = \int_0^{\pi} R(\theta, x \cos \theta + y \sin \theta) d\theta \quad (2.6)$$

İki boyutlu ara değerlemeden (interpolasyon) kaçınmak ve ara değerlendirme sırasında oluşacak hataları en aza indirmek için radon dönüşümü θ açısı kadar döndürülerek, “s” doğrusu boyunca ara değerlendirme yapılır.

$A(m, n) \mid [m=0 \dots M-1, n=0 \dots N-1]$ resmi için en yakın komşulukların ara değerlemesi:

$$R(\theta, \rho) = \begin{cases} \frac{\nabla_x}{[\sin \theta]} \sum A(m, [am + b]) & [\sin(\theta)] > \frac{\sqrt{2}}{2} \\ \frac{\nabla_y}{[\cos \theta]} \sum A([a'm + b'], n) & [\sin(\theta)] \leq \frac{\sqrt{2}}{2} \end{cases} \quad (2.7)$$

Parantezin içinde yer alan değerler, en yakın tamsayıya yuvarlanmıştır. düzeltme için verilen değerler (ofset) ve eğim değerleri aşağıda yer almaktadır.

$$a = \frac{\Delta x \cos \theta}{\Delta y \sin \theta} \quad b = \frac{\rho - x_{\min} \cos \theta - y_{\min} \sin \theta}{\Delta y \sin \theta} \quad (2.8)$$

$$a' = \frac{1}{a} \quad b' = \frac{\rho - x_{\min} \cos \theta - y_{\min} \sin \theta}{\Delta x \cos \theta} \quad (2.9)$$

Lineer ara deęerleme için donüşüm formülü ise řu řekildedir:

$$R(\theta, \rho) = \begin{cases} \frac{\Delta x}{[\sin \theta]} \sum_m (1-w)A(m, [am+b]) + wA(m, [[am+b]]+1) & [\sin \theta] > \frac{\sqrt{2}}{2} \\ \frac{\Delta y}{[\cos \theta]} \sum_m (1-w)A([a'n+b'], n) + wA([[a'n+b']] + 1, n) & [\sin \theta] \leq \frac{\sqrt{2}}{2} \end{cases} \quad (2.10)$$

2.2.7. Geniřletme ve ařınma algoritması

Geniřletme ve ařınma matematiksel yapı cebirinde 2 temel iřlemdir. Bu iki iřlem daha sofistike iřlemler elde etmek, için deęiřik kombinasyonlarda uygulanabilir.

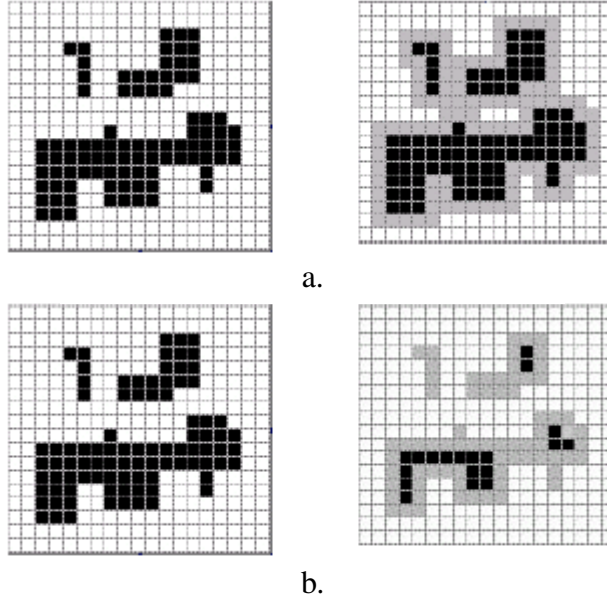
Geniřletme, iki kümeyi, řekillendirme elemanlarının kümesinin vektör toplamasını kullanarak birleřtiren morfolojik (yapısal) donüşümüdür. Ařınma ise siyah piksellerin tamamlayıcısını geniřleterek ve sonuçlandırıcı nokta kümesinin tamamlayıcısını alarak elde edilir [16].

B 'nin 3x3 'lük kare bir řekillendirme elemanını temsil ettięini varsayalım. A görüntüsünün geniřlemesi řekillendirme elemanlarını kullanarak, (2.10) ve (2.11) olarak tanımlanan dięer bir ikili sayı sistemi olan B içinde sonuçlanır ve řekil 2.5.' te gösterilir.

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \quad (2.10)$$

Benzer olarak görüntü A 'nın ařınması da řu řekilde tanımlanır.

$$A \otimes B = \{z \mid (B)_z \subseteq A\} \quad (2.11)$$



Şekil 2.5. Genişletme ve aşınmanın gösterimi a. Genişletme b. Aşınma

Genellikle genişleme ve aşınma diğer iki işlemle birleştirilir:

1- Açılış : Önce görüntüyü şekillendirme elemanlarıyla aşındırma ve sonra sonucu aynı şekillendirme elemanlarını kullanarak genişletme yoluyla elde edilir. Bir görüntünün açılması dış hatları pürüzsüzleştirir, dar istmusları (berzah) kırar ve küçük adaları yok eder. Şu şekilde tanımlanır:

$$A \circ B = (A \otimes B) \oplus B \quad (2.12)$$

2- Kapanış : Önce görüntüyü şekillendirme elemanlarıyla genişletme ve sonra sonucu aynı şekillendirme elemanlarını kullanarak aşındırma yoluyla elde edilir. Bir görüntünün kapanması dış hatları pürüzsüzleştirir , dar kırılmaları ve uzun , ince körfezleri eritir, küçük delikleri yok eder ve konturlar içinde boşlukları doldurur.

Şu şekilde tanımlanır:

$$A \bullet B = (A \oplus B) \otimes B \quad (2.13)$$



a.



b.



c.

Resim 2.2. Açılış ve Kapanış işlemlerinin araç resmine uygulanması a. Orijinal gri ölçekli girdi görüntüsü, b. Açılıştan sonraki işlem görmüş görüntü, c. Çıkarımdan (a' dan b) sonraki işlem görmüş görüntü.

Bu bölümde, PTS sisteminin ana basamağı olan plaka bölgesinin adaylarını bulmak için morfolojik metot kullanılmıştır. Bir plakanın zeminleriyle yüksek kontrast içinde olan birkaç karakterden oluştuğu bilinmektedir. Böylece plakadaki karakterleri bulanıklaştırmak için bir açılış işlemi kullanılabilir.

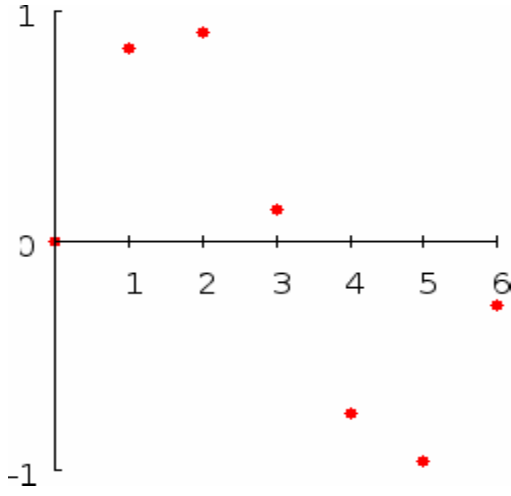
Öncelikle girdi görüntüsü gri ölçeğe transfer edilir ve sonra açılış işlemi olarak işler. Yapılandırma elamanıyla yapılan açılış işlemi, bir görüntüden b' den daha küçük olan detayları çıkarır. Bu yüzden eğer yapılandırma elamanı doğru seçilmiş ise, plakanın konumu açılıştan sonra bulanıklaşacaktır.(Resim 2.2.a) Orijinal görüntüden açılış görüntüsü çıkarılarak iki görüntü arasında çok büyük fark olduğu için plakanın yeri aydınlanacaktır. (Resim 2.2.b)

2.2.8. Plaka bölgesinin ölçeklendirilmesi

Bulunan plaka bölgesinin ölçeklendirilerek karakter ayrıştırma işlemine hazırlanması gerekmektedir. Yeniden ölçeklendirme işlemi interpolasyon (ara değerlendirme) ile gerçekleştirilir [15].

Ara değerlendirme elde var olan noktalar kullanılarak, elde olmayan diğer noktaların arasında kalan değerleri belirlemektir (Kinder et. al. 1999).

X	f(x)
0	→ 0
1	→ 0.8415
2	→ 0.9093
3	→ 0.1411
4	→ -0.7568
5	→ -0,9589
6	→ -0.2794

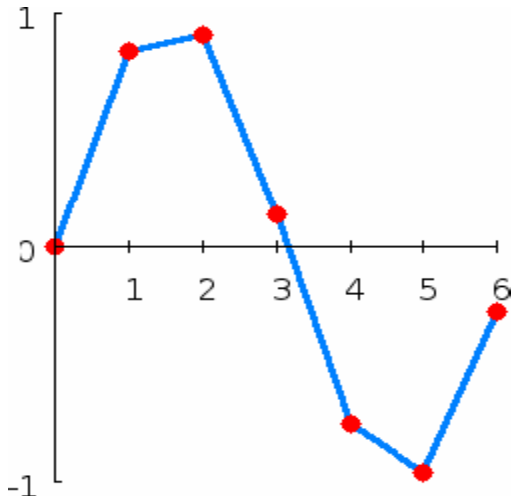


Şekil 2.6. Ara değerleri bulunacak noktalar (Kinder v.d. 1999)

Yukarıda verilen örnek için ara değerlendirme yöntemi incelenecek olursa; doğrusal ara değerlendirme; (x_a, y_a) , (x_b, y_b) noktalarını kullanarak;

$$y = y_a + \frac{(x - x_a)(y_b - y_a)}{(x_b - x_a)} \quad (2.14)$$

denklemini ile (x, y) noktaları belirlenir. Ancak doğrusal ara değerlemenin hızı çok yüksek olmasına rağmen doğruluk derecesi çok yüksek değildir.

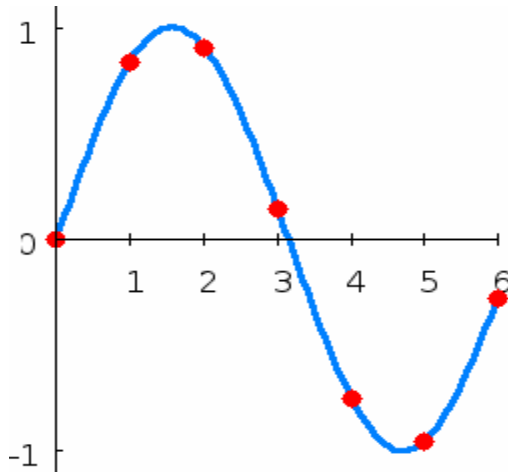


Şekil 2.7. Doğrusal ara değerlendirme sonuçları (Kinder v.d. 1999)

Polinomsal Ara Değerleme: Doğrusal ara değerlemenin geliştirilmiştir. Doğrusal ara değerlemede, iki nokta arasında çizilen doğru üzerinden işlem yapılırken, polinomsal ara değerlemede yüksek dereceden, ön bilgi olarak verilmiş tüm noktaları üzerinde barındıran bir polinom esas alınarak işlem yapılır. Bu sayede önceden verilmiş olan iki nokta arasında ki nokta ya da noktalar tespit edilir .

$$f(x) = -0.0001521x^6 - 0.003130x^5 + 0.07321x^4 - 0.3577x^3 + 0.2255x^2 + 0.9038x \quad (2.15)$$

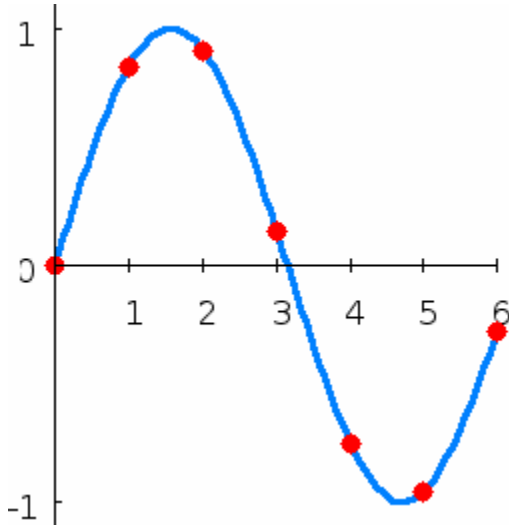
Doğruluk derecesi, doğrusal ara değerlemeye göre oldukça fazla olmasına rağmen, işlem yükü çok ağırdır.



Şekil 2.8. Polinomsal ara değerleme sonuçları (Kinder v.d. 1999)

Spline Ara değerleme: Polinomsal ara değerlemenin doğruluk derecesine çok yakın sonuçlar üretir. Bu işlemler için polinomsal ara değerleme kadar yüksek dereceden işlem yapmaması olumlu bir yönü olarak ortaya çıkar. Bu sebeple ara değerleme yöntemlerinin içinde en kullanışlı olanıdır.

$$f(x) = \begin{cases} -0.1522x^3 + 0.9937x & x \in [0.1] \\ -0.01258x^3 - 0.4189x^2 + 1.4126x - 0.1396 & x \in [1.2] \\ 0.1403x^3 - 1.3359x^2 + 3.2467x - 1.3623 & x \in [2.3] \\ 0.1579x^3 - 1.4945x^2 + 3.7225x - 1.8381 & x \in [3.4] \\ 0.05375x^3 - 0.2450x^2 + 1.2756x - 4.8259 & x \in [4.5] \\ -0.1871x^3 + 3.3573x^2 - 19.3370x - 34.9282 & x \in [5.6] \end{cases} \quad (2.16)$$



Şekil 2.9. Spline ara değeri (Kinder v.d. 1999)

2.3. Karakter Ayırıştırma Yöntemleri

Bu kısımda kopartılmış plaka görüntüsü üzerindeki karakterlerin resim bilgisi olarak ayrıştırılması üzerinde durulacaktır. Plaka resmi üzerinde yapılan karakter ayrıştırma işleminin çıktısı, karakter örüntülerinin siyah-beyaz formdaki resimleri olmaktadır. Bu resimler bir sonraki modül olan karakter algılama modülüne girdi olarak verilmektedir.

2.3.1. Damla renklendirme yöntemi

Damla renklendirme yöntemi (“Blob Colouring”) temelde alan büyütme (“region growing”) tekniğine dayanmaktadır. Alan büyütme tekniği 8-bağlantılı pikseller (“8-connected component”) metodunu kullanmaktadır [17]. Yöntemin uygulanış amacı, bütün plaka resmi üzerinde, birbirine bağlı olan grup pikselleri ortaya çıkarmak ve bunları “etiketlemek” tir. Etiketleme süreci tamamlandıktan sonra ise, çıkan grupların karakter olup olmadığının belirlenmesi gerekmektedir [17].

Metodun uygulanması için önce aracın gri seviyeli plakası siyah-beyaz forma dönüştürülmektedir. Alınan plakanın üzerinde 8-bağlantılı piksel metodu uygulanmaktadır. Yani piksel komşusu olan 8 pikselle kıyaslanmaktadır. Komşu piksel, kıyaslanan pikselle aynı değeri taşıyorsa her iki piksel bir renk olarak işaretlenmektedir. Bu işaretleme bitişik olan tüm komşu pikseller için uygulanmaktadır. Sonuçta plaka üzerinde etiketleme “labeling” yapılmış farklı renk kodlarıyla gruplanmış damlalar (“blob”) elde edilmiş olmaktadır [10].

Elde edilmiş damlaların büyüklüklerine bakılmalıdır. Örneğin, plaka çıkarımında kullanılmış olan tekniklerden, çevreleyen kutu metodu gibi bir yöntem ile damla büyüklükleri karşılaştırılmakta, çok küçük ve çok büyük parçalar , değerlendirme dışı bırakılmaktadır. Bu sayede, plaka etrafında yer alan büyük çizgiler, gürültüler ve karakter olamayan ancak kullanılabilir “Hiragana” stili özel imgeler plaka kapsamı dışında tutulmaktadır [10].

Gürültülerin karakter olarak algılanmaması bu yöntemin avantajlarından birini teşkil etmektedir. Diğer avantaj ise, sadece tek satır plakaların incelenmesi yerine çift satırdan oluşan plakaların da değerlendirilmesidir. Gürültü kapsamında olmayan ancak plaka üzerinde var olan bilgilerin yok sayılması, bu yöntemin bir diğer olumlu yönünü oluşturmaktadır [17].

Birbirine yapışık karakterlerin ayırt edilememesi, karakterlere dokunan plaka dış çizgisinin oluşturduğu büyük objenin değerlendirmeye alınması, bu yöntemin

dezavantajları olarak sunulmaktadır. Damla ayıklamanın sağlıklı yapılması için kullanılan büyüklüklerinin tahminlenmesi, başlı başına yöntem kısıtlılık getirmektedir [17].

2.3.2. Karakterler arası boşluk takibi

Bu metot plaka üzerindeki karakterler arasındaki boşlukları sınır noktaları olarak kullanmaktadır. Yapılandırılan işlem bir tarayıcı çizgi yardımıyla, rotasyon problemi giderilmiş siyah/beyaz formundaki plaka resmi üzerinde soldan başlayarak “siyah” renkli piksel taşımayan tüm kolonların işaretlenmesi ile yapılmaktadır [18].



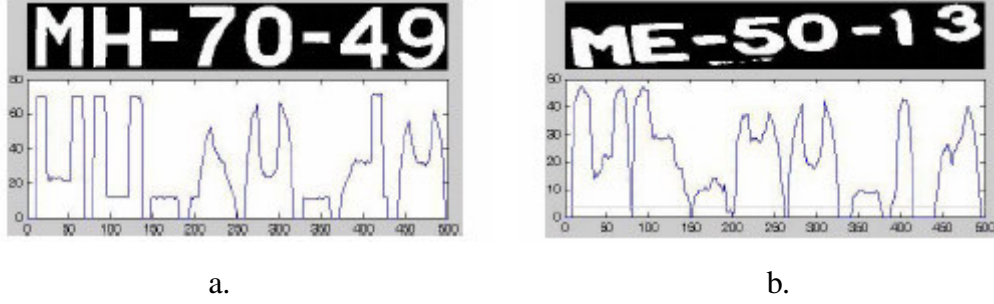
Resim 2.3. Plaka üzerindeki harfler arasında boşluk takibi

Bu yöntem çok temiz elde edilmiş siyah/beyaz plaka görüntüleri üzerinde basit ve hızlı bir çözüm üretmektedir [18]. Yani bu basit metodun başarılı olabilmesi için önceki adımlardan iyi kalitede bir plaka görüntüsü bu aşamaya aktarılmış olmalıdır.

