

**TEK KARTLI BİLGİSAYARDA ÇALIŞACAK
BİR LINUX İŞLETİM SİSTEMİNİN DERLENMESİ
VE GERÇEK ZAMANLI ÇALIŞAN BİR UYGULAMANIN
OLUŞTURULMASI**

Serhat TATLIDİL
Yüksek Lisans Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı
Ocak-2010

JÜRİ VE ENSTİTÜ ONAYI

Serhat TATLIDİL'in "Tek Karthı Bilgisayarda Çalışacak Bir Linux İşletim Sisteminin Derlenmesi ve Gerçek Zamanlı Çalışan Bir Uygulamanın Oluşturulması" başlıklı Elektrik-Elektronik Mühendisliği Anabilim Dalındaki, Yüksek Lisans Tezi 28.12.2009 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Adı-Soyadı	İmza
Üye (Tez Danışmanı) : Yard. Doç. Dr. HAKAN GÜRAY ŞENEL
Üye : Yard. Doç. Dr. ATAKAN DOĞAN
Üye : Yard. Doç. Dr. ÖZGÜR YILMAZEL

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... tarih ve sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

TEK KARTLI BİLGİSAYARDA ÇALIŞACAK BİR LINUX İŞLETİM SİSTEMİNİN DERLENMESİ VE GERÇEK ZAMANLI ÇALIŞAN BİR UYGULAMANIN OLUŞTURULMASI

Serhat TATLIDİL

**Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik – Elektronik Mühendisliği Anabilim Dalı**

**Danışman : Yard.Doç. Dr. Hakan Güray ŞENEL
2009, 82 sayfa**

Bu çalışmanın amacı bir tek kartlı bilgisayar için özel bir işletim sistemi oluşturmaktır. İşletim sisteminin Linux tabanlı olmasına karar verilmiş ve kişisel bilgisayarda yer alan Debian GNU/Linux dağıtımından yararlanılmıştır. İşletim sisteminin tek kartlı bilgisayara aktarılma işlemi tamamlandıktan sonra karta bir Internet kamerası görüntü işleme amacıyla takılarak gömülü bir sistem meydana getirilmiştir. Yeni gömülü sistemde çalışmak üzere cepheden çekilmiş bir insan yüzünün gerçek zamanlı olarak tespit edilmesi ve tespit edilen yüzün takip edildiği bir uygulama gerçekleştirilmiştir. Yüz tespit yöntemi olarak üç aşamadan oluşan bir nesne tespit algoritmasının kullanıldığı çalışmada, yüz tespitinin başarılı olmasının ardından tespit edilen yüzün uyarlanabilir bir nesne takip algoritması ile takip edildiği bu uygulama C programlama dili ile kodlanmıştır. Sistemin performansı ve uygulama sırasında karşılaşılan sorunlar da tartışılmıştır.

Anahtar Kelimeler: Tek Kartlı Bilgisayar, Gömülü Sistemler, Yüz Tespit, Debian GNU/Linux, Yüz Tespit.

ABSTRACT

Master of Science Thesis

**BUILDING A LINUX OPERATING SYSTEM FOR A SINGLE BOARD
COMPUTER AND IMPLEMENTING A REAL-TIME APPLICATION**

Serhat TATLIDİL

**Anadolu University
Graduate School of Sciences
Electrical & Electronics Engineering Program**

**Supervisor: Assist. Prof. Dr. Hakan Güray ŞENEL
2009, 82 pages**

The main goal of this study was to build a special Operating System for a Single Board Computer. Application of a Linux-based Operating System was decided and Debian GNU/Linux which is installed on a personal computer was used. After porting of the operating system to a Single Board Computer, a new embedded system was created by plugging a USB webcam into the board for image processing purposes. A real-time frontal face detection and tracking application was implemented to work on this new embedded system. A three phase object detection algorithm was used as the face detection method in this work. After detecting the face, the detected face was tracked by using an adaptive object tracking algorithm that was coded in C programming language. Performance issues and problems encountered in the application process are presented.

Keywords: Single Board Computer, Embedded Systems, Debian GNU/Linux,
Face Detection, Face Tracking.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİLLER DİZİNİ	vi
SİMGELER ve KISALTMALAR DİZİNİ	viii
1 GİRİŞ	1
2 YÜZ TESPİTİ, YÜZ TAKİBİ ve GÖMÜLÜ SİSTEMLER	4
2.1 Yüz Tespit Çalışmaları.....	4
2.2 Yüz Takip Çalışmaları	6
2.3 Gömülü Sistemler	8
2.3.1 Gömülü Sistemlerde Yüz Tespit ve Takip Uygulamaları	11
3 YÜZ TESPİT VE TAKİBİ İÇİN KULLANILAN YÖNTEMLER	18
3.1 VIOLA ve JONES Yüz Tespit Algoritması.....	18
3.1.1 Tümevsel Görüntü	19
3.1.2 Öğrenme Algoritması.....	20
3.1.3 Ardarda Sıralı Sınıflandırıcı	21
3.2 CAMSHIFT Yüz Takip Algoritması	21
3.2.1 Renk Olasılık Dağılımları	23
3.2.2 Algoritma	25
4 UYGULAMANIN GENEL YAPISI	27
4.1 Donanımların ve İşletim Sisteminin Seçilmesi	27
4.2 Geliştirme Ortamının Hazırlanması	30
4.3 Özel Uygulama İçin Kullanılacak Yöntem ve Araçların Belirlenmesi. 31	
5 ANA MAKİNENİN HAZIRLANMASI	34
5.1 DEBIAN GNU/LINUX Kurulumu	34
5.1.1 Kurulum Sırasında Ağ Yapılandırması.....	35

5.2	Kurulum Sonrası Yapılması Gereken Ayarlar	36
5.2.1	Debian GNU/Linux'ta Çekirdek Derlenmesi	36
5.2.2	Kurulum Sonrası Ağ Yapılandırılması	37
5.2.3	Usb Cihaz Ayarlamaları	38
5.2.4	OV51X-JPEG Sürücü Kurulumu	40
5.2.5	OpenCV Kütüphanesi Kurulumu	43
5.2.6	Sistemden Görüntünün Alınması İçin Gerekli Ayarlamalar	44
5.2.7	Sistemde Çalışacak Çekirdeğin Derlenmesi	47
5.2.8	Sistemde Çalışacak Çekirdeğe Internet Kamerasına Ait Modülün Eklenmesi	48
5.2.9	Code::Blocks Kurulumu	48
6	SBC'NİN HAZIRLANMASI	50
6.1	SBC'yi Disketten Başlatma	50
6.2	DOM'un Hazırlanması	52
6.3	Temel Linux Ayarları	54
6.3.1	Temel Dizinlerin Oluşturulması	54
6.3.2	Dizinelere Erişimlerin Düzenlenmesi	54
6.3.3	/dev Dizini Altında Aygıtların Oluşturulması	54
6.3.4	Kütüphanelerin Taşınması	55
6.3.5	Temel Uygulamaların Busybox İle Sisteme Kurulması	56
6.4	Sistemin Kendi Başına Yüklenebilmesi İçin Gerekli Ayarlamalar	57
6.4.1	SBC' ye Özgün Çekirdeğin Oluşturulması	57
6.4.2	Sistemin Başlatılmasından Sonra Kullanılacak Programların Ayarlanması	58
6.4.3	Önyükleyicinin Oluşturulması	60
6.5	Sistem Kendi Başına Yüklendikten Sonra Yapılacak Ayarlamalar	62
6.5.1	Sistem Kullanıcıları ve Gruplar	62
6.5.2	Sistemin Ağ'a Bağlanması	63
6.5.3	Sisteme Bir Internet Daemon (inetd) Eklenmesi	65
6.5.4	Sisteme Ftp Sunucu Eklenmesi	66
6.5.5	Sisteme Telnet Sunucu Eklenmesi	67

7	YÜZ TESPİT VE TAKİP UYGULAMASI.....	69
7.1	Uygulamanın Derlenmesi ve Çalıştırılması	69
7.2	Uygulama Performansı	70
8	SONUÇLAR VE ÖNERİLER	73
	KAYNAKLAR	75
	EK: CD İÇERİĞİ.....	82

ŞEKİLLER DİZİNİ

1-1: Yüz tespitine ait örnek resim [70]	1
2-1: Öznitelik değişimsiz yaklaşım örneği [69].....	5
2-2: Rowley'nin yüz tespit algoritması [70]	6
2-3: Ten rengine dayalı örnek resim [1].....	7
2-4: Birchfield örnek yüz takip resimleri [43]	7
2-5: Örnek resim dizisi – Maurer ve Malsburg yöntemi [4].....	8
2-6: Örnek resim dizisi – Matthews ve Baker [61].....	8
2-7: Örnek resim dizisi – Williams ve ark. [54]	8
2-8: Örnek Ana Makine / Hedef Bağlantısı [16].....	9
2-9: Örnek Geliştirme Aracı [55, 56].....	9
2-10: Chu'nun kullandığı donanım [14]	12
2-11: Chu'nun algoritması [14].....	12
2-12: Sun'ın 1 boyutlu HMM'si [8].....	13
2-13: CAMEO sistemi [9].....	14
2-14: CAMEO ile ilgi alanında yer alan yüzün tespiti [9].....	14
2-15: CAMEO ile yüze göre hareketin sınıflandırılması [62]	15
2-16: McCready'nin yöntemi [66]	16
2-17: Pun'un sistem diyagramı ve düzeneği [67]	16
2-18: Lee'nin önerdiği yöntem [71].....	17
3-1: Dikdörtgen öznitelikler	18
3-2: Tümlevsel görüntü değerinin hesaplanması	20
3-3: AdaBoost tarafından seçilen ilk iki öznitelik	21
3-4: Ardarda sıralı sınıflandırıcı.....	21
3-5: CAMSHIFT algoritması	23
3-6: RGB renk küpü	24
3-7: HSV renk sistemi.....	24
3-8: Yüz takip ve ten rengi olasılık dağılım resmi.....	26
4-1: Geliştirme yöntemi	27
4-2: ICOPTECH – Vortex86 tek kartlı bilgisayar	29
4-3: Geliştirme ortamı blok şeması.....	30
4-4: Geliştirme ortamı	31

4-5: Projede oluşturulan gömülü sistem.....	31
4-6: Genişletilmiş öznitelikler.....	32
4-7: Projeye özel algoritma	33
5-1: Debian kurulum ekranı	34
5-2: Debian ağ yapılandırma ekranı.....	35
5-3: /etc/network/interfaces dosyası.....	37
5-4: /etc/hosts dosyası	38
5-5: Ana Makinedeki USB aygıtlar hakkında bilgi edinilmesi.....	40
5-6: Sürücü yüklendikten sonraki ov51x-jpeg mesajı.....	41
5-7: /dev/video içeriği	42
5-8: xawtv uygulaması ile modül kurulumunun test edilmesi	42
5-9: Grafiksel karşılayıcı ayar penceresi.....	44
5-10: “Deny TCP connections to Xserver” işareti	45
5-11: /etc/X11/xinit/xserverrc dosyası	46
5-12: X Window erişim hakkının verilmesi.....	46
5-13: Code::Blocks	49
6-1: Sistem BIOS Ekranı.....	51
6-2: Advanced CMOS SETUP Ekranı.....	51
6-3: /etc/init.d/rcS dosyası.....	59
6-4: /etc/fstab dosyası.....	60
6-5: /etc/lilo.conf dosyası.....	61
6-6: /etc/nsswitch.conf dosyası	63
6-7: Ethernet yapılandırma kontrolü	64
6-8: Telnet ile ana makineden sisteme bağlanma	68
7-1: Örnek resimler	71
7-2: Yüz takip performansı	72
7-3: Sistem CPU ve hafıza kullanımı.....	72
8-1: Sistem disk kullanım miktarı.....	73

SİMGELER ve KISALTMALAR DİZİNİ

i (x, y)	Resmin x,y noktasındaki değeri
ii (x, y)	Resmin x,y noktasındaki tümleysel görüntü değeri
s (x, y)	Resmin x,y noktasındaki kümülatif satır toplam değeri
h_j(x)	Sınıflandırıcı fonksiyon
f_j	Öznitelik değeri
θ_j	Eşik değeri
p_j	Parity değeri
M_n	n'inci moment
s	Takip penceresi ebadı
A	Amper
AC	Alternatif Akım
CAMEO	Camera Assisted Meeting Event Observer
CAMSHIFT	Sürekli Uyarlanabilir Ortalama Kayma Algoritması - Continuously Adaptive Mean Shift
CFA	Rekabetçi Öznitelik Analizi - Competitive Feature Analysis
CMOS	Bütünleyici Metal Oksit Yarıiletken - Complementary Metal Oxide Semiconductor
CRC	Çevrimsel Artıklık Kodu - Cyclic Redundancy Check
DC	Doğru Akım
DOM	Disk On Module
DSP	Sayısal İşaret İşlemcisi - Digital Signal Processor
DVI	Sayısal Görüntü Arayüzü - Digital Visual Interface
fps	Saniyedeki Resim Sayısı - Frame per Second
FPGA	Alanda Programlanabilir Kapı Dizileri - Field Programmable Gate Array
HMM	Saklı Markov Modeli - Hidden Markov Model
HSV	Renk Tonu Doygunluk Değer - Hue Saturation Value
I/O	Giriş/Çıkış - Input/Output
IDE	Entegre Geliştirme Ortamı - Integrated Development Environment

MB	Mega Byte
MBR	Ana Ön Yükleme Kaydı - Master Boot Record
PCA	Ana Bileşen Analizi - Principal Component Analysis
PDA	Avuç içi Bilgisayar - Personal Digital Assistant
QMP	Nicemlenmiş Büyüklük/Faz - Quantized Magnitude/Phase
RGB	Kırmızı Yeşil Mavi – Red Green Blue
SBC	Tek Kartlı Bilgisayar - Single Board Computer
SoC	System-on-Chip
USB	Universal Serial Bus
V	Volt
v4l	Video4linux
W	Watt
YCbCr	Bir renk koordinat sistemi – Y Luminance, CbCr ise Chrominance olarak isimlendirilir.

1 GİRİŞ

Verilen bir resimde herhangi bir yüz olup olmadığının tespitinin yapılması ve bulunan her yüze ait konum bilgisinin geri döndürülmesi işlemine "yüz tespiti (*face detection*)" denir [81].

Gerçek zamanlı bir resim dizisi içinde sürekli olarak yüzün konumu ve duruş biçiminin tahmin edilmesi ise "yüz takibi (*face tracking*)" olarak adlandırılır [81].

"Tespit oranı (*detection rate*)" doğru olarak tespit edilen yüz sayısının insan tarafından tespit edilen yüz sayısına oranıdır. Yüz detektörleri iki tip hata yapabilir: "Yanlış ret hatası (*false negatives*)" yüzlerin tespit edilmeden geçilmesi, "Yanlış kabul (*false positives*)" ise yüz olarak tespit edilen bölgede bir yüz olmama durumudur [81]. Aşağıdaki resimde yer alan 57 yüzün tamamı doğru olarak tespit edilmiş ancak 1 adet de yanlış kabul hatası bulunmaktadır [70].



Şekil 1-1: Yüz tespitine ait örnek resim [70]

Yüz tespiti ile ilgili çalışmalarda önerilen metotların performansı diğer çalışmalarda belirtilen tespit oranları ile karşılaştırma yapılarak ölçülür. Değerlendirme yapılırken öğrenme zamanı, çalışma zamanı, öğrenme aşamasında kullanılan örnek resim sayısı gibi başka kriterlerden de yararlanır.

Gömülü sistemler, özel amaçlı işlerde kullanılmak üzere tasarlanmış, sadece beklenen görevleri yerine getirmek üzere programlanmış, gerçek-zamanlı olarak çalışması istenen özel aygıtlardır [68]. Cep telefonları, taşınabilir bilgisayarlar, otonom çalışan video gözetleme sistemleri, robotlar gibi gömülü sistemler, günlük yaşamımızda yer almaya başlamıştır [82]. Gürbüz bir yüz tespit ve takip detektörüne sahip bir gömülü sistem ile sayısız uygulama gerçekleştirilir. Örneğin, bir robot-insan arayüzü oluştururken robotun karşısındaki insanın yüzünü algılaması ve iletişim esnasında ona bakması sağlanabilir veya cep telefonu üzerinden video konferans esnasında kameradan alınan görüntünün tamamı yerine sadece yüzün olduğu bölge gönderilerek bant genişliği korunabilir [28].

Tez çalışmasında gömülü sistemde çalışan özel bir Linux işletim sistemi hazırlandıktan sonra gerçek zamanlı bir yüz tespit ve takip uygulamasının gerçekleştirimi anlatılacaktır. Gerçek zamanlı olarak çalışan bir yüz tespit ve takip uygulamasının geliştirilmesinde aşağıda belirtilen zorluklar bulunur [81]:

- Takip edilen nesnenin düzensiz bir şekilde hareket etmesi nedeniyle yöntemlerde sabit ebatlı bir arama penceresi kullanılamaz
- Alınan resimlerdeki gürültü yüzün tespit ve takip işlemini zorlaştırır
- El, kol ve resimlerde bulunan diğer yüzler gibi çeldiriciler (distractors) yüz tespitini güçleştirir
- Önerilen tüm yöntemler ışık ve aydınlatma değişimlerinden etkilenmektedir
- Yüzün bazı kısımlarının başka nesnelere arkasında kalması (occlusion) tespiti güçleştirir
- Poz, duruş, yüz ifadeleri, sakal, gözlük gibi kişiye özel farklılıklar sabit özelliklerin belirlenmesini engeller

- Kamera ve resim kalitesi farklılıkları nedeniyle yöntemlerde sabit değerli fonksiyonlar kullanılamaz

Tezin bu bölümünde tez içinde bahsedilecek kavramlara ait tanımlar ve dikkate alınması gereken zorluklara ait bilgiler verilerek kısa bir giriş yapılmıştır. İkinci bölümde ise literatürde yer alan yüz tespit ve yüz takip yöntemlerine değinildikten sonra güncel gömülü sistemler hakkında bilgi verilecektir. Son olarak, gömülü sistemlerde gerçekleştirilen yüz tespit ve takip uygulamalarına bazı örnekler verilecektir.

Tezin üçüncü bölümünde gerçekleştirimi yapılan yüz tespit ve takip yöntemleri detaylı bir şekilde anlatılacaktır. Dördüncü bölümde ise uygulamanın hazırlanması aşamasında izlenen adımlar, oluşturulan geliştirme ortamı ve proje algoritmasına ait akış diyagramı verilecektir.

Tezin beşinci ve altıncı bölümlerinde sırasıyla ana makine ve tek kartlı bilgisayarın hazırlanması sırasında gerçekleştirilen adımlar anlatılacaktır. Yedinci bölümde ise uygulamanın sisteme aktarılması için gerekli prosedür anlatıldıktan sonra uygulamaya ait performans bilgileri verilecektir.

Tezin sekizinci ve son bölümünde ise yedinci bölümde ele alınan performans verilerinin bir değerlendirmesi yapılacak ve gelecekteki çalışmalar için bazı öneriler getirilecektir.

2 YÜZ TESPİTİ, YÜZ TAKİBİ VE GÖMÜLÜ SİSTEMLER

Görüntü işlemede yüz ile ilgili kullanılan tanımlama ve deyimler şöyle belirtilir [81]:

"*Yüzün yerini belirleme (face localization)*" resimdeki tek bir yüzün konumunun hesaplanmasını amaçlar. "*Yüze ait özniteliklerin tespiti (facial feature detection)*" resimde sadece bir yüz bulunduğu varsayımıyla gözler, burun, göz çukuru, ağız vb. yüze ait özniteliklerin varlığını ve konumunu tespit etmeyi hedefler. "*Yüz tanıma (face recognition veya face identification)*" bir girdi resmini veritabanında yer alan diğer resimlerle karşılaştırır ve eğer bir eşleşme bulursa sonucu bildirir. "*Yüzden kimlik tanıma (face authentication)*" bir girdi resmindeki bireyin kimliğinin doğrulanmasını amaçlar. "*Yüze ait ifadelerin tanınması (facial expression recognition)*" insanların mutlu, üzgün, iğrenmiş vb. etkin yüz ifadelerinin tanımlanması ile ilgilidir.

2.1 Yüz Tespit Çalışmaları

Resim veya video görüntülerindeki yüzlerin tespiti için önerilen yöntemler dört gruba ayrılır [81]:

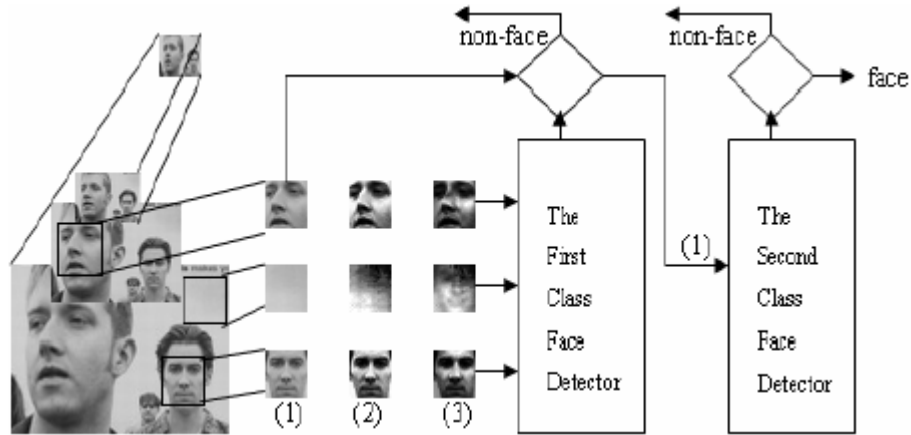
1. Bilgi tabanlı metotlar (Knowledge-based methods):

Bu yaklaşımı kullanan yüz tespit metotları, araştırmacının insan yüzü hakkında sahip olduğu bilgilerden türettiği kurallara bağlıdır. Örneğin, "*bir resimde genellikle iki göz birbirine ağız ve buruna göre simetrik olarak bulunur*" bilgisinden bağıl pozisyon ve uzaklık verilerinden yararlanılarak çeşitli kurallar oluşturulur ve yüz tespit edilir. İnsan yüzüne ait bilgilerden iyi tanımlanmış kurallar elde edilmesi aşamasının zorluğu yaklaşımın yaygın kullanımını engeller [81].

2. Öznitelik değişimsiz yaklaşımlar (Feature invariant approaches):

İnsanların farklı poz ve aydınlık durumlarında bile yüzleri tespit etme özelliğinden yola çıkarak, değişimlerden etkilenmeyen bazı özellik ve özniteliklerin olması gerektiği savunulur. Önce yüze ait öznitelikler tespit edilir ardından yüzün varlığı araştırılır. Elde edilen özniteliklerden bir istatistiksel model oluşturularak yüzün varlığı doğrulanır.

Yaklaşımına bir örnek olarak, üç adet özellikten yararlanılan gürbüz bir yüz tespit algoritması verilebilir [69]: Algoritmada "zayıf sınıflandırıcıları (weak classifiers)" tasarlarlarken kullanılan "birleştirilmiş ortalama yüz özneliği (unified average face feature)" ve "yerel öznelik (local feature)" olarak adlandırılan iki çeşit dikdörtgen öznelik kullanılır. Son olarak iki özellik birleştirilerek bir "öğrenme algoritması (learning algorithm)" ile "ardarda sıralı (cascade)" yüz detektörü elde edilir. Yapılan eklentiler sayesinde, yüzün bazı kısımları başka nesnelere arkasında kalsa bile yüzler tespit edilir.



Şekil 2-1: Öznitelik değişimsiz yaklaşım örneği [69]

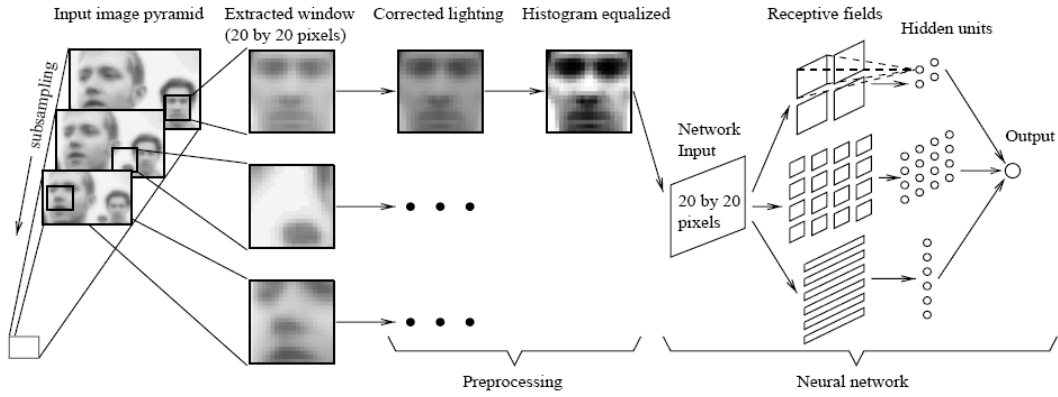
3. Şablon eşleştirme metotları (Template matching methods):

Şablon eşleştirmede, cepheden çekilmiş bir yüz resmi gibi standart bir yüz örüntüsü, önceden elle tanımlanır (predefined) veya parametreler halinde ifade edilir. Verilen bir resimdeki yüz çeperi, gözler, burun ve ağız için birbirlerinden

bağımsız olarak "ilinti (correlation)" değerleri hesaplanarak yüzün varlığına karar verilir. Yaklaşım resim skalası, poz ve biçim farklarındaki değişikliklerden etkilenir [81].

4. Görünüş tabanlı metotlar (Appearance-based methods):

Görünüş tabanlı metotlarda, resimlerde yer alan örneklerden öğrenme yoluyla çeşitli şablonlar oluşturulur. İstatistiksel analiz ve "otomatik öğrenme (machine learning)" yöntemlerinden faydalanılarak bir yüzün olduğu ve olmadığı bölgelere ait karakteristik bilgiler elde edilir. Rowley ve ark. [70] "sinir ağları (neural networks)" tabanlı, gri skalada ve cepheden çekilmiş yüzlerin tespit edildiği bir sistem geliştirmiştir. Sistemde bir veya daha fazla sinir ağı alt pencereler üzerinde inceleme yaparak her bir pencerenin yüz içerip içermediğini kontrol etmektedir. Verilen resim filtreler ve seçiciden oluşan iki aşamadan geçer. Filtrelerde olası yüz bölgeleri farklı skalalarda taranır. Seçicide ise farklı filtrelerden gelen sonuçlar birleştirilir ve üst üste binen tespitler elenerek yüz tespit edilir.



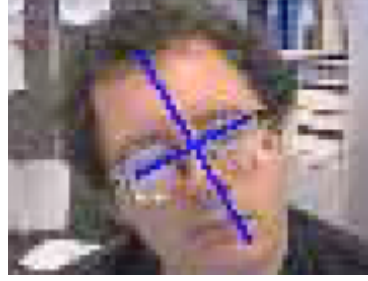
Şekil 2-2: Rowley'nin yüz tespit algoritması [70]

2.2 Yüz Takip Çalışmaları

Yüz takibi için önerilen yöntemler dört grupta toplanır [5]:

1. Ten rengine dayalı metotlar (skin-color based tracking):

Az sayıda işlem gerektirdiği için ten rengine dayalı metotlar oldukça popülerdir. Önerilen yöntemler ışık ve aydınlatma değişimleri ile arkaplan resminde ten rengine benzeyen renklerden etkilenir [5].



Şekil 2-3: Ten rengine dayalı örnek resim [1]

2. Biçim tabanlı metotlar (Shape based methods):

Biçim tabanlı metotlar arkaplan renginden ve aydınlatma değişimlerinden etkilenmez. Fakat, yöntemler yoğun arkaplan gürültüsüne karşı aşırı hassas oldukları için gürbüz değildir [5].



Şekil 2-4: Birchfield örnek yüz takip resimleri [43]

3. Öznitelik tabanlı metotlar (Feature based methods):

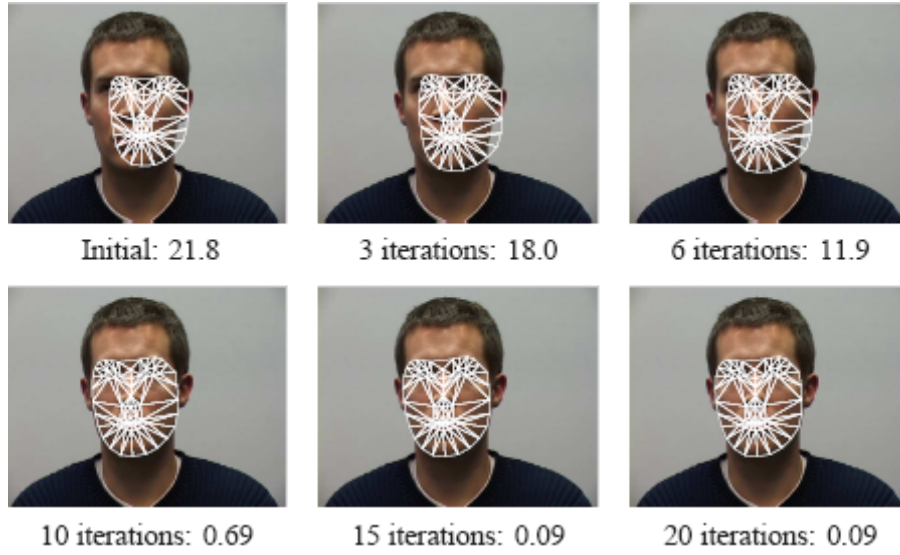
Öznitelik tabanlı metotlar yüz takibi için çıkarım yapılarak elde edilen öznitelikleri kullanır. Bu metotlar yüksek işlem gücü gerektirir ve gerçek zamanlı bir uygulama oluşturulamaz [5].



Şekil 2-5: Örnek resim dizisi – Maurer ve Malsburg yöntemi [4]

4. Otomatik öğrenme tabanlı metotlar (Machine learning based methods):

Otomatik öğrenmeye dayalı metotlar öznitelik veya model tabanlı olabilir: Öznitelik tabanlı metotlar "*görüntü eğrilerinin sürerliğine (persistence of image curves)*" veya görüntünün duruşuna, model tabanlı metotlar ise hem iki boyutlu hem de üç boyutlu olarak görüntü eğrilerine dayanmaktadır [54].



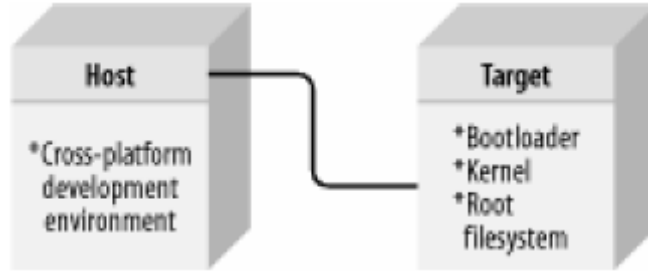
Şekil 2-6: Örnek resim dizisi – Matthews ve Baker [61]



Şekil 2-7: Örnek resim dizisi – Williams ve ark. [54]

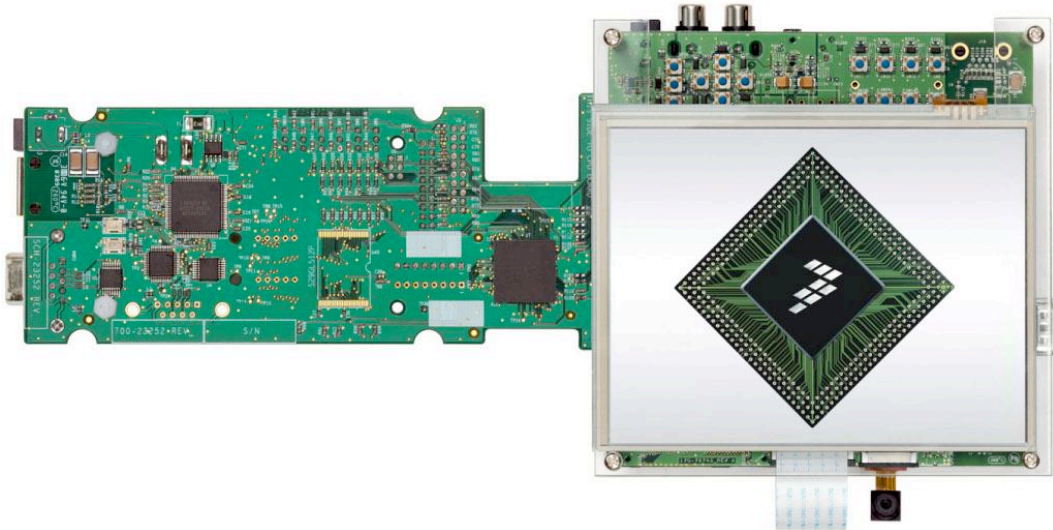
2.3 Gömülü Sistemler

Gömülü sistemlerin geliştirme ortamında tüm geliştirmelerin yapıldığı bir "ana makine (host)" ve geliştirilen sistemin çalışacağı bir "hedef sistem (target)" yer alır [15, 16].



Şekil 2-8: Örnek Ana Makine / Hedef Bağlantısı [16]

Gömülü sistem üzerinde çalışacak bir uygulama hazırlarken "derleyici (compiler)" ve "hata ayıklayıcı (debugger)" araçları kullanılır [68]. Sistem geliştirme aşamasında nihai sisteme göre daha fazla veriyolu desteği ve test imkanı sağlayan "geliştirme araçlarından (development kits)" yararlanır [55, 56].



Şekil 2-9: Örnek Geliştirme Aracı [55, 56]

Bir gömülü sistemin çekirdeğini yapılacak işin kapsamına göre özelleştirilmiş işletim sistemi kullanan bir mikroişlemci oluşturur [68]. Özelleştirilmiş işletim sistemleri, aktarımlar yolu ile ticari olmayan bir Linux işletim sisteminde yer alan ücretsiz araçlarla veya RedHat, VxWorks, Green Hills - Integrity, Timesys Linux, MQX, QNX vb. ticari işletim sistemleri kullanılarak hazırlanır [73-79]. Bir gömülü Linux sistemi oluştururken aşağıda belirtilen adımlar izlenir [16]:

- Gömülü sistem gereksinimleri belirlenir
- Gereksinimler doğrultusunda donanım seçimleri yapılır
- Sistem donanımları hazırlanır
- Ana makinede ilgili yazılım bileşenleri bir araya getirilir
- Sistem için "*çekirdek (kernel)*" derlenir
- Sistemde bir "*kök dosya sistemi (root file system)*" yaratılır
- Sistemde temel yazılım ayarları yapılır
- Sistemin açılışını sağlayacak "*önyükleme (boot)*" yazılımı hazırlanır
- Sistem ile ana makinenin haberleşmesi için ethernet, seri port gibi iletişim protokollerine ait ayarlar yapılır
- Sistemde çalışması planlanan özelleşmiş yazılımlar ana makinede derlenir ve sisteme taşınır.

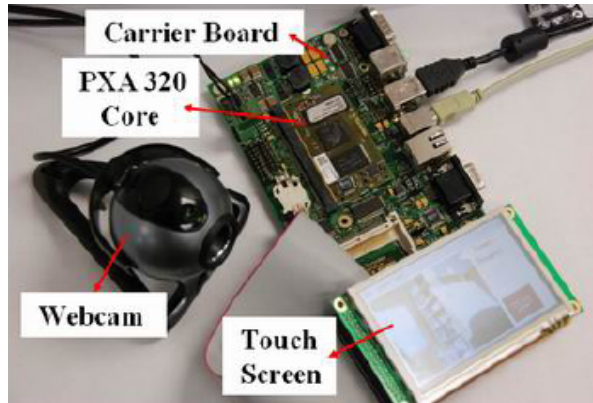
Gömülü sistemler "*Alanda Programlanabilir Kapı Dizileri (Field Programmable Gate Array - FPGA)*", "*sayısal işaret işlemcisi (Digital Signal Processor - DSP)*" veya "*tek kartlı bilgisayar (Single Board Computer - SBC)*" mimarileri kullanılarak oluşturulur:

- FPGA programlanabilir mantık blokları ve bloklar arasındaki ara bağlantılardan oluşan sayısal tümleşik devrelerdir [3].

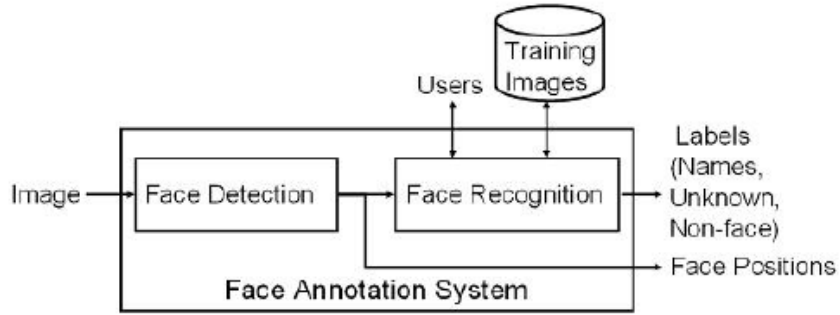
- DSP sayısal işaret işleme görevini yerine getirmek için tasarlanan, eniyileştirilmiş mimariye sahip özel mikroişlemcidir [57].
- SBC tek devre kartı üzerinde çalışan tam bir bilgisayardır [68]. Sistem tek veya çift çekirdekli bir işlemci, bellek, giriş/çıkış sinyalleri, ekran kartı, ethernet kartı ve ses kartı gibi bileşenlerden oluşur [68]. Depolama amacıyla Disk On Module (DOM), Disk On Chip veya Compact Flash gibi depolama aygıtlarından faydalanılır [68]. Farklı form factor'lerde ve farklı veriyollarına sahip SBC'ler geliştirilmiştir:
 - Advanced Mezzanine Cards – AMC kartlar, PcPMC kartlar, Embedded Compact Extended (ECX), Mini-ITX, PICMG, PXI, VMEbus, VPX, VXI ve Computer-on-Module (COMs) kartları gibi temel fonksiyonları içeren ancak daha büyük sistemler veya taşıyıcı kartlara takılıp entegre olarak çalışan SBC'ler [68].
 - Tüm bileşenleri kart üstünde olan tümleşik yapıda ve herhangi bir arkayüz mimarisine bağlı olmadan çalışan PC/104, PC/104-Plus, PCI-104, EPIC ve EBX form factor'lerine sahip stack tipi SBC'ler [68].

2.3.1 Gömülü sistemlerde yüz tespit ve takip uygulamaları

Literatürdeki yüz tespit algoritmaları, kişisel bilgisayarlarda uygulanmak ve tespit oranı çalışma verimliliğinden daha önde tutularak hazırlanır [14]. Chu ve ark. [14] yukarıda belirtilen sebeplerden ötürü herhangi bir kıyas yapmaksızın Toradex firmasının XScale PXA 320 işlemcili Orchid Standard Carrier Board'unu ve bir Internet kamerası kullanarak yüz tespit ve tanıma uygulaması sunmuştur.



Şekil 2-10: Chu'nun kullandığı donanım [14]



Şekil 2-11: Chu'nun algoritması [14]

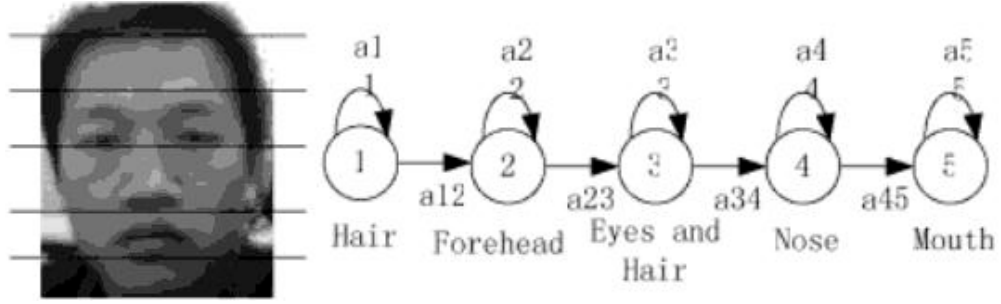
“*Face Sensing Engine*” ismindeki gömülü sistemlerde kullanılmak üzere hazırlanan ticari bir “*ara katman yazılımı (middleware)*” sayesinde elde edilen kazanımlar şöyledir [64]:

- Cihazlar sadece tanımlı kişilerce kullanılır
- Ortak kullanılan cihazlarda kişiye özel verilerin diğer kullanıcılar tarafından erişilmesi engellenir
- Fotoğraf makinelerinde tespit edilen yüze otomatik olarak odaklanarak kırmızı veya kapalı göz gibi resim hataları engellenir
- Bir resimde tespit edilen kişiye ait veritabanında bulunan tüm resimler ekrana getirilir
- İnternet üzerinden iletişim esnasında bir animasyon görüntüsü kullanılarak kişisel güvenlik sağlanır
- Yüz fotoğrafları dijital imza olarak kullanılır
- Araba sürücülerinin uyuduğu tespit edilerek olası kazalar önlenir.

Robot sekreter Saya isimli proje kapsamında insanların yüz hareketlerine göre hareket eden bir robot yaratmak için gürbüz bir yüz takip algoritması geliştirilmiştir [7]. Çalışmada “*kromatik renk uzayında (chromatic color space)*” cilt rengi filtrelemesi ve Gauss cilt rengi modeline dayalı bir

metot benimsenerek OpenCV kütüphanesi [59] ile JAVA programlama dillerinde kodlama yapılmıştır [12].

Sun ve ark. [8] Linux işletim sistemi, OV7640 CMOS sensör ve Samsung S3C2410A işlemcisinin bulunduğu bir gömülü sistemde "*Saklı Markof Modelinin (Hidden Markov Model - HMM)*" esas alındığı bir yüz tanıma algoritması sunmuştur.



Şekil 2-12: Sun'ın 1 boyutlu HMM'si [8]

Firewire kameralar ve taşınabilir bir görüntü işleme iş istasyonundan meydana gelen CAMEO (the Camera Assisted Meeting Event Observer) sisteminde yüz tespit ve takip işlemi yapılır [9]. Sistem "*çerçeve farkları (frame differencing)*" ve "*ortalama kayma (mean shift)*" [51] algoritmasının esas alındığı uyarlamalı (adaptive) renk takibi algoritmaları ile gerçekleştirilmiştir: Yüz tespiti için Schneiderman'ın [10] çalışmasında sunulan metot, tespit edilen yüzün takip edilmesi için ise Comaniciu ve ark.'nın [11] geliştirdikleri metot kullanılmıştır.

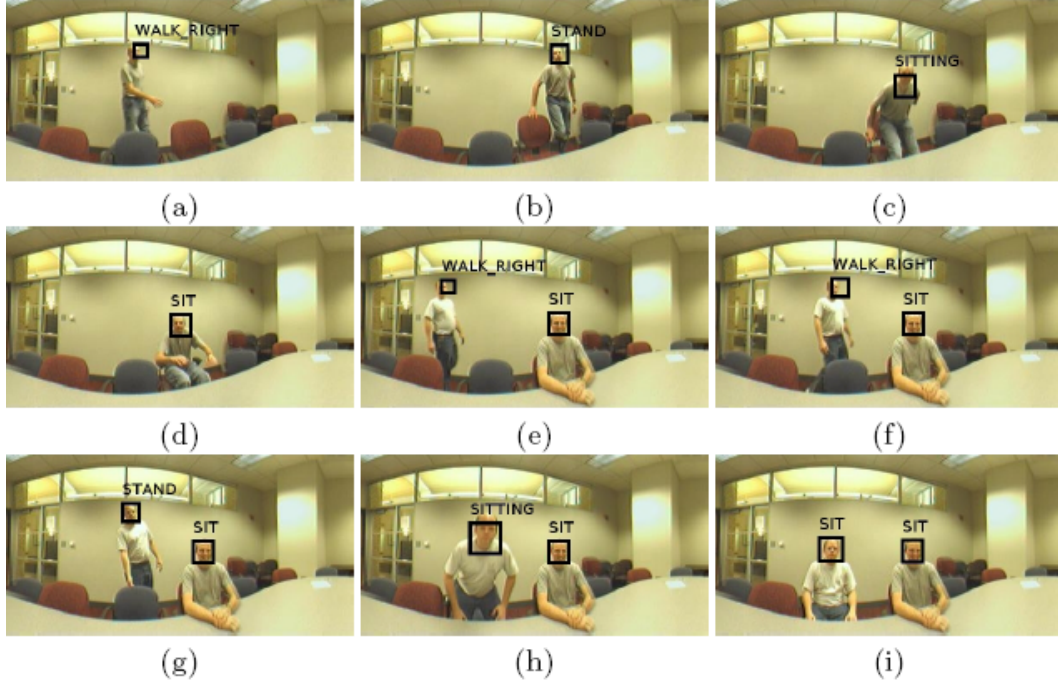


Şekil 2-13: CAMEO sistemi [9]



Şekil 2-14: CAMEO ile ilgi alanında yer alan yüzün tespiti [9]

CAMEO sisteminin kullanıldığı bir diğer çalışmada, HMM ve Kalman filtrelerinin geliştirilmiş bir hali olan "*Dinamik Bayes Ağları (Dynamic Bayesian Networks)*" kullanılmıştır [62]. OpenCV kütüphanesinin kullanıldığı çalışmada videodaki kişilerin yüzlerinin tespitinin ardından oturma, sağa – sola doğru yürüme gibi kişilerin gerçekleştirmekte oldukları hareketler tahmin edilmektedir.

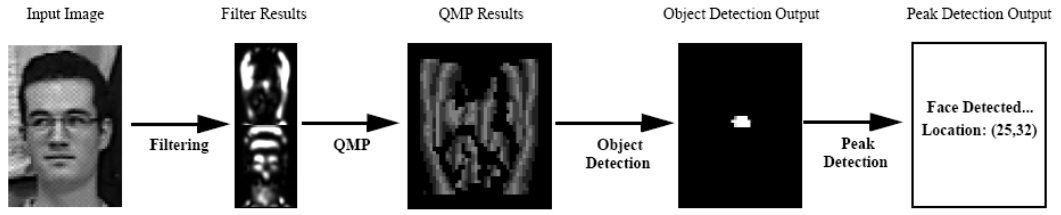


Şekil 2-15: CAMEO ile yüze göre hareketin sınıflandırılması [62]

Bigdeli ve ark. [65] otomatik yüz tespit ve tanıma işlemini gerçekleştirmek için FPGA tabanlı bir gömülü sistem tasarımı yapmıştır. GNU GCC derleyicisini ve C programlama dilinin kullanıldığı çalışmada, hedef sistem Altera Nios II işlemcili bir Stratix FPGA geliştirme aracıdır.

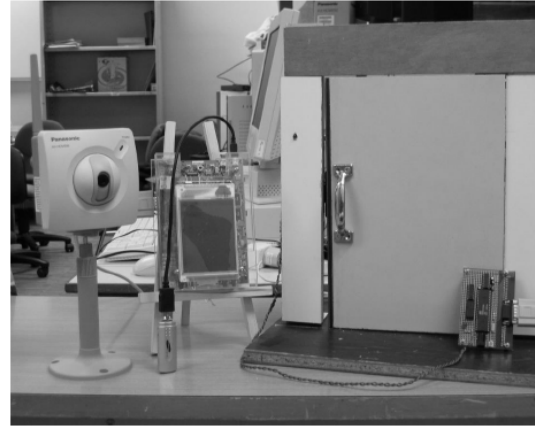
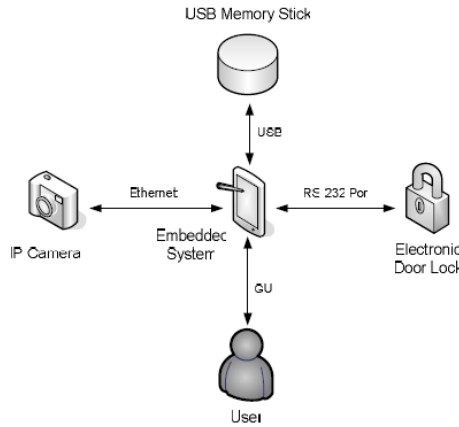
McCready, Transmogripher-2 isimli çok kartlı FPGA platformu için bir yüz tespit algoritması geliştirmiştir [66]. Girdi resmi aşağıdaki aşamalardan geçirilir:

- Resme G2/H2 filtreleri ile filtrelenir
- Filtrelenen resim "*nicemlenmiş büyüklük/faz (Quantized Magnitude/Phase - QMP)*" biçimine çevrilir
- QMP sonuçlarına göre "*rekabetçi öznelik analizi (Competitive Feature Analysis - CFA)*" öğrenme algoritması uygulanarak olasılık ve nesnenin pozisyon bilgileri elde edilir
- Sonuç değeri eşik değeri ile karşılaştırılarak bir karar verilir.



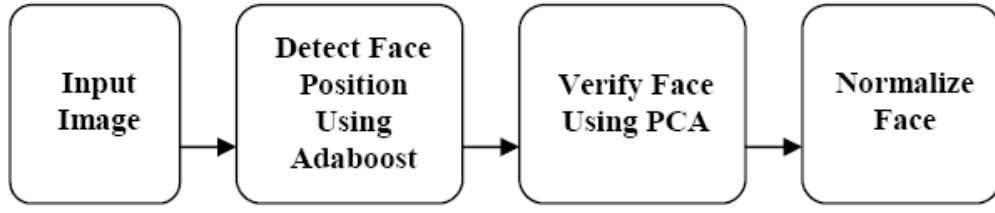
Şekil 2-16: McCready'nin yöntemi [66]

Pun ve ark.[67] yüz tespiti ve tanınması ile çalışan gömülü kapı geçiş kontrol sistemi geliştirmiştir. Yüz tespitini yarı otomatik gerçekleştiren sistem, yüz tanıma işlemi için göz pozisyon bilgisinden yararlanır. Çalışmada Linux işletim sistemi ve Intel XScale PXA255 400Mhz işlemcili bir geliştirme aracı kullanılmıştır.



Şekil 2-17: Pun'un sistem diyagramı ve düzeneği [67]

Lee ve ark. [71] TMS320C6711 DSP'li bir sistem kullanarak İnsan-Robot etkileşimi için insan yüzlerinin tespiti ve yüze ait ifadelerin tanınması yönünde bir çalışma yapmıştır. Yüz tespitinde AdaBoost algoritması [6], yüz tanımda "ana bileşen analizi (Principal Component Analysis - PCA)", yüze ait ifadelerin anlaşılmasında ise "Gabor dalgacıkları (wavelet)" kullanılır.



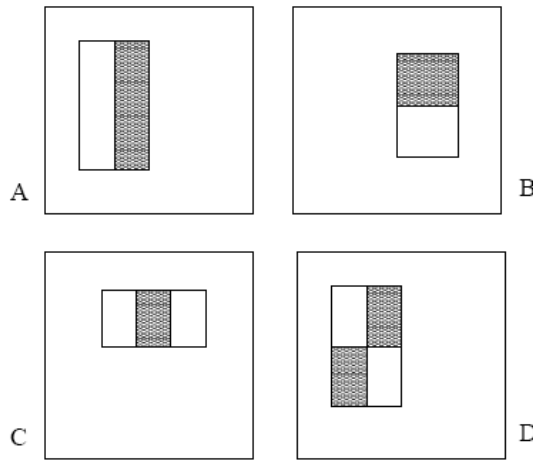
Şekil 2-18: Lee'nin önerdiği yöntem [71]

3 YÜZ TESPİT VE TAKİBİ İÇİN KULLANILAN YÖNTEMLER

3.1 VIOLA ve JONES Yüz Tespit Algoritması

Viola ve Jones algoritmasında üç ana özellik bulunur [2]: Resim detektörünün kullanacağı özniteliklerin hesaplanmasını sağlayan "tümlevsel görüntü (*integral image*)" gösterimi, AdaBoost algoritması baz alınarak büyük veri setlerinden küçük sayılarda görsel özniteliklerin seçilerek verimli sınıflandırıcılar türeten bir öğrenme algoritması ve türetilen sınıflandırıcıların ardarda sıralı bir şekilde birleştirilmesiyle daha karmaşık sınıflandırıcıların elde edilmesidir.

Çalışmada üç çeşit dikdörtgen öznitelik değeri kullanılır. İki dikdörtgenli öznitelik değeri iki dikdörtgen bölgedeki pikseller toplamının birbirinden çıkarılması ile elde edilir (Şekil 3-1A ve B). Dikdörtgen bölgeler aynı boyutlu, aynı şekle sahip ve yatay veya dikey olarak birbirine bitişik (*adjacent*) olmalıdır. Üç dikdörtgenli öznitelik değeri, dıştaki dikdörtgenlerin piksel toplamından merkezde bulunan dikdörtgenin piksel toplamının çıkarılmasıyla elde edilir (Şekil 3-1C). Dört dikdörtgenli öznitelik değeri ise çapraz dikdörtgenlerin toplamı birbirinden çıkarılarak hesaplanır (Şekil 3-1D).



Şekil 3-1: Dikdörtgen öznitelikler

3.1.1 Tümlevsel görüntü

Dikdörtgen özniteliklerin hesap edilmesi için tümlevsel görüntü gösterimi kullanılır. Resmin x, y noktasındaki tümlevsel görüntü değeri x, y noktasının solunda ve üstünde kalan tüm piksel değerlerinin toplamıdır.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3.1)$$

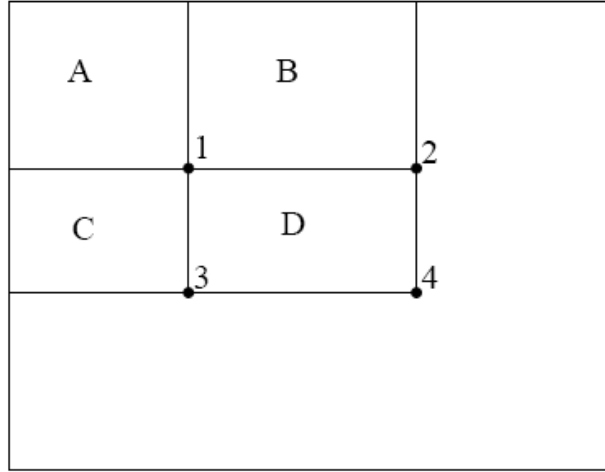
Denklem 3.1'de $ii(x, y)$ tümlevsel görüntü değerini, $i(x, y)$ değeri de orijinal resmin x, y noktasındaki değerini göstermektedir. Aşağıdaki özyineli işlemler sayesinde tümlevsel görüntü değeri hesaplanır:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (3.2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (3.3)$$

Denklem 3.2 ve 3.3' teki $s(x, y)$ kümülatif satır toplamını gösterir, $s(x, -1)$ ve $ii(-1, y)$ değerleri 0 olarak kabul edilir.

Tümlevsel görüntü gösterimi ile dikdörtgen bir bölgeye ait piksel toplamı referans dört nokta kullanılarak hesaplanır. Örneğin, Şekil 3.2' de D olarak isimlendirilen bölgedeki piksellerin toplamını hesaplamak için 1'den 4'e kadar belirtilen noktalar referans olarak alınır. 1 numaralı noktadaki tümlevsel görüntü değeri A olarak adlandırılan bölgenin piksel toplamıdır. 2 numaralı noktadaki değer A + B, 3 numaralı noktadaki değer A + C ve 4 numaralı noktadaki değer ise A + B + C + D' dir. Tüm bilgiler bir araya getirilerek, D bölgesindeki piksel toplamı $4 + 1 - (2 + 3)$ şeklinde tümlevsel görüntü değerleri kullanılarak hesaplanır. İki dikdörtgenli öznitelik değerinin hesabı için referans altı nokta, üç dikdörtgenli öznitelik değeri için ise referans sekiz nokta kullanılarak istenen değerler hesaplanır.



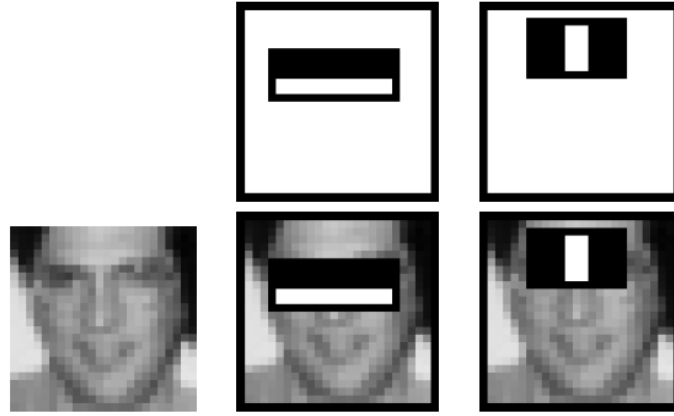
Şekil 3-2: Tümevsel görüntü değerin hesaplanması

3.1.2 Öğrenme algoritması

AdaBoost öğrenme algoritmasının bir versiyonu daha küçük öznitelik setlerinin seçilmesinde ve sınıflandırıcıların eğitiminde kullanılır. "Zayıf öğrenici (*weak learner*)" her bir öznitelik için optimal eşik sınıflandırıcı fonksiyonunu belirler. Basit bir sınıflandırıcı $h_j(x)$, böylece f_j öznitelik değeri, θ_j eşik değeri ve p_j parity değerleri kullanılarak elde edilir. Buradaki x değeri bir resmin 24x24'lük bir alt resim penceresidir.

$$h_j(x) = \begin{cases} 1 & \text{eğer } p_j f_j(x) < p_j \theta_j \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.4)$$

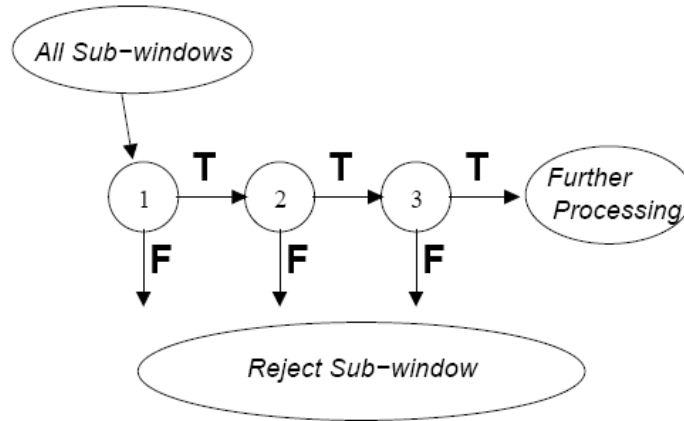
Yüz tespiti için AdaBoost algoritması tarafından seçilen ilk iki öznitelik Şekil 3-3' te verilmiştir. İlk öznitelik göz bölgesindeki pikseller burun ve yanaklara göre daha koyu renklidir özelliğini kullanırken, ikinci öznitelik gözler burun kıvrımına göre daha koyudur özelliğini baz almıştır.



Şekil 3-3: AdaBoost tarafından seçilen ilk iki öznelik

3.1.3 Ardarda sıralı sınıflandırıcı

Karar alma sürecinde ardarda sıralı sınıflandırıcılar kullanılır. İlk sınıflandırıcıdan gelen pozitif sonuç, ikinci bir sınıflandırıcıyı çalıştırır, benzer şekilde ikinci sınıflandırıcı üçüncüyü tetikler ve böyle devam eder. Herhangi bir anda negatif sonuç gelmesi durumunda alt pencere reddedilir (Şekil 3-4).

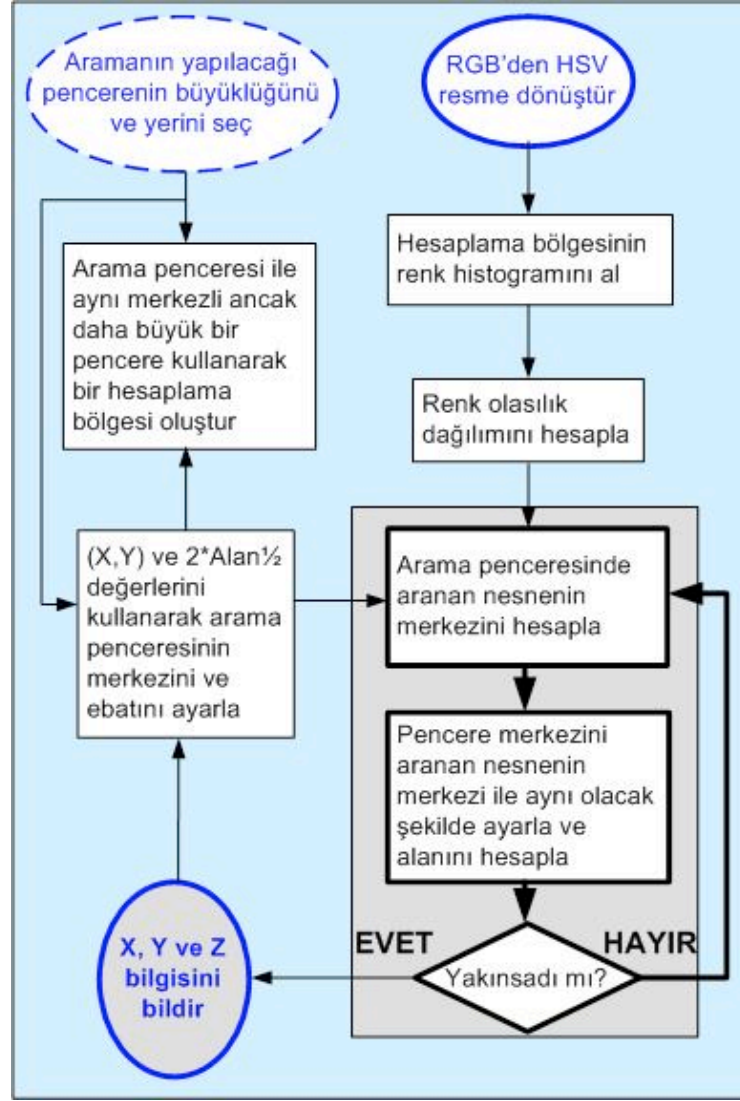


Şekil 3-4: Ardarda sıralı sınıflandırıcı

3.2 CAMSHIFT Yüz Takip Algoritması

Bradski [1] "sürekli uyarlanabilir ortalama kayma algoritması (continuously adaptive mean shift -

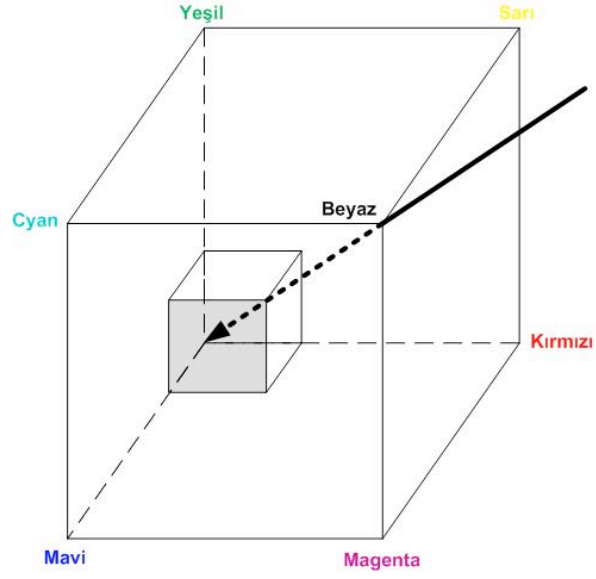
CAMSHIFT) " adını verdiđi yüz takip yönteminde, ortalama kayma algoritmasına bir nesne takipçisi ve ten rengi algoritması eklemiştir [45 – 49]. Algoritmada gürbüz istatistiki verilere ulaşmak için "*ilgi alanından (region of interest)*" uzaktaki veriler ihmal edilir. Gürültü ve çeldiriciler gibi olumsuz etkenlerin giderilmesinde "*olasılık dağılımlarına (probability distribution)*" göre çalışan ortalama kayma algoritması kullanılır [50, 51]. Renkli resim karelerinden oluşan dizilerdeki nesnelerin takibi için renk histogramlarından yararlanır [1]. Her yeni gelen resimde renk dağılımı deđiştirdiğinden ortalama kayma algoritması takip edilen nesnenin olasılık dağılım deđişimlerine göre dinamik olarak uyarlanır. X ve Y düzlemleri ile yüzü gösteren alandaki ten rengine ait olasılık dağılımı takip edilir. Yüzün bulunduđu alan ise Z düzlemi (kamera ile olan mesafe) ile doğru orantılı olarak hesaplanır. Şekil 3-5'te CAMSHIFT algoritmasının akış diyagramı verilmiştir.



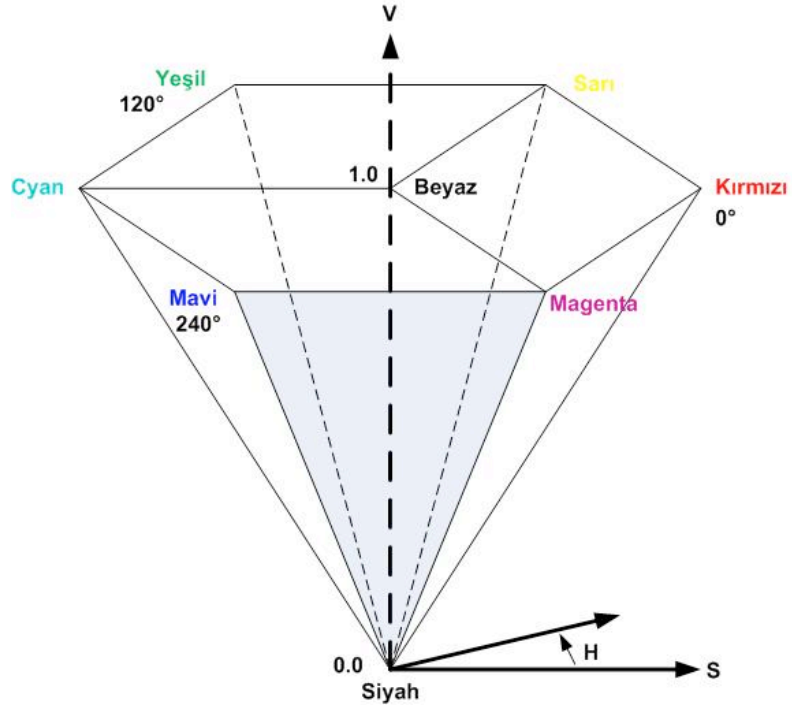
Şekil 3-5: CAMSHIFT algoritması

3.2.1 Renk olasılık dağılımları

CAMSHIFT algoritmasında "Renk Tonu Doygunluk Değer (Hue Saturation Value - HSV)" renk sistemi kullanılmıştır [52, 53]. "Kırmızı, Yeşil ve Mavi (Red Green Blue - RGB)" renk uzayının beyazdan siyaha doğru giden ve Şekil 3-6'da ok şekli ile gösterilen diyagonal çizgi boyunca izdüşümünün (projection) alınması ile HSV sistemini gösteren altı kenarlı koni (hexcone) şekli elde edilir (Şekil 3-7).



Şekil 3-6: RGB renk küpü



Şekil 3-7: HSV renk sistemi

HSV renk sisteminde V ekseninden azalarak gidilirse, daha küçük altı kenarlı koniler ile daha küçük ve daha koyu renkli RGB renk alt küpleri elde edilir (Şekil 3-6). Böylelikle, renk tonu doygunluk ile parlaklık değerlerinden ayrıştırılır

ve H kanalına ait deęerin 1 boyutlu histogramı kaydedilerek bir renk modeli oluşturulur. Oluşturulan renk modelinde, yeni alınan her resim karesinden bir ten rengi olasılık dağılımı hesaplanarak 0.0 deęerinin en olasılıksız, 1.0 deęerinin ise en olası olduđu ayırık bir olasılık deęer aralıęı elde edilir.

3.2.2 Algoritma

CAMSHIFT algoritması içinde yer alan ortalama kayma algoritması şöyle çalışır (Şekil 3-5’te gri arkaplanlı kısım):

1. Bir arama penceresi ebadı seç
2. Arama penceresinin ilk konumunu seç
3. Arama penceresindeki ortalama konum deęerini hesapla
4. Üç numaralı adımda hesaplanan ortalama konum deęeri merkez olacak şekilde arama penceresini yeniden konumlandır
5. Üç ve dört numaralı adımdaki işlemleri yakınsama gerçekleşene kadar tekrarla

CAMSHIFT algoritmasında arama penceresinin altında kalan dağılıma ait “alan” olarak düşünölen sıfırncı moment bilgisi kullanılarak her bir resim karesindeki pencere ebadı resimdeki verilere göre sürekli uyarlanır. Sıfırncı moment, x ve y ’nin birinci momentleri aşığıdaki denklemler kullanılarak hesaplanır:

$$M_{00} = \sum_x \sum_y I(x, y) \quad (3.5)$$

$$M_{10} = \sum_x \sum_y xI(x, y) ; M_{01} = \sum_x \sum_y yI(x, y) \quad (3.6)$$

Yukarıdaki denklemlerde $I(x, y)$ resmin (x, y) noktasındaki piksel (olasılık) deęeri, x ve y ise arama penceresinin aralık deęerini gösterir.

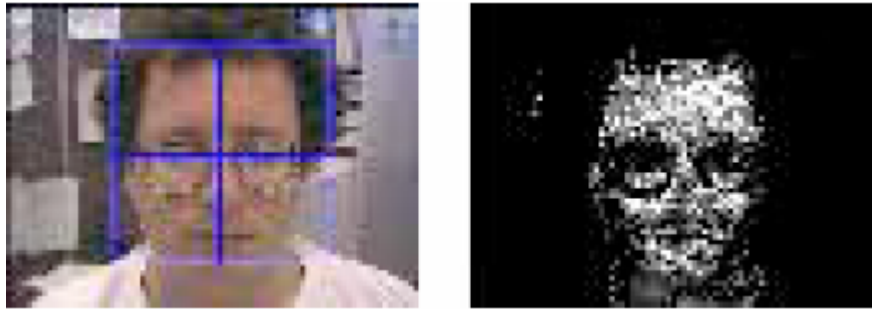
Arama penceresinin konum bilgisi ve 2 boyutlu renk olasılık dağılımlarında maksimum piksel değeri 255 olabileceğinden dolayı, pencere ebadı hesabında kullanılan s değeri aşağıdaki denklemler kullanılarak hesaplanır:

$$x_c = \frac{M_{10}}{M_{00}} ; y_c = \frac{M_{01}}{M_{00}} \quad (3.7)$$

$$s = 2 * \sqrt{\frac{M_{00}}{256}} \quad (3.8)$$

Şekil 3-5'te akış diyagramı verilen CAMSHIFT algoritması şöyle uygulanır:

1. Arama penceresinin ilk konumunu belirle
2. Ortalama kayma algoritmasındaki adımları gerektiği kadar tekrarla ve son sıfıncı moment değerini kaydet
3. Pencere eni s ve pencere boyu $1.2s$ (Denklem 3.8) olacak şekilde arama penceresinin ebadını ayarla
4. İkinci ve üçüncü adımdaki işlemleri yakınsama gerçekleşene kadar devam ettir.



Şekil 3-8: Yüz takip ve ten rengi olasılık dağılım resmi

4 UYGULAMANIN GENEL YAPISI

Tez projesi Şekil 4-1’de akış diyagramı verilen geliştirme yöntemine göre gerçekleştirilmiştir.



Şekil 4-1: Geliştirme yöntemi

4.1 Donanımların ve İşletim Sisteminin Seçilmesi

Debian, Linux çekirdeğini kullanan %100 özgür bir işletim sistemidir [19 - 21]. İşletim sisteminin merkezinde bir çekirdek (kernel) yer alır. Debian işletim sistemini oluşturan temel araçların çoğunun GNU projesinden gelmesi sebebiyle işletim sistemi Debian GNU/Linux olarak adlandırılır [22]. Debian'ın her

sürümüne bir kod adı ve sürüm numarası verilir [21]. Mevcut kararlı (stable) sürümün adı "*Lenny*" ve sürüm numarası 5.0' dır. Kritik çalışmalarda güvenilir olması sebebiyle sadece kararlı sürümdeki paketler kullanılır. Kararlı sürümdeki paketlerin testlerden geçmiş olması sebebiyle tez çalışması sırasında kullanılacak işletim sistemi olarak Debian GNU/Linux seçilmiştir.

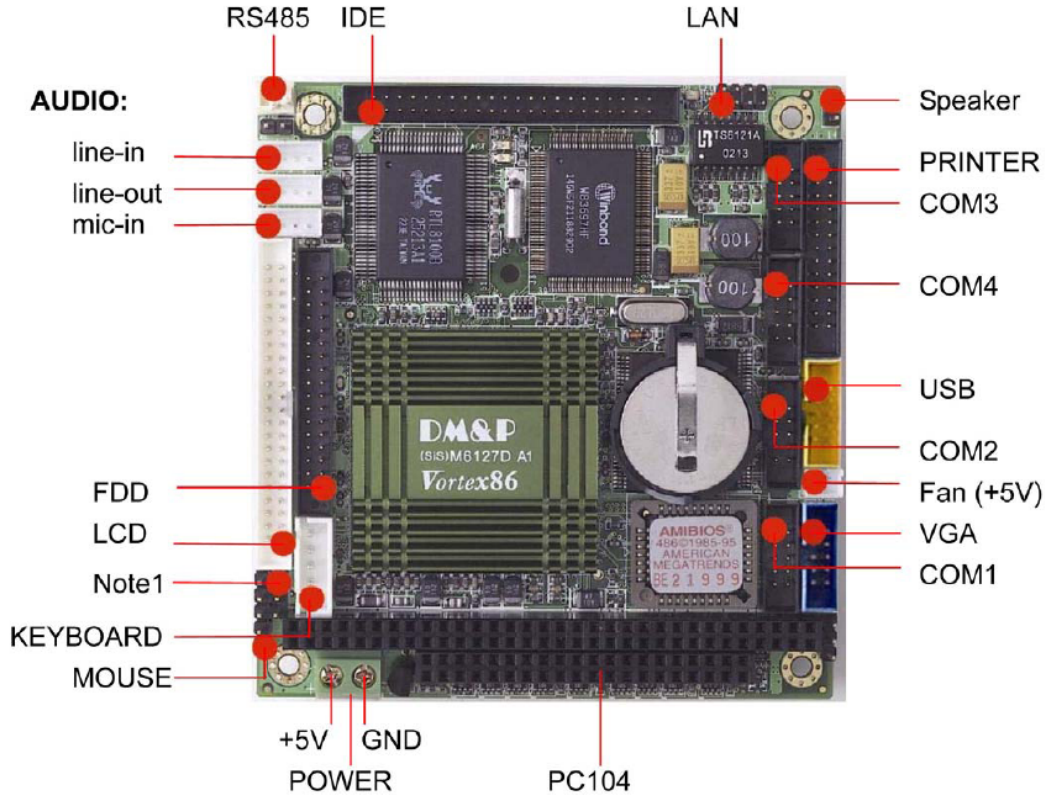
Debian GNU/Linux minimum donanım gereksinimleri [83] ve projede ethernet üzerinden akacak veri miktarına bağlı olarak tez çalışmasında kullanılacak SBC aşağıda belirtilen minimal kriterleri sağlamalıdır:

- Uygulama versiyonlarına bağlı olarak 32 - 64 MB Disk Alanı
- 100 Mhz çalışma frekansına sahip bir işlemci
- 64 MB RAM
- Internet kamera bağlantısı için bir adet USB portu
- Önyükleme yapabilmek için bir adet disket sürücü portu
- SBC'nin ethernet üzerinden kontrol edilmesi ve Internet kamerasından alınan görüntünün ana bilgisayara aktarımı için 100Mbps'lik bir ethernet portu

"*ICOPTTECH*" firmasının "*Vortex86-6071LV - PC/104 Embedded Vortex86LV (Low Voltage) 166 MHz System-on-Chip CPU Module with 4S/CRT/LCD/Ethernet/Audio/128MB SDRAM on board*" adlı SBC'si yukarıda belirtilen tüm gereksinimleri sağladığından tez çalışmasında kullanılmak üzere temin edilmiştir [17, 18]. SBC'nin teknik özellikleri ve konnektör yerleşimini gösteren resmi aşağıda verilmiştir:

- Gerçek zamanlı saati olan DM&P(SiS) Vortex86™ System-on-Chip CPU-166MHz SoC kart
- PC104 Veriyolu
- AMI BIOS
- 128MB SDRAM
- AGP V.2.0 uyumlu VGA ve STN/DSTN/TFT Flat Panel arabirim desteği

- Realtek 8100B 10/100Mbps Ethernet
- AC97 V2.1 Audio
- 1 adet 16MB'nin üstündeki Disk On Module'leri çalıştıracak Enhanced IDE portu
- 1 adet FDD disket sürücü portu
- 3 adet RS-232 seri portu
- 1 adet RS-232/485 portu
- 1 adet Paralel port
- 2 adet USB V.1.1 port



Şekil 4-2: ICOPTECH – Vortex86 tek kartlı bilgisayar

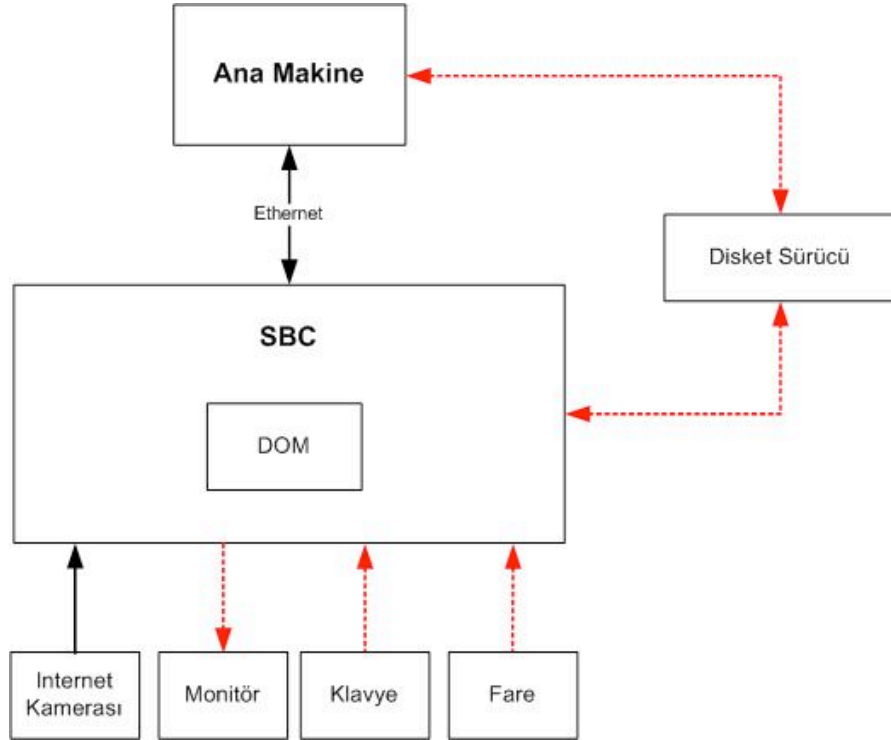
"Fly" marka "WC - OML 3000" model bir Internet kamerası ve 64 MB'lık bir DOM temin edilmiştir.

4.2 Geliştirme Ortamının Hazırlanması

Geliştirme ortamının donanım alt yapısı aşağıdaki gibidir:

- SBC
- Hard Disk görevi gören bir DOM
- Linux ve geliştirme araçlarının bulunduğu bir ana makine
- SBC'yi önyüklemek için bir disket sürücü
- SBC ile ana makine arasında ethernet bağlantısı kurulması için bir çapraz (cross) ethernet kablosu
- Fare, klavye ve monitör
- İnternet kamerası

Aşağıda blok diyagramı verilen geliştirme ortamında kesikli oklar ile gösterilen bağlantılar sadece geliştirme aşaması sırasında kullanılmıştır:



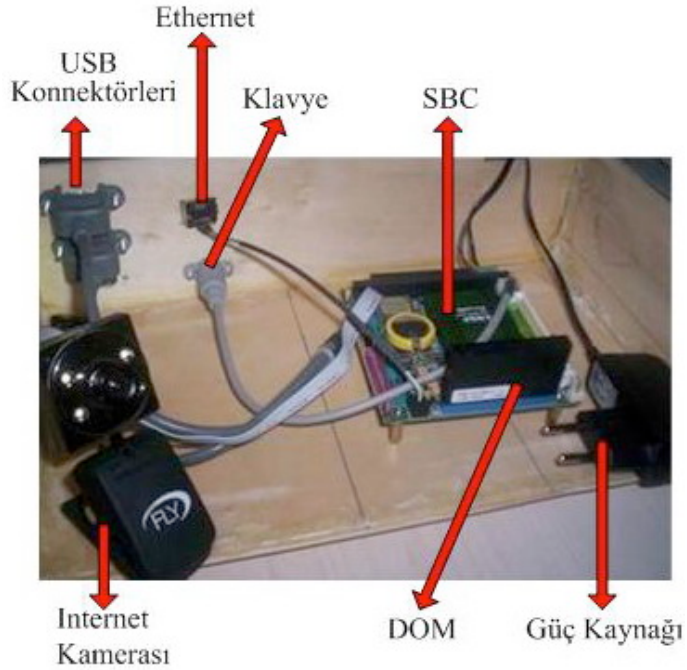
Şekil 4-3: Geliştirme ortamı blok şeması



Şekil 4-4: Geliştirme ortamı

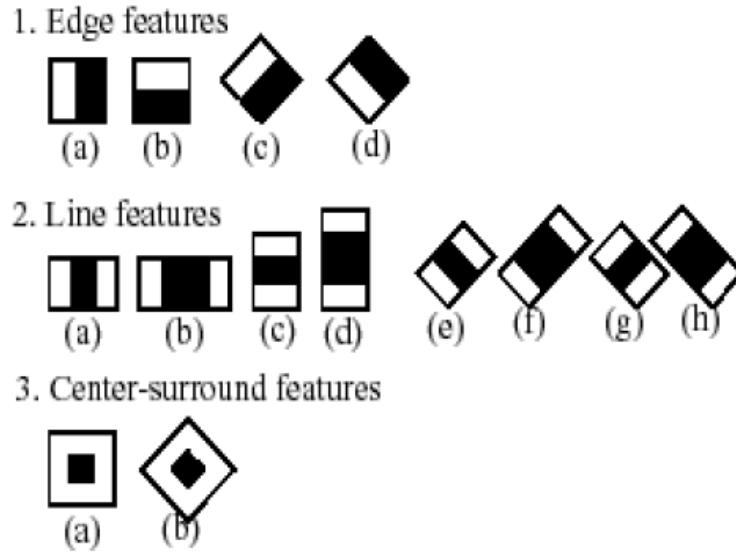
4.3 Özel Uygulama İçin Kullanılacak Yöntem ve Araçların Belirlenmesi

Özel uygulamanın gereksinimlerine göre belirlenen SBC arayüz bağlantıları yapılır (Şekil 4-5).



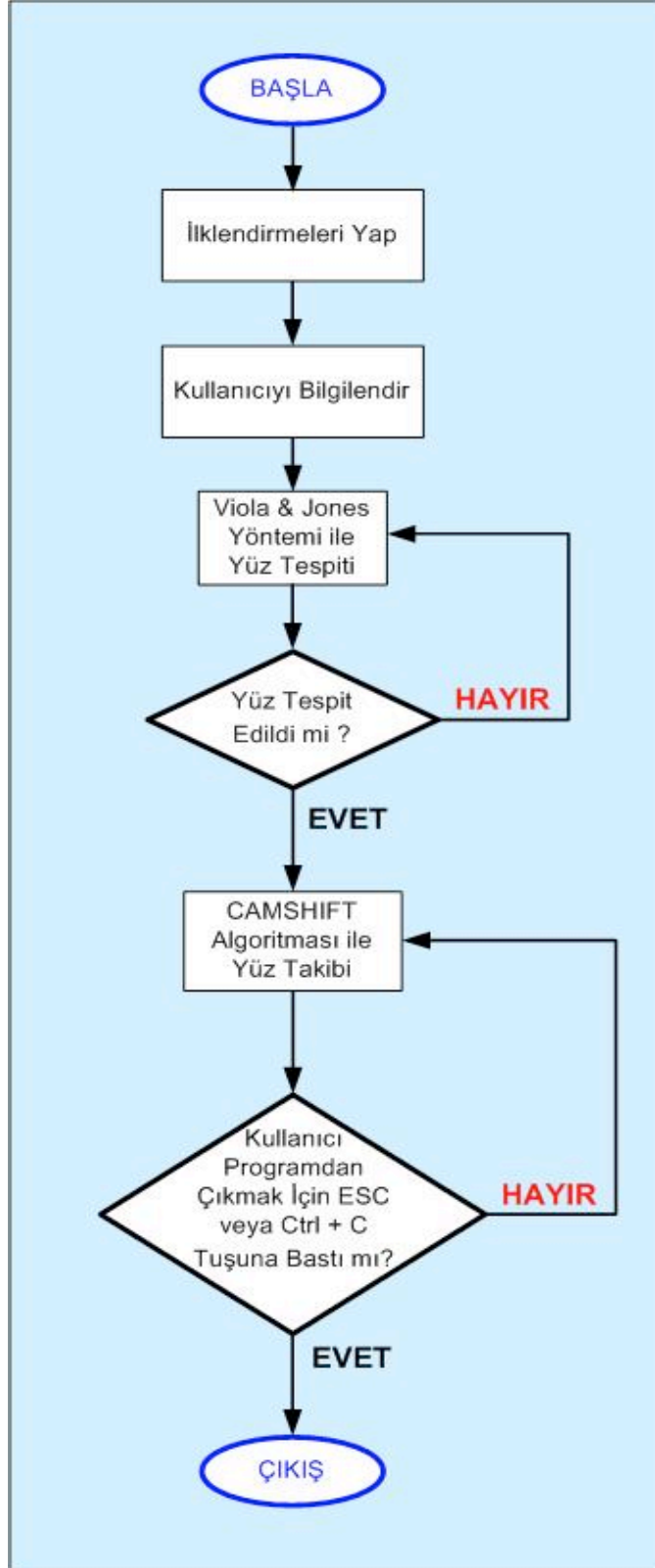
Şekil 4-5: Projede oluşturulan gömülü sistem

Tez çalışması sırasında yüz tespit yöntemi olarak Viola ve Jones algoritması [2], yüz takibi için ise CAMSHIFT algoritması [1] benimsenmiştir. Algoritmaların gerçekleşmesi sırasında OpenCV kütüphanesi, "gcc" derleyicisi ve "Code::Blocks" geliştirme ortamından yararlanılmıştır [59, 13]. Viola ve Jones tarafından önerilen dikdörtgen öznitelikler üzerine eklenti yapılarak geliştirilen öznitelikler kullanılmıştır (Şekil 4-6) [80]:



Şekil 4-6: Genişletilmiş öznitelikler

Yapılan seçimler doğrultusunda, özel algoritmaya ait akış diyagramı belirlenmiştir (Şekil 4-7).



Şekil 4-7: Projeye özel algoritma

5 ANA MAKİNENİN HAZIRLANMASI

5.1 DEBIAN GNU/LINUX Kurulumu

Debian GNU/Linux işletim sisteminin dağıtımı İnternet üzerinden yapılır. İşletim sisteminin DVD'sini elde etmek için uygun mimari tipine (örneğin i386) ait iso uzantılı dosya indirilir ve bir DVD' ye yazdırılır. Tez çalışması için sadece DVD-1 iso dosyası [23] ile İnternet üzerinden dağıtılan bazı paketler kullanılmıştır [36, 42-44].

Debian DVD'si, ana makinedeki DVD sürücüsüne yerleştirilir ve BIOS ayarları öncelikli olarak DVD sürücüsünden ön yüklenecek şekilde yapılarak kurulum ekranına ulaşılır (Şekil 5-1).



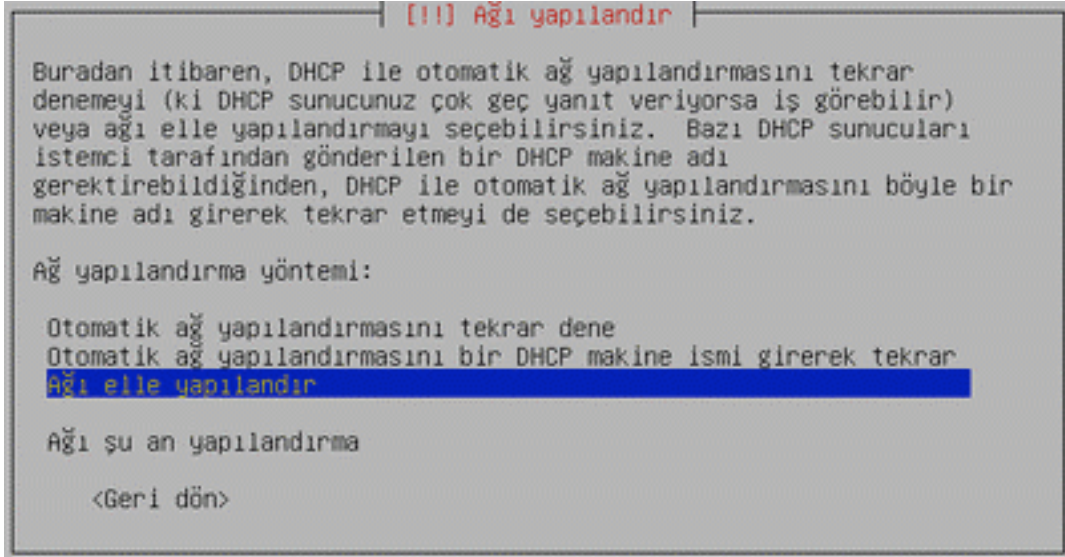
Şekil 5-1: Debian kurulum ekranı

Ana makinede temel kurulum ağ yapılandırma ayarı haricinde resmi kurulum dokümanında belirtildiği gibi yapılır [24]. Temel kurulum tamamlandıktan sonra, otomatik olarak sırasıyla saat dilimi seçimi, root parolası

belirlenmesi ve ana makinede normal kullanıcı haklarına sahip olacak bir kullanıcının eklenmesi işlemleri yapılır.

5.1.1 Kurulum sırasında ağ yapılandırması

Kurulum yöneticisi, ön tanımlı olarak dhcp yapılandırması gerçekleştirir. Ortamda herhangi bir dhcp sunucusu varsa sunucu bulunur ve ağ yapılandırma işlemi tamamlanır. Ancak ortamda dhcp sunucusu yoksa, ekrana ağ yapılandırma menüsü çıkar (Şekil 5-2).



Şekil 5-2: Debian ağ yapılandırma ekranı

Menüden "Ağı elle yapılandır" seçeneği seçilir. Statik IP ataması yapmak için ekrana sırasıyla gelen pencerelerde IP adresi "192.168.1.100", alt ağ maskesi "255.255.255.0", ağ geçidi "192.168.1.1", alan adı sunucusu "192.168.1.255", makine adı "tuba" ve alan adı