T.C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ İŞLEME TEKNİKLERİNİ KULLANARAK OPTİK İŞARET TANIMA SİSTEMİ

Asmaeil Ammarah Abdullah BALQ

Danışman	**Dr. Öğr. Üyesi Yasemin GÜLTEPE**
Jüri Üyesi	**Dr. Öğr. Üyesi Abdulkadir KARACI**
Jüri Üyesi	**Dr. Öğr. Üyesi Halil İbrahim AKYÜZ**
Jüri Üyesi	**Dr. Öğr. Üyesi Javad RAHEBI**
Jüri Üyesi	**Dr. Öğr. Üyesi Mohammad M. GOMROKİ**

DOKTORA TEZİ
GENETİK VE BİYOMÜHENDİSLİK ANA BİLİM DALI

KASTAMONU – 2019

TEZ ONAYI

Asmaeil Ammarah Abdullah BALQ tarafından hazırlanan **"Görüntü İşleme Tekniklerini Kullanarak Optik İşaret Tanıma Sistemi"** adlı tez çalışması aşağıdaki jüri üyeleri önünde savunulmuş ve **oy birliği** ile Kastamonu Üniversitesi Fen Bilimleri Enstitüsü **Genetik ve Biyomühendislik Ana Bilim Dalı**'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Danışman	Dr. Öğr. Üyesi Yasemin GÜLTEPE Kastamonu Üniversitesi	.....
Jüri Üyesi	Dr. Öğr. Üyesi Abdulkadir KARACI Kastamonu Üniversitesi	.....
Jüri Üyesi	Dr. Öğr. Üyesi Halil İbrahim AKYÜZ Kastamonu Üniversitesi	.....
Jüri Üyesi	Dr. Öğr. Üyesi Javad RAHEBI Türk Hava Kurumu Üniversitesi	.....
Jüri Üyesi	Dr. Öğr. Üyesi Mohammad M. GOMROKI Türk Hava Kurumu Üniversitesi	.....

19/06/2019

Enstitü Müdürü	Prof. Dr. Hasbi YAPRAK	.....

TAAHHÜTNAME

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildirir ve taahhüt ederim.

Asmaeil Ammarah Abdullah BALQ

ÖZET

Doktora Tezi

GÖRÜNTÜ İŞLEME TEKNİKLERİNİ KULLANARAK OPTİK İŞARET TANIMA SİSTEMİ

Asmaeil Ammarah Abdullah BALQ
Kastamonu Üniversitesi
Fen Bilimleri Enstitüsü
Genetik ve Biyomühendislik Ana Bilim Dalı

Danışman: Dr. Öğr. Üyesi Yasemin GÜLTEPE

Bu tezde görüntü işleme teknolojisi ile Optik İşaret Tanıma (OİT) sisteminin geliştirilmesi amaçlamıştır. Hedeflenen amaç doğrultusunda çoktan seçmeli test sınavları cevap kâğıtlarının görüntü işlemeye dayalı okunması için sistem tasarlanmış ve sistem uygulaması ile elde edilen sonuçlar değerlendirilmiştir. Sistem, Microsoft Visual Studio 2013 ile Visual Basic (VB) programlama dili kullanılarak geliştirilmiştir. Sınav kâğıtlarının okunması ve değerlendirmesi işlemi eğiticilerin zamanının önemli bir kısmını alan önemli bir aktivitedir. Bu işlemin doğru ve hatasız bir şekilde gerçekleştirilmesi eğitimin değerlendirilmesi açısından son derece önem arz etmektedir. Gerçekleştirilen sistemde işaretlenen bir veya birden fazla seçeneği tespit etmek için farklı yöntemler kullanılmıştır. Bu yöntemler, optik cevap anahtarı şablonu ve anahtar noktanın (key point) hesaplanmasına dayanmaktadır. OİT, cevap kâğıtlarını ve cevap anahtarı şablonunu tarayıcı yardımıyla bilgisayara aktardıktan sonra oluşan görüntüler, sistem için girdi olarak kullanılmıştır. Bu sistem dört aşamalı olarak çalışmaktadır. İlk aşamada koordinatlar seçilerek şablon oluşturulur. Tüm koordinatlar kaydedilir. İkinci aşama olarak, anahtar nokta algılama algoritması uygulanır. Üçüncü aşama olarak ise optik notun cevap sayfaları otomatik olarak üç ilgi alanına (öğrenci kimliği, öğrenci adı ve çoktan seçmeli sorular) ayrılır ve her bir kabarcık sütun veya satırdaki siyah pikseli projeksiyon profili ve eşik tekniği kullanarak hesaplar. Son aşama olan dördüncü aşamada ise sınav kâğıdı önceden kaydedilmiş dosyadaki cevap anahtarı ile karşılaştırılarak ve otomatik olarak doğru cevapların sayısını hesaplayarak her öğrenci için bir istatistik sonuç elde edilir. Sistemin her bir sınav kâğıdı başına düşen işlem süresi 1 saniyeden azdır. Sistem performansı; işlem sürelerinin başarım karşılaştırılması ve doğruluk oranı üzerine odaklanarak literatürdeki üç farklı çalışma ile karşılaştırılmıştır. Karşılaştırma sonucu işlem süresi, her bir sınav kâğıdı için 1 saniyeden uzun sürmüştür. İşlem sonuçları çalışmaya bağlı olarak %97.6-%100 arasında doğruluk oranı ile gerçekleştirilmiştir. Bu karşılaştırma sonucunda daha doğru olduğu tespit edildi.

Anahtar Kelimeler: Görüntü işleme, optik işaret tanıma, eşik, anahtar nokta algılama.

2019, 82 Sayfa
Bilim Kodu: 101

ABSTRACT

PhD. Thesis

OPTICAL SIGNAL RECOGNITION SYSTEM USING IMAGE PROCESSING TECHNIQUES

Asmaeil Ammarah Abdullah BALQ
Kastamonu University
Institute of Science and Technology
Department of Genetic and Bioengineering

Supervisor: Assist. Prof. Dr. Yasemin GÜLTEPE

In this thesis, it is aimed to develop Optical SignaL Recognition (OIT) system with image processing technology. For the purpose of the aim, the system was designed for the image processing based reading of multiple-choice test exams and the results obtained with the system application were evaluated. The system was developed using Microsoft Visual Studio 2013 with Visual Basic (VB) programming language. Reading and evaluation of exam papers is an important activity which takes a significant part of the time of the trainers. Accurate and accurate operation of this process is of utmost importance for the evaluation of education. In addition, this system will be able to process quickly for processing hundreds or thousands of optical answer sheets. One or more options are identified in the system. The method used is based on the calculation of template and the key point. The images that were created after the OMR answer sheets and the answer key template were transferred to the computer with the help of the scanner were used as input. This system works in four stages. In the first step, the template is created by selecting the coordinates. All coordinates are saved. As the second stage, the key point detection algorithm is applied. As a third step, the optical answer pages are automatically divided into three areas of interest (student ID, student name and multiple choice questions) and calculate each bubble column or black pixel in row using projection profile and threshold technique. In the fourth stage, a statistical result is obtained for each student by comparing the exam paper with the answer key in the previously recorded file and automatically calculating the number of correct answers. The processing time for each paper of the system is less than 1 second. The performance of the developed system is compared with three different studies. These comparisons focus on the performance comparison and accuracy rate of processing times. The result of the comparison procedure took more than 1 second for each examination paper. In addition, the results of the study were performed with an accuracy rate of 97.6% - 100% depending on the study.

Key Words: Image processing, optical mark recognition, threshold, key-point detection

2019, 82 Pages
Science Code: 101

TEŞEKKÜR

Tez çalışması süresince sonsuz ilgi ve sabır ile değerli katkılarını hiçbir zaman esirgemeyen danışmanım Dr. Öğr. Üyesi Yasemin GÜLTEPE'e çok teşekkür ederim.

Tez jürimde olmayı kabul ederek zaman ayırdıkları ve bu çalışmaya yaptıkları önemli katkıları ve destekleri için değerli jüri üyeleri Dr. Öğr. Üyesi Abdülkadir KARACI'a, Dr. Öğr. Üyesi Halil İbrahim AKYÜZ'e, Dr. Öğr. Üyesi Javad RAHEBI'a ve Dr. Öğr. Üyesi Mohammad M. GOMROKI'a teşekkür ediyorum.

Hayatım boyunca bana olan sevgilerini, inançlarını her an hissettiren, beni sabırla dinleyen, destek olan ve bu tez süreci boyunca da hep yanımda olan anneme ve babama minnettarlığımı sunuyorum.

Bu süreçte bana hep yanımda olduğunu hissettiren, çalışmalarımı sürdürebilmem için hayatımı kolaylaştırmaya uğraş veren ve bana kendimi her zaman şanslı hissettiren değerli eşime ve varlıklarıyla bana mutluluk ve yaşama sevinci veren sevgili çoçuklarıma teşekkür ederim.

Kastamonu Üniversitesinde aldığım eğitim için yaptıkları destekten dolayı Libya Hükümetine ve Türkiye'deki Libya Büyükelçiliğine teşekkür ederim.

Asmaeil Ammarah Abdullah BALQ
Kastamonu, Haziran, 2019

İÇİNDEKİLER

	Sayfa
TEZ ONAYI	ii
TAAHHÜTNAME	iii
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
TABLOLAR DİZİNİ	xii
1. GİRİŞ	1
2. ALT YAPI VE İLGİLİ ÇALIŞMALAR	5
2.1. Dijital Görüntü İşleme	5
2.2. Dijital Görüntü Temsili	6
2.3. Temel Görüntü Ölçümleri	7
2.3.1. Çözünürlük	7
2.3.2. Piksel Bit Derinliği	7
2.3.3. KYM Modelini Kullanarak Renk Temsili	8
2.4. Dijital Görüntü Kaynakları	9
2.5. Dijital Görüntü Türleri	10
2.5.1. İkili Görüntüler	10
2.5.2. Gri Tonlu Görüntüler	10
2.5.3. Renkli Görüntüler	11
2.5.4. Çok Bantlı Görüntüler	11
2.6. Dosya Biçimleri	11
2.7. Renkli Görüntüyü Gri Tonlu Görüntüye Çevirme	12
2.8. Görüntü Bölümleme Tabanlı Eşikleme Yöntemi	14
2.8.1. Yerel Eşikleme	13
2.8.2. Küresel Eşikleme	14

2.9. Optik İşaret Tanımaya Yönelik Çalışmalar 15
3. YÖNTEM VE SİSTEM TASARIMI 21
3.1. Uygulamanın Yapısı 22
3.2. Yöntem Tasarımı 22
3.2.1. Birinci Adım 24
3.2.2. İkinci Adım 24
3.2.3. Üçüncü adım 29
3.2.4. Dördüncü adım 32
3.2.5. Beşinci adım 33
3.2.6. Altıncı adım 34
3.2.7. Yedinci adım 41
3.2.8. Sekizinci adım 42
3.3. Sistem Arayüz Tasarımı 42
3.3.1. Şablon Tasarımı 43
3.3.2. Yeni Şablon Tasarımı 43
3.3.3. Doğru Cevap Tasarımı 45
3.3.4. Tüm Görüntü İşlemelerin Kontrol Arayüzü 45
3.3.5. Sonuç Arayüzü Tasarımı 46
4. SONUÇ VE TARTIŞMA… 48
4.1. Deney/Test Çalışması ve Sonuçları 48
4.1.1. Deneysel Çalışma-1 48
4.1.2. Deneysel Çalışma-2 49
4.1.3. Deneysel Çalışma-3 50
4.1.4. Deneysel Çalışma-4 52
4.1.5. Deneysel Çalışma-5 53
4.2. Diğer Çalışmalarla Karşılaştırma 56
5. SONUÇLAR VE GELECEK ÇALIŞMALAR 59
KAYNAKLAR 61
EKLER…….…….. 64
EK1 Çalışmada kullanılan diğer tip cevap kağıtları …….……..... 65
EK2 OİT kılavuzu ……..……..... 67
EK3 Visual basic kodu …….……..... 68
ÖZGEÇMİŞ……. 82

SİMGELER VE KISALTMALAR DİZİNİ

DPI	Dots Per Inch
ICR	Akıllı Karakter Tanıma
IDE	Tümleşik Geliştirme Ortamı
KYM	Kırmızı, Yeşil, Mavi
LPI	Line Per Inch
MCQ	Çoktan Seçmeli Soru
OCR	Optik Karakter Tanıma
OİT	Optik İşaret Tanıma
PPI	Pixel Per Inch
RBG	Kırmızı Mavi Yeşil
SDK	Yazılım Geliştirme Kiti

ŞEKİLLER DİZİNİ

		Sayfa
Şekil 2.1.	Koordinat Düzenleri	6
Şekil 2.2.	Çıkarıcı Renk Karışımı	9
Şekil 2.3.	Elektromanyetik Spektrum	9
Şekil 2.4.	Gri Tonlamalı Bir Palet	10
Şekil 2.5.	Farklı Büyüklükteki Siyah Alanlar İçin Çıkış Eşiği Ölçümü	19
Şekil 3.1.	Uygulama Şeması	21
Şekil 3.2.	Yöntemin Temel Adımlarının Akış Şeması	23
Şekil 3.3.	Öğrenci Numarası Bölgesi	24
Şekil 3.4.	Öğrenci İsim Bölgesi	24
Şekil 3.5.	Çoktan Seçmeli Soru Bölgesi	25
Şekil 3.6.	Barkod Bölgesi	25
Şekil 3.7.	Şablondaki Önemli Alanlar	28
Şekil 3.8.	Bir Dosyada Saklanan Şablonun Koordinatları	29
Şekil 3.9.	Başlangıç Noktası Algılama Algoritması	31
Şekil 3.10.	Bitiş Noktası Algılama Algoritması	32
Şekil 3.11.	Dönüş Açısının (θ) hesaplanması	33
Şekil 3.12.	OİT Kağıdının Bölünmesi	33
Şekil 3.13.	Grup gösterimi akış şeması	35
Şekil 3.14.	Öğrenci numarası	37
Şekil 3.15.	Karakter Alanı	39
Şekil 3.16.	Cevap grubunu temsil etmektedir	39
Şekil 3.17.	Merkez Kabarcığın yerinin gösterimi	40
Şekil 3.18.	Cevap alanını kırptıktan sonra kayıtlı dosyanın sonucu	41
Şekil 3.19.	Uygulamanın Kullanıcı Arayüzü	43
Şekil 3.20.	Şablon Menüsü Görünümü	43
Şekil 3.21.	Şablonun Koordinatlarını Seçmek İçin Kullanıcı Arayüzü	44
Şekil 3.22.	ToolStrip1.Visible Bar ve ToolStrip2.Visible Bar Gösterimi	44
Şekil 3.23.	Doğru Cevap Menüsü Görünümü	45
Şekil 3.24.	Cevap tasarımı ara yüzü	45
Şekil 3.25.	Yeni ve Eski Doğru Cevap Açma Penceresinin Açıklanması	46
Şekil 3.26.	İşlem Kontrol Arayüzü	46
Şekil 3.27.	Ekran Çıktısı	47
Şekil 4.1.	(A) 100 Cevap Ve "16605" Öğrenci Kimliğini İçeren Cevap Anahtarı ve 53 Doğru Cevaplı (B) Örnek Cevap Sayfasını İçeren (A) Cevap Anahtarı ile Tip I Test Edilmesi	48
Şekil 4.2.	100 Test İçin Tip I Örnek Sonuçları	49
Şekil 4.3.	72 Cevap İçeren (A) Cevap Anahtarı Ve (B) 57 Doğru Cevap İçeren Öğrenci Testinden Oluşan Tip II'nin Test Edilmesi	49

Şekil 4.4. 86 Test İçin Tip II Örnek Sonuçları 49
Şekil 4.5. Excel Formunda Nihai Sonuçlar 50
Şekil 4.6. (A) 40 Cevap Anahtarı Ve (B) 22 Doğru Cevap İçeren, Öğrenci Adı "BUSE" Ve Öğrenci Kimliği "166211018" Olan Öğrenci Kağıdı İle Tip III Test Edilmesi 51
Şekil 4.7. 86 Test İçin Tip III Örnek Sonuçları..... 51
Şekil 4.8. Excel Sayfasında Nihai Sonuçlar..... 52
Şekil 4.9. 300 Soru İçeren (A) Cevap Anahtarı Ve (B) 111 Doğru Cevap İçeren Öğrenci Testiyle Tip V'nin Test Edilmesi..... 52
Şekil 4.10. 20 Test İçin Tip V Örnek Sonuçları..... 53
Şekil 4.11. 11 adet hatalı dosya ile birlikte 82 dosyayı test işleminde gereken işlem süresi (CPU süresi)..... 53
Şekil 4.12. Dosyalanmış arama için artırım miktarındaki değişim gösterimi..... 54
Şekil 4.13. Dhatalı dosya olmaksızın 82 dosyayı test işleminde gereken işlem (CPU süresi)..... 54
Şekil 4.14. İşlemci zamanı sonuçlarin statistiksel gösterimi..... 55
Şekil 4.15. Test sayısına göre işlem zamanı..... 56
Şekil 4.16. Sonuçların İstatistiksel Gösterimi..... 58

TABLOLAR DİZİNİ

		Sayfa
Tablo 2.1.	Çeşitli Çaplardaki İşaretleme Sonuçları…….	19
Tablo 4.1.	Literatürdeki Çalışmalar ile Tez Çalışması Sonuçlarının Karşılaştırılması…….	55
Tablo 4.2.	Çeşitli Çaplardaki İşaretleme Sonuçları…….	57

1. GİRİŞ

Eğitim sistemindeki en önemli trendlerden biri olan Optik İşaret Tanıma (OİT)'dır. OİT, 1960'lı yıllardan beri eğitim alanında yaygın olarak kullanılmaktadır. Bu sistemler çoktan seçmeli şıklar içeren optik formlar aracılığı ile öğrenci cevaplarının değerlendirilmesinde öğretmenlere kolaylık sağlamaktadır (Haag vd., 2006).

Sumitra ve Gaikwad (2015) çalışmasına göre, OİT teknolojisinin temelinde işaretli dokümanı taramak, önceden tanımlanmış bölgeleri yakalamak ve işaretlerin bulunduğu veya bulunmadığı yerleri kaydetmek için bir tarayıcı kullanılır. Yazdırılan kağıttaki işaretler bir kalem veya kurşun kalem kullanılarak işaretlenmiştir. Bu teknoloji, 1950'lerde Amerika Birleşik Devletleri'nde kullanılmaya başlanmıştır. Genel olarak öğrencilerin çoktan seçmeli testlerin değerlendirilmesinde kullanılmaktadır ve o zamandan günümüze kullanımı uzanmaktadır. Çoktan Seçmeli Soru (Multiple Choice Question, MCQ), çoğunlukla öğrencilerin okullardaki akademik performansını değerlendirmek için kullanılmaktadır. MCQ, eğitim sisteminin önemli bir parçasıdır. Önemli testler ayrıca öğrencinin akademik performansını değerlendirmek için çoktan seçmeli soruları kullanırlar. Bugün okullarda, kolejlerde ve sınıflarda, OİT teknolojisi kullanılmaktadır. Sınavlar, OİT cevap tablosu kontrol sistemi kullanılarak yapılmaktadır. Çünkü bu teknolojiyi kullanarak sınavların yapılması çok daha kolay, güçlü ve ucuz hale gelir (Sumitra ve Gaikwad, 2015). Ayrıca özel bir okuyucu alma ihtiyacı ortadan kalkar.

Lin vd. (2013) çalışmasında Barkod teknolojisi, Otomatik Tanımlama ve Veri Yakalama (AIDC)'nin en önemli parçalarından biri olarak kabul edilmektedir. Barkod, insan müdahalesi olmadan verilerin depolanmasını ve geri alınmasını otomatikleştiren sembolik olarak tanımlanabilir. Bu teknoloji, veri toplama ve envanter kontrolü amacıyla otomatik tanımlama ile birlikte birçok endüstriyel ürün için yaygın olarak kullanılmaktadır.

Optik Karakter Tanıma, iki sınıfa ayrılmaktadır: Çevrimdışı tanıma ve Çevrimiçi tanıma. Çevrimdışı tanımada kaynak ya bir görüntü ya da belgenin taranmış bir formudur. Çevrimiçi tanımada ise ardışık noktalar zamanın bir fonksiyonu olarak

gösterilmekte ve vuruş sırası da dahil edilmektedir (Arica ve Yarman-Vural, 2001; Plamondon ve Srihari, 2000).

Suratanee'nın (2014) çalışmasında Akıllı Karakter Tanıma sistemi el yazısı metnini okumak için kullanılmıştır. İşlem oldukça karmaşıktır ve doğru sonuçlar için daha fazla el yazısı örneği gerekir.

Tez çalışmasının önemi beş başlık altında toplanabilir. Tasarlanan yazılım sistemi kısa sürede büyük miktarda veriyi işlemek için kullanılabilir, ayrıca çok sayıda cevapların yakalanmasına ve puanlanmasına yardımcı olabilir. Bununla birlikte bu sistem elle düzeltme yerine elektronik düzeltme için çok hızlıdır. Dahası, sistemi kolay kullanıcı ara yüzüne sahip olarak tasarlanmıştır. Sonunda, sistemin gereksinimlerine kolayca ulaşılabilir (verileri toplamak ve işlemek için gereken bir tarayıcı ve bilgisayar).

Tez çalışması, çoktan seçmeli cevap kâğıdından sınav puanını saymak için kullanılabilecek optik işaret okuyucuyu geliştirmeyi hedeflemektedir. Bu hedef ulaşmak için sistem dolu baloncukların bulunması için farklı tipte OİT tabakalarını tanıyacak şekilde tasarlamaktır. Bir sonraki amaç ise görsel temel programlama 2013 kodunu kullanarak optik işaret tanıma ile görüntü işlemektir. Sonuncusu bilimsel literatüre yeni bir çalışma sağlamaktır.

Bu çalışmadaki ana sorun, sonuç sürecini test etme ve analiz etmenin karmaşıklığıdır. Bu süreç içerisinde görevleri tamamlamak için zaman harcanır. Ortalama olarak, öğretmenlerin %95'i öğrencilerin testlerini manuel olarak değerlendirirken, çoğu öğretmen hala sınıfları test etmek için dış hizmetleri kullanmaktadır. Bu hizmetler gerekenden daha fazla zaman ve çaba gerektirebilir. Artık çoğu okul ve üniversite OİT yazılımını kullanarak karanlık izleri tespit edip okumakta daha doğru sonuçlar vermektedir (Reg, 2013).

Öğrenciler cevaplarını veya diğer kişisel bilgilerini, önceden basılmış bir sayfada işaretlenmiş daireleri karartmak suretiyle işaretlerler. Daha sonra, sayfa otomatik olarak bir tarama makinesi tarafından derecelendirilir. Bazen bazı öğrenciler

kabarcığın üzerine işaretlemeleri çok hafif veya kabarcığın yarısından daha az doldurur veya bir soruda kabarcığından fazlasını doldurur.

Bu durumlara yönelik basit çözümler önerilmektedir. Bu çözüm önerileri öğretmenlere yardımcı olabilecek hızlı ve ekonomik bir yol sunar.

Önerilen sistemde, mükemmel bir sonuç elde etmek için OİT levhalarını işleyen bir uygulama geliştirdik. Bu sistem, şablon ve anahtar nokta tespiti (başlangıç noktası ve bitiş noktası) oluşturulmasına bağlıdır. Yöntem genel olarak beş bölüme ayrılmıştır. İlk olarak, tüm kağıtlar tarayıcı kullanılarak bilgisayara girilir ve yazılım sistemine girdi olarak verilir. İkinci olarak, koordinatlar (başlangıç noktası ve bitiş noktası, başlangıç noktası ve bitiş noktasının uzunluğu ve genişliği, işaretlemelerim ve grup bölgelerinin boyutu) seçilerek şablon oluşturulur. Tüm koordinatlar dosyada saklanır. Üçüncü olarak, anahtar nokta tespit algoritmasını uygulanır. Daha sonra, bölgeler otomatik olarak üç bilgi alanına (öğrenci kimliği, öğrenci adı ve çoktan seçmeli sorular) ayrılır. Her bir işaretlemenin kolon ve satırdaki her bir noktası, projeksiyon profili ve eşik teknikleri kullanılarak hesaplanır. Cevap sayfasındaki projeksiyon profili ve eşikler, tüm bilgileri açıklamak için kullanılır. Sonuçlar daha sonra dosyaya kaydedilir. Son olarak, sınav kâğıdı önceden kaydedilmiş cevap anahtarı ile karşılaştırılır ve otomatik olarak doğru cevapların sayısı hesaplanır.

Bu tez genel olarak beş bölümden oluşmaktadır ve her bölümün içeriği aşağıda özetlenmektedir.

İkinci bölüm, araştırmayla ilgili bilgilere odaklanan literatür taraması hakkında bilgi vermektedir. Dergiler veya diğer referanslar aracılığıyla bulunan her olgu ve bilgiler karşılaştırılarak önerilen yöntem için daha iyi yöntem seçilmiştir.

Üçüncü bölümde, yöntem ve uygulama hakkında bilgi verecektir. Sistem tamamlanana kadar her aşama için yöntem detaylı olarak açıklanmıştır.

Dördüncü bölüm, deney sonuçları ve diğer çalışmalarla karşılaştırmaları açıklanmaktadır. Sonuçlar tablolar ve şekillerle sunulmuştur.

Beşinci Bölüm, önerilen yöntemin tartışması ve sonuçları verilerek bunun geliştirilmesine yönelik gelecek çalışmalardan bahsedilmiştir.

2. ALT YAPI VE İLGİLİ ÇALIŞMALAR

Dijital görüntü işleme tekniklerinin çoğu dijital görüntü işleme olarak adlandırılır ve 1960'lı yıllarda Maryland'de Jet Propulsion Laboratuvarı, MIT, Bell Labs, Maryland Üniversitesi'nde uydu görüntülerine, fotoğraf standartları dönüştürme, tıbbi görüntüleme, görüntülü telefon, karakter tanıma ve fotoğraf geliştirmede uygulanmak üzere geliştirilmişti. Ancak bu dönemin bilgi işlem yöntemeleri ile işleme maliyeti oldukça yüksektir.

Günümüzde, görüntü işleme hızla büyüyen teknolojiler arasındadır. Çeşitli alanlarda ve uygulamalarda analiz için etkili bir araç haline gelmiştir**.** Örneğin, endüstrilerde ürünün kalitesini ve verimliliğini arttırmak için otomatik görsel denetim sisteminde, tıp alanında görüntülerdeki belirtileri tespit etmek için kullanılır. Akıllı taşıma sistemlerinde otomatik plaka tanıma ve trafik işareti tanıma işlemlerinde kullanılabilir. Uzaktan algılama alanında, dünyanın yüzeyinin fotoğraflarını uzaktan algılama uydularında veya bir uçakta monte edilmiş çok spektrumlu tarayıcılarda tanıma işlemi için kullanılır ve otomatik tanıma sisteminde metni optik karakter tanıma veya onay kutusu kullanarak görüntüden çıkarmak için kullanılır. Optik karakter tanıma veya onay kutusu ve optik işaret tanımayı kullanarak işaretleme değerleri kullanarak görüntüden metin ayıklarlar.

Bu bölümde tez çalışması için gerekli alt yapı bilgileri ve daha önce bu konuda gerçekleştirilmiş olan çalışmalar yer almaktadır. Alt yapı bilgileri ile ilgili olarak renkli görüntüyü gri tonlamalı görüntüye dönüştürmek için çeşitli algoritmalar ve farklı teknikler hakkında bilgiler yer almaktadır. Ayrıca farklı segmentasyon teknikleri ve bunların bilgisayar ortamında uygulamaları hakkında bilgiler yer almaktadır. Optik işaret tanıma ile ilgili diğer çalışmalar hakkında bilgi verilmiştir.

2.1. Dijital Görüntü İşleme

Dijital görüntü, bir görüntüleme cihazında görüntülendiğinde (örneğin, bir bilgisayar monitörü) kare resim öğelerine (piksel) dönüşen elektronik bir dosyadır.

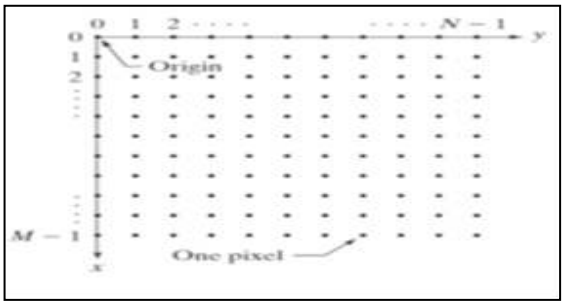
Görüntülenen görüntülerin her birinin kendi adresi, boyutu ve renk sunumu olan binlerce veya milyonlarca pikselden oluşan iki boyutlu bir matrisidir (Puglia, 2000).

2.2. Dijital Görüntü Temsili

Bir görüntü, x ve y uzaysal (düzlemsel) koordinatlar olan iki boyutlu bir fonksiyon f (x, y) olarak tanımlanabilir ve herhangi bir çift koordinattaki f büyüklüğü, o noktadaki görüntünün yoğunluğu olarak adlandırılır. Gri seviye terimi genellikle tek renkli görüntülerin yoğunluğunu ifade etmek için kullanılır. Renkli görüntüler, tek tek görüntülerin bir kombinasyonu ile oluşturulur. Örneğin, KYM (Kırmızı, Yeşil, Mavi) renk sisteminde bir renkli görüntü, kırmızı (R), yeşil (G) ve mavi (B) birincil (veya bileşen) görüntüler olarak adlandırılan üç tek renkli görüntüden oluşur. Bu nedenle, tek renkli görüntüler için geliştirilen tekniklerin çoğu, üç bileşenli görüntüleri tek tek işleyerek renkli görüntülere genişletilebilir.

2.2.1. Koordinat Düzenleri

Örnekleme, gerçek sayıların bir matrisidir. Bir görüntü f(x, y) 'nin örneklendiğini, sonuçta elde edilen görüntünün M satırları ve N sütunları olduğunu farz edelim. Görüntünün M * N boyutunda olduğunu söyleyebiliriz. Koordinatların değerleri ayrı miktarlardadır. Görüntü kökeni (x, y) = (0,0) olarak tanımlanır. Görüntünün ilk sırasındaki sonraki koordinat değerleri (x, y) = (0,1) şeklindedir. Şekil 2.1'de piksel koordinat düzenlerini göstermektedir. X'in 0 ile M-1 arasında değiştiğini ve y ile uzamsal koordinatlar olarak adlandırıldığı ve 0 ile N arasında olduğunu, sütunlara başvurmak için x öğesini ve satırlara başvurmak için y öğesinin kullanıldığını unutulmamalıdır.



Şekil 2.1. Koordinat düzenleri

2.2.2. Görüntülerin Matris Olarak Gösterimi

Bir sayısal görüntü, analog resmin örneklenmesi ve kuantalanması sonucunda elemanları reel sayılardan oluşan bir matris formunda ifade edilir. f(x,y) şeklindeki bir sayısal görüntü, M satır N sütundan oluşmuş M*N elemanlı bir matris gösterilmiştir.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots\dots\dots. & f(0,N-1) \\ f(1,0) & f(1,1) & \dots\dots\dots & f(1,N-1) \\ f(M-1,0) & f(M-1,1) & \dots\dots\dots & f(M-1,N-1) \end{bmatrix} \tag{2.1}$$

Denklem 2.1'de sayısallaştırılmış görüntüyü ifade eden matris formu gösterilmiştir (Gonzalez, Woods ve Eddins 2004).

2.3. Temel Görüntü Ölçümleri

Her statik dijital görüntünün üç temel ölçümü vardır. Bunlar şu şekilde sıralanabilir: Uzamsal çözünürlük, piksel bit derinliği ve renktir. Her ölçüm için seçilen özellikler, orijinal fotoğrafı veya sahneyi temsil etmek için yakalanan elektronik bilginin miktarını belirler. Genel olarak değerler bu ölçüler içinde ne kadar yüksekse, orijinalden daha fazla miktarda fotoğraf detayını temsil eden daha fazla veri elde edilir.

2.3.1. Çözünürlük

Uzamsal çözünürlük, bir görüntüdeki birkaç yönden tanımlanabilen piksel sayısıdır. En yaygın ölçümler şunlardır: DPI (Dots Per Inch, inç başına düşen nokta sayısı), PPI (Pixels Per Inch, inç başına düşen piksel sayısı) ve LPI (Lines per Inch, inç başına düşen satır).

2.3.2. Piksel Bit Derinliği

Bu ölçü bazen piksel derinliği bazen de renk derinliği olarak adlandırılır. Bu, her piksel için kaydedilen bilgi miktarıyla gerçekte gösterilebilen gölgelerin sayısını tanımlar. Bilgisayarlar ikili sayı sistemi ile çalışırlar; her veri biti 1 veya 0'dır.

Görüntüdeki her piksel, bir ikili basamak dizisiyle gösterilir ve basamak sayısı bit derinliği olarak bilinir. Raster formatındaki veriler, konumsal detayların gerçek durumunu bir kafes (ızgara) sistemi veya daha çok bir satranç tahtası şeklinde temsil eder. Bu nedenle, bir bit görüntü, tek bir piksele yalnızca 0 veya 1 (siyah veya beyaz) olmak üzere iki değerden birini atayabilir: 8 bitlik (2^8) gri tonlamalı bir görüntü, 256 renkten birini tek bir piksele atayabilir. 24 bit ($2^{(3 \times 8)}$) KYM görüntü (her biri kırmızı, yeşil ve mavi renk kanalları için 8 bit), tek bir piksele 16,8 milyon renkten birini atayabilir. Bit derinliği, bu ikili sayıların sayısı için 1 ve 0 olası kombinasyonlarının sayısını ve dolayısıyla her piksel tarafından temsil edilebilecek gri renk tonlarının sayısını belirler. Denklem 2.2'de gösterildiği gibi gri tonlu görüntüleri 2, 4, 6, 12, 16 ve 32 bit derinliği, 8 bpp (yani piksel başına 1 bayt) değerinde hesaplayabiliriz (Puglia, 2000).

$$\text{Gölge sayısı} = 2^{x}, \quad x = \text{bit derinliği} \tag{2.2}$$

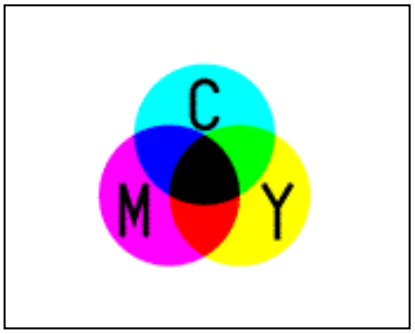
2.3.3. KYM Modelini Kullanarak Renk Temsili

KYM renk modeli, her bir pikselin üç sayısal değer olarak temsil edildiği bir ek renk modelidir. İlk değer kırmızı, ikincisi yeşil ve üçüncüsü mavidir. Bu değerler ekranda sunulan rengi oluşturmak için kullanılır. Diğer her değer karışımı farklı renk anlamına gelir. KYM renk küpünde karıştırmanın en temel kuralı şöyledir (Koirala, 2007).

- R+G+B = Beyaz.
- R+G = Sarı.
- R+ B = Eflatun.
- G+B =Açık Mavi.

CMYK (Cyan, Magenta, Yellow ve Black) veya Açık mavi, Eflatun, Sarı ve Siyah, baskı için standart mürekkep renkleri olan çıkarıcı renklerdir. Bu, bir görüntüyü her yazdırdığımızda, baskıyı üretmek için CMYK mürekkepleri kullanıyoruz demektir. Birçok profesyonel yazıcı veya yayıncı, basımdan önce basılacak görüntülerin CMYK'ye dönüştürülmesini gerektirir. Bunun nedeni, KYM renk spektrumunun

(göstergeler) CMYK spektrumu (mürekkep) ve KYM'den CMYK'ye dönüşüm sırasında Şekil 2.2'te gösterildiği gibi çok daha geniş olmasıdır.

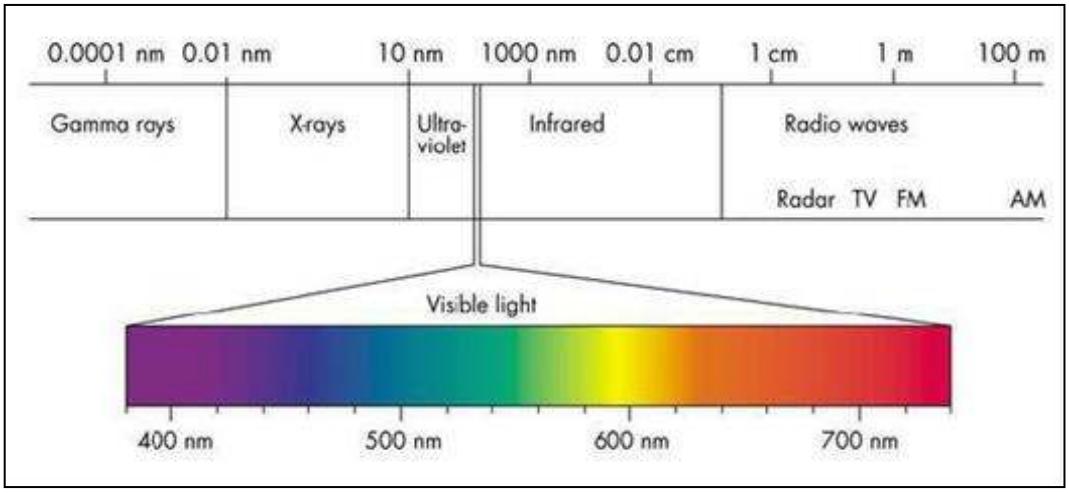


Şekil 2.2. Çıkarıcı renk karışımı (renklerin maddesel karışımları)

2.4. Dijital Görüntü Kaynakları

Görüntülerin ana kaynağı gama ışınlarından (en yüksek enerji) radyo dalgalarına (en düşük enerji) kadar uzanır. İnsan gözü, Şekil 2.3'de gösterildiği gibi, elektromanyetik spektrumda yaklaşık 400 nm (nanometre) (mor) ve 700 nm (kırmızı) arasında görünür renkleri görebilir.

Bir elektromanyetik spektrumun en uzun dalga boyundan en kısa dalga boyuna kadar sıralanması, radyo dalgaları, mikrodalga, kızılötesi, görünür ışık, morötesi, x-ışınları ve gamma ışınları biçiminde olur. Frekansı en küçük elektromanyetik dalgalar radyo dalgaları olarak adlandırılır.



Şekil 2.3. Elektromanyetik spektrum

2.5. Dijital Görüntü Türleri

Dijital görüntüler genel olarak dört farklı biçimde sınıflandırılabilir: İkili, gri tonlamalı, renkli ve çok bantlı görüntülerdir. Dijital görüntü türleri hakkında kısa bilgiler aşağıdaki alt bölümlerde ayrıntılı olarak anlatılmıştır.

2.5.1. İkili Görüntüler

İkili görüntüler, her bir pikseli temsil etmek için yalnızca bir bit kullanır. Bir bit, yalnızca iki durumda açık veya kapalı olarak mevcut olabileceğinden, ikili görüntüdeki her piksel, genellikle siyah veya beyaz olmak üzere iki renkten biri olmalıdır.

2.5.2. Gri Tonlu Görüntüler

Bu resim türü, piksel başına 1 bayt kullanır ve her piksel yoğunluğu 0-255 arasında değişir. Şekil 2.4'de gösterilen gri tonlama, görünür renk olmadan gri tonlarının bir dizisidir. En karanlık olası gölge siyahtır. Bu iletilen veya yansıtılan ışığın tamamen yokluğu anlamına gelmektedir (bilgisayar ekranında görüntü, vb.). KYM bileşenlerin parlaklık seviyeleri, her biri ondalık 0 ile 255 arasında bir sayı veya ikilik sayı sistemine göre 00000000 ile 11111111 arasında temsil edilir.

KYM'deki her piksel için;

Gri tonlamalı görüntü, R=G=B ile temsil edilir.

Siyah, R=G=B=0 veya R=G=B=00000000 ile temsil edilir.

Beyaz, R=G=B=255 veya R=G=B= 11111111 ile temsil edilir.

Gri seviyenin ikili temsili 8 bit olduğu için, bu görüntüleme yöntemine 8 bit gri tonlama denir. Şekil 2.4'de, gri tonlamalı görüntülerin bir örneği gösterilmektedir.

Şekil 2.4. Gri tonlamalı bir palet

2.5.3. Renkli Görüntüler

Renkli bir görüntü, her biri belirli bir konumdaki görüntünün kırmızı, yeşil ve mavi seviyelerine karşılık gelen üç sayılı piksellerden oluşur. KYM, ışığın karıştırılması için ana renklerdir. Katkı maddesi olarak adlandırılan bu birincil renkler, renklerin karıştırılmasında kullanılan açık mavi, eflatun ve sarı renklerinden çıkarmalıdır. Doğru miktarlarda kırmızı, yeşil ve mavi ışığı karıştırarak herhangi bir renk oluşturulabilir. Her temel renk için 256 seviye varsayarsak, her bir renk pikseli belleğin üç baytında (24 bit) saklanabilir. Bu yaklaşık olarak 16.7 milyon farklı olası renge karşılık gelir (Sachs, 2003).

2.5.4. Çok Bantlı Görüntüler

Çok bantlı görüntüler genellikle normal insan algısal aralığı dışındaki bilgileri içerir. Bu kızılötesi, ultraviyole, X-ışını, radar verisini içerebilir. Bunlar, genel anlamda görüntü değildir. Şekil 2.3'te gösterildiği gibi temsil edilen bilgi, insanlar tarafından doğrudan görülememektedir.

2.6. Dosya Biçimleri

Görüntüleri saklamak için dört temel format vardır (Caroline, 2010).

- İmlenmiş Görüntü Dosyası Biçimi (Tagged Image File Format, TIFF): TIFF görüntüleri çok büyük dosya boyutları oluşturur. TIFF görüntüleri sıkıştırılmamış ve bu nedenle çok sayıda ayrıntılı görüntü verisi (bu yüzden dosyalar çok büyüktür) içerir. TIFF'ler aynı zamanda renk açısından da son derece esnektir (baskı için gri tonlu, CMYK veya web için KYM olabilirler).

- Birleşmiş Fotoğraf Uzmanları Grubu (Joint Photographic Experts Group, JPEG): JPEG, bu tip görüntü formatı için bu standardı yaratan Birleşik Fotoğraf Uzmanları grubu anlamına gelmektedir. JPEG dosyaları, birçok bilgiyi küçük boyutlu bir dosyada saklamak için sıkıştırılmış görüntülerdir. Dijital fotoğraf makinelerinin çoğu fotoğrafları JPEG

biçiminde depolar, çünkü diğer formatlarda kullanabileceğinizden daha fazla fotoğraf çekebilirsiniz. JPEG dosyaları genellikle web'deki fotoğraflarda kullanılır, çünkü bunlar kolayca web sayfasına yüklenen küçük bir dosya halindedir ve iyi görünürler.

- Grafik Değişim Biçimi (Graphic Interchange Format, GIF): Grafik Değişim Biçimi anlamına gelir. Bu türdeki görüntüleri sıkıştırmaktadır. Ancak JPEG'den farklı olarak sıkıştırma kayıpsız olarak gerçekleşmektedir. Ayrıca GIF'ler web için daha uygundur Ancak baskı için son derece sınırlı bir renk aralığına sahiptirler.

- Taşınabilir Ağ Grafikleri (Portable Network Graphics, PNG): Taşınabilir Ağ Grafikleri anlamına gelir. Sadece web görüntüleri için kullanılır, asla baskı görüntüleri veya fotoğraflar için kullanılmaz. PNG, büyük bir dosya büyüklüğüne sahip olması nedeniyle JPEG kadar iyi değildir.

2.7. Renkli Görüntüyü Gri Tonlu Görüntüye Çevirme

Renkli görüntünün gri tonlamalı bir görüntüye dönüştürülmesi, renkli görüntü hakkında daha fazla bilgi gerektirirmektedir. KYM renk değerleri, hafiflik, renk parlaklığı ve renk özellikleriyle gösterilen üç boyutlu XYZ olarak temsil edilir. Renkli görüntünün kalitesi, dijital aygıtın destekleyebileceği bit sayısına göre belirlenen renge bağlıdır. Basit renkli görüntü 8 bit ile temsil edilir, yüksek renkli görüntü 16 bit kullanılarak temsil edilir, gerçek renkli görüntü 24 bit ile temsil edilir ve derin renkli görüntü ise 32 bit ile temsil edilir. Bit sayısı dijital cihaz tarafından desteklenen maksimum farklı renk sayısına karar verir. Her Kırmızı, Yeşil ve Mavi 8 bitse, KYM kombinasyonu 24 bit kaplar ve 16.777.216 sayıda farklı renkleri destekler. 24 bit, renkli görüntüdeki bir pikselin rengini temsil eder. Gri tonlamalı görüntü 8 bit değeri kullanılarak parlaklık ile temsil edilir. Gri tonlamalı bir görüntünün piksel değerinin parlaklığı 0 ile 255 arasında değişir. Renkli bir görüntünün gri tonlu bir görüntüye dönüştürülmesi KYM değerlerini (24 bit) gri tonlama değerine (8 bit) dönüştürür. Renkli Görüntüyü Gri Tonlamalı Görüntüye Dönüştürme, üç farklı teknikle sağlanabilir:

- Yoğunluk Yöntemi: Denklem 2.3 'te gösterildiği gibi KYM kanallarının ortalamasıdır (Jack, 2007).

$$Intensity = \frac{(R+G+B)}{3} \tag{2.3}$$

- Luster Yöntemi: Denklem 2.4'te gösterilen minimum ve maksimum KYM değerlerinin ortalamasıdır (Kanan ve Cottrell, 2012).

$$Luster = \frac{1}{2}\left(\max(R,G,B)\min(R,G,B)\right) \tag{2.4}$$

- Parlaklık Yöntemi: KYM kanallarının ağırlıklı bir bileşimini kullanarak insan parlaklığı algısına uyacak şekilde tasarlanmıştır Denklem 2.5'te gösterildiği gibi parlaklık, görüntü işleme yazılımı tarafından kullanılan standart algoritmadır (Pratt, 2007).

$$Luminance = 0.3R + 0.59G + 0.11B \tag{2.5}$$

2.8. Görüntü Bölümleme Tabanlı Eşikleme Yöntemi

Eşikleme, görüntü bölümlemenin en sade yöntemidir. Eşikleme, bazı eşiklerin altındaki tüm pikselleri sıfıra ve o eşiğe ilişkin tüm pikselleri bire dönüştürerek, gri tonlu ikili görüntüler oluşturur (Bryan, 1998). Otomatik eşikleme tekniği küresel ve yerel eşikleme olmak üzere iki temel gruba ayrılabilir.

2.8.1. Yerel Eşikleme

Yerel eşiklemede imge içersindeki her bir bölütü elde edebilmek için birden fazla eşik değeri seçilir. Bir eşik T (x, y), Denklem 2.6'de gösterildiği gibi bir değerdir:

$$\mathrm{b}(\mathrm{x},\mathrm{y}) = \begin{cases} 0 & if\ \ I(x,y) \leq T(x,y) \\ & f \\ 1 & Otherwise \end{cases} \tag{2.6}$$

(x, y) binarize edilmiş görüntüdür ve [0, 0] 'da I (x, y) lokal adaptif tekniğe Niblack ve Bernsen gibi yaklaşılabilir (Singh vd, 2011).

Niblack'ın algoritması, yerel ortalama ve yerel standart sapmanın hesaplanmasına dayanan yerel bir eşikleme yöntemidir. Eşik, aşağıdaki formül 2.7 ile belirlenir.

$$\mathrm{T(x,y) = m(x,y) + k * s(x,y)} \tag{2.7}$$

m(x, y) ve s(x, y), yerel penceredeki piksellerin yerel ortalama ve standart sapmasıdır ve k bir önyargıdır. k=0.5 olarak ayarlanabilir. Yerel ortalama m(x, y) ve standart sapma s(x, y), pikselin yerel değerindeki karşıtlığa göre eşiğin değerini ayarlar (Firdousi ve Parveen, 2014).

Bernsen tarafından önerilen bu teknik, yerel eşik değerini belirlemek için yerel kontrast değerini kullanan yerel bir binarizasyon tekniğidir. Her piksel için yerel eşik değeri (x, y) şu ilişki tarafından hesaplanır:

$$T(x,y) = \frac{I_{max}+I_{min}}{2} \tag{2.8}$$

Denklem 2.8'de, I_{max} ve I_{min}, (x, y) merkezli bir w × w penceresindeki maksimum ve minimum gri seviye değerlerini temsil etmektedir (Kaur, 2011).

2.8.2. Küresel Eşikleme

Ön plan ve arka plan nesneleri arasındaki yoğunluk dağılımı çok farklı olduğunda, küresel (global) eşikleme yöntemi kullanılır. Ön plan ve arka plan nesneleri arasındaki farklar çok belirgin olduğunda, her iki nesneyi birbirinden ayırmak için tek bir eşik değeri kullanılabilir. Bu tip eşik değerde, eşik T değeri sadece pikselin özelliğine ve görüntünün gri seviye değerine bağlıdır. En yaygın kullanılan bazı küresel eşikleme yöntemleri; Otsu yöntemi ve entropi tabanlı eşiklemedir (Kumbhar ve Holambe, 2015).

2.9. Optik İşaret Tanımaya Yönelik Çalışmalar

1950'li yılların ortasından başlayıp günümüze kadar donanım cihazı ve yazılım paketi olarak ticari OİT sistemleri geliştirilmeye devam etmektedir (Rakesh vd., 2013). OİT ile ilgili çalışmalar hız kesmemiş, çevre koşulları, ortam ve karmaşık zeminler, video görüntüleri üzerinde yapılan çalışmalar devam etmektedir.

Chinnasarn ve Rangsanseri (1999) çalışmasında, görüntü tabanlı OİT çalışmalarında mikrobilgisayar ve görüntü tarayıcı kullanılmıştır. Sistem çalışması iki modda ayırt edilebilir: Öğrenme modu ve tanıma modu. Öğrenme modunda dikey ve yatay projeksiyon profiline odaklanılmıştır. Projeksiyon profili, görüntünün satır ve sütunları için piksel sayılarının histogramıdır. Yatay projeksiyon profili, bir belge görüntüsündeki metin satırını yatay sayısını gruplanmak için uygulanabilir histogramdır. Dikey projeksiyon profili, çok sütunlu döküman görüntülerinde sütunların segmentasyonu için tasarlanmıştır. Dikey projeksiyon profili, her alandaki her bir daire sınırını tanımlamak için kullanılır. Sol ve sağ siyah piksellerin sayısı sayılır ve daha sonra her dairedeki siyah piksellerin sayısı korunur. Algoritma dikey ve yatay projeksiyon profili ile kontrol edilir. Çalışma modunda, tüm görüntüler taranır ve görüntüler ikili görüntülere dönüştürülür. Sadece piksel başına bit kullanılır, sonra her bir cevap kâğıdının üç bölüme ayrılır: Öğrenci alan kodu, konu alan kodu ve cevap alanıdır. Öğrenci ve konu alan kodu için her daire içindeki koyu piksellerin sayısı sayılır ve maksimum değer pozisyonu aranır. Her sorunun cevabı için, maksimum değer pozisyonu, listedeki gerçek cevap öğesi modeli ile karşılaştırılır, eğer her ikisi de aynı verilen konumdaysa, listenin sonuna kadar bir sonraki cevap öğesine geçilir.

Krishna vd. (2013) çalışmasında, "Programlama mantığı kullanarak düşük maliyetle yüksek hızlı optik işaret okuyucu donanım uygulaması" sunmuşlardır. Çoktan seçmeli soruları işaretlemek için düşük maliyetli ve yüksek hızlı bir OİT sistemi prototipi geliştirmişlerdir. Bu yaklaşımın yeniliği, yüksek işlem hızına ulaşmak için komple sistemin tek bir düşük maliyetli Alan Programlanabilir Kapı Dizisi (Field Programmable Gate Array, FPGA) içine uygulanmasıdır.

Hasan ve Kareem (2015) çalışmasında, Çoklu Bağlantı Mimarisini Değiştirme (Modify Multi-Connect Architecture, MMCA) ile OİT çalışması yapmışlardır. Bu çalışma, çoktan seçmeli testler için yeni bir teknik olan "Çoklu Bağlantı Yapısını Değiştiren Birleşik Bellek" kullanan bir OİT geliştirilmesine odaklanmıştır. Ortalama işlem hızı dakikada 60 sayfadan fazladır.

Al-marakeby (2013) çalışmasında, çok çekirdekli işlemci sisteminde çalışan optik işaret tanıma sistemi için düşük maliyetli ve hızlı bir çözüm önerisi sunmuştur. Buna göre cevap sayfası bir dijital kamera kullanılarak çekilir ve görüntü işlenir. Başlangıçta sayfanın sınırları bulunur, daha sonra işaretlemeler tespit edilir. Hızlı teknikler, rotasyon düzeltmesi olmayan işaretlemeleri tespit etmek için kullanılır.

Bayar (2016) çalışmasında, çoktan seçmeli sınavlarda kullanılan otomatik derecelendirme sistemlerine yeni bir bakış açısı getirmiştir. Otomatik bir derecelendirme sistemi geliştirmek için Hough dönüşüm tekniğinin uyarlanması, çalışmasının ana odak noktasıdır. Hough dönüşümü, bir veri kümesinde veya bir görüntüdeki olası bir satıra yerleştirilen noktaların sayısını saymayı mümkün kılan bir yöntemden oluşur. Bu yöntem, eğim kesişme bilgisine dayalı olarak oluşturulan bir formdaki çizgilerin temsil edilmesine odaklanır. Çalışmada 1000'den fazla sınav kâğıtları test edilmiştir. Öğrencilerin kâğıtlarından birini doldurmak için gereken ortalama süre 10.583 saniyedir.

Zampirolli vd. (2013) çalışmasında, çoktan seçmeli sınavlar için otomatik bir düzeltme sistemi önermişlerdir. Sistem cevap seçeneklerini tespit etmek için bir görüntü işleme tekniği kullanır. Her cevap sayfası taranır ve ikili formata dönüştürülür. Daha sonra bölümleme için basit morfolojik operasyonlar kullanılır. Bu metodoloji Çoktan Seçmeli Testlerin otomatik olarak düzeltilmesi çin gerçek zamanlı probleme uygulanır. Öncelikle, kullanıcı görüntüyü diske kaydetmek için kameranın ön tarafındaki cevap anahtarını konumlandırır ve ardından test tipini değerlendirmek için görüntü ölçülür. Daha sonra öğrenci cevap kâğıtları okunabilir.

Sattayakawee (2013), Tayland'ın optik olmayan geleneksel grid cevap kâğıtlarının kullanarak test puanlaması için algoritma önermiştir. Algoritma, projeksiyon profili ve eşikleme yöntemlerine dayanmaktadır.

Pegasus Görüntüleme Şirketi, belge görüntülerinden OİT tanıma için bir Yazılım Geliştirme Kiti (Software Development Kit, SDK) sunmuştur. SDK şablon tanıma modu ve serbest tanıma modunu desteklemektedir. Bir OİT alanı, değerlendirilecek belirli sayıda sütun ve işaretleme satırlarını içeren bir dikdörtgen alan olarak tanımlanır. SDK bölgeyi yatay olarak taradıktan sonra aralarındaki boşluklardan ayrı olarak işaretlemeleri bulmak için dikey olarak da tarayabilir. Daha sonra, belirtilen işaretleme şekline dayanarak, hangi işaretleme alanlarının "doldurulmuş" olarak nitelendirileceğini belirlemek için koyu pikselleri sayarak işaretlemelerin ayrık alanlarını tarar. Pegasus'un tekniği, düz kağıdın baskısını ve tasarımını destekleyebilirken, ancak okuldaki uygulamada, çoktan seçmeli cevap tanıma başarı oranı, sınavın gerekliliklerini yerine getiremez (Parul vd., 2012).

Lomte, Mastud ve Thite (2012) çalışmasında, barkod algılama için bir görüntü işleme prosedürü sunmuşlardır. Önerilen sistemde, mükemmel bir sonuç elde etmek amacıyla içindeki gürültüyü ortadan kaldırmak için görüntü işleyen bir sistem oluşturmuşlardır. Görüntü eğer renkli görüntü ise görüntü yakalandıktan sonra, gri tonlamalı görüntüye dönüştürülür ve filtreleme işlemi gerçekleştirilir. Daha sonra önerilen model kullanılarak eşikleme algoritması ve post filtreleme işlemi gerçekleştirilir. Önerilen sistem barkodu çok hassas bir şekilde tanıyabilir ve dijital görüntülerden barkodları tanıyan güçlü geliştirici kütüphanesi kullanır. Sistem, barkodun yönünü 0^0 ila 360^0 derece arasında algılayabilir. Siyah beyaz, gri tonlamalı, paletlenmiş ve renkli resimlerdeki birden fazla barkodu algılar. Tanınan her barkodun türünü döndürür. Sistem yaklaşık 11 endüstriyel barkod tipini tanıyabilir. Hızlı, doğru ve kolay kullanımı sayesinde iş piyasasında çok faydalıdır.

Gyamfi ve Missah (2017) çalışmasında, MCQ tipi sınavlardan sonra öğrencilerin cevaplarını tespit etmek için bir OİT sistemi önermişlerdir. Tasarlanan sistem, görüntü ön işleme stratejileri ile piksel tabanlı denetimsiz sınıflandırma yaklaşımını kullanmıştır. Hız ve doğruluk açısından nesnel tabanlı denetimli veya denetimsiz

sınıflandırma ile OİT sistemlerinin verimliliklerini karşılaştırmıştır. Çalışma, önerilen örneklem tarafından kullanılacak bir OİT şablonunun tasarımında 50 öğrencinin fikirlerini dahil ederek başlar. Çalışma, üç görüntü işleme öncesi stratejisinin, iki boyutlu medyan filtrelemenin, kontrast sınırlı uyarlamalı histogram eşitlemesinin, tarama çizgilerinin ve standart Hough dönüşüm tekniklerinin etkilerini hesaplamak için altı doğruluk parametresi kullanmıştır. Bu stratejilerin önerilen sistemin doğruluğunu artırdığı kanıtlanmıştır. Çalışma sonunda işaretlemenin tespit edilmesinde işaretleme alanının merkezi ve çevresindeki piksellerin tespitine yönelik stratejiler önerilmiştir. Bu etiketler, satır-sütun tek boyutlu dizi matrislerinde tutulmuştur. Araştırma daha sonra önerilen piksel tabanlı eğitimsiz sınıflandırma OİT algoritmasının, nesne tabanlı eğitilmemiş sınıflandırmasının istatistiksel olarak hızlı ve doğru olduğu sonucuna varmıştır.

Saengtongsrikamon, Meesad ve Sodsee (2009) çalışmasında, basit tarayıcıda uygulanması ve OİT makinesi olarak kullanılması için OİT yazılımı geliştirmeye yönelik çalışmalar yapmışlardır. Bu çalışma Java dili kullanılarak geliştirilmiştir. Çalışmada, Şelale Modeli gelişimin ana yapısı olarak seçilmiştir. 4 çözünürlükte 5 tarayıcı ile 50 farklı öğrenci tarafından işaretlenmiş 50 adet test-cevap sayfası tarandıktan sonra, 1000 adet resim üretilmiş ve program tarafından yürütülmeye hazır hale getirilmiştir. Sonuç olarak, bu değerlendirme yoluyla yorumlanan 1.000 görüntü vardır. Uygulamada yanlış bir cevap tespit edilmemiştir. Sonuç olarak, değerlendirme sonuçları olumlu olmuştur, tahmin edilen hata sayısı 1.000'de 1 veya 0.1'in altındadır. Geliştirilen yazılımın kalitesi ve güvenilirliği bu nedenle kabul edilebilir sınırlar içerisindedir.

Lais (2002) çalışmasında, OCR'nin basılı veya yazılı metin karakterlerinin optik olarak taranan bitmap'lerinin ASCII gibi karakter kodlarına dönüştürülmesini tanımlamıştır. Bu, basılı kopya materyalleri, bilgisayarda düzenlenebilmesinin ve başka şekilde değiştirilebilecek veri dosyalarına dönüştürmenin etkili bir yoludur.

Talib vd. (2015) çalışmasında, yöntem iki aşamadan oluşmaktadır. Yaklaşım, eğitim aşaması ve tanıma aşaması olarak adlandırılan iki evreye sahiptir. Eğitim aşamasında, bir web kamerası kullanılarak OİT sayfa görüntüsü yakalanır, daha

sonra görüntü, yumuşatma filtre teknikleri kullanılarak işlenir. Ardından, şablon olarak kullanılan soru numarasıyla birlikte bir set cevap bloğu etrafında bir dikdörtgen ilgilenilen alan (ROI) seçilir. Daha sonra tanıma aşamasında, eşleme, şablon görüntüsünün OİT sayfasına yerleştirilmesiyle yapılır. Son olarak, şablon ve aday görüntü, yoğunluk değerleri kullanılarak karşılaştırılır ve aday yanıtın şablonla eşleşip eşleşmediğine karar verilir.

Patil vd. (2016) çalışmasında, OİT cevap sayfası tarama işlemi ve IR Sensörleri için FPGA uygulamasını kullanmışlardır. Bu çalışma, minimum düzeyde tanınabilir işaretleme alanına odaklanılmıştır. Minimum algılanabilir siyah işaretleme alanı çapını, IR sensörü ile ölçmek için çeşitli çaplarda bir işaretleme alanının içinden geçirilmelidir. Bu işlem, Şekil 2.5 'de gösterilmiştir.

Şekil 2.5. Farklı büyüklükteki siyah alanlar için her defasında çıkış eşiği ölçümü

Çıktı eşiği, Tablo 2.1'de gösterilen çeşitli büyüklükteki siyah kabarcıklar için her seferinde ölçülür.

Tablo 2.1. *Çeşitli çaplardaki işaretleme sonuçları*

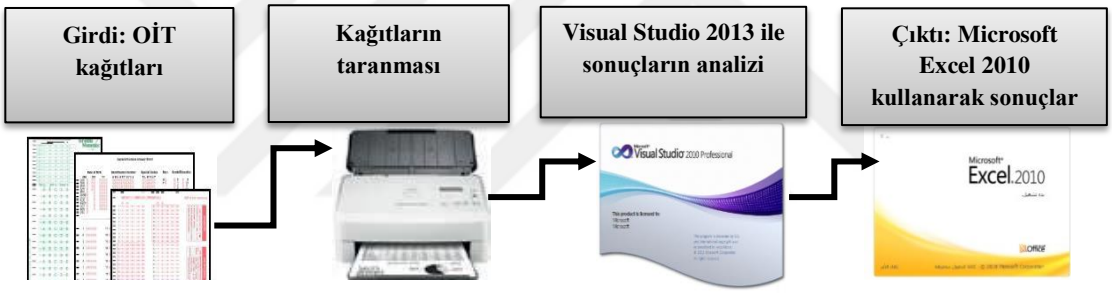
Çaplardaki İşaretlemeler (cm)	**Durum**
1,8	Tespit edildi
1,4	Tespit edildi
1,25	Tespit edildi
1	Tespit edildi
0,8	Tespit edilmedi
0,6	Tespit edilmedi
0,4	Tespit edilmedi

Bu bölümde, görüntü işleme kullanarak çoktan seçmeli testler için bir OİT'nin geliştirilmesine odaklanan çalışmalar sunulmuştur. Bu tez çalışmasında kullanılan yöntem, şablon cevap sayfası ve anahtar nokta tespiti oluşturulmasına dayanmaktadır. Tez çalışmasında geliştirilen yöntem ait tüm adımlar üçüncü bölümde açıklanmıştır.

3. YÖNTEM VE SİSTEM TASARIMI

Microsoft Visual Studio, Microsoft tarafından üretilen ve konsollar, grafik kullanıcı arayüzleri, Windows formları, web servisleri ya da web uygulamaları oluşturmak için kullanılan bir Tümleşik Geliştirme Ortamı (Integrated Development Environmet, IDE)'dır. Microsoft Visual Studio (Visual Studio.NET) birkaç farklı programlama dili (Visual Basic, Visual C #, Visual C ++, Jscript, ASP.NET) içermektedir.

Temel bir Visual Basic uygulaması Şekil 3.1.'de gösterilmektedir. Bu çalışmada, Microsoft Visual Basic (VB) kullanılmıştır. Bu program, iş sektöründe milyonlarca satır kod ve her türlü uygulamalarda kullanılan dünyanın her yerindeki en popüler programlama dilidir.



Şekil 3.1. Uygulama şeması

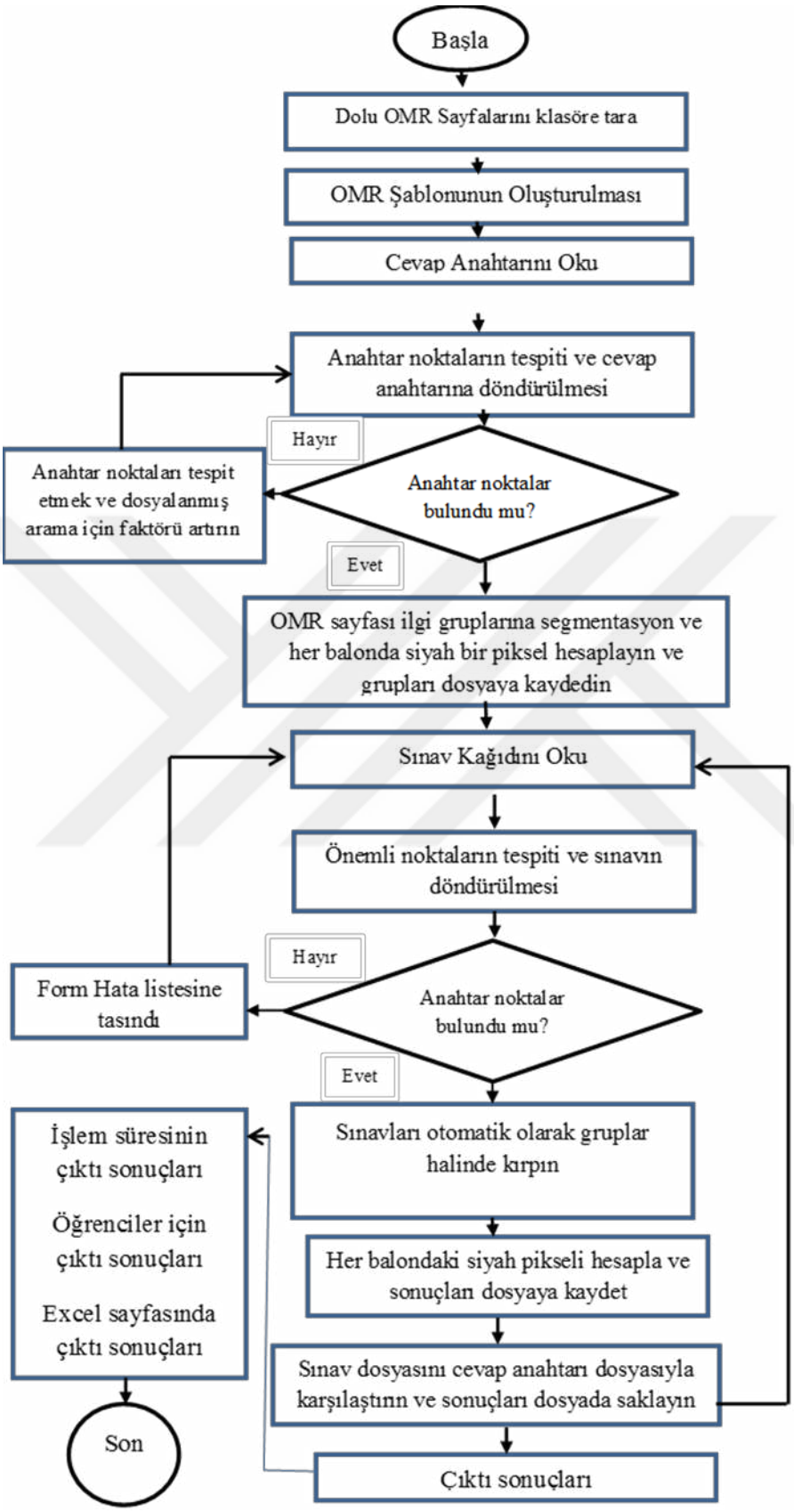
Bu program dijital görüntü işlemeye dayalı çoktan seçmeli soruların düzeltilmesi amacıyla gelişmiş ve otomatik sistem olarak kullanılır. Bu sistemde, OİT cevap kâğıtlarına girmek ve sayfa şablonunu bilgisayara aktarmak için tarayıcı kullanılmıştır. Tüm görüntüler yazılım sistemine girdi olarak verilecektir. Bu teknik beş ana adımdan oluşmaktadır. İlk aşamada, şablonun koordinatları seçilerek ve bir dosyada saklanmak koşuluyla şablon oluşturulur. İkinci aşamada anahtar noktası tespiti ve cevap anahtarının gerekli alanlarının kırpılması yapılır. Üçüncü aşama anahtar nokta tespiti ve sınav kâğıdını ilgi alanlarına bölme işlemleri yapılır. Dördüncü aşama, tüm sınavları cevap anahtarıyla karşılaştırma ve sonucu dosyaya kaydetmeyi içerir. Son aşama ise sonuçların gösterilmesidir.

3.1. Uygulamanın Yapısı

Gerçekleştirilen sistemde, mükemmel bir sonuç elde etmek için OİT cevap kâğıtlarını işleyen bir uygulama oluşturulacaktır. Şekil 3.1'de tüm uygulama gösterilmektedir.

3.2. Yöntem Tasarımı

Başvurulan yöntem, cevap sayfası şablonu ve anahtar nokta tespiti (başlangıç noktası ve bitiş noktası) oluşturulmasına dayanır. Yöntemin genel adımları Şekil 3.2'de sunulmuştur.



Şekil 3.2. Yöntemin temel adımlarının akış şeması

Özellikle, bu yöntemin adımları aşağıdaki alt bölümlerde ayrıntılı olarak açıklanmaktadır.

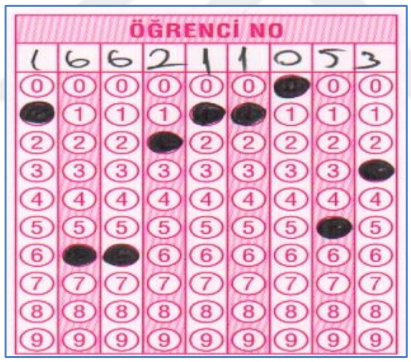
3.2.1. Birinci Adım

OİT görüntülerinin girilmesinde, kâğıtlarının taramak, verileri bilgisayara aktarmak ve jpeg görüntüsü ile bilgisayarın belleğinde saklamak için tarayıcı kullanılmıştır.

3.2.2. İkinci Adım

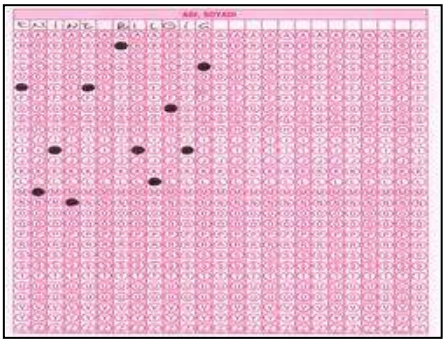
Aday taslaklarının seçiminin yapılmasında şablondaki bölgelerin konumu, yöntemin başarısı için çok önemlidir. Oluşturabileceğiniz dört bölge türü vardır. Bu bölge, formdaki OİT işaretleme alanlarını, onay kutularını veya başka bir tam şekli içeren bir alandır.

- Öğrenci numarası bölgesi: öğrenci numarası 10 haneli, 0-9 rakamlar kullanılarak işaretlenir. Şekil 3.3'de gösterilmiştir.



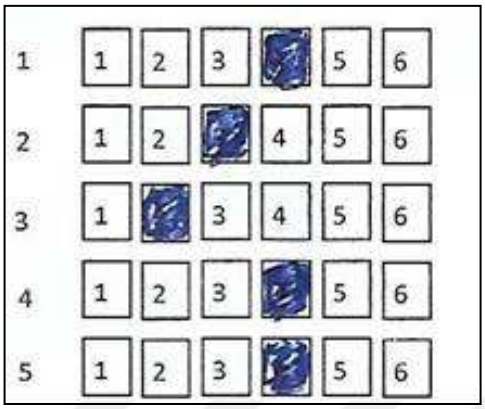
Şekil 3.3. Öğrenci numarası bölgesi

- Öğrenci isim bölgesi: 27 (A-Z) alfabe harfinden oluşur. Şekil 3.4'de gösterilmiştir.



Şekil 3.4. Öğrenci isim bölgesi

- Çoktan Seçmeli Soru Bölgesi: çoktan seçmeli sorular, öğrencilerin bir listedeki seçimlerden en iyi cevabı seçerek yaptığı bir işaretleme şemasıdır. Şekil 3.5'de çoktan seçmeli soru örneği gösterilmiştir.



Şekil 3.5. Çoktan seçmeli soru bölgesi

- Barkod Bölgesi: Barkod, isimlerden kimlik numaralarına ve diğer bilgilere kadar, istenen herhangi bir bilgiyi kodlamak için kullanılır. Şekil 3.6'da, borkod bölgesi örneği gösterilmiştir.

Şekil 3.6. Barkod bölgesi

Koordinatların başlangıcını ve sonu seçilerek koordinat seçimi yapılmaktadır. Şekil 3.7'de koordinat seçimi gösterilmiştir.

Akış olarak şablon koordinatlarını seçmek için kullanılan algoritma şu şekildedir:

Fareye sol tıklandığında // sol tıkın kullanılmasına izin ver
Seçenek işaret alanı kontrol edilirse // işaret alanı seçimi için:

bubx1 ← koordine etmek için x’e tıklayın

buby1 ← koordine etmek için y’e tıklayın

SonDeğerİşaretleme alanı seçenekleri başlangıç noktası kontrol edilirse // başlangıç noktası seçimi için

stpointx1 ← koordine etmek için x’e tıklayın

stpointy1 ← koordine etmek için y’e tıklayın

Son Değer

enpointx1 ← koordine etmek için x’e tıklayın

enpointy1 ← koordine etmek için y’e tıklayın

Son Değer

grpp.gridx1 ← koordine etmek için x’e tıklayın

grpp.gridy1 ← koordine etmek için y’e tıklayın

Son Değer

Son Değer

Fare sağ tıklandığında //sağ tıkın kullanılmasına izin verin

Seçenek işaret alanı kontrol edilirse //işaret alanı boyutu seçimi için

Bubx2 ← koordine etmek için x’e tıklayın

Buby2 ← koordine etmek için y’e tıklayın

Son DeğerİşaretleMe alanı seçenekleri başlangıç noktası kontrol edilirse

Stpointx2 ← koordine etmek için x’e tıklayın

Stpointy2 ← koordine etmek için y’e tıklayın

Son DeğerİşaretleMe alanı seçenekleri bitiş noktası kontrol edilirse // bitiş noktası seçimi için

Enpointx2 ← koordine etmek için x’e tıklayın

Enpointy2 ← koordine etmek için y’e tıklayın

Son Değeri

şaretleme alanı seçenekleri kontrol edilirse // alan seçimi için

grpp.gridx2 ← koordine etmek için x’e tıklayın

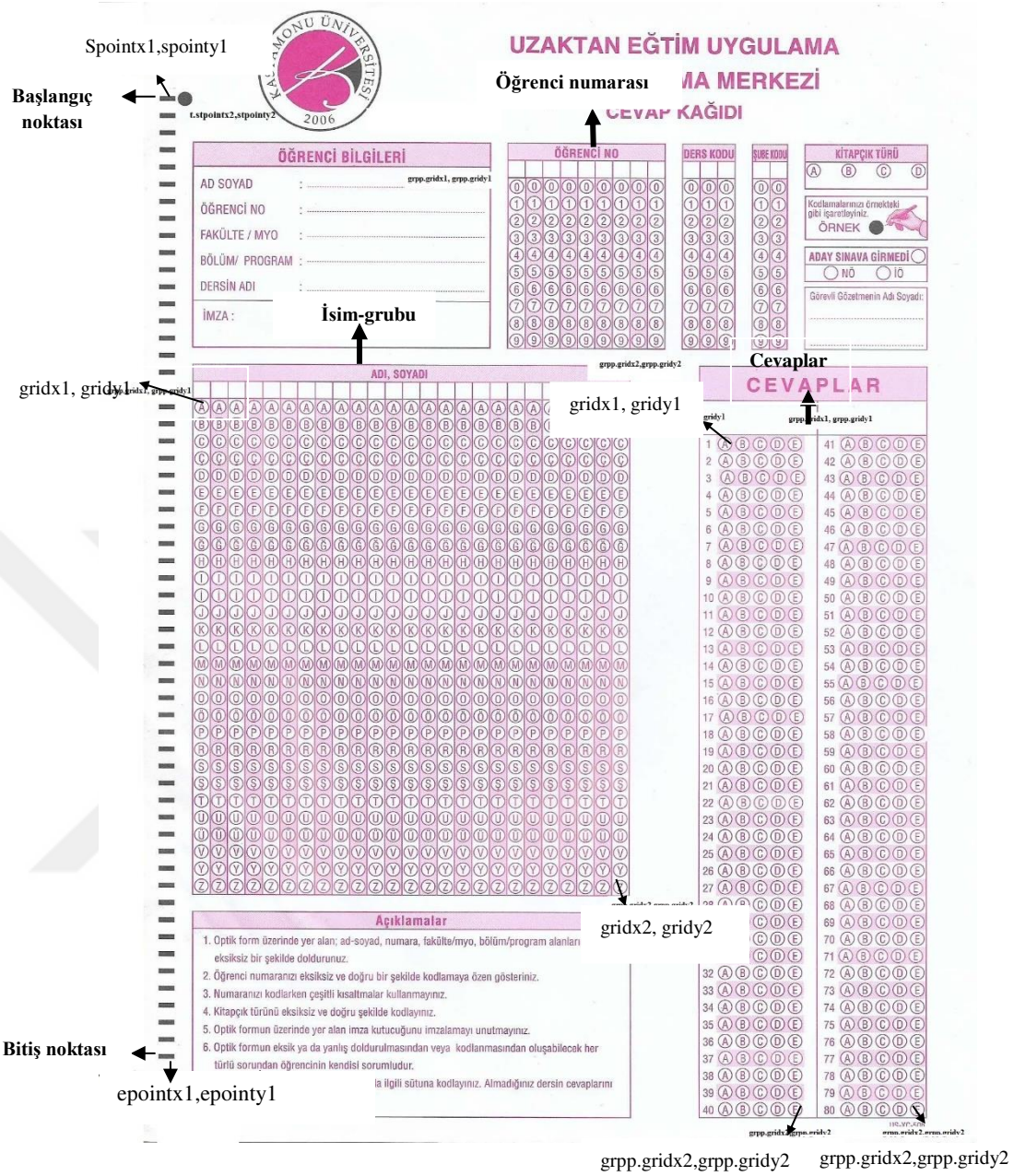
grpp.gridy2 ← koordine etmek için y’e tıklayın

Son Değer

Son Değer

Şablonun koordinatlarını bilgisayarda herhangi bir dosyaya kaydetme işlemi için geliştirilen algoritma aşağıdaki verilmiştir.

```
Private Sub ToolStripButton5
File nam = "temp\" & ToolStripTextBox4.Text & ".tmp" // bilgi metnini yazınız
My file = File.Create Text (dosya adı)        // dosyaı aç
Myfile.WriteLine (Boyut.Genişlik, Boyut. Yükseklik) // şablonun boyutlarını
saklamak için
Myfile.WriteLine (stpointx1, stpointy1, stpointx2, stpointy2…etc.) //
başlangıç ve bitiş noktalarını saklamak için.
her i grup türü        // her grubun koordinatlarını saklamak için döndürün.
Myfile.WriteLine (i.tür, i.isim, i.sıra, i.sutun, grpp.gridx1, grpp.gridy1
grpp.gridx2 , grpp.gridy2 …gibi.) // her grubun koordinatlarını saklamak için.
Next
My file.Close ( ) // close file.
```



Şekil 3.7. Şablondaki önemli alanlar

Şekil 3.8'de, dosyadaki şablon koordinatlarının sonuçları gösterilmektedir. Bu sonuçlara ait açıklama aşağıda verilmiştir:

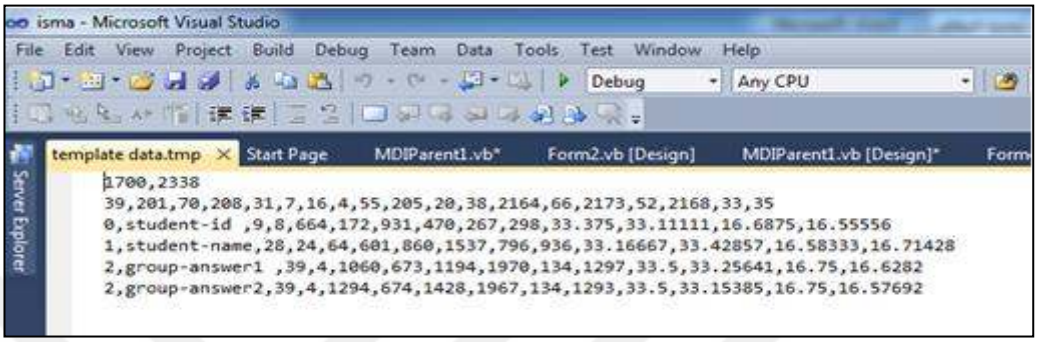
Satır 1: Şablonun boyutu (genişlik ve yükseklik).

Satır 2: Şablonun koordinatları (başlangıç noktası, bitiş noktası, başlangıç noktasının merkezi ve bitiş noktasının merkezi, vb.).

Satır 3: Öğrenci numarası bölgesini koordinatları

Satır 4: İsim bölgesinin koordinatları,

Satır 5 ve 6: Cevap grubunun koordinatları.



Şekil 3.8. Bir dosyada saklanan şablonun koordinatları

3.2.3. Üçüncü Adım

Bu aşamada anahtar noktaların tespiti yapılmaktadır. Şablon oluşturma ve şablona ait koordinatları dosyaya kaydet etme aşamasından sonraki aşamada geliştirilen algoritma cevap anahtarının ana noktalarını tespit etme amacı ile oluşturulmuştur. Bu noktalar iki kısma ayrılmıştır. Birinci kısım, algılama algoritmasını şablonun başlangıç noktasına uygulamaktadır. İkinci kısım da ise algılama algoritmasını şablonun bitiş noktasına uygulanmaktadır.

Başlangıç noktası tespiti için uygulanan ilk adımda renkli pikseli gri piksele dönüştürülmesi ve insan gözü için gerekli düzeltme yapılmaktadır. Siyah/beyaz ekranda renk olmadığı için, renkler grinin farklı tonlarıyla ifade edilir. Gözün, renklere farklı tepkisi dikkate alındığında parlaklık işareti Denklem 3.1'de belirtilen oranlardan renklerin karıştırılmasıyla oluşur.

$$COLOR \ = (0.3R + 0.59G + 0.11B) \tag{3.1}$$

Bu durumda sadece mavi renk ifade edilmek istendiğinde, R=G=0, B=1 (max) olacak ve parlaklık işareti Y=0.11 olarak açık gri ton ile mavi ifade edilecektir. Renkli pikselin gri piksele dönüşüm algoritması aşağıda verilmiştir:

C ← Pikseli getir (x, y) // piksel değerini oku

Gri ← C.R * 0.2989 + C.G * 0.587 + C.B * 0.114 // Parlaklık yöntemini renk pikselini gri piksele dönüştürmek için parlaklık yöntemini uygulanması.

Bir sonraki adımda gri pikselleri siyah piksellere dönüştürmek için eşik değeri uygulamaktır. 128 eşik değeri kullanılmıştır. Eşik değerinden daha küçük olan piksel, beyaza ve geri kalanı ise siyah piksele çevrilir. Gri pikselin siyah ve beyaz piksele dönüşüm algoritması aşağıda verilmiştir:

Eşikler ← 128 // eşik değeri.

Eğer gri< sholds ise// eşik değerleri uygulayarak gri görüntüyü ikili görüntüye dönüştürme.

c←0 // siyah olarak sınıflandırılan piksel.

else c←255 // beyaz olarak sınıflandırılan piksel.

Son adımda algılama algoritması aşağıdaki gibi başlangıç noktasına uygulanır:

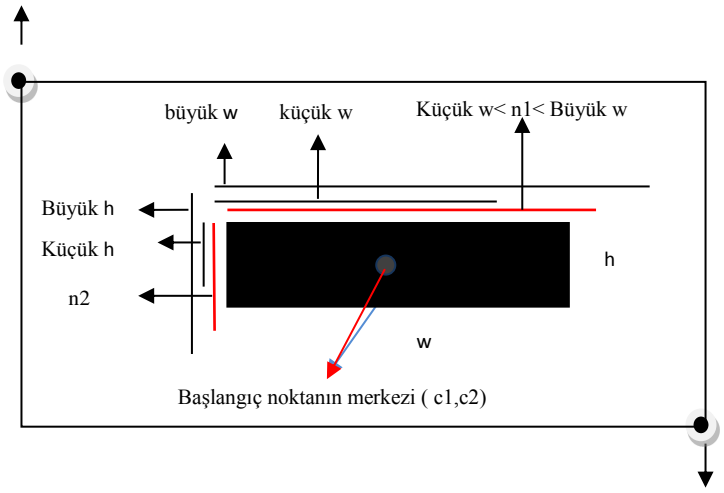
- Algoritma, n1 şablonunun başlangıç noktası uzunluğuna yakın olan ilk siyah yatay çizgiyi bulmak için başlangıç noktasının arama alanının içinde algılayacaktır.
- Algoritma, n2 şablonunun başlangıç noktasının genişliğine yakın olan ilk siyah dikey çizgiyi bulmak için arama alanının içinde algılayacaktır.
- Başlangıç noktasının (c1, c2) merkezini Denklem 3.2 ve 3.3'te olduğu gibi bulunuz.

  C1 = t.stpointx1 + ((t.stpointx2 - t.stpointx1) / 2) (3.2)

  C2 = t.stpointy1 + ((t.stpointy2 - t.stpointy1) / 2) (3.3)

Bu adımlar Şekil 3.9’da gösterilmektedir.

Araştırma alanın başlangıcı



Araştırma alanın bitiş noktası

Şekil 3.9. Başlangıç noktası algılama algoritması

Burada w: anahtar noktasının genişliği ve h: anahtar noktasının yüksekliğidir. Küçük w: w- hata oranı * w, Büyük w: w + hata oranı * w; Küçük h: h- hata oranı * h, Büyük h: h + hata oranı * h. hata oranıdır. Başlangıç noktasının uzunluğunun veya genişliğinin maksimum % 25'idir.

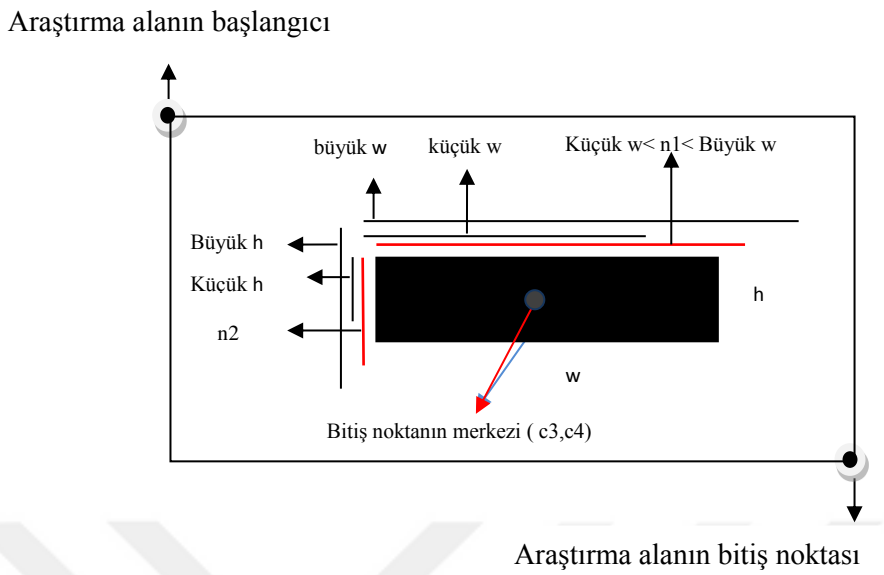
Bitiş noktası algılama: Bitiş noktasını tespit etmenin ilk adımı, parlaklık yöntemini kullanarak pikselleri gri piksellere dönüştürmek ve gri pikselleri siyah piksellere dönüştürmek için eşik değerini uygulamaktır. Bir sonraki adımda, algılama algoritmasını aşağıdaki gibi bitiş noktasına uygulanır:

- Algoritma, n1 şablonunun bitiş noktası uzunluğuna yakın olan ilk siyah yatay çizgiyi bulmak için bitiş noktasının arama alanının içinde algılayacaktır.
- Algoritma, n2 şablonunun bitiş noktasının genişliğine yakın olan ilk siyah dikey çizgiyi bulmak için arama alanının içinde algılayacaktır.
- Başlangıç noktasının (c1, c2) merkezini Denklem 3.4 ve 3.5'te olduğu gibi bulunuz.

$$C3 = t.enpointx1 + ((t.enpointx2 - t.enpointx1) / 2) \quad (3.4)$$

$$C4 = t.enpointy1 + ((t.enpointy2 - t.enpointy1) / 2) \quad (3.5)$$

Bu adımlar Şekil 3.10'de gösterilmektedir.



Şekil 3.10. Bitiş noktası algılama algoritması

3.2.4. Dördüncü adım

Başlangıç noktası ve bitiş noktası tespit edildikten sonra, algoritma başlangıç noktası ve bitiş noktasının merkezinden doğru açıyı (θ) Denklem 3.6 ve 3.7 ile hesaplanır. Bu adımın amacı, resmi normal dikdörtgene geri yüklemek için hesaplanan bir açıda döndürmektir. Düz çizginin θ açısı denklemi şu şekilde verilir:

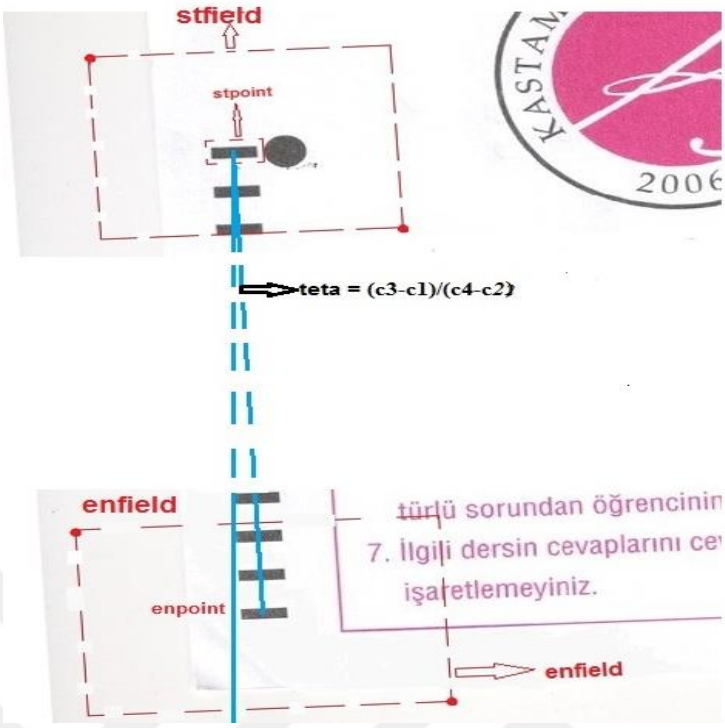
$$\theta = (c3 - c1)/(c4 - c2) \tag{3.6}$$

$$\text{Dönme açısı} = \theta * 180/\pi \tag{3.7}$$

Bu algoritmada şu fonksiyon kullanılır:

mat.Rotate At (θ, New Point(t.stpointcx, t.stpointcy)) // dönüş için

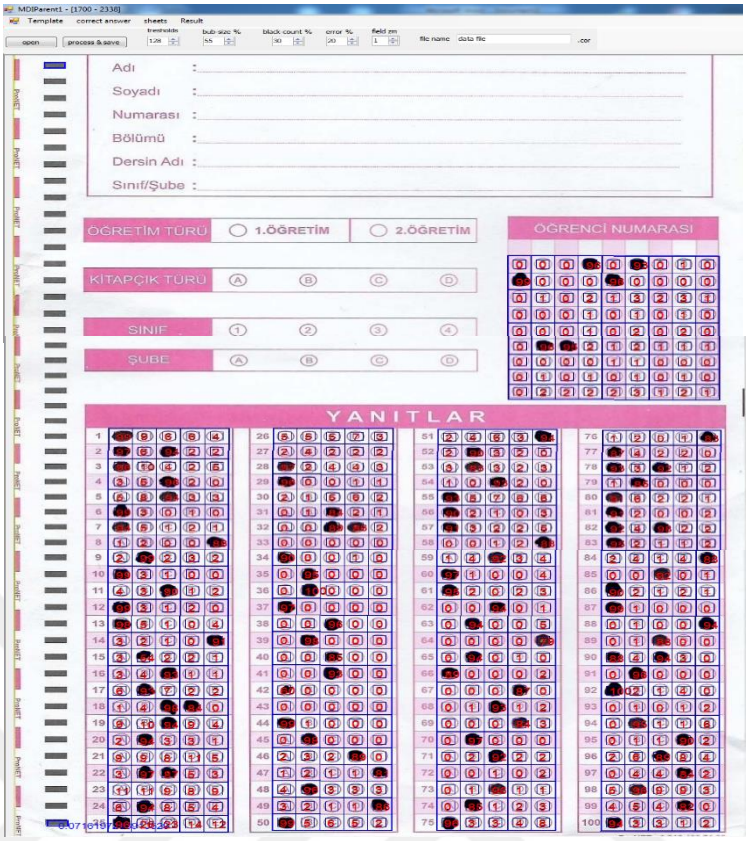
Dönüş açısı, Şekil 3.11'de gösterilmektedir.



Şekil 3.11. Dönüş açısının(θ) hesaplanması

3.2.5. Beşinci adım

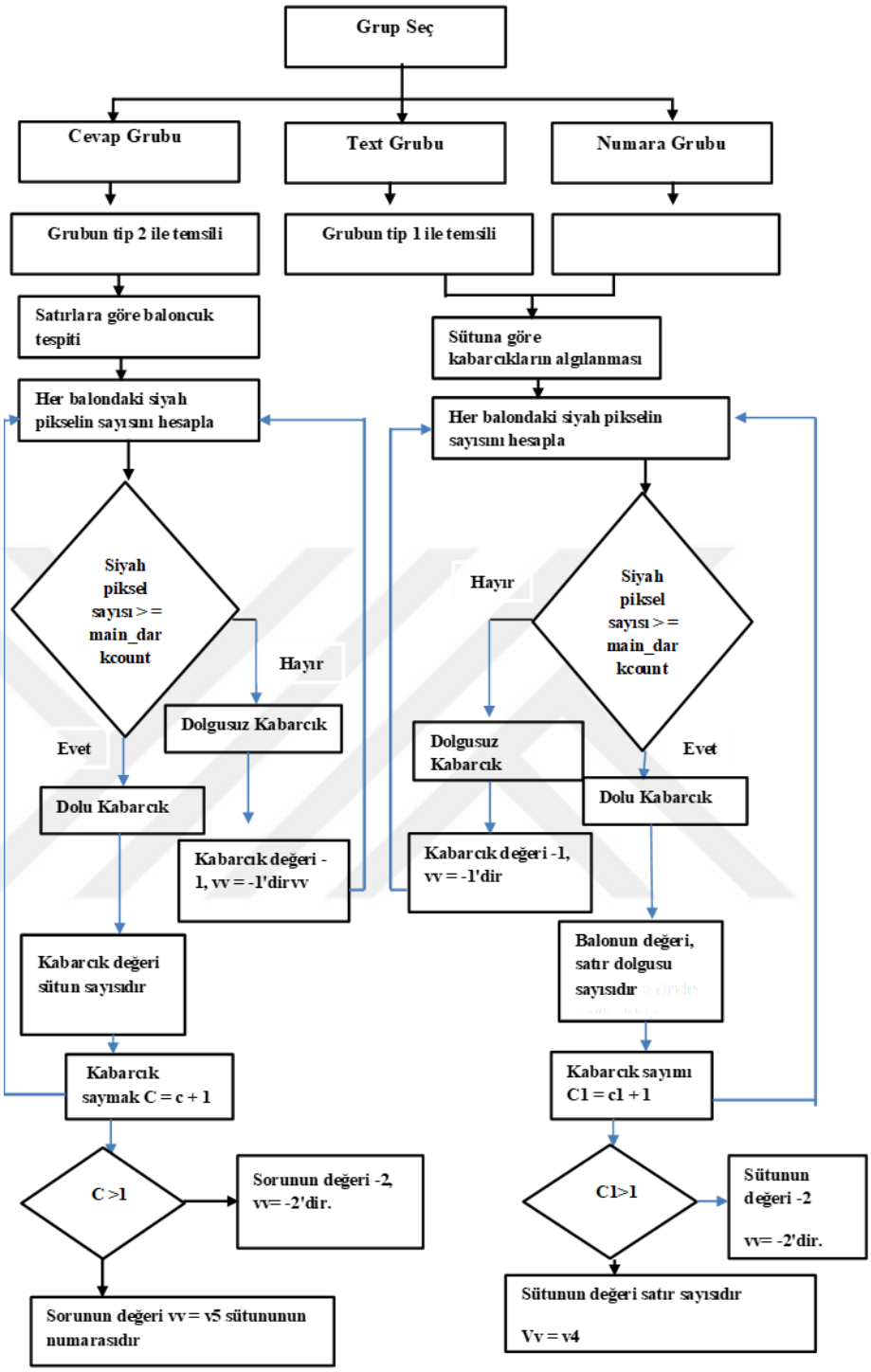
OİT sayfası, Şekil 3.12'te gösterildiği gibi öğrenci kimliği, öğrenci adı ve cevap alanları olmak üzere üç bölgeye otomatik olarak kırpılacaktır. Kırpma işlemi görüntülerin boyutunu azaltacaktır (Krishna, vd., 2013). Bu adımda, işaretlenecek her bir cevap kağıdının alan içine yerleştirilmesi ve daha sonra projeksiyon profili ve eşik yöntemleri uygulanması gerekir. Projeksiyon profili ve eşikler, cevap kâğıdındaki tüm bilgileri tanımlamak için kullanılır. Cevaplar için yatay projeksiyon profili seçim veya işaret alanı sınırını belirlemek ve her satırdaki işaret alanı veya seçimler için siyah değerlerin sayılmasında kullanılır. Öğrenci kimliği ve öğrenci adı için her seçim veya işaret alanı sınırı için dikey projeksiyon profili tanımlaması ve her bir kolonun her bir sütununda işaret alanı veya seçim için siyah değerlerin sayılması ve her satırdaki işaret alanları için siyah değerlerin sayılması gerekir.



Şekil 3.12. OİT kağıdının bölünmesi

3.2.6. Altıncı adım

Bu adımın amacı, grupların temsili için gruplar arasındaki sayıyı her bir farklı grup için şekil 3.13 te gösterildiği gibi tanımlamaktır:



Şekil 3.13. Grup gösterimi akış şeması

- Öğrenci numarası alanı 9 rakamlı bir dizidir ve dikey yönde düzenlenmiştir. Algoritmada tip 0 ile temsil edilir. Bu türün işlenmesi her balondaki siyah piksel sayısının sayılması ve bunları koyu sayılar (koşul) ile karşılaştırmak için sütunlar kullanılarak yapılır. Bu çalışmada 30

koşulunu kullandık ve bu koşul otomatik olarak değiştirilebilmektedir. Siyah piksel sayısının koyu sayısından fazla olması durumunda, işaret alanı doldurulur. Aşağıdaki olasılıkları göz önünde bulundurmak için sütunlarda daha fazla doldurulmuş alan olabilir. Sütunda bir seçim yapılması durumunda, seçim değeri satır sayısıdır. Sütunda herhangi bir seçim yapılmadığı takdirde, seçim değeri -1'dir. Sütunda seçili durumdan daha fazlası durumunda, seçim değeri -2'dir. Şekil 3.14'te öğrenci numarası bölgesini temsil eden algoritma gösterilmektedir. Algoritma aşağıdaki gibidir:

Eğer tip = 0 ise v1 ←col v2 ←satır // satır ve sütun sayısına bakılır
Sütunun Dikey yön stili için iç içe döngü yapısı uygulanır
gx ← 0 v1 için // her bir sütun için deneyin
vv ← -1 // cevap yok ise bakın.
vlu ← " " // satır değerine bakın
For gy ← 0 To v2 // her hiç satıra bakın
v3 ← gy : v4 ← gx // v3 satırı ve v4 sütunu temsil eder.
lbs ← 0 // siyah pikseller sayılır Bu adımda, işaret alanının merkezinin yeri tespit edilir.
Cellw←gridgw * cellwper / 2 // işaret alanının çapı hesaplanır
lbw=-cellw to +cellw // işaret alanının genişliği hesaplanır.
lbh=-cellw to +cellw // işaret alanının yüksekliği hesaplanır.
newcol ← pic.GetPixel // işaret alanındaki her bir siyah piksel okunur.
Bu adımda, piksellerin dönüşümü ve siyah piksel sayısının hesaplanması gerçekleştirilir.
gry← gray (newcol) // önceki bölümdeki renkli pikselinden gri piksele dönüşüm algoritmasında gösterildiği gibi renk piksel gri piksele dönüştürülür.
Eğer gry = 0 ise // Sınıflandırılmış Piksel için siyah olarak koşullandırılır
lbs += 1 // Siyah piksel sayısı bir artırılır.
Son tekrar
Son tekrar

Bu adımda, işaret alanındaki siyah pikselin toplamını koşulla karşılaştırılır.

Eğer lbs >= main_darkcount ise

v5 ← v3 // satır sayısı

Eğer v5 = v3 İse vv = v5 : c=c+1

Eğer c >1

vv=-2

Son tekrar

Sonuçların elde edilmesi

Eğer vv = -1 ise vlu ← vv // cevap yoktur.

Başka eğer vv=-2 ise vlu ← vv // seçilen birden fazla cevap vardır.

Eğer vlu ← vv // cevap vardır

Bu adımda sonuçlar bir dosyaya kaydedilir

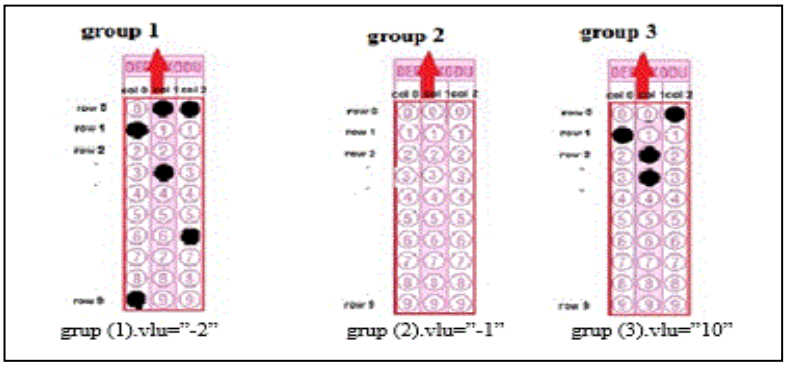
Corfile.Write(vlu) // Sonuçlar bu işlemlerden sonra dosyada saklanır

vlu = " " : vv = -1 :C=0

Son tekrar

Son tekrar

Algoritmanın sonu



Şekil 3.14. Öğrenci numarası

- Karakter bölgesi alanı 28 harften oluşmakta (A-Z) ve dikey yönde düzenlenmektedir.

- Tüm karakterler "ABCDEFGHIJKLMNOPQRSTUVWXYZ" matrisinde saklanır. Algoritmada tip 1 ile temsil edilmektedir. Bu türün işlenmesi, her sütunda koyu alanı algılamak için sütun kullanılarak yapılır. Sütundaki seçim değeri satır sayısıdır ve Şekil 3.15'de gösterildiği gibi

matris konumu değerine göre temsil edilen satır sayısına eşittir. Algoritma aşağıdaki gibidir:

Eğer tip = 0 ise v1 ←col v2 ←satır // satır ve sütun sayısına bakılır.
Sütunun Dikey yön stili için iç içe döngü yapısı uygulanır
gx ← 0 To v1 // her sütun için tekrar edilir
vv ← -1 // cevap yok.
vlu ← " " // sütunun değerine bakılır
gy ← 0 To v2 // her sütun için tekrar edilir
v3 ← gy : v4 ← gx // v3 satırı ve v4 sütunu temsil eder.
lbs ← 0
Bu adımda, işaret alanının merkezinin yeri tespit edilir.
Cellw←gridgw * cellwper / 2 // İşaret alanının çapı hesaplanır.
lbw=-cellw to +cellw // işaret alanının genişliği hesaplanır.
lbh=-cellw to +cellw // işaret alanının yüksekliği hesaplanır.
newcol ← pic.GetPixel (t.stpointcx + lbw + .gridx1 + v4 * .gridgw),
(t.stpointcy + lbh + .gridy1 + v3 * .gridgh)
Bu adımda, piksellerin dönüşümü ve siyah piksel sayısının hesaplanması gerçekleştirilir.
gry← gray (newcol)
Eğer gry = 0 ise lbs += 1 // Piksel Siyah olarak sınıflandırılır ve siyah sayılır
Son tekrar: son tekrar.
Bu adımda, işaret alanındaki siyah pikselin toplamını koşulla karşılaştırılır.
Eğer lbs >= main_darkcount ise
v5 ← v3 // satır sayısı
Eğer v5 = v3 else vv = v5 : c=c+1
Eğer c >1
vv=-2
Son tekrar
Bu adımda sonuçlar elde edilir
Eğer vv = -1 ise vlu ← vv // cevap yoktur.
Eğer vv= -2 ise vlu ← vv // Bir fazla seçilen cevap vardır.
Eğer nm(vv) ← vv :vlu← nm(vv) // cevap vardır.

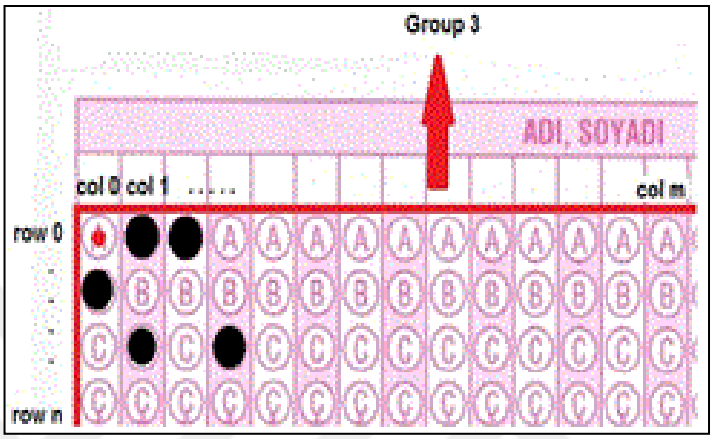
Bu adımda sonuçlar bir dosyaya kaydedilir

Corfile.Write(vlu) // Sonuçlar bu işlemlerden sonra dosyada saklanır

vlu = " " : vv = -1 :C=0

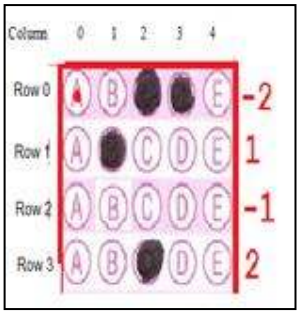
Son tekrar

Algoritma sonu.



Şekil 3.15. Karakter alanı

- Cevap Bölgesi, soru gruplarını içermektedir. Her sorunun seçenekleri vardır. Bu alanlar tip 2'de temsil edilmektedir. Bu bölgenin işlenmesi, her satırdaki siyah alanı algılamak için satır kullanılarak yapılır. İki cevap seçeneği seçildiğinde, soru değeri -2'dir, seçimin yapılmadığı durumda ise soru değeri -1 olur, bir seçenek seçilirse soru değerini alır. Şekil 3.16'da gösterildiği gibi sütun sayısıdır.



Şekil 3.16. Cevap grubunu temsil etmektedir

Tüm bu süreçler şekilde gösterildiği gibi dosyada saklanır. Algoritma aşağıdaki gibidir:

İşaret alanı algılama tespiti yapılır

Eğer .tip = 2 ise v1 ←row v2 ←col // satır ve sütun sayısına bakılır

Sütunun Dikey yön stili için iç içe döngü yapısı uygulanır

gx ← 0 To v1 // her satır için tekrar edilir

vv ← -1

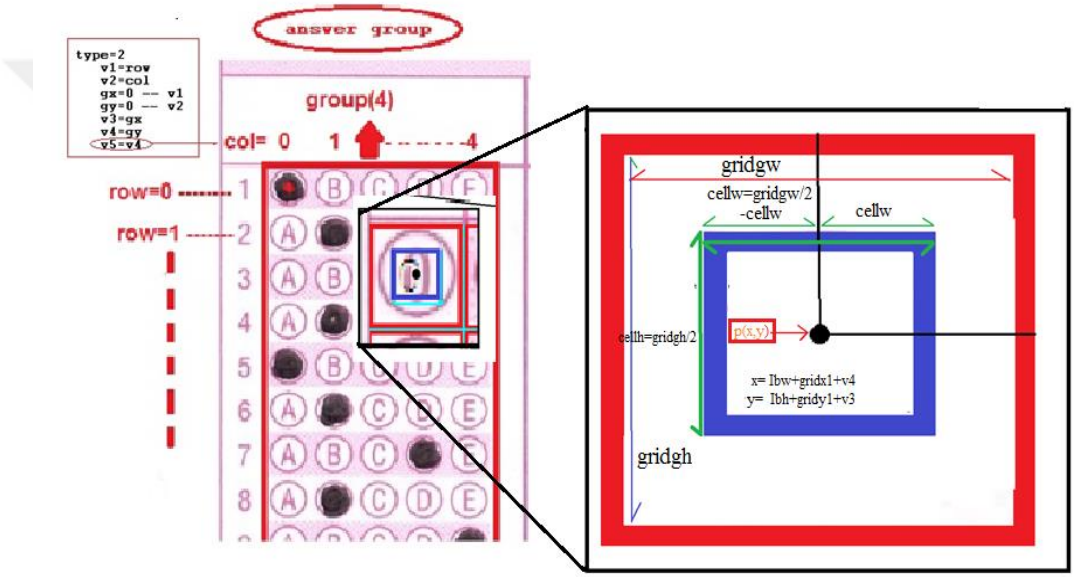
vlu ← " " // satır değerine bakılır

gy ← 0 To v2 // her sütun için tekrar edilir

v3 ← gy : v4 ← gx

lbs ← 0 // Piksel sayısı hesaplanır

Bu adımda, Şekil 3.17'da gösterildiği gibi merkez kabarcığın yerini bulacağız.



Şekil 3.17. Merkez Kabarcığın yerinin gösterimi

Bu adımda, işaret alanının merkezinin yeri tespit edilir.

Cellw←gridgw * cellwper / 2

lbw=-cellw to +cellw // İşaret alanının genişliği hesaplanır.

lbh=-cellw to +cellw // İşaret alanının yüksekliği hesaplanır.

newcol ← pic.GetPixel (t.stpointcx + lbw + .gridx1 + v4 * .gridgw),

(t.stpointcy + lbh + .gridy1 + v3 * .gridgh).

Bu adımda, piksellerin dönüşümü ve siyah piksel sayısının hesaplanması gerçekleştirilir

gry← gray (newcol).

Eğer gry = 0 ise lbs += 1 // Piksel Siyah olarak sınıflandırılır ve siyah

piksel sayılır.

Son tekrar: Son tekrar

Bu adımda, işaret alanındaki siyah pikselin toplamını koşulla karşılaştırılır.

Eğer lbs >= main_darkcount ise // koşulla karşılaştırılır.

v5 ← v4 // v5'deki satır sayısı kaydedilir

Eğer v5 = v4 ise vv = v5 : c=c+1

Eğer c >1

vv=-2

Son tekrar

Bu adıma sonuçlar elde edilir

Eğer vv = -1 ise vlu ← vv // cevap yoktur.

Başka eğer vv=-2 ise vlu ← vv // birden fazla cevap seçilmiştir.

Eğer vlu ← vv //cevap vardır.

corfile.Write(vlu)

vlu = " " , vv = -1

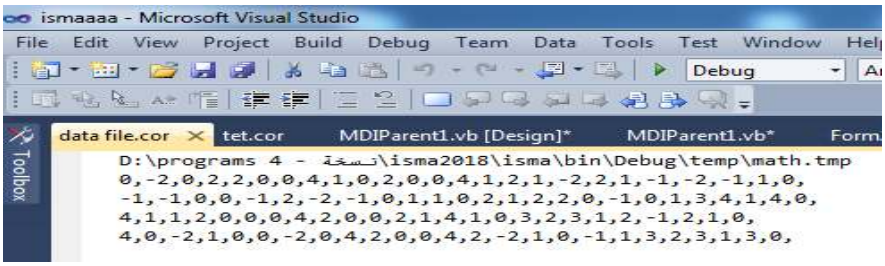
Son Tekrar

Son tekrar

Algoritma sonu

3.2.7. Yedinci Adım

Tüm sınav kâğıtlarının önceden kayıtlı cevap anahtarı dosyasıyla karşılaştırılıp okunması adımı bu bölümde yer almaktadır. Bu adımda, algoritma her bir cevap kâğıdını otomatik olarak okur ve 2-5 arası adımları tekrar eder ve daha sonra, test kâğıdının tespit edilen cevaplarını Şekil 3.18'da önceden kaydedilmiş dosya cevap anahtarıyla karşılaştırır ve sonuçları dosyaya kaydeder.



Şekil 3.18. Cevap alanını kırptıktan sonra kayıtlı dosyanın sonucu

3.2.8. Sekizinci adım

Tüm sınav kâğıtları işlendikten sonra, sonuçlar excel sayfasında çıktı olarak gönderilir. Bu sonuçlar, her öğrenci için öğrenci numarası, öğrenci adı ve puanları gibi tüm bilgileri içerir. Elde edilen sonuçlar dördüncü bölümde gösterilecektir.

Akış olarak excel sayfasındaki sonuçların algoritması:

```
Dim excel As New Microsoft.Office.Interop.Excel.ApplicationClass //
uygulama sınıfının belirlenmesi
Dim wBook As Microsoft.Office.Interop.Excel.Workbook      // çalışma
kitabının belirlenmesi
Dim wSheet As Microsoft.Office.Interop.Excel.Worksheet   // çalışma
sayfasının belirlenmesi
wBook = excel.Workbooks.Add(System.Reflection.Missing.Value)
wSheet = wBook.Sheets(1)      //ilk çalışma sayfasına sonuçların
kaydedilmesi
i = 0 to DataGridView1.RowCount - 1
j = 0 To DataGridView1.ColumnCount - 1
wSheet.Cells(i + 1, j + 1).value = DataGridView1.Rows(i).Cells(j).Value
//gridview'den verilerin okunması ve verileri çalışma sayfasına
kaydedilmesi.
Sıradaki  j
Sıradaki i
algoritma sonu.
```

3.3. Sistem Arayüz Tasarımı

Bu uygulamada, çıktıyı görüntülemek için bir etikete ihtiyacımız vardır. Ekranda ne olduğunun tanımlaması için Metin Kutusu, Düğmeler gibi diğer başka etiketlere de ihtiyaç olacaktır. Genel olarak bu uygulama, grafiksel kullanıcı ara yüzünü kullanan 5 büyük tasarımdan oluşur: şablon tasarımı, doğru cevap, sayfa işleme ve sonuç görüntüleme. Tasarım ara yüzü, Şekil 3.19'de gösterilmiştir.

3.3.1. Şablon Tasarımı

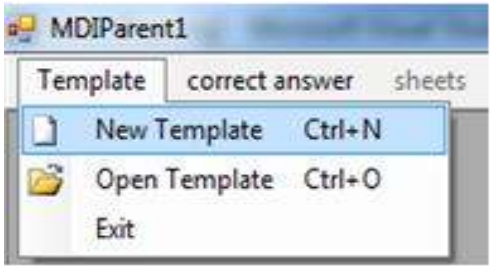
Şablonun tasarımı üç önemli alt işlevi içerir: Yeni bir şablon oluşturma, Eski bir şablon dosyasını açma ve çıkış.

3.3.2. Yeni Şablon Tasarımı

Başlangıçta, Şekil 3.19'de gösterildiği gibi şablonun koordinatlarını almak amacıyla yeni görüntü şablonunu açmak için kullanılır.

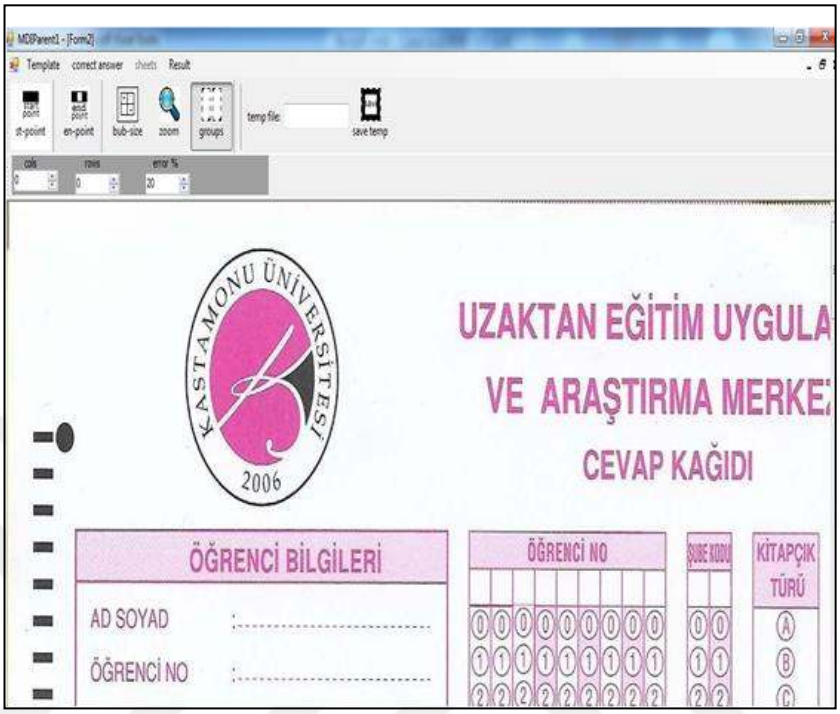
Şekil 3.19. Uygulama kullanıcı arayüzü

İkincisi Şekil 3.20'de gösterildiği gibi eski şablonu açmak için kullanılır ve sonuncusu ise menüden çıkmak için kullanılır.



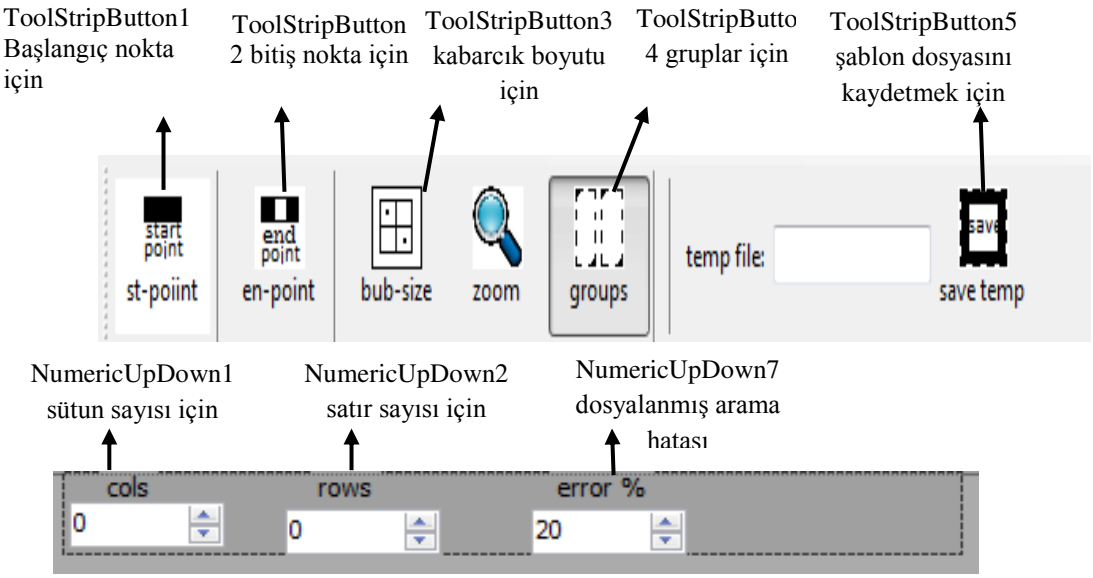
Şekil 3.20. Şablon menüsü görünümü

Yeni şablona tıkladığınızda, Şekil 3.21'te gösterildiği gibi şablon oluşturmak için pencere görünecektir.



Şekil 3.21. Şablonun koordinatlarını seçmek için kullanıcı arayüzü

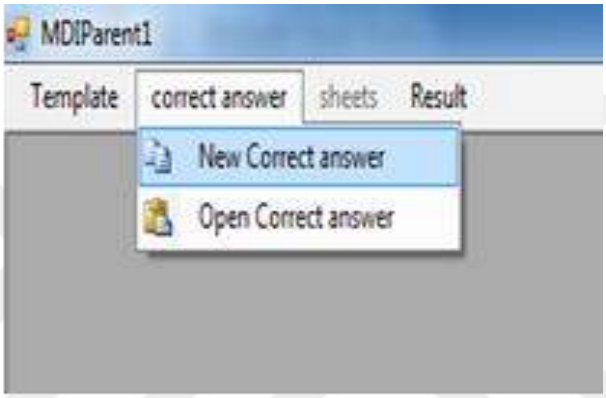
Bu tasarım, Şekil 3.22 'te gösterilen iki araç çubuğundan oluşmaktadır. Araç çubuklarının amacı, kullanıcının ne yapması gerektiğini açıklamaktır.



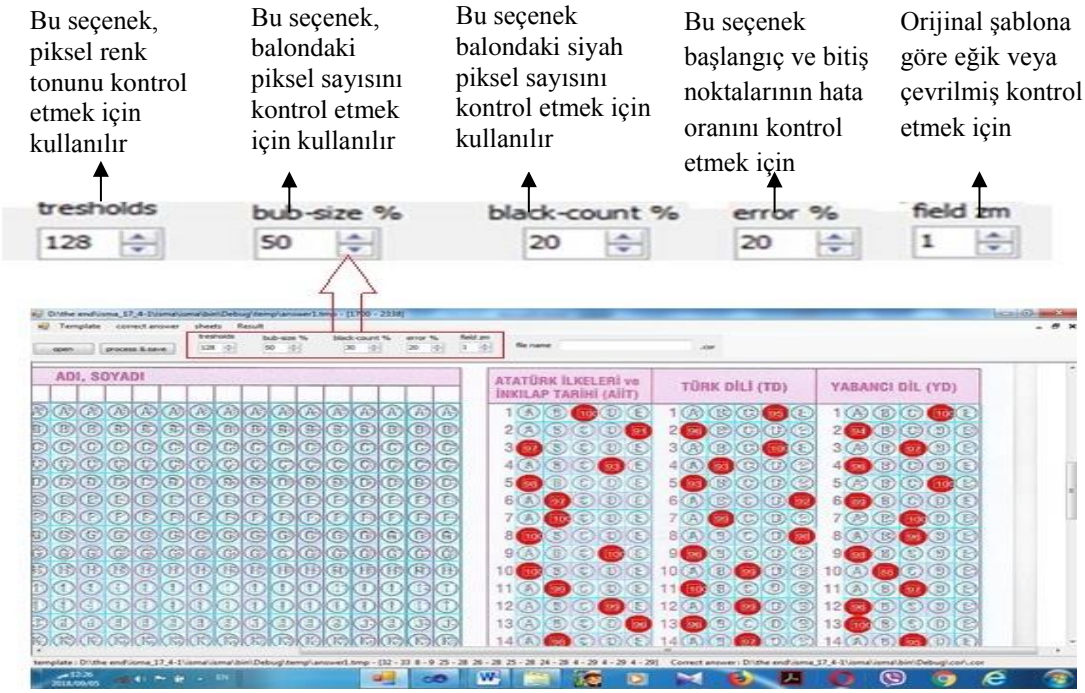
Şekil 3.22. Toolstrip1.visible bar ve toolstrip2.visible bar gösterimi

3.3.3. Doğru Cevap Tasarımı

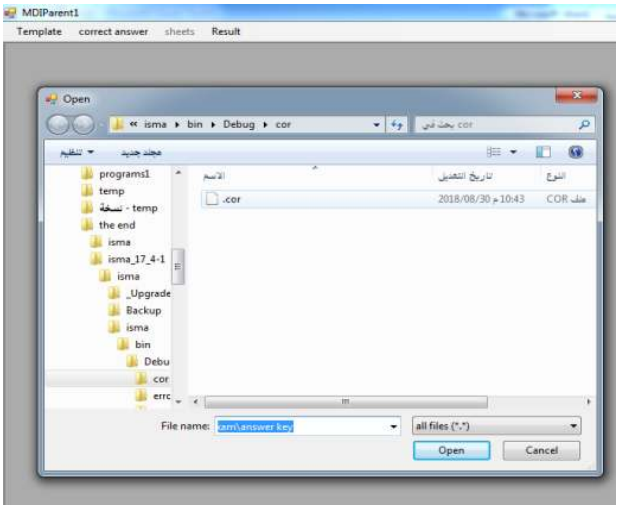
Cevabın tasarımında iki önemli alt fonksiyon bulunur: Yeni bir doğru cevap oluşturun ve eski bir cevap dosyasını Şekil 3.23'de gösterildiği gibi açın. İlki, Şekil 3.24'de gösterilen faktörlere dayanarak bunları okumak ve düzeltmek için yeni bir cevap açmak için kullanılır. İçerdiği faktörler, piksel sayısı, balon büyüklüğü vb. gibi kontroller için kullanılır. İkincisi, dosyayı açmak için kullanımı, Şekil 3.25'te gösterildiği gibi eski cevapları kullanmaktır.



Şekil 3.23. Doğru cevap menüsü görünümü



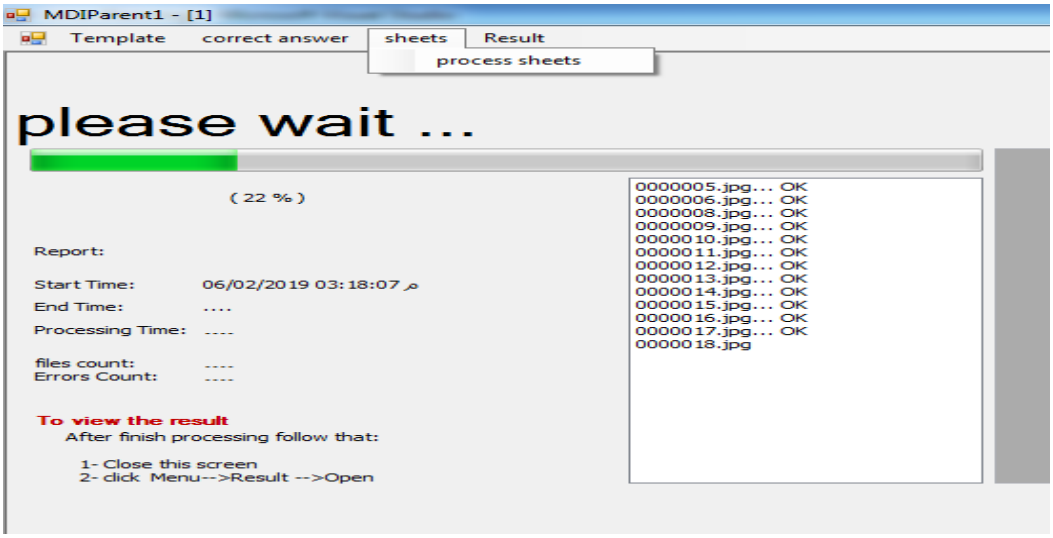
Şekil 3.24. Cevap tasarımı ara yüzü



Şekil 3.25. Yeni ve eski doğru cevap açma penceresinin açıklanması

3.3.4. Tüm Görüntü İşlemelerin Kontrol Arayüzü

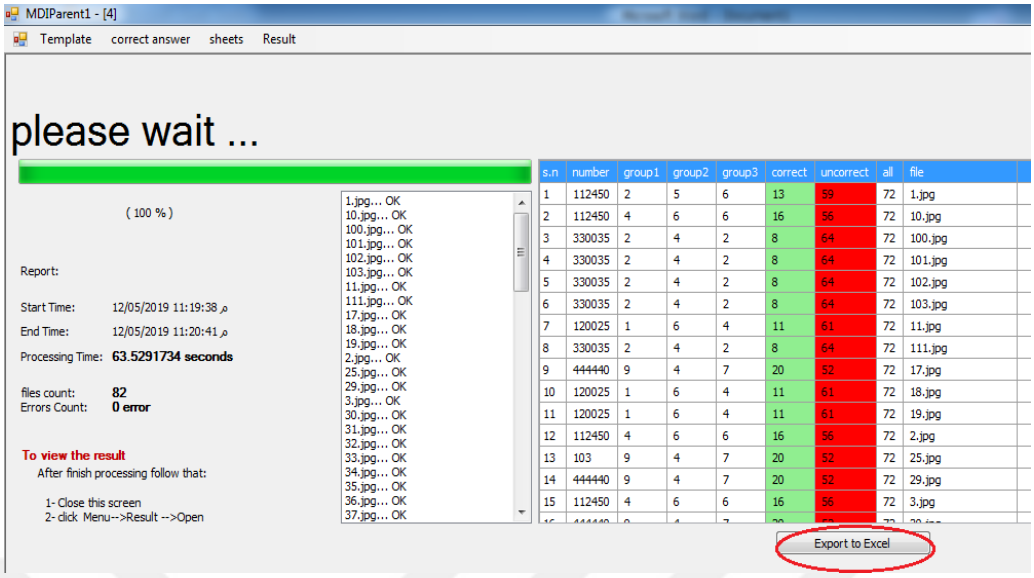
İşlem kontrolü tasarımı, görüntü dosyalarını okuma işlemini yürütmek için kullanılır ve her görüntü dosyasını, anahtar görüntü dosyasıyla çalışma klasöründe karşılaştırır. Bu tasarımı, tüm modelleri cevap anahtarıyla karşılaştırıp ve işlem süresini, dosya sayısını ve hata dosya sayısını hesaplayarak sonlandırır. Şekil 3.26'te gösterilmiştir.



Şekil 3.26. İşlem kontrol arayüzü

3.3.5. Sonuç Arayüzü Tasarımı

Bu formun işlevi, tüm sonuçları Şekil 3.27'te gösterildiği gibi bir rapor biçiminde göstermektir. Kullanıcı sonuçları Microsoft Excel Tablosuna (.xls) dönüştürebilir.



s.n	number	group1	group2	group3	correct	uncorrect	all	file
1	112450	2	5	6	13	59	72	1.jpg
2	112450	4	6	6	16	56	72	10.jpg
3	330035	2	4	2	8	64	72	100.jpg
4	330035	2	4	2	8	64	72	101.jpg
5	330035	2	4	2	8	64	72	102.jpg
6	330035	2	4	2	8	64	72	103.jpg
7	120025	1	6	4	11	61	72	11.jpg
8	330035	2	4	2	8	64	72	111.jpg
9	444440	9	4	7	20	52	72	17.jpg
10	120025	1	6	4	11	61	72	18.jpg
11	120025	1	6	4	11	61	72	19.jpg
12	112450	4	6	6	16	56	72	2.jpg
13	103	9	4	7	20	52	72	25.jpg
14	444440	9	4	7	20	52	72	29.jpg
15	112450	4	6	6	16	56	72	3.jpg

Şekil 3.27. Ekran Çıktısı

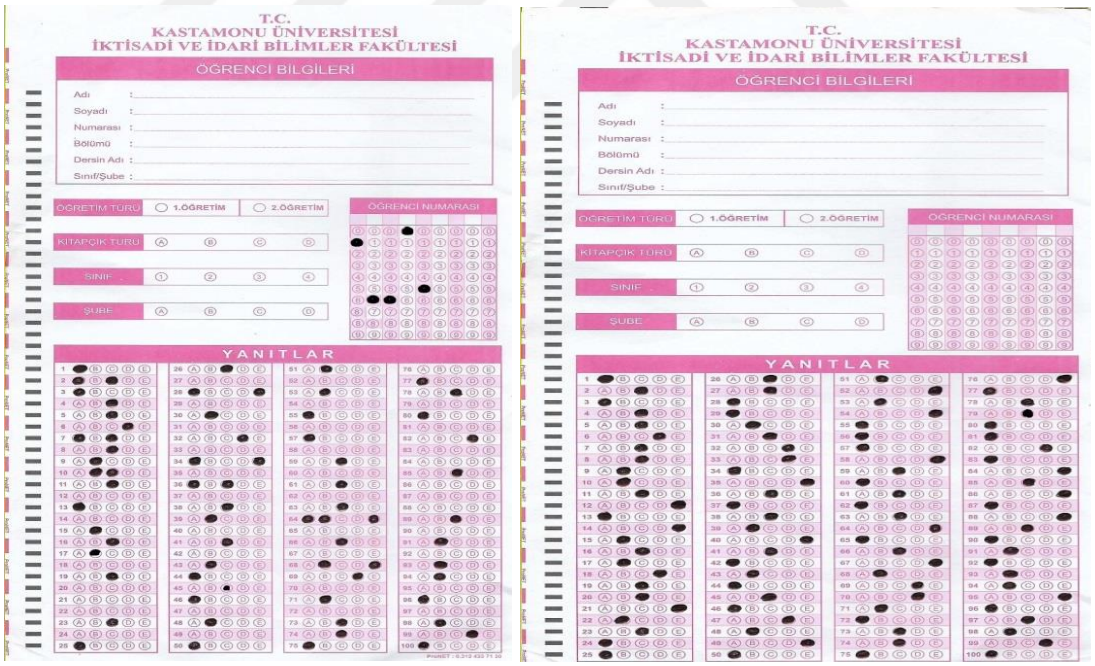
4. SONUÇ VE TARTIŞMA

4.1. Deney/Test Çalışması ve Sonuçları

Bu bölümde araştırma için yapılan deneylerin sonuçları sunulacaktır. Sonuçlar farklı renk vurgularıyla tabloda gösterilecektir. Doğru cevaplar yeşil renkle vurgulanacak ve yanlış cevaplar ise kırmızı olarak gösterilecektir. Sonuçlar bir excel sayfasında gösterilecektir.

4.1.1. Deneysel Çalışma-1

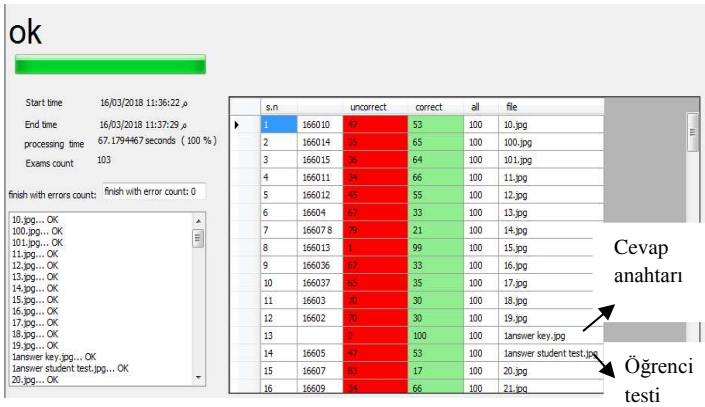
Birinci test çalışmasında, sistem 100 öğrencinin cevap kâğıtları test edilmiştir (her sınav kâğıdında 100 soru ve öğrenci numarası vardır). Bu testte kullanılan cevap anahtarı ve örnek cevap sayfası Şekil 4.1 (a) ve Şekil 4.1 (b) 'de gösterilmiştir.



(a) (b)

Şekil 4.1. (a) 100 Cevap ve "16605" öğrenci kimliğini içeren cevap anahtarı ve 53 doğru cevaplı (b) örnek cevap sayfasını içeren (a) cevap anahtarı ile Tip I test edilmesi

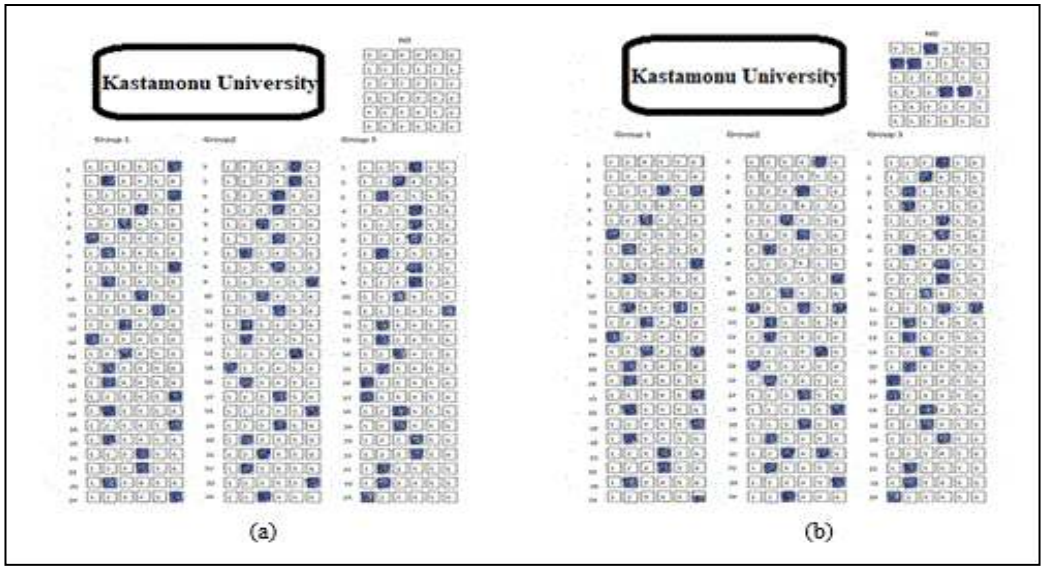
Örnek öğrenciye ait cevap sayfası ile cevap anahtarının karşılaştırılması sonucunda elde edilen sonuçlar, Şekil 4.2'de gösterilmiştir.



Şekil 4.2. 100 Test için tip I örnek sonuç listesi

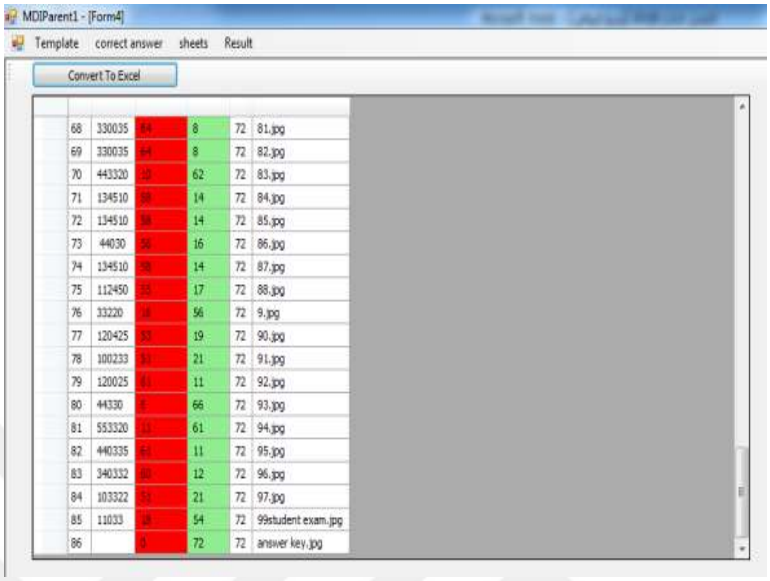
4.1.2. Deneysel Çalışma-2

Bu test çalışmasında, sistem her sınav kâğıdının 78 sorudan oluşan 100 öğrencinin sınav kâğıdına cevap kâğıdına girdiği tür II testine tabi tutulmuştur ve bazı sorular, Şekil. 4.3 (a) cevap anahtarı ve Şekil 4.3 (b) ise 57 doğru cevap içeren öğrenci testinden oluşan Tip II'nin testi gösterilmiştir. Sonuçlar, Şekil 4.3'te gösterildiği gibi rapor olarak Excel formatında sağlanmıştır.



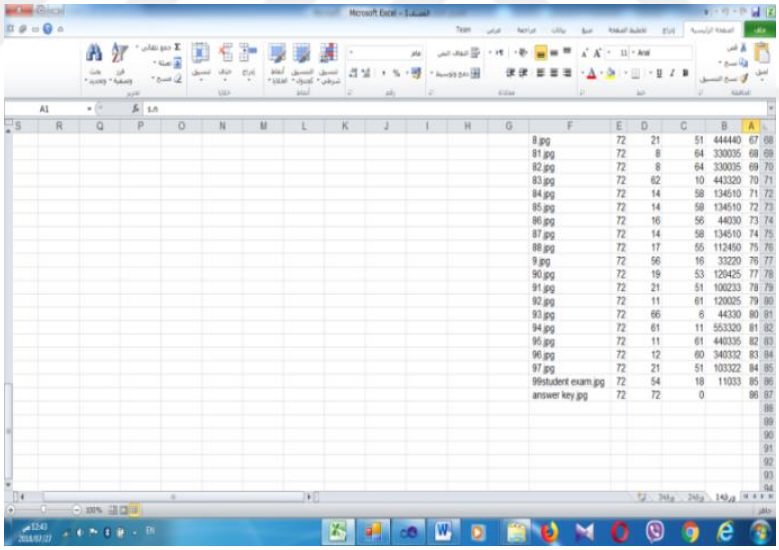
Şekil 4.3. 72 Cevap içeren (a) cevap anahtarı ve (b) 57 doğru cevap içeren öğrenci testinden oluşan Tip II'nin test edilmesi

Sonuçlar, Şekil 4.4'te gösterilmiş ve sonuçları Şekil 4.5'te gösterildiği gibi Excel sayfasına aktarılmıştır.



68	330035	64	8	72	81.jpg
69	330035	64	8	72	82.jpg
70	443320	10	62	72	83.jpg
71	134510	58	14	72	84.jpg
72	134510	58	14	72	85.jpg
73	44030	56	16	72	86.jpg
74	134510	58	14	72	87.jpg
75	112450	55	17	72	88.jpg
76	33220	16	56	72	9.jpg
77	120425	53	19	72	90.jpg
78	100233	51	21	72	91.jpg
79	120025	61	11	72	92.jpg
80	44330	6	66	72	93.jpg
81	553320	11	61	72	94.jpg
82	440335	61	11	72	95.jpg
83	340332	60	12	72	96.jpg
84	103322	51	21	72	97.jpg
85	11033	18	54	72	99student exam.jpg
86		0	72	72	answer key.jpg

Şekil 4.4. 86 Test için Tip II örnek sonuçlar



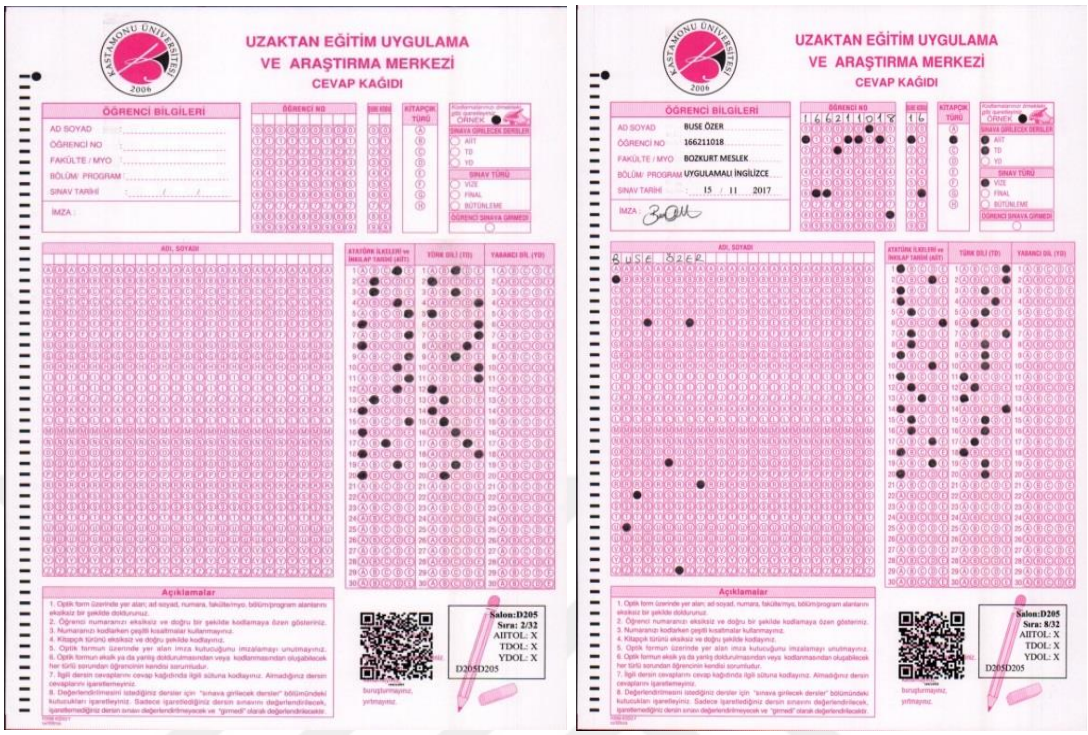
8.jpg	72	21	51	444440	67
81.jpg	72	8	64	330035	68
82.jpg	72	8	64	330035	69
83.jpg	72	62	10	443320	70
84.jpg	72	14	58	134510	71
85.jpg	72	14	58	134510	72
86.jpg	72	16	56	44030	73
87.jpg	72	14	58	134510	74
88.jpg	72	17	55	112450	75
9.jpg	72	56	16	33220	76
90.jpg	72	19	53	120425	77
91.jpg	72	21	51	100233	78
92.jpg	72	11	61	120025	79
93.jpg	72	66	6	44330	80
94.jpg	72	61	11	553320	81
95.jpg	72	11	61	440335	82
96.jpg	72	12	60	340332	83
97.jpg	72	21	51	103322	84
99student exam.jpg	72	54	18	11033	85
answer key.jpg	72	72	0		86

Şekil 4.5. Excel formunda nihai sonuçlar

4.1.3. Deneysel Çalışma- 3

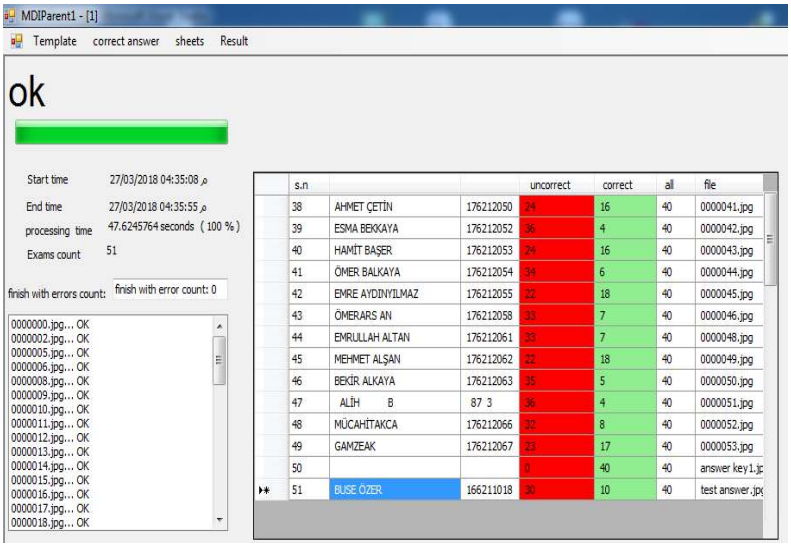
Bu testte, Şekil 4.6 (a) ve Şekil 4.6 (b) 'de gösterildiği gibi 4 önemli bölge (öğrenci numarası, öğrenci adı, cevaplar ve barkod bölgesi) içeren Tip III test edilmiştir.

50 öğrencinin cevap kâğıtlarının sonuçları Şekil 4.7'de verilmiştir.



(a) (b)

Şekil 4.6. (a) 40 cevap anahtarı ve (b) 22 doğru cevap içeren, öğrenci adı "BUSE" ve öğrenci kimliği "166211018" olan öğrenci kağıdı ile Tip III test edilmesi



Şekil 4.7.86 Test için Tip III örnek sonuçlar

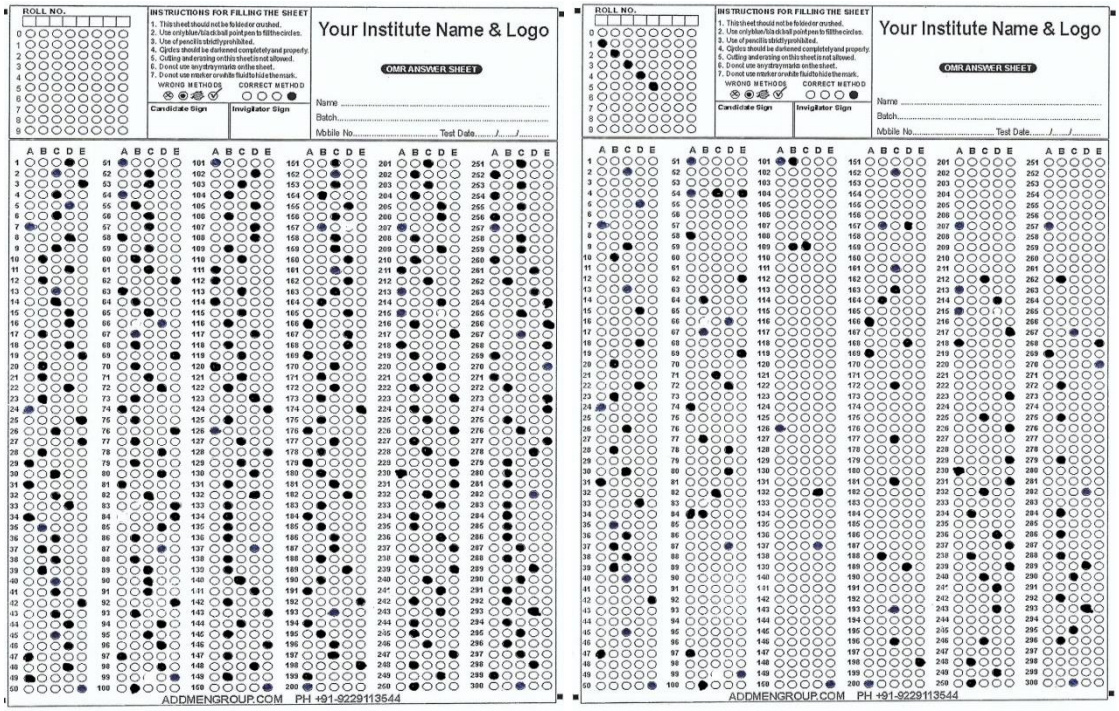
Şekil 4.8'de sonuçlar excel formunda sunulmuştur.

K	J	I	H	G	F	E	D	C	B	A
file	all	uncorrect	correct	barcode	group2	group1	name	sub	ogrenci no	s.n
0000005.jpg	40	25	15	166211015	10	5	MEHTAP POLAT	16	166211015	1
0000006.jpg	40	35	5	166211016	1	4	MUSTAFAÖZTEKKELİ	16	166211016	2
0000008.jpg	40	26	14	166211020	9	5	MELEK ÖZALP	16	166211020	3
0000009.jpg	40	36	4	166211024	0	4	ESRA KOYUNLU	16	166211024	4
0000010.jpg	40	22	18	166211026	8	10	SEMRA KOÇ	16	166211026	5
0000011.jpg	40	34	6	166211032	2	4	BERK HARMANDAR	12	166211032	6
0000012.jpg	40	20	20	166211034	11	9	SİNEM GÜLMEZ	16	166211034	7
0000013.jpg	40	24	16	166211038	8	8	ÇİĞDEM GÖK	16	166211038	8
0000014.jpg	40	35	5	166211042	2	3	FATMA EFE	16	166211042	9
0000015.jpg	40	21	19	166211044	10	9	ZÜBEYDE DİKYO L	16	166211044	10
0000016.jpg	40	35	5	166211045	0	5	MERYEM DEMİRKIRAN	16	166211045	11
0000017.jpg	40	17	23	166211048	12	11	MUHAMMED ÇELİK	16	166211048	12
00000171.jpg	40	18	22	166211048	12	10	MUHAMMED ÇELİK	16	166211048	13
0000018.jpg	40	31	9	166211049	4	5	ENGİN ÇAKIR	16	166211049	14
0000019.jpg	40	24	16	166211052	10	6	SELİNAY BOZKURT	16	166211052	15
0000020.jpg	40	34	6	166211053	1	5	EMİNE BİLGİÇ	16	166211053	16
0000021.jpg	40	30	10	166211059	8	2	KÜBRAARSLAN	16	166211059	17

Şekil 4.8. Excel sayfasında nihai sonuçlar

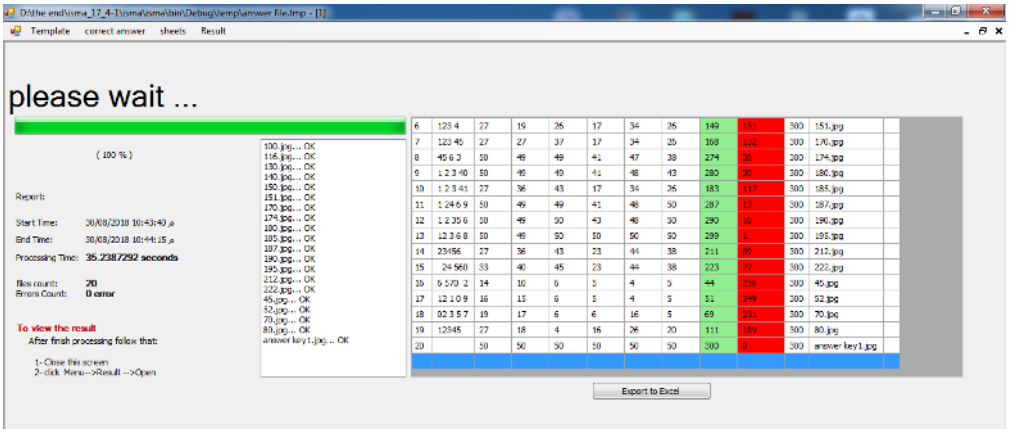
4.1.4. Deneysel Çalışma-4

Sistem, öğrencinin cevap kâğıtları ve her sınav kâğıdında 300 soru ve öğrenci numarası içerecek şekilde V Tipi için test edilmiştir. Bazı sorular Şekil 4.9 (a) ve 4.9 (b) 'de gösterildiği gibi cevapsız veya birden fazla cevaba sahiptir. Test sonuçları Şekil 4.10'da sunulmuştur.



(a) (b)

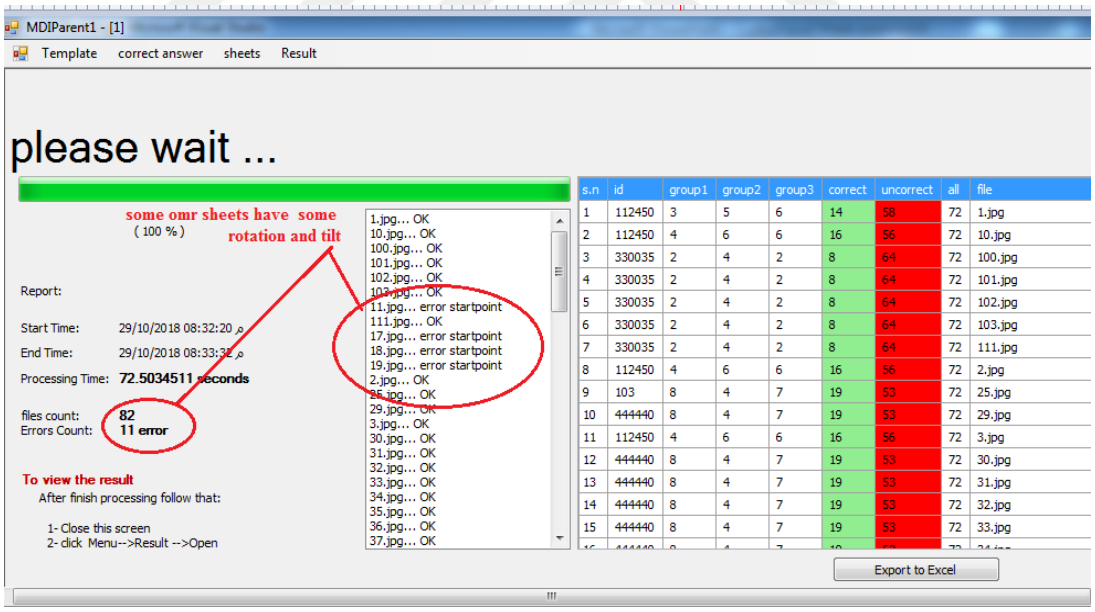
Şekil 4.9. 300 Soru içeren (a) cevap anahtarı ve (b) 111 doğru cevap içeren öğrenci testiyle tip v'nin test edilmesi



Şekil 4.10. 20 Test için tip V örnek sonuçları

4.1.5. Deneysel Çalışma-5

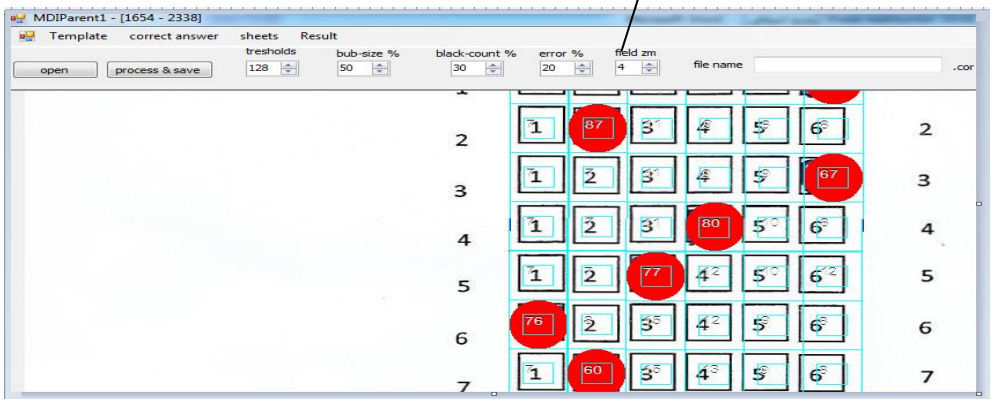
Formun taranan görüntüsü yatırıldığında döndürme işlemi yapılması gerekir. Şekil 4.11'de, tasarlanan sistem 11 adet hatalı dosya ile birlikte 82 dosyayı test etmiştir.



Şekil 4.11. 11 adet hatalı dosya ile birlikte 82 dosyayı test işleminde gereken işlem süresi (CPU süresi)

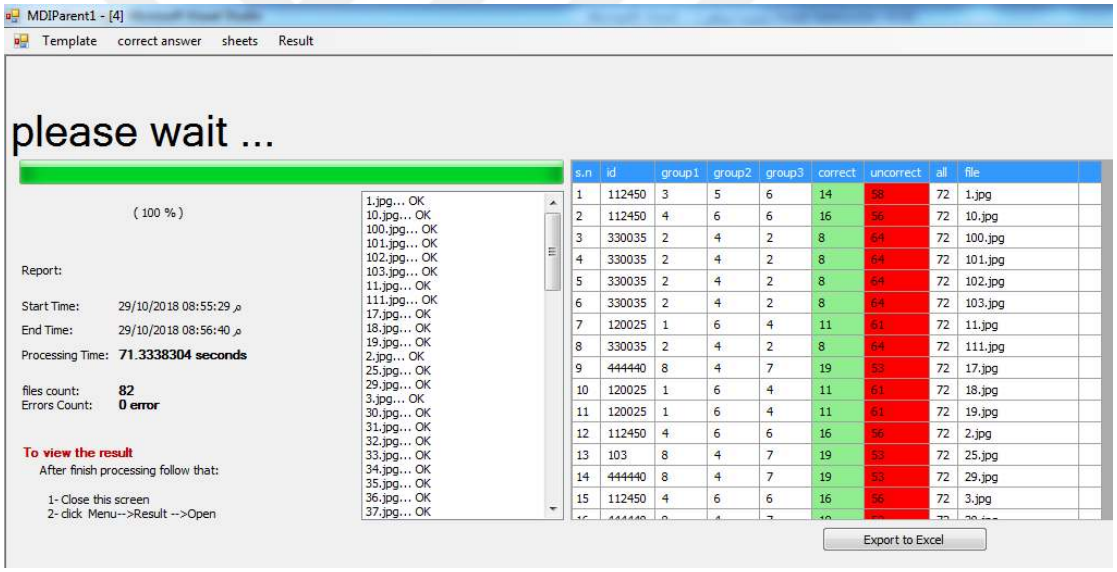
Bu sorunu çözmek için dosyalanmış aramayı Şekil 4.12'de gösterildiği gibi arttırılmalıdır.

Dosyalanmış arama için artırım faktörü



Şekil 4.12. Dosyalanmış arama için artırım miktarındaki değişim gösterimi

Şekil 4.13'te sistem 82 hata testi dosyası olmadan test edilmiştir.



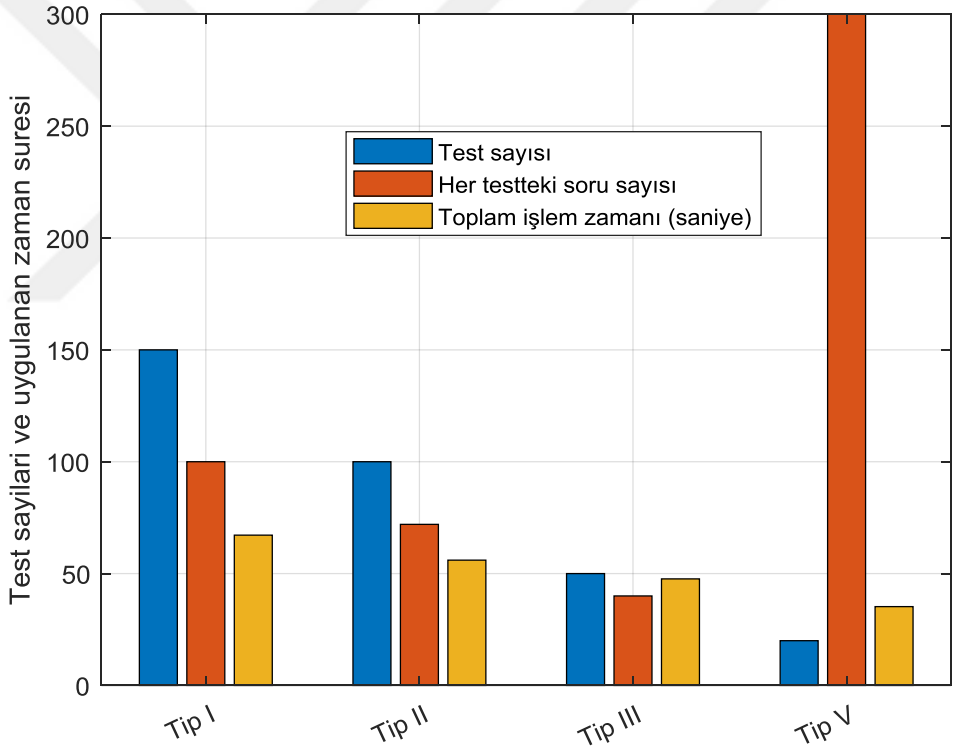
Şekil 4.13 Dhatalı dosya olmaksızın 82 dosyayı test işleminde gereken işlem süresi (CPU süresi).

Bu deneylerde işlem süresi, Tablo 4.1'de gösterildiği gibi soru sayısı arttığında form tipinin artmasına bağlıdır.

Tablo 4.1. *Tüm OİT Kağıtları İçin (İşlemci zamanı)*

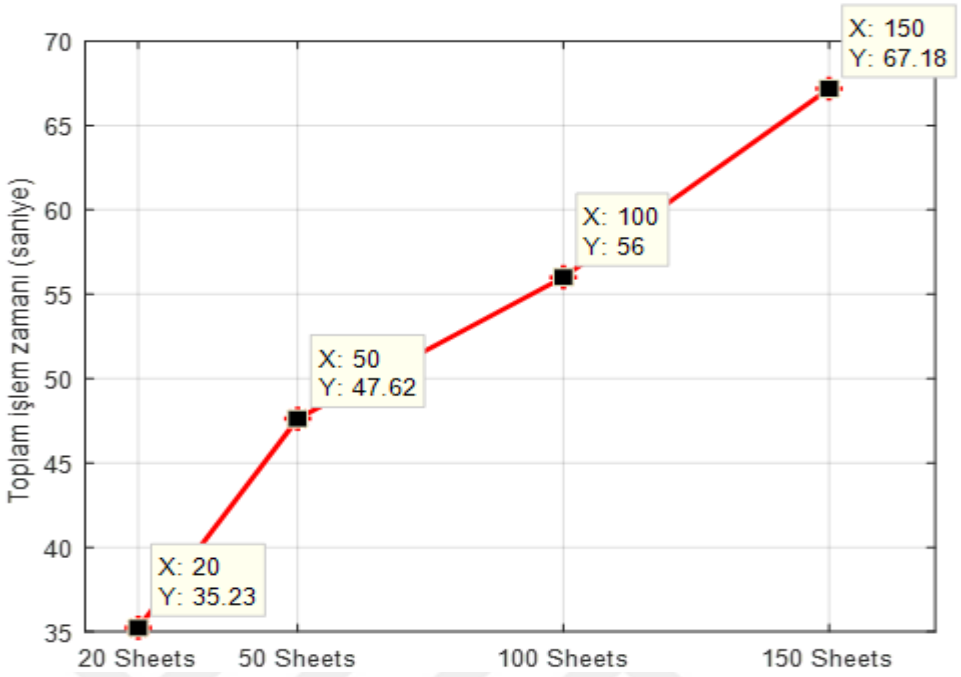
Cevap anahtarı tipi	Test sayısı	Her testteki soru sayısı	Her test için işlem zamanı (saniye)	Toplam işlem zamanı (saniye)
Tip I	150	100	0.67179	67,179
Tip II	100	72	0.56002	56,002
Tip III	50	40	0.933	47,624
Tip V	20	300	1.7615	35,23

Bu sonuçlar görsel olarak şekil 4.14 de gösterilmektedir.



Şekil 4.14. İşlemci zamanı sonuçların istatistiksel gösterimi

Test sayısına göre işlem zamanı şekil 4.15te gösterilmektedir.



Şekil 4.15. Test sayısına göre işlem zamanı

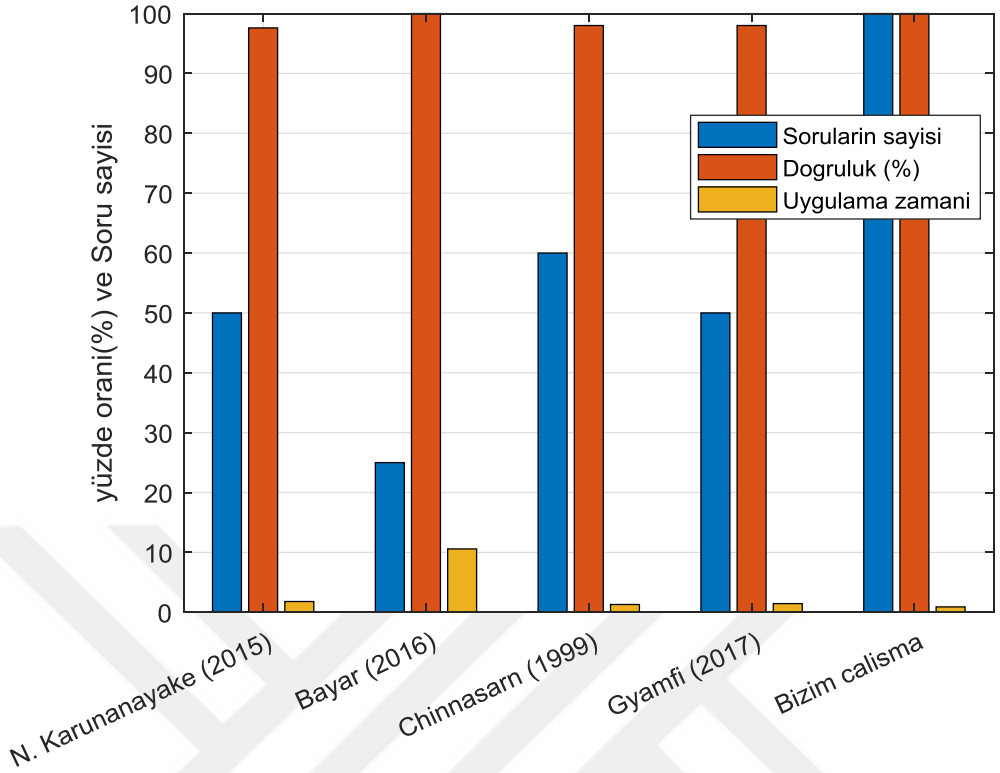
4.2. Diğer Çalışmalarla Karşılaştırma

Bu bölümde, sonuçlar ikinci bölümde sunulan bir önceki çalışma ile karşılaştırılacaktır. Bu karşılaştırmada, doğruluk sonucuna ve işlem süresine odaklanılmıştır. Tablo 4.2'de bulunan karşılaştırmalar bazı çalışmaların doğruluğunu göstermektedir.

Tablo 4.2. *Literatürdeki çalışmalar ile tez çalışması sonuçlarının karşılaştırılması*

Önceki Çalışmalar	**Yöntem**	**Grupların Tespiti**	**Her Testin Soru Sayısı**	**Her Test İçin İşlem Süresi**	**Doğruluk (%)**
G. Bayar (2016)	Sistem, görüntüye yerleştirilen nesneleri ve şekilleri tanımlamak için kullanılabilecek Hough dönüşüm tekniğini kullanmıştır.	Öğrencilerin numaraları ve çoktan seçmeli sorular	25	10.583 sn.	100%
Nalan. Karunanayak (2015)	Sistem, seçilen tüm bölgeleri doğru cevaplarla seçti, baskı kağıdına elle işaretledi	Öğrencilerin numaraları ve çoktan seçmeli sorular	40	2 saniyeden az	97.6%
E.Gyamfi ve Y.Missah (2017)	Denetimsiz sınıflandırmaya dayalı tüm kabarcıklar için merkez piksel değerlerini saptamak için farklı bir teknik önermişlerdir.	Öğrencilerin numaraları ve çoktan seçmeli sorular	50	1.45 sn	98%
Chinnasarn ve Rangsanseri (1999)	Önerilen teknik öğrenme moduna ve tanıma moduna bağlıdır.	Öğrencilerin numaraları ve çoktan seçmeli sorular	60	1.30 ve 1.19 saniye	Yüksek Başarı
Önerilen Yöntem (2019)	Önerilen teknik, şablon görüntüsünün oluşturulması ve kilit noktaların tespiti üzerine kuruludur	Öğrencilerin numaraları çoktan seçmeli sorular ve barkod	40	1 saniyeden az	100%

Bu sonuçlar görsel olarak şekil 4.16'de gösterilmektedir.



Şekil 4.16. Sonuçların istatistiksel gösterimi

5. SONUÇLAR VE GELECEK ÇALIŞMALAR

Optik işaret tanımanın en yaygın kullanımı, öğrencinin çoktan seçmeli bir sınava verilen yanıtları veya bir anket ya da geri bildirim formuna verilen yanıtları işlemektir. Tipik olarak sorular kâğıt üzerinde verilmektedir ve öğrenciler cevaplarını özel önceden basılmış formlara işaretlemektedirler. Bu formlar daha sonra otomatik olarak okunur (Nerkar, 2015).

Bu çalışma, görüntü işleme yöntemi kullanarak çoktan seçmeli testler için bir OİT sistemi geliştirilmesine odaklanmıştır. Çok sayıda formun okuma ve işleme için bir çözüm sağlayan algoritmanın değerlendirilmesi için Visual Studio entegre geliştirme ortamında (IE) Visual Basic (VB) dili kullanılmıştır. Sistem, çoktan seçmeli testlerinin puanlamasını otomatik olarak kontrol etmekte ve sonuçta yüksek doğrulukla bunu başarmaktadır. Sistemimizde kullanılan metodoloji, şablon cevap kâğıdı oluşturulmasına ve kilit nokta tespitine (başlangıç noktası ve bitiş noktası) dayanmaktadır.

Sonuçlar çoktan seçmeli cevap kâğıtlarının sınav puanının hesaplanması için kullanılabilir. Sistem, kolay kullanıcı ara yüzü ile tasarlanmış ve minimum maliyetle uygulanmıştır. Ayrıca değerlendiricinin çoktan seçmeli cevap anahtarının doğruluğunu ve verimliliğinin yakalaması için yardımcı olacaktır. Sistem başka farklı türde cevap kâğıtlarına da uygulanabilir. Her bir form için işlem süresi bir saniyeden az sürmektedir. Kısacası, bu sistem 10 dakikada 500 sayfa okuyabilir.

Bu tez, karmaşık görüntü işleme teknikleri üzerinde optik işaret tanıma tekniği için etkili bir alternatif yöntem sunmuştur. Önerilen sistem OİT verilerini daha hızlı ve doğru bir şekilde değerlendirmek için kullanışlıdır. Bu sistem, optik işaret tanıma tesislerinin kırsal alanlardaki okullar gibi kolay bulunamadığı bölgelerde kullanışlıdır. Tez çalışmasında OİT, barkod ve görüntü işleme teknikleri alanlarında araştırma yapmak için bazı önemli sonuçlar ortaya konulmuştur.

Çalışmanın bir sonraki adımında yapılacak çalışmalar aşağıdaki şekilde sıralanabilir:

- Öğrencilerin online olarak formları doldurmasıyla online sonuç değerlendirmesi yapılabilen ve öğrencilerin sonuçlara bu sitedeki hesaplama yoluyla ulaşabilecekleri bir web sitesi oluşturulabilir.
- Bu uygulamayı, oylama, topluluk anketleri vb. diğer alanlarda da kullanılabilir.
- Tarayıcı yerine web kamera veya telefon kullanarak OİT kâğıt değerlendirilebilir.
- Zamanı kısaltmak adına OCR teknolojisini, öğrenci kimliğini ve öğrenci adını tanımlama amacıyla da kullanılabilir.

KAYNAKLAR

Al-marakeby, A. (2013). Multi-Core Processors for Camera based OMR. *International Journal of Computer Applications*, *68*(13), 1-5.

Arica, N. ve Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *31*(2), 216-233.

Bayar, G. (2016). The use of hough transform to develop an intelligent grading system for the multiple choice exam papers. *Karaelmas Fen ve Mühendislik Dergisi, 6(1), 100-104.*

Chinnasarn, K. ve Rangsanseri, Y. (1999). Image-processing-oriented optical mark reader. *In Applications of digital image processing* 3808, 207-209.

Gonzalez, R. C., Woods, R. E. ve Eddins, S. L. (2004). *Digital Image Processing Using MATLAB*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Gyamfi, E. O. ve Missah, Y. M. (2017). Pixel-Based Unsupervised Classification Approach for Information Detection on Optical Markup Recognition Sheet. *Advances in Science, Technology and Engineering Systems Journal*, *2*(4), 121-132.

Hasan, R. H. ve Kareem, E. A. (2015). An Image Processing Oriented Optical Mark Reader Based on Modify Multi-Connect Architecture MMCA. *International Journal of Modern Trends in Engineering and Research (IJMTER)*, *2*(7).

Karunanayake, N. (2015). OMR sheet evaluation by web camera using template matching approach. *International Journal for Research in Emerging Science and Technology*, *2*(8).

Kaur, N. ve Kaur, R. (2011). "A review on various methods of image thresholding" *International Journal on Computer Science and Engineering , 3(11).*

Kumbhar, G. P. ve Holambe, N. S. (2015). A Review of Image Thresholding Techniques", *International Journal of Advanced Research in Computer Science and Software Engineering, 5(6).*

Krishna, G., Rana, H. M., Madan, I. ve Sahu, R. (2013). Implementation of OMR technology with the help of ordinary scanner. *International Journal of Emerging Technology and Advanced engineering*, *3*(4), 714-719.

Koirala, P. (2007). RGB Color Space.*Department of Computer Science and Statistics.Brigham University of Joensuu*.

Kanan, C. ve Cottrell, G.W. (2012). Color-to-grayscale: Does the method matter in image recognition?, PLoS ONE *7*(1).

Lais, S. (2002). QuickStudy: Optical Character Recognition, ComputerWorld Magazine.

Lomte, V.M., Lomte, R.M., Mastud, D. ve Thite, S. (2012). Robust Barcode Recognition Using Template Matching. *International Journal of Engineering Sciences & Emerging Technologies*, *2*(2), 59-65.

Nerkar, B. (2015). "Optical Markup Recognition for Exam system," *International Journal of Emerging Technology and Advanced engineering, 5(2).*

Plamondon, R. ve Srihari, S. N. (2000). Online and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(1), 63-84.

Puglia, S. (2000). "Technical Primer," in *Handbook for Digital Projects: A Management Tool for Preservation and Access*, ed. Maxine K. Sitts (Andover, MA: Northeast Document Conservation Center, 2000), 84.

Patil, A., Naik, M. ve Ghare, P. H. (2016). FPGA implementation for OMR Answer sheet scanning and IR Sensors,*International Journal of Electrical, Electronics and Data Communication*, 4(11).

Rakesh, S., Atal, K. ve Arora, A. (2013). Cost effective optical mark reader. *International Journal of Computer Science and Artificial Intelligence*, *3*(2), 44.

Singh, T. R., Roy, S., Singh, O. I., Sinam, T. ve Singh, K. (2011). A new local adaptive thresholding technique in binarization, *International Journal of Computer Science*, 8(6), 271-277.

Sumitra, M. ve Gaikwad, B. (2015). Image processing based OMR sheet scanning. *Int J Adv Res Electron Commun Eng*, *4*(3), 519-522.

Sattayakawee, N. (2013). Test scoring for non-optical grid answer sheet based on projection profile method. *International Journal of Information and Education Technology*, *3*(2), 273-277.

Saengtongsrikamon, C., Meesad, P. ve Sodsee, S. (2009). Scanner-based optical mark recognition, *Information Technology Journal*, *5*(1), 69-73.

Firdousi, R. ve Parveen, S. (2014). "Local Thresholding Techniques in Image Binarization", *International Journal Of Engineering And Computer, 3(3), 4062-4065*.

Sachs, J. (2003). Digital Image Basics**,** 08.05.2018 tarihinde http://vision.ucsd.edu/~nalldrin/files/digital_image_basics.pdf adresinden alınmıştır.

Talib, A. B., Ahamad, N. B. ve Tahar, W. (2015). OMR Form Inspection by Web Camera using Shape Based Matching Approach, *International Journal of Research in Engineering and Science*, 3(4), 29-35.

Zampirolli, A. F., Gonzalez, Q. J. ve Oliveira Neves, R. P. (2013). Automatic correction of multiple-choice tests using digital cameras and image processing, Anais WVC 2013.

URL_1 Online Available: Reg, T. (2013). Bubble Sheet Testing in Education's**.**

URL_2 Online Available: Pratt, W. (2007). Color-to-Grayscale: Does the Methodhttps://journals.plos.org/plosone/article?id=10.1371/journal.

URL_3 Online Available: Jack ,K. (2007). Color-to-Grayscale: Does the Method, https://journals.plos.org/plosone/article?id=10.1371/journal.

URL_4 Online Available: Caroline. (2010). The 5 Types of Digital Image Files: TIFF, JPEG, GIF, PNG athttps://www.ivanexpert.com/blog.

URL_5 Online Available: Bryan S. Morse.(1998–2000). Lecture 4: Thresholding at https:*homepages.inf.ed.ac.uk/rbf/CVonline/.../MORSE/threshold.pdf.*

URL_6 Online Available: Parul, M ., Monga, H.ve Kaur, M. (2012). A novel Optical Mark Recognition Technique Based on Biogeography Based Optimization, *nternational Journal of Information Technology and Knowledge Management, 5(2), 331-333.*

EKLER

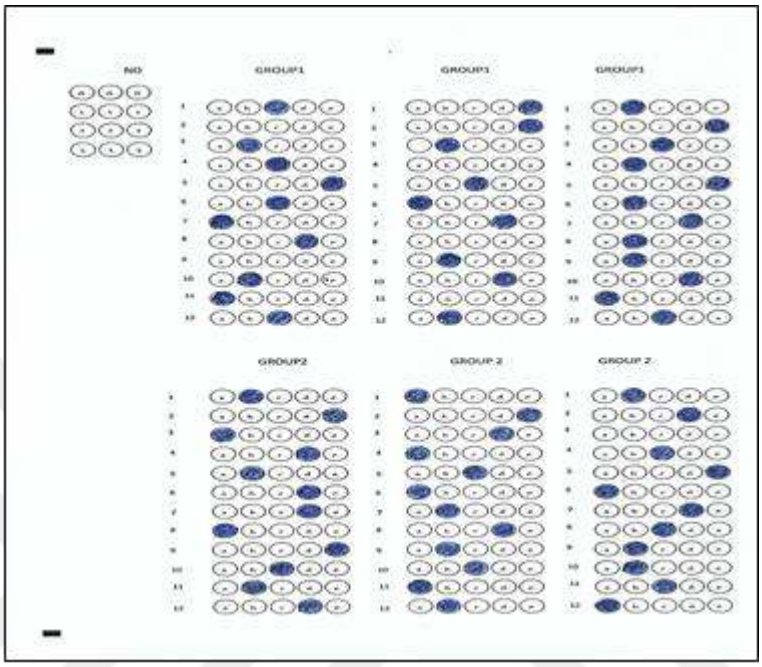
EK 1 **Çalışmada kullanılan diğer tip cevap kağıtları**

EK 2 **OİT kılavuzu**

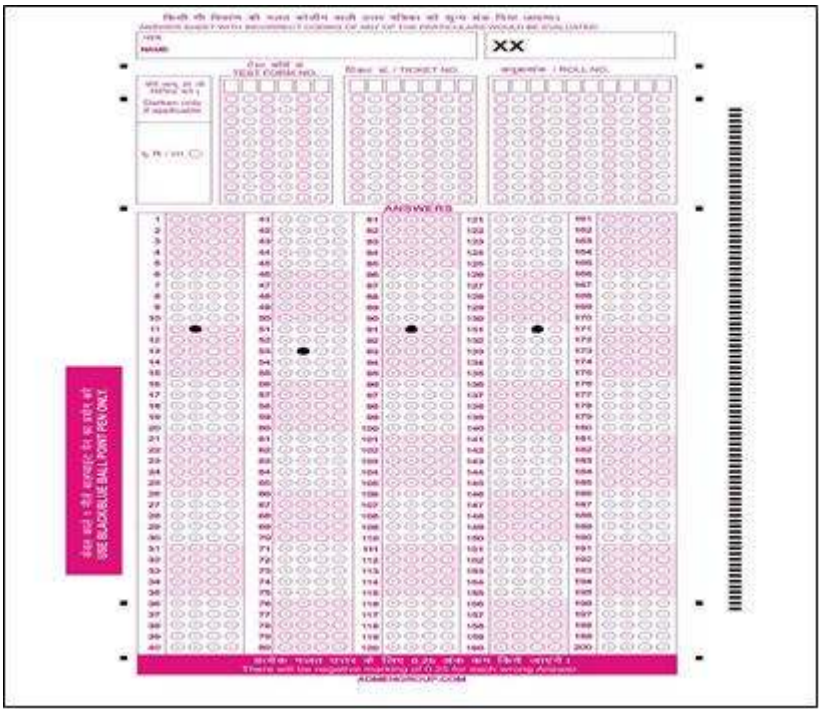
EK 3 **Visual basic kodu**

EK 1 Çalışmada kullanılan diğer tip cevap kağıtları

TİP I

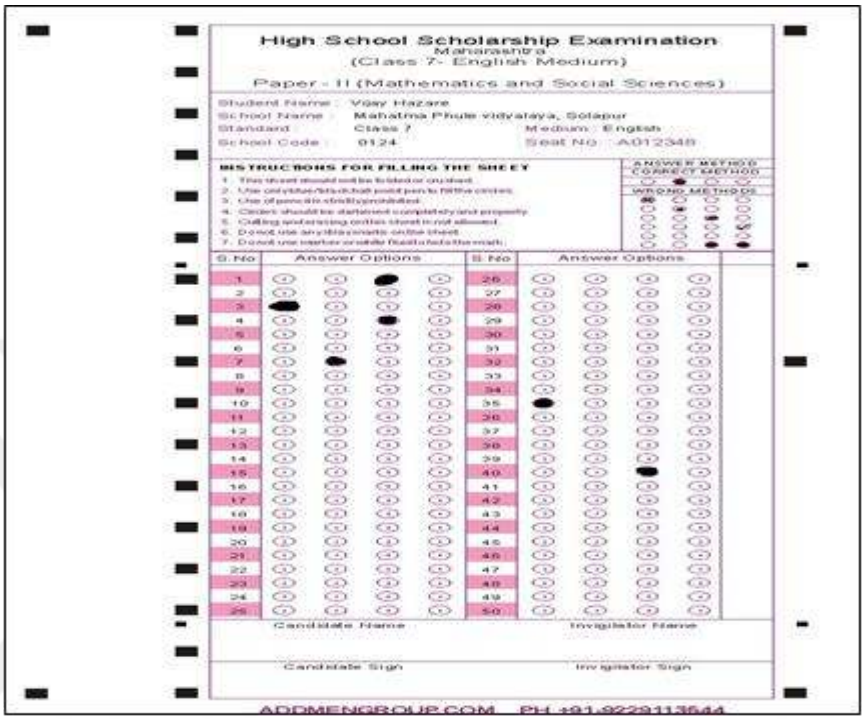


TİP II

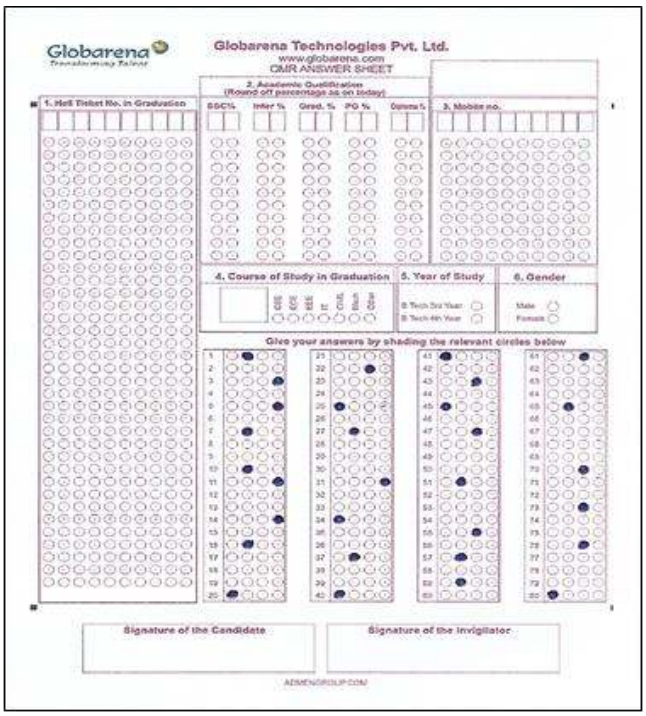


EK 1' ün devamı

TİP III

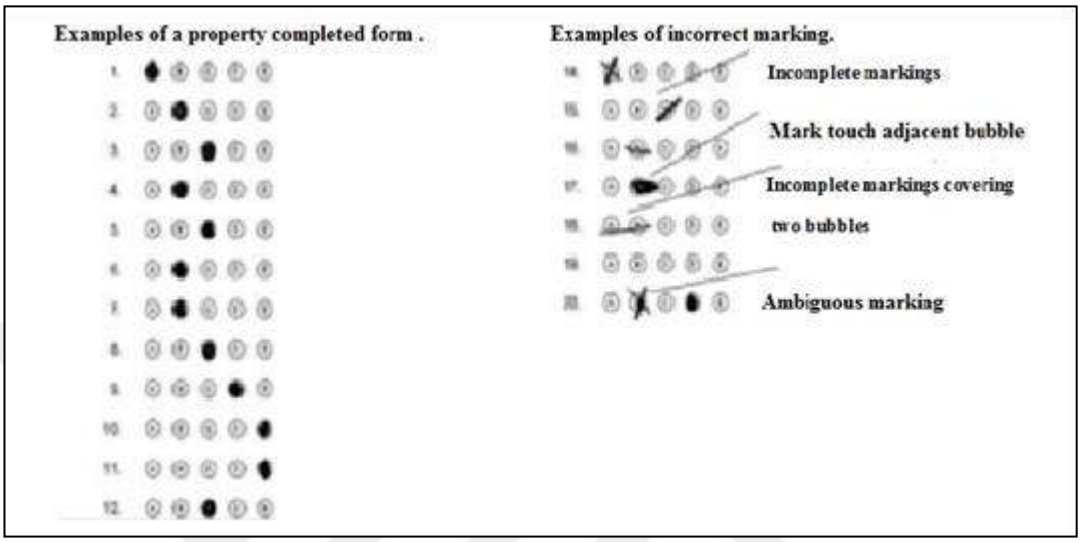


TİP V



EK 2 Oit kılavuzu

Doğru OİT İŞARETLEME ŞEKLİ Doğru OİT İŞARETLEME ŞEKLİ



EK 3 Visual basic kodu

```
Imports System.Windows.Forms
Imports System.IO

Public Class MDIParent1
    Public main_zm = 1
    Public main_err_per As Integer = 20
    Public cellwper As Single = 0.5
    Public cellw As Integer
    Public sholds = 128
    Public main_darkcount As Integer = 30
    Public temp, cor As String
    Public main_tmp As New temp
    Public grp As temp.grouptype
    Public main_cor() As String
    Dim corfile As IO.StreamReader
    Dim ss() As String
    Dim rr() As String
    Dim tt() As String
    Public org, cpy, tmp As Bitmap


    Private Sub ShowNewForm(ByVal sender As Object, ByVal e As
EventArgs) Handles NewToolStripMenuItem.Click
        OpenFileDialog1.InitialDirectory =
IO.Directory.GetCurrentDirectory & "\images"
        OpenFileDialog1.Filter = "all files (*.*)|*.*"
        If (OpenFileDialog1.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then
            Panel1.Visible = True
            ToolStrip2.Visible = True
            ToolStrip1.Visible = True
            ToolStripButton1.Checked = True
            ToolStripButton2.Checked = False
            ToolStripButton4.Checked = False
            ToolStripButton6.Checked = False
            ToolStripButton8.Visible = False
            ToolStripComboBox2.Visible = False
            ToolStripLabel1.Visible = False
            ToolStripLabel2.Visible = False
            ToolStripComboBox1.Visible = False
            ToolStripTextBox1.Visible = False
            StatusStrip.Items(0).Text = ">>Start point"
            ToolStripComboBox1.SelectedIndex = 0
            org = Image.FromFile(OpenFileDialog1.FileName)
            cpy = Image.FromFile(OpenFileDialog1.FileName)
            tmp = Image.FromFile(OpenFileDialog1.FileName)
            Form2.PictureBox1.Image = org
            Form2.MdiParent = Me
```

EK 3' ün devamı

```
Me.SheetsMenu.Enabled = False
            Form2.Show()
            Form2.WindowState = FormWindowState.Maximized
        End If

    End Sub

    Private Sub OpenFile(ByVal sender As Object, ByVal e As
EventArgs) Handles OpenToolStripMenuItem.Click
        OpenFileDialog1.InitialDirectory =
IO.Directory.GetCurrentDirectory & "\temp"
        OpenFileDialog1.Filter = "all files (*.*)|*.*"
        If (OpenFileDialog1.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then

            temp = OpenFileDialog1.FileName
            Me.SheetsMenu.Enabled = False
            Me.correctAnswerMenu.DropDownItems(0).Enabled = True
            ToolStripStatusLabel.Text = "template : " & temp & "
Correct answer : "
        End If

    End Sub


Private Sub ExitToolsStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs) Handles ExitToolStripMenuItem.Click
        Me.Close()
    End Sub

Private Sub CopyToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CopyToolStripMenuItem.Click
        'Form3.MdiParent = Me
        'Form3.Show()
        TextBox1.Text = ""
        Button2.Enabled = False
        Panel2.Visible = True
        ToolStrip3.Visible = True
        NumericUpDown3.Value = 128
        NumericUpDown4.Value = 50
        NumericUpDown5.Value = 30
        NumericUpDown6.Value = 20

    End Sub
    Private Sub ProcessSheetsToolStripMenuItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
ProcessSheetsToolStripMenuItem.Click
        Form1.MdiParent = Me
        Form1.Show()
```

EK 3 ' ün devamı

```
Form1.WindowState = FormWindowState.Maximized
    End Sub

    Private Sub MDIParent1_Resize(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Resize
        Form1.WindowState = FormWindowState.Maximized
        Form2.WindowState = FormWindowState.Maximized
        Form3.WindowState = FormWindowState.Maximized
        Form4.WindowState = FormWindowState.Maximized
    End Sub

    Private Sub PasteToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
PasteToolStripMenuItem.Click
        OpenFileDialog1.InitialDirectory =
IO.Directory.GetCurrentDirectory & "\cor"
        OpenFileDialog1.Filter = "all files (*.*)|*.*"
        If (OpenFileDialog1.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then

            cor = OpenFileDialog1.FileName
            main_cor = File.ReadAllLines(cor)

            ss = File.ReadAllLines(main_cor(0))
            tt = main_cor(1).Split(",")

            sholds = tt(0)
            cellwper = tt(1)
            main_err_per = tt(2)
            main_darkcount = tt(3)
            main_zm = tt(4)

            NumericUpDown3.Value = tt(0)
            NumericUpDown4.Value = tt(1) * 100
            NumericUpDown5.Value = tt(2)
            NumericUpDown6.Value = tt(3)
            NumericUpDown8.Value = tt(4)

            rr = ss(1).Split(",")
            With main_tmp
                .stpointx1 = rr(0)
                .stpointy1 = rr(1)
                .stpointx2 = rr(2)
                .stpointy2 = rr(3)
                .stpointw = rr(4)
                .stpointh = rr(5)
                .stpointmw = rr(6)
                .stpointmh = rr(7)
                .stpointcx = rr(8)
                .stpointcy = rr(9)
                .stpointper = rr(10)
```

EK 3 ' ün devamı

```
                .enpointx1 = rr(11)
                .enpointy1 = rr(12)
                .enpointx2 = rr(13)
                .enpointy2 = rr(14)
                .enpointcx = rr(15)
                .enpointcy = rr(16)
                .bubw = rr(17)
                .bubh = rr(18)
            End With
            For i As Integer = 2 To ss.Length - 1
                rr = ss(i).Split(",")
                grp.type = rr(0)
                grp.name = rr(1)
                grp.row = rr(2)
                grp.col = rr(3)
                grp.gridx1 = rr(4)
                grp.gridy1 = rr(5)
                grp.gridx2 = rr(6)
                grp.gridy2 = rr(7)
                grp.gridw = rr(8)
                grp.gridh = rr(9)
                grp.gridgw = rr(10)
                grp.gridgh = rr(11)
                grp.gridmgw = rr(12)
                grp.gridmgh = rr(13)


                main_tmp.group.Add(grp)
                Next

Me.correctAnswerMenu.DropDownItems(0).Enabled = True
Me.SheetsMenu.Enabled = True
ToolStripStatusLabel.Text = "template : " & main_cor(0) & "
Correct answer : " & cor
End If

End Sub

Private Sub OpenToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
OpenToolStripMenuItem1.Click
        If Form4.IsHandleCreated Then Form4.Close()
        Form4.MdiParent = Me
        Form4.Show()
        Form4.WindowState = FormWindowState.Maximized
End Sub

Private Sub ErrorListToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ErrorListToolStripMenuItem.Click
        If Form6.IsHandleCreated Then Form6.Close()
```

EK 3 'ün devamı

```
        Form6.MdiParent = Me
        Form6.Show()
        Form6.WindowState = FormWindowState.Maximized
    End Sub


Private Sub MDIParent1_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        With Me

            .ToolStrip2.Width = .Width
            .ToolStrip1.Width = .Width
            .ToolStripButton1.Image = Image.FromFile("icons\a.bmp")
            .ToolStripButton2.Image = Image.FromFile("icons\b.bmp")
            .ToolStripButton4.Image = Image.FromFile("icons\d.bmp")
            .ToolStripButton5.Image = Image.FromFile("icons\e.bmp")
            .ToolStripButton6.Image = Image.FromFile("icons\g.bmp")
            .ToolStripButton7.Image = Image.FromFile("icons\f.bmp")
            .ToolStripButton8.Image =
Image.FromFile("icons\gr.jpg")
        End With
End Sub

Private Sub ToolStripButton1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolStripButton1.Click

        ToolStripButton1.Checked = True
        ToolStripButton2.Checked = False

        ToolStripButton4.Checked = False
        ToolStripButton6.Checked = False

        StatusStrip.Items(0).Text = ">>Start point"

    End Sub

Private Sub ToolStripButton2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolStripButton2.Click

        ToolStripButton2.Checked = True
        ToolStripButton1.Checked = False

        ToolStripButton4.Checked = False
        ToolStripButton6.Checked = False

        StatusStrip.Items(0).Text = ">>Start field"


End Sub
```

EK 3 'ün devamı

```
Private Sub ToolStripButton4_CheckedChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles ToolStripButton4.Click
        ToolStripButton8.Visible = Not ToolStripButton8.Visible
        ToolStripComboBox2.Visible = Not ToolStripComboBox2.Visible
        ToolStripLabel1.Visible = Not ToolStripLabel1.Visible
        ToolStripLabel2.Visible = Not ToolStripLabel2.Visible
        ToolStripComboBox1.Visible = Not ToolStripComboBox1.Visible
        ToolStripTextBox1.Visible = Not ToolStripTextBox1.Visible

    End Sub
    Private Sub ToolStripButton4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton4.Click


        ToolStripButton1.Checked = False
        ToolStripButton2.Checked = False

        ToolStripButton4.Checked = True
        ToolStripButton6.Checked = False

        StatusStrip.Items(0).Text = ">>groups"


    End Sub

    Private Sub ToolStripButton5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton5.Click
        Dim filenam As String
        Dim myfile As StreamWriter
        'writes the information to text
        filenam = "temp\" & ToolStripTextBox4.Text & ".tmp"
        myfile = File.CreateText(filenam)
        With Form2
            myfile.WriteLine(Form2.zoom.Size.Width & "," &
Form2.zoom.Size.Height)

            myfile.WriteLine(.t.stpointx1 & "," & .t.stpointy1 &
"," & .t.stpointx2 & "," & .t.stpointy2 & "," & .t.stpointw & "," &
.t.stpointh & "," & _
                                .t.stpointmw & "," & .t.stpointmh &
"," & .t.stpointcx & "," & .t.stpointcy & "," & .t.stpointper & ","
& .t.enpointx1 _
                                 & "," & .t.enpointy1 & "," &
.t.enpointx2 & "," & .t.enpointy2 & "," & _
                                  .t.enpointcx & "," & .t.enpointcy &
"," & .t.bubw & "," & .t.bubh)
 For Each i As temp.grouptype In .t.group
```

EK 3 'ün devamı

```
myfile.WriteLine(i.type & "," & i.name & "," & i.row & "," & i.col
& "," & i.gridx1 - .t.stpointcx & "," & i.gridy1 - .t.stpointcy &
"," & _
                     i.gridx2 - .t.stpointcx & "," & i.gridy2 -
.t.stpointcy & "," & i.gridw & "," & i.gridh & "," & i.gridgw & ","
& i.gridgh & "," & i.gridmgw & "," & i.gridmgh)


            Next

            myfile.Close()
            'main_tmp = .t
            .t.copyto(main_tmp)
        End With
        Me.Text = Directory.GetCurrentDirectory & "\" & filenam
        temp = Directory.GetCurrentDirectory & "\" & filenam
        ToolStripStatusLabel.Text = "template : " & Me.Text & "
Correct answer : "
        correctAnswerMenu.DropDownItems(0).Enabled = True

    End Sub
    Private Sub ToolStripButton6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton6.Click


        ToolStripButton1.Checked = False
        ToolStripButton2.Checked = False

        ToolStripButton4.Checked = False
        ToolStripButton6.Checked = True

        Me.StatusStrip.Items(0).Text = ">>bubble size"

    End Sub
    Private Sub ToolStripButton7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton7.Click
        Form5.Visible = Not Form5.Visible

    End Sub


    Private Sub ToolStripButton8_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton8.Click
        Form2.addgroup()
    End Sub
```

EK 3' ün devamı

```
Private Sub NumericUpDown1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles NumericUpDown1.Click

        Form2.drawboxgroup()
    End Sub

    Private Sub NumericUpDown1_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown1.ValueChanged

    End Sub

    Private Sub NumericUpDown2_Click(ByVal sender As Object, ByVal
e As System.EventArgs) Handles NumericUpDown2.Click

        Form2.drawboxgroup()
    End Sub


    Private Sub ToolStrip1_ItemClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
ToolStrip1.ItemClicked

    End Sub

    Private Sub ToolStripComboBox1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripComboBox1.Click

    End Sub

    Private Sub ToolStripComboBox1_TextChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
ToolStripComboBox1.TextChanged
        Form2.grpp.type = ToolStripComboBox1.SelectedIndex
    End Sub

    Private Sub ToolStripTextBox1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripTextBox1.Click

    End Sub

    Private Sub ToolStripTextBox1_TextChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
ToolStripTextBox1.TextChanged
        Form2.grpp.name = ToolStripTextBox1.Text
    End Sub
```

EK 3 'ün devamı

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        OpenFileDialog1.InitialDirectory =
IO.Directory.GetCurrentDirectory & "\images"
        OpenFileDialog1.Filter = "all files (*.*)|*.*"
        If (OpenFileDialog1.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then

            Form3.pic = Image.FromFile(OpenFileDialog1.FileName)

            Form3.MdiParent = Me
            Form3.Show()
            Form3.PictureBox1.Image = New Bitmap(Form3.pic)
            Form3.WindowState = FormWindowState.Maximized
            Button2.Enabled = True
        Else : MsgBox("no files open ") : Exit Sub
        End If

    End Sub
    Private Sub ToolStripTextBox1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripTextBox1.Click

    End Sub

    Private Sub ToolStripTextBox1_TextChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
ToolStripTextBox1.TextChanged
        Form2.grpp.name = ToolStripTextBox1.Text
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click

        Form3.Close()
        Form3.pic = Image.FromFile(OpenFileDialog1.FileName)

        Form3.PictureBox1.Image = Form3.pic
        Form3.MdiParent = Me
        Call Form3.scan_cor()
        Form3.Show()

        Form3.WindowState = FormWindowState.Maximized
    End Sub

    Private Sub NumericUpDown3_ValueChanged_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown3.ValueChanged
sholds = NumericUpDown3.Value
    End Sub
```

EK 3' ün devamı

```
Private Sub NumericUpDown4_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown4.ValueChanged

cellwper = NumericUpDown4.Value / 100
    End Sub
Private Sub NumericUpDown5_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown5.ValueChanged
        main_darkcount = NumericUpDown5.Value
    End Sub

    Private Sub NumericUpDown6_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown6.ValueChanged
        main_err_per = NumericUpDown6.Value
    End Sub

    Private Sub NumericUpDown7_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown7.ValueChanged
        main_err_per = NumericUpDown7.Value
    End Sub

    Private Sub Panel2_Paint(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.PaintEventArgs) Handles Panel2.Paint

    End Sub

    Private Sub MenuStrip_ItemClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
MenuStrip.ItemClicked

    End Sub
    Private Sub NumericUpDown8_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
NumericUpDown8.ValueChanged
        main_zm = NumericUpDown8.Value
    End Sub
End Class

Public Class Form2
    Public zoom As Bitmap
    Dim npp As New Pen(Color.DarkOliveGreen, 15)
    Dim lef As Boolean = False
    Public t As New temp
    Public grpp As temp.grouptype
    Public gg As Graphics
    Public x1, y1, x2, y2 As Integer
    Public xx1, yy1, xx2, yy2 As Integer
```

EK 3 'ün devamı

```
    Public mmx, mmy, ffx As Integer
    Dim org, tmp, pic5 As Bitmap
    Public testx, testy As Integer
    Dim ax1, ay1, ax2, ay2 As Integer
    Public lbw, lbh, lbs As Integer
    Dim newcol, col As Color
    Dim r1, r2, r3, r4, rr, gr, bb, aa, gry As Byte
    Dim bubx1, buby1 As Integer

Private Sub PictureBox1_MouseClick(ByVal sender As Object, ByVal e
As System.Windows.Forms.MouseEventArgs) Handles
PictureBox1.MouseClick
        Beep()
        If e.Button = Windows.Forms.MouseButtons.Left Then
            lef = True
            If MDIParent1.ToolStripButton6.Checked Then
                bubx1 = e.X
                buby1 = e.Y
            End If
            If MDIParent1.ToolStripButton1.Checked Then
                t.stpointx1 = e.X
                t.stpointy1 = e.Y
            End If

            If MDIParent1.ToolStripButton2.Checked Then
                t.enpointx1 = e.X
                t.enpointy1 = e.Y
            End If
            If MDIParent1.ToolStripButton4.Checked Then
                grpp.gridx1 = e.X
                grpp.gridy1 = e.Y
            End If
        End If
        If e.Button = Windows.Forms.MouseButtons.Right Then
            If lef Then
                If MDIParent1.ToolStripButton6.Checked Then
                    t.bubw = e.X - bubx1
                    t.bubh = e.Y - buby1
                    Me.Text = t.bubw & " - " & t.bubh
                End If
                If MDIParent1.ToolStripButton1.Checked Then
                    t.stpointx2 = e.X
                    t.stpointy2 = e.Y
                    t.stpointw = t.stpointx2 - t.stpointx1
                    t.stpointh = t.stpointy2 - t.stpointy1
                    t.stpointmw = t.stpointw / 2
                    t.stpointmh = t.stpointh / 2
                    t.stpointcx = t.stpointx1 + t.stpointmw
                    t.stpointcy = t.stpointy1 + t.stpointmh
                    t.stpointper = MDIParent1.main_err_per
                End If
```

EK 3 'ün devamı

```
If MDIParent1.ToolStripButton2.Checked Then
                    t.enpointx2 = e.X
                    t.enpointy2 = e.Y
t.enpointcx = t.enpointx1 + ((t.enpointx2 - t.enpointx1) / 2)
t.enpointcy = t.enpointy1 + ((t.enpointy2 - t.enpointy1) / 2)
                End If
                If MDIParent1.ToolStripButton4.Checked Then
                    If Not (t.bubh = 0 Or t.bubw = 0 Or e.X <
grpp.gridx1 Or e.Y < grpp.gridy1 Or grpp.gridx1 = Nothing) Then
                        grpp.gridx2 = e.X
                        grpp.gridy2 = e.Y
                        grpp.type =
MDIParent1.ToolStripComboBox1.SelectedIndex
                        grpp.name =
MDIParent1.ToolStripTextBox1.Text
                        If grpp.type < 3 Then
                            grpp.gridw = grpp.gridx2 - grpp.gridx1
                            grpp.gridh = grpp.gridy2 - grpp.gridy1
                            grpp.col = Math.Round(grpp.gridw /
t.bubw) : grpp.row = Math.Round(grpp.gridh / t.bubh)
MDIParent1.NumericUpDown1.Value = grpp.col
MDIParent1.NumericUpDown2.Value = grpp.row
drawboxgroup()
End If
Else : MsgBox("check bubble size or left click first ...")
End If
End If
Else
MsgBox("click left button first")
            End If
        End If
    End Sub

Private Sub Form2_Disposed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Disposed
        MDIParent1.ToolStrip1.Visible = False
        MDIParent1.ToolStrip2.Visible = False
        MDIParent1.Panel1.Visible = False
        Form5.Hide()
    End Sub

Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        pic5 = New Bitmap(32, 32)
        zoom = PictureBox1.Image
        org = MDIParent1.org
        tmp = MDIParent1.tmp
        gg = Graphics.FromImage(zoom)
        npp.Alignment = Drawing2D.PenAlignment.Inset
End sub
```

EK 3 'ün devamı

```
Private Sub PictureBox1_MouseMove(ByVal sender As Object, ByVal e
As System.Windows.Forms.MouseEventArgs) Handles
PictureBox1.MouseMove

        If MDIParent1.ToolStripButton4.Checked Then
            If e.Button = Windows.Forms.MouseButtons.Right Then

                grpp.gridx2 = e.X
                grpp.gridy2 = e.Y
                gg.DrawImage(tmp, 0, 0)
                gg.DrawRectangle(Pens.Red, New
Rectangle(grpp.gridx1 - grpp.gridmgw, grpp.gridy1 - grpp.gridmgh,
grpp.gridx2 - grpp.gridx1 + grpp.gridgw, grpp.gridy2 - grpp.gridy1
+ grpp.gridgh))
                PictureBox1.Image = zoom
            End If
        Else
            If e.X >= 14 And e.Y >= 14 And e.X + 14 < zoom.Width
And e.Y + 14 < zoom.Height Then Form5.PictureBox1.Image =
zoom.Clone(New Rectangle(e.X - 14, e.Y - 14, 28, 28),
Imaging.PixelFormat.Format24bppRgb) : Form5.Focus()
        End If
    End Sub
Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles PictureBox1.Click

    End Sub
    Public Sub drawboxgroup()
        If grpp.type = 3 Then
            gg.DrawRectangle(npp, New Rectangle(grpp.gridx1,
grpp.gridy1, grpp.gridx2 - grpp.gridx1, grpp.gridy2 - grpp.gridy1))

        Else
            grpp.col = MDIParent1.NumericUpDown1.Value
            grpp.row = MDIParent1.NumericUpDown2.Value

            Me.Text &= "  " & grpp.col & " - " & grpp.row
            grpp.gridgw = grpp.gridw / grpp.col
            grpp.gridgh = grpp.gridh / grpp.row
            grpp.gridmgw = grpp.gridgw / 2
            grpp.gridmgh = grpp.gridgh / 2
            gg.DrawImage(tmp, 0, 0)
            For i = 0 To grpp.col
                For j = 0 To grpp.row
                    lbs = 0
                    For lbw As Integer = 0 To grpp.gridgw *
MDIParent1.cellwper
                        For lbh As Integer = 0 To grpp.gridgh *
MDIParent1.cellwper
```

EK 3 'ün devamı

```
newcol = zoom.GetPixel(lbw + grpp.gridx1 - grpp.gridmgw + i *
grpp.gridgw, lbh + grpp.gridy1 - grpp.gridmgh + j * grpp.gridgh)
                            aa = newcol.A
                            rr = newcol.R
                            gr = newcol.G
                            bb = newcol.B
  gry = 0.2989 * rr + 0.587 * gr + 0.114 * bb

If gry < MDIParent1.sholds Then lbs += 1
Next
Next
lbs = lbs / (grpp.gridgw * grpp.gridgh) * 100
  gg.DrawRectangle(Pens.Blue, New Rectangle(grpp.gridx1 -
grpp.gridmgw + i * grpp.gridgw, grpp.gridy1 - grpp.gridmgh + j *
grpp.gridgh, grpp.gridgw, grpp.gridgh))
gg.DrawString(lbs, New Font("arial", 8), Brushes.Red, grpp.gridx1 -
grpp.gridmgw + i * grpp.gridgw, grpp.gridy1 - grpp.gridmgh + j *
grpp.gridgh)
Next
Next
End If
PictureBox1.Image = zoom
    End Sub
    Public Sub addgroup()

t.group.Add(grpp)

MDIParent1.ToolStripComboBox2.Items.Add(MDIParent1.ToolStripComboBo
x2.Items.Count + 1 & grpp.name)
        If grpp.type < 3 Then
            gg.DrawRectangle(npp, New Rectangle(grpp.gridx1 -
grpp.gridmgw, grpp.gridy1 - grpp.gridmgh, grpp.gridw + grpp.gridgw,
grpp.gridh + grpp.gridgh))
            gg.DrawString("saved...", New Font("arial", 6),
Brushes.Yellow, grpp.gridx1 - grpp.gridmgw, grpp.gridy1 -
grpp.gridmgh)
        Else
            gg.DrawRectangle(npp, New Rectangle(grpp.gridx1,
grpp.gridy1, grpp.gridx2 - grpp.gridx1, grpp.gridy2 - grpp.gridy1))
            gg.DrawString("saved...", New Font("arial", 6),
Brushes.Yellow, grpp.gridx1, grpp.gridy1)

     End If
        PictureBox1.Image = zoom
    End Sub

    Private Sub PictureBox1_MouseHover(ByVal sender As Object,
ByVal e As System.EventArgs) Handles PictureBox1.MouseHover

    End Sub
End Class
```

ÖZGEÇMİŞ

Adı Soyadı : Asmaeil Ammarah Abdullah BALQ

Doğum Yeri ve Yılı : Agelit/Libya,1976

Medeni Hali : Evli

Yabancı Dili : İngilizce ve Türkçe

E-posta : blag.ismael@gmail.com

Eğitim Durumu

Lise : Al-Agelat Lisesi, Libya

Lisans : Eğitim Üniversitesi, Libya

Yüksek Lisans : Omdurman Islam Üniversitesi, Sudan

Mesleki Deneyim

İş Yeri : Zawiya Üniversitesi, Libya