



T.C.

NİĞDE ÖMER HALİSDEMİR ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

MEKATRONİK MÜHENDİSLİĞİ

2016-2017 EÖY LİSANS BİTİRME TEZİ PROJESİ

KAZA TESPİT SİSTEMİ

HAZIRLAYANLAR

Yasemin YILDIZ

130607012

Meltem MUHSUROĞLU

130607034

DANIŞMAN

Yrd. Doç. Dr. İlyas KACAR

NİĞDE, 2017

T.C.
NİĞDE ÖMER HALİSDEMİR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ
BİTİRME TEZİ KABUL VE ONAY BELGESİ

Bölümümüz 130607012 numaralı öğrencisi Yasemin Yıldız' ın, 130607034 numaralı öğrencisi Meltem Muhsuroğlu' nun “Kaza Tespit Sistemi” başlıklı Bitirme Tezi çalışması aşağıdaki jüri üyeleri tarafından Mekatronik Mühendisliği Bölümü’nde Bitirme Tezi olarak Oy Birliği/Oy Çokluğu ile kabul edilmiştir.

Danışman: Yrd. Doç. Dr. İlyas KACAR

İmzası

Üye: Unvanı Adı Soyadı : Yrd. Doç. Dr. M. Kürşat YALÇIN

İmzası

Üye: Unvanı Adı Soyadı: Öğrt. Gör. Dr. M. Ali EROĞLU

İmzası

Tezin Savunulduğu Tarih: 19/06/2017

Bitirme Tezi dersi kapsamında yapılan bu çalışma, ilgili jüriler tarafından değerlendirme sonucunda Mekatronik Mühendisliği Bölümü’nde Bitirme Tezi çalışması olarak kabul edilmiştir.

İmza

Bölüm Başkanı V.

Doç. Dr. Murat BARUT

T.C.

NİĞDE ÖMER HALİSDEMİR ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

DOĞRULUK BEYANI

Bitirme tezi olarak sunduğumuz bu çalışmayı tüm akademik kurallara ve Niğde Ömer Halisdemir Üniversitesi Yayın Etiği Komisyonu Yönergesine uygun olarak gerçekleştirdiğimizi ve sunduğumuzu; bu kurallar ve ilkelere aykırı hiç bir yol ve yardıma başvurmaksızın bizzat hazırladığımızı beyan ederiz.

Tezimizle ilgili yaptığımız beyana aykırı bir durum saptanırsa ortaya çıkacak tüm ahlaki ve hukuki sonuçlara katlanacağımızı bildiririz.

İmza

Yasemin Yıldız

İmza

Meltem Muhsuroğlu

TEŞEKKÜR

Öncelikle bu projede bizden hiçbir yardımını esirgemeyen değerli danışman hocamız Yrd. Doç. Dr. İlyas KACAR' a, Elektrik Elektronik Mühendisliği Bölümü hocalarımızdan Öğr. Gör. İbrahim YÜCEL' e, Yrd. Doç. Dr. Yasin KABALCI' ya ve Mekatronik Mühendisliği Bölümü hocalarımızdan Yrd. Doç. Dr. Mehmet Kürşat YALÇIN'a, Öğrt. Gör. Dr. M. Ali EROĞLU'na tüm çalışmalarımızda gerek laboratuvar ortamında sağladıkları imkânlar, gerekse bilgi alma aşamasında gösterdikleri bütün yardımlarından dolayı teşekkür ederiz.

Sanayi danışmanımız SKY Robot Teknolojileri Sanayi Ticaret Ltd. Şti' ne proje kapsamında sağladığı imkânlar için teşekkür ederiz.

Ayrıca projemize “TUBİTAK 2241-A Sanayi Odaklı Lisans Bitirme Tezi Destekleme Programı” kapsamında destek olan Türkiye Bilimsel ve Teknolojik Araştırma Kurumuna teşekkür ederiz.

Özellikle hayatımızın her aşamasında her konuda bizim yanımızda olan ailelerimize sonsuz teşekkürlerimizi sunarız.

Yasemin Yıldız

Meltem Muhsuroğlu

ÖZET

Günümüz dünyasında gün geçmeden birçok yenilikler gerçekleşmekte, her yeni üretilen ürün kendinden öncekini gölgede bırakarak yoluna devam etmektedir.

Bu projede, kamera yardımıyla alınacak görüntünün işlenmesi ile kaza yapmış olan araç tespit edilecek ve acil servise bilgi aktarımı yapılacaktır. Bu sayede erken uyarı-müdahale gerçekleştirilmiş olacaktır. Böylece de can kaybını en düşük seviyede tutmak ve insan hayatının önemini en üst seviyeye çıkarmak amaçlanmıştır.

Kaza tespit sistemi bir adet güneş paneli, bir adet akü ve kameradan oluşmaktadır. Sistemin kontrolü raspberry pi 3 ve hazırladığımız arayüz programı ile yapılmaktadır. Ara yüz programı Microsoft Visual Studio programı üzerinde, Visual Basic dilinde yazılmış ve kameradan alınan görüntünün işlenmesi sağlanmıştır.

Anahtar kelimeler: Microsoft Visual Studio, Visual Basic, Güneş paneli, Akü, Raspberry Pi 3, Kaza Tespit Sistemi

ABSTRACT

In today's world, many innovations are taking place before the day, and every new product is going on its way by leaving itself in the shade.

In this project, the accidental vehicle will be detected and the emergency service information will be transmitted by processing the images taken by a camera. Early warning intervention will be done at this point. Thus, it is aimed to keep the loss of life at the lowest level and bring the importance of human life to the highest level.

The accident detection system consists of one solar panel, one battery and a camera. The system is controlled by Paspberry Pi 3 and the interface program that we have prepared. The interface program is written in Microsoft Visual Studio program by using Visual Basic language and images from the camera are processed bu interface program.

Key words: Microsoft Visual Studio, Visual Basic, Solar Panel, Battery, Raspberry pi 3, Accident Detection System

İÇİNDEKİLER

BİTİRME TEZİ KABUL VE ONAY BELGESİ.....	ii
DOĞRULUK BEYANI	iii
TEŞEKKÜR.....	iv
ÖZET.....	v
ABSTRACT	vi
İÇİNDEKİLER.....	vii
ŞEKİLLER LİSTESİ.....	ix
TABLO LİSTESİ	x
ÖNSÖZ.....	xi
BÖLÜM 1	1
1.1 Giriş ve Teorik Çerçeve	1
1.2 Çalışmanın Amacı	2
1.3 Çalışmanın Önemi.....	3
1.4 Projenin Gerçekleşme Düzeyi	4
BÖLÜM 2	6
2.1 Literatür Özeti	6
BÖLÜM 3	8
3.1 Arayüz Programı	8
3.1.1 Kameradan Görüntü Alma	8
3.1.2 Görüntü Üzerinde Riskli Bölgenin İşaretlenmesi	10
3.1.3 Kaza Anında Görüntünün Kaydedilmesi	12
3.1.4 Hassasiyet Ayarlama	13
3.1.5 SMS Modülü	13
3.2 Sistemin Gerekli Güç Hesabı	14
3.3 Raspberry İçin 32 Gb lık SD Karta Windows 10 Kurulumu	16
3.3.1 Dashboard Kurulumu	19
3.3.2 Windows IoT Kurulumu	20
BÖLÜM 4	23
4.1 Bütçe ve Gerekçesi.....	23
BÖLÜM 5	24
5.1 Sonuçlar.....	24
5.2 İleride Yapılması Planlanan Çalışmalar	27
KAYNAKLAR.....	29

EKLER	31
Ek 1: Microsoft Visual Basic İle Ana Program.....	31
ÖZGEÇMİŞ	44
1	44
2	45

ŞEKİLLER LİSTESİ

Şekil 1.1 Tasarlanan Sistemin Şeması.....	1
Şekil 3.1. Hazırlanan Ara Yüz Programı.....	9
Şekil 3.2 İşlenecek Olan Görüntünün Bölümleri	10
Şekil 3.3 İşlenecek Olan Riskli Bölgeler	11
Şekil 3.4 Eğimli Bağlantılarda Perspektif Nedeni İle Ufka Doğru Gittikçe Eğimli ..	12
Olacak Yol Çizgilerine Göre Ayarlanabilen Tarama Çizgileri	12
Şekil 3.5 Manual Olarak Görüntü Kaydetme Butonu	13
Şekil 3.6 Hassasiyet Ayar Butonu.....	13
Şekil 3.7 Gsm Modem İle Sms Atma.....	14
Şekil 3.8 Online Olarak Sms Atma	14
Şekil 3.9 Raspberry Pi 3	16
Şekil 3.10 Raspberry Pi Resmi Sitesinde İndirilebilecek İşletim Sistemleri	17
Şekil 3.11 Windows Yükleme Dosyası	18
Şekil 3.12 IoT Dashboard Penceresi 1	19
Şekil 3.13 IoT Dashboard Penceresi 2	20
Şekil 3.14 My Device Komut Penceresi	21
Şekil 3.15 Raspberry Pi 3 Windows IOT Core Kurulum Sonrası Ekranı	22
Şekil 5.1 Yazılım Arayüzü	24
Şekil 5.2 Yazılım Arayüzü 2	25
Şekil 5.3 Sistemin Güneş Paneli Ve Akü Montajı	25
Şekil 5.4 Raspberry Pi 3 ve Kızılötesi Kamera	26
Şekil 5.3 “S” Şeklindeki Virajlı Bir Yol	27
Şekil 5.4 Bir Kavşak İçin Kaza Senaryosu	28

TABLO LİSTESİ

Tablo 1.1 2006- 2015 Yılları Arasındaki Trafik Kazası İstatistikleri	3
Tablo 1.2 İş –Zaman Çizelgesi (Gant Chart)	5
Tablo 3.1 Malzemelerin Harcayacakları Enerji Hesabı	15
Tablo 3.2 Sistemin Güç Hesabı.....	15
Tablo 4.1 Bütçe Planı	23

ÖNSÖZ

Hazırlanan bu projede İlk olarak ilgili alan ve çalışmanın amacından bahsedilmiş olup konunun önemi özetlenmiştir.

Daha sonra, Niğde Ömer Halisdemir Üniversitesi tez yazım kuralları dikkate alınarak, giriş, literatür taraması ve bütçe hesabı gibi kısımlar detaylıca açıklanmıştır. Projede kullanılan tüm yazılım kodları Ek halinde bu belgenin sonuna eklenmiştir.

Daha sonra yapılacak geliştirme çalışmaları için detaylı bir açıklayıcı belge olduğu kanaatindeyiz.

BÖLÜM 1

1.1 Giriş ve Teorik Çerçeve

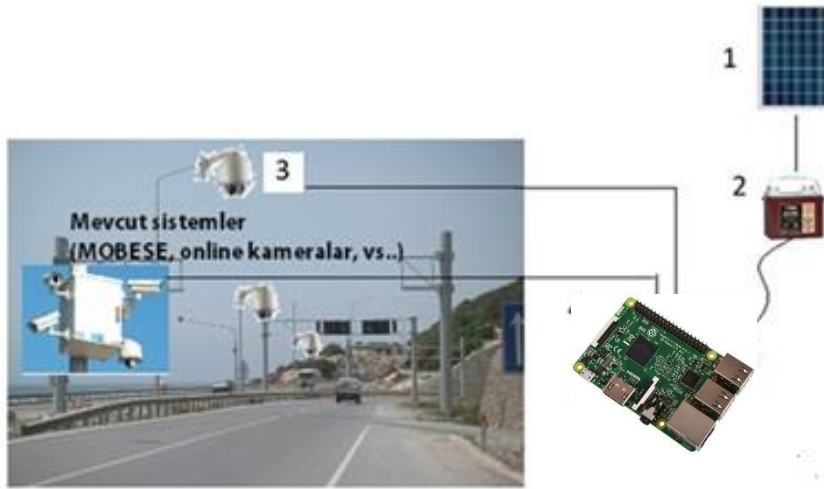
Tasarlanan sistemin şeması Şekil 1.1 de verilmiştir. Sistemin esası; otoyola kurulacak kameralar ağı sayesinde kazaların tespit edilebilmesidir.

Sistemi oluşturan temel elmanlar Şekil 1.1 den de görüleceği üzere;

- Kamera
- Güneş paneli-akü
- İşlemci

Ünitelerinden oluşmaktadır. “Kaza tespitini” sağlayacak ana fikir ise; sistemin çalışma algoritması 33 milisaniyede bir kameradan alınan anlık ardışık iki görüntü arasında, kullanıcının belirlediği güvenlik bölgelerinde meydana gelen sıradışı hareketleri algılamak esasına dayanmaktadır. Kıyaslama algoritması 33 milisaniye farkla alınan ardışık görüntülerdeki, belirlenen piksellerde kırmızı, mavi, yeşil renk yoğunluklarını hesaplayıp ardışık görüntüler arasında her bir pikseldeki renk yoğunluğu farkına bakmaktadır. Böylece İki ardışık görüntü arasında fark olup olmadığı kararını vermektedir.

Kameralar otoyolda özellikle, kaza riski olan bölgelere yerleştirilecektir.



Şekil 1.1 Tasarlanan Sistemin Şeması

- Projede kullanılan kamera kızıl ötesi olup, geceleri de görüntüyü en hassas biçimde almaktadır.
- Sistem güneş paneliyle çevrilmiştir. Bu sayede gün boyu alınan güneş enerjisi elektrik enerjisine çevrilmiş olup yaptığımız sistemin elektrik enerjisi tamamen karşılanmıştır. Ayrıca güneş panellerine bağladığımız akü sayesinde bu elde etmiş olduğumuz elektrik enerjisinin kullanılmayan fazla kısmı aküye depolanarak güneş yokken elektrik aküden karşılanmaktadır.
- Kullandığımız işlemci kameradan 33 milisaniyede bir alınan iki ardışık görüntüyü karşılaştırarak alınan görüntülerdeki araç, sürekli olarak duruyorsa veya kaza yapmışsa bunu tehlike olarak algılayıp 112 acil servise sinyal yollayacaktır. Bu karşılaştırma Visual basic programıyla gerçekleştirilecek yazılım ile yapılacaktır.
- Oluşturulan sistemin konumu belirli olup, 112 acile yollanacak olan sinyalde direkt olarak yer tespiti sağlanacaktır.

1.2 Çalışmanın Amacı

Projenin amacı; ücra köşelerde, kimsenin sıklıkla geçmediği yerlerde kazayı bildirecek kimsenin olmadığı durumlarda veya en sık kazaların meydana geldiği yerlerde, bu kazaların algılanıp acil servise bildirilmesi ve bu sayede insan hayatına verilen önemin en üst seviyelere çıkarılması, can kaybının ise en alt seviyelere indirilmesidir.

1.3 Çalışmanın Önemi

Günümüzde her ne kadar teknoloji sürekli olarak gelişse de ufak bir dikkatsizlik insan hayatında ciddi tehlikelere yol açmakta ve maalesef ki can kayıplarına sebebiyet vermektedir. Tablo 1.1’de görüldüğü üzere günümüze kadar trafik kazalarında birçok insan hayatını kaybetmiştir. Bu proje sayesinde trafik kazalarının erken tespiti ve müdahalesi sağlanarak can kayıplarının en aza indirilmesi amaçlanmıştır.

Tablo 1.1 2006- 2015 Yılları Arasındaki Trafik Kazası İstatistikleri

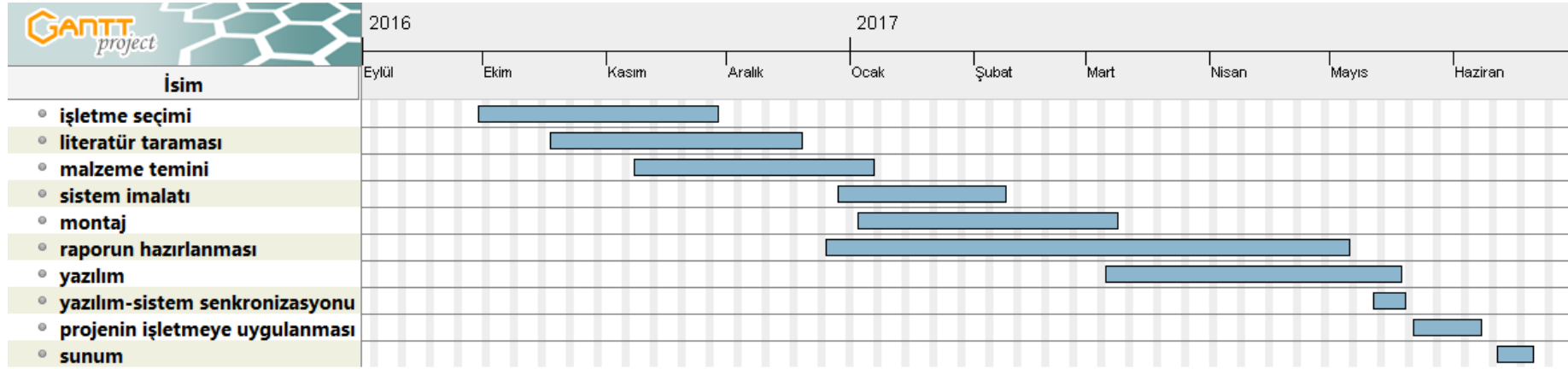
[Türkiye İstatistik Kurumu, “Karayolu Trafik Kaza İstatistikleri”,
<http://www.tuik.gov.tr/PreHaberBultenleri.do?id=18510>]

Yıl	Toplam Kaza Sayısı	Ölümlü Yaralanmalı Kaza Sayısı	Maddi Hasarlı Kaza Sayısı	Ölü Sayısı			Yaralı Sayısı
				Toplam	Kaza Yerinde	Kaza Sonrası	
2006	728 755	96 128	632 627	4 633	4 633	-	169 080
2007	825 561	106 994	718 567	5 007	5 007	-	189 057
2008	950 120	104 212	845 908	4 236	4 236	-	184 468
2009	1 053 346	111 121	942 225	4 324	4 324	-	201 380
2010	1 106 201	116 804	989 397	4 045	4 045	-	211 496
2011	1 228 928	131 845	1 097 083	3 835	3 835	-	238 074
2012	1 296 634	153 552	1 143 082	3 750	3 750	-	268 079
2013	1 207 354	161 306	1 046 048	3 685	3 685	-	274 829
2014	1 199 010	168 512	1 030 498	3 524	3 524	-	285 059
2015	1 313 359	183 011	1 130 348	7 530	3 831	3 699	304 421

1.4 Projenin Gerçekleşme Düzeyi

Projemizin kurulumu laboratuvar ortamında yapılmıştır. Denemeler ilk olarak laboratuvar da deney düzeneği oluşturulup oyuncak arabalarla kaza tespiti yapılarak sağlanmıştır. Oyuncak arabalarla olan deney başarılı olup daha sonraki denemeler otoyol üzerinde kontrollü şartlar altında başarıyla gerçekleştirilmiştir.

Tablo 1.2 İş –Zaman Çizelgesi (Gant Chart)



Proje için planlanan iş paketleri ve iş-zaman çizelgesi Tablo 1.2’ de verilmiştir. Planlanan iş paketleri herhangi bir aksaklık olmaksızın başarı ile gerçekleştirilmiştir. Tablo 1.2’ de verilen iş-zaman çizelgesinde de görüldüğü üzere projenin hayata geçmesi için ilk adımlara Eylül-2016 itibariyle başlanmıştır.

Malzeme temini literatür taraması başladıktan 15 gün kadar sonra başlamış ve Aralık ayında son bulması planlanmıştır ancak projedeki bazı problemler (yazılımsal hatalar) sebebiyle malzeme alımı Haziran-2017 tarihine kadar devam etmiştir. Malzeme teminin başlamasıyla birlikte sistemin kurulumu gerçekleştirilmiştir. Aynı zamanda rapor hazırlıkları da başlamış ve sistemin denemeleri bitene kadar devam etmiştir. Kaza tespit sisteminin montajı tamamen bittikten hemen sonra deneme aşamasına geçilmiştir ve denemelerden olumlu sonuç alınarak sistem tamamen hayata geçmiştir. Proje süreci, Haziran-2017 ayında yapılan sunum ile son bulmuştur.

BÖLÜM 2

2.1 Literatür Özeti

Yapılan literatür taramasından; mevcut çalışmaların ya akıllı telefon sensörleri sayesinde ya da araç üzerinde mevcut kamera ve bilgisayar kullanıldığı gözlemlenmiştir. Buradan yola çıkarak, yapay görme tekniklerinin uygulaması esaslı bu çalışmayı yaptık.

Guido ve arkadaşları (2014) taşıt takibindeki kaza oranını azaltmak üzere güvenlik analizi yapmıştır. Araç filolarındaki birbirini takip eden araçların takiplerini Vissim yazılımı sayesinde genetik algoritma esaslı yapmışlardır. Trafikteki araçların birbirleri ile etkileşimi modelini oluşturmuşlardır [1]. Kansız (2013) bir kaza sırasında erken teşhis şansı kazandıracak ve ölümcül yaralanma riskini azaltacak bir telefon uygulaması geliştirmiştir [2]. Hu ve arkadaşları (2004) 3D model tabanlı araç takibi kullanarak kaza tahmini yapmak için bir olasılık modeli önermişlerdir. Hareket yörüngeleri de dâhil olmak üzere alınan veri ile 3D model tabanlı araç takibi yapmışlardır [3]. Sadeky ve arkadaşları (2010) gerçek zamanlı kaza tespiti için akış eğrisindeki değişim histogramı ile otomatik olarak kaza tanımlamasının yapılması önermişlerdir [4]. Matsushita ve arkadaşları (2002) hem zamana bağlı hem mekâna bağlı Markov rastgele alanı kaynak göstererek, kavşaklardaki trafik görüntüleri için bir uygulama algoritması geliştirmişlerdir [5]. White ve arkadaşları (2011) Iphone Ios ve Google Android tabanlı akıllı telefonların tüm sensörleri, web, ağ bağlantı özellikleri kullanılarak alınan ivme ve akustik veriler kullanılarak otomatik olarak fotoğraflar, GPS koordinatları, VOIP iletişim kanalları yardımıyla algılanan ve kazayı kayıt altına alan bir uygulama yapmışlardır [6]. Calafate ve arkadaşları (2011) On Board Diagnostics (OBD-II) Android tabanlı araç izleyen bir uygulama tasarlamışlardır [7]. Lagassey (2008) araç kaza tespiti, kaydedilmesi ve raporlanması için transdüktör kullanarak, bölgedeki ses çıkışlarını alıp akustik dalgalanmalarla kaza yer tespiti yapmıştır [8]. Ki ve Lee (2007) kavşaklardaki trafik kazalarını raporlamak için grafik tabanlı bir kaza tespit algoritması önermişlerdir [9]. Mackey ve arkadaşları (2000) bir veya daha fazla kamerayı aracın etrafına yerleştirerek kaza anında gerekli

verilerin araç tarafından otomatik olarak gönderilip bu verilere kablosuz bağlantı ile internet üzerinden ulaşılabilen bir sistem yapmışlardır [10].

BÖLÜM 3

3.1 Arayüz Programı

Sistemin tüm kodları Microsoft Visual Studio programında yazılmıştır. Microsoft Visual Basic dilinde yazılan kodlar EK 1 de verilmiştir. Şekil 3.1 de ise hazırlanan arayüz programı tanıtılmıştır.

Arayüz programı;

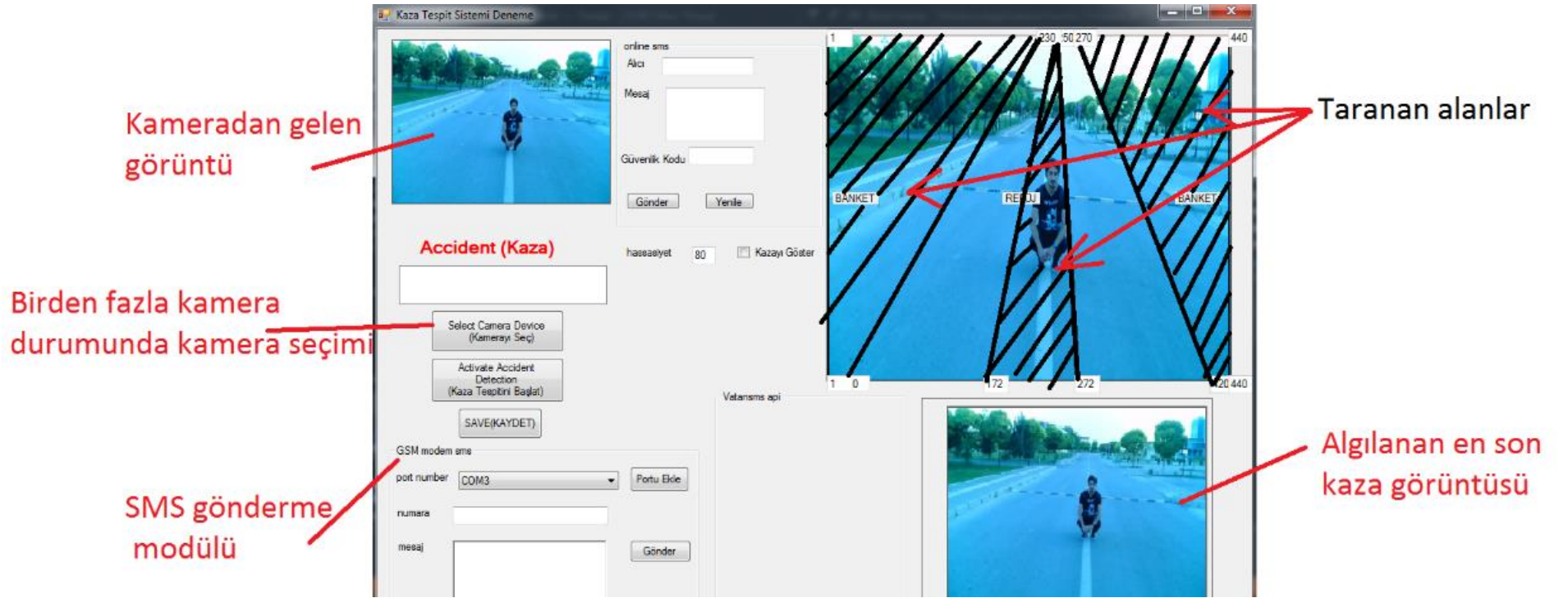
- Kamera seçimi ve görüntü alma
- Alınan görüntü üzerinde kaza tespit yapılması
- Kaza tespiti durumunda kaza sahnesinin fotoğrafının kaydedilmesi ve SMS ile bilgilendirme

modüllerinden oluşmaktadır.

3.1.1 Kameradan Görüntü Alma

Program ilk açıldığında Raspberry Pi sistemine takılı kaç adet kamera var ise listeleyerek kullanıcının istediği kamerayı seçmesine olanak tanımaktadır. Kamera görüntüsünün arayüz formuna aktarılabilmesi için Aforge Kütüphanesi ve Aforge fonksiyonları kullanılmıştır. Bu kütüphane ve fonksiyonlar yardımıyla takılan harici kameradan istenilen çözünürlükte görüntü alınması sağlanmıştır. Kamera seçimi kullanıcıya bağlı olup kamera ve çözünürlük seçimi kamerayı seç butonu ile çalışmaktadır.

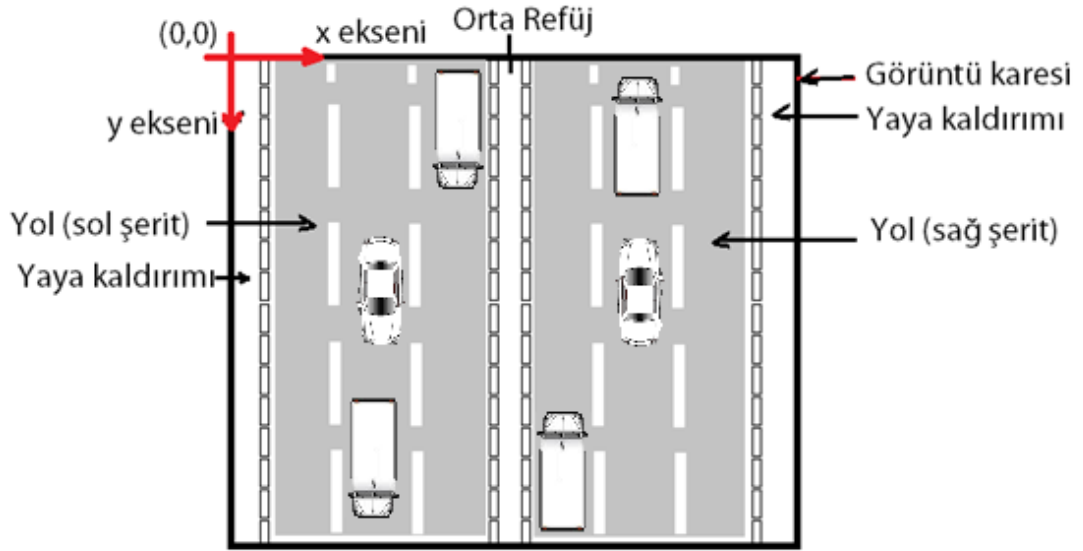
Bu ayarlar kurulum esnasında bir sefere mahsus yapılmakta, daha sonra ise sistem kendi kendine çalışabilmektedir. Uzak masaüstü bağlantısı ile sistem istenildiği zaman tekrar ayarlanabilmektedir.



Şekil 3.1. Hazırlanan Ara Yüz Programı

3.1.2 Görüntü Üzerinde Riskli Bölgenin İşaretlenmesi

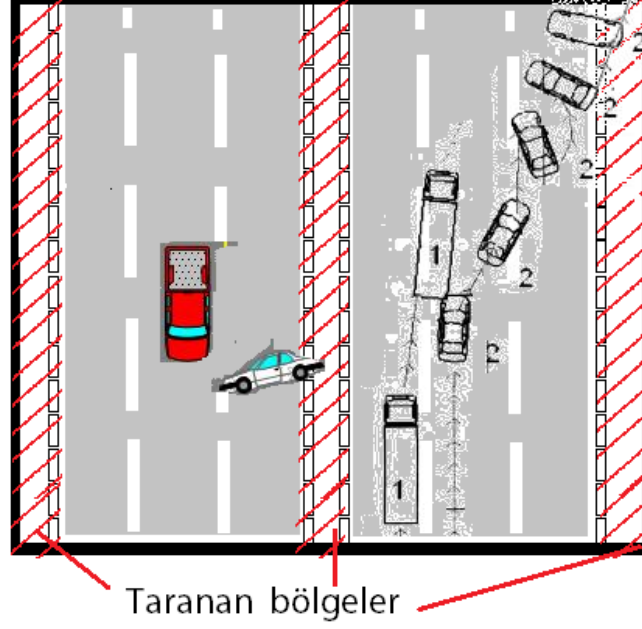
Kamera seçimi yapıldıktan sonra görüntü picturebox1 de real time video stream olarak görüntülenmektedir. Görüntü alımı başladıktan sonra asıl işlemin yapılacağı PB1 adlı pictureboxa anlık görüntü aktarılmaktadır. PB1 objesi (Şekil 3.1 üzerinde “taranan alanlar” oku ile gösterilmiştir), üzerinde görüntü işleme yapılan kutudur. Kameradan her 1/33 saniyede alınan görüntü işlenmektedir. Bu süre kullanıcı tarafından 1/1000sn ye kadar seçilebilmektedir. Şekil 3.2 de görüntü düzlemi görülmektedir. Koordinat eksen takımı, karenin sol üst köşesini başlangıç (0,0) kabul edecek şekilde gösterilmiştir.



Şekil 3.2 İşlenecek Olan Görüntünün Bölümleri

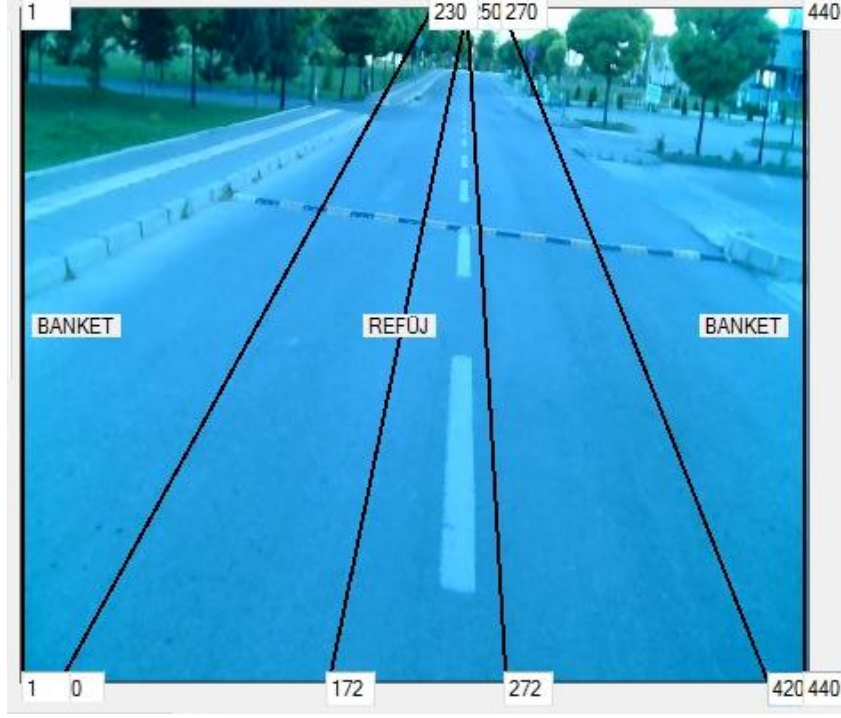
“Kaza tespitini” sağlayacak ana fikir ise; ardışık karelerin kıyaslanarak kare içerisinde sürekli olarak bekleme yapmakta olan unsur(lar)un tespiti ve/veya bekleme yapmakta olan unsurun ilk alınan görüntüsüyle son alınan görüntüsünün incelenip kıyaslanarak kazalı olup olmadıklarının ve konumlarının belirlenebilmesidir. Şekil 3.3 ten anlaşılacağı üzere orta refüj, sağ-sol banketler riskli bölge olarak işaretlenmektedir. Çünkü eğer bir kaza durumu yoksa yolun sağ ve sol şeritleri daima düzgün bir trafik akışı içerisinde olacaktır. Bir kaza durumunda ise trafik tıkanmakta ve orta refüj ve banketlerde durağan nesneler oluşmaktadır. Böylesi bir olağan dışı durumun oluşması

bir alarm ihtimalini göz önünde bulundurarak sahnenin fotoğrafını çekip durumu çağrı merkezine bildirmektedir.



Şekil 3.3 İşlenecek Olan Riskli Bölgeler

Ancak Şekil 3.3 te verilen sahne, kameranın otoyolun tam orta noktasına ve üst tarafına, tepeden bakacak şekilde yerleştirilmesi durumunu göstermektedir. Ancak her yol şartı bu şekilde değildir ya da kullanıcı kameranın oryantasyonunu, değişik şekillerde ayarlamak isteyebilir. Bu gibi durumlarda perspektif etkisinden dolayı, yolun şerit çizgileri, kaldırım taşlarının oluşturduğu çizgiler ufka doğru gittikçe birleşiyorlarmışçasına eğimli hale gelmektedir. Bu gibi durumlar için de, kullanıcının taranacak bölgeyi sınırlayan çizgileri ayarlamaya olanak sağlaması için Şekil 3.4 te görüleceği üzere ayarlanabilen tehditler oluşturulmuştur. Bu işlemler arayüz programında PB1 adlı görüntü penceresi üzerinde bulunan kuların koordinat değerlerini değiştirmek ile mümkün olabilmektedir.

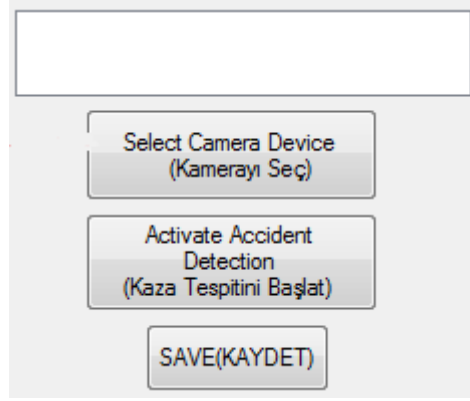


Şekil 3.4 Eğimli Bağlantılarda Perspektif Nedeni İle Ufka Doğru Gittikçe Eğimli Olacak Yol Çizgilerine Göre Ayarlanabilen Tarama Çizgileri

Kaza tespit sistemi algoritması kaza tespiti başlat butonu ile aktive edilmektedir. Bu butona basılınca kameradan anlık olarak alınan görüntü PB1 adındaki pictureboxta işlenekte, herhangi bir kaza algılanması durumunda şüpheli bölge görüntü üzerinde dikdörtgen içine alınmaktadır.

3.1.3 Kaza Anında Görüntünün Kaydedilmesi

Kullanıcı tarafından belirlenen bölgelerde kaza tespiti yapılırsa, kaza anının delili için kullanılması gerekir düşüncesiyle kazanın yapıldığı an görüntü olarak veritabanına otomatik olarak kaydedilmektedir. Ayrıca kullanıcı isterse şüpheli gördüğü anları kendisi de Şekil 3.5 te görüldüğü gibi “Kaydet” butonu ile manuel olarak veritabanına kaydedebilmektedir.



Şekil 3.5 Manual Olarak Görüntü Kaydetme Butonu

3.1.4 Hassasiyet Ayarlama

Aradışık görüntülerin kıyaslanması aşamasında iki ardışık görüntünün arasındaki kırmızı, mavi ve yeşil renklerinin yoğunlukları arasındaki farkın belli bir eşik değerine göre karşılatırılması gerekmektedir. Bu ayar “hassasiyet ayarı” adını verdiğimiz textbox ile yapılmaktadır (Şekil 3.6). Sayı değeri küçükse daha hassas karşılaştırma işlemi yapılır.



Şekil 3.6 Hassasiyet Ayar Butonu

3.1.5 SMS Modülü

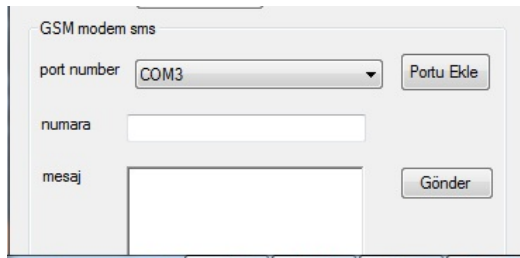
Kaza algılandığında, algılanan kazayı gerekli yerlere bildirmek için 4 adet sinyal (sms) gönderme algoritması programa dâhil edilmiştir. 4 farklı yöntemle SMS iletilmesini garantilemek amaçlanmıştır.

Birinci algoritma, online olarak bir internet sitesi vasıtasıyla istenilen numaralara algılanan kaza durumunu kullanıcı girişi ile iletmek için yazılmıştır.

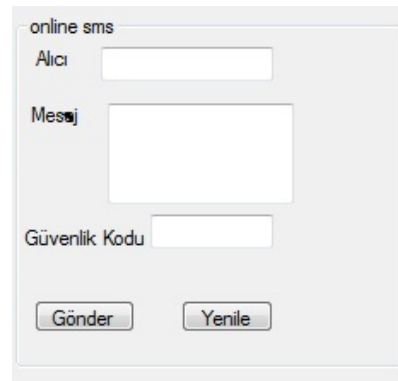
İkinci algoritma, harici olarak sisteme eklenen GSM modem kullanılarak SMS iletimi için yazılmıştır.

Üçüncü algoritma, Visual Basic dili için özel geliştirilmiş olan Textmagicrest kütüphanesi kullanılarak yazılıma eklenen Textmagicrest fonksiyonları ile SMS api uygulamasının kullanımı ile sağlanmıştır.

Dördüncü algoritma, temeli yine SMS api uygulamasına dayanan Türk bir şirketin sağlamış olduğu online olarak kullanılan özel bir algoritma ile sağlanmıştır. Burada birden fazla SMS'i, birden fazla alıcıya ulaştırmak için visual basic için özel olarak yazılan algoritma kullanılmıştır.



Şekil 3.7 Gsm Modem İle Sms Atma



Şekil 3.8 Online Olarak Sms Atma

3.2 Sistemin Gerekli Güç Hesabı

Sistemde güç çeken elemanlar Kızılötesi Kamera + Raspberry pi olup toplam güç tüketimi 10 watt dır. Niğde ilinin en az güneş görme süresi 3.9 saattir. 21 Aralık en uzun gece olup karanlık süresi 14.1 saattir. Dolayısıyla sistem güç hesabı Tablo 3.1 ve Tablo 3.2 de görüldüğü üzere, gece için 14.1 saat gündüz için 9.9 saat çalışacak şekilde hesaplanmıştır.

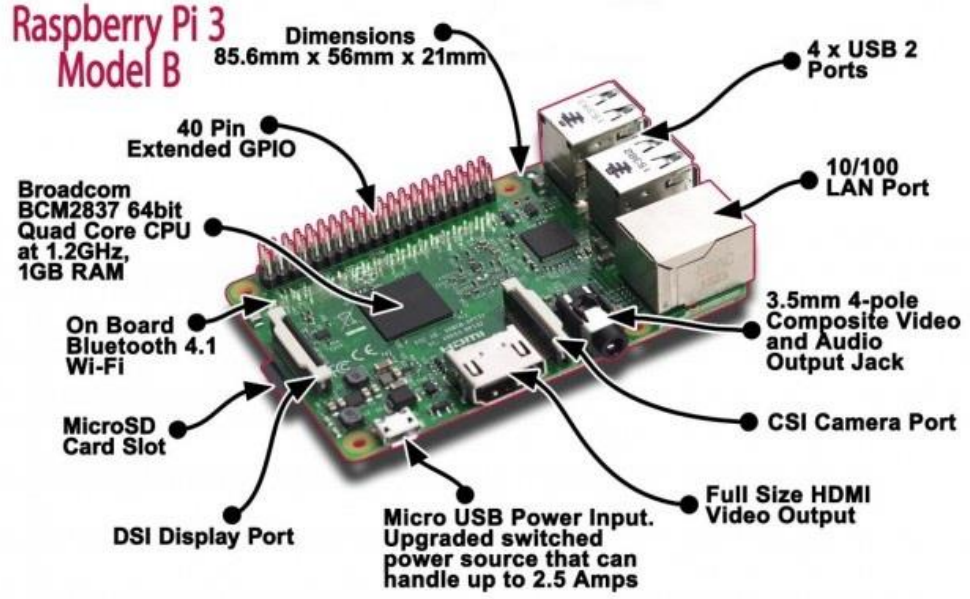
Tablo 3.1 Malzemelerin Harcayacakları Enerji Hesabı

Enerji	Kamera (6 Watt)	Raspberry Pi (4 Watt)
Gece Enerji Gereksinimi (14.1 saat)	14.1 saat*6W=84.6 Wh	14.1 saat*4W=56.4 Wh
Gündüz Enerji Gereksinimi (9.9 saat)	9.9 saat*3W=29.7 Wh	9.9 saat*4W=39.6 Wh
Toplam	114.3 Wh	96 Wh
Genel Toplam	210.3 Wh	

Tablo 3.2 Sistemin Güç Hesabı

Sistemin enerjisini karşılayacak güneş paneli minimum güç hesabı	210.3 Wh/3.9 h=53.92 W
Sistemin enerjisini depolayacak akü minimum akım hesabı	210.3 Wh/12 V=17.525 Ah*PF (Power Factor)

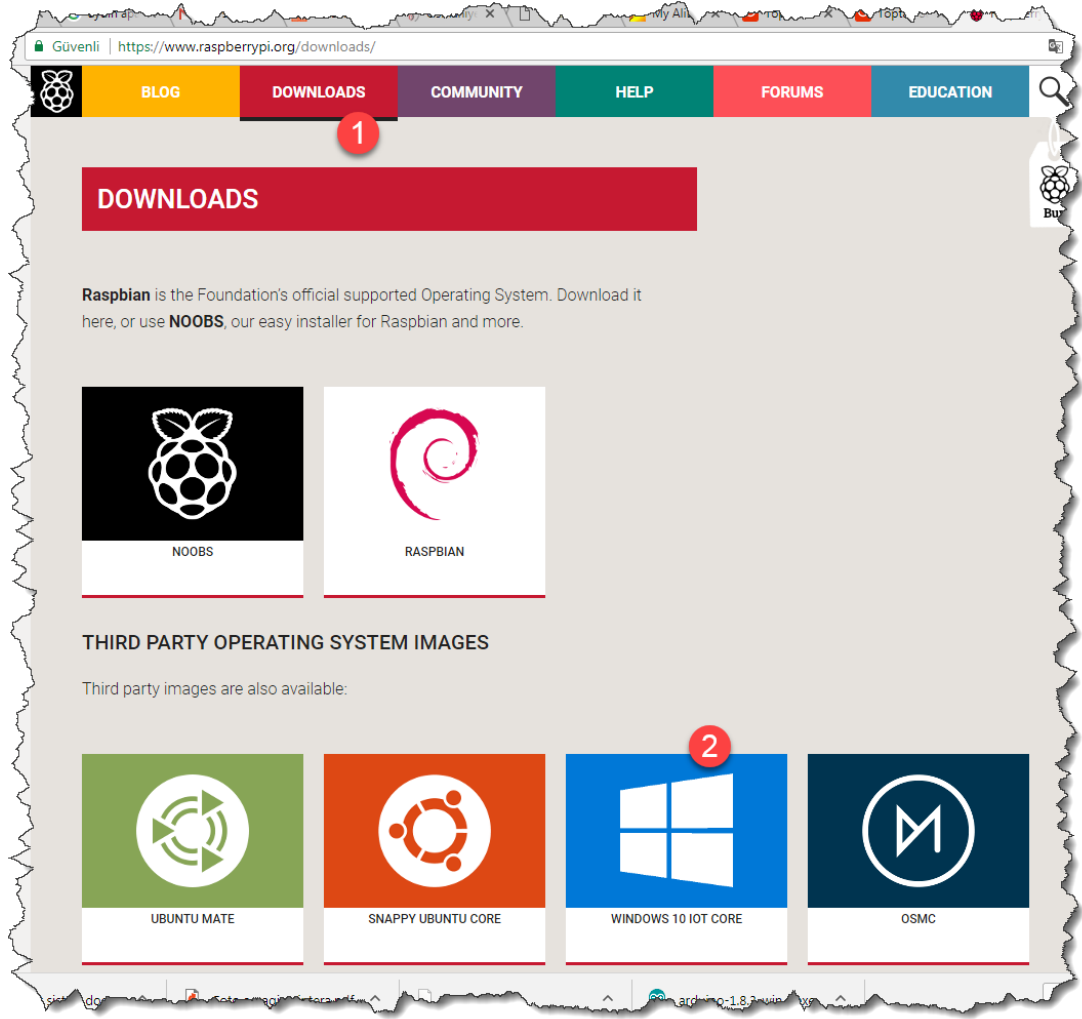
3.3 Raspberry İçin 32 Gb lık SD Karta Windows 10 Kurulumu



Şekil 3.9 Raspberry Pi 3

Şekil 3.9[11] te bir Raspberry Pi 3 tanıtılmıştır. Şekil 3.10 da ise kendi sitesinde, çeşitli işletim sistemleri için ücretsiz bulunan indirme linkleri gösterilmiştir. Buradan anlaşılacağı üzere Raspberry Pi 3 ile uyumlu olan tek Windows esaslı işletim sisteminin “Windows 10 IoT Core” olarak adlandırılmaktadır.

Downloads-Windows IoT üzerinde bulunan [Get Windows 10 Iot Core Insider Preview](#) bağlantısına tıklayarak, **Windows 10 Iot Core Insider Preview** içindeki **Select the edition** seçeneği altındaki **Windows 10 IoT Core Insider Preview-Build 15051** (yoksa en son çıkanı) seçip **Confirm** diyerek **Select to device** kısmına **Raspberry Pi** seçip **Download Now** butonuna basarak indiriyoruz. Tabi indirme işlemi yapabilmemiz için öncelikle Windows mail hesabımıza giriş yaparak indirmemiz bekleniyor.



Şekil 3.10 Raspberry Pi Resmi Sitesinde İndirilebilecek İşletim Sistemleri

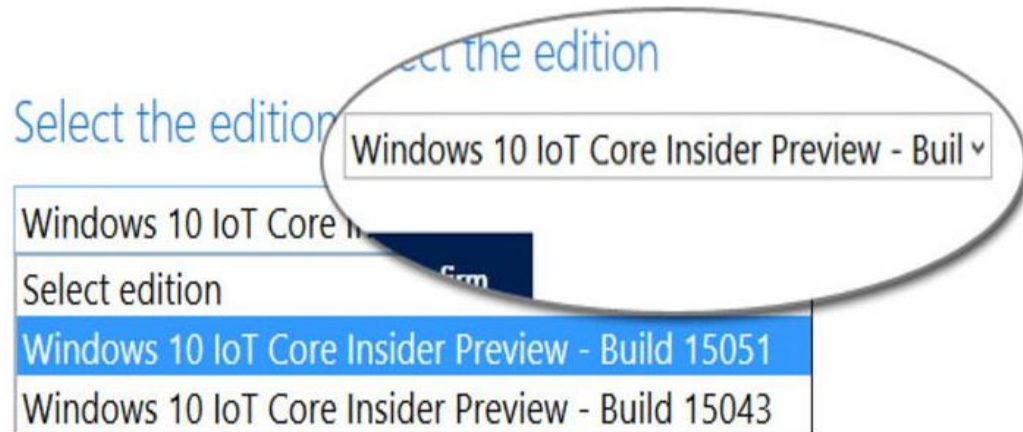
İndirdiğimiz, **Windows10_InsiderPreview_IoTCore_RPi_ARM32_en-us_15051.**

iso şeklinde ya da benzer isme sahip olan dosyamızı açıp, içerisinde ***.msi** uzantılı windows yükleme dosyasını çalıştırarak işleme başlayabiliriz.

Windows 10 IOT Core Insider Preview

Download Windows 10 IoT Core Insider Preview to get started developing for Internet

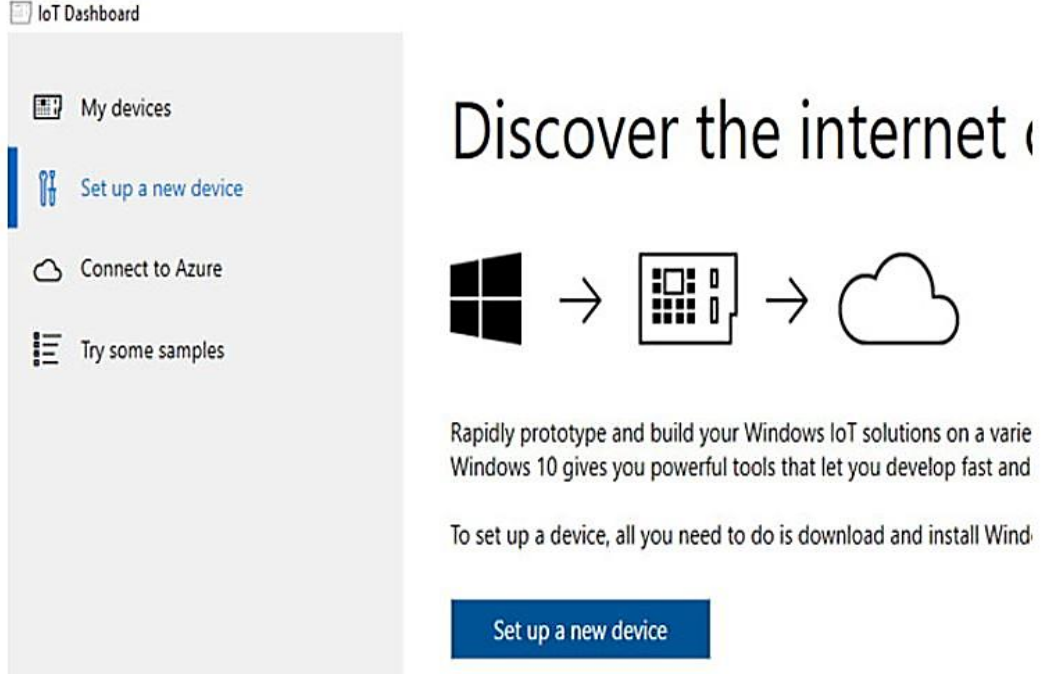
Refer back to [Windows on Devices](#) for additional downloads and developer tools.



Şekil 3.11 Windows Yükleme Dosyası

3.3.1 Dashboard Kurulumu

[Downloads-Windows IoT](#) sayfasında bu seferde **Get Windows 10 IoT Core Dashboard** bağlantısına tıklayarak Dashboard indiriyoruz. Insider ve Dashboard aynı Windows 10 cihazda kurulmalıdır çünkü sadece Windows 10 cihazlarda çalışmaktadır.



Şekil 3.12 IoT Dashboard Penceresi 1

IoT Dashboard

My devices

Set up a new device

Connect to Azure

Try some samples

Sign in

Settings

Set up a new device

First, let's get Windows 10 IoT Core on your device.

Device type
Raspberry Pi 2 & 3

OS Build
Windows Insider Preview

Sign in as a Windows Insider.

Drive
E: 14Gb JFNK TECH USB CARD READER L

Device name
minwingc

New Administrator password
•••••

Confirm Administrator password
•••••

☒ Wi-Fi Network Connection

Select

Only 2.4 GHz WiFi networks that have already been connected to will appear in this list

☒ I accept the software license terms

Download and install

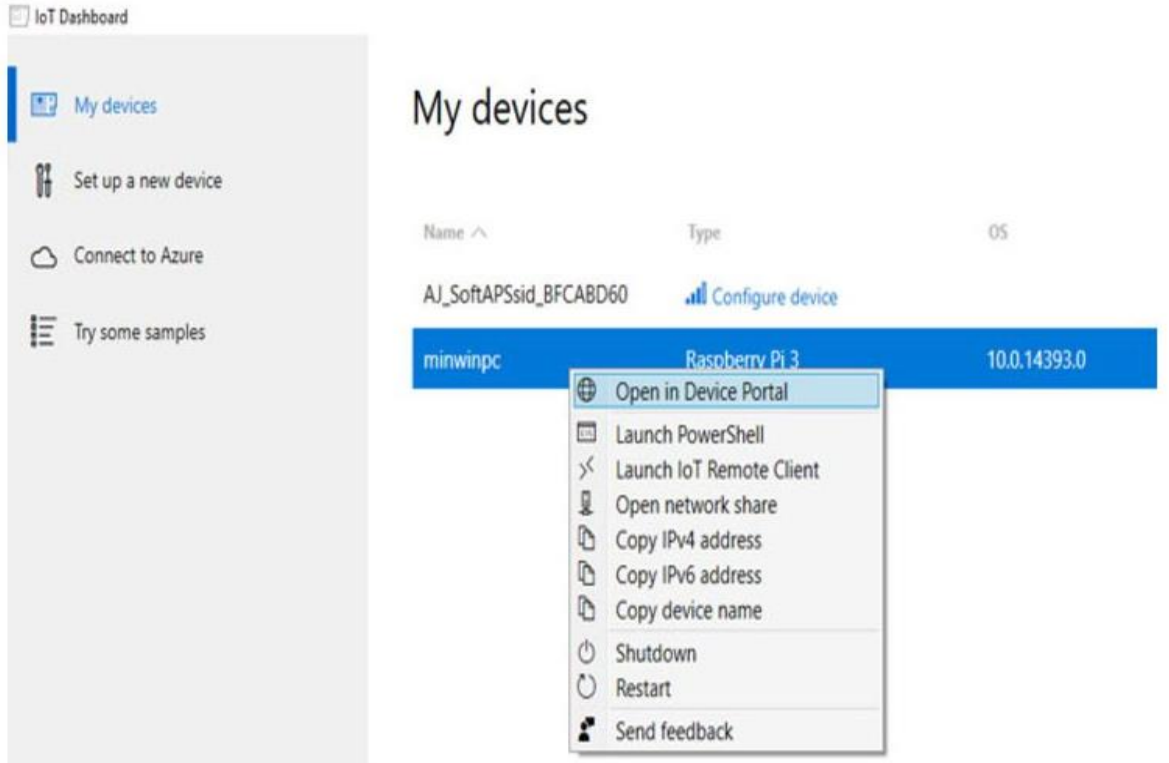
Şekil 3.13 IoT Dashboard Penceresi 2

3.3.2 Windows IoT Kurulumu

Windows 10 işletim sistemi yüklü bir cihaza **Windows 10 Iot Core Dashboard** ve **Windows 10 Insider Preview IoT Core** sürümlerini kurmalıyız.

Yükleme sonrasında **Windows 10 IoT Core Dashboard** açılacaktır. 16 ve üstü **sd kart**ınızı cihaza **Sd Kart Okuyucu** ile takıp çıkan ekrandan **Setup a new device** sekmesinden **Raspberry Pi 2&3**'ü seçip sd kartın konumunu, admin şifremizi seçip internet üzerinden sd karta kurulumu gerçekleştirebiliriz.

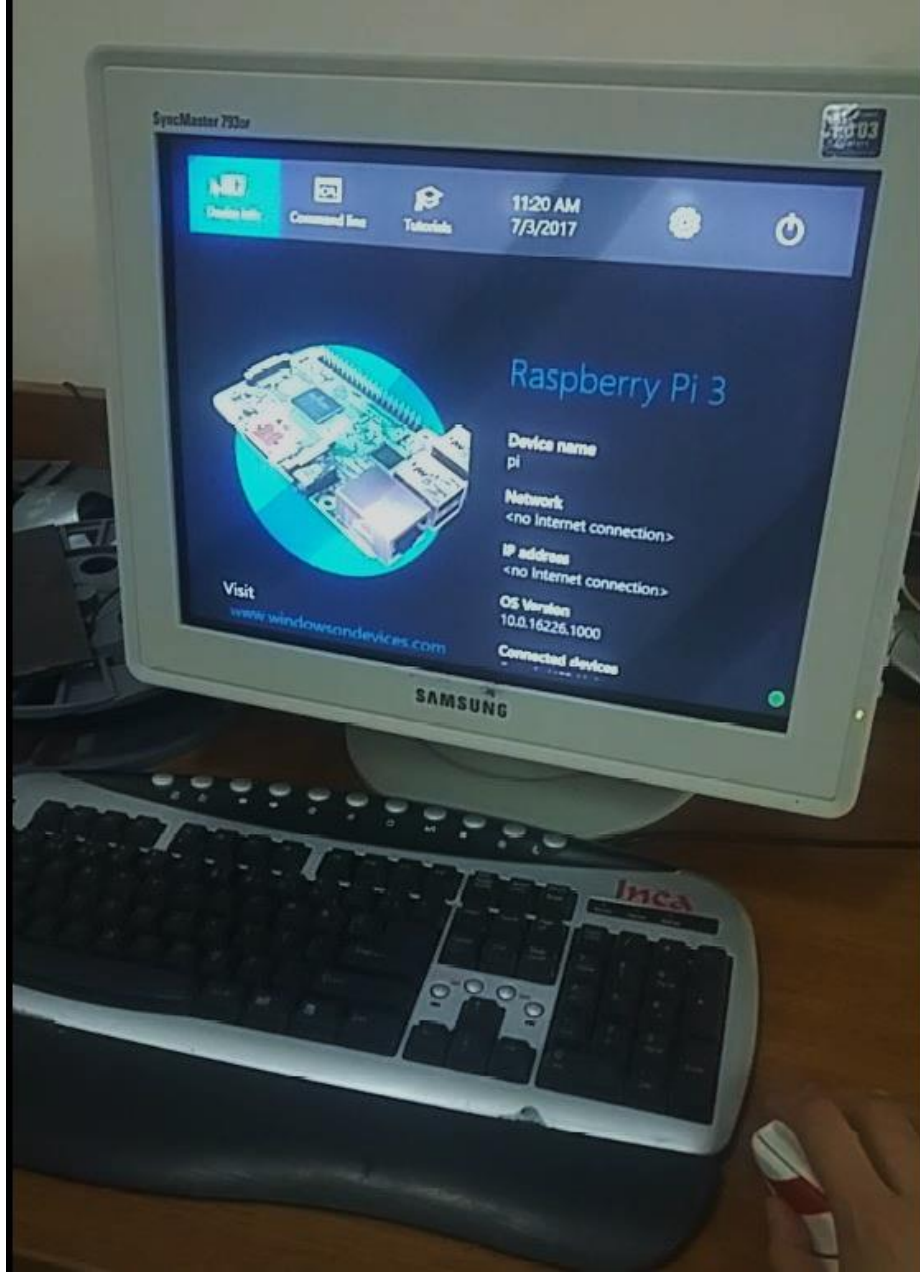
Kurulum tamamlandığında cihaza SD kartı takıp My Devices kısmından Windows 10 cihazımızdan kontrol edebiliriz. Network'e bağlantı sağlamayı unutmamalıyız.



Şekil 3.14 My Device Komut Penceresi

My devices kısmından; cihazı kapatabilir, resetleyebilir, Web Device Portal’a girebilir, IP adresi, hostname kopyalayabilir, Powershell’i açabilir, IoT Remote Client’i çalıştırabiliriz.

Cihazı 8080 portu üzerinden web arayüzüne (Web Device Portal’a erişim sağlayabiliriz.) Birkaç hazır örneği IoT Dashboard üzerinden çalıştırabiliriz.



Şekil 3.15 Raspberry Pi 3 Windows IOT Core Kurulum Sonrası Ekranı

BÖLÜM 4

4.1 Bütçe ve Gerekçesi

Proje kapsamında proje için gerekli malzemeler satın alınmıştır. Satın alınan ürünler ve fiyatları Tablo 4.1’de belirtilmiştir.

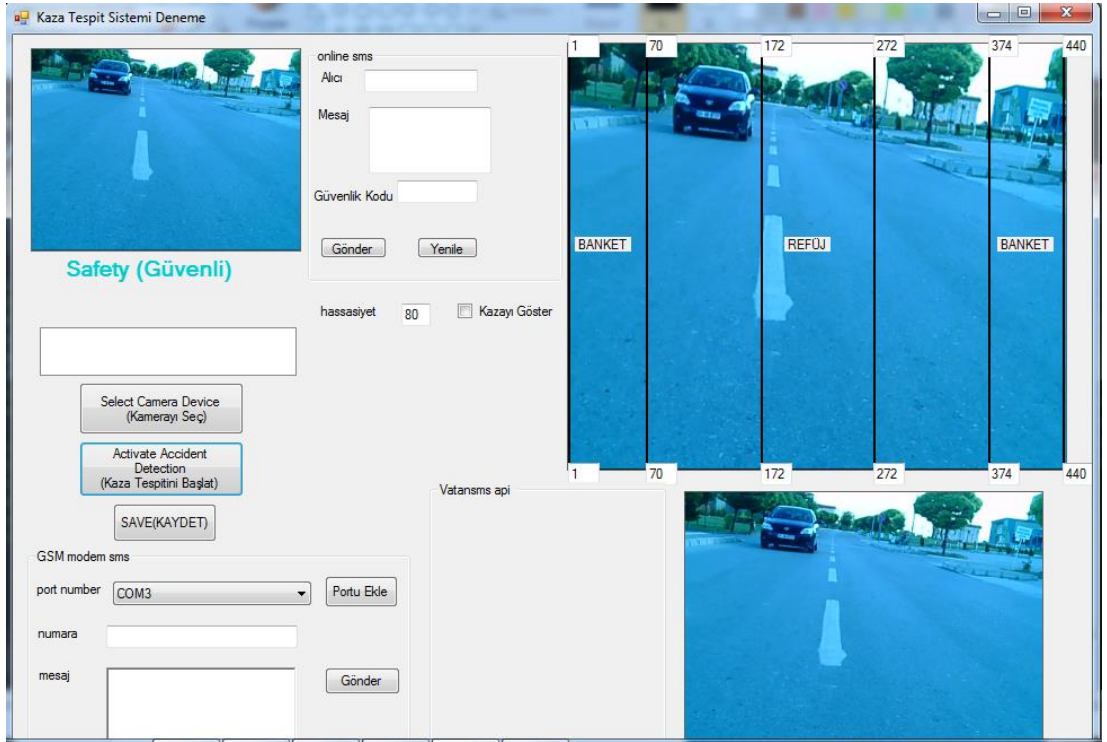
Tablo 4.1 Bütçe Planı

TÜKETİME YÖNELİK MAL VE MALZEME ALIMLARI		
Laboratuvar Malzemesi Alımları	Sarf	Kamera hareket ekseni için 2 adet servo motor 824,86TL, sistem için sarf malzeme 49,99 TL
Bilgisayar Malzemesi Alımları	Sarf	Gece görüşlü dome kamera 1.445,08 TL, Kamera motoru 379,90 TL
Elektrik-Elektronik Malzemesi Alımları	Sarf	Elektronik Kart(ardiuno) 145.72 TL, Raspberry Pi 3: 149.86 TL, Güneş paneli 299.99 TL, Servo motor sürücü kartı 68.51 TL, Jel akü 428.00 TL
Baskı ve Cilt Giderleri		97.95 TL
Fotokopi Giderleri		49.95 TL
Ulaşım Giderleri(Uçak ve yataklı tren hariç)		59.95 TL TL kargo bedeli(internette sipariş verilecek tüm ürünler için)
TOPLAM		3.999,76 TL

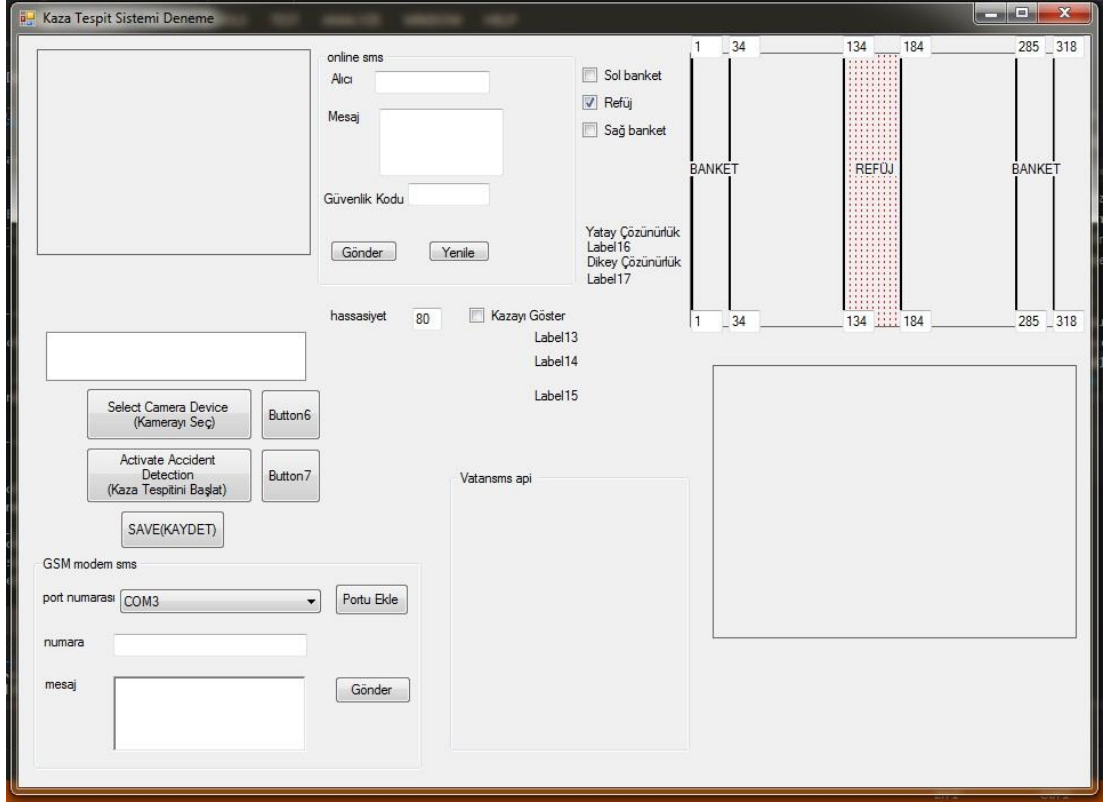
BÖLÜM 5

5.1 Sonuçlar

Projemizde ilk olarak Visual Studio programında Visual Basic dili ile görüntü işleme algoritması yazıldı ve denemeleri başarılı bir şekilde gerçekleştirildi. Daha sonra sistemin güneş paneli ve akü montajı yapılarak proje tamamlandı.



Şekil 5.1 Yazılım Arayüzü



Şekil 5.2 Yazılım Arayüzü 2



Şekil 5.3 Sistemin Güneş Paneli Ve Akü Montajı



Şekil 5.4 Raspberry Pi 3 ve Kızılötesi Kamera

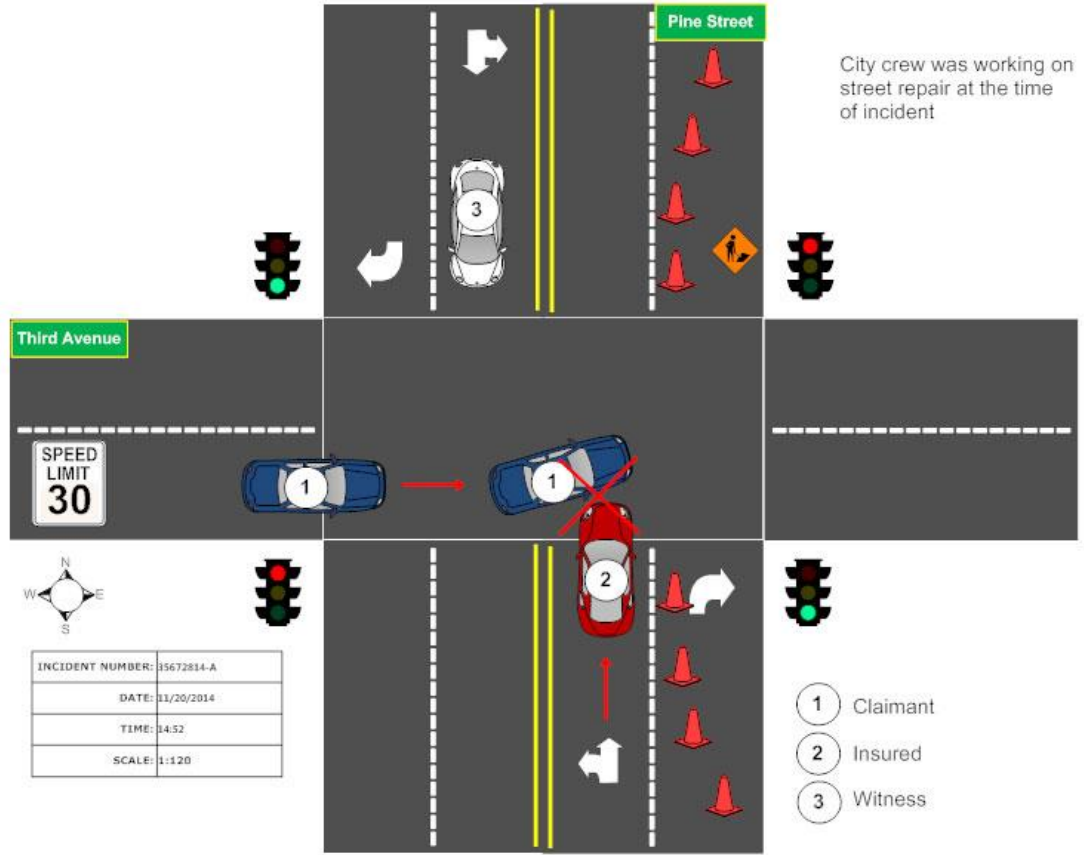
5.2 İleride Yapılması Planlanan Çalışmalar

Bu proje kapsamında hedeflenen iş paketleri başarı ile gerçekleştirilmiştir. Ancak sistemin ticari hale gelebilmesi için kazaların algılama algoritmasının daha hassas hale getirilmesi gerekmektedir. Bu nedenle bu projenin devamı niteliğindeki ileriki çalışmalarda sistemin yazılımsal olarak geliştirilmesi planlanmaktadır.

Elbette her türlü yol durumu için yazılımın adaptif olması gerekmektedir. Bu yol durumlarından birisi Şekil 5.3 [12] de diğeri de Şekil 5.4 [13] de görülmektedir. Sırası ile bu şekiller “S” şeklindeki virajlı bir yolu ve bir kavşağı göstermektedir. Kıvrımlı yollar için nonlinear bir yol çizgisi oluşturulmasına müsaade edecek bir algoritma geliştirilecektir. Kavşaklar için ise yuvarlak şekilli kavşak göbeği için elips şeklinin algoritması oluşturulacaktır.



Şekil 5.3 “S” Şeklindeki Virajlı Bir Yol



Şekil 5.4 Bir Kavşak İçin Kaza Senaryosu

Son olarak ise; kaza algılaması gerçekleştikten sonra sürekli olarak bekleme yapmakta olan unsur(lar)un araç olup olmadıklarının tespiti ve/veya hasar oranının tahmini işlemi algoritmaya katılacaktır.

KAYNAKLAR

- [1]. G. Guido, F. F. Saccomanno, A. Vitale, V. Gallelli, D. Rogano, “A calibration framework of car following models for safety analysis based on vehicle tracking data from smartphone probes”, *International Journal of Mobile Network Design and Innovation (IJMNDI)*, Vol. 5, No. 4, 2014.
- [2]. A. O. Kansız, “Accident recognition using mobile phone”, *Signal Processing and Communications Applications Conference (SIU)*, syf. 1-4, 24-26 Nisan 2013.
- [3]. W. Hu, X. Xiao, D.Xie, T. Tan, S. Maybank , “ Traffic accident prediction using 3D model – based vehicle tracking”, *IEEE Transactions on Vehicular Technology*, Vol. 53 syf 677-694, May 2004.
- [4]. S. Sadeky, A. A. Hamadiy, B. Michaelisy, U. Sayed, “Real – Time Automatic Traffic Accident Recognition Using HFG”, *Pattern Recognition (ICPR)*, 20th International Conference on İstanbul, *Journal of Signal and Information Proccesing*, 2010.
- [5]. Y. Matsushita, K. Ikeuchi, M. Sakauchi, S. Kamijio ,“ Traffic monitoring and accident detection at intersections” *EEE Transactions on Intelligent Transportation Systems* vol.1 page 108-118. *Inst. of Ind. Sci, Tokyo Univ. Japan*, 06/08/2002.
- [6]. J. White, C. Thompson, H. Turner, B. Dougherty, D. C. S. White , “WreckWatch Automatic Traffic Accident Detection and Notification with smartphones ” 16: 285. Doi: 10.1007/s11036-011-0304-8, H. et al. *Mobile Netw Appli.*, 22 March 2011
- [7]. C.T. Calafate, J.C. Cano, P. Manzoni, J. Zaldivar “ Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones”, *Department of Computer Engineering, Universitat Politcnica de Valencia, Camino de Vera S/N, 46022, Spain*, 4-7 oct. 2011.

[8]. P. J.Lagassey, “Advanced automobile accident detection, data recordation and reporting system”, Patent number: US7348895 B2, 25 march 2008.

[9]. Y.K. Ki, D.Y. Lee, “A Traffic Accident Recording and Reporting Model at Intersections”, IEEE Transactions on Intelligent Transportation Systems (Volume:8, Issue: 2 , page 188-194), Dept of Comput. Sci. & Eng, Korea Univ. Seoul, 04/06/2007.

[10]. J.J. Mackey, R. Pandolfi, C.J. Brogan, “Mobile vehicle accident data system” Patent number: US 6141611 A, Phoenix Group, Inc, 31 oct. 2000.

[11]. Kerem İzgöl, “Raspberry Pi 3: İlk İzlenimler”, http://maker.robotistan.com/wp-content/uploads/2016/03/Raspberry_Pi_3_breakdown-696x390.jpg, Yükleme tarihi: 5 Mart 2016 Erişim tarihi: 02.07.2017

[12] Aimelaime at French Wikipedia, Virage en épingle sur le circuit de F1 de Monaco- Wikimedia, https://commons.wikimedia.org/wiki/File:Epingle_Monaco.jpg Yükleme tarihi: 12:28, 3 Mayıs 2009, erişim tarihi: 02.07.2017

[13] Smartdraw, Accident Reconstruction, ©1994-2017 SmartDraw, LLC, <https://www.smartdraw.com/accident-reconstruction/>

EKLER

Ek 1: Microsoft Visual Basic İle Ana Program

```
' ** **
Imports AForge.Video
Imports AForge.Video.DirectShow
Imports System.IO
Imports System.IO.Ports
Imports System.Threading
Imports TextmagicRest
Imports TextmagicRest.Model
Imports System.Runtime.InteropServices
Imports System.Math

Public Class Form1
    Dim yataycarpan As Integer
    ' WebCam Defenitions
    ' Const WM_CAP As Short = &H4005

    'Const WM_CAP_DRIVER_CONNECT As Integer = WM_CAP + 10
    ' Const WM_CAP_DRIVER_DISCONNECT As Integer = WM_CAP + 11
    'Const WM_CAP_EDIT_COPY As Integer = WM_CAP + 30
    'Const WM_CAP_SET_PREVIEW As Integer = WM_CAP + 50
    'Const WM_CAP_SET_PREVIEWRATE As Integer = WM_CAP + 52
    'Const WM_CAP_SET_SCALE As Integer = WM_CAP + 53
    'Const WS_CHILD As Integer = &H40000000
    'Const WS_VISIBLE As Integer = &H10000000
    'Const SWP_NOMOVE As Short = &H25
    'Const SWP_NOSIZE As Short = 1
    'Const SWP_NOZORDER As Short = &H45
    ' Const HWND_BOTTOM As Short = 1
    Dim kamera As VideoCaptureDevice
    Dim bmp As Bitmap
    'Dim iDevice As Integer = 0 'Current device ID
    'Dim hWnd As Integer ' Handle to preview window
    Dim RecTop, RecBottom, RecLeft, RecRight As Integer
    Dim motiondet As Boolean

    Private Sub Captured(sender As Object, EventArgs As NewFrameEventArgs)

        bmp = DirectCast(eventArgs.Frame.Clone(), Bitmap)

        On Error Resume Next
        PictureBox1.Image = DirectCast(eventArgs.Frame.Clone(), Bitmap)
        PB1.Image = DirectCast(eventArgs.Frame.Clone(), Bitmap)
    End Sub

    ' Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    'Label10 = PB1.Image.Size.Height
    ' iDevice = lstDevices.SelectedIndex
    ' displays a list of Devices
    ' LoadDeviceList()

    ' make the first Device as the default selection
    ' lstDevices.SelectedIndex = 0
```

```

'   OpenPreviewWindow()
' End Sub

'   Private Sub OpenPreviewWindow()
'Dim piccapture As PictureBox = PictureBox1
'Dim iHeight As Integer = piccapture.Height
'Dim iWidth As Integer = piccapture.Width

'   Bir önceki görüntüyü pictureboxta açma
'   hWnd = capCreateCaptureWindowA(iDevice, WS_VISIBLE Or WS_CHILD, 0, 0,
640, 480, piccapture.Handle.ToInt32, 0)

'
'   Kamera cihazına bağlanma
'
'   If SendMessage(hWnd, WM_CAP_DRIVER_CONNECT, iDevice, 0) Then
'
'Set the preview scale
'
'       SendMessage(hWnd, WM_CAP_SET_SCALE, True, 0)
'
'   Set the preview rate in milliseconds
'
'       SendMessage(hWnd, WM_CAP_SET_PREVIEWRATE, 66, 0)
'
'Start previewing the image from the camera
'
'       SendMessage(hWnd, WM_CAP_SET_PREVIEW, True, 0)
'
'Resize window to fit in picturebox
'
'       SetWindowPos(hWnd, HWND_BOTTOM, 0, 0, piccapture.Width,
piccapture.Height, SWP_NOMOVE Or SWP_NOZORDER)

'   End If

' End Sub

'Private Sub LoadDeviceList()
'Dim strName As String = Space(100)
'Dim strVer As String = Space(100)
'Dim bReturn As Boolean
'Dim x As Integer = 0

'
'   Load name of all available devices into the lstDevices
'
'   Do
'
'       Get Driver name and version
'
'       bReturn = capGetDriverDescriptionA(x, strName, 100, strVer, 100)
'
'   If there was a device add device name to the list

```

```

'
'      If bReturn Then lstDevices.Items.Add(strName.Trim)
'      x += 1
'      Loop Until bReturn = False
'End Sub

Private Sub SelCamDev_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SelCamDev.Click

    Dim kameras As VideoCaptureDeviceForm = New VideoCaptureDeviceForm

    If kameras.ShowDialog() = Windows.Forms.DialogResult.OK Then
        kamera = kameras.VideoDevice
        AddHandler kamera.NewFrame, New NewFrameEventHandler(AddressOf
Captured)

        kamera.Start()

    End If
    PB1Timer.Enabled = True
    'Label16.Text = PB1.Size.Height
    'Label17.Text = PB1.Size.Width

End Sub

Private Sub ActMotDet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ActMotDet.Click

    PB2.Image = PictureBox1.Image
    DoTimer.Enabled = True

End Sub

Private Sub DoTimer_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles DoTimer.Tick
    Compare_Images()
End Sub
Private Sub MotionLabelTimer_Tick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles MotionLabelTimer.Tick
    Label1.Visible = False
    MotionLabelTimer.Enabled = False

End Sub

Private Sub PB1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles PB1.Paint
    Dim x, y As Integer
    Dim blackPen As New Pen(Color.Black, 2)
    Dim rect As New Rectangle(134, 0, 50, 238)
    Dim yataycarpan As Integer

    ' sol banket kontrolü

```

```

        If yataycarpan = 0 Then yataycarpan = 1
        If CheckBox1.Checked = True Then
            For x = TextBox2.Text * yataycarpan To TextBox12.Text *
yataycarpan Step 5
                For y = 1 To 239 * yataycarpan Step 5
                    e.Graphics.DrawEllipse(Pens.Red, x, y, 1, 1)
                Next y
            Next x

        Else
            End If

        'refüj kontrolü
        If CheckBox2.Checked = True Then

            For x = TextBox1.Text * yataycarpan To TextBox9.Text * yataycarpan
Step 5
                For y = 1 To 239 * yataycarpan Step 5
                    e.Graphics.DrawEllipse(Pens.Red, x, y, 1, 1)
                Next y
            Next x

        Else
            End If

        'sağ banket kontrolü
        If CheckBox3.Checked = True Then

            For x = TextBox10.Text * yataycarpan To TextBox11.Text *
yataycarpan Step 5
                For y = 1 To 239 * yataycarpan Step 5
                    e.Graphics.DrawEllipse(Pens.Red, x, y, 1, 1)
                Next y
            Next x

        Else
            End If

        If TextBox1.Text = "" Then TextBox1.Text = 134
        If TextBox2.Text = "" Then TextBox2.Text = 134
        If TextBox3.Text = "" Then TextBox3.Text = 134
        If TextBox9.Text = "" Then TextBox9.Text = 134
        If TextBox10.Text = "" Then TextBox10.Text = 134
        If TextBox11.Text = "" Then TextBox11.Text = 134
        If TextBox12.Text = "" Then TextBox12.Text = 134
        If TextBox13.Text = "" Then TextBox13.Text = 134
        If TextBox14.Text = "" Then TextBox14.Text = 134
        If TextBox15.Text = "" Then TextBox15.Text = 134
        If TextBox16.Text = "" Then TextBox16.Text = 134
        If TextBox17.Text = "" Then TextBox17.Text = 134

        'e.Graphics.DrawRectangle(blackPen, rect)
        On Error Resume Next
        ' rofuj linelari
        Dim point1 As New Point(TextBox1.Text, 0)

```

```

Dim point2 As New Point(TextBox14.Text, PB1.Height)
Dim point3 As New Point(TextBox9.Text, 0)
Dim point4 As New Point(TextBox15.Text, PB1.Height)
'sağ basket line
Dim point5 As New Point(TextBox2.Text, 0)
Dim point6 As New Point(TextBox3.Text, PB1.Height)
Dim point7 As New Point(TextBox12.Text, 0)
Dim point8 As New Point(TextBox13.Text, PB1.Height)
'sol basket line
Dim point9 As New Point(TextBox10.Text, 0)
Dim point10 As New Point(TextBox16.Text, PB1.Height)
Dim point11 As New Point(TextBox11.Text, 0)
Dim point12 As New Point(TextBox17.Text, PB1.Height)

e.Graphics.DrawLine(blackPen, point1, point2)
e.Graphics.DrawLine(blackPen, point3, point4)
e.Graphics.DrawLine(blackPen, point5, point6)
e.Graphics.DrawLine(blackPen, point7, point8)
e.Graphics.DrawLine(blackPen, point9, point10)
e.Graphics.DrawLine(blackPen, point11, point12)

TextBox1.Location = New Point(PB1.Location.X + Val(TextBox1.Text), 0)
TextBox14.Location = New Point(PB1.Location.X + Val(TextBox14.Text),
PB1.Height)
TextBox9.Location = New Point(PB1.Location.X + Val(TextBox9.Text), 0)
TextBox15.Location = New Point(PB1.Location.X + Val(TextBox15.Text),
PB1.Height)

TextBox2.Location = New Point(PB1.Location.X + Val(TextBox2.Text), 0)
TextBox3.Location = New Point(PB1.Location.X + Val(TextBox3.Text),
PB1.Height)
TextBox12.Location = New Point(PB1.Location.X + Val(TextBox12.Text),
0)
TextBox13.Location = New Point(PB1.Location.X + Val(TextBox13.Text),
PB1.Height)
TextBox10.Location = New Point(PB1.Location.X + Val(TextBox10.Text),
0)
TextBox16.Location = New Point(PB1.Location.X + Val(TextBox16.Text),
PB1.Height)
TextBox11.Location = New Point(PB1.Location.X + Val(TextBox11.Text),
0)
TextBox17.Location = New Point(PB1.Location.X + Val(TextBox17.Text),
PB1.Height)

If Label11.Visible = True And RecYes.CheckState = CheckState.Checked
Then
    e.Graphics.DrawRectangle(Pens.White, RecLeft, RecTop, RecRight -
RecLeft, RecBottom - RecTop)
    e.Graphics.DrawRectangle(Pens.White, RecLeft + 1, RecTop + 1,
RecRight - RecLeft + 1, RecBottom - RecTop + 1)
    e.Graphics.DrawRectangle(Pens.White, RecLeft - 1, RecTop - 1,
RecRight - RecLeft - 1, RecBottom - RecTop - 1)

End If

```

```

End Sub
' ardışık görüntü kıyaslama
Private Sub Compare_Images()

    Dim c1, c2 As System.Drawing.Color
    Dim x, y, sen_delta, basla, bitis As Integer
    Dim dG, dB, dR, c1G, c1B, c1R, c2G, c2B, c2R As Integer
    Dim solbanket, sagbanket, refuj As Boolean

    On Error Resume Next
    RecButtom = 0
    RecRight = 0
    RecTop = PictureBox1.Height
    RecLeft = PictureBox1.Width

    Label16.Text = PB1.Image.Size.Width
    Label17.Text = PB1.Image.Size.Height
    yataycarpan = Int(PB1.Image.Size.Width / 320)

    'sol banket kontrolü
    If CheckBox1.Checked = True Then
        For x = TextBox2.Text * yataycarpan To TextBox12.Text *
yataycarpan Step 5
            For y = 1 To 239 * yataycarpan Step 5

                ' getting pixel color (R,G,B) from both PictureBoxes at
the same X,Y
                ' ** both PictureBoxes must be 319x239 !!!!! **
                c1 = CType(PB1.Image, Bitmap).GetPixel(x, y)
                c2 = CType(PB2.Image, Bitmap).GetPixel(x, y)

                ' calculating colors Delta between two PictureBoxes (at
the same X,Y)
                ' immitating ABS function. for some odd reason using ABS
causes overflow

                'Delta R
                If c1.R >= c2.R Then dR = Abs(c1.R - c2.R)
                'Delta G
                If c1.G >= c2.G Then dG = Abs(c1.G - c2.G)
                ' Delta B
                If c1.B >= c2.B Then dB = Abs(c1.B - c2.B)

                sen_delta = TextBox6.Text ' <-- adjust sensitivity here
(low = more sensitive ; high = less sensitive)

                'check if color Delta between two PictureBoxes (at the
same X,Y) crossed the "threshold"
                If dR > sen_delta Or dG > sen_delta Or dB > sen_delta Then

                    If x < RecLeft Then RecLeft = x
                    If x > RecRight Then RecRight = x
                    If y < RecTop Then RecTop = y
                    If y > RecButtom Then RecButtom = y

                    ' MOTION DETECTED

                    Label1.Visible = True
                    Label2.Visible = False

```

```

        ' this timer is only to show the Label "Motion" for
few milliseconds otherwise it will not show
        MotionLabelTimer.Enabled = True

        'if MOTION DETECTED then match PB2 to PB1
        PB2.Image = PB1.Image
        PB2.Image.Save(".\refuj kazası.jpg")

        ' there is no reason to continue checking all the rest
of the pixels for "change"
        ' its enough that one pixel is changed inorder to
declare Motion "Detection"
        Exit Sub

    Else
        ' MOTION NOT DETECTED
        Label1.Visible = False
        Label2.Visible = True

        NoMotionTimer.Enabled = True

    End If

Next y
Next x
Else
End If

'refuj kontrolü
If CheckBox2.Checked = True Then

    For x = TextBox1.Text * yataycarpan To TextBox9.Text * yataycarpan
Step 5
        For y = 1 To 239 * yataycarpan Step 5

            ' getting pixel color (R,G,B) from both PictureBoxes at
the same X,Y
            ' ** both PictureBoxes must be 319x239 !!!!! **
            c1 = CType(PB1.Image, Bitmap).GetPixel(x, y)
            c2 = CType(PB2.Image, Bitmap).GetPixel(x, y)

            ' calculating colors Delta between two PictureBoxes (at
the same X,Y)
            ' immitating ABS function. for some odd reason using ABS
causes overflow

            'Delta R
            If c1.R >= c2.R Then dR = Abs(c1.R - c2.R)
            'Delta G
            If c1.G >= c2.G Then dR = Abs(c1.G - c2.G)
            ' Delta B
            If c1.B >= c2.B Then dR = Abs(c1.B - c2.B)

            sen_delta = TextBox6.Text ' <-- adjust sensitivity here
(low = more sensitive ; high = less sensitive)

```

```

        'check if color Delta between two PictureBoxes (at the
        same X,Y) crossed the "threshold"
        If dR > sen_delta Or dG > sen_delta Or dB > sen_delta Then

            If x < RecLeft Then RecLeft = x
            If x > RecRight Then RecRight = x
            If y < RecTop Then RecTop = y
            If y > RecBottom Then RecBottom = y

            ' MOTION DETECTED

            Label1.Visible = True
            Label2.Visible = False

            ' this timer is only to show the Label "Motion" for
            few milliseconds otherwise it will not show
            MotionLabelTimer.Enabled = True

            'if MOTION DETECTED then match PB2 to PB1
            PB2.Image = PB1.Image
            PB2.Image.Save(".\refuj kazası.jpg")

            ' there is no reason to continue checking all the rest
            of the pixels for "change"
            ' its enough that one pixel is changed inorder to
            declare Motion "Detection"
            Exit Sub

        Else
            ' MOTION NOT DETECTED
            Label1.Visible = False
            Label2.Visible = True

            NoMotionTimer.Enabled = True

        End If

    Next y
Next x

Else
End If

'sağ banket kontrolü
If CheckBox3.Checked = True Then

    For x = TextBox10.Text * yataycarpan To TextBox11.Text *
yataycarpan Step 5
        For y = 1 To 239 * yataycarpan Step 5

            ' getting pixel color (R,G,B) from both PictureBoxes at
            the same X,Y

            ' ** both PictureBoxes must be 319x239 !!!!! **
            c1 = CType(PB1.Image, Bitmap).GetPixel(x, y)
            c2 = CType(PB2.Image, Bitmap).GetPixel(x, y)

```

```

' calculating colors Delta between two PictureBoxes (at
the same X,Y)
' immitating ABS function. for some odd reason using ABS
causes overflow

'Delta R
If c1.R >= c2.R Then dR = Abs(c1.R - c2.R)
'Delta G
If c1.G >= c2.G Then dR = Abs(c1.G - c2.G)
' Delta B
If c1.B >= c2.B Then dR = Abs(c1.B - c2.B)

sen_delta = TextBox6.Text ' <-- adjust sensitivity here
(low = more sensitive ; high = less sensitive)

'check if color Delta between two PictureBoxes (at the
same X,Y) crossed the "threshold"
If dR > sen_delta Or dG > sen_delta Or dB > sen_delta Then

    If x < RecLeft Then RecLeft = x
    If x > RecRight Then RecRight = x
    If y < RecTop Then RecTop = y
    If y > RecButtom Then RecButtom = y

    ' MOTION DETECTED

    Label1.Visible = True
    Label2.Visible = False

    ' this timer is only to show the Label "Motion" for
few milliseconds otherwise it will not show
    MotionLabelTimer.Enabled = True

    'if MOTION DETECTED then match PB2 to PB1
    PB2.Image = PB1.Image
    PB2.Image.Save(".\refuj kazası.jpg")

    ' there is no reason to continue checking all the rest
of the pixels for "change"
    ' its enough that one pixel is changed inorder to
declare Motion "Detection"
    Exit Sub

Else
    ' MOTION NOT DETECTED
    Label1.Visible = False
    Label2.Visible = True

    NoMotionTimer.Enabled = True

End If

Next y
Next x
Else
End If

PB2.Image = PictureBox1.Image
End Sub

```

```

    Private Sub NoMotionTimer_Tick(sender As Object, e As EventArgs) Handles
NoMotionTimer.Tick
        Label2.Visible = False
        NoMotionTimer.Enabled = False

    End Sub

    Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox3_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox12_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox13_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox14_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox15_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox16_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub
    Private Sub TextBox9_KeyPress(sender As Object, e As KeyPressEventArgs)
        If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
            e.Handled = True
        End If
    End Sub

```

```

End Sub
Private Sub TextBox10_KeyPress(sender As Object, e As KeyPressEventArgs)
    If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
        e.Handled = True
    End If
End Sub
Private Sub TextBox11_KeyPress(sender As Object, e As KeyPressEventArgs)
    If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
        e.Handled = True
    End If
End Sub
Private Sub TextBox17_KeyPress(sender As Object, e As KeyPressEventArgs)
    If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
        e.Handled = True
    End If
End Sub
Private Sub TextBox4_KeyPress(sender As Object, e As KeyPressEventArgs)
    If Not (Char.IsNumber(e.KeyChar) = True) And e.KeyChar <>
ChrW(Keys.Back) Then
        e.Handled = True
    End If
End Sub
' fotoğrafı kullanıcıya kaydetme
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    '
    ' Copy image to clipboard
    '

    '
    ' Get image from clipboard and convert it to a bitmap
    '

    PB2.Image = PB1.Image

    Button1.Enabled = False

    If SaveFileDialog1.ShowDialog = DialogResult.OK Then
        PB2.Image.Save(SaveFileDialog1.FileName, Imaging.ImageFormat.Bmp)
    End If

End Sub
' online sms atma
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
    Dim alici As HtmlElement =
WebBrowser1.Document.GetElementById("number")
    Dim mesaj As HtmlElement =
WebBrowser1.Document.GetElementById("message")
    Dim guvenlik kodu As HtmlElement =
WebBrowser1.Document.GetElementById("recaptcha_response_field")
    Dim gonder As HtmlElement =
WebBrowser1.Document.GetElementById("submit")

```

```

        ' Dim guvenlik As HtmlElement =
WebBrowser1.Document.GetElementById("recaptcha_challenge_image")

        alici.SetAttribute("value", TextBox4.Text)
        mesaj.SetAttribute("value", TextBox5.Text)
        guvenlikodu.SetAttribute("value", TextBox7.Text)
        gonder.InvokeMember("click")
        ' guvenlik.InvokeMember("Image", PictureBox2.Image)
End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles
Button3.Click
    WebBrowser1.Navigate("http://smsfree4all.com/free-text-turkey.php")
End Sub

Private Sub WebBrowser1_DocumentCompleted(sender As Object, e As
WebBrowserDocumentCompletedEventArgs) Handles WebBrowser1.DocumentCompleted
    On Error Resume Next
    Dim nesne As Object

    nesne = WebBrowser1.Document.DomDocument.body.createControlRange()
    nesne.add(WebBrowser1.Document.DomDocument.Image(2))
    nesne.exeCommand("Copy")
    PictureBox2.Image = My.Computer.Clipboard.GetImage

End Sub
' GSM modem ile sms atma
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    Try
        Dim ports As String() = SerialPort.GetPortNames
        Dim port As String
        For Each port In ports
            ComboBox1.Items.Add(port)
        Next port
        ComboBox1.SelectedIndex = 0

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles
Button4.Click
    Try
        With SerialPort1
            .PortName = ComboBox1.Text
            .BaudRate = 115200
            .Parity = Parity.None
            .StopBits = StopBits.One
            .DataBits = 8
            .Handshake = Handshake.RequestToSend
            .DtrEnable = True
            .RtsEnable = True
            .NewLine = vbCrLf
            .Open()
        End With
    End Try

```

```

        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles
Button5.Click
        Try
            If SerialPort1.IsOpen Then
                With SerialPort1
                    .Write("AT" & vbCrLf)
                    .Write("AT+ CMGF = 1 " & vbCrLf)
                    .Write("AT + CMGS = " & Chr(34) & TextBox8.Text & Chr(34)
& vbCrLf)
                    .Write(RichTextBox1.Text & Chr(26))
                    MsgBox("başarılı ")
                End With
            Else
                MsgBox("başarısız")
            End If
        Catch ex As Exception

        End Try
    End Sub
    ' textmagic api ile sms atma
    Public Function SendMessage()
        Dim client = New Client("test", "my-api-key")
        Dim link = client.SendMessage("Hello from TextMagic API",
"054155247689")
        If link.Success Then
            Console.WriteLine("Message with ID {0} has been successfully
sent", link.Id)
        Else
            Console.WriteLine("Message was not sent due to following
exception: " + link.ClientException.Message)
        End If
    End Function

End Class

```

ÖZGEÇMİŞ

1

Kimlik Bilgileri

Adı Soyadı: Yasemin YILDIZ

Baba Adı: Hüseyin

Anne Adı: Mühibe

Doğum Yeri: Adana

Doğum Yılı: 12.08.1994



Orta öğrenimini Mersin’de tamamladıktan sonra 2013 yılında Niğde Niğde Ömer Halisdemir Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümünde lisans öğrenimine başladı. 2017 yılında lisans öğrenimine devam etmektedir.

Haberleşme Bilgileri

E-posta: lostrss@gmail.com

Kimlik Bilgileri

Adı Soyadı: Meltem Muhsuroğlu

Baba Adı: Ahmet

Anne Adı: Semiha

Doğum Yeri: Kayseri

Doğum Yılı: 01.04.1995



Orta öğrenimini Kayseri’de tamamladıktan sonra 2013 yılında Niğde Ömer Halisdemir Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümünde lisans öğrenimine başladı 2017 yılında lisans öğrenimine devam etmektedir.

Haberleşme Bilgileri

E-posta: meltemmuhsuroglu00@gmail.com