2.3.3. Dikey izdüşüm yöntemi

Plakadaki karakterlerin ayrıştırılması için diğer bir yöntem de plakadaki karakterlerin dikey olarak izdüşümlerinin çıkarılması ve bunların takibidir. Plakanın yerleştirilmesinden sonra, karakterleri ayrıştırmak için dikey bir izdüşüm yapılmıştır. Şekil 2.10.a' da görüldüğü gibi, izdüşüm konturu bu karakterler arasındaki yüksek

pozitif değeri ve aralıklar arasındaki çok düşük pozitif değerleri göstermektedir. Düşük değer alanlar arasındaki mesafe karakterin genişliği olarak kabul edilebilir [16].



Şekil 2.10. İki ayrı plakanın izdüşümlerinin çıkartılması (Cheokman W.U *et al.* 2005)

Şekil 2.10.b. karakterlerin tam olarak yatay yönde olmadıkları ve izdüşüm konturündeki boşlukların her bir karakteri doğru olarak ayrıştırmayacağı genişlikte olduğu durumu göstermektedir. Bu durumda her karakter için en uygun yerleştirmeyi bulmak için, uyarlanabilen bir eşik seçilecek ve sonra her karakterin kesin sınırını belirlemek için arama algoritması uygulanacaktır. 147 görüntünün veri tabanını test ederek, yaklaşık %95'i yukarıda bahsedilen algoritmaya göre, doğru bir şekilde parçalanabilir [16].

2.3.4. Maskeleye yöntemi

Dijital görüntüler göz önüne alındığında, noktalar, satırlar ve köşeler olmak üzere üç temel çeşit gri seviye kesikliği olduğu görülür. Ayrıca, bir dijital görüntüde kesiklikleri ortaya çıkarmanın birçok yolu da vardır. Fakat, boşlukları ortaya çıkarmanın en çok kullanılan yolu maskeleye algoritması kullanmaktır. Maskeleye algoritması, istenilen kesiklik türü için uygun maskeleye kullanmayı ve görüntünün hedef bölgesine maskeleye uygulamayı içerir.

Noktaları belirleme:

Bir görüntüdeki izole noktaları ortaya çıkarmanın en kolay ve etkili yolu, gri

seviyeleri arka plandan ve çevreden belirgin şekilde ayıran, farklı nokta veya noktaları bulmaya yarayan maskeleme kullanmaktır. Böylelikle yüksek geçişli filtrelemede kullanılan maskeleme, homojen veya homojene yakın bölgedeki böyle izole nokta veya noktaları kolaylıkla belirleyebilmektedir. Amaca uygun 3x3 maskeleme örneği Çizelge 2.2.' de gösterilmiştir [19].

Çizelge 2.2. Nokta belirlemesi için 3x3 maskelemesi

-1	-1	-1
-1	8	-1
-1	-1	-1

Diğer bir taraftan bu maskeleme, yüksek geçiş filtreleme olayından farklıdır. Sıfır değer gösteren nokta veya noktalara homojen (gri basamak değerleri eşit olan) etraftaki noktaları oluşturan yüksek geçiş filtreleme olayından farklıdır. Bu, 3x3 maskelemesi için, maskenin merkez katsayısının 9 dan 8' e değişimidir.

Satırların belirlenmesi:

Bir görüntüdeki satır farklı gri basamak değerlerine sahip olan çevresinden ayırt edilebilir. Çizgi yatay, dikey veya herhangi bir yöne olabilir. Eğer söz konusu çizginin yönü biliniyorsa, o zaman çizgi piksellerine daha kuvvetli cevap verecek maskeleme kullanılır. Sıra pikselleri daha net olan (daha büyük katsayılar) ve diğerleri (sıra olmayan) 0 olan maskelemenin katsayısı seçilir. Eğer çizgi yataysa, Çizelge 2.3.' te gösterilen maskeleme çizgiyi gösterebilir [19].

Çizelge 2.3. Yatay Çizgiler için örnek Maskeleme

-1	-1	-1
2	2	2
-1	-1	-1

Burada eğer maskeleme sabit arka plana sahip yatay çizgisi olan görüntüye uygulanırsa, çizgi maskenin orta sırasından geçirildiğinde maksimum etki elde edilir. Benzer yaklaşım, Çizelge 2.4.' te sırasıyla maskelemeleri verilen +45 derece yönündeki ve – 45 derece yönündeki dik çizgilerin bulunmasında da kullanılabilir.

Çizelge 2.4. Değişik Yönlerdeki Çizgi maskelemeleri

-1	2	-1
-1	2	-1
-1	2	-1

Dik

-1	-1	2
-1	2	-1
2	-1	-1

+45° yönünde

2	-1	-1
-1	2	-1
-1	-1	2

-45° yönünde

Köşelerin belirlenmesi:

Bir görüntüdeki köşe, piksel yoğunluğunun koyudan açığa ve açıktan koyuya ani değişimidir. Başka bir deyişle, köşeler bir görüntüde nesnelerin kenarlarıyla kesişen yerlerdir. Nesnelerin kenarlarına ait olan bu pikselleri bulmaya köşe belirleme denir. Yoğunluğun hızla değiştiği yerleri arayan köşeleri bulmak için şu iki kriter kullanılır:

- Yoğunluğun ilk türevinin eşikten büyüklük olarak daha geniş olduğu yerlerde
- Yoğunluğun ikinci türevinin sıfır çaprazlamasının olduğu yerlerde

Bu operatörleri kullanırız çünkü, ilk türevin önemi bir görüntünün içindeki bir noktanın köşesinin varlığını bulmada kullanılabilir. Buna benzer olarak, ikinci türevin işareti köşe pikselinin hayali(imgesel) düz çizgiye sahip köşenin etrafındaki ikinci türevle, köşenin orta noktasının kenarında sıfır çaprazlayacak ikinci türevin ekstrem pozitif ve negatif değerlerini bağlayan (birleştiren) özelliklere sahip köşenin karanlık mı (koyu), aydınlık mı(açık) tarafında olduğunu belirlemede kullanılabilir. İkinci türevin bu sıfır çaprazlama özelliği, kalın köşelerin merkezlerinin belirlenmesinde oldukça faydalıdır [19].

Bir görüntüdeki birinci sıra türevler, derece derece değişen operatörler kullanarak hesaplanabilir ve ikinci türevler Laplacian kullanarak elde edilir. Bir görüntünün eğimi $f(x,y)$, (x,y) yerleşiminde şu şekilde tanımlanır:

$$\nabla f = [G_x; G_y] = \left[\frac{\partial f}{\partial x}; \frac{\partial f}{\partial y} \right] \quad (2.17)$$

Ve eğimin büyüklüğü şöyle ifade edilir:

$$\text{mag}(\nabla f) = [G_x^2; G_y^2]^{1/2} \quad (2.18)$$

Görüntünün eğiminin hesaplanması her bir piksel yerleşiminde kısmi türevlerin $\partial f / \partial x$ ve $\partial f / \partial y$ elde edilmesine dayanır.

Eğer bir 3x3 alanı şekil 2.17.(a) daki görüntüde yan alandaki gri seviyeleri temsil ediyorsa, z5 noktasındaki birinci sıra kısmi türevi hesaplamanın yollarından biri Roberts çapraz türev işlemini kullanmaktır.

$$G_x = (z9 - z5) \text{ ve } G_y = (z8 - z6) \quad (2.19)$$

Çizelge 2.5. a. Bir görüntünün 3x3 alanı b. Roberts maskelemesi

Z1	z2	Z3
Z4	z5	Z6
Z7	z8	Z9

a.

-1	0	-1	0
0	1	0	1

b.

Çizelge 2.6.'daki 3x3 maskelemesi kullanılarak, benzer bir yaklaşım kısmi türevlerin şu şekilde verilmesinde kullanılabilir:

$$\begin{aligned} G_x &= (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \\ G_y &= (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \end{aligned} \quad (2.20)$$

Burada 3x3 görüntü alanındaki birinci ve üçüncü satırları arasındaki fark, türevi x yönünde ortalamaktadır, üçüncü ve birinci sütunlar arasındaki fark ta türevi y yönünde ortalamaktadır.

Çizelge 2.6. Kenar belirlemesi için Prewitt maskelemesi

-1	-1	-1
0	0	0
1	1	1

-1	-1	-1
0	0	0
1	1	1

Sobel maskelemesi denilen, orta noktaya daha çok önem verilerek pürüzsüzleştirmeyi sağlayan maskeleme katsayısındaki bazı küçük değişiklikler Çizelge 2.7.'de gösterilmiştir.

Çizelge 2.7. Köşe belirlemesi için Sobel maskelemesi

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Sobel maskelemesi, Prewitt maskelemesinden daha üstün gürültü baskılama özelliği sağlar. Bir görüntünün Laplacian' ı ikinci sıra türevidir; şu şekilde tanımlanır:

$$\nabla f^2 = \left[\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right] \quad (2.21)$$

Laplacian, köşe belirlemede orjinal formunda kullanılmaz çünkü, gürültüye çok duyarlıdır. Bu yüzden sıfır çaprazlama yöntemiyle, köşeleri bulmak için öncü olarak yumuşatmayla(düzgünleştirme) kombine edilebilir. Çizelge 2.8.' de 5x5 Gaussian maskelemesinin Laplacian örneği verilmiştir:

Çizelge 2.8. Köşe belirlemek için Gaussian fonksiyonunun Laplacian' ını kullanan bir 5x5 maskelemesi

0	0	1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

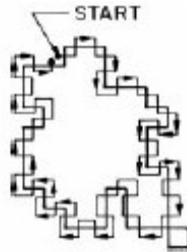
Özet olarak Sobel metodu köşeleri, türeve Sobel ortalaması kullanarak bulur. Görüntünün eğiminin maksimum olduğu yerlerdeki noktalarda köşeleri yansıtır. Prewitt metodu, köşeleri türeve Prewitt ortalaması kullanarak bulur. Görüntünün eğiminin maksimum olduğu yerlerdeki noktalarda köşeleri yansıtır. Robert metodu, köşeleri türeve Robert ortalaması kullanarak bulur. Görüntünün eğiminin maksimum olduğu yerlerdeki noktalarda köşeleri yansıtır.

Gaussian'ın Laplacian metodu, bir görüntüyü Gaussian'ın Laplacian filtresiyle filtreledikten sonra, sıfır çaprazlamaya arayarak köşeleri bulur. Sıfır çapraz metodu, sizin belirleyeceğiniz bir filtreyle bir görüntüyü filtreledikten sonra, sıfır çaprazlamaya bakarak köşeleri bulur [19].

2.3.5. Parça etiketleme

Herhangi bir başarılı görüntü parçalamanın (ayrıştırmanın) sonucu, bariz farklı parçanın içinde bulunan her pikseli etiketlemektir. Etiketlemenin bir yolu, bir görüntünün her pikseline etiket numarası veya parçasının işaretini eklemektir. Daha kısa bir yol ise, her parçanın kapalı konturunu belirlemektir. Eğer gerekliyse, bir konturun içindeki her pikseli etiketlemek için kontur doldurma teknikleri kullanılabilir. Aşağıda en çok kullanılan bir kontur izleme tekniği anlatılmıştır.

Şekil 2.11. ikili görüntü örneğinde kavramsal kontur takipçisi beyaz arka plandan kapalı konturla gösterilen siyah piksel alanına hareket etmeye başlamıştır. Kontur takipçisi siyah piksele geçtiğinde sola döner ve yandaki piksele devam eder. Eğer o piksel siyahsa, takipçi tekrar sola döner, eğer piksel beyazsa takipçi sağa döner. Bu işlemi takipçi başlama noktasına geri dönene kadar devam eder. Bu basit kontur takipçisi, bir sınırdaki çıkıntı pikselleri gözden kaçırabilir.



Şekil 2.11. Kontur Takibi

Diğer bir teknik, kontur takipçisinin geri beslemesi olarak bilinir. Bu algorithmada, eğer takipçi beyazdan siyaha piksel geçişi yaparsa, önceki başlangıç noktasına geri döner ve doğru bir dönüş yapar. Takipçi, beyazdan beyaza geçiş yaptığında doğru dönüş yapar.

Kontur takipçisi bir konturu izlerken, her sınır pikselinin koordinat listesini yaratabilir. Alternatif olarak, sınırdaki bazı referans piksellerinin koordinatları kayıt

edilebilir ve sınır, göreceli hareket koduyla tanımlanabilir. Bu şu demektir ki ele alınan pikseller bazı kodlama (şifreleme) teknikleriyle etiketlenmiştir.

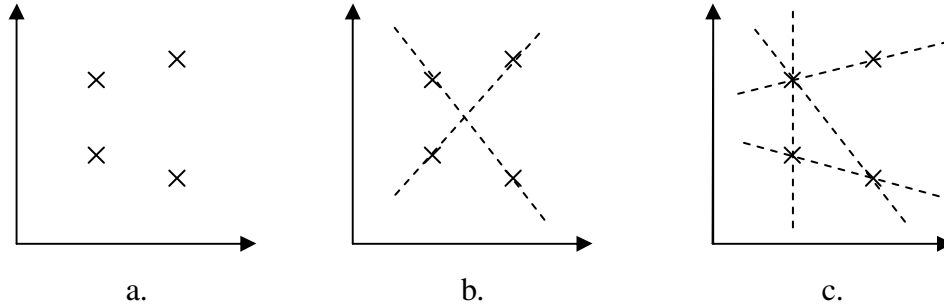
2.3.6. Hough Transformu (Geçişi)

Hough transformu, bir görüntüdeki belirli bir şeklin özelliklerini izole etmekte kullanılabilen bir tekniktir. İstenilen özelliklerin bazı parametrik şekillerde (formlarda) belirtilmesini gerektirdiğinden, klasik Hough transformu genellikle en çok elips, daire, çizgi gibi sıradan şekillerin belirlenmesinde kullanılır [19].

Hough transformunu anlamak için, Hough aralığını bilmek önemlidir. Hough aralığındaki her nokta (r, θ) , θ açısında bir satırla ve orjinal data alanındaki orijinden r uzaklığıyla kesişir. Hough aralığındaki bir fonksiyonun değeri veri alanında bir hat boyunca olan nokta yoğunluğunu verir. Hough transformu bunu aşağıdaki metodu kullanarak sağlar.

Orjinal alandaki (mesafe) her nokta için, Belirgin farklı açılardan o noktaya giden bütün hatları ele alın. Her bir θ açısı için o açıdaki nokta boyunca olan satıra mesafeyi hesapla ve r değeri vererek önceden seçilen ayrıştırılmayı kullanarak mesafeyi ayrıştır. Hough aralığının çapraz ayrıştırmasını yap. Bu, Hough aralığında bir takım kutuların oluşmasına sebep olacaktır. Bu kutular Hough akümülatörleri olarak adlandırılır. Yukarıda bahsedilen her bir satır için Hough akümülatörünün (r, θ) noktasında bir sayı (değer) artırılır (sıfırdan başlayarak). Tüm noktalar boyunca bütün satırları ele aldıktan sonra, yüksek değerdeki bir Hough akümülatörü büyük olasılıkla noktalar hattıyla kesişir. Gerçekte değişmeyen şekilde (sabit) dağılmış noktalar için, her bir Hough kutusu Poisson yayılmış (dağıtılmış) sayısına sahip olmalıdır. İlgili Poisson dağıtımının sonunda olan sayının temel standart datanın sonucu olması küçük bir olasılıktır, büyük ihtimalle bazı noktaların satır sonucudur. Bir satırdaki noktaların sayısı için önceden örnek (model) vermek, orijine uzaklık ve ilgili açıda satır olup olmadığına dair tam bir Bayesian değerlendirmesi sağlar. Basit bir örnek olarak, birtakım farklı görüntü noktasına satır parçalarının uydurulması

ortak problemi verilebilir (ör: köşe dedektörünün sonucu olan piksel yerleşimi). Şekil 2.12 bu probleme bazı olası çözümleri göstermektedir. Burada istenilen satır parçalarının sayısının önceden bilinmeyişi (ve satır parçalarını neyin oluşturduğunun belirsizliği) bu problemi sınırlandırılmaz hale getirmektedir.

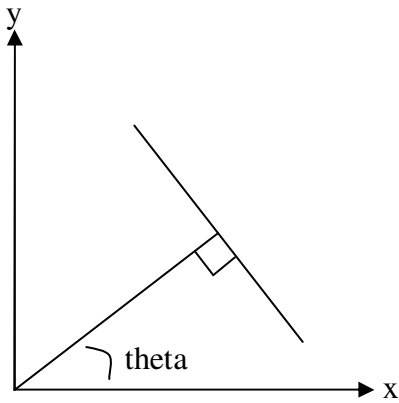


Şekil 2.12. Koordinat noktaları, b., ve c. olası düz çizgi ayarlamaları.

Bir satır parçasını analitik olarak birçok şekilde ifade edebiliriz. Fakat, bir küme satır tanımlamak için basit bir denklem eşitlik 2.22' deki gibi tanımlanabilir;

$$x \cos \theta + y \sin \theta = r \quad (2.22)$$

r , orijinden bu satıra olan uzunluk ve θ , r ' nin Şekil 2.13.' te olduğu gibi x eksenine göre yönlendirilmesidir. Bu satırdaki herhangi bir nokta için (x,y) , r ve θ sabittir.



Şekil 2.13. Düz satırın parametrik tanımı

Bir görüntü analiz bağlamında, görüntüdeki köşe parçalarının (ör: x_i, y_i) nokta koordinatları bilinir ve bunlar parametrik çizgi (sattır) denkleminde sabit sayılar olarak görev yaparlar. r ve θ ise bizim arayacağımız bilinmeyen değişkendir. Eğer her bir (x_i, y_i) tarafından tanımlanan olası (r, θ) değerlerini işaretleyecek olursak kartezyen görüntü alanındaki noktalar polar Hough parametresi alanındaki şekillerin (ör: sinüzoid) yerini belirler. Bu noktadan şekle transformasyonu, düz sattır için Hough transformasyonudur. Hough parametre alanına bakıldığında, kartezyen görüntü alanındaki aynı doğru üzerindeki noktalar ortak (r, θ) noktasında kesişen şekillere uyum sağladığı için kolayca belirgin hale gelirler.

Transformasyon, Hough parametre alanının ölçülebilir aralıklarla veya akümülatör hücreleriyle sayısal olarak belirtilmesiyle sağlanır. Algoritma devam ettiği sürece, her (x_i, y_i) farklı (r, θ) şekline transfer edilir ve bu şekil boyunca uzanan akümülatör hücreleri arttırılır. Akümülatör dizilişinde ortaya çıkan uç noktalar, görüntüde kesişen düz satırların olduğunun güçlü kanıtıdır.

