

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TRAFİK OTOMASYONUNDA GÖRÜNTÜ TABANLI
ARAÇ TESPİT SİSTEMİNİN GELİŞTİRİLMESİ VE FPGA
İLE GERÇEKLENMESİ**

Elektronik ve Haberleşme Müh. Ferhat YALDIZ

**FBE Elektrik Müh. Anabilim Dalı Kontrol ve Otomasyon Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Doç. Dr. Abdullah BAL

İSTANBUL, 2010

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ÇİZELGE LİSTESİ	ix
ÖNSÖZ.....	x
ÖZET.....	xi
1. GİRİŞ.....	1
2. TRAFİK KONTROL SİSTEMLERİ.....	3
2.1 Zaman Ayarlı Kontrol Sistemi	3
2.2 Manyetik Algılayıcı Tabanlı Kontrol Sistemi	4
2.3 Ultrasonik veya Lazer Algılayıcı Tabanlı Kontrol Sistemi	4
2.4 Görüntü İşleme Tabanlı Kontrol Sistemi.....	6
3. BİLGİSAYAR ORTAMINDA ARAÇ TESPİT SİSTEMİ.....	10
3.1 Görüntünün RGB'den Gri Seviyeye Dönüşümü	12
3.2 Yol Maskelenmesi	13
3.3 Arka Plan Öğrenilmesi	14
3.4 Eşikleme (Thresholding)	17
3.5 Morfolojik Kapama (Closing) İşlemi	20
3.6 Medyan Filtreleme.....	25
3.7 Tanımlanmış Bölgelerde Araç Tespitinin Yapılması	28
3.7.1 Araç Tespiti Bölgelerinin Tanımlanması ve Başarımı	28
4. GÖMÜLÜ SİSTEMİN DONANIMSAL BİLEŞENLERİ.....	35
4.1 FPGA	35
4.2 FPGA Geliştirme Kartı	37
4.2.1 Xilinx Spartan 3E FPGA	38
4.3 Görüntü Sensörü	40
4.3.1 Görüntü sensörü kartı	40
5. GÖMÜLÜ SİSTEMİN GERÇEKLENMESİ.....	42
5.1 FPGA Tabanlı Gömülü Sistem Geliştirme	42
5.1.1 HDL (Donanım Tasarım Dili) Tabanlı Tasarım Geliştirme	42
5.1.1.1 VHDL Donanım Tanımlama Dili	43
5.1.2 İşlemci Tabanlı Tasarım Geliştirme	44
5.2 FPGA Tabanlı Araç Tespit Sistemi	47
5.2.1 Microblaze Tabanlı Sistem Geliştirme	47

5.3	PC Programı	49
6.	SONUÇLAR VE ÖNERİLER.....	54
	KAYNAKLAR	55
	ÖZGEÇMİŞ.....	57

SİMGE LİSTESİ

$B(p)$	Arkaplan öğrenimi sonucu
$D(k, p)$	Güncel görüntünün arkaplanının çıkarılması
k	Görüntü karesi sayısı
$M(p)$	Yol maskelenmesinde kullanılacak ikili görüntü
n	Arkaplan öğrenimine giren görüntü karesi sayısı

KISALTIMA LİSTESİ

ADC	Analog to Digital Converter
CAD	Computer Aided Design
CCD	Charge-Coupled Device
CLB	Configuration Logic Block
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Devices
DAC	Digital to Analog Converter
DDRSDRAM	Double-Data Rate Synchronous Dynamic RAM
FPGA	Field Programmable Gate Array
Fps	Frame per second
HDL	Hardware Description Language
I2C	Inter-integrated Circuit
JTAG	Joint Test Action Group
IEEE	Institute of Electrical and Electronics Engineers
LCD	Liquid Crystal Display
LUT	Look-up-table
MPMC	Multi Port Memory Controller
MUX	Multiplexer
PC	Personal Computer
PCB	Printed Circuit Board
PLL	Phase-locked loop
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RGB	Red Green Blue
ROI	Region of Interest
SCCB	Serial Camera Control Bus
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver Transmitter
VGA	Video Graphics Array
VHDL	VHSIC Hardware Description Language
VHSIC	Very High-Speed Integrated Circuit

ŞEKİL LİSTESİ

Şekil 2.1	Trafik ışıklarının bir kavşağa yerleştirilmesinin temsili gösterimi.	3
Şekil 2.2	Manyetik tabanlı kontrol sisteminin çalışması.....	4
Şekil 2.3	Ultrasonik sensörler ile araç tespiti. (a) Ultrasonik sensörün yerleşimi ve araç var iken cevabı. (b) Araç yok iken cevabı.	5
Şekil 2.4	Lazer algılayıcı birimi.	6
Şekil 2.5	Ultrasonik veya lazer algılayıcıların kavşak araç tespiti ve trafik araç yoğunluğunu belirleme amacı ardışıl olarak ile yerleşimi.....	6
Şekil 2.6	Kamera tabanlı araç tesbit sistemi (Phuket, Tayland).....	7
Şekil 2.7	Kameranın trafik ışıklarının yanına yerleştirilmesi (Teksas, ABD).....	8
Şekil 2.8	Kameranın farklı bir şekilde yerleştirilmesi (Missouri, ABD).....	8
Şekil 2.9	Kavşak araç tespit sisteminde araç tanıma bölgelerinin yazılımda tanımlanmış hali.9	
Şekil 3.1	Analizi yapılacak olan video verisinin ardışıl görüntüleri.	10
Şekil 3.2	Araç tespiti için görüntü işleme tekniklerinin adımlarının gösterilmesi.	11
Şekil 3.3	RGB renk uzayının gösterilmesi.	12
Şekil 3.4	Video görüntüsünün RGB uzayından gri seviyeye dönüştürülmüş hali.	13
Şekil 3.5	Yol maskelenmesi için resim işleme programında hazırlanan M(p).....	13
Şekil 3.6	Yol maskelenmesi işlemi yapıldıktan sonraki görüntülerden bazıları.	13
Şekil 3.7	Yol maskelenmesi işleminin kamera görüş açısı otoyolun iki yönünü de alması durumunda istenilen yön için yapılması. (Bhatti., 2006)	14
Şekil 3.8	Arkaplan çıkarımının gerçekleştirilmesi, (a) $n = 5$, (b) $n = 10$, (c) $n = 15$, (d) $n = 20$, (e) $n = 50$, (f) $n = 100$	15
Şekil 3.9	Arkaplan çıkarımından sonra hesaplanacak olan $D(k,p)$ 'nin her bir görüntü için bulunması, (a) $k = 10$, (b) $k = 20$, (c) $k = 30$, (d) $k = 40$, (e) $k = 50$, (f) $k = 70$	16
Şekil 3.10	Bazı görüntülere ait histogram bilgileri, (a) koyu görüntü, (b) açık görüntü.....	17
Şekil 3.11	Eşikleme değeri 30 için görüntüler, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.18	
Şekil 3.12	Eşikleme değeri 50 için görüntüler, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.19	
Şekil 3.13	Eşikleme değeri 100 için görüntüler, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.19	
Şekil 3.14	Parmak izi görüntüsüne morfolojik işlemlerin uygulanması, (a) orijinal görüntü, (b) yapısal işlem elemanı, (c) daraltma, (d) açma, (e) açma sonucundaki görüntünün genişletilmesi, (f) açma işlemi sonucundaki görüntünün kapatılması.....	21
Şekil 3.15	Kapatma (Closing) işleminin şablonları.....	22
Şekil 3.16	Morfolojik kapama işlemi sonuçları, eşikleme değeri = 30, (a) yapısal eleman,	

(b) k = 20, (c) k = 30, (d) k = 50, (e) k = 70.....	23
Şekil 3.17 Morfolojik kapama işlemi sonuçları, eşikleme değeri = 30, (a) yapısal eleman, (b) k = 20, (c) k = 30, (d) k = 50, (e) k = 70.....	24
Şekil 3.18 Morfolojik kapama işlemi sonuçları , eşikleme değeri = 30, (a) yapısal eleman, (b) k = 20, (c) k = 30, (d) k = 50, (e) k = 70.....	25
Şekil 3.19 Medyan işleminin tuz ve biber (salt and papper) gürültüsüne sahip bir görüntüye uygulanması.....	26
Şekil 3.20 Detayları anlaşılamayan bir PCB görüntüsüne medyan filtrenin uygulanması ile detaylarının belirgin hale getirilmesi.....	26
Şekil 3.20 3x3 medyan filtre sonuçları, (a) k = 20, (b) k = 30, (c) k = 50, (d) k = 70.....	27
Şekil 3.21 5x5 medyan filtre sonuçları, (a) k = 20, (b) k = 30, (c) k = 50, (d) k = 70.....	27
Şekil 3.22 Araç tespit bölgelerinin tanımlanması.....	28
Şekil 3.23 Video verisi üzerinde tanımlanmış araç tespit bölgelerlerinden araç geçişi, (a)k = 17, (b) k = 31, (c) k = 36, (d) k = 46, (e) k = 64, (f) k = 69, (g) k = 72, (h) k = 87, (i) k = 95, (j) k = 110.	30
Şekil 3.24 Eşik yüzdesinin %15 olduğunda tanımlanan araç tespit bölgelerlerinden araç geçişi, (a)k = 17, (b) k = 31, (c) k = 36, (d) k = 46, (e) k = 64, (f) k = 69, (g) k = 72, (h) k = 87, (i) k = 95, (j) k = 110.	33
Şekil 3.25 Trafik sinyalizasyon süresinin ayarlanması için yoğunluk tespit bölgeleri, (a) gündüz, (b) gece durumu için araç ışıklarının algılanması. (Bhatti, 2006).....	34
Şekil 4.1 FPGA yapısı.	36
Şekil 4.2 FPGA lojik hücrelerin gerçekleşme mimarileri.....	36
Şekil 4.3 Digilent Spartan 3E geliştirme kartı görünüşü.....	37
Şekil 4.4 Digilent Spartan 3E geliştirme kartı blok diagramı.....	38
Şekil 4.5 Spartan 3E FPGA'in CLB ve slice yapısı, (a) genel, (b) detaylı.....	39
Şekil 4.6 Spartan 3E FPGA'in SLICEMve SLICEL yapısı.	39
Şekil 4.7 Görüntü sensör donanımı, (a) sensör kartı, (b) temsili çizim.....	40
Şekil 4.8 OV6630 görüntü sensörü çalışma blok çizimi.	41
Şekil 5.1 FPGA temelli uygula geliştirmenin safhaları. (Aydın, 2005).....	42
Şekil 5.2 Microblaze işlemcisinin detayları.	45
Şekil 5.3 Microblaze işlemcisi ile sistem kurma.	46
Şekil 5.4 Microblaze işlemcisi ile kurulan sistem.	47
Şekil 5.5 Microblaze işlemcisi ile kurulan sistemin birbirleri ile oluşturulan bağlantıları.....	48

Şekil 5.6	Sistemin adres haritası.....	48
Şekil 5.7	Sistemin blok diagram olarak raporunun gösterilmesi.....	49
Şekil 5.8	PC programının görünüşü.	50
Şekil 5.9	PC programının ilk ayarlarının yapılmış durumu.	51
Şekil 5.10	Görüntülerin gönderilmesi.	51
Şekil 5.11	Arkaplanın hesaplanıp PC programına gönderilmesi.....	52
Şekil 5.12	Araç belirlenmiş alanlarda mevcutken PC programı görüntüsü.	53
Şekil 5.13	Araç belirlenmiş alanlarda mevcut değilken PC programı görüntüsü.	53

ÇİZELGE LİSTESİ

Çizelge 3.1 Araç tespiti için tanımlanmış ilgili bölgelere ilişkil bilgiler.....	29
Çizelge 3.2 Araç tespiti için tanımlanmış ilgili bölgelere ilişkil bilgiler. (Eşik yüzdesinin %10 olması durumu.).....	31
Çizelge 3.3 Araç tespiti için tanımlanmış ilgili bölgelere ilişkil bilgiler. (Eşik yüzdesinin %15 olması durumu.).....	31
Çizelge 3.4 Araç tespiti için tanımlanmış ilgili bölgelere ilişkil bilgiler. (Eşik yüzdesinin %20 olması durumu.).....	31
Çizelge 3.5 Araç tespiti için tanımlanmış ilgili bölgelere ilişkil bilgiler. (Eşik yüzdesinin %25 olması durumu.).....	32
Çizelge 3.6 Araç tespiti için tanımlanmış ilgili bölgelere ilişkil bilgiler. (Eşik yüzdesinin %40 olması durumu.).....	32
Çizelge 5.1 Microblaze işlemcisi ile kurulan sistemin detaylı açıklaması.....	48

ÖNSÖZ

Bu tezin hazırlanmasındaki katkılarından dolayı sayın Doç. Dr. Abdullah BAL'a, TÜBİTAK - UEKAE çalışanlarından sayın Dr. Mehmet SEZGİN'e ve bana her konuda destek olan eşime teşekkür ederim.

ÖZET

Trafik otomasyon sistemleri çok hızlı bir şekilde gelişmektedir. Artan şehirleşme neticesinde otoyollar, kavşaklarla birlikte her gün trafiğe çıkan araç sayısı artmaktadır. Kavşak ve yolların kontrolü gittikçe daha zorlaşmakta ve akıllı sistemlere ihtiyaç duyulmaktadır. Önceleri zaman ayarlı kavşak ışıklı sistemleri yeterli iken günümüzde artan trafik akışı neticesinde bu şekildeki sistemler yetersiz kalmaktadır.

Akıllı kavşak ve otoyol sistemlerinde algılayıcı tabanlı akıllı sistemler öne çıkmaktadır. Bu algılayıcı sistemler arasında manyetik, ultrasonik ve görüntü tabanlı algılayıcılar sayılabilir. Görüntü işleme teknolojisinin taban alındığı çözümler işlevselliği, kolay kurulumu, idamesinin basitliği, verimliliği ve uzun ömürlülüğüyle avantajlıdır. Gerçek zamanlı olarak kamera vasıtasıyla video görüntülerin alınması ve bu görüntülerin işlenmesi neticesinde o andaki yoğunluğa bağlı olarak kavşak ve otoyoldaki trafik akışının yönlendirilmesi en doğru biçimde gerçekleştirilebilir. Günümüzde gömülü sistemler genel olarak mikroişlemcili, DSP'li ve FPGA tabanlı olarak ayrılmaktadır. Gerçek zamanlı görüntü işleme veri akışı ve veri işleme yoğunluğu çok yüksek olduğundan işlemlerin paralel olarak donanımsal seviyede yapılması vazgeçilmezdir. Yani FPGA tabanlı görüntü işleme sistemleri günümüzde askeri ve sivil uygulamalarda ön plandadır. Diğer sistemler veri akışı ve işlem yoğunluğunun altında kalmaktadır.

Bu çalışmada gerçek zamanlı video görüntüsündeki araç tespiti yapılmaya çalışılmıştır. Öncelikle kaydedilmiş trafik video verisi üzerinde çalışılmış ve MATLAB'de araç tespiti için gerekli adımlar belirlenmiştir. Bu çalışmadan sonra video görüntülerinin aktarımı ve görüntü işleme için belleğe yazılma için gerekli olan yazılımlar FPGA için yapılmıştır. Daha sonra gerekli araç takibi için adımları FPGA ortamına aktarılacaktır.

ABSTRACT

Traffic automation systems are increasingly deployed in today's world. As a result of increased urbanization highways, intersections are proliferating, and the number of vehicles in the traffic are continuously increasing.

As a consequence of increased number of highways and intersections, the control of those is getting harder and it necessitates intelligent control systems. Even though the utilization of time-adjustable intersection light systems was sufficient for earlier traffic conditions, they are no longer adequate for today's requirements.

Sensor based intelligent systems are prevailing amongst all intelligent intersection and highway control systems. Magnetic, ultrasonic and image based sensors can be listed in those sensor-based control systems. The solutions that employ image processing technologies are more advantageous over others thanks to their better functionality, ease of deployment, ease of maintenance, efficiency and longer life-time. By means of real-time video images fed through camera and by processing those real-time video images traffic flow at intersections and on highways can be directed based on instantaneous traffic load. Nowadays, embedded systems can be generally classified as DSP-based and FPGA-based. In real-time image processing systems data flow and data processing jobs are very frequent, thus, it's indispensable to process those events in parallel at hardware-level. In other words, nowadays FPGA-based image processing systems are prevailing in military and civil applications. Other systems are under the intensity of data flow and processing jobs.

In this study, we investigated the detection of vehicles in real-time video images. First, we worked on recorded traffic video data and determined the necessary steps involved for vehicle detection in MATLAB. Then, we developed the embedded software for FPGA to collect video data and writing this data onto the memory for later video processing. We later plan to port the necessary steps for vehicle detection to FPGA.

1. GİRİŞ

Otomasyon sistemleri günümüzün vazgeçilmez sistemleri arasında yer almakta olup çok hızlı ilerlemeler göstermektedir. Gelecekte de en fazla önemli olacak sistemlerin başında gelmektedir. Trafik kontrol sistemlerinin ana amacı trafik akışının düzenlenmesi olup otomasyon sistemlerindeki bu ilerlemelerden yararlanmakta ve çok daha hızlı ve doğru karar veren otonom sistemler haline gelmektedir. Kavşak ve yan yol birleşimli otoyoldaki trafik akışını düzenlemek en önemli sorunların başında gelmektedir. Trafik ışıklı önceden zaman ayarlı kontrol sistemleri günümüzdeki trafik akışının kontrolünde yetersiz kalmakta, ultrasonik ve manyetik sensörlü sistemler ise pahalılığı, kurulum ve bakım zorluğu gibi nedenlerden dolayı mühendisleri yeni arayışlara itmektedir. Görüntü işleme tabanlı otomasyon sistemleri endüstrinin her alanında uygulanmakta olup, en akıllı sistemler arasında yer almaktadır. Tekstil endüstrisinden, tarım endüstrisine, biyoloji alanından, imalat sektörüne, askeri sistemlerden, plaka ve yüz tanımaya kadar geniş alanlarda kullanılmaktadır. Kamera tabanlı trafik kontrol sistemleri gelecekte adından daha çok söz ettirecek olup vazgeçilmez bir hale gelecektir.

Araç tespit sistemi için ultrasonik algılayıcıların ardışıl olarak belirli aralıklarda kavşağa yaklaşma yerlerinde kullanılmasının gerektiğini ve bu algılayıcılardan toplanan verilerin kavşak kontrol bilgisayarına gönderilmesini önerilmektedir (Sakakira vd., 2005). Ayrıca bu sistemin görüntü işleme tabanlı trafik kontrol sistemine göre karşılaştırılması bu çalışmada yapılmış olup maliyet, kurulum ve bakım kolaylıkları açısından dezavantajlı olduğunu belirtmişlerdir.

Gömülü sistem tabanlı çözümlerin performansı yüksek, uyarlanabilir, maliyeti düşük ve az güç tüketen sistemler olduğundan dolayı araç tespit sistemlerinde PC tabanlı çözümler yerine kullanılması gerektiği belirtilir (Bramberger vd.).

Görüntü tabanlı trafik kontrol sistemlerinde araştırmalar, araç tespiti, araç yoğunluk tespiti ve sınıflandırma, gece, sis ve yağmur gibi durumlardaki başarımlarını artırma, kavşak kontrol sistemlerinin yeteneğinin görüntü tabanlı sistemler ile birleşimi ve plaka tanıma şeklinde özetlenebilir. Bu araştırmalarda çeşitli yöntemler uygulanmakta olup, bu yöntemlerin üzerinde çalışacağı platformların tasarımı da görüntü işlemenin zor olduğundan dolayı diğer araştırma konularındandır.

Bu çalışmada öncelikle trafik kontrol sistemleri incelenmiş olup avantajları ve dezavantajları belirtilmiştir. Daha sonra görüntü işleme sistemlerine değinilmekte olup, sonraki bölümde

kaydedilmiş video verisi üzerinde bilgisayar ortamında MATLAB yazılımı kullanılarak araç tespiti yapılmaya çalışılmış olup ilgili adımlar belirlenmiştir. Daha sonra gerçek zamanlı görüntü işleme platformunda günümüzde en fazla kullanılan FPGA yapıları, tez çalışmasında kullanılan geliştirme kitine ve görüntü sensörüne değinilmiştir. Video görüntülerinin aktarılması ve geçici belleğe yazılması işlemi için gerekli sistem ve sistemi oluşturan alt modüller genişletilebilir ve uyarlanılabilir olarak tasarlanılmış olup FPGA yongasında gerçekleştirilmiştir. MATLAB ortamında belirlenen adımların daha sonra FPGA ortamında aktarılması amaçlanmıştır.

Bu tez çalışması ile kavşak ve otoyol trafik otomasyonu için gerekli olan görüntülerden araç tespitinin yapılması amaçlanmıştır. Böylelikle trafik akışı en hassas biçimde ayarlayacak olan akıllı sistem geliştirilmeye çalışılmış olup ayrıca ucuz, verimli ve geliştirmeye açık bir kontrol sistemi hedeflenmiştir.

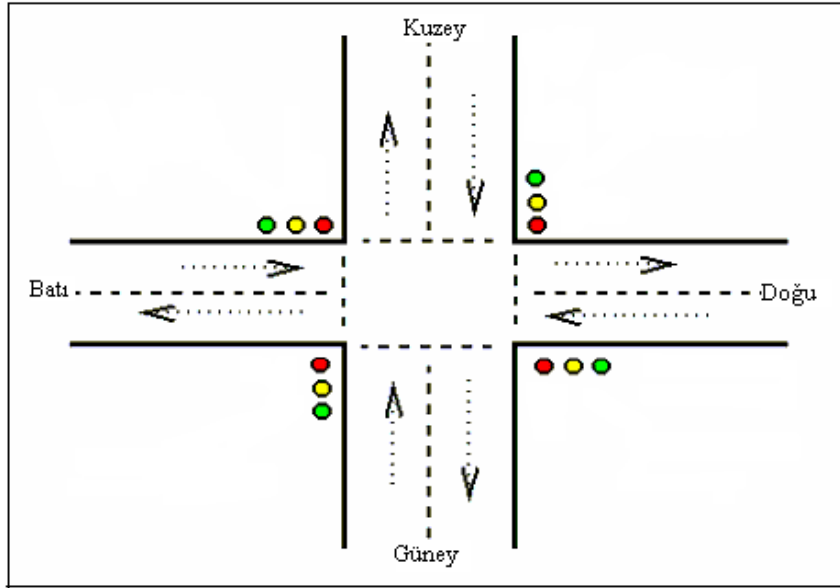
2. TRAFİK KONTROL SİSTEMLERİ

Trafik kontrol sistemlerinin temel amacı trafik akışının yönlendirilmesi olup herhangi oluşacak karmaşıklığı engellemektir. Kavşak ve tali yol geçişlerindeki trafik ışıklarının kontrolü ana amaçtır. Gün geçtikçe kontrol sistemlerinden daha akıllı, daha az hatalı ve daha performanslı çalışması beklenmektedir. Trafik kontrol sistemlerini 4 başlıkta toplayabiliriz.

- 1) Zaman ayarlı kontrol sistemi
- 2) Manyetik algılayıcı tabanlı kontrol sistemi
- 3) Ultrasonik veya lazer algılayıcı tabanlı kontrol sistemi
- 4) Görüntü işleme tabanlı kontrol sistemi

2.1 Zaman Ayarlı Kontrol Sistemi

Zaman ayarlı kontrol sistemleri trafik ışıklarını kontrol etmede kullanılan en basit ve en kolay kullanım sistemidir. Bir kavşağı ele alırsak kavşağın her bir tarafında bulunan araç ve yayalar için önceden belirlenmiş süre ve sırada geçiş hakkı tanınmaktadır. Şekil 2.1’de bir kavşakta trafik ışıklarının yerleşimi gösterilmiştir.

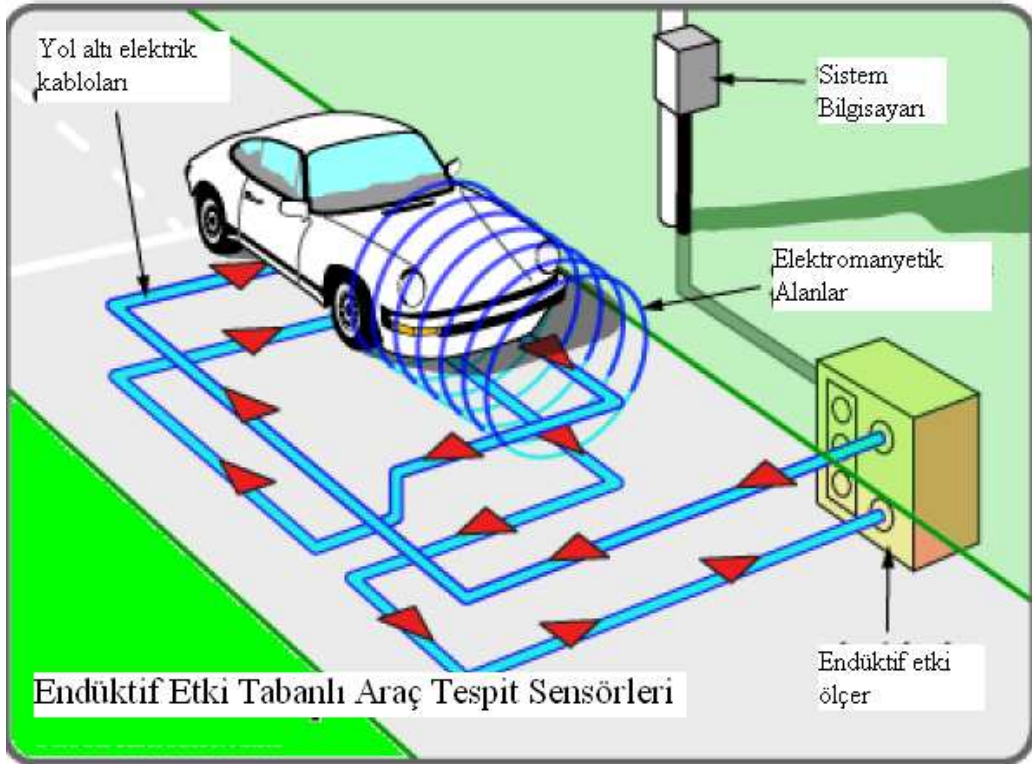


Şekil 2.1 Trafik ışıklarının bir kavşağa yerleştirilmesinin temsili gösterimi.

Bu sistem gün içerisinde oluşan yaya ve araç akışının daha önceden uzun bir süre izlenmesi ve buna göre trafik ışıklarının gün içerisinde ayarlanabilecek şekilde kontrol biriminin programlanmasıyla bir adım öne götürülebilir. Ancak yine de bu iyileştirmeler günümüz trafik yoğunluğuna ve gelecekte artmasına cevap veremeyecektir. Günümüzde bu şekilde önceden ayarlı trafik ışık sistemlerinin kontrolünde PLC tabanlı sistemler kullanılmaktadır.

2.2 Manyetik Algılayıcı Tabanlı Kontrol Sistemi

En yaygın kullanılan araç tespiti yöntemidir. Tespiti yapılacak yol sökölerek geniş daireler şeklinde kablolar döşenir. Bu kablolar endüktif etkinin (inductive loop) deęişiminin algılanması amacıylaadır. Araç yoldan geđerken bu kabloların üzerinde oluşan endüktif etki ana kontrol birimi tarafından algılanır ve ilgili kontrol işlemleri yapılır. Şekil 2.2’de endüktif etki tabanlı kontrol sistemi gösterilmiştir. Otopark girişlerinde, site girişlerinde ve yollarda sıkça görülür. Oldukça basit olmasının yanında kurulumunun zorluğu ve kurulduktan sonraki bakımının maliyetli olması sistemin dezavantajlı yönüdür. Ayrıca yoğun trafik akışı veya trafik akışının durması durumunda endüktif etki ölçer kablolarda elektromanyetik alan deęişimi algılamayacağından trafik kontrolü doğru bir şekilde sağlanamayacaktır.



Şekil 2.2 Manyetik tabanlı kontrol sisteminin çalışması.

2.3 Ultrasonik veya Lazer Algılayıcı Tabanlı Kontrol Sistemi

Kavşak veya yol birleşim yerlerindeki trafik ışıklarına ulaşan yolların bir veya daha fazla noktasına yerleştirilen alıcı-verici birimlerinden algılanan araç tespiti bilgilerinin ana bilgisayar tarafından değerlendirilip trafik ışıklarının yönlendirilmesi sistemidir.

Şekil 2.3’de Ultrasonik sensörün yerleşimi, araç var iken ve araç yok iken cevabı gösterilmiştir (Sakakira vd., 2005). Ultrasonik verici tarafından yayılan ultrasonik dalganın alıcıya ulaşması neticesinde yansıma zamanı ölçülür ve araç tespitinin kararı verilir.



(a)



(b)

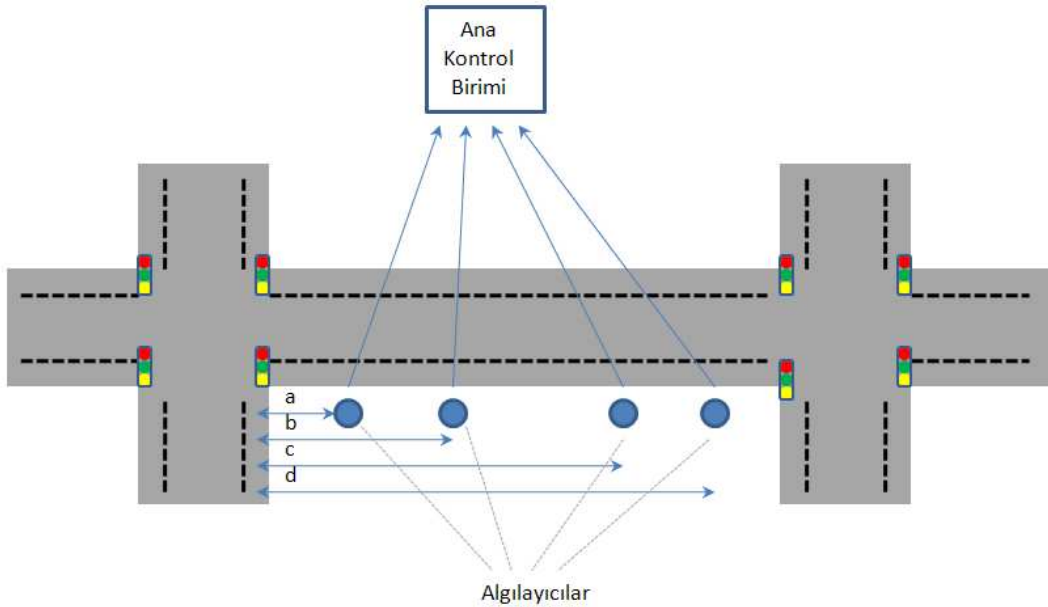
Şekil 2.3 Ultrasonik sensörler ile araç tespiti. (a) Ultrasonik sensörün yerleşimi ve araç var iken cevabı. (b) Araç yok iken cevabı (Sakakira vd., 2005).

Aynı şekilde lazer algılayıcıda araç tespitinde kullanılır. Farkı dalga gönderimi yerine lazer demeti gönderilmesidir. Böyle bir lazer algılayıcı Şekil 2.4'te gösterilmiştir.



Şekil 2.4 Lazer algılayıcı birimi.

Kavşağa yaklaşımdaki belirli aralıklarla yerleştirelen bu ultrasonik veya lazer algılayıcı birimlerden gelen bilgiler ana kontrol biriminde araç var yok bilgisinin yanında trafik araç yoğunluğu hakkında da bilgi verir ki bu da trafik ışıklarının sürelerinin doğru bir şekilde ayarlanmasına çok etkindir (Sakakira vd., 2005). Aşağıda a,b,c ve d mesafelerinde yerleştirilmiş ardışıl kavşak topolojisi gösterilmiştir (Şekil 2.5). Algılayıcıların sayısının artırılması ile daha doğru trafik yoğunluğu hesaplanabilir. Ancak kurulum maliyetinin ve kurulum sonrası idamesinin zorluğu da artacaktır.



Şekil 2.5 Ultrasonik veya lazer algılayıcıların kavşak araç tespiti ve trafik araç yoğunluğunu belirleme amacı ardışıl olarak ile yerleşimi.

2.4 Görüntü İşleme Tabanlı Kontrol Sistemi

Trafik araç tespiti sistemlerinde işlevsel olarak en karışık olanıdır. Video kaynağından elde

edilen ardışıl görüntü verilerinin işlenmesi neticesinde araç tespiti gerçekleştirilir. Kamera sistemi ilgilenilen alanın (ROI, Region of Interest) karşısına konumlandırılır ve belli bir alanın izlenmesi sağlanır. Bu sistem ayrıca kurallara uymayan sürücülerin anlık görüntülerinin alınmasını da sağlamaktadır. Ayrıca araç plaka tanınmasında da yararlanır. Yeni nesil sistemlerde ise RF iletişim ile ambulanslara yol verilmesi sağlanmaktadır. Ultrasonik ve lazer sistemlerinin belirli bir alanda birden çok kurulması gerekirken, kamera sistemi ile belirli bir alanın izlenebilmesi sağlandığından bir adet görüntüleme sistemi yeterli olmaktadır.

Aşağıda bu sistemin konumlanmasına ait iki örnek verilmiştir.



Şekil 2.6 Kamera tabanlı araç tespit sistemi (Sakakira vd., 2005).



Şekil 2.7 Kameranın trafik ışıklarının yanına yerleştirilmesi (Teksas - ABD , modot.mo.gov/stlouis/links/signalcameras.htm).

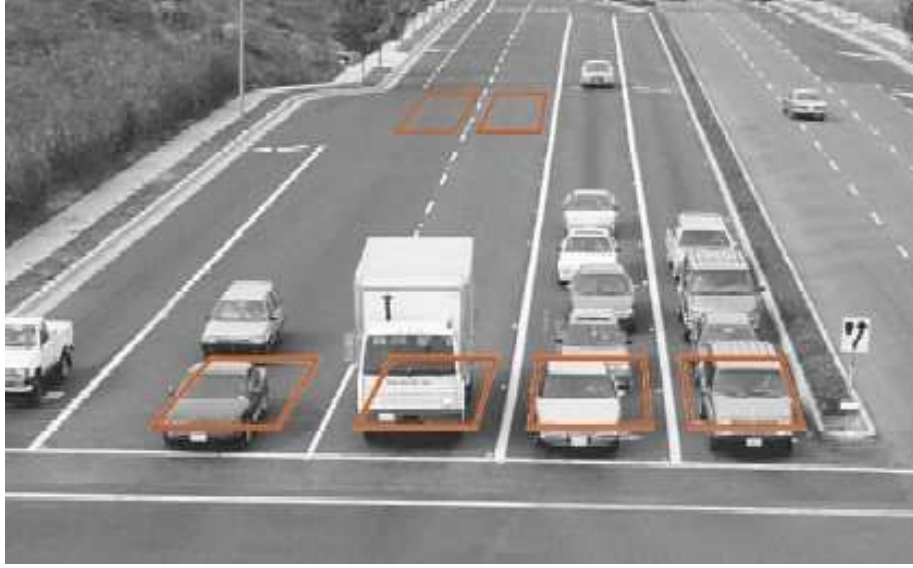


Şekil 2.8 Kameranın farklı bir şekilde yerleştirilmesi (Missouri – ABD, modot.mo.gov/stlouis/links/signalcameras.htm).

Bu sistemlerin kurulumu maliyetli olmasına karşın uzun ömürlülüğü, kurulumdan sonra bakımının kolaylığı, değişik yerlere kolayca uyarlanabilir olması avantajlarındandır. Kamera tabanlı araç tespit sistemlerinde iyileştirilmesi gerekenler arasında, rüzgar ve titreşimlere karşı önlemler, karlı ve yağmurlu havalarda doğru tespit edebilmesi, ortak ışık yoğun olmadığı durumlarda yani gece durumunda tespitin artırılması yer almaktadır.

Şekil 2.9'de kamera tabanlı kavşak trafik kontrol sistemi için araç tespit bölgesi gösterilmiştir.

Bu alanlar bilgisayar yazılımında tanımlanmaktadır. En soldaki alana araç girmediği müddetçe sola dönüş trafik ışığı yanmadığından bunun için ayrılacak zaman diğer yönlere aktarılacak olup sürücüler beklemeyecekler ve trafik araç birikmesi yaşanmayacaktır.



Şekil 2.9 Kavşak araç tespit sisteminde araç tanıma bölgelerinin yazılımda tanımlanmış hali (modot.mo.gov/stlouis/links/signalcameras.htm).

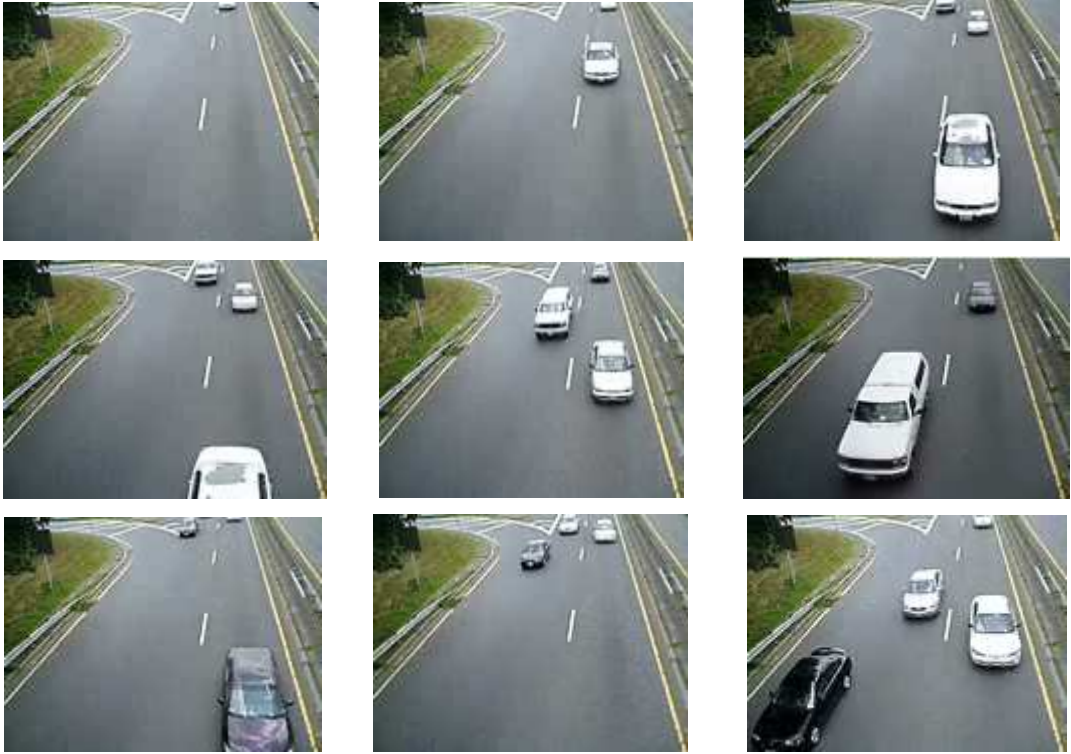
3. BİLGİSAYAR ORTAMINDA ARAÇ TESPİT SİSTEMİ

Bu bölümde daha önceden kaydedilmiş video görüntüleri üzerinden çeşitli yöntemler ve adımlar uygulanarak araç tespiti yani araç var ve yok bilgisi üretilmiştir. MATLAB geliştirme ortamı kullanılarak belirlenen bu adımlar daha sonra gömülü sisteme aktarılacaktır. Uygulanması istenilen her yere PC tabanlı bir çözüm hem çok pahalı hem de yavaş kalacağından dolayı gömülü sistem çözümleri vazgeçilmezdir.

Hareket halindeki bir veri kümesinde nesne veya örüntü bulma ve tanıma işlemi sistemin karmaşıklığını arttırmakta, algoritma ve efektif programlama sorunlarına neden olmaktadır. Bu sorunlar arasında arka plan çıkarımı, görüntülerdeki objelerin kenarlarının tespiti ve sayımı, gürültüden arındırma sayılabilir (Atkociunas vd., 2005).

Araç tespiti yazılımı birden çok adımdan oluşmakta olup her biri farklı görüntü işleme tekniklerindedir. Bu adımların birbirine bağlanması yani birinin çıktısının diğerinin girdisi olmasıyla tüm sistem tek halde çalışmaktadır.

Şekil 3.1’de kaydedilmiş video verisine ilişkin görüntüler verilmiştir.



Şekil 3.1 Analizi yapılacak olan video verisinin ardışıl görüntüleri.

Video verisi aşağıdaki özelliklere sahiptir.

Görüntü formatı : RGB

Görüntü sayısı : 120

Görüntü boyutu : 120 piksel x160 piksel

Şekil 3.2’de araç tespiti için video verisi üzerinde yapılan görüntü işleme teknikleri adımları belirtilmiştir.

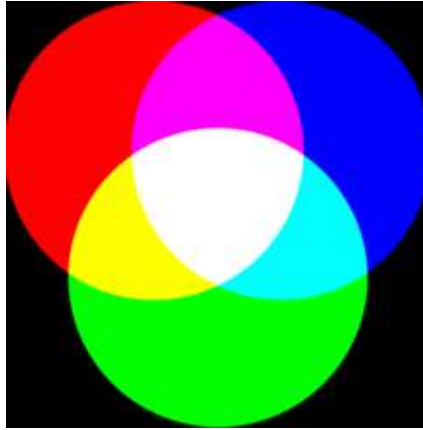


Şekil 3.2 Araç tespiti için görüntü işleme tekniklerinin adımlarının gösterilmesi.

3.1 Görüntünün RGB'den Gri Seviyeye Dönüşümü

RGB ifadesi görüntüdeki her bir pikselin Kırmızı-Yeşil-Mavi bileşenleri ile ifade edilmesidir. Renkli resimde denilmektedir. RGB çoğunlukla kabul edilen kamera çıkış formatıdır. Boyutunun fazla olması nedeniyle görüntüler işlenmeden önce gri seviyeye dönüştürülerek değerlendirilmektedir. Yazılımdaki bellek yerinden tasarruf edilmiş olup ayrıca ilgili işlemlerin daha hızlı çalışması sağlanacaktır.

RGB ifadesi 'Red'-'Green'-'Blue' kelimelerinin birleşmesi ile oluşur ve mevcut bütün renkler bu 3 ana rengin birleşimi ile ifade edilir. En fazla kullanılan renk uzayıdır. Diğer renk uzayları arasında PAL, CMYK ve HSV'yi sayabiliriz. Bu renklerin değişik oranda birleşmesiyle farklı ara renkler ortaya çıkmaktadır. Örneğin bu 3 renk %100 oranında birleştirilirse ortaya siyah renk çıkmaktadır.



Şekil 3.3 RGB renk uzayının gösterilmesi.

En fazla kullanılan renk uzayı olmasının nedenlerinden 1953'de ilk fotoğraf makinesi Polaroid'de kullanılması ve televizyonlarda standart olarak kabul edilmesini sayabiliriz. Günümüzde tarayıcılarda, ekranlarda, televizyonlarda ve fotoğraf makinelerinde standart olarak kullanılır.

Gri seviyede bir görüntüye ait pikseller '0' ile '255' arasındaki değerleri almaktadır ve bu yoğunluk bilgisi siyahtan beyaza kadar uzanmaktadır. RGB renk uzayına ait bir görüntünün piksellerinin gri seviyede gördüğü karşılık aşağıdaki denklemde verilmiştir. Bu denklem sayesinde gri seviyeye dönüşen görüntü üzerinde çalışmak daha kolay ve hızlı olacaktır.

$$\text{Gri Seviye} = 0.2989 * R (\text{Kırmızı}) + 0.5870 * G (\text{Yeşil}) + 0.1140 * B (\text{Mavi}) \quad (3.1)$$



Şekil 3.4 Video görüntüsünün RGB uzayından gri seviyeye dönüştürülmüş hali.

3.2 Yol Maskelenmesi

Yol maskelenmesi işlemi video verisindeki görüntülerdeki ölü bölgelerin kaldırılmasını ifade eder ki bu da görüntü işleme adımlarındaki işlem yoğunluğunun azaltılması demektir (Bhatti., 2006). Şayet kamera görüntüsü otoyolun iki yönünü de alıyorsa ilgilenilen alan (ROI) yol maskelenmesi sayesinde çıkartılır. Yol maskelenmesi aşağıdaki denkleme göre bulunur.

$$I_{\text{yeni}}(p) = M(p) * I_{\text{eski}}(p) \quad (3.2)$$

$I_{\text{yeni}}(p)$; gri seviye yeni görüntü, $I_{\text{eski}}(p)$; gri seviye eski görüntü, $M(p)$; ikili maskeyi ifade etmektedir. Yani $M(p)$, kaldırılmak istenen bölge için '0' bırakılmak istenen bölgeler için '1' değerine sahip olan piksel boyutları $I_{\text{eski}}(p)$ ile aynı olan ikili görüntüyü (binary image) ifade etmektedir.



Şekil 3.5 Yol maskelenmesi için resim işleme programında hazırlanan $M(p)$.



Şekil 3.6 Yol maskelenmesi işlemi yapıldıktan sonraki görüntülerden bazıları.



Şekil 3.7 Yol maskelenmesi işleminin kamera görüş açısı otoyolun iki yönünü de alması durumunda istenilen yön için yapılması (Bhatti., 2006).

3.3 Arka Plan Öğrenilmesi

Arka planın doğru bir şekilde öğrenilmesi araç tespitinin en önemli aşamalarındandır. Arka plan çıkarımında sabit duran nesnelere yok edilmesi ardışıl görüntüler üzerinde gerçekleştirilir. Arka plan çıkarımında görüntüdeki gürültüler ve çeşitli faktörlerin dikkate alınması önemlidir. Bu faktörler arasında yağmurlu, karlı, sisli havalarda ve ağaçların hareketini sayabiliriz. Aşağıdaki eşitlikte verildiği gibi arka plan çıkarımı görüntüye ait her bir piksel değeri için hesaplanmaktadır.

$$B(p) = \sum_k \frac{I_{yeni(k,p)}}{n} \quad (3.3)$$

Yukarıdaki denklemde k; görüntü sırası, p; piksel sırası, n; arka plan çıkarımı için alınmış olan görüntü sayısını, B(p) ise arka plandaki ilişkin piksel değerini göstermektedir. Bulunan arka planın yeni gelen görüntü ile çıkarımı neticesinde arka plan çıkarımı tamamlanmış olmaktadır. Aşağıdaki denklemde D(k,p); arka plan çıkarımı sonucundaki görüntünün piksel değerlerini, I(k,p); yeni gelen görüntünün ilgili piksel değerini göstermektedir.

$$D(k,p) = I(k,p) - B(p) \quad (3.4)$$

Şeki 3.8'den görüldüğü üzere arka plan çıkarımına ne kadar çok sayıda görüntü girerse sonucun doğruluğu da o kadar artmaktadır. Şekil 3.9'da D(k,p) görüntüleri gösterilmiştir.



(a)



(b)



(c)



(d)

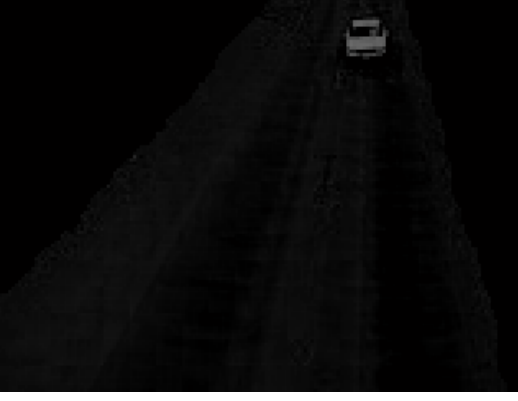


(e)



(f)

Şekil 3.8 Arka plan çıkarımının gerçekleştirilmesi, (a) $n = 5$, (b) $n = 10$, (c) $n = 15$, (d) $n = 20$, (e) $n = 50$, (f) $n = 100$.



(a)



(b)



(c)



(d)



(e)



(f)

Şekil 3.9 Arka plan çıkarımından sonra hesaplanacak olan $D(k,p)$ 'nin her bir görüntü için bulunması, (a) $k = 10$, (b) $k = 20$, (c) $k = 30$, (d) $k = 40$, (e) $k = 50$, (f) $k = 70$.

3.4 Eşikleme (Thresholding)

Şekil 3.9'daki görüntülerde görüleceği gibi gürültü çok fazladır ve bu gürültüler kaldırılmasındaki ilk adım görüntüye eşikleme işleminin uygulanmasıdır. Daha sonra filtre işlemleri uygulanarak görüntüdeki gürültüler kaldırılır. Eşikleme iki tür yapılabilir:

- 1) Sabit değer ile eşikleme
- 2) Histogram sonucuna göre eşikleme

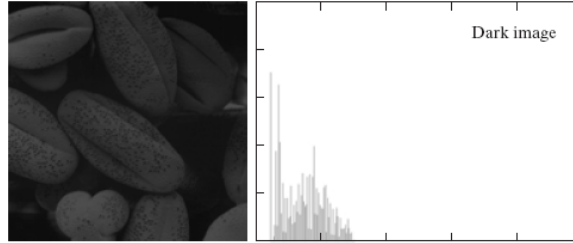
Sabit değer ile eşikleme eşitliği (3.5) aşağıdaki gibi ifade edilir. $D(k,p)$ 'nin eşik değerinden büyük olması durumunda bu piksel'in değeri $M(k,p)$ 'ye aktarılır ancak diğer durumda $M(k,p)$ değeri '0' olmaktadır.

$$M(k,p) = D(k,p) \quad (3.5)$$

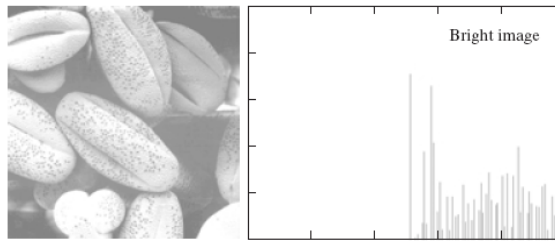
İkinci yöntemde ise öncelikle görüntünün histogramı hesaplanır. Histogram bize görüntüdeki piksel değerlerinin yoğunlukları hakkında bilgi verir. Aşağıda $h(r_k)$ ilgili gri seviyeye ait piksel sayısının n_k 'nin değeridir. Bu değerlerin n piksel sayısına bölümü sonucunda $p(r_k)$, gri seviye bilgisinin görüntüde bulunma ihtimalidir.

$$h(r_k) = n_k \quad (3.6)$$

$$p(r_k) = n_k/n \quad (3.7)$$



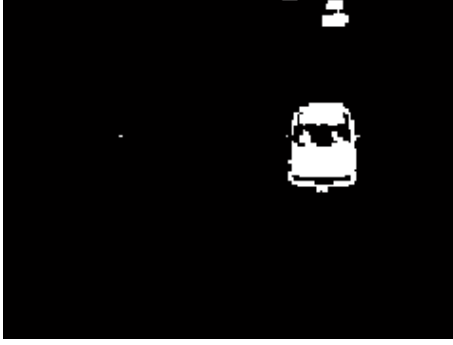
(a)



(b)

Şekil 3.10 Bazı görüntülere ait histogram bilgileri, (a) koyu görüntü, (b) açık görüntü (Gonzales vd., 2002).

Otsu uyarlamalı eşik belirleme algoritması gri seviyeli görüntülerin histogramlarında eşik değeri belirlenmesinde kullanılmaktadır. Uyarlamalı olduğundan dolayı görüntülerdeki farklı hatalarda verdiği sonuç etkilidir. Her k eşik değeri için tanımlanmış iki piksel kümesi arasındaki ayırım değeri hesaplandıktan sonra bu değer en büyük olduğu nokta optimal eşik değeridir. Uyarlamalı eşik bulma işlem yükü getireceğinden basitliğinden dolayı birinci yöntem seçilmiştir. Elde edilen görüntülerden bazıları aşağıda verilmiştir.



(a)



(b)

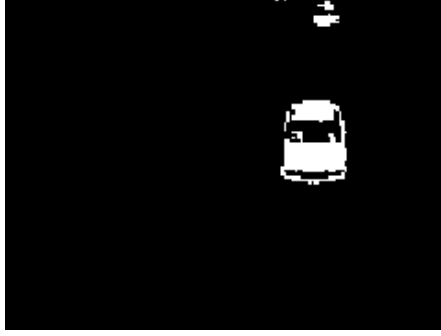


(c)



(d)

Şekil 3.11 Eşikleme değeri 30 için görüntüler, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.



(a)



(b)

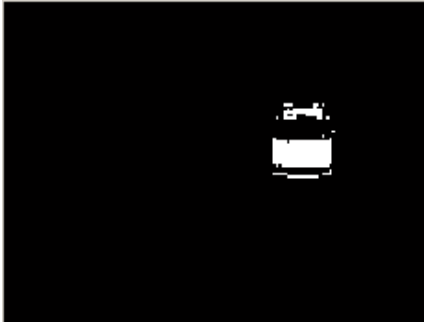


(c)



(d)

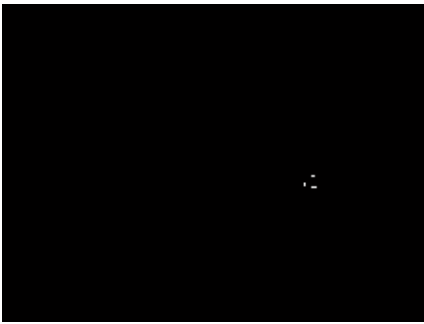
Şekil 3.12 Eşikleme değeri 50 için görüntüler, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.



(a)



(b)



(c)



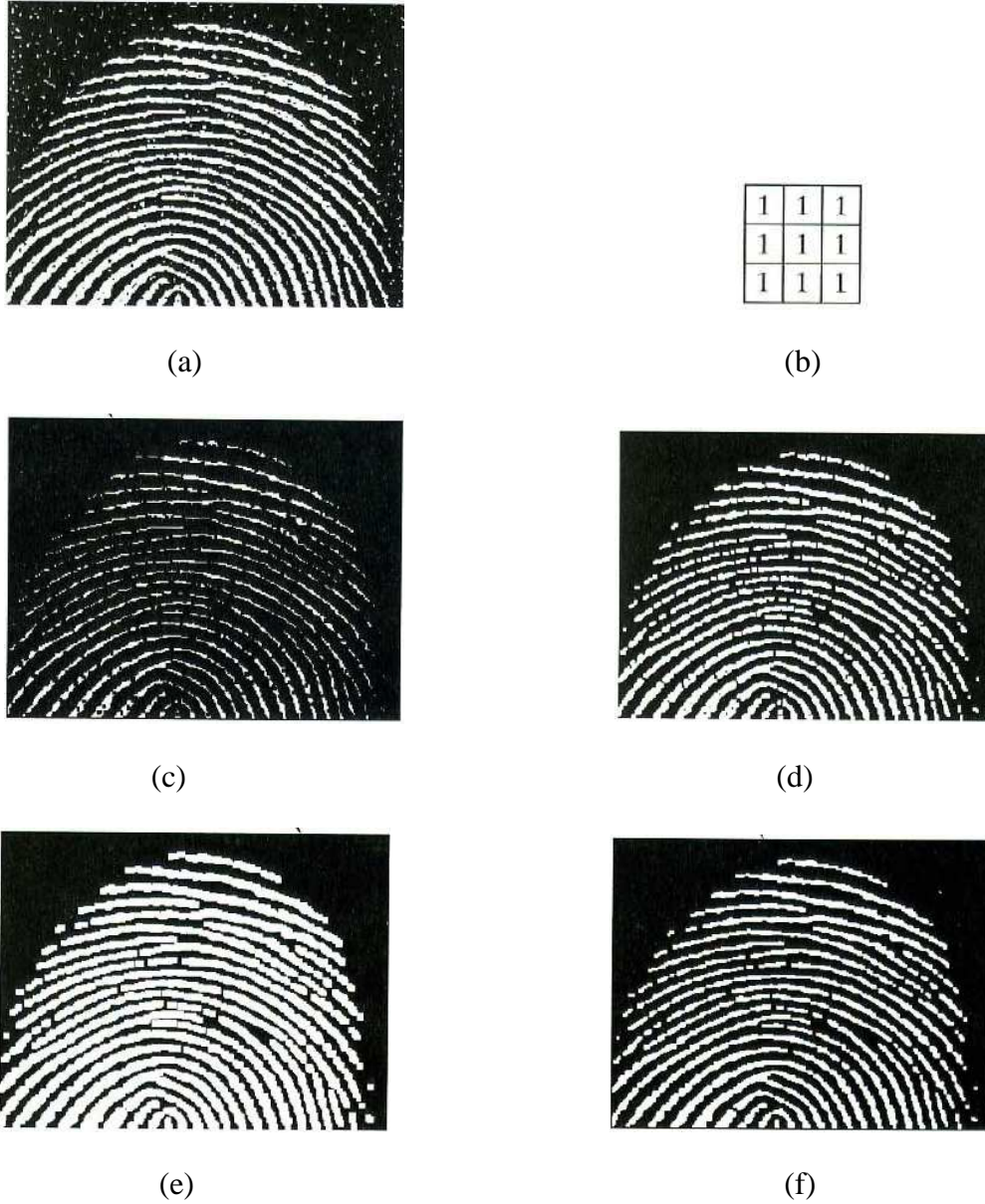
(d)

Şekil 3.13 Eşikleme değeri 100 için görüntüler, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.

Eşiklemenin farklı deęerleri için yukarıdaki sonuçlar incelendiğinde bu deęer arttığında küçük nesnelerin tespit edilememesi ve koyu renkteki araęların tespitinin zorluğu ile karşı karşıya kalınmıştır.

3.5 Morfolojik Kapama (Closing) İşlemi

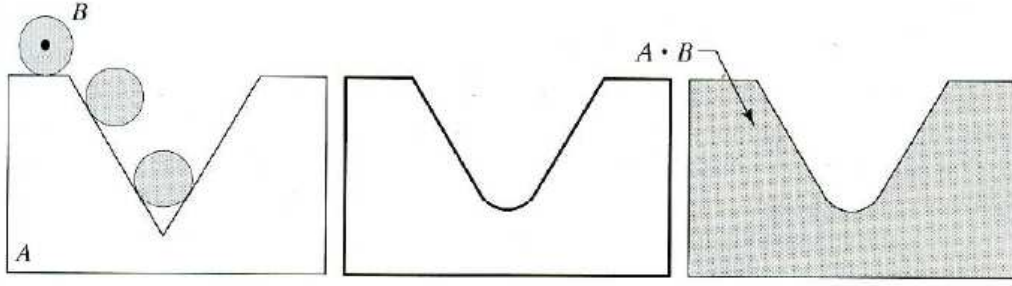
Morfolojik operatörler günümüzde görüntü işlemede çok fazla kullanılmaktadır. Eksik noktaların tamamlanması, fazlalıkların giderilmesi ve gürültüden arındırma kullanıldığı alanlardandır. Morfolojik işlemler matematiksel morfoloji ve küme teorisine dayanmaktadır. Radar ve uydu görüntülerinin tamamlanması, parmak izi tanımlamaları, eksik ve hasar görmüş resim ve metinlerin yeniden tanımlanmasını kullanılan alanlara örnek olarak verebiliriz. Genişletme (dilation), daraltma (erosion), kapama (closing) ve açma (opening) morfolojik işlemlerdendir. Bir parmak izi görüntüsüne çeşitli morfolojik işlemlerin uygulanması durumunda oluşan yeni görüntüler Şekil 3.14’de gösterilmiştir.



Şekil 3.14 Parmak izi görüntüsüne morfolojik işlemlerin uygulanması, (a) orijinal görüntü, (b) yapısal işlem elemanı, (c) daraltma, (d) açma, (e) açma sonucundaki görüntünün genişletilmesi, (f) açma işlemi sonucundaki görüntünün kapatılması (Gonzales vd., 2002).

Morfolojik kapama işlemi görüntüdeki yakın noktaların birleştirilmesi, görüntü üzerindeki girinti veya çıkıntıların yok edilmesi işlemlerini gerçekleştirir. Morfolojik kapama işleminin eşitliği (3.8)'de gösterilmiştir. B, belirlenen yapısal elemanı göstermektedir. Morfolojik kapama işleminde A görüntüsünün B ile önce genişletilmesi (dilation), daha sonra çıkan görüntünün daraltılması (erosion) işlemi yapılmaktadır. Görüntüde bulunan girintiler ve etrafı çevrilmiş bölgelerin doldurulması bu işlemin başarılarındandır.

$$A \cdot B = (A \oplus B) \ominus B \quad (3.8)$$

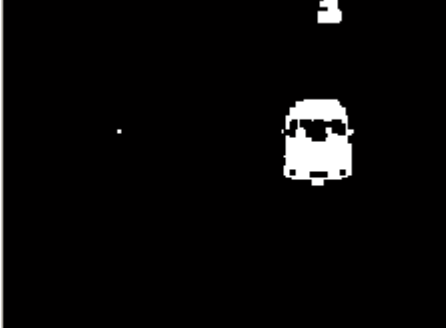


Şekil 3.15 Kapama (Closing) işleminin şablonları (Gonzales vd., 2002).

Şekil 3.16, 3.17 ve 3.18’de eşikleme değeri 30 için üç farklı yapısal eleman için analiz sonuçları verilmiştir. Görüntü sırası k ile gösterilmektedir. Karşılaştırma amaçlı eşikleme bölümündeki analizler ile aynı sıradaki görüntüler verilmiştir. Kapama işleminde kullanılan yapısal elemanın boyutunun büyümesi durumunda özellikle koyu araçlar daha belirgin hale gelmektedir. Bu durum sonraki adımlar için istenen bir sonuçtur.

1	1
1	1

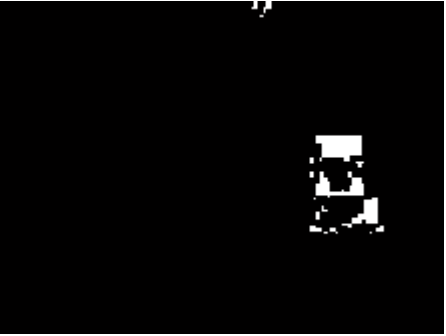
(a)



(b)



(c)



(d)

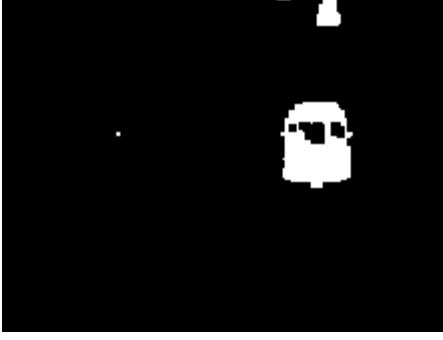


(e)

Şekil 3.16 Morfolojik kapama işlemi sonuçları, eşikleme değeri = 30, (a) yapısal eleman, (b) $k = 20$, (c) $k = 30$, (d) $k = 50$, (e) $k = 70$.

1	1	1
1	1	1
1	1	1

(a)



(b)



(c)



(d)

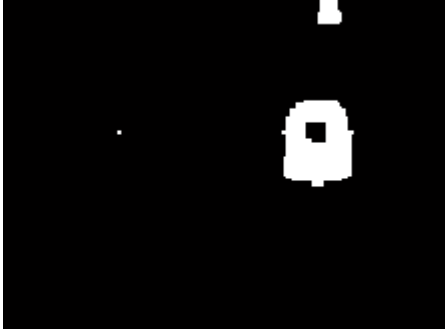


(e)

Şekil 3.17 Morfolojik kapama işlemi sonuçları, eşikleme değeri = 30, (a) yapısal eleman, (b) $k = 20$, (c) $k = 30$, (d) $k = 50$, (e) $k = 70$.

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

(a)



(b)



(c)



(d)

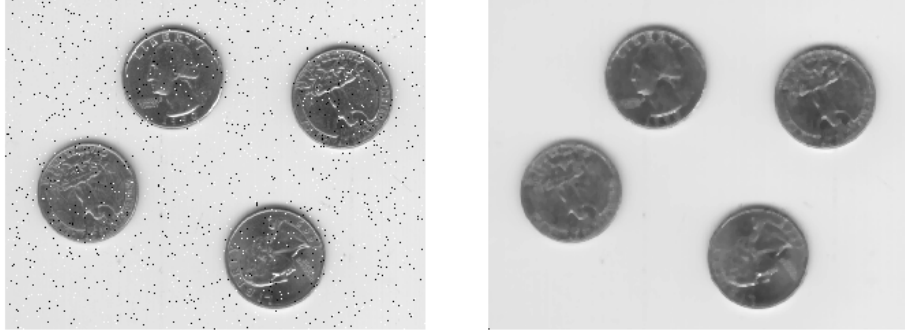


(e)

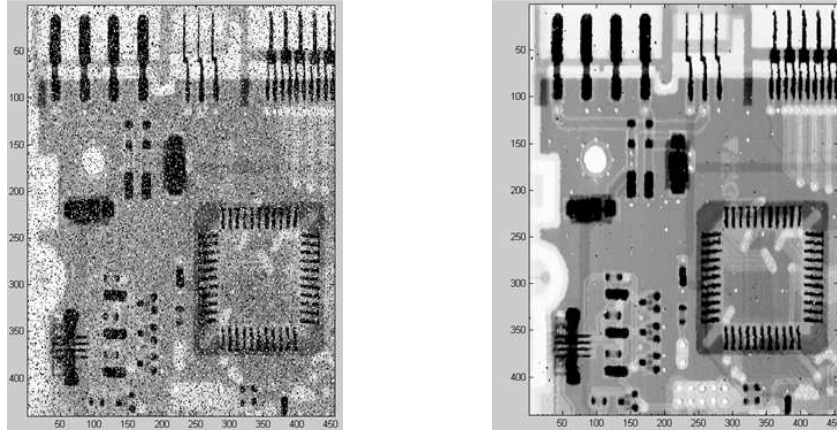
Şekil 3.18 Morfolojik kapama işlemi sonuçları , eşikleme değeri = 30, (a) yapısal eleman, (b) $k = 20$, (c) $k = 30$, (d) $k = 50$, (e) $k = 70$.

3.6 Medyan Filtreleme

Medyan ve ortalama (mean) filtreleme görüntü işlemede çok fazla kullanılmaktadır. Gürültü azaltılmasını sağlamakta olup obje kenarlarını korumasından dolayı konvolüsyon işleminden daha avantajlıdır. Önceden tanımlanan komşuluklardaki piksellerin değerleri artan şekilde sıralanır ve ortadaki alınır ki böylelikle yumuşatma işlemini gerçekleştirir.

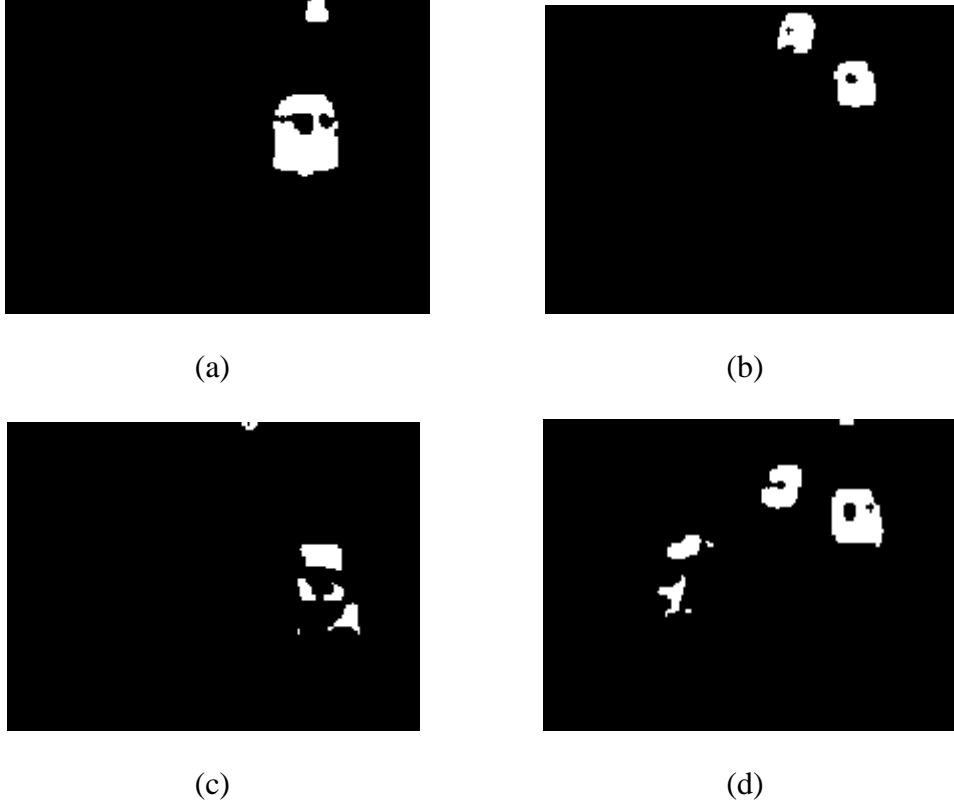


Şekil 3.19 Medyan işleminin tuz ve biber (salt and papper) gürültüsüne sahip bir görüntüye uygulanması (Gonzales vd., 2002).

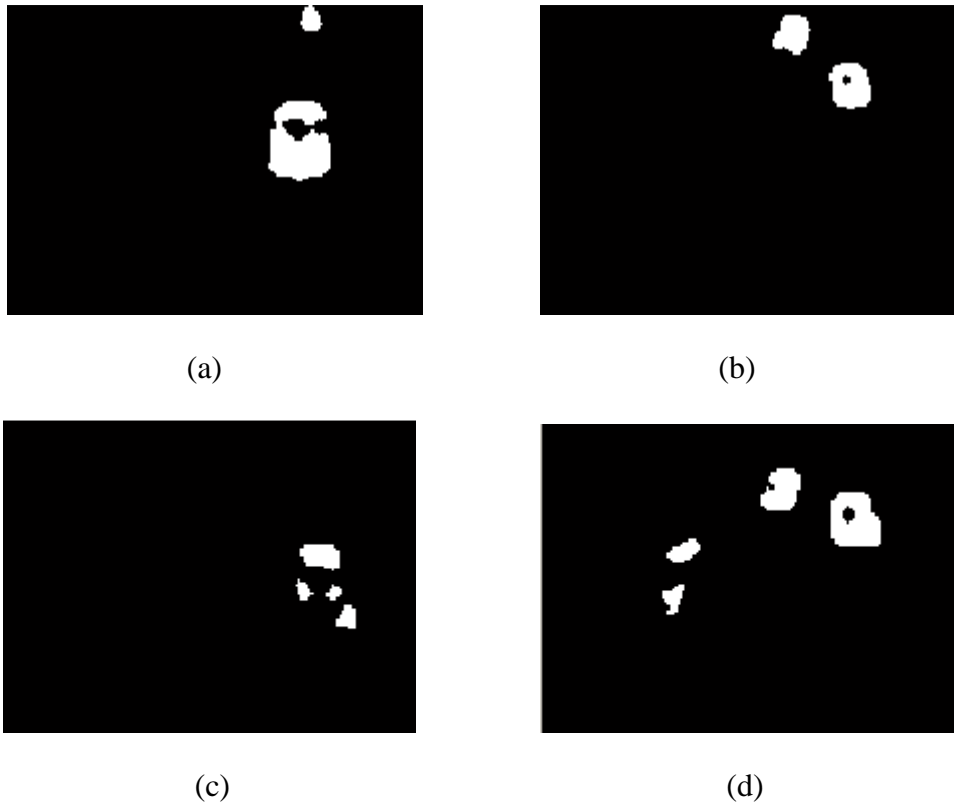


Şekil 3.20 Detayları anlaşılamayan bir PCB görüntüsüne medyan filtrenin uygulanması ile detaylarının belirgin hale getirilmesi (Gonzales vd., 2002).

Şekil 3.17'deki sonuçları verilen koşullardan sonra 3x3 komşulukta medyan filtre uygulanmasının görüntü üzerindeki iyileştirilme etkileri aşağıda görülmektedir. (Şekil 3.20) Aynı koşullardan sonra 5x5 komşulukta medyan filtre uygulanmasının sonuçları ise şekil 3.21'de verilmiştir.



Şekil 3.20 3x3 medyan filtre sonuçları, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.



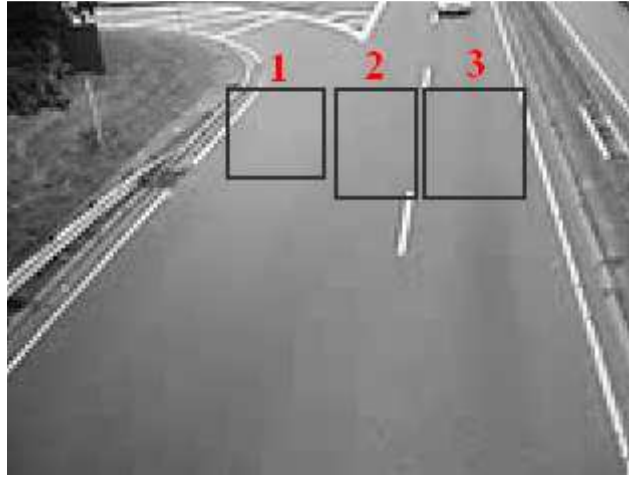
Şekil 3.21 5x5 medyan filtre sonuçları, (a) $k = 20$, (b) $k = 30$, (c) $k = 50$, (d) $k = 70$.

3.7 Tanımlanmış Bölgelerde Araç Tespitinin Yapılması

Bu aşamaya kadar araç tespitine yönelik tüm iyileştirmeler yapılmış olup bu bölümde araç tespiti için tanımlanan bölgelerdeki yoğunluk bilgisine bakılacak ve böylelikle araç var-yok bilgisi hesaplanacaktır. Şayet amaç araç tespiti ve takibi olsaydı görüntü üzerindeki detaylı bir inceleme yapılması ve nesnelere tespit gerekecekti. Burada bağlı komponent analizi ve etiketleme (connected-component analysis and labelling) işlemi yapılması gerekir (Bhatti., 2006).

3.7.1 Araç Tespiti Bölgelerinin Tanımlanması ve Başarımı

Kameranın konumlanmasına ve araç tespit bölgesinin değişimine göre ilgili bölgeler tanımlanmalıdır. Şekil 3.22’de tespit bölgelerinin tanımlanması görülmektedir. Burada kamera açısı ve izlenecek bölgelerin doğru bir şekilde tespiti sistemin başarımını arttıracaktır. Ele aldığımız video verisinin detaylı analizi yapıldıktan sonra 3 adet bölge tanımlanmıştır. Bu şekilde bu sistemin kurulacağı yerde kameranın kurulduktan sonra öncelikle alınan video verisinin detaylı incelenmesi yapılması gerekmektedir. Bu bölgeler aşağıda gösterilmiştir.



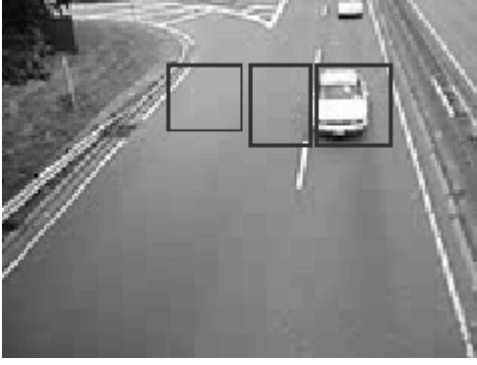
Şekil 3.22 Araç tespit bölgelerinin tanımlanması.

Video verisinde yukarıdaki tanımlanan bölgelere ilişkin araç geçiş zamanları Şekil 3.23’de verilmekte olup 10 adet araç bu bölgelerden geçmektedir. Video verisi 120x160 piksel boyutunda olup 120 görüntüden oluşmaktadır. Görüntü sırası k ile belirtilmiştir.

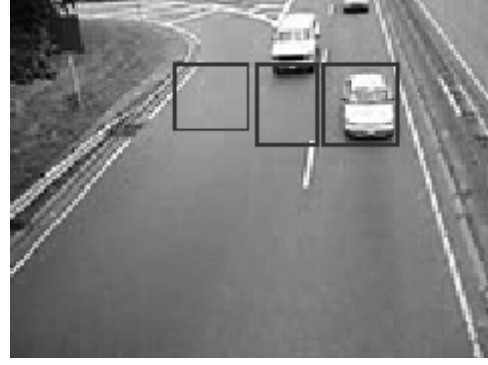
Çizelge 3.1’de tanımlanan bölgelere ait bilgiler verilmiştir.

Çizelge 3.1 Araç tespiti için tanımlanmış ilgili bölgelere ilişkin bilgiler.

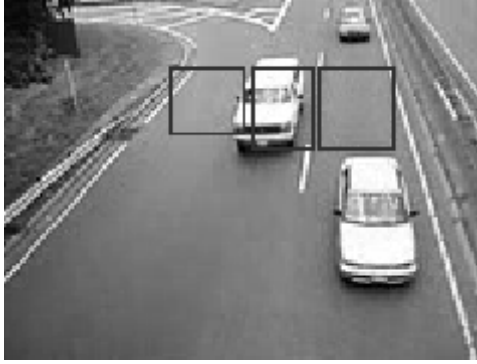
Bölge	Bölge piksel bilgileri	Bölge alanı
1	(56, 23) (80, 23) (56,45) (80, 45)	24x22
2	(83, 23) (103, 23) (83,50) (103, 50)	20x27
3	(105, 23) (130, 23) (105,50) (130, 50)	25x27



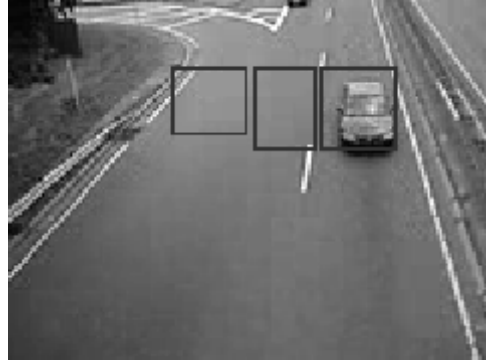
(a)



(b)



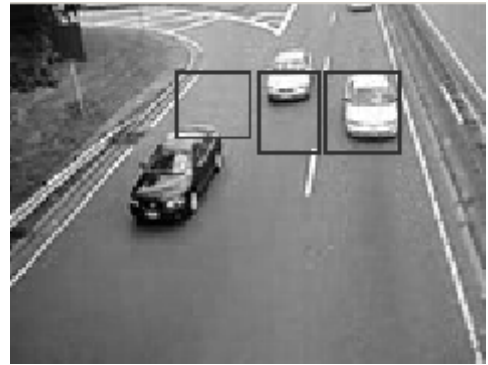
(c)



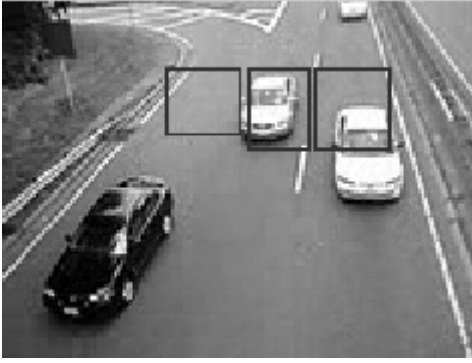
(d)



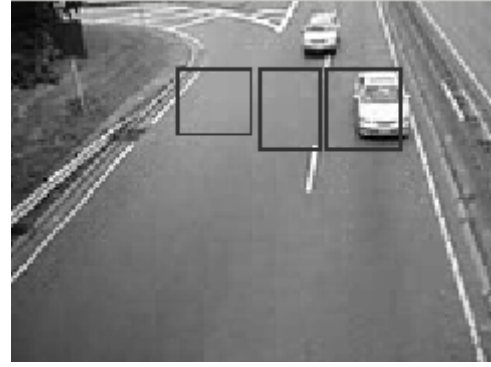
(e)



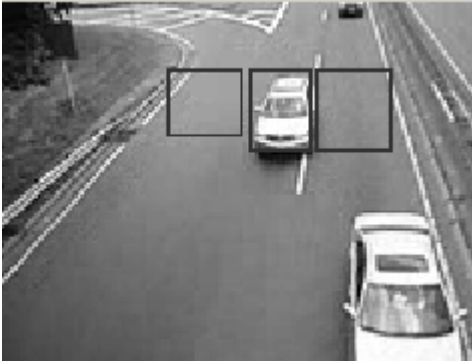
(f)



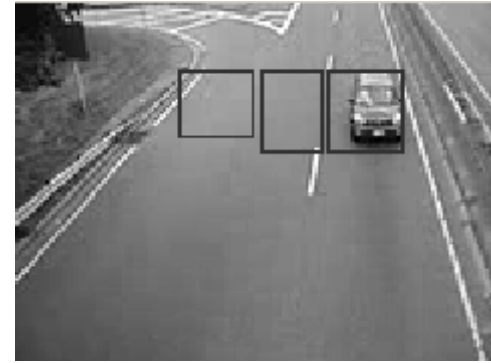
(g)



(h)



(i)



(j)

Şekil 3.23 Video verisi üzerinde tanımlanmış araç tespit bölgelerinden araç geçişi, (a) $k = 17$, (b) $k = 31$, (c) $k = 36$, (d) $k = 46$, (e) $k = 64$, (f) $k = 69$, (g) $k = 72$, (h) $k = 87$, (i) $k = 95$, (j) $k = 110$.

Yukarıda belirtilen bölgeler için tasarlanılan sistemin başarısı tanımlanan alanların belirli yüzdeleri eşik kabul edilerek Çizelge 3.2, Çizelge 3.3, Çizelge 3.4, Çizelge 3.5 ve Çizelge 3.6'da verilmiştir. Eşik kabul edilen alan yüzdesinin üzerindeki '255' değerleri sayısı araç tespitinin var olması diğer durumlar araç tespitinin olmaması demektir. Şekil 3.24'de eşik yüzdesi %15 için araç tespit görüntüleri verilmiştir. Eşik yüzdesinin %10 ve %15 olduğu durumlarda tanımlanan bölgeler için başarı oranı %100'dür. Koyu renkteki araçların tespiti için video verisi üzerindeki görüntü işleme teknikleri başarılı olmuştur, ancak eşik yüzdesinin %20 ve daha fazlası olduğu durumlarda bu söylenemez. Araç tespiti için yapılan iyileştirmeler yetersiz kalmıştır. Bu sistemin gerçekte uygulanmasında ise %10 ve %15 eşik düzeyi yeterlidir. Gece, karlı ve sisli hava durumları ise incelenmesi gereken durumlar arasındadır. Bu sayılan durumlar için araç tespiti yapılabilmesi için araç ışıklarının yanması durumundan yararlanılabilir (Bhatti, 2006).

Çizelge 3.2 Araç tespiti için tanımlanmış ilgili bölgelere ilişkin bilgiler (Eşik yüzdesinin %10 olması durumu).

Görüntü sırası (k)	Araç tespit	Eşik Yüzdesi	Başarım
17	Araç var	%10	%100
31	Araç var		
36	Araç var		
46	Araç var		
64	Araç var		
69	Araç var		
72	Araç var		
87	Araç var		
95	Araç var		
110	Araç var		

Çizelge 3.3 Araç tespiti için tanımlanmış ilgili bölgelere ilişkin bilgiler (Eşik yüzdesinin %15 olması durumu).

Görüntü sırası (k)	Araç tespit	Eşik Yüzdesi	Başarım
17	Araç var	%15	%100
31	Araç var		
36	Araç var		
46	Araç var		
64	Araç var		
69	Araç var		
72	Araç var		
87	Araç var		
95	Araç var		
110	Araç var		

Çizelge 3.4 Araç tespiti için tanımlanmış ilgili bölgelere ilişkin bilgiler (Eşik yüzdesinin %20 olması durumu).

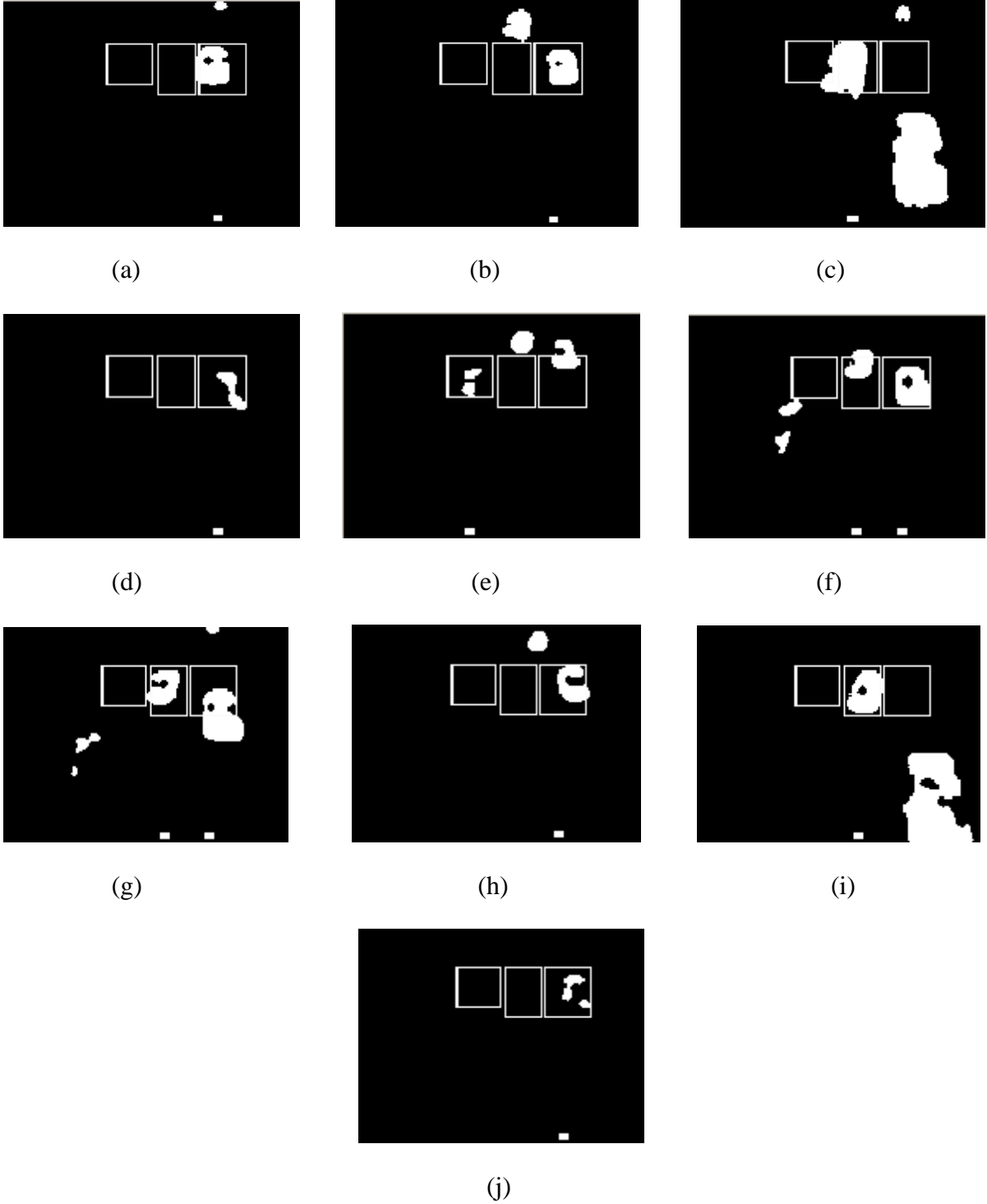
Görüntü sırası (k)	Araç tespit	Eşik Yüzdesi	Başarım
17	Araç var	%20	%70
31	Araç var		
36	Araç var		
46	Araç yok		
64	Araç yok		
69	Araç var		
72	Araç var		
87	Araç var		
95	Araç var		
110	Araç yok		

Çizelge 3.5 Araç tespiti için tanımlanmış ilgili bölgelere ilişkin bilgiler (Eşik yüzdesinin %25 olması durumu).

Görüntü sırası (<i>k</i>)	Araç tespit	Eşik Yüzdesi	Başarım
17	Araç var	%25	%70
31	Araç var		
36	Araç var		
46	Araç yok		
64	Araç yok		
69	Araç var		
72	Araç var		
87	Araç var		
95	Araç var		
110	Araç yok		

Çizelge 3.6 Araç tespiti için tanımlanmış ilgili bölgelere ilişkin bilgiler (Eşik yüzdesinin %40 olması durumu).

Görüntü sırası (<i>k</i>)	Araç tespit	Eşik Yüzdesi	Başarım
17	Araç var	%35	%70
31	Araç var		
36	Araç var		
46	Araç yok		
64	Araç yok		
69	Araç var		
72	Araç var		
87	Araç var		
95	Araç var		
110	Araç yok		



Şekil 3.24 Eşik yüzdesinin %15 olduğunda tanımlanan araç tespit bölgelerinden araç geçişi, (a) $k = 17$, (b) $k = 31$, (c) $k = 36$, (d) $k = 46$, (e) $k = 64$, (f) $k = 69$, (g) $k = 72$, (h) $k = 87$, (i) $k = 95$, (j) $k = 110$.

Kavşak otomasyonunda sinyalizasyon süresinin ayarlanmasında araç yoğunluk tespiti yapılması gerekmektedir. Kameranın bakış açısındaki şerit için tespit bölgeleri geniş tutulmalıdır (Bhatti, 2006). Bu durumdaki yoğunluk görüntüleri Şekil 3.25’de gösterilmiştir.



(a)



(b)

Şekil 3.25 Trafik sinyalizasyon süresinin ayarlanması için yoğunluk tespit bölgeleri, (a) gündüz, (b) gece durumu için araç ışıklarının algılanması (Bhatti, 2006).

4. GÖMÜLÜ SİSTEMİN DONANIMSAL BİLEŞENLERİ

Proje tasarımının gerçekleşmesinde kullanılan donanımsal bileşenler aşağıda detaylı olarak anlatılmıştır.

4.1 FPGA

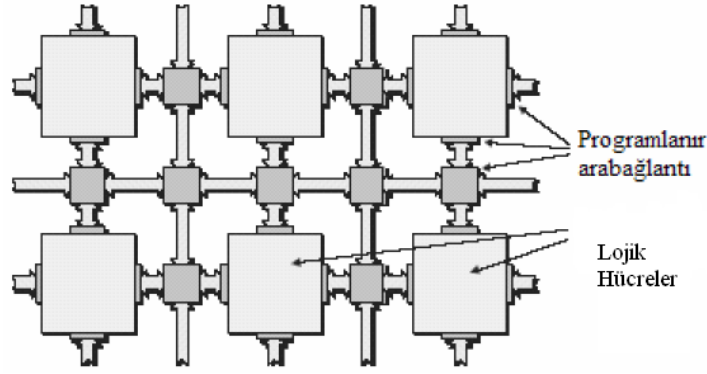
FPGA (Field Programmable Gate Array – Alan programlanabilir kapı dizisi), programlanabilir lojik bloklar dizisi ve bunların bağlanması için gerekli olan ara bağlantılardan oluşmuş sayısal tasarım dünyasında çok fazla kullanılmakta olan tümdevrelerdir (Şekil 4.1). Yeniden konfigüre edilebilmesinin SRAM (statik RAM) teknolojisi ile olması sayesinde binlerce defa programlanabilmektedir. CPLD'den (Complex Programmable Logic Devices – Kompleks programlanabilir lojik tümdevre) en temel farkları çok daha fazla lojik işlem kapasitesine sahip olması ve SRAM tabanlı yani her güç verilmesinde konfigüre edilmesinin gerekmesidir. CPLD'ler flash tabanlıdır yani her güç verilmesinde konfigüre edilme ihtiyacı duymazlar. FPGA'ler farklı seviyelerdeki lojik uygulamaları gerçekleştirebilmektedir. Günümüzde SRAM tabanlı FPGA yapısı çoğunluktadır.

Aşağıda FPGA'lerin mimarilerine göre çeşitleri verilmiştir:

- 1) Sigorta tabanlı.
- 2) Antisigorta tabanlı.
- 3) EPROM, EEPROM veya Flash tabanlı.
- 4) SRAM tabanlı.

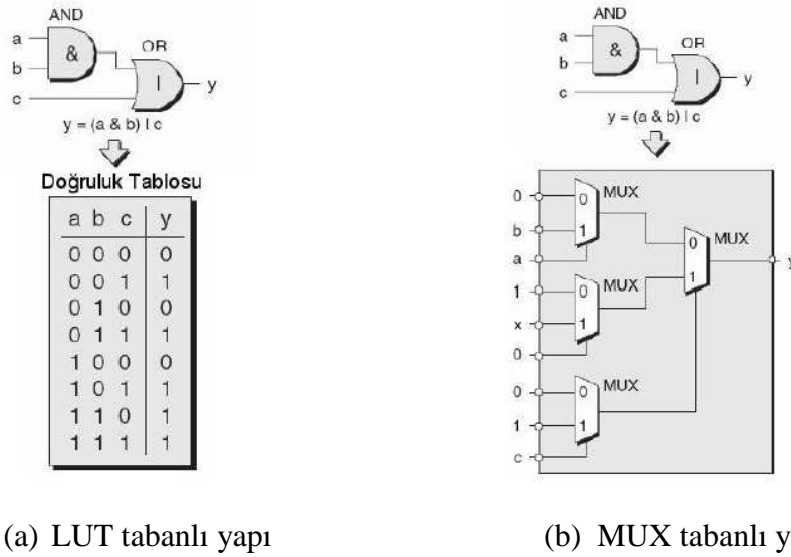
FPGA'lerin avantajlarını ise şu şekilde sıralayabiliriz :

- Paralel çalışma.
- Güç verimi ve hız kabiliyetleri.
- Yapısını sistem çalışırken değiştirebilme.
- Standart ve süper bilgisayarların emülasyonu.
- ASIC'lerin (Application Spesific Integrated Circuit) ön kabul testleri.
- Zamanlama yeteneklerinin çok iyi olması.
- Çok fazla karmaşık lojik devre uygulamaları ve çok fazla lojik kapasite ihtiyacı.



Şekil 4.1 FPGA yapısı.

Lojik hücreler kendilerinden istenen yapıyı LUT ya da MUX tabanlı gerçekleyebilirler. LUT tabanlı yapı SRAM tabanlı olup giriş değerlerinin alacağı durumlar için çıkış değerini başvuru tablosunda (LUT, look-up table) tutmaktadır. MUX tabanlı yapı farklı yapıdaki çoğullayıcılar (multiplexer) ile çıkış değerini giriş değerlerinin farklı değerlerine göre güncellemektedir.



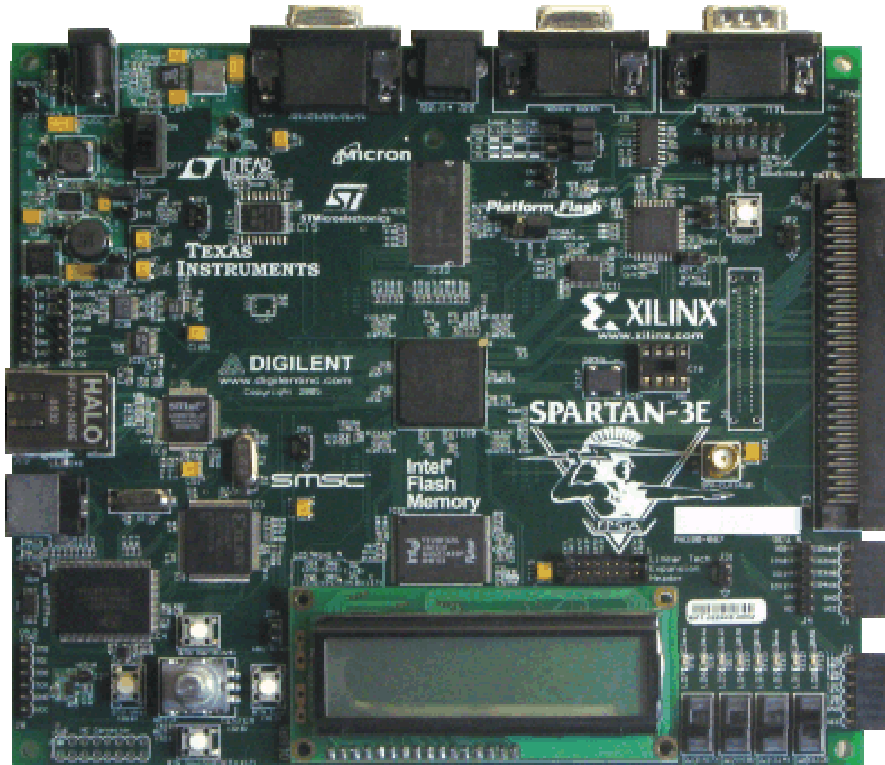
Şekil 4.2 FPGA lojik hücrelerin gerçeklenme mimarileri.

FPGA'lerin konfigürasyonu için gerekli olan derlenmiş program harici bir hafıza entegresinde saklanmaktadır. Bu durum ise tasarımın korunması durumunu göz önüne getirmektedir. Bu duruma karşı alınacak bir önlem ise konfigürasyon dosyasını şifrelemektir. Tabii ki bu durumda şifrelenmeyi destekleyen FPGA seçmek gerekmektedir. Şifre anahtarı FPGA'ye yüklenir ve şifrelenmiş konfigürasyon dosyasını FPGA'ye yüklemek geri okuma modunu kaldıracaktır. FPGA'in enerjisi kesildiğinde şifre anahtarının kaybolmaması için harici bir pil ile şifre anahtarının yüklü olduğu kısım beslenmelidir.

4.2 FPGA Geliştirme Kartı

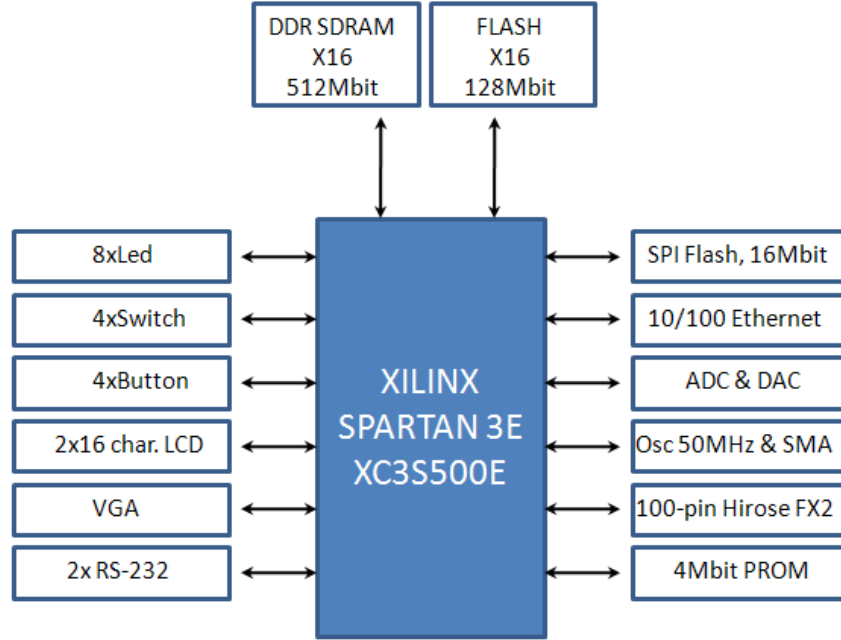
Proje süresince Digilent firmasının Spartan 3E starter kit isimli ürünü kullanılmıştır. Aşağıda bu geliştirme kartı üzerinde bulunan özellikler verilmiştir :

- XC3S500E Spartan 3E FPGA.
- 4Mbit konfigürasyon PROM.
- 64 macrocell XC2C64A CPLD.
- 512Mbit DDR SDRAM. (16-bit veri yolu genişliği, 100+ MHz)
- 128Mbit paralel NOR flash.
- 16Mbit SPI flash.
- 2x16 karakter LCD.
- PS/2 fare veya klavye girişi, VGA ekran çıkışı, 10\100 Mbit Ethernet arayüzü.
- 2xRS-232 arayüzü.
- 8-led, 4-anahtar, 4-button.
- SPI arayüzüne sahip ADC ve DAC.
- Ayrı bir programlayıcıya ihtiyaç duymama.



Şekil 4.3 Digilent Spartan 3E geliştirme kartı görünüşü.

Bu geliştirme kartı üzerinde FPGA'ye saat girişi sağlayacak 50MHz frekasında osilatör bulunmaktadır, ayrıca harici saat girişi için de bir SMA bulunmaktadır.

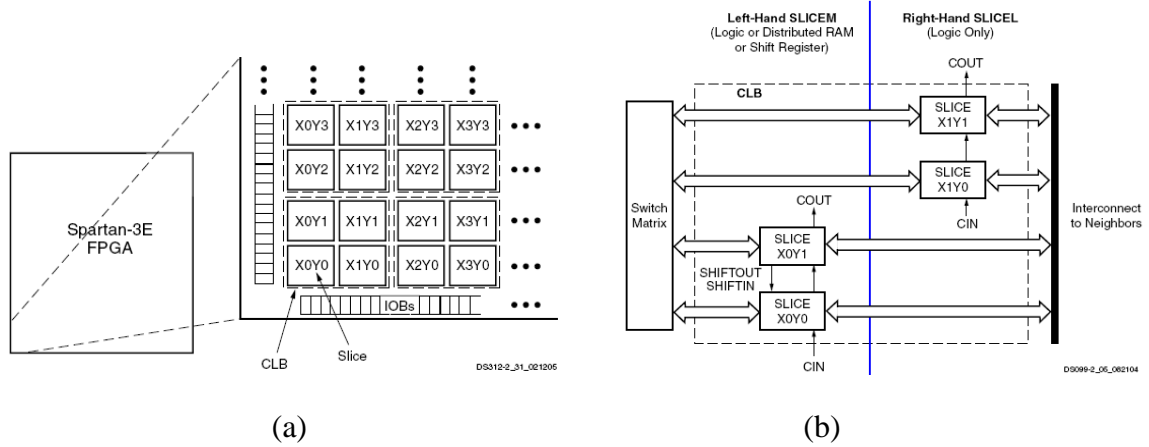


Şekil 4.4 Digilent Spartan 3E geliştirme kartı blok diagramı.

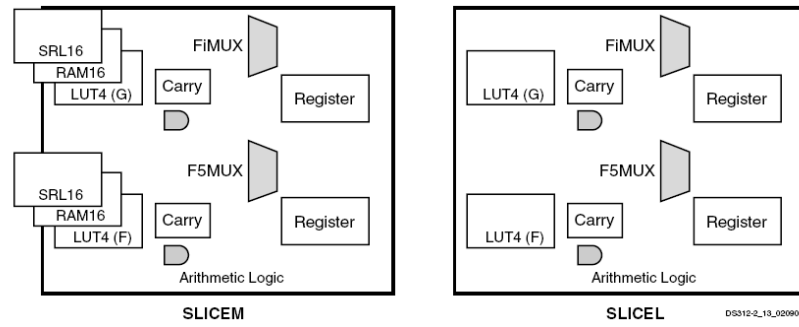
4.2.1 Xilinx Spartan 3E FPGA

Spartan FPGA ailesi Xilinx firmasının orta düzey özellikli olan FPGA ailelerine verdiği isimdir. Virtex II, Virtex 4, Virtex 5 ve Virtex 6 FPGA aileleri Xilinx firmasının yüksek performanslı FPGA'leridir. Geliştirme kartı üzerinde Spartan 3E ailesinden XC3S500E FPGA mevcuttur. Bu FPGA'in özellikleri olarak,

- 10.476 lojik hücre, 1164 CLB (configurable logic block),
- FBGA-320 kılıf,
- 20 adet 18x18 çarpıcı,
- 20 adet 18Kbit Blok RAM,
- 73Kbit dağıtılmış RAM
- 4 adet dijital saat yönetici birimi (PLL),
- 232 kullanıcı I/O,
- 90nm üretim prosesi,
- LVCMOS, LVTTTL, SSTL, HSTL ve LVDS standartlarına destekleyen I/O,
- JTAG ile programlamayı sayabiliriz.



Şekil 4.5 Spartan 3E FPGA'in CLB ve slice yapısı, (a) genel, (b) detaylı.



Şekil 4.6 Spartan 3E FPGA'in SLICEM ve SLICEL yapısı.

Şekil 4.5'den görüldüğü gibi Spartan 3E FPGA tümdevresi CLB'lerden (configurable logic block) oluşmakta, her CLB ise SLICEM ve SLICEL olarak 2 kısımdan meydana gelmektedir. SLICEM, SLICEL'den farklı olarak SRL16 kayma yazmacına ve RAM16 hafızasına sahiptir. (Şekil 4.6) Her bir SLICE 2 adet LUT4 yapısı içermekte olup, 1 CLB'de toplam 4 SLICE yani 8 adet LUT yapısı vardır. XC3S500E FPGA tümdevresinde 1164 CLB, yani 9312 LUT vardır. Bunun eşdeğeri ise 10.476 lojik hücre (lojik cell) demektir.

FPGA'in giriş ve çıkış portları 1.2V, 1.5V, 1.8V, 2.5V ve 3.3V gerilim seviyelerini desteklemektedir. Sahip olduğu 232 I/O pini 116 farksal sinyalleşme (differential signalling) mimarisinde kullanılabilir. Blok RAM, çift port (dual port) yapısını desteklediğinden çift port RAM olarak kullanılabilir, yani bir porttan yazma yapılırken diğer porttan okuma gerçekleştirilebilir.

4.3 Görüntü Sensörü

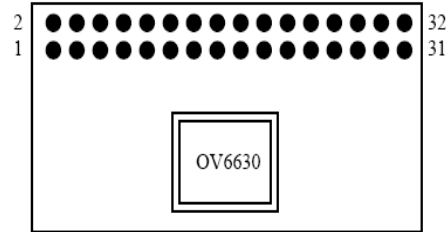
Görüntünün optik bir mercek yardımıyla sensörler tarafından algılanıp dijital bilgiye dönüştürülmesi işlemini gerçekleştiren tümdevre yapıları olan görüntü sensörleri günümüzde CMOS ve CCD olmak üzere ikiye ayrılmaktadır. CMOS ve CCD sensörler kamerada sayısal film gibi davranmaktadır. CCD teknolojisi kamera endüstrisi için geliştirilmiş bir teknoloji olup uzun süredir kullanılmaktadır. Günümüzde yüksek kalitedeki kameralar CCD teknolojisine sahiptir. CCD sensörler ışığa duyarlı fotodiyotlardan oluşmakta olup düşük ışık şiddetine oranla elektrik gerilimi üzerlerinde üretilir. Işığa karşı daha fazla hassas olduğundan CMOS sensörlerden daha kaliteli görüntüler elde edilir. Düşük ışık ortamlarında bile detaylı görüntü sağlaması ve düşük arkaplan gürültüsü üretmesi avantajlarından, üretiminin pahalı olması, ağ tabanlı kameraya entegrasyonun zorluğu ve ekrana çok parlak ışık geldiğinde görüntünün altında ve üzerinde siyah çizgi oluşturması ise dezavantajlarından. CMOS teknolojisi ise bellek, mikroişlemci ve diğer elektronik elemanlarda kullanıldığı için standart bir teknoloji olup, ucuz ve uygulaması kolaydır. Gürültü oranının yüksekliği ve düşük ışık hassasiyeti dezavantajlarından. Günümüzde ise CMOS sensörlerin görüntü kaliteleri CCD sensörlere yaklaşmıştır. CMOS sensörünün çıkışının dijital olması CCD sensör tabanlı sistemlerdeki ekstra ADC ve işlemci kullanılması gibi maliyet artırıcı faktörleri gereksiz kılmaktadır. Maliyet, temin edilebilirlik ve uygulama kolaylığından dolayı CMOS sensör seçilmiştir.

4.3.1 Görüntü sensörü kartı

Bu çalışmada üzerinde OmniVision firmasının OV6630 görüntü sensörü bulunan C3038 kamera modül kartı kullanılmıştır. Bu kart üzerinde objektif ve OV6630 sensörü için gerekli olan pasif elemanlar bulunmaktadır.



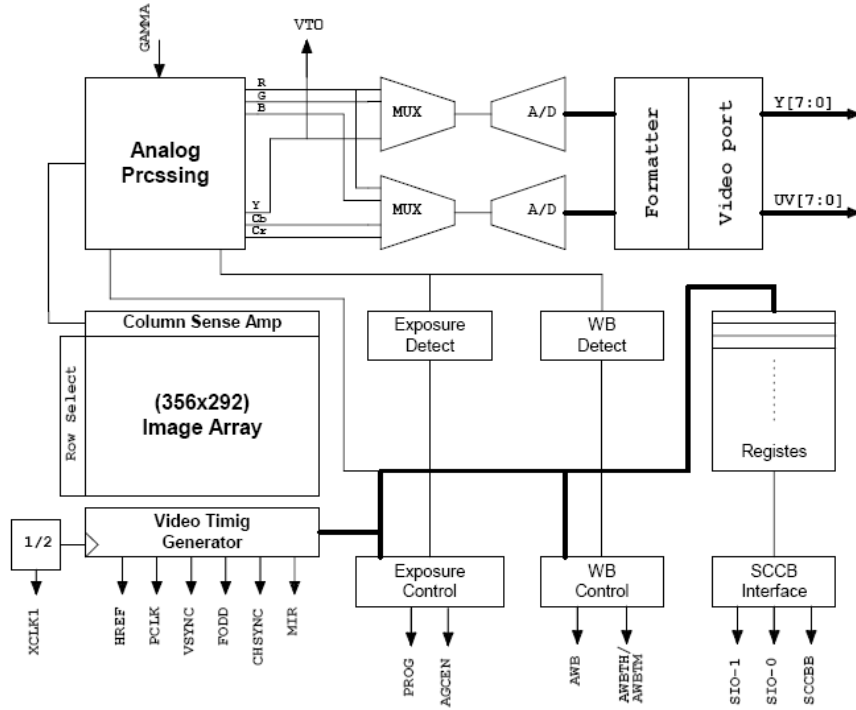
(a)



(b)

Şekil 4.7 Görüntü sensör donanımı, (a) sensör kartı, (b) temsili çizim.

OV6630 görüntü sensörü 352x288 çözünürlükte, 60 fps görüntü hızında renkli görüntü alabilmektedir. 1/4" lense sahiptir ve 3.3V gerilimde çalışmaktadır. Gerekli olan bütün ayarları SCCB iletişim protokolü üzerinden yapılmaktadır. SCCB protokolü, I2C protokolüne benzer ancak OmniVision tarafından değişiklikler yapılmış çoklu kamera bağlanması sistemine uygunlaştırılmıştır. Çıkışı sayısal veri olduğundan herhangi bir devre tasarımı durumu yoktur. YCrCb 4:2:2, GRB 4:2:2 ve RGB formatlarında görüntü verisi verebilmekte, 8/16 bit ITU-601, ITU-656 ve ZV port video verisini desteklemektedir. OV6630 görüntü sensörü çalışırken 20mA'den az akım çekmekte olup düşük güç tüketen uygulamalar için idealdir. OV6630 görüntü sensörünün çalışma blok diyagramı şekil 4.8'de gösterilmiştir. Analog işlem katından sonra görüntü üzerine kazanç arttırımı, dengeleme, parlaklık, kontrast ayarlamaları, pozlama, keskinlik ve pencereleme gibi işlemler görüntü sensörü tarafından yapılabilmektedir.



Şekil 4.8 OV6630 görüntü sensörü çalışma blok çizimi.

5. GÖMÜLÜ SİSTEMİN GERÇEKLENMESİ

Bu bölümde öncelikle FPGA tabanlı gömülü sistem oluşturma yöntemleri anlatılmıştır. Daha sonra ise uygulamamıza özel olarak tasarlanılan görüntü işleme amaçlı gömülü sistem detayları ve çalışması anlatılacaktır.

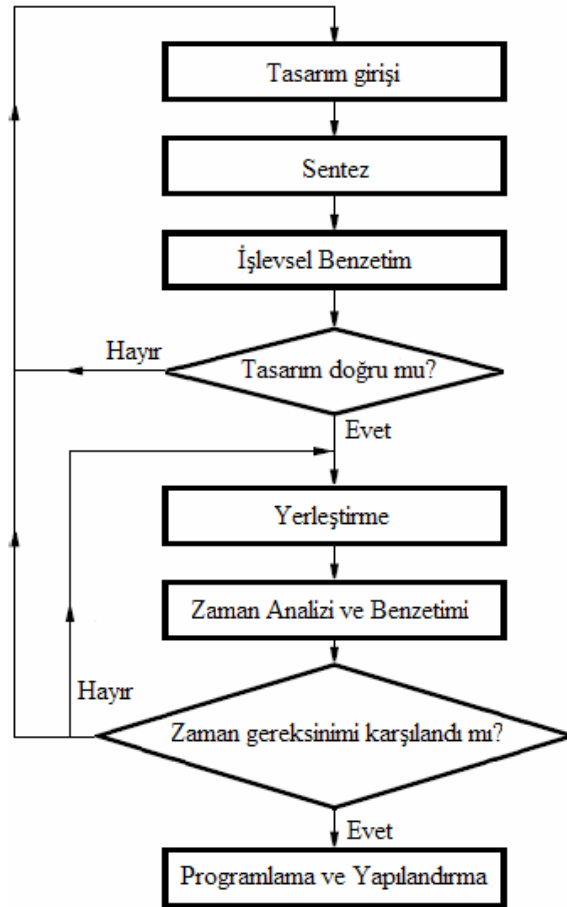
5.1 FPGA Tabanlı Gömülü Sistem Geliştirme

FPGA tabanlı gömülü sistem oluşturma yöntemleri aşağıda verilmiştir.

- 1) HDL tabanlı tasarım geliştirme.
- 2) İşlemci tabanlı tasarım geliştirme.

5.1.1 HDL (Donanım Tasarım Dili) Tabanlı Tasarım Geliştirme

HDL temelli gömülü sistem geliştirmede izlenecek yol aşağıda verilmiştir. Günümüzde uygulama geliştirmenin tüm safhalarında CAD (Computer Aided Design) programlarından yararlanılmaktadır.



Şekil 5.1 FPGA temelli uygulama geliştirmenin safhaları (Aydın, 2005).

Tasarım girişi şematik tabanlı veya yüksek seviye programlama dili (VHDL veya Verilog) tabanlı olmaktadır. Tasarımların büyüdüğü durumlar için şematik tabanlı tasarım yetersiz kalmaktadır. Esnekliği, genişleyebilirliği, üreticiden bağımsızlığı ve teknolojilerdeki geçiş kolaylığı nedeniyle programlama dilleri kullanarak tasarımı metin şekilde oluşturma kullanıcılar arasında çok tercih edilmektedir. Sentezleme aşamasında, tasarım girişindeki gerçekleştirilmek istenen lojik devreler CAD araçları tarafından gerçekleştirilir. Sentezlenen devrenin benzetimi test programları yazılarak işlevsel benzetim aşamasında gerçekleştirilir. Doğrulandıktan sonra yerleştirme aşamasına geçilir. Yine bu aşamada CAD araçları sentezleme ile belirtilen lojik devreleri seçilen FPGA türüne göre yerleştirme işlemini yapar. Lojik devreler arasındaki tüm arabağlantılar bu aşamada yönlendirilirler. Yerleştirilmesi bitmiş tasarımın yollarının gecikmesinin incelenmesi ve başarımı istenen lojik devrelerin maksimum çalışma frekansı belirlenir ve gereksinim karşılanmazsa iyileştirmelere gidilir. Bu aşamanın istenen performansı sağlamasından sonra ilgili CAD aracı ile FPGA tümdevresi programlanır. Sistem çalışırken lojik devrelerin ayıklama (debug) çalışmaları ise kullanıcıya çok büyük bir kolaylık sağlamaktadır. Xilinx firmasının ChipScope adlı ayıklama aracı projenin gerçekleştirilme boyunca etkin bir şekilde kullanılmıştır.

5.1.1.1 VHDL Donanım Tanımlama Dili

Tasarımın büyüdüğü durumlar için HDL kullanılması kaçınılmazdır. Aşağıda HDL ile tasarımın nedenleri verilmiştir.

- 1) Tasarımların büyümesi nedeniyle sentezlenmesinin ve doğruluğunun zorlaşması.
- 3) Tasarımın firmalardan bağımsız ve kısa sürede gerçekleştirilmek istenmesi.
- 4) Sonradan yapılacak değişikliklerin kısa sürede etkinleştirilmesi.
- 5) Taşınabilirlik, test edilebilirlik, genişleyebilirlik ve modülerlik açısından çok iyi olması.

Akademik çalışmalar, kompleks tasarımlar ve savunma sanayiinde genellikle VHDL tercih edilirken, Verilog ise endüstriyel uygulamalarda tercih edilir. VHDL'in kullanılma amaçlarını aşağıdaki gibi sayabiliriz. (Navabi, 1998)

- 1) Dökümantasyon.
- 2) Doğrulama.
- 3) Sayısal sistemlerin gerçekleştirilmesi.

VHDL dili 1980'lerde geliştirilmiş olup, ilk versiyonu VHDL 87'dir. Daha sonra geliştirilerek VHDL 93 sürümü oluşturulmuştur. VHDL, IEEE 1076 standardı ile IEEE

tarafından standartlaştırılan ilk donanım tanımlama dilidir. VHDL devre sentezleme ve benzetimi için kullanılmaktadır. Tamamen benzetimi yapılmasına karşın tüm yapıları sentezlenememektedir. VHDL tanımlama dili günümüzde FPGA, CPLD ve ASIC tasarımlarında kullanılmaktadır.

5.1.2 İşlemci Tabanlı Tasarım Geliştirme

Günümüzde FPGA üreticileri soft-işlemci (IP olarak) veya hard-işlemci desteği ile FPGA yongalarına destek vermektedirler. İşlemci kullanarak mikroişlemcilerin esnekliği ile FPGA'in paralel işleme gücünü birleştirmek esas amaçlarındandır. Ayrıca tüm alt sistem bileşenleri üretici tarafından verildiğinden bunları işlemciye bağlamak kolay olacak olup tasarım süreci çok hızlanacaktır. Aşağıda en büyük iki FPGA üreticisinin sağladığı soft-işlemciler verilmiştir.

- 1) Microblaze (Xilinx)
- 2) Nios II (Altera)

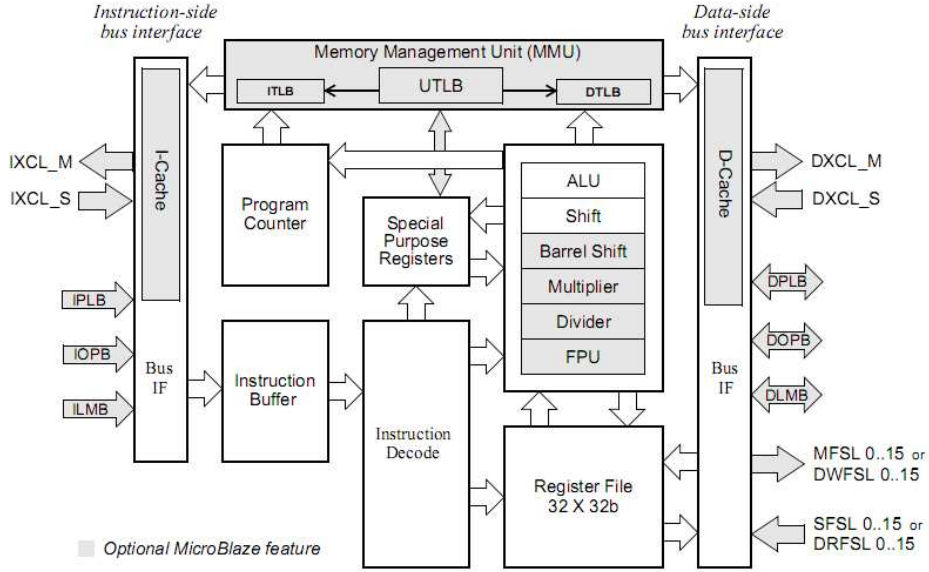
Xilinx firmasının Spartan 3E FPGA geliştirme kitine sahip olduğumuzdan dolayı Microblaze işlemcisi hakkında aşağıda detaylı bilgi verilmiştir. Hard-işlemci desteği ise Spartan 3E FPGA ailesinde mevcut değildir.

MicroBlaze, 32-bit Harvard mimarisi yapısındadır. Donanımsal çarpma (hardware multiply) ve bölme blokları (hardware divider) vardır. Kayan noktala aritmetiğe FPU birimi ile destek vermektedir. PLB, OPB ve FSL gibi kullanıcıya esnek veri yolları sunmaktadır. Aşağıda MicroBlaze'in yapısı verilmiştir. MMU (Memory Management Unit) ile modern işletim sistemlerini desteklemektedir. Donanımsal debug özelliği kullanıcıya yazılımı test etme imkanı vermektedir.

Microblaze işlemcisinin sabit olan yani konfigüre edilemeyen özellikleri ise aşağıda verilmiştir.

- 1) 32 adet 32-bit genel amaçlı yazmaç.
- 2) 32-bit komut kümesi. (2 adresleme modu ve 3 operand desteği)
- 3) 32-bit adres yolu.

Microblaze işlemcisinin detayları Şekil 5.2'de verilmiştir. Buradan da görüldü gibi kullanıcının konfigüre edebileceği bir yapı mevcuttur.



Şekil 5.2 Microblaze işlemcisinin detayları.

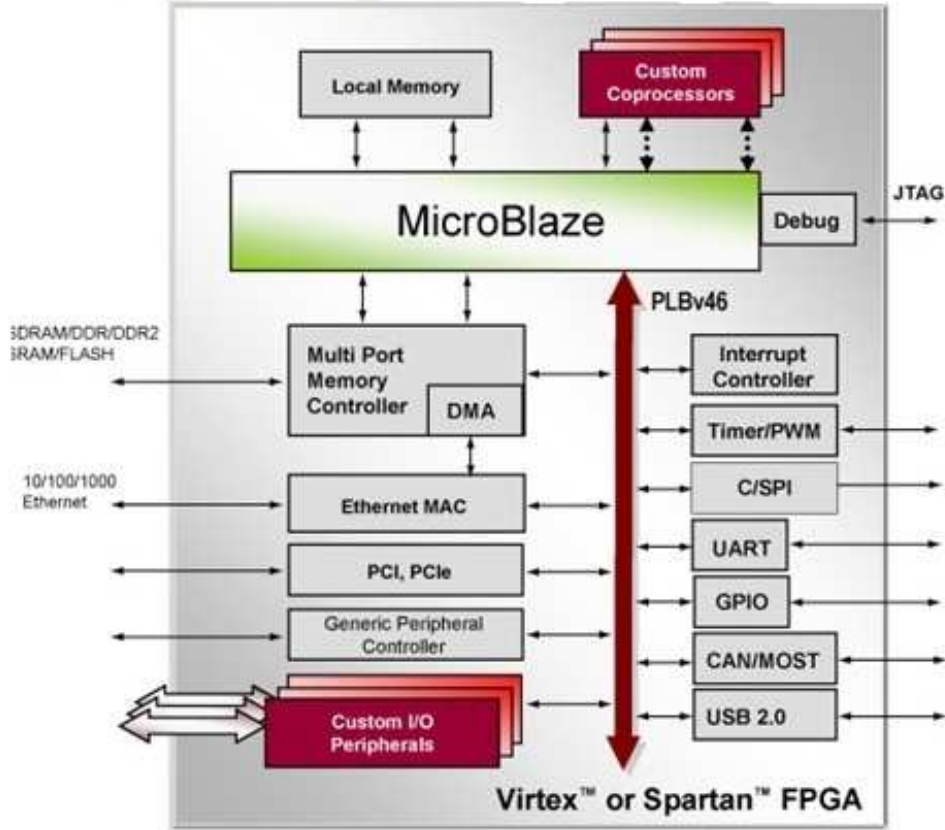
Microblaze işlemcisi, Xilinx firmasının sağladığı EDK(Embedded Development Kit) ile beraber gelmektedir ve EDK lisansı ile kullanılmaktadır. EDK lisansı olduğu zaman sınırsız sayıda kullanılabilir yani sayı sınırı yoktur. Çoklu işlemci desteği vardır. Yani bir FPGA yongasında birden fazla işlemci yerleştirilerek bunlar birbirleri ile konuşturulabilmekte ki, bu işlem yükünün çok fazla olduğu uygulamalar için önemlidir. FPGA içerisinde ne kadar bir alan kapladığı ve ne kadar kaynak tükettiği ise tamamen kullanıcının işlemciyi konfigüre etmesiyle ilgilidir. Çalışma frekansı ise kullanıcının hangi tür FPGA yongası kullandığına bağlıdır. Donanımsal FPU ise IEEE-754 uyumlu olup, uygulamaya göre değişmekle beraber bazı uygulamalarda 40 kata varan performans artışı sağlanmaktadır. Microblaze işlemcisi çok çeşitli hafıza yongalarına destek vermektedir. EDK ortamı çok esnek yapıya sahip olmakla beraber gerek normal ayıklama araçları gerekse de ChipScope tabanlı lojik ayıklama özelliklerine destek sunması önemli avantajlarındandır.

Microblaze tabanlı sistem geliştirmenin kullanıcıya sağladığı kolaylıklar arasında aşağıdaki özellikleri sayabiliriz.

- 1) Doğruluğu kanıtlanmış ve uzun yıllardır geliştirilmiş CPU yapısı.
- 2) CPU ile kullanılacak çeşitli çevresel birimlerin kullanıcıya sunulması.
- 3) C dili ile yazılım geliştirebilme.
- 4) Linux tabanlı modern işletim sistemi desteği.
- 5) FPGA'in en büyük yeteneği olan paralel işleme için kullanıcıya özel veri yolu.

Bu özelliklerden en sonuncusu ise çok önemli olmaktadır. Microblaze işlemcisinin kullanıcıya sunduğu veri yolları aşağı verilmiştir.

- 1) FSL(Fast Simplex Link) kanalları.
- 2) PLB v46 veri yolu.



Şekil 5.3 Microblaze işlemcisi ile sistem kurma.

Şekil 5.3'den görüldü gibi kullanıcının kendi uygulaması için tasarladığı işlemci için veri yolu vardır. Bu veri yolu FSL kanalları olup, gecikmesi çok düşük olup veri yoğunluğunun çok fazla olduğu uygulamalarda tercih edilmektedir. Kullanıcının kendi tanımladığı blokları Microblaze'e bağlamasının diğer bir yolu ise PLB v46 veri yoluna ilişkilendirilmesidir. Bu yol ise FSL kanalları kadar hızlı olmamakla beraber Microblaze tabanlı sistemlerde kullanılmaktadır.

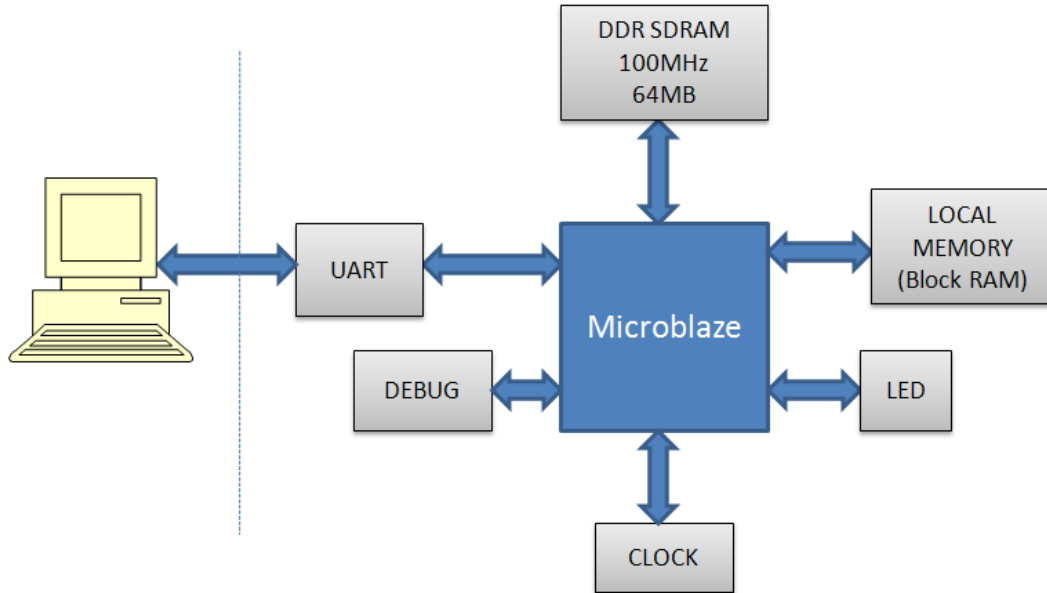
Kullanıcı bu işlemciyi JTAG yoluyla ayıklama işlemi gerçekleştirebilir. Xilinx EDK ortamının kullanıcıya sağladığı IP blokları Şekil 5.3'de görülmektedir. Bu hazır olan IP yapıları kullanıcının istediği gömülü sistemi çok hızlı bir şekilde gerçekleştirmesine olanak vermektedir.

5.2 FPGA Tabanlı Araç Tespit Sistemi

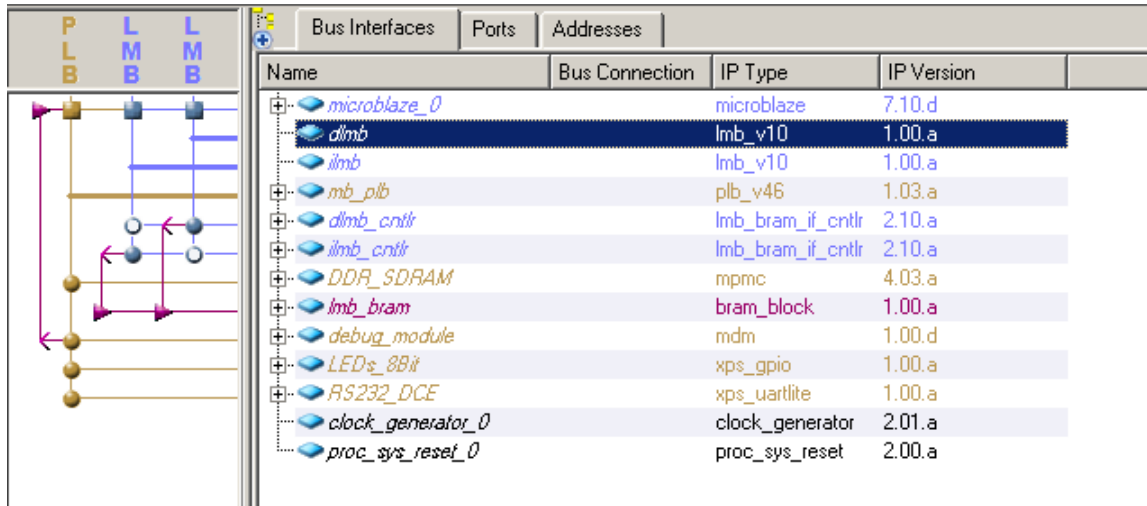
Araç tespiti adımlarının belirlenmesi ve ilgili başarımlar oranları 3.bölümde anlatılmıştır. Bu adımların gömülü sisteme aktarılması ve kurulan mimari aşağıda anlatılmıştır. Microblaze tabanlı oluşturulan sistem Xilinx EDK programında tasarlanmıştır. Video verisinin aktarılması ve video işleme sonuçlarının gösterilmesi için PC tabanlı kullanıcı arayüzü Borland C++ Builder ortamında tasarlanmıştır.

5.2.1 Microblaze Tabanlı Sistem Geliştirme

Spartan 3E geliştirme kiti üzerinde görüntü işleme amaçlı oluşturulan sistem EDK ortamında Microblaze tabanlı olup Şekil 5.4’de bileşenleri verilmiştir. Uart bileşeni PC programı ile veri alışverişinde kullanılır. PC programı tarafından UART ile gönderilen video görüntülerinin yazılmasında ve işlenmiş olan görüntülerin saklanması için kullanılan ve dış dünyada bulunan DDRSDRAM’i kontrolü için DDR SDRAM bloğu mevcuttur. LOCAL MEMORY bloğu ise derlenmiş program’ı içermektedir. LED ise yazılım aşamalarının gösterilmesi amaçlıdır. Diğer bir blok olan DEBUG ise yazılımsal ayıklama içindir. Şekil 5.5’de oluşturulan bu sistemin yol bağlantı arayüzleri gösterilmiştir. Şekil 5.6’da bu sistemin adres haritası verilmiştir



Şekil 5.4 Microblaze işlemcisi ile kurulan sistem.



Şekil 5.5 Microblaze işlemcisi ile kurulan sistemin birbirleri ile oluşturulan bağlantıları.

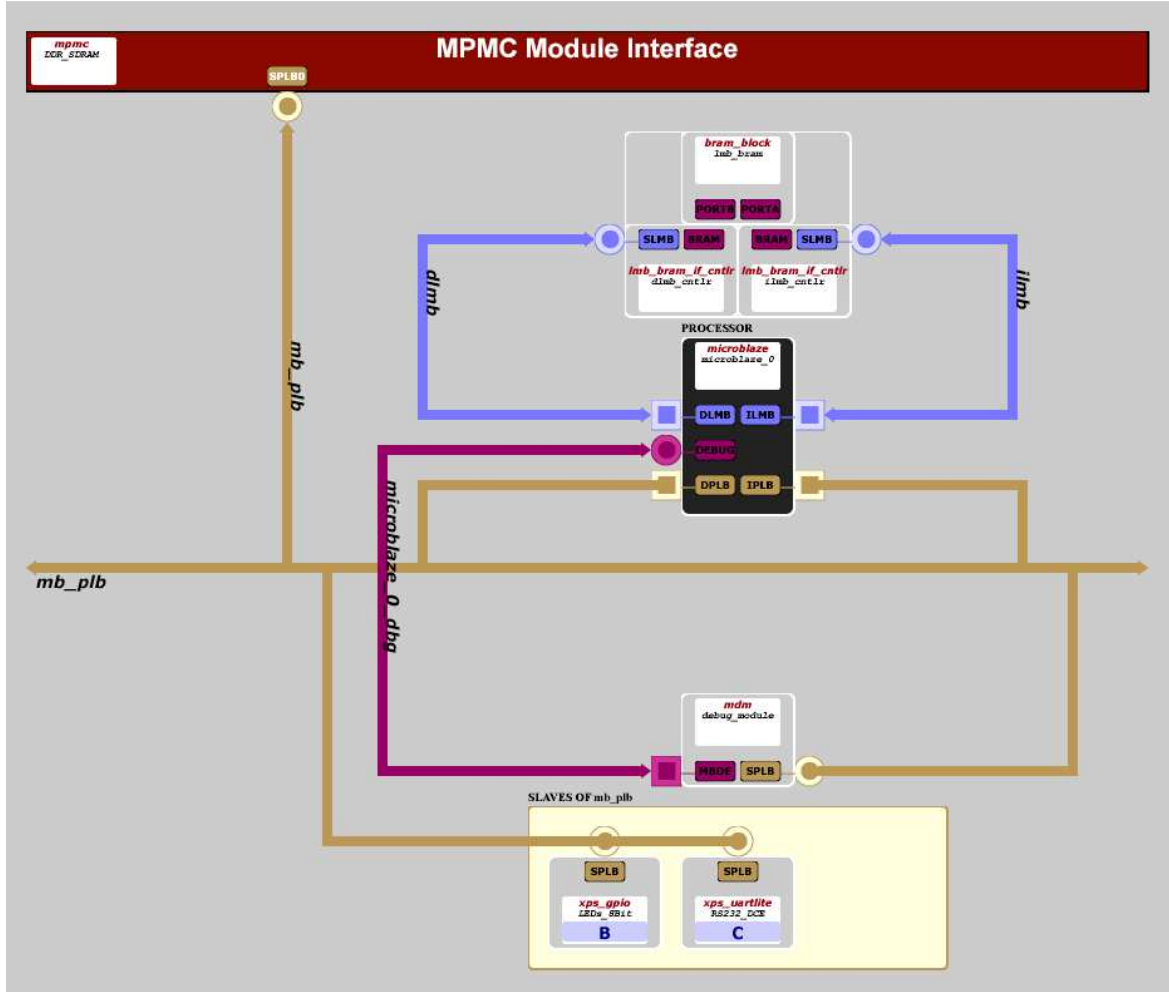
Bağlantıların detaylı anlamları ise aşağıdaki çizelgede verilmiştir.

Çizelge 5.1 Microblaze işlemcisi ile kurulan sistemin detaylı açıklaması

Blok Adı	Anlamı
<i>Microblaze_0</i>	Microblaze işlemcisi
<i>dlmb</i>	İşlemci belleği için veri yolu
<i>ilmb</i>	İşlemci belleği için komut yolu
<i>mb_plb</i>	Çeşitli blokların bağlanması için PLB yolu.
<i>dlmb_cntrl</i>	İşlemci belleği kontrol bloğu (Veri)
<i>ilmb_cntrl</i>	İşlemci belleği kontrol bloğu (Komut)
<i>DDR_SDRAM</i>	PLB yolu için DDRSDRAM bloğu
<i>Lmb_bram</i>	İşlemci belleği (Veri ve komut)
<i>Debug_module</i>	Yazılımsal ayıklama için gerekli blok
<i>LEDs_8Bit</i>	Yazılım aşamalarını göstermek için tanımlanan LED kontrol bloğu

Instance	Name	Base Address	High Address	Size	Bus Interface(s)	Bus Connection
mb_plb	C_BASEADDR			U	Not Applicable	
dlmb_cntrl	C_BASEADDR	0x00000000	0x00003fff	16K	SLMB	dlmb
ilmb_cntrl	C_BASEADDR	0x00000000	0x00003fff	16K	SLMB	ilmb
LEDs_8Bit	C_BASEADDR	0x81400000	0x8140ffff	64K	SPLB	mb_plb
RS232_DCE	C_BASEADDR	0x84000000	0x8400ffff	64K	SPLB	mb_plb
debug_module	C_BASEADDR	0x84400000	0x8440ffff	64K	SPLB	mb_plb
DDR_SDRAM	C_MPMC_BASEADDR	0x8c000000	0x8cffffff	64M	SPLB0	mb_plb

Şekil 5.6 Sistemin adres haritası.



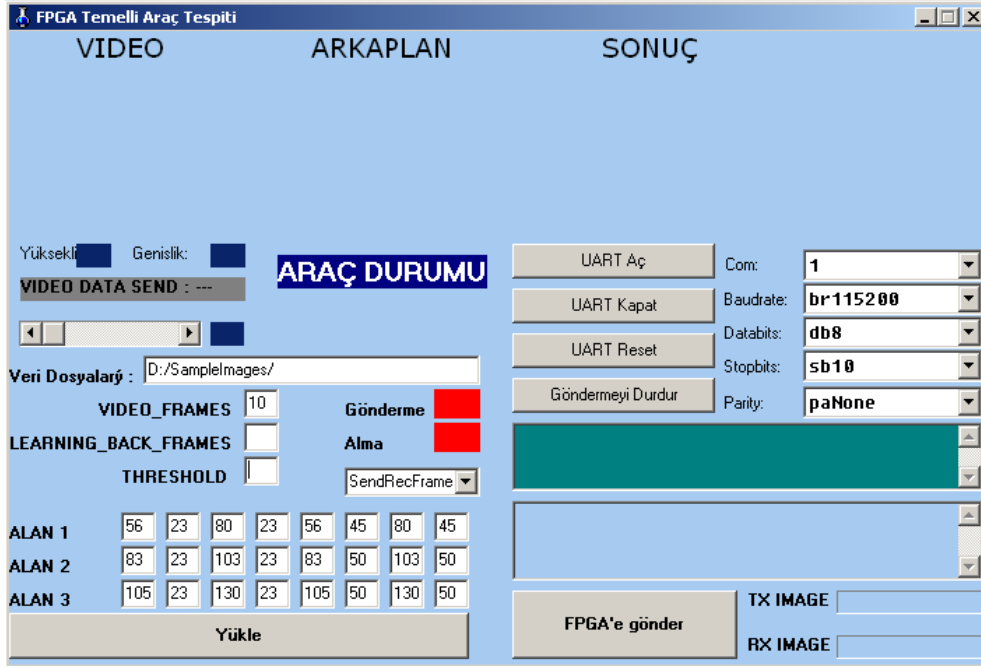
Şekil 5.7 Sistemin blok diagram olarak raporunun gösterilmesi.

MPMC (Multi Port Memory Controller) arayüzü dış dünyadaki hafızaların Microblaze işlemcisine bağlanması için gerekli olan yapı olmakta olup, işlemciye PLB yolundan bağlanmıştır.

Daha önce MATLAB ortamında belirlenen araç tespit sistemi adımları C dilinde yazılmış olup EDK ortamında derlenmiştir. Bu yazılım ilgili adımları gerçekleştiren kısımlardan oluşmakta olup ayrıca UART arayüzü ile gelen verilerin DDRSDRAM'e yazılması ve işlenen görüntü verilerinin tekrar UART ile gönderilmesini gerçekleştirmektedir.

5.3 PC Programı

PC programı Borland C++ Builder 6.0 kullanılarak yazılmıştır. Seri port komponenti olarak TMSA32 kullanılmıştır. Aşağıda yazılımın görüntüsü gösterilmektedir. Yazılım belirlenen dosyadan görüntüleri okumakta, ilgili tüm parametreleri FPGA'ye göndermekte ve geri gönderilen görüntü ve araç tespit bilgilerini ekranda göstermektedir.



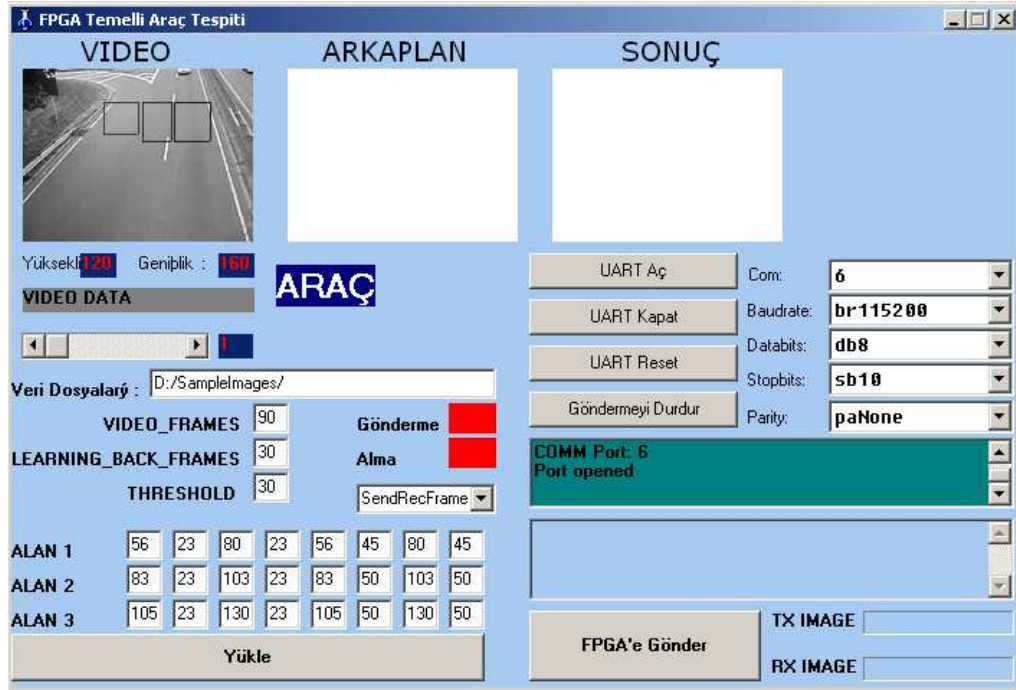
Şekil 5.8 PC programının görünüşü.

PC programındaki sağ kısım RS232 arayüzünün çeşitli ayarlarını içermekte olup, görüntü akışını gösteren özelliklere sahiptir. Alan tanımlamaları ise ilgili görüntünün neresinde araç tespiti yapılacağına ilişkin piksel koordinatların ayarlamakta kullanılmaktadır. Analizini yaptığımız görüntülerde 3 farklı tanımlanan alanda araç tespiti yapılacaktır. Görüntü dosyalarının yeri seçilebilmektedir. Ayrıca diğer parametrelerin anlamları aşağıda verilmiştir.

Çizelge 5.1 Microblaze işlemcisi ile kurulan sistemin detaylı açıklaması

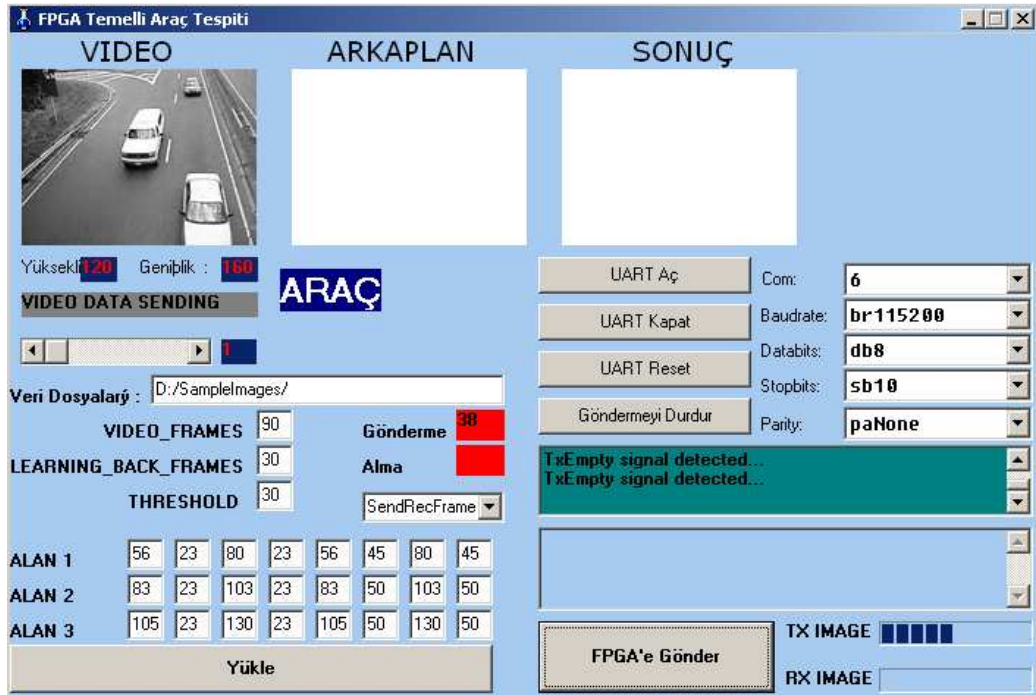
Parametre Adı	Açıklaması
<i>VIDEO_FRAMES</i>	Görüntü işleme için kullanılacak olan toplam görüntü sayısı
<i>LEARNING_BACK_FRAMES</i>	Arkaplan çıkarı için hesaba katılacak görüntü sayısı
<i>THRESHOLD</i>	Eşikleme değeri

Şekil 5.9'da programın ilk ayarları yapılmış olup FPGA'e gönderime hazır durumundaki görüntüsü verilmiştir. Buradan da görüldüğü gibi tanımlanan alanların ilgili görüntüdeki karşılığı gösterilmiş olup , 90 görüntü sırasıyla ayarlanılan RS232 hızında gönderilmektedir. Arkaplan öğrenimi için 30 görüntü belirlenmiş ve eşikleme için 30 değeri verilmiştir. Görüntü aktarımı hızı 115200 baud hızında olacaktır.



Şekil 5.9 PC programının ilk ayarlarının yapılmış durumu.

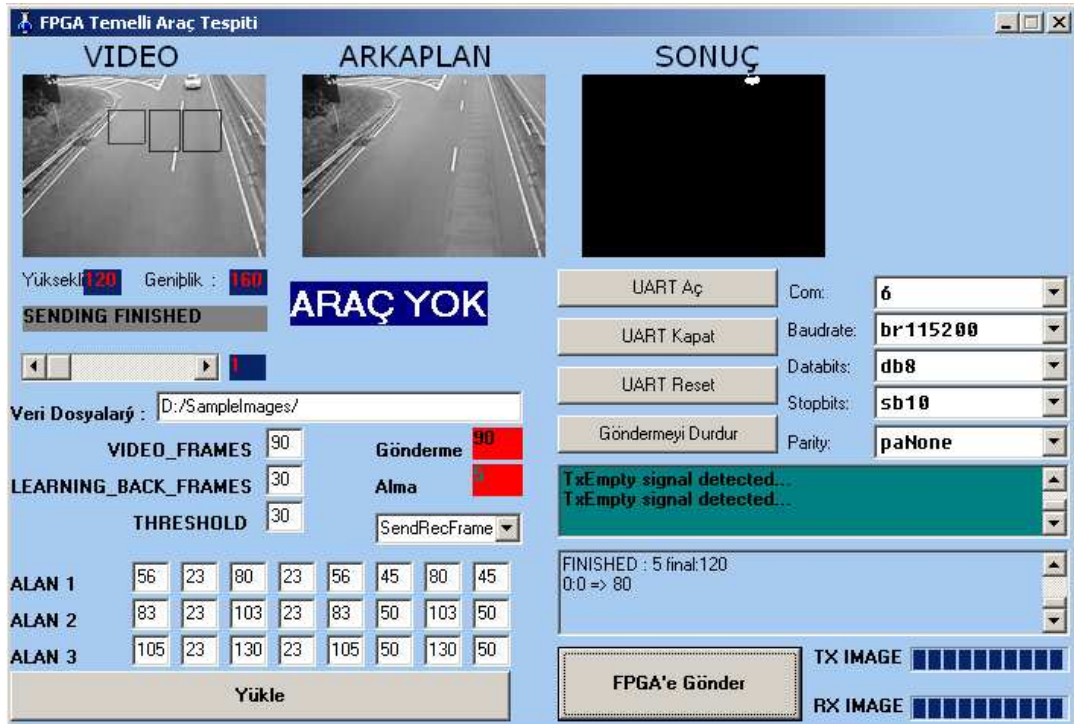
Kullanıcı arayüzünde araç tespiti için seçilmiş olan alanlar ekranda gösterilmektedir. Görüntüler gönderilmeye devam ettikçe gönderilen görüntüler yine bu ekranda gösterilmiş olup gönderilme işlemi 'FPGA'e Gönder'e basılınca başlar ve etkileşimli olarak kullanıcılara ilerleme gösterilir. Şekil 5.10'da aktarılma esnasında PC programı gösterilmiştir. Gönderilen görüntü sayısı yine bu programda gösterilir.



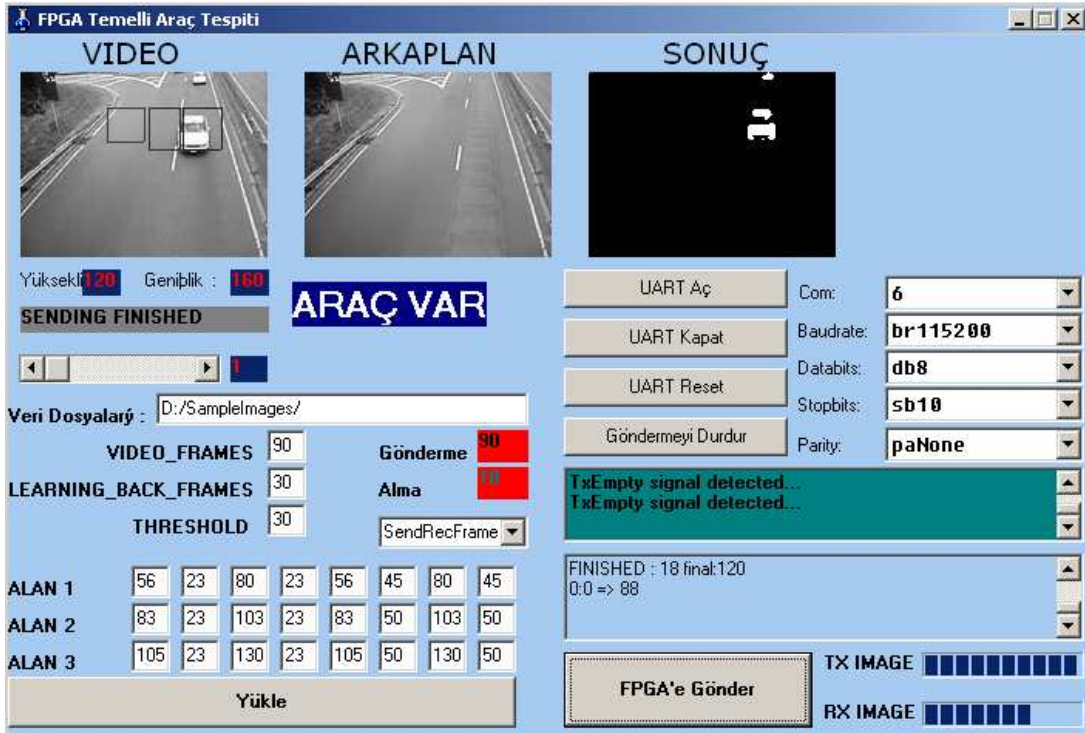
Şekil 5.10 Görüntülerin gönderilmesi.

Şekil 5.11’de ise arkaplan bilgisinin FPGA tarafından çıkarılması ve PC’ye göndermesi sonucu verilmiştir. Araç bilgisi ise varken ‘ARAÇ VAR’ ve yokken ‘ARAÇ YOK’ şeklinde ekrana bastırılmaktadır. PC programının en sağındaki alan FPGA tarafından hesaplanan en son aşamadaki yani tüm adımlardan geçmiş görüntü bilgisini vermektedir. Bu alan bilgileri sırasıyla gönderilir. Ayrıca ‘ARAÇ VAR’ ve ‘ARAÇ YOK’ bilgisi de FPGA tarafından her görüntünün sonuna eklenerek gönderilir. Şekil 5.12’de tanımlanmış alanlarda araç varken, şekil 5.13’de ise araç yokken ilgili PC programının görüntüleri verilmiştir.

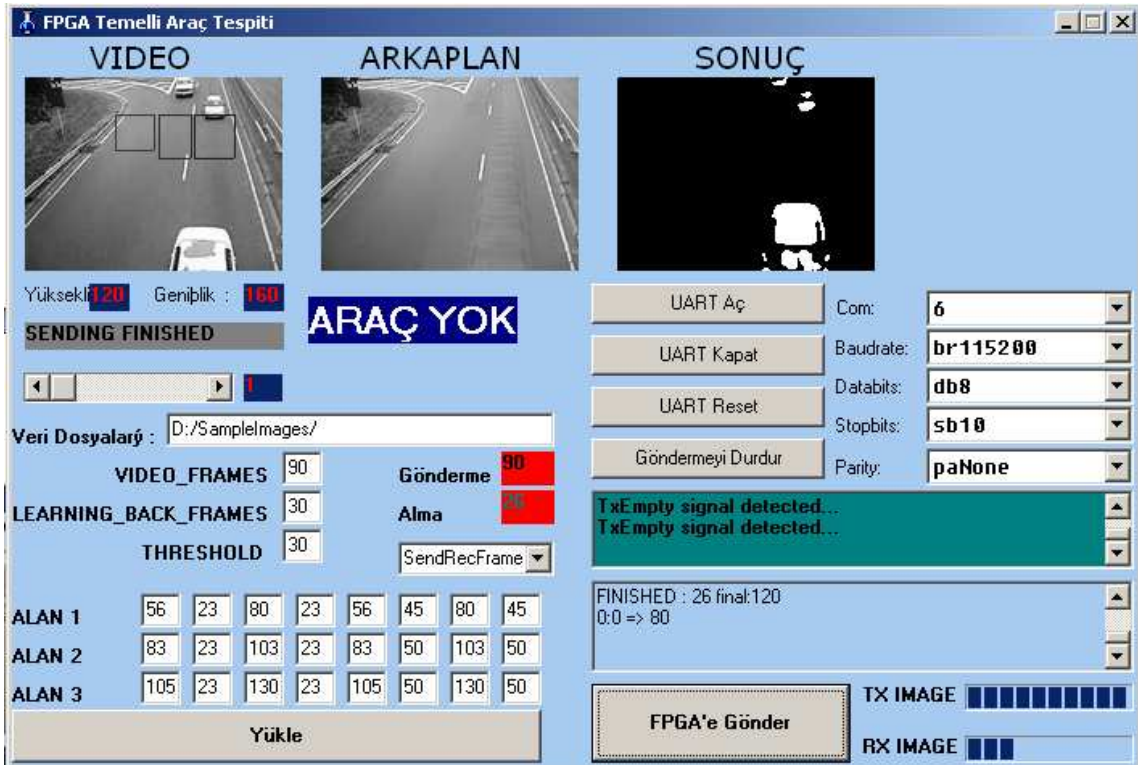
Bölüm 3.7’deki eşik yüzdeleri uygulandığında aynı başarımlar yüzdeleri FPGA yazılımında da elde edilmiştir.



Şekil 5.11 Arkaplanın hesaplanıp PC programına gönderilmesi.



Şekil 5.12 Araç belirlenmiş alanlarda mevcutken PC programı görüntüsü.



Şekil 5.13 Araç belirlenmiş alanlarda mevcut değilken PC programı görüntüsü.

6. SONUÇLAR VE ÖNERİLER

Bu çalışmada yol görüntülerinin analizi gerçekleştirilerek içerisindeki araç bilgileri algılanmıştır. Araç tespiti öncelikle kaydedilmiş video verisi üzerinden bilgisayar ortamında yapılmış, bu amaç için tüm adımlar ve başarımları belirlenmiştir. Daha sonra ise mevcut FPGA geliştirme donanımında bu adımların aktarılması için bir PC programı yazılmış olup, FPGA içerisinde işlemcili bir platform kurulmuştur. Bilgisayar ortamındaki başarımların FPGA temelli kurulan gömülü sistemde de sağlanmıştır.

Video görüntülerindeki araçların tespit hızı karşılaştırıldığında FPGA'nin PC'ye göre çok yüksek hıza sahip olduğu görülmüştür. PC yerine FPGA temelli sistemin kullanımı maliyet açısından büyük avantajlar sunmaktadır. Bu tür görüntü tabanlı trafik sinyalizasyon sisteminin otomasyonu ile trafik ışıklarında araçların bekleme süresi otomatik olarak ayarlanabilecektir. Bunun sonucunda zaman kazanımı, çevre kirliliği ve gereksiz yakıt tüketimi gibi bir çok konuda önemli avantajlar sağlanmış olacaktır.

İşlemci temelli kurulacak sisteme ek olarak desteklenen hızlı arayüzlerden ilgili algoritma adımlarını gerçekleştirecek yardımcı işlemcinin HDL ile yazılması neticesinde güçlü ve esnek bir gerçek zamanlı görüntü işleme sistemi yapılmış olacaktır. Böyle kurulan bir görüntü işleme platformu trafik sistemleri, endüstriyel, tıp ve savunma alanlarında görüntü işleme uygulamalarına cevap verecektir.

KAYNAKLAR

- Atkociunas, E., Blake, R., Juozapavicius, A. ve Kaziminianec, M. (2005), "Image processing in Road Traffic Analyses", *Nonlinear Analysis:Modelling and Control*, Volume 10, No:4, 315-332.
- Aydın, A., (2005), "FPGA Yonga Mimarisi ve Kullanımı", Bitirme Ödevi, Süleyman Demirel Üniversitesi, Isparta.
- Bhatti, Z. E., (2006) "Morphological Image Processing Based Road Traffic Control Systems".
- Blackham, B., (2006) "The Development of a Hardware Platform for Real-time Image Processing." School of Electrical, Electronic and Computer Engineering, The University of Western Australia.
- Baş, E., (2007) "Road and Traffic Analysis from Video", Master of Sciences Thesis, Koç University, August 2007, İstanbul.
- Bramberger, B.,Pflugfelder,R.P., Maier, A., Rinner, B., Strobl, B. ve Schwabach, H., "A Smart Camera for Traffic Surveillance"
- Cahou, Vu Ky. ve Vinh, Truong Quang., "Embedded Image Processing system on FPGA".
- Chin, L., (2006) "FPGA-based Embedded Vision Systems." School of Electrical, Electronic and Computer Engineering, The University of Western Australia.
- Çuhadaroğlu, B., (2008), "ITS Uygulamaların Ankara'da Kullanımı ", Ankara.
- Deng, L. Y., Tang,N.C., Lee, D., Wang, C.T. ve Lu, M.C., (2005), "Vision Based Adaptive Traffic Signal Control System Development", 19th International Conference on Advanced Information Networking and Application (ANIA'05).
- Gonzales, C. ve Woods, R.E. (2002), *Digital Image Processing*, Prentice Hall, New Jersey.
- Higashikuba, M., Hinenoya, T. ve Takeuchi, K., "Traffic Queue Length Measurement Using an Image Processing Sensor".
- Johnston, C T., Gribbon , K T. ve Bailey, D G. "Implementing Image Processing Algorithms on FPGAs".
- McCurry, P., Morgan, F. Ve Kilmartin, L. "Xilinx FPGA implementation of a pixel processor for object detection applications".
- Motkowski, M., (1991), "Traffic Signal Control Systems".
- Özcan, A.R. (2009), "Gerçek Zamanlı Lineer Görüntü İşleme Algoritmalarının FPGA'de gerçekleştirilmesi", Bitirme Tezi\Projesi, Yıldız Teknik Üniversitesi, İstanbul.
- Sakakibara, H., Aoki, M. ve Matsumoto, H., (2005) "Advanced Traffic Control System Installed in Phukey City, Kingdom of Thailand", SEI Technical Review, Information & Communication Systems, Number 60, June 2005, Wien, 54-58.
- Siyal, M., (2004), "A Novel Image Processing Based Approach For Real-Time Road Traffic Applications", IEEE.

İNTERNET KAYNAKLARI

www.ieeexplore.ieee.org

www.ovt.com

www.mathworks.com/matlabcentral

www.modot.mo.gov/stlouis/links/signalcameras.htm

www.xilinx.com

www.xilinx.com/tools/microblaze.htm

www.wikipedia.org

www.digilentinc.com

ÖZGEÇMİŞ

Doğum tarihi	06.01.1981	
Doğum yeri	Samsun	
Lise	1994-1999	Atatürk Sağlık Meslek Lisesi, KONYA
Lisans	1999-2004	Yıldız Üniversitesi Elektrik ve Elektronik Fak. Elektronik ve Haberleşme Mühendisliği Bölümü
Yüksek Lisans	2007-2010	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik Müh. Anabilim Dalı, Kontrol ve Otomasyon Programı
Çalıştığı kurum	2004-Devam	TÜBİTAK UEKAE, Uzman Tasarım Müh.