2.4. Karakter Tanıma İle İlgili Yöntemler

Optik karakter tanıma genellikle OCR ile gösterilen bilgisayar çıktısı metin görüntülerini makinede düzenlenebilir metin haline veya karakterlerin resimlerini onları ASCII veya Unicode da temsil eden standart şifreleme şemalarına çevirmek üzere dizayn edilmiş bilgisayar yazılımını içerir. OCR, yapay zeka ve makine vizyonu alanındaki araştırma kolu olarak başlamıştır. Alandaki akademik araştırmalar devam etmekle birlikte, OCR üzerindeki odak noktası kanıtlanabilir tekniklerin uygulanmasına kaymıştır.

İnsan, tanıma işlemini okudukça yapar; basılı karakteri alarak, şeklini yorumlayarak ve anlamını atayarak. Diğer bir tarafta, bilgisayarlar bu beceriyi kendiliğinden başaramazlar, onlara bir karakter görüntüsünden metnin nasıl çıkarılacağını öğretilmesi gerekir.

1950 de AFSA'da kriptanalist) olan Birleşmiş Milletler Ulusal Güvenlik Ajansı (NSA) önde geleni David Shepard'a, Japonların PURPLE diplomatik şifresini kıran Frank Rowlett tarafından Ajans için data otomasyon prosedürleri hakkında tavsiyelerde bulunmak üzere Dr Louis Tordella ile çalışma teklifi götürüldü. Bu, yazılı mesajların bilgisayar işlemleri için makine diline çevrilmesi sorununu da içeriyordu. Shepard, bunu gerçekleştirecek bir makinenin yapılmasının olası olduğuna karar verdi ve bir arkadaşı olan Harvey Cook'un da yardımıyla tavan arasında günler haftalar boyunca uğraşıp 'Gizmo' yu yarattı. Daha sonra Shepard, Akıllı Makineler Araştırma Kurumu'nu (IMR) kurdu. Ticari işlemlerde kullanılan dünyanın ilk OCR sistemlerini bu şirket yaydı. Gismo ve daha sonra da IMR sistemleri görüntü analizi kullanırken karakter eşlemenin aksine ve bazı font çeşitlemelerini kabul ederken, Gismo oldukça kapalı dikey düzenlemeyle sınırlı kalmıştı. Halbuki, daha sonra çıkan ticari IMR tarayıcıları gerçek dünya dökümanları üzerinde pratik bir gereklilik olan taranmış alanın herhangi bir yerindeki karakterleri analiz etti. İlk ticari sistem, Readers Digest'te 1955'te kuruldu. Bu, yıllar sonra Readers Digest tarafından Smithsonian'a sergilenmek üzere bağışlanmıştır. İkinci sistem, hesap çıkarma amacıyla kredi kartı bilgilerini okuması için Kaliforniya'daki Standard petrol şirketine satılmıştır. Diğer sistemler, 1950'lerin sonuna kadar IMR tarafından fatura koçanı okuyucusu olarak Ohio Bell telefon şirketine, telgraf ve bilgisayar yazısı mesajlarını okumak ve iletmek için sayfa tarayıcısı olarak Amerikan hava kuvvetlerine satılmıştır. IBM ve diğerleri daha sonra Shepard OCR patentleriyle lisanslanmıştır. Bu, OCR sistemlerinin kısa bir geçmişidir.

Karakter tanımayı gerçekleştirmenin iki metodu vardır: şablon tabanlı tanıma ve özelliğe dayalı tanıma. Bu metotların her biri için yüzlerce sayıda algoritma olmasına rağmen, tüm algoritmaların altında yatan prensip bu iki kategoriye dayanır. Şablon tabanlı tanıma, şablon adı verilen bir grup bilinen karakter görüntüsüyle her bir girdi karakter görüntüsünü eşleştirir. Şablonlarla girdi görüntüsünü karşılaştırınca, en iyi benzerlik doğru tanımlamayı verir. Fakat, özelliğe dayalı (özellik tabanlı) tanımada, bu yaklaşım tanımlama veritabanıyla girdi görüntü belirleme özelliklerini veya etiketlerini karşılaştırma eğilimindedir. Girdi özellikleri girdi görüntüsüne en çok uyan karakter olarak tanımlanır.

Sonuç olarak plaka tanıma sistemlerinin son safhası olan karakter tanıma ayrıştırılan karakterlerin tanınarak alfa nümerik değerlere çevrilmesi işlemidir. Karakter tanıma ile ilgili sık başvurulan bazı yöntemler aşağıda verilmiştir.

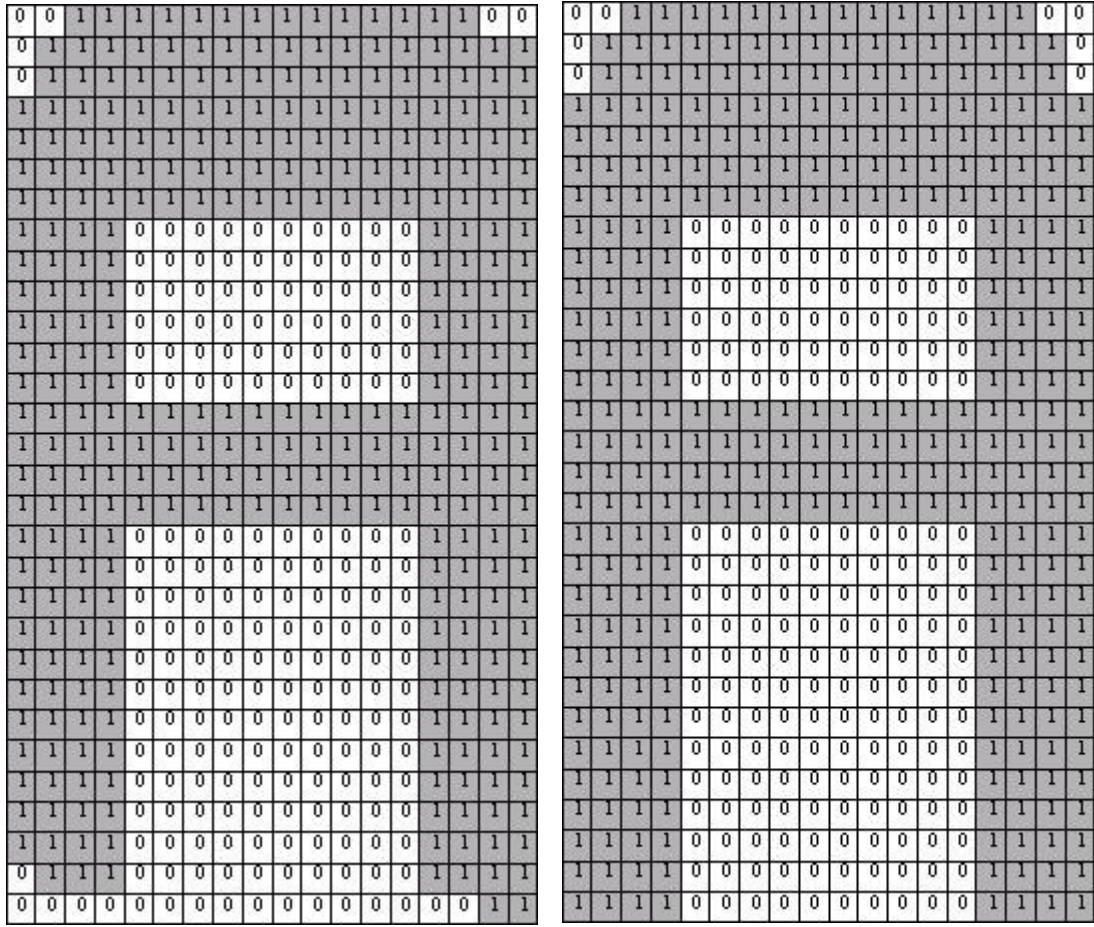
2.4.1. Şablon eşleştirme

Şablon eşleştirme (“Template matching”) yönteminde karakterlerine ayrılmış plakadaki her karakter resmi daha önceden veri tabanına kayıtlı karakterlerle kıyaslanır ve birbirlerine benzerliği bulunmaya çalışılır. Bilinmeyen karakterler ile, şablonun benzerliği ise “Euclidean Uzaklığı” ile hesaplanır [15]. Euclidean uzaklığı, matematikte iki noktanın birbirine olan uzaklığı olarak ifade edilir.

Şablon eşleştirme yönteminden yüksek başarı elde etmenin yolu aynı karakterlerden şablon kütüphanesinde (veritabanında) fazla sayıda bulunmasıdır. Bu şekilde bir karakter daha fazla şablon ile karşılaştırılacak ve daha yakınsak bir benzerlik elde etmek mümkün olacaktır. Ancak veritabanının çok fazla sayıda örnek karakter (şablon) içermesi karşılaştırma süresinin uzamasına dolayısıyla sistemin yavaşlamasına neden olacaktır.

Şablon eşleştirme yönteminde plaka resmi üzerinde ayrıştırılan karakter resmi veritabanındaki şablonlarla piksel piksel karşılaştırılır (Şekil 2.14.) ve en iyi sonucu üreten yani en çok benzeyen şablon resmin karşılığı olan karakter olarak kabul edilir.

Şablon eşleştirme yöntemi için gerekli olan en önemli önkoşul şablonların boyutlarının birbirine eşit olmasıdır. Plaka üzerinden kopartılmış karakter resminin de şablonların boyutlarına eşitlendikten sonra karşılaştırma işleminin yapılması gerekir. Aksi takdirde sistemin başarılı sonuçlar üretmesi mümkün değildir. Sistemin başarılı oranının yüksek olmasını sağlayan diğer önemli şablon boyutlarının büyük tutulmasıdır.



a.

b.

Şekil 2.14. Şablon eşleştirme yöntemi a. Plaka üzerinden ayrıştırılan karakter resmi
b. Veritabanında karşılaştırma sonucu bulunan karakter (en yakın karakter)

2.4.2. Momentler yardımıyla optik karakter tanıma

OCR(Optical Character Recognition) resim formatındaki herhangi bir dosya içindeki yazıyı tanıyarak, sonradan tekrar düzenlenebilecek metin biçimine dönüştürmektir.

Pattern recognition (örüntü tanıma) bilim disiplininin amacı nesnelere bir kategoriye koymak veya sınıflamaktır. Bu nesnelere, uygulamaya göre görüntü, ses ya da sınıflandırılması istenen başka bir işaret olabilir ve genel olarak pattern (örüntü) olarak adlandırılır. Karakter (harf veya sayı) tanıma, örüntü tanımanın başka bir önemli alanı “Optical Character Recognition” (OCR) sistemleridir. Optik Karakter

Tanıma (OCR), elektronik görüntüler üzerindeki karakterlerin ya da metin bilgilerinin okunarak ASCII koda dönüştürülmesi işlemidir. OCR metodolojileri kullanılarak makineler tarafından yazılmış karakterler, el yazısı karakterler ve işaretler kolaylıkla okunup ASCII koda dönüştürülebilirler. OCR, makineler tarafından yazılmış karakterlerin okunup tanınmasında kullanılan teknolojidir.

Karakterlerin tanınmasında birkaç OCR metodolojisi kullanılır. Bunlar arasında topolojik tabanlı olan, yapay sinir ağlarına dayanan, matris eşleme ile gerçekleşen, öznitelik analizi ile çalışan ve eğrileri izleyerek karakter tanıma işlemini gerçekleştiren metodolojiler yer almaktadır. Karakterlerin daha büyük doğrulukla tanınması için, okunan verilere ön işleme ve son işleme algoritmaları uygulanmaktadır. Bu algoritmalar, formların belirlenmesinde, formlar üzerindeki belirli alanların silinmesinde, form ve karakterlerin düzeltilmesinde ve görüntünün iyileştirilmesinde kullanılırlar. Karakterlerin büyük doğrulukla okunmasında, sisteme sunulan elektronik görüntünün mümkün olduğunca keskin (sharp), temiz ve düzgün olması oldukça büyük önem taşımaktadır. Buna rağmen araç plaka resimleri genellikle kirlili ve yıpranmışlardır. Tanıma işleminden önce uygulanacak ön işleme algoritmaları ile formun elektronik görüntüsü temizlenir ve karakterler mümkün olduğunca iyileştirilir.

Karakterlerin tanınması sırasında, kullanıcı tarafından kabul edilmeyen ya da doğruluk kriterini yakalayamayan karakterler reddedilmelidir (tanınmıyor olarak değerlendirilmelidir). Örneğin doğruluk kriteri olarak % 99.98 kabul edilirse, doğru olmadığı kabul edilen birçok karakter reddedilecektir. Eğer doğruluk kriteri daha aşağı sınırlara çekilirse, tanınan karakter sayısı artacaktır. Buna rağmen % 99 oranındaki doğruluk kriterleri sıklıkla kullanılmaktadır.

İçinde OCR tarafından okunamayan karakterler de bulunan reddedilen karakterler doğrulukla belirlenebilmelidir. Bu işlem genelde karakterin ekrana getirilerek el ile girişi yapılması süreci ile gerçekleştirilir.

Doğruluğu iyileştirmenin en son basamağı ise OCR ya da el ile girişi yapılmış tüm

karakterlerin işlenmesi, doğruluğunun istenilen kriteri yakaladığının sınanması ya da kullanıcı gereksinimini karşılayıp karşılamadığının belirlenmesidir. Son işleme işlemleri ile sözü edilen son basamak gerçekleştirilebilir. Ticari olarak var olan algoritmalar ile bu adım kolaylıkla uygulanabilir. Verinin arıtılması adı da verilen bu işlemler, karakter tanımadaki doğruluğu büyük oranda arttırmaktadır.

Geometrik momentler normalize ve merkezi normalize olmak üzere ikiye ayrılır. Normalize momentler yalnız öteleme ve ölçekleme için değişmezdir. Amaç öteleme, ölçekleme ve döndürme için değişmez dönüşümler bulmaktır. Bu nedenle Hu tarafından tanımlanan, öteleme, ölçekleme, ve döndürme için değişmez olan 7 adet momente değinilmiştir.

$I(x,y)$ bir sürekli görüntü fonksiyonu olsun. Bu görüntünün $p+q$ uncu dereceden geometrik momentleri şöyle tanımlanır:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q I(x, y) dx dy \quad (2.23)$$

$p, q = 0, 1, 2, \dots$

Geometrik momentler görüntü hakkında geniş bir bilgi sağlar ve pattern recognition için popülerdir.

Geometrik dönüşümlerde değişmezlik pattern recognition da istenen bir özelliktir. (2.23)' de tanımlanan momentler görüntünün koordinatlarına bağımlıdır ve değişmezlik özelliği göstermezler. Bu sorun, momentlerin normalize türlerinin uygun kombinasyonlarını tanımlamakla giderilir. Amaç aşağıda verilen dönüşümlerde değişmezliği olan momentler tanımlamaktır.

$$\begin{aligned} x' &= x + a \\ y' &= y + b \end{aligned} \quad (2.24)$$

Öteleme:

$$\begin{aligned} x' &= \alpha.x \\ y' &= \alpha.y \end{aligned} \quad (2.25)$$

Ölçekleme:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.26)$$

Döndürme:

$$\mu_{pq} = \iint I(x, y)(x - \bar{x})^p (y - \bar{y})^q dx dy \quad (2.27)$$

Burada;

$$\begin{aligned} \bar{x} &= \frac{m_{10}}{m_{00}} \\ \bar{y} &= \frac{m_{01}}{m_{00}} \end{aligned} \quad \text{dır.}$$

$$n_{pq} = \frac{\mu_{pq}}{\mu_{00}^y}$$

Normalize merkezi momentler:

$$y = \frac{p + q + 2}{2} \quad (2.28)$$

Bu momentler öteleme ve ölçekleme için değişmezdir.

1962' de Hu öteleme, ölçekleme ve döndürme için değişmez olan yedi moment tanımlamıştır. Bunlar:

$$p + q = 2$$

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
p + q &= 3 \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (\eta_{03} - 3\eta_{21})(\eta_{03} + \eta_{21}) \\
\phi_6 &= (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}) \\
&\quad \cdot \left[(\eta_{03} + \eta_{21})^2 - 3(\eta_{30} + \eta_{12})^2 \right] \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \\
&\quad \cdot \left[(\eta_{03} + \eta_{21})^2 - 3(\eta_{12} + \eta_{30})^2 \right]
\end{aligned} \tag{2.29}$$

Bu momentlerin ilk altısı aynı zamanda aynalamada da değişmezdir. Aynalama durumunda yedinci momentin mutlak değeri değişmez, işareti değişir. Bu büyüklükler oldukça değişik değerler alabilirler. Pratikte doğruluk sorunlarından kaçınmak ve değişim aralığını kısıtlamak için momentlerin mutlak değerlerinin logaritmaları kullanılır.

3. PLAKA TANIMA SİSTEMİ

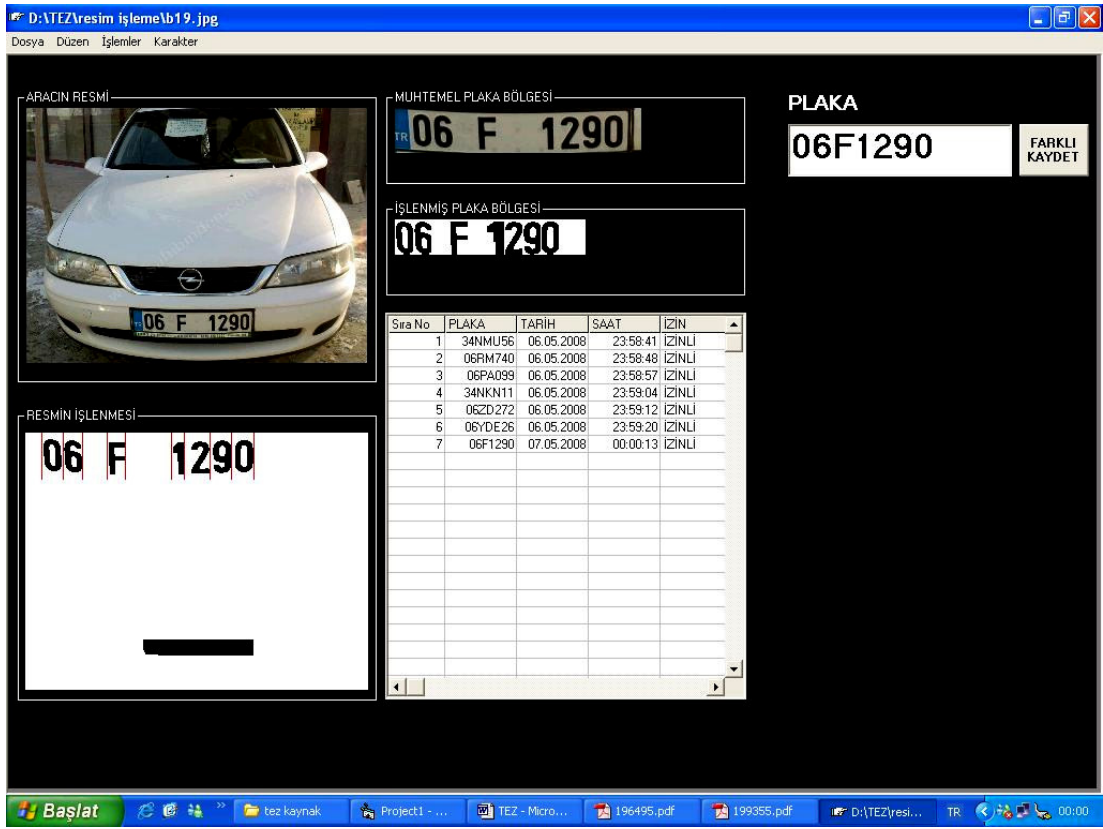
Bu bölümde tez süresince üzerine çalışılan ve tasarlanan Plaka Tanıma Sistemi anlatılacaktır. Hazırlanan plaka tanıma yazılımında 2 MP çözünürlükteki cep telefonu kameralarıyla çekilmiş resimlerden ve İnternet üzerindeki araç satış sitelerinden alınan araç resimlerinden yararlanılmıştır. Resimler günün değişik saatlerinde, farklı ışık ortamlarında, kapalı ve açık mekanlarda, değişik iklim şartlarında ve farklı arka plan fonlarında çekilmiş resimlerdir. Bu özelliği ile sabit bir kameradan, sabit bir ortamda çekilmiş resimlere göre işlenmesi ve plakaların okunması daha zor resimleri işleyebilen esnek bir yazılım geliştirilmiştir. Ayrıca kullanılan resimlerde araçların ön ve arkalarından değişik açılarla çekilmiş resimlerinin kullanılmasına özen gösterilmiştir.

Araç resimleri 640x480 piksel boyutlarında çekilmiş veya bu boyuta getirilmiştir. Ancak yazılıma girdi olarak verilen bu resimler, işlenme hızının düşmemesi amacıyla yazılım tarafından 320x240 piksel boyutuna indirgenerek plaka bölgeleri bulunmuş ve daha sonraki modüllerde yine 640x480 piksel boyutlu girdi resmine ait muhtemel plaka bölgesi veya bölgeleri kullanılmıştır.

Üretilen yazılım Microsoft Visual Basic 6.0 dilinde yazılmıştır. Resim 3.1.' de üretilen programın arayüzü gösterilmiştir.

Bu çalışmada yapılan Plaka Tanıma Sistemi üç modülden oluşmaktadır. Bunlar:

- Plaka bölgesinin bulunması,
- Plakadaki karakterlerin ayrıştırılması,
- Plakanın okunması' dır.



Resim 3.1. Üretilen plaka tanıma yazılımının arayüzü

3.1. Plaka Bölgesinin Bulunması

Plaka bölgesinin bulunması çalışmanın ilk ve en önemli safhasıdır. Çünkü bu modülün performansı bundan sonraki modüllerin performansını doğrudan etkileyecektir.

Plaka bölgesinin bulunması birtakım ön işleme algoritmalarından ve resim işleme algoritmalarından oluşmaktadır. Plaka bölgesinin bulunması modülü; Resmi yeniden boyutlandırma, Gri seviyeye indirgeme, Histogram eşitleme, Thresholding algoritması, Smearing algoritması, Morfolojik filtreleme ve Bölge seçim algoritmalarından oluşmaktadır.

3.1.1. Yeniden boyutlandırma

Sisteme girdi olarak verilen araç resimleri 640x480 piksel boyutundadır. Ancak sistemin plaka bölgesinin bulunması ile ilgili kısımda 320x240 piksel boyutunda resimler kullanılmıştır. Bunun nedeni çalışma hızının arttırılmak istenmesidir. 320x240 piksel boyutuna küçültülmüş resim boyutu $\frac{1}{4}$ katına indiği için işleme hızı da dört katına çıkacaktır.

640x320 piksel boyutundaki resim sisteme girer girmez bir kopyası alınmakta ve orijinal resim aşağıdaki kodla $\frac{1}{4}$ boyuta indirgenmektedir.

```
Picture1.PaintPicture Picture1.Picture, 0, 0, Picture1.ScaleWidth / 2,
Picture1.ScaleHeight / 2, 0, 0, Picture1.ScaleWidth, Picture1.ScaleHeight
```

3.1.2. Gri seviyeye indirgeme

Sisteme girdi olarak alınan araç resimleri, 16 milyon renk bilgisini barındırmaktadırlar. Bu 16 milyon renk her piksel için temel renkler olan kırmızı (red), yeşil (green) ve mavi (mavi) renklerin karışımlarından oluşmaktadır. Bu üç ana renk, resim üzerindeki her noktada (piksel), 0-255 arası bir değer ile belirtilmektedirler. Örneğin gerçek mavi rengi, Kırmızı(0), Yeşil(0), Mavi(255) karışımı ifade etmektedir. Bu 16 milyon ihtimal, RGB(0,0,0)-RGB(255,255,255) aralığı değerleri ile tanımlanmaktadırlar.

Gerçek renkli araç resimleri üzerinde amaca yönelik resim işleme işlemleri yapmanın bazı zorlukları vardır [10]. Bu zorluklar şöyle sıralanabilir:

- a. Plaka bölgesinin bulunması için gerçekleştirilen algoritmaların tamamı RGB renk uzaylarının üçünde birden çalışmaktadır. Renkli resimlerde bu durum çözümsüzlüğe neden olabilmektedir. Örneğin iki komşu pikselin kırmızı renk düzlemindeki fark değerleri birbirine çok yakın olabilirken, yeşil renk

düzlemindeki fark değerleri birbirlerine çok uzak olabilmektedir. Bu nedenle yeşil renk düzlemi üzerindeki geçiş bilgisi yakalanabilirken, kırmızı renk düzleminde bu geçişin algılanması güçleşmektedir.

- b. Bir başka problem ise sınıflandırmadır. Gözle birbirine çok yakın olarak nitelendirilen iki renk tonunun, gerçekte rakamsal olarak RGB değerlerinin birbirinden uzak oluşları, sınıflandırmada güçlükler neden olmaktadır [19].

Sayılan nedenle renkli resimlerdeki RGB değerlerinin kullanılması, görüntü analizi açısından karmaşıklığın artmasına neden olmaktadır. Bu nedenle renkli resimden gri seviyeye transfer sorunlarımızın çözümü ve bir standartlaşma olarak karşımıza çıkmaktadır.

Tez kapsamında geliştirilen sistemde 320x240 piksel büyüklüğe yeniden boyutlandırılan resim gri seviyeli forma dönüştürülmektedir (Resim 3.2.).



Resim 3.2. Orijinal 320x240 resim ve gri seviyeye indirgenmiş hali

Resim üzerinde, renk bilgisinin karmaşıklığı ile taşınan fazla ve anlamsız bilgiler, resmin gri seviyeye indirgenmesiyle ortadan kalkmaktadır (Resim 3.2.). Bu indirgeme sistemin çalışma hızını da belirli bir oranda yükseltmektedir. Çünkü RGB ile 3 farklı düzlemde çalışmak yerine, sadece tek gri düzlem üzerinde çalışılmaktadır [20].

Gerçek renk – gri skala dönüşümü; resimdeki RGB değerlerini NTSC formatına çevirerek, renk ve saturasyon değerlerini sıfıra çekerek gerçekleştirilir.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,30 & 0,59 & 0,11 \\ 0,610 & -0,207 & -0,32 \\ 0,21 & -0,52 & 0,31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.1)$$

I ve Q renk fark işaretleridir ve I ve Q' nun sıfıra eşitlenmesi ile, Y hesaplanarak gri skalaya geçilmiş olunur.

$$\begin{aligned} Y &= 0,30R + 0,59G + 0,11B \\ I &= a(R - Y) \\ Q &= b(B - Y) \end{aligned} \quad (3.2)$$

Aşağıda gerçek renkten gri forma dönüştürme için kullanılan kod satırları verilmiştir.

```
For j = 0 To picheight - 1
  For i = 0 To picwidth - 1
    s = GetPixel(h1a, ia, ja)
    Getcolor s, r, g, b
    k = 0.59 * g + 0.3 * r + 0.11 * b
    SetPixelV h, i, j, RGB(k, k, k)
  Next
  Picture1.Refresh
  DoEvents
Next
```

3.1.3. Histogram Eşitleme

Tez çalışması boyunca çekilen resimler günün değişik zamanlarında olmasından dolayı farklı parlaklık kontrast seviyelerine sahiptirler. Resimlerin bazıları kapalı havalarda veya akşam üzeri çekildiğinden dolayı koyu gri tonlar resim üzerinde yoğunlaşmaktadır. Bazı resimler ise direk güneş ışığının arabanın plakasına vurması veya direk fotoğraf makinesi objektifine gelmesi ile plaka bölgesinin parlaklığını aşırı artmıştır. Bu etkenler plaka bölgesinin seçilmesini zorlaştırıcı bir görünüm oluşturmaktadır. Bazı görüntülerde ise, parlaklık / kontrast dengesi bozukluğu, aşırı ton yakınlığı oluşturmakta ve bu durum plakanın resim üzerinde kaybolmasına yol açmaktadır.

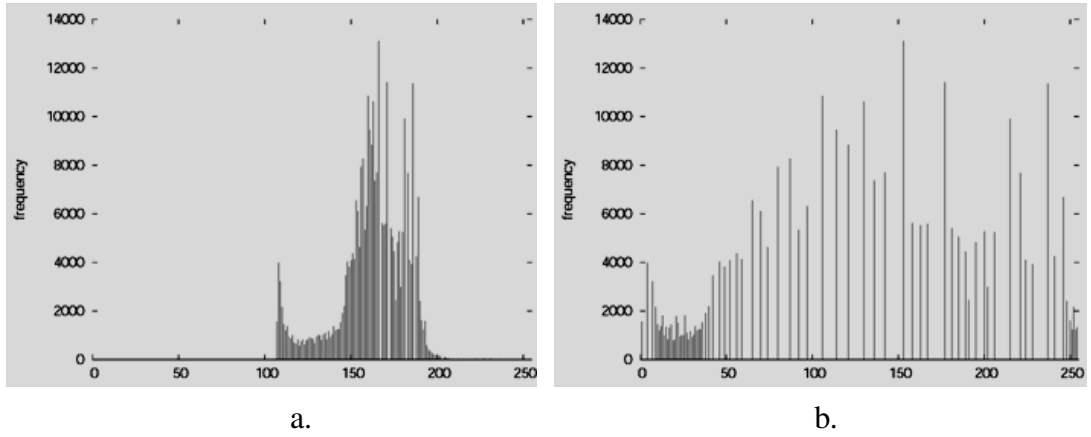
Parlaklık ve kontrast seviyelerindeki bu dengesizlikler nedeniyle oluşan problemleri aşabilmek için, resmin gri ton dağılımının homojen yapılandırılması sağlanmalıdır [9]. Her resimde gri ton dağılımı farklı olduğu için gri ton dağılım skalasında homojenliği sağlamak için histogram eşitleme (“histogram equalization”) yöntemi kullanılmaktadır.

Histogram eşitleme resimlerdeki parlaklık ve kontrastı dengelemek için kullanılan bir tekniktir. Amacı resim içerisinde kullanılan gri tonları tüm spektruma yaymaktır. Histogram eşitlemesi gri seviyeye resmin normalize edilmiş histogramını kullanır.

Bir resmin histogramı aşağıdaki bağıntı ile hesaplanabilir.

$$S_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} \quad (3.3)$$

Burada s orijinal resimdeki her r ye karşılık gelen değerdir. T(r) transfer fonksiyonunu, n_j görüntüdeki j. Seviyesinin kaç kere tekrarlandığını ve n ise görüntüdeki toplam piksel sayısını göstermektedir. Düşük kontrasta sahip bir resmin histogram grafiği Şekil 3.1.a’ daki gibi olsun.



Şekil 3.1. Histogram eşitleme işlemi a. Düşük kontrasta sahip resmin histogramı , b. Histogram eşitleme işleminden sonra resmin histogramı

Bu grafik, gri seviyede olasılık yaklaşımını verdiği sürece, histograma,

$$s = T(r) = \int_0^r p_r(w)dw$$

eşitliğini uygulayarak olasılık eşitlenebilir ve bu yeni piksel değerleri ile resmi yeniden oluşturabilir. Resmin yeni histogramı Şekil 3.3.b'deki gibi olur.

Tez çalışması kapsamında histogram eşitleme işlemi şöyle yapılmıştır: Eşitleme için önce kümülatif histogram hesaplanır. Kümülatif histogramda her değer kendisinden öncekilerin toplamı ile kendisinin toplamına eşittir. Yani:

$$\begin{aligned} KH[0] &= H[0] \\ KH[1] &= H[0] + H[1] \\ KH[2] &= H[0] + H[1] + H[2] \\ KH[3] &= H[0] + H[1] + H[2] + H[3] \\ &\dots \end{aligned} \tag{3.4}$$

.....

(KH: Kümülatif Histogram, H: Histogram)

Kümülatif histogramın sırayla hesaplandığını düşünürsek, $KH[n]$ hesaplanırken $KH[n-1]$ zaten hesaplanmış olacaktır bu nedenle kümülatif histogramı $KH[n] = KH[n-1]$

1] + H[n] şeklinde hesaplamak da mümkündür. Aşağıdaki yordamda kümülatif histogram hesaplamaktadır.

```

For i = 0 To 255
    renk(i) = 0
Next i
For ja = 0 To picheight - 1
    For ia = 0 To picwidth - 1
        sa = GetPixel(ha, ia, ja)
        getcolor sa, ra, ga, ba
        ka = 0.56 * ga + 0.33 * ra + 0.11 * ba
        SetPixelV ha, ia, ja, RGB(ka, ka, ka)
        renk(ka) = renk(ka) + 1
    Next
Next
DoEvents
Next
Picture1.Refresh
Picture2.Picture = Picture1.Image
Picture2.Refresh
For i = 1 To 255
    renk(i) = renk(i) + renk(i - 1)
Next i

```

Kümülatif histogramı elde ettikten sonra, kümülatif histogram değerleri yeni resimde olması istenilen maksimum renk değeriyle çarpılıp, resimdeki toplam nokta sayısına bölünerek normalize edilirler. Bu işlemi yapan yordam aşağıdadır.

```

For i = 0 To 255
    renk(i) = Round((renk(i) * 255 / (val(picwidth) * val(picheight))), 0)
Next i

```

Böylece normalize olmuş histogram elde edilmiş olur. Bu histogramı kullanarak resmin renk değerleri tekrar güncellenir. Yani her noktanın değeri ile normalize

olmuş histogramdaki karşılığına bakılır ve noktanın rengi histogramdaki karşılığına eşitlenir. Aşağıda bu işlemi yapan yordam görülmektedir.

```

For ja = 0 To picheight - 1
  For ia = 0 To picwidth - 1
    sa = GetPixel(ha, ia, ja)
    getcolor sa, ra, ga, ba
    ka = 0.56 * ga + 0.33 * ra + 0.11 * ba
    SetPixelV ha, ia, ja, RGB(renk(ka), renk(ka), renk(ka))
  Next
DoEvents
Next

```

Resim 3.3.' te tez kapsamında kullanılan bir araç resminin gri seviyeye indirgenmiş halindeki resmi ve histogram eşitleme işlemi uygulandıktan sonraki resmi verilmiştir.



a.

b.

Resim 3.3. Araç resmi üzerinde histogram eşitleme işleminin uygulanması, a. Gri seviyeye indirgenmiş araç resmi b. Histogram eşitleme işlemine tabi tutulmuş araç resmi

Yukarıdaki şekilde görüldüğü üzere histogram işleme işleminden geçirilmiş araç resmi aydınlanmış plaka bölgesi çok daha belirgin hale gelmiştir. Bu şekilde resim siyah beyaz forma dönüştürülmeden önce daha kararlı hale getirilmiştir.

3.1.4. Siyah-beyaz forma dönüştürme (Thresholding)

Bir resmin ön ve arka plan bilgilerinin çıkartılması ve birbirinden ayrılması amacıyla, resmi siyah ve beyaz olmak üzere iki renge dönüştürme işlemine siyah/beyaz forma dönüştürme (“binarization”) denmektedir.

Ön plan, arka plan ayrımı iyi bir eşikleme (“Thresholding”) değeri saptaması ile gerçekleştirilmektedir. Gri seviye değerleri, eşik değerini geçemeyen pikseller arka plan, gecebilen pikseller ise ön plan üzerinde durmaktadır. Bu şekilde sınıflandırma yapılarak resim siyah beyaz forma dönüştürülmektedir.

Thresholding gri seviye resmi sadece siyah ve beyaz renkten oluşan iki renkli resme çevirir [19]. Thresholding görüntü işleminin kullanıldığı çalışmalarda ön işlem algoritması olarak kullanılır.

Plaka bölgesinin bulunması için bir ön işlem algoritması olarak kullanılan Thresholding yöntemi araç resimde art alan ile resimdeki nesnelerin ayırt edilmesini sağlar. Resmin gri seviye histogramında “T” Threshold değerinden büyük veya küçük olmasına göre piksellerin değeri siyah yada beyaza çekilerek, art alan ve nesneler ayrıştırılır. Thresholding ayrıca, kompleks renk bilgisine sahip olan resimlerin işlenmesini kolaylaştırmak için kullanılmaktadır. RGB renk modunda beyaz renk her üç kanalın da 255 olduğu, siyah renk ise her kanalın 0 olduğu durumdur. Thresholding koyu ton renkleri siyaha açık ton renkleri beyaz renge dönüştürmektedir. Görüntü işlenirken siyah renk 1, beyaz renk 0 olarak kodlanmaktadır.

Genellikle resim işleme yazılımlarında Threshold eşik değeri 256 renk değerinin tam ortası olan 128 olarak alınmaktadır. Geliştirilen sistemin eşik değeri ise değişken tanımlanmıştır. Renk tonu ortalaması 100’eşit veya 100’ü geçen araç resimlerinde yani parlaklığı yüksek resimlerde plaka bölgesinin parlama ihtimalini ortadan kaldırmak için eşik değeri yüksek tutulmuş ve 128 olarak belirlenmiştir. Renk tonu ortalaması 100 değerinin altında kalan resimlerde yani koyu ve karanlık tonların

hakim olduđu resimlerde plaka bölgesindeki beyaz renkli kısımların ayırt edilebilmesi için eşik değeri daha düşük tutulmuş ve 110 olarak belirlenmiştir. Bu rakamların elde edilebilmesi için 150' den fazla araç resminde testler yapılmış ve Resim 3.4.' te görüldüğü gibi en iyi siyah-beyaz dönüşüm sonuçları bu değerlerde alınmıştır.



Resim 3.4. Histogram eşitleme işleminden sonra siyah-beyaz forma dönüştürülmüş resim

Aşağıda siyah-beyaz dönüşüm için kullanılan kodlar verilmiştir.

```
aa = 0
```

```
bb = 0
```

```
cc = 0
```

```
For j = 1 To picheight - 1 Step 2
```

```
    For i = 1 To picwidth - 1 Step 2
```

```
        s = GetPixel(h, i, j)
```

```
        r = getred(s)
```

```
        g = getgreen(s)
```

```
        b = getblue(s)
```

```
        aa = aa + r
```

```
        bb = bb + g
```

```

        cc = cc + b
    Next i
Next j
aa = Round((aa / (val(picheight / 2) * val(picwidth / 2))), 0)
bb = Round((bb / (val(picheight / 2) * val(picwidth / 2))), 0)
cc = Round((cc / (val(picheight / 2) * val(picwidth / 2))), 0)
rrr = Round(((aa + bb + cc) / 3), 0)
If rrr >= 100 Then
    rrr = 128
Else
    rrr = 110
End If
For j = 1 To picheight - 1
    For i = 1 To picwidth - 1
        s = GetPixel(h, i, j)
        r = getred(s)
        g = getgreen(s)
        b = getblue(s)
        If r < rrr And g < rrr And b < rrr Then
            r = 0
            g = 0
            b = 0
        Else
            r = 255
            g = 255
            b = 255
        End If
        SetPixelV h, i, j, RGB(r, g, b)
    Next i
Next j

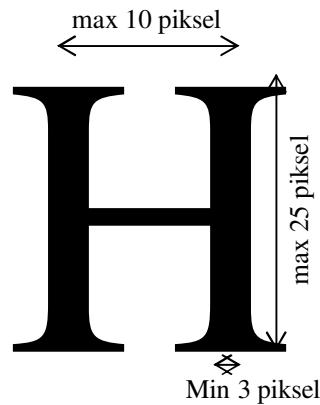
```


3.1.5. Morfolojik filtrelemeler

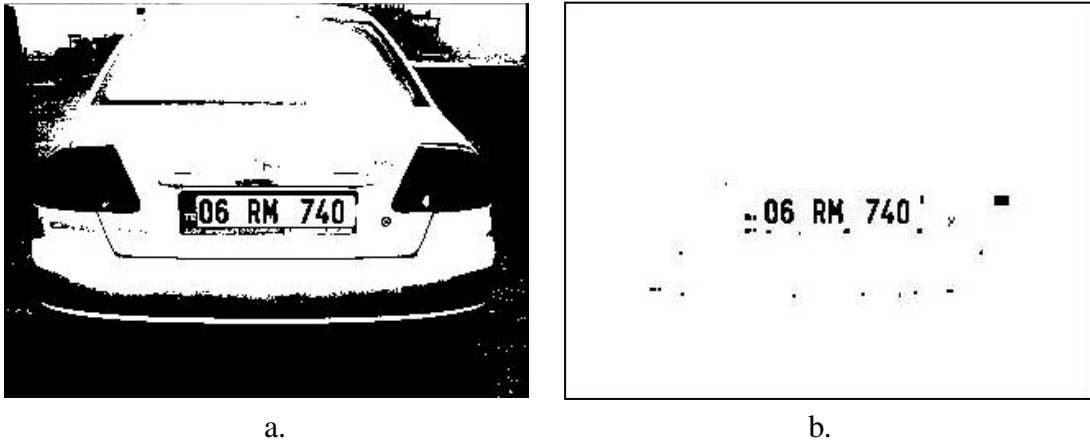
Morfolojik filtreleme plaka bölgesi bulma işleminde işlemleri siyah-beyaz forma dönüştürülmüş (ikilik resim) araç resmindeki plaka bölgesi olma ihtimali olmayan bölgelerin görüntüden elenmesi yani resmin gereksiz kısımlarının atılması için filtreleme yapılması işlemidir. Tez çalışması kapsamında da plaka bölgesi bulma modülünde morfolojik filtreleme işlemleri aynı amaç için kullanılmıştır.

Siyah-beyaz forma dönüştürülmüş araç resminde plaka bölgesi olamayacak kadar yatayda ve dikeyde geniş siyah bölgeler veya yine plaka karakterlerinin bir kısmı olamayacak kadar küçük siyah bölgeler mevcuttur. İşte tez çalışma sırasında kullanılan filtreleme işlemleri bu bölgeleri resim üzerinden yok ederek plaka bölgesi dışında mümkün olduğu kadar az siyah bölgenin bulunmasını sağlamaktır.

Sistemin Plaka yeri bulma modülünün girdi olarak aldığı araç resimleri 320x240 piksel boyutundadır ve bu resimler 2-3 metre mesafeden çekilmiştir. Bu veriler ışığında gidildiğinde plaka üzerindeki bir karakterin maksimum yatay genişliği en fazla 10 piksel olmaktadır (Şekil 3.2.) ve dikey olarak yüksekliği en fazla 25 piksel olmaktadır. Aynı zamanda hiçbir karakterin et kalınlığı 3 pikselden daha ince olmamaktadır. Bu verilere göre bu şartları taşımayan resmin siyah bölgeleri morfolojik filtreleme yöntemiyle elenmekte ve resmin sarf plaka bölgesiyle bırakılmasına çalışılmaktadır (Resim 3.5.).



Şekil 3.2. Plaka üzerindeki karakterlerin maximum ve minimum ölçüleri



Resim 3.5. a. Siyah-beyaz resim b. Morfolojik filtreleme işleminden geçmiş resim

Morfolojik filtreleme işlemi yapan kod bloğu aşağıda verilmiştir.

```

For j = 0 To picheight - 1
    ss = 0
    For i = 0 To picwidth - 1
        s = GetPixel(h, i, j)
        r = getred(s)
        s1 = GetPixel(h, i + 1, j)
        r1 = getred(s1)

        If r = 0 Then
            ss = ss + 1
        End If
        If ss = 1 And r1 = 255 Then
            SetPixelV h, i, j, RGB(255, 255, 255)
            ss = 0
        End If
        If r = 255 Then ss = 0
        If i = picwidth - 1 And ss >= 10 Then
            For ii = i - ss To i
                SetPixelV h, ii, j, RGB(255, 255, 255)
            
```

```

                Next ii
                ss = 0
            End If
        Next i
    Next j
    ss = 0
    For i = 0 To picwidth - 1
        ss = 0
        For j = 0 To picheight - 1
            s = GetPixel(h, i, j)
            r = getred(s)
            If r = 0 Then
                ss = ss + 1
            End If
            If ss = 1 And r1 = 255 Then
                SetPixelV h, i, j, RGB(255, 255, 255)
                ss = 0
            End If
            If r = 255 Then ss = 0
        Next j
    Next i

```

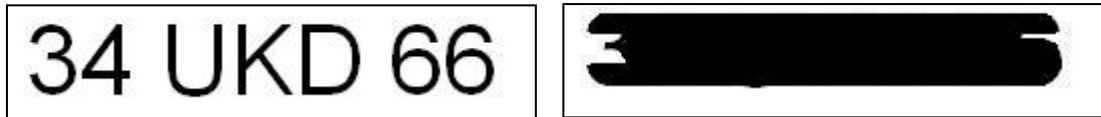
3.1.6. Smearing algoritması

İkilik resme uygulanan smearing algoritması, metin içeren görüntüden metin alanlarını arka alandan ayırt etmek için kullanılan uygun bir algoritmadır. Bu algoritma kullanılırken karakterlerin yatay ekseninde düz olması, verimi artırmak için gereklidir.

Algoritma görüntü hem yatay hem de dikey olarak taranmaktadır. Yatayda ve dikeyde yan yana dizilmiş beyaz piksellerin sayısının önceden belirlenmiş bir eşik değerinden küçük olması durumunda bu piksellerin renginin siyaha çevrilmesi

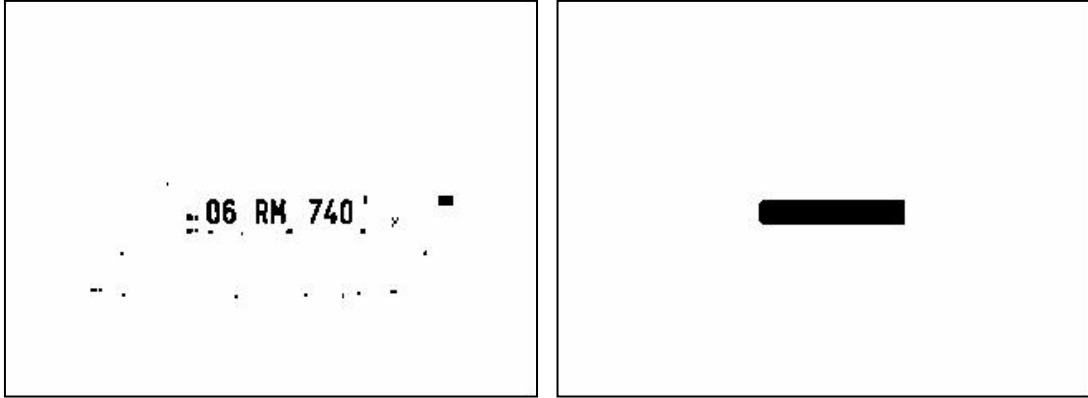
prensibine dayanmaktadır. Görüntüdeki bu beyaz bloklar bu eşik değerinden büyükse aynen kalmaktadır. Smearing algoritması karakterin boyutlarına ve görüntüdeki gürültüye duyarlı olduğundan eşik değeri atanırken görüntü örneklerinin incelenmesi gerekmektedir.

Smearing algoritmasının yatay çizgiler halinde uygulaması Şekil 3.3.' te gösterilmiştir. Eşik değeri karakterler arası güvenli açıklık göz önünde bulundurularak belirlenmiştir. Plaka genişliği yaklaşık olarak 7 veya 8 karakterdir ve karakterler arası açıklığın maksimum olduğu bölge harf kümesiyle rakam kümeleri arasındaki mesafedir. Bununla birlikte, kimi karakterlerin kendi bölgelerinde sola ve sağa dayalı bölgeler olmaktadır. Buda 320x240 piksel boyutlarındaki bir araç görüntüsündeki bir plakada maksimum 30 piksele karşılık gelmektedir. Böylece yatay smearingde yan yana beyaz renk gruplarının bulunduğu piksel sayısı 30'in altındaysa bu piksellerdeki renkler siyah yapılmaktadır. Bu işlemden sonra plaka karakterlerini oluşturan harf ve rakamların da iç kısımları siyah dolacağı dikeyde ard arda gelen beyaz piksellerin siyah yapılması için eşik değerinin 5 olması yeterli gelmiştir.



Şekil 3.3. Yatay Smearing işlemi

Smearing algoritması ile siyah renge boyanan muhtemel plaka bölgeleri dışında kalan diğer bölgelerin de yok edilmesi gerekmektedir. Bu işlem için yine morfolojik filtreleme işlemine ihtiyaç vardır. Smearing algoritması ile doldurulan muhtemel plaka bölgelerinin yatayda genişliği 320x240 resimde en az 50 piksel olmaktadır. Dikeyde yüksekliği ise en az 8 piksel olmaktadır. Bu şartların sağlamayan resim üzerindeki bölgeler filtreleme işlemiyle yok edilmekte ve böylece araç resmi üzerinde sadece muhtemel plaka bölgesi ya da bölgeleri kalmaktadır (Resim 3.6.). Resimdeki tüm gürültüler yok edilmektedir.



Resim 3.6. Smearing ve filtreleme işlemlerinden geçirilerek resim üzerinde plaka bölgesinin bulunması

Aşağıda smearing algoritmasına ait kodlar verilmiştir.

```

For j = 0 To picheight - 1
    ss = 0
    anah = 0
    For i = 0 To picwidth - 1
        s = GetPixel(h, i, j)
        r = getred(s)
        s1 = GetPixel(h, i + 1, j)
        r1 = getred(s1)
        If r = 0 And r1 = 255 Then anah = 1
        If anah = 1 And r = 255 Then ss = ss + 1
        If r = 255 And r1 = 0 And ss <= 30 Then
            For ii = i - ss + 1 To i
                SetPixelV h, ii, j, RGB(0, 0, 0)
            Next ii
            ss = 0
            anah = 0
        End If
    End For
    If ss > 30 Then
        anah = 0
    End If
End For

```

```

                ss = 0
            End If
        Next i
    Next j

    ss = 0
    anah = 0
    For i = 0 To picwidth - 1
        ss = 0
        anah = 0
        For j = 0 To picheight - 1
            s = GetPixel(h, i, j)
            r = getred(s)
            s1 = GetPixel(h, i, j + 1)
            r1 = getred(s1)
            If r = 0 And r1 = 255 Then anah = 1
            If anah = 1 Then ss = ss + 1
            If r = 255 And r1 = 0 And ss <= 5 Then
                For ii = j - ss + 1 To j
                    SetPixelV h, i, ii, RGB(0, 0, 0)
                Next ii
                ss = 0
                anah = 0
            End If
            If ss > 5 Then
                anah = 0
                ss = 0
            End If
        Next j
    Next i

```

Smearing ve filtreleme işlemlerinin ardından plaka bölgesi araç resmi üzerinde

bulunmuştur. Artık bu plaka bölgesinin resim üzerinden doğru bir şekilde kopartılması gerekir. Bunun için plakanın en sol ve en üst köşe noktaları $(x1,y1)$, en sağ ve en alt noktaları $(x2,y2)$ tespit edilir (Şekil 3.4.), en bu koordinatların oluşturduğu dikdörtgen çerçeve, resme ilk olarak girdi olarak verilen 640×480 piksel büyüklüğündeki resimden oranlanarak kopartılır (Resim 3.7.) ve plaka bölgesi bir sonraki modül olan karakterlerin ayrıştırılması modülüne girdi olarak sunulur.



Şekil 3.4. Plaka bölgesinin koordinatlarının belirlenmesi



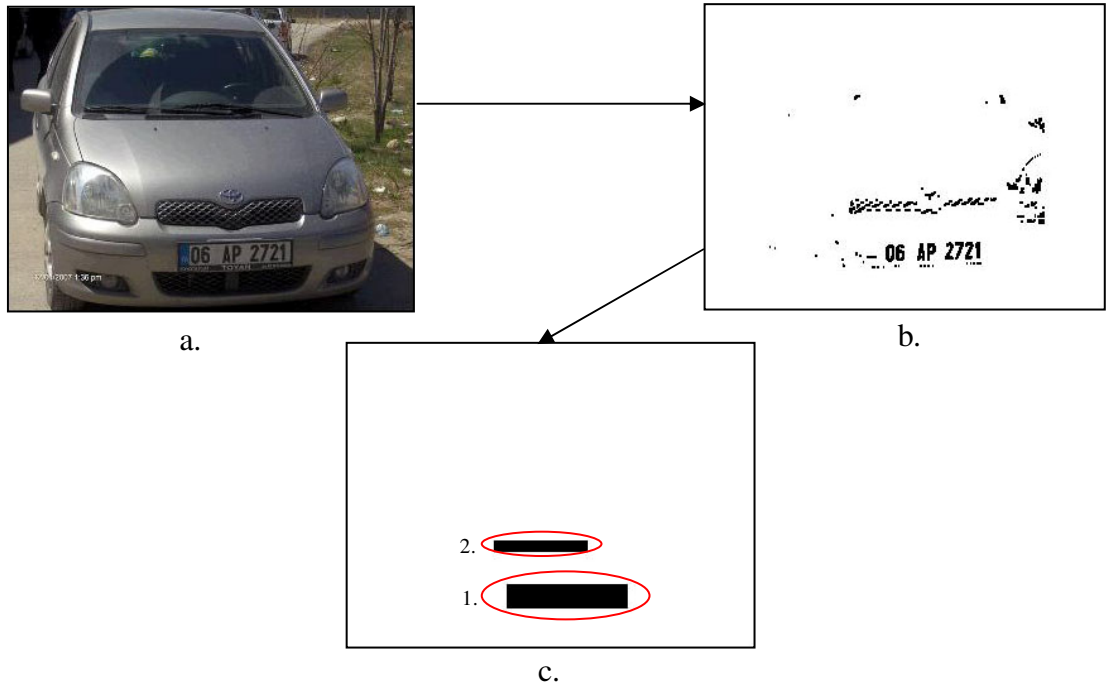
Resim 3.7. Plaka bölgesinin kopartılması

Bazı araç resimlerinde smearing ve filtreleme işlemlerinden sonra birden fazla aday plaka bölgesi kalabilmektedir (Resim 3.8.). Geliştirilen sistem böyle durumlarda

işaretlenen aday plaka bölgelerinin en boy oranlarına bakmaktadır. Çekilen araç resimleri üzerinde yapılan incelemelerde plakaların en boy oranı ortalama 5,5 ile 6,5 arasında değişmektedir. Ancak resmin işlenmesi sırasında plaka üzerindeki istenmeyen aşınmaların veya fazla alan bırakılması ihtimaline karşılık oluşturulan yazılımda bu oran 5 ile 7 arasında kabul edilmiştir.

Smearing algoritmasının uygulanmasının ardından resim üzerinde birden fazla alan kaldıysa tasarlanan sistemde şöyle bir yol izlenmektedir:

- Aday plaka bölgelerinin en boy oranlarına bakılır, belirlenen limitler dışında kalan alan varsa bu alanlar elenir,
- Eğer belirlenen limitler içerisinde birden fazla aday plaka bölgesi kaldıysa bu aday bölgeler 6,0 en/boy ortalamasına yakınsaklıklarına göre derecelendirilir,
- 6,0 ortalamaya en yakın aday bölgeden başlanarak bölge okunmaya çalışılır, eğer 1. aday plaka bölgesi okunamadıysa 2. aday plaka bölgesine bakılır ve bu böylece devam eder.



Resim 3.8. Birden fazla aday plaka bölgesi işlemi, a. Aracın resmi, b. Smearing algoritması öncesi son durum, c. İki aday bölge derecelendirme işlemi

3.2. Plakadaki Karakterlerin Ayrıştırılması

Plakadaki karakterlerin ayrıştırılmasında modülünde, bir önceki modülün koparttığı plaka bölgesi üzerindeki karakterlerin resim bilgisi olarak ayrıştırılması ile ilgili yapılan işlemler anlatılacaktır. Bu bölümde plakanın ölçülendirilmesi ve karakterlerin sağ-sol, üst-alt sınırlarının belirlenmesi üzerinde durulacaktır.

Bir önceki modülden bu modüle sisteme ilk giriş yapan plaka resmi, sisteme girdi olarak alınan 640x480 piksel boyutundaki araç resmine ait plaka bölgesidir. Bir önceki modülde bulunan 320x240 resme ait siyah-beyaz plaka bölgesi kullanılmaz. Bunun nedeni, plaka boyutunun karakter tanımada dezavantaj oluşturacak seviyede küçük olması ve resmin tümüyle birlikte siyah-beyaz dönüşüme tutulmasından dolayı renk ayrımının karakter tanımada kullanılmasını güçleştirecek şekilde ayrıntısız olmasından kaynaklanmaktadır.

Karakter ayrıştırma modülüne girdi olarak kullanılan plaka resmi renklidir. Bu nedenle plaka resmi önce Gri seviyeye indirgeme, Histogram eşitleme, Siyah-beyaz, dönüşüm, morfolojik filtrelemeler işlemlerini içine alan bir önışleme işlemlerinden geçirilir. Daha sonra plaka bölgesi üzerindeki karakterler alanları ölçülendirilir ve smearing algoritması ve komşu bölge seçim algoritmaları ile plaka karakterlerinin yerleri tek tek belirlenir ve bir sonraki modüle karakter resimleri aktarılır.

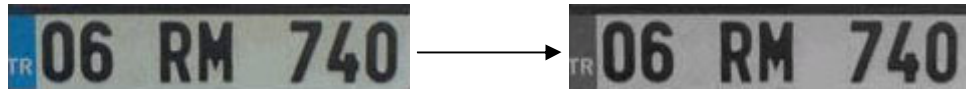
3.2.1. Önışleme algoritmaları

Önışleme işlemleri gri seviyeye indirgeme, histogram eşitleme, ikili resme çevirme ve morfolojik filtreleme işlemlerini içermektedir. Bu işlemlerden bir önceki modülde ayrı ayrı başlıklar halinde bahsedildiği için bu bölümde ayrı başlıklar kullanılmayacak ve önışleme algoritmaları başlığı halinde anlatılacaktır.

Daha önceden kullanılan bu yöntemlerin tekrar niye kullanıldıkları sorusu akla gelebilir. Bunu nedenleri şunlardır:

Plaka bölgesi bulma algoritmasında, işlemleri hızlandırmak için 320x240 resim kullanılmıştır ve buradaki plaka bölgesi küçük ve ayrıntısızdır. Bu bölgenin direk olarak karakter ayrıştırma ve tanıma algoritmalarında kullanılması performansı düşürecektir. İkinci nedeni ise bir önceki plaka bölgesi bulma modülündeki algoritmalar resmin tümü için kullanıldıklarından bulunan eşik değerleri resmin tamamı için kullanılmıştır. Plaka bölgesindeki karakterlerin arka plandan daha net ve daha ayrıntılı olarak ayrılması için kullanılan aynı tür algoritmalar içerisinde kullanılacak eşik değerlerinin doğal olarak değişmesi gerekecektir.

Gerçek renkli plaka resmi önce bilinen gri seviyeye indirgeme katsayı hesaplamalarıyla gri seviyeye çekilir (Resim 3.9.).



Resim 3.9. Plaka resmini gerçek renkten gri seviyeye indirilmesi

Plaka bölgesindeki siyah beyaz kontrastını dengelemek ve plaka üzerinde bulunabilecek yabancı maddeleri (örneğin çamur gibi) ve plaka karakterleri dışındaki karakterleri arka plana yaklaştırmak için histogram eşitlemesi işlemine tabi tutulur (Resim 3.10.).



Resim 3.10. Histogram eşitlemesi işleminden geçirilmiş plaka resmi

Histogram eşitleme plaka üzerindeki karakterlerin arka plandan ayrıştırılması için kötü bir sonuç veriyor gibi algılanabilir. Ancak bu sadece bir göz yanılgısıdır. Büyük bir çoğunlukla siyah-beyaz forma dönüştürme işleminde histogram eşitleme işleminden geçirilmiş plaka resimlerinin, geçirilmeyen plaka resimlerine göre daha doğru sonuçlar ürettiği Resim 3.11.' da net olarak görülmektedir.

Siyah beyaz dönüşüm için kullanılan algoritma yine thresholding algoritmasıdır ancak plaka resminin renk ağırlığı değiştiği için, daha açık bir ifadeyle resimdeki açık tonların artmasından dolayı threshold eşik değeri daha düşük olarak belirlenmiştir. Bunun nedeni mümkün olduğu kadar plaka karakterleri dışında bulunan gürültülü bölgelerin resimden yok edilmesidir. Eşik değeri resim renk değeri ortalaması 95' in üzerinde olan resimlerde yani 255 beyaz sınırına daha yakın resimlerde 105, altında olan resimlerde yani 0 siyah sınırına daha yakın olan koyu tonlu resimlerde 95 olarak belirlenmiştir.

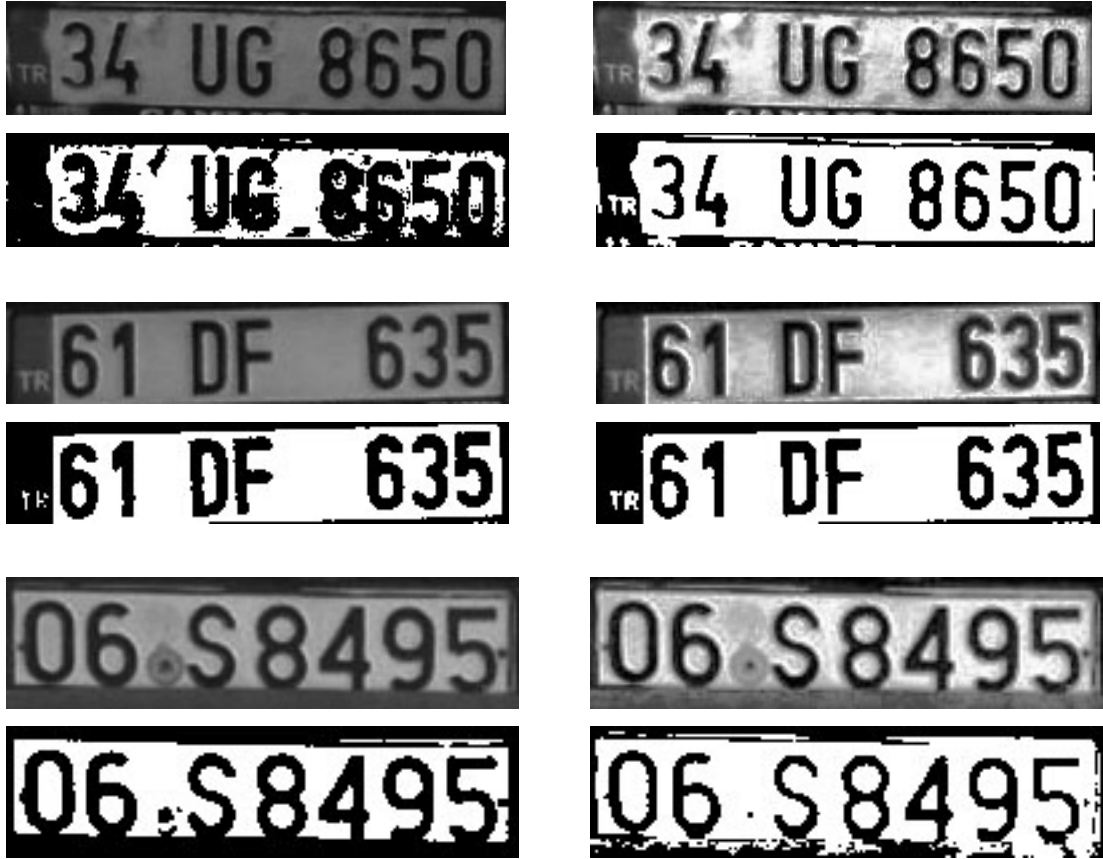
```

For j = 1 To picheight - 1 Step 2
    For i = 1 To picwidth - 1 Step 2
        s = GetPixel(h, i, j)
        r = getred(s)
        aa = aa + r
    Next i
Next j
aa = Round((aa / (val(picheight / 2) * val(picwidth / 2))), 0)
If aa >= 95 Then
    rr = 105
Else
    rr = 95
End If
For j = 1 To picheight - 1
    For i = 1 To picwidth - 1
        s = GetPixel(h, i, j)
        r = getred(s)
        If r < rr Then
            r = 0; g = 0; b = 0
        Else
            r = 255; g = 255; b = 255
        End If
        SetPixelV h, i, j, RGB(r, g, b)
    Next i
Next j

```

Next i

Next j



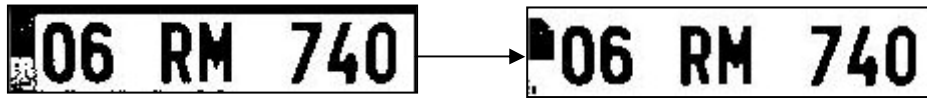
a.

b.

Resim 3.11. a. Gri seviye ve siyah-beyaz forma dönüştürülmüş plaka resimleri, b. Histogram eşitlenmiş ve siyah-beyaz forma dönüştürülmüş plaka resimleri.

Morfolojik filtrelemeler plaka karakterlerinin ayrıştırılması modülünde hayati bir önem kazanmaktadır. Çünkü karakterlerin sağlıklı bir biçimde zeminden ve birbirlerinden ayrıştırılabilmeleri için resimdeki gürültülerin mümkün olduğu nispette karakterlere dokunmadan yok edilebilmeleri gerekmektedir. Morfolojik filtreleme işlemlerinde birtakım istatistiki ölçülendirmeler kullanılmaktadır. Örneğin Türk plaka standartlarındaki sivil plakalar 7-8 karakterden oluşmaktadır. Karakterler arasındaki boşluklarda dikkate alındığında plakalardaki hiçbir karakterin yatay genişliği plakanın toplam genişliğinin $1/7'$ ini geçmemektedir. Bu nedenle yatayda

1/7 plaka ölçüsünü aşan tüm siyah satırlar beyaz yapılmaktadır. Bu işlemden sonra plaka üzerindeki gürültüler büyük oranda yok olacaktır. Aynı zamanda plakadaki yatay ve dikey uzunluğu 2 pikselin altında genişliğe sahip siyah bölgeler de beyaza dönüştürülmektedir. Bu yöntemlerle siyah-beyaz resimdeki gereksiz ayrıntılar ve gürültüler çok büyük oranda yok edilmektedir (Resim 3.12.)



Resim 3.12. Morfolojik filtreleme işlemi

Kullanılan morfolojik filtrelemeye ait kod bloğu aşağıda verilmiştir.

```

For j = 0 To picheight - 1
    ss = 0
    For i = 0 To picwidth - 1
        s = GetPixel(h, i, j)
        r = getred(s)
        ii = i
        If r = 0 Then
            ss = ss + 1
        Else
            If ss >= Round(picwidth / 7, 0) Then
                For ii = i - ss + 1 To i
                    SetPixelV h, ii, j, RGB(255, 255, 255)
                Next ii
                ss = 0
            End If
        End If
    End For
    If r = 255 Then ss = 0
    If i = picwidth - 1 Then
        If ss >= 10 Then

```

```

        For ii = i - ss + 1 To i
            SetPixelV h, ii, j, RGB(255, 255, 255)
        Next ii
        ss = 0
    End If
End If
s1 = GetPixel(h, i + 1, j)
r1 = getred(s1)
If r = 0 And r1 = 255 And ss <= 2 Then
    For ii = i - ss To i
        SetPixelV h, ii, j, RGB(255, 255, 255)
    Next ii
    ss = 0
End If
Next i
Next j

ss = 0
anah = 0
For i = 0 To picwidth - 1
    ss = 0
    anah = 0
    For j = 0 To picheight - 2
        s = GetPixel(h, i, j)
        r = getred(s)
        s1 = GetPixel(h, i, j + 1)
        r1 = getred(s1)
        If (r = 255 And r1 = 0) Or (j = 0 And r = 0) Then anah = 1
        If anah = 1 Then ss = ss + 1
        If r = 0 And r1 = 255 And ss <= 2 Then
            For ii = j - ss To j
                SetPixelV h, i, ii, RGB(255, 255, 255)
            
```

```

        Next ii
        ss = 0
        anah = 0
    End If
    If ss > 3 Then anah = 0
Next j
Next i

```

3.2.2. Karakter en/boy orantısı ve ölçülendirme

Türk plaka standartlarındaki sivil plakalar üç kısımda oluşur. İlk kısımda plakanın bağlı bulunduğu il kodunu gösteren iki haneli sayı, ikinci kısımda 1,2 veya 3 adet harf ve üçüncü kısımda 2 ile 4 haneli sayı bulunur (bkz. Bölüm 1, Sayfa 3).

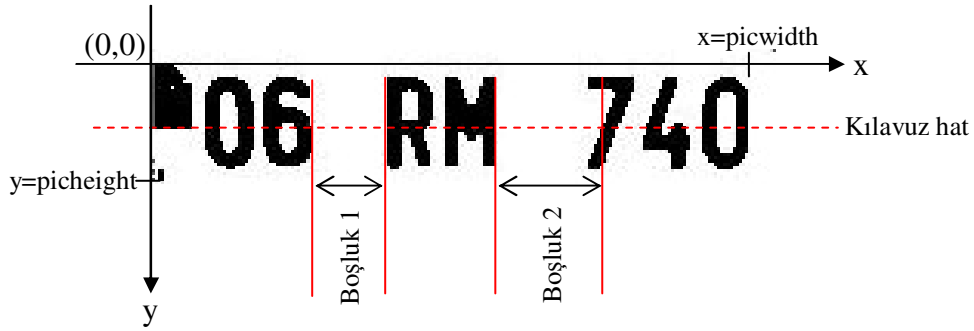
Karakterlerin diziliminde şöyle bir kombinasyon mevcuttur:

- İkinci kısımda 1 adet harf varsa üçüncü kısımda 4 adet rakam bulunur
- İkinci kısımda 2 adet harf varsa üçüncü kısımda 3 veya 4 adet rakam bulunur
- İkinci kısımda 3 adet harf varsa üçüncü kısımda 2 adet rakam bulunur

Bu veriler yapılacak işlemlerle ilgili güzel bir ipucu vermektedir. Şöyle ki; plakalarda yataydaki en büyük boşluklar harf kümesi ile rakam kümeleri arasından bulunmaktadır (Şekil 3.5.).

Burada dikkat edilmesi gereken nokta boşlukları resmin dikey ekseninin hangi noktasında arayanacağıdır. Bunun için bir klavuz hat belirlenmesi ve o hat boyunca boşluk aranması gerekmektedir. Bunun için en uygun satır resmin dikey uzunluğunun aritmetik ortasıdır.

Yani $y = \text{picheight} / 2$ dikey noktasında, $x=0$ to $x=\text{picwidth}$ doğrusu boşlukların bulunmasında doğru sonucu verecektir.



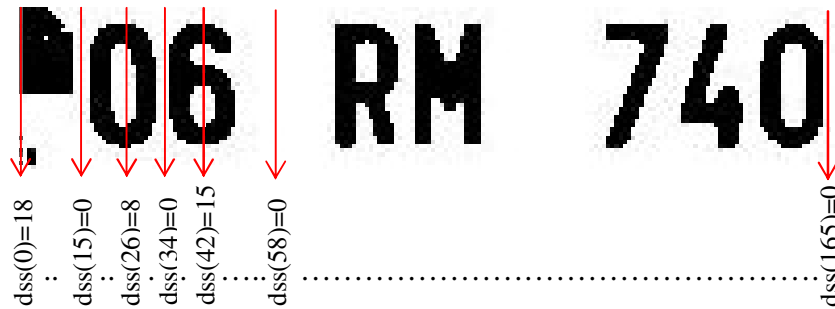
Şekil 3.5. Plaka üzerindeki boşlukların yakalanması

Plakadaki bu boşluklar bulunduktan sonra boşlukları referans olarak dikey tarama işlemine geçilir. Dikey tarama;

$x=0$ ' dan $x=picwidth$ değerine ilerlerken her x değeri için

$y=0$ ' dan $y=picheight$ değerine kadar taranmasıdır.

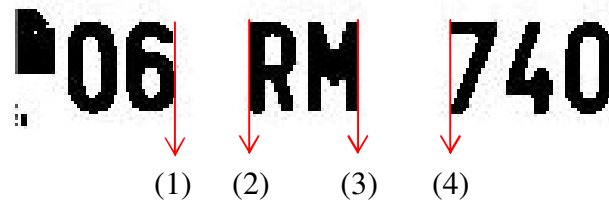
Burada diziler kullanılmaktadır. Şöyle ki; her x değeri için taranan sütundaki siyah piksellerin sayısı 2' yi geçerse (0-2 arası gürültü kabul edilmektedir) o sütunda karakter olduğu kabul edilir böylece o sütununu (y eksenini) taşıdığı siyah piksel sayısı dizi değişkenine aktarılır. (Şekil 3.6.).



Şekil 3.6. Plakanın dss() dizisine dikey değerleriyle sayısal olarak aktarılması

Boşluklar referans alınarak yapılan dikey tarama işleminde harf kümesinin yani plakanın ikinci kısmının yatayda başladığı ve bittiği noktaların koordinatlarını dizi

değişkenler içerisindeki değerler kullanılarak kolaylıkla bulunabilir [22]. Bu uzun boşluklardan 1.boşluktan sola doğru yapılan dikey tarama işlemi plakanın iki haneli il kodunu oluşturan birinci kısmının yataydaki bitiş noktasına ait koordinatı, 2. boşluktan sağa doğru yapılan dikey tarama işlemi plakanın üçüncü kısmının başlangıç koordinatını da vermektedir (Şekil 3.7.)



Şekil 3.7. Plaka üzerindeki kısım başlangıç bitiş noktalarının yakalanması

- | | |
|-------------------------------|-----------------------------------|
| 1) 1. kısım bitiş noktası | 2) İkinci kısım başlangıç noktası |
| 3) İkinci kısım bitiş noktası | 4) Üçüncü kısım başlangıç noktası |

Bu dört başlangıç ve bitiş noktasının bulunmasını sağlayan kod bloğu aşağıda verilmiştir.

```

baraj = 2
For i = 0 To 500
    dss(i) = 0
Next i
ss = 0
For i = 0 To picwidth - 1
    For j = 0 To picheight - 1
        s = GetPixel(h, i, j)
        r = getred(s)
        If r = 0 Then ss = ss + 1
    Next j
    dss(i) = ss
    If ss <= baraj Then
        dss1(i) = 0
    Else

```

```

        dss1(i) = 1
    End If
    If dss1(i) = 0 Then
        dss1(i) = dss1(i - 1) + 1 ' + i
    End If
    ss = 0
Next i

    anah = 0
    For i = 0 To picwidth - 1
        If dss1(i) = 1 And dss1(i + 1) = 1 And anah = 0 Then
            aa = i
            anah = 1
        End If
        If dss1(i) = 1 And dss1(i + 1) = 1 Then bb = i + 1
    Next i

    eb = dss1(aa)
    For i = aa To bb
        If dss1(i) > eb Then
            eb = dss1(i)
            m = i
        End If
    Next i

    For i = m - eb To m
        dss1(i) = 0
    Next i

    eb1 = dss1(aa)
    For i = aa To bb
        If dss1(i) > eb1 And dss1(i) < eb Then

```

```

        eb1 = dss1(i)
        n = i
    End If
Next i

```

```

If m > n Then
    gec = eb
    eb = eb1
    eb1 = gec
    gec = n
    n = m
    m = gec
End If

```

```

kp(1) = m - eb
kp(2) = m
kp(3) = n - eb1
kp(4) = n

```

Plaka resmine ait sütunların taşıdıkları piksel değerleri dss() dizisinde tutulmaktadır. dss() dizisinin 0' dan farklı değer taşıyan indeksleri bire bir dss1() dizisine "1" değeri olarak aktarılır. dss1() dizisinin diğer indekslerinin değerleri ise "0" olarak kalır. Böylece plakanın dikeydeki boş ve karakter içeren sütunlarına ait yatay koordinat bilgileri dss1() dizisine aktarılmış olur (Şekil 3.8.).

dss1(x) 1.. 0. 1.. 0. 1.. 0.... 1.. 0. 1.. 0... 1.. 0. 1.. 0. 1..

Şekil 3.8. Plakanın diziye 1 ve 0 olarak aktarılması

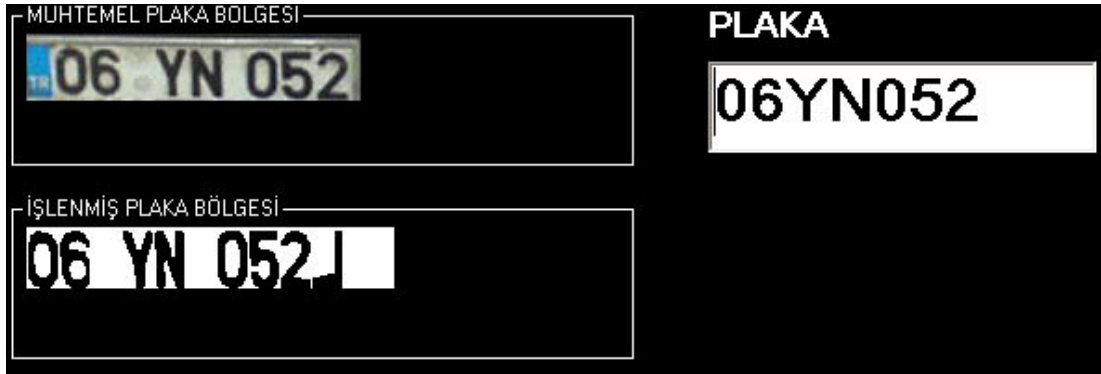
Dss1() dizisi artık plakanın karakterlerinin yatay sınırlarına ait koordinat bilgilerini taşımaktadır. Buna göre 1. büyük boşluğundan önceki dizideki iki adet “1” bloğu plakanın 1. kısmına ait yani il kodunu gösteren kısmın başlangıç ve bitiş değerlerini göstermektedir. Dizinin daha önce “1” değeri içeren indeksleri varsa bu bloğun gürültü içerdiği anlaşılır ve resimden bu bölüm atılır. 1. büyük boşluk ile 2. büyük boşluk arasındaki kısım harflerin bulunduğu kısmın başlama ve bitiş noktalarını vermektedir. Aynı zamanda orta kısımda bulunan harf sayısı da öğrenilmiş olur.

Daha önce de anlatıldığı gibi orta kısımdaki harf sayısı son kısımda bulunan rakam sayısını da vermektedir. Buna göre plakanın 3. kısmında bulunan rakamların da başlangıç ve bitiş noktalarını dizi kullanarak tespit etmiş olunur (Şekil 3.9.). Ortadaki kısmın taşıdığı harf sayısının 2 olduğu durumda hatırlanacağı üzere sonda 3 veya 4 sayı bulunmakta idi. Bu problem yaratan bir durumdur ancak eldeki plaka ölçü bilgileri bu noktada bize yardımcı olmaktadır. Şöyle ki; plakanın 3 kısmındaki karakterler arası boşlukların ortalaması eşik değeri olarak kullanılabilir. 2 harfli plakalarda 3. kısımda 4. karakter bilgisi varsa bu kısmın 3. karakteri ile 4. karakteri arasındaki boşluğa bakılır. Bu boşluk ortalama karakterler arası boşluğun 1 piksel fazlasından daha fazlaysa 4. karakter bölgesi dikkate alınmaz. Eğer arasındaki boşluk ortalama ölçüler içindeyse 4. karakter alınır. Eğer her şeye rağmen 4 karakter, bir karakter değil de resimdeki bir gürültü ise bu Karakter tanıma modülünde ayrılır (Resim 3.13.). Tüm bu işlemlerin sonucunda plakadaki tüm karakterlerin yatay ekseninde başlangıç ve bitiş noktaları tespit edilmiş olur (Resim 3.14.).

Dss1(x) 111110000111110000111110000000011111100011111100000000001111100011111000011111

Anlamsız Bölge **0 6 R M 7 4 0**

Şekil 3.9. Plakanın karakterlerinin yatayda başlangıç ve bitiş noktalarının diziyeye yüklenmesi



Resim 3.13. Plaka tanıma sisteminde Karakter ayrıştırma modülünün tespit edemediği plaka 3. kısım 4. karakter gürültüsünün karakter tanıma modülünde elenmesi



Resim 3.14. Plakadaki tüm karakterlerin yatay ekseninde başlangıç ve bitiş noktalarının tespit edilmesi

Plakanın karakterlerinin yatay ekseninde başlangıç ve bitiş noktaları artık belirlenmiştir. Bundan sonraki işlem her karakterin dikeydeki başlangıç ve bitiş noktalarının işaretlenmesidir. Bu işlem smearing algoritması ile gerçekleştirilir.

Her karakterin yataydaki minimum ve maksimum koordinatları (x_{min}, x_{max}) bellidir. Her karakter aralığı $y=0$ ' dan $y=picheight$ noktasına kadar yatay taramaya tabi tutularak karakterin dikeydeki en küçük ve en büyük değerleri bulunmuş olur (Resim 3.15.). Böylece her karakterin yatayda ve dikeyde başlangıç ve bitiş noktaları belirlenerek karakterler birbirlerinden ayrıştırılmış olur ve her bir karakterin koordinatları son modül olan karakter tanıma modülüne gönderilir.



Resim 3.15. Karakterlerin yatay ekseninde taranarak dikey sınırlarının belirlenmesi

Aşağıda karakterlerin dikey sınırları belirlenmesi için oluşturulan kod bloğu verilmiştir. Buradaki xp() dizisi karakterlerin yatay sınırlarını ya() dikey sınırlarını tutan dizilerdir.

```

c = 0
anah = 0
For k = 0 To ss - 1
    i = Round((xp(k) + xp(k + 1)) / 2, 0)
    For j = 0 To picheight - 1
        s = GetPixel(h, i, j)
        r = getred(s)
        s1 = GetPixel(h, i, j + 1)
        r1 = getred(s1)
        If j = 0 And r = 0 Then anah = 1
        If r = 0 Then anah = 1
        If r = 0 And anah = 1 Then
            c = c + 1
            If (r = 0 And r1 = 255) Or (r = 0 And j = picheight - 1) Then
                If c >= 15 Then
                    If j = picheight - 1 Then
                        ya(k) = j
                    Else
                        ya(k) = j + 1
                    End If
                    yu(k) = j - c
                    If yu(k) < 0 Then yu(k) = 0
                    SetPixelV h, i, yu(k), RGB(200, 20, 10)
                    SetPixelV h, i, ya(k), RGB(200, 20, 10)
                    anah = 0
                    c = 0
                Else
                    c = 0
            End If
        End If
    Next j
Next k

```

```

                                anah = 0
                                End If
                            End If
                        End If
                    Next j
                    c = 0
                    anah = 0
                Next k

```

Karakterlerin dikey sınırları belirlenirken bazı özel durumlar oluşabilmektedir. Deforme olmuş plakalarda karakterlerin bazı ince kısımları önişleme algoritmalarından geçerken yok olabilmektedir (Resim 3.16.). Bu duruma karşı şöyle bir önlem geliştirilmiştir. Koptuğu tespit edilen karakterin (tarama esnasında tespit edilen aralık) kopan parçası ile birlikte toplam karakter yüksekliği diğer karakterlerin yüksekliklerinin ortalamasının 5 piksele kadar yakınsaklığında ise sistem karakterin deformasyondan dolayı koptuğuna karar vermekte ve tek karakter olarak sınırlarını belirlemektedir. Aksi durumda ayrı duran bölgeyi sistem gürültü olarak algılamakta ve resimden bu kısım atılmaktadır.



Resim 3.16. Deforme olmuş karakter ve sistem tek karakter olarak kabul edilmesi

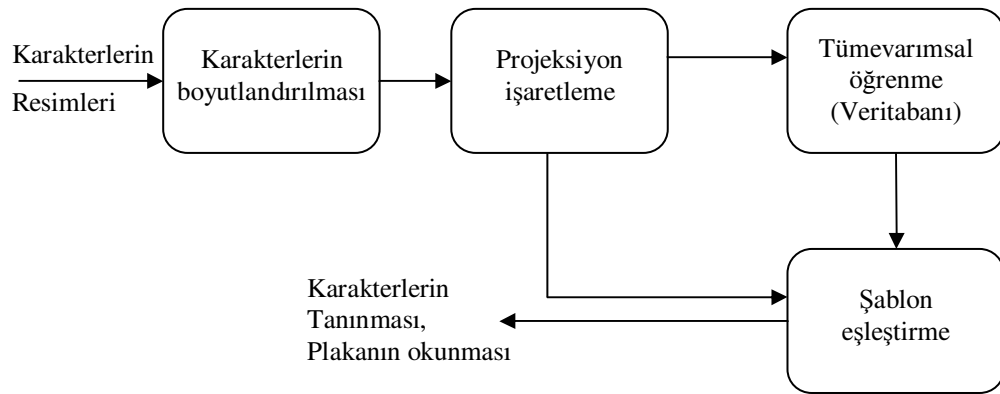
3.3. Plakanın Okunması (Karakter Tanıma)

Plakanın okunması modülü plakadaki karakterlerin tanınması için kullanılan algoritmaları içerisinde barındırmaktadır. Karakter tanıma (“Character Recognition”), karakterlerin ayrıştırılması modülünden gelen karakterlere ait koordinat bilgilerine göre, seçilmiş plaka üzerindeki karakter görüntülerini (Şekil 3.23.) çıkararak girdi olarak kullanılmaktadır. Bu modülün amacı, karakter resimlerini metinsel karşılıklarına çevirmektir.

Tez çalışması kapsamında doğrudan karakter tanıma yapan bir sistem geliştirilmiştir. Bu sistem dört aşamadan oluşmaktadır. Bunlar:

- Karakterlerin boyutlandırılması
- Projeksiyon işaretleme
- Tümevarımsal öğrenme
- Şablon eşleştirme

Projeksiyon işaretleme, tümevarımsal öğrenme ve şablon eşleştirme aşamaları iç içe çalışan birbirleriyle sürekli etkileşim halinde olan algoritmalarıdır (Şekil 3.10.). Ancak konunun daha iyi anlaşılabilmesi için bu üç algoritma ayrı ayrı anlatılacaktır. Tümevarımsal öğrenme sistemin karakterleri öğrenmesi için kullanılan bir algoritmadır ve projeksiyon işaretleme algoritması ile ortak çalışır. Şablon eşleştirme öğrenilmiş karakterlerin karşılaştırma metodu ile tanınması için kullanılan bir algoritmadır ve yine projeksiyon işaretleme ile etkileşimli olarak çalışır [22].



Şekil 3.10. Karakter tanıma modülünün çalışması

3.3.1. Karakterlerin boyutlandırılması

Karakter ayrıştırma modülünden gelen karakterlere ait koordinatlara göre sistem bu karakterlerin her birini resim olarak plaka üzerinden kopartıp belleğe alır. Karakter tanıma algoritması olarak şablon eşleştirme metodu kullanıldığı için, birinci kural

tüm karakterlerin en ve boylarının eşit olmasıdır. Eğer eşitleme işlemi yapılmazsa şablon eşleştirme metodunu verimli sonuçlar vermesi mümkün değildir. Buna göre, tasarlanan sistemde karakter şablonları 16x32 piksel boyutunda kullanılmıştır. Plaka resmi üzerinden kopartılan karakterlerin tamamının boyutu önce 16x32 piksel büyüklüğe eşitlenir (Şekil 3.11.). Daha sonra projeksiyon işaretleme aşamasına geçilecektir.



Şekil 3.11. Plakadan kopartılan karakterlerin boyutlarının eşitlenmesi (16x32 piksel)

Aşağıda Karakterin karakterlerin plakadan kopartılması, karakterlerin yeniden boyutlandırılması ile ilgili oluşturulan kod bloğu verilmiştir.

anah = 0 'Her karakter için Yüst ve Yalt değerleri tam olarak ayarlanıyor

For k = 0 To ss - 1

For j = yu(k) To ya(k)

For i = xp(k) To xp(k + 1)

s = GetPixel(h1, i, j)

r = getred(s)

If anah = 0 And r = 0 Then

yu(k) = j

anah = 1

End If

If anah = 1 And r = 0 Then ya(k) = j

Next i

Next j

anah = 0

Next k

anah = 0 *'Her karakter için Xsol ve Xsag değerleri tam olarak ayarlanıyor*

For k = 0 To ss - 1

For i = xp(k) To xp(k + 1)

For j = yu(k) To ya(k)

s = GetPixel(h1, i, j)

r = getred(s)

If anah = 0 And r = 0 Then

Xsol(k) = i

anah = 1

End If

If anah = 1 And r = 0 Then Xsag(k) = i

Next j

Next i

anah = 0

Next k

For j = 0 To picheight - 1 ***'PICTURE1 BOSALTILYOR***

For i = 1 To picwidth - 1

SetPixelV h, i, j, RGB(255, 255, 255)

Next i

Next j

c = 1

For k = 0 To ss - 1 ***'Picture2' den Picture1' e 16x32 boyutunda harfler kopyalanıyor***

StretchBlt Picture1.hdc, c, 2, 16, 32, Picture2.hdc, Xsol(k), yu(k), Xsag(k) -

Xsol(k) + 1, ya(k) - yu(k) + 1, vbSrcCopy

c = c + 17

Next k

3.3.2. Projeksiyon İşaretleme

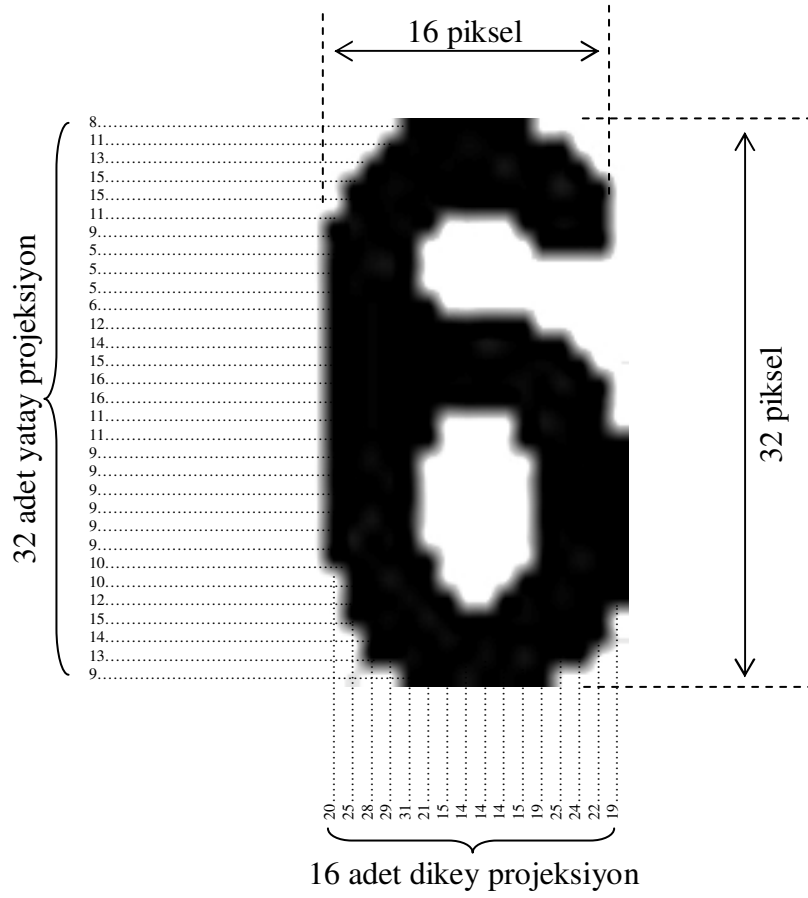
Önceki çalışmalardaki şablon eşleştirme yönteminin kullanımlarında karakterin pikselleri matris olarak ifade edilmekte yani karaktere ait her piksel matrisin bir değerini oluşturmakta olduğu görülmektedir. Örneğin karakterin boyutu 16×32 ise aynı boyutta matris tanımlanarak, $16 \times 32 = 512$ ayrı değer takip etme zorunluluğu vardır. Bu durumun karakter tanıma modülünün, karakteri tanımak için kullandığı süreyi arttıracak aşıkardır [22].

Tez çalışması kapsamında projeksiyon işaretleme yöntemi kullanılarak şablon çıkarma ve şablon eşleştirme yolu yöntemi tercih edilmiştir. Böylece bir karakter için 512 yerine $16 + 32 = 48$ adet değer takip edilmesi yeterli olacaktır.

Projeksiyon işaretlemeyle dayalı şablon çıkarma yönteminde her bir karakterin yatay ve dikey projeksiyonlarına bakılır (Şekil 3.12) ve bu projeksiyon yansımaları sayısal olarak elde edilip bir diziye atılır [22]. Eğer karakter sistem tarafından öğrenilecekse oluşturulan dizi o karaktere ait bir kayıt altına (şablon) veritabanına yazılır. Eğer karakter okunacaksa çıkarılan projeksiyon veritabanındaki şablonla karşılaştırılır.

Yatay ve dikey projeksiyona bağlı şablon çıkarma yönteminde karakter önce y_{min} ' den y_{max} ' a kadar yatay olarak satır satır taranır ve her satırın taşıdığı siyah piksel sayısı yatay projeksiyonun 32 adet değerini oluşturur. Daha sonra karakter x_{min} ' den x_{max} ' a kadar dikey olarak sütun sütun taranır ve her sütunun taşıdığı siyah piksel sayısı dikey projeksiyonunun 16 adet değerini oluşturur. Böylece her karakter için 48 adet değer yatay ve dikey projeksiyonla çıkartılmış olur (Şekil 3.12.). Çıkartılan değerler işlenmek üzere dizi değişkenine aktarılır.

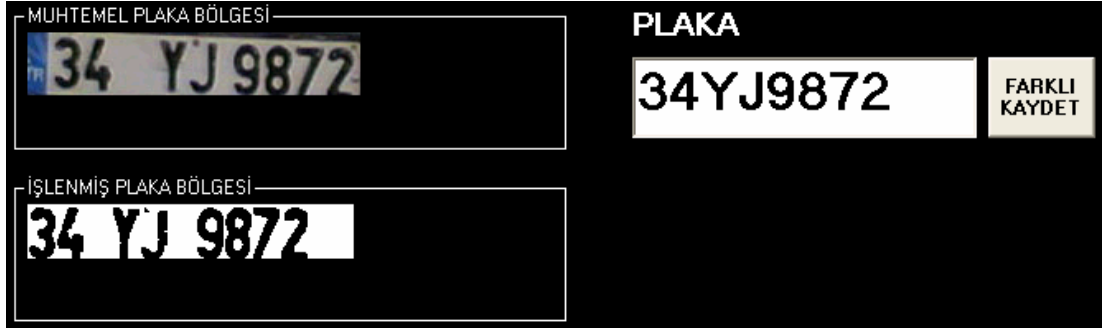
Sonuç olarak plakadaki her karakterin şablonu çıkartılmış ve dizi değişkenine aktarılmıştır. Eğer karakter okuma işlemi yapılacaksa her bir karaktere ait şablon kendi grubundaki (harf veya rakam) karakterlere veritabanından karşılaştırılır. Eğer öğrenme işlemi yapılacaksa veritabanına kendi grubuna şablon kaydedilir [22].



Şekil 3.12. Karakterin yatay ve dikey projeksiyonlarının çıkartılması

3.3.3. Tümevarımsal öğrenme ve şablon eşleştirme




Tümevarımsal öğrenme esasında karakterlerin veritabanına kendisine ait şablonlarıyla kaydedilmesi işlemidir. Ancak karakterler sistem çalışmadan önce peşinen veritabanına öğretilmez. Tasarlanan sistemde, sistemin çalışması esnasında plakanın karakterleri ayrıştırılıp resimleri çıkartıldığında plakası bir yandan sistem tarafından okumaya başlanmıştır bir yandan da insan tarafından veritabanına yazılır. Veritabanında her karakterden örnekleme sayısı birkaç taneyi geçtikten itibaren sistem bulduğu şablonlarla veritabanını karşılaştırır ve plakayı doğru okuma oranı gittikçe artmaya başlar. Bir taraftan da sistem tarafından yanlış okunan veya okunamayan karakterler kullanıcı tarafından düzeltilip veritabanına kaydedilir (Resim 3.17.). Bu şekilde tasarlanan sistemde sürekli bir öğrenme süreci vardır.



Resim 3.17. Plakanın sisteme öğretilmesi

Her bir karakter aynı sayıda şablon olarak veritabanına yazılmaz. Kaydetme işlemi sistem yanlış okuma yaptıkça veya okuyamadıkça tekrarlanır. Bilgisayarın algılama mekanizması insan gibi olmadığından 3-9, 5-6, 0-8 gibi birbirine benzeyen karakterlerin okunması zordur ve kolaylıkla bu karakterler sistem tarafından karıştırılabilmektedirler (Çizelge 3.1). Bu yüzden bu karakterlerden veri tabanındaki örnek sayısı ne kadar fazla olursa sistemin yapacağı hata olasılığı da o kadar düşer. Her karakterin okuma zorluğuna göre yeterince sayıda veritabanına yazıldığında yani sisteme öğretildiğinde okuma işlemi hatasız olarak yapılacaktır.

Çizelge 3.1. Şablon metodunun uygulanmasıyla elde edilen arşivlenmiş yanlış sonuçlar ve tekrarlı öğrenme sonucu doğru okuma

Çıkarılmış karakter			
Şablon modunun uygulanmasıyla elde edilen sonuç	9	6	0
Tekrarlı öğrenme sonucu doğru tanıma	3	5	8

Tümevarımsal öğrenmenin isminden gelen özelliği her karakter için sistemin oluşturduğu 48 adet değerden tek bir sonuca varılmasıdır. Şablonu oluşturan 48 adet

projeksiyon değeri 1 tek karakteri ifade eder ve öğrenme her karakter için 48 ayrı değerle o karakterin öğrenilmesi yani parçalardan tüme varılmasıdır.

Öğrenme işleminin farklı bir özelliği daha vardır. Plakadaki harf ve rakam grupları karakter ayrıştırma modülü tarafından önceden tespit edilmişti. Sistemin okuma performansını arttırmak hata yapma ihtimalini temelli ortadan kaldırmak amacıyla plakaya ait sayılar farklı tabloya, harfler farklı tabloya kaydedilir. Bu şekilde okuma işleminde de her karakter kendi sınıfına ait veritabanı tablosundan aranacak ve daha kısa sürede daha doğru sonuca ulaşılabilecektir. Özellikle B ve 8, O ve sıfır(0), G ve 6 ve bunun gibi karakterlerde yapılabilecek hataları ortadan kaldırmak için bir harf şablonu tablosu ve sayı şablonu tablosu oluşturulmuştur. Bu şekilde harfler ve rakamlar bağımsız olarak tanınacak ve O ve 0 (sıfır), 8 ve B arasında olduğu gibi karakterler arasındaki karışıklık çözülecektir.

Tasarlanan sistemin karakter okumayla ilgili son kısmı şablon eşleştirmedir. Şablon eşleştirme plakanın okunmasını sağlayan algoritmadır. Ayrıştırılan plaka karakterlerinin projeksiyon işaretleme ile şablonu çıkartılır ve bu karaktere ait şablon daha önceden öğretilmiş veritabanındaki kendi karakter grubuyla, yani harf karakterleri harf şablonları tablosundaki, sayı karakterleri sayı şablonları tablosundaki kayıtlarla karşılaştırılarak okuma işlemi gerçekleştirilir.

Harf guruplarıyla sayı gurupları arasında yeterince boşluk bulunmayan, aşırı deforme olmuş veya çok kirli plakaların bazılarında harf sayısı sistem tarafından tespit edilememiştir. Harf sayısı tespit edilemediği için plakanın 3. kısmındaki rakamların sayısı da sistem tarafından kararsız bir algoritmayla alınmıştır. Bu durumda okuma işlemi için çıkartılan şablonlar her iki veritabanı tablosuyla karşılaştırılıp doğru sonuca ulaşılmaya çalışılmıştır.

4. SONUÇ VE ÖNERİLER

Literatürdeki araç plaka tanıma uygulamaları incelendiğinde tasarlanan sistemin uygun fiziki şartlar altında oldukça iyi sonuçlar ürettiğini söylemek yanlış olmayacaktır. Uygun fiziki şartlardan kasıt, çekilen araç resminin mümkün olduğu nispette aracın önünden veya arkasından 60 ila 90 derece arası bir açıdan olması, plaka bölgesinin yer düzlemine en fazla 20 derecelik bir farkla paralellik göstermesidir. Literatürdeki plaka tanıma sistemlerinin hemen hemen tamamında başarı oranının yüksek olması için bahsedilen fiziki şartların sağlanması gerekmektedir. Sistem bir otoparkın girişinde kullanılması durumunda yerleştirilecek kameranın araç plakalarının ortalama yerden yüksekliği seviyesine yerleştirilme performansı arttıracak diğer bir etkidir.

Tasarlanan sistemin literatürdeki örneklerine göre en büyük farkı ve aynı zamanda en büyük avantajı karakterlerin tanınması ve plakanın okunması modülünde ortaya çıkmaktadır. Projeksiyon işaretlemeye dayalı şablon eşleştirme yöntemi karakterlerin okunma hızını arttırmıştır. Aynı zamanda sistem sürekli bir öğrenme halinde olduğu için hatalı sonuç üretme ihtimali oldukça düşürülmüştür. Tasarlanan sistemin özellikle kamu veya özel alanlara ait otoparkların giriş çıkışlarında %100' e yakın bir performansla çalışacağını tahmin etmek yanlış bir yaklaşım olmayacaktır. Çünkü sisteme kaydedilen bir araç bir sonraki gelişinde rahatlıkla tanınabilecektir.

Sistemin testlerinde, çalışması esnasında ilk 150 aracın plakası sisteme kaydedilmiştir. Bu 150 araca ait plakaların karakterlerinin %70' ine yakını sistem kendisi tanımış ancak veritabanını güçlendirmek için yine de kayıt işlemi yapılmış sisteme her bir karakterler Çizelge 4.1' de verilen sayıda veritabanına kaydedilmiştir.

Karakterler daha önce de bahsedildiği gibi yatay ve dikey projeksiyon değerleriyle, yani her bir karakter 48 adet değerle kaydedilmiştir. Oluşturulan sistem her an öğrenmeye açık bir sistem olduğu için buradaki her bir karakter için yapılan kayıt sayıları sabit değildir, her zaman sistem yeni kayıtlarla zenginleştirilip daha yetenekli

bir okuma niteliğine kavuşturulabilir.

Çizelge 4.1. Yapılan testlerde sistemin öğretilmesi için kullanılan karakter sayıları

Rakam	Adet	Harf	Adet	Harf	Adet
0	148	A	52	L	18
1	79	B	27	M	17
2	80	C	19	N	23
3	166	D	24	O	2
4	140	E	30	P	14
5	76	F	22	R	23
6	145	G	16	S	14
7	53	H	17	T	14
8	66	I	1	U	17
9	58	J	10	V	25
		K	19	Y	15
				Z	26

Veritabanını fazlaca şişirmek bir dezavantaj olarak düşünülebilir ama geliştirilen sistemde bu problem, programın çalışmaya başladığı sırada veritabanının belleğe yüklenmesiyle ortadan kaldırılmıştır. Üstelik tüm veritabanı belleğe yüklendiği sırada yapılan ölçümde, veritabanının bellekte 0,5 Mbyte' tan daha az yer tuttuğu tespit edilmiştir.

Daha sonra sistemde kaydı bulunmayan günün değişik saatlerinde, değişik iklim koşullarında ve değişik mekanlarda uygun mesafeden ve açıdan çekilmiş 200 aracın resmi sisteme girdi olarak verilmiş ve oldukça memnun edici sonuçlar alınmıştır. Bu 200 aracın tanınmasıyla ilgili deneysel sonuçları gösterir tablo Çizelge 4.2' de verilmiştir.

Çizelge 4.2. Deneysel sonuçlar

	Sayı	Yüzde
Yakalanmış plaka	192/200	%96
Doğru okuma (Tüm resimler)	185/200	%92,5
Doğru okuma (Yakalanmış plakalar)	185/192	%96,35
Tek karakteri yanlış okunanlar	3/200	%1,5

Sonuçlar incelendiğinde özellikle yakalanmış plakalardaki plaka okuma performansı oldukça yüksektir. Bunun nedeni sürekli öğrenebilen bir sistem olmasından kaynaklanmaktadır. Sistemin toplam başarısı tüm girdi yapılan resimlerdeki hatasız plaka okuma oranı olan %92,5 olduğunu söylemek gerekir.

Oluşturulan plaka yazılımı Microsoft Visual Basic 6.0 programlama dilinde geliştirilmiştir. Oluşturulan yazılım Intel Centrino 1.6 Ghz işlemciye, 512 Mbyte RAM belleğe sahip bir dizüstü bilgisayarda yazılmış ve testleri yapılmıştır. Yapılan testlerde sistemin bu bilgisayarda resmin sisteme girmesinden, plakanın okunması arasındaki işlemleri yani bir aracın plakasını 3,5 - 4,5 saniye arasında okuduğu yapılan ölçümlerde tespit edilmiştir. Daha hızlı ve bellek kapasitesi daha yüksek bir bilgisayarda bu sürenin 2,5 - 3,5 saniye arasına düşeceği öngörülmektedir.

Tasarlanan sistem geliştirmeye açık bir sistemdir. Sistem bir taşıt resminin plakasının okunması için kullandığı tüm zamanın 2/3' ünü plakanın resimden kopartılması sırasında harcamaktadır. Bu işlemler için daha hızlı algoritmalar geliştirilmesi durumunda çalışma hızı daha da artacağı öngörülmektedir. Yine de, bu sistemin geliştirilmesi için, daha başka araştırmalar yapılmalıdır. Sistem bazı durumlarda çok iyi sonuçlar üretememektedir. Örneğin plaka bölgesinin yarısının gölgede kaldığı,

diğer yarısının parladiđı veya plakanın aşırı parladiđı, plakanın fazla eğik olduđu resimlerde plaka bölgesi yakalanamamıştır. Deforme olmuş veya çok kirli plakalarda da okuma problemi ortaya çıkmaktadır. Bu tür problemleri düzeltebilecek algoritmalar tasarlanan sisteme monte edilerek başarı oranı çok daha arttırılabilir.

KAYNAKLAR

- [1] Setchell, C. J., "Applications of Computer Vision to Road-traffic Monitoring", Phd Thesis, *University of Bristol*, Bristol, 170 (1997).
- [2] Dawson, J. A. L., Gatling, I., "Electronic road pricing in Hong Kong", *Transportation Research*, 20:129-134 (1986).
- [3] Gillan, W. J., "PROMETHEUS-Reducing traffic congestion by advanced technology In Roads and Traffic 2000", *International Road and Traffic Conference*, Berlin, 111-115 (1988).
- [4] Stoelhurst, H. J., Zandbergen, A. J., "The Development of a road pricing system in the Netherlands", *Traffic Engineering and Control*, 31:66-71 (1990).
- [5] Williams, P. G., Kirby, H. R., Montgomery, F. O., Boyle, R. D., "Evaluation of video-recognition equipment for number-plate matching", *2nd. International Conference on Road Traffic Monitoring*, London, 13-17 (1989).
- [6] Fahmy, M. M., "Computer vision application to automatic number plate recognition", *36th. International Symposium on Automotive Technology and Automation*, Aachen, 24-26 (1993).
- [7] Auty, G., Corke, P., Dunn, P., Jensen, M., Macintyre, I., Mills, D., Nguyen, H., Simons, B., "An image acquisition system for traffic monitoring applications", *SPIE: Cameras and System for Electronic Photography and Scientific Imaging*, 2416:118-133 (1995).
- [8] Barroso, J. R., Dagless., E. L. and Bulas-Crus, J., "Number Plate Reading Using Computer Vision", *IEEE- International Symposium on Industrial Electronics ISIE'97*, Universidade do Minho, (1997).
- [9] Draghici, S., "A Neural Network Based Artificial Vision System for License Plate Recognition System", *Inter. Journal of Neural Systems*, 8:113-126 (1997).
- [10] Cinsdikici, M.G., "Akıllı Trafik Sistemleri için Araç Plakası Tanıma Modülü Tasarımı ve Geliştirilmesi", Doktora Tezi, *Ege Üniversitesi Fen Bilimleri Enstitüsü*, İzmir, 13-38 (2004).
- [11] Huang J., Kumar S.R., Mitra M., Zhu W.J., Zabih R., "Color-Spatial Indexing and Applications", *International Journal of Computer Vision*, 3:245-268 (1999).
- [12] Chun B.T., Yoon H.S., "A Method to Extract Vehicle Number Plates by Applying Signal Processing Techniques", *Journal of Korea Electronic Engineering Association*, Korea, 92-100 (1993).
- [13] Johnson M., "Image Transform-Hough Transform", Intelligence Systems Lecture 10, Technical Report ID:59.318, Computer Science, Massey University Albany, <http://www.massey.ac.nz/notes/59318/110.html> , (1996).
- [14] Min J.Y., Choi J.U., "Development of License Plate Extraction Algorithm Using Knowledge", *5.th World Congress on Intelligent Transport Systems Proceeding*, Seoul Korea, 40-52 (1998).
- [15] Çamaşırcıoğlu, E., "Araç Plakası Tanıma ve Algılama", Yüksek Lisans Tezi, *Ankara Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 14-20 (2007).

- [16] Cheokman, W.U., Lei C.O., Chan H.W., Tong S.K., Kengchung N.G., , "A Macao License Plate Recognition System", *Faculty of Information Technology*, Macao University of Science and Technology, 3-5 (2005).
- [17] Coetzee C., Botha C.P., Weber D., "PC Based Number Plate Recognition System", *Proc. IEEE International symposium on Industrial Electronics*, University of Stellenbosch, 604-610 (1998).
- [18] Parisi R., Claudio E.D.D, Lucarelli G., Orlandi G., "Car Plate Recognition by Neural Networks and Image Processing", *IEEE Proceedings of Int. Symposium on Circuits and Systems*, Zhejiang University, 195-198 (1998).
- [19] Özbay S., "Automatic Vehicle Identification By Plate Recognition", Yüksek Lisans Tezi, *Gaziantep University Graduate School Of Natural & Applied Sciences*, Gaziantep, 30-38 (2006).
- [20] Cinsdikici M., Yüçetürk A.C., Öztürk Y., "Partitioned Generalized Euclidean Distance", *The Proceedings of the Twelfth International Symposium on Computer and Information Sciences (ISCIS'XII)*, Antalya, 623-628 (1997).
- [21] Gradshteyn, I. S., Ryzhik, I. M., "Hessian Determinants." *14.314 in tables of Integrals and Products*, Academic Press, 1069 (2000).
- [22] Yalım B., Doğan N., "Türk Taşıt Plaka Standartları İçin Plaka Tanıma Sistemi", *Bilişim Teknolojileri Dergisi*, 1:47-50 (2008).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : YALIM, Beytullah
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 05.02.1979 MustafaKemalpaşa
 Medeni hali : Evli
 Telefon : 0 (312) 358 43 77
 Faks : 0 (312) 278 00 80
 e-mail : yalimegemen@hotmail.com

Eğitim Derece

Eğitim Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Kocaeli Üni./T.E.F/Bilg. Öğrt. Bölümü	2000
Lise	Demirtaşpaşa Teknik Lisesi	1995

İş Deneyimi Yıl

Yıl	Yer	Görev
2000-2001	M.K.Coşkunöz ATL	Bilgisayar Öğretmeni
2001-2008	Güvercinlik ATL,TL,EML	Bilgisayar Bölüm Şefi

Yabancı Dil

İngilizce

Yayımlar

1. Yalım B., Doğan N., “Türk Taşıt Plaka Standartları İçin Plaka Tanıma Sistemi”, **Bilişim Teknolojileri Dergisi, Gazi Üniversitesi Bilişim Enstitüsü**, Cilt 1 Sayı 1 : 47-50, Ocak 2008.

Hobiler

Bilgisayar teknolojileri, Yüzmek, Gezmek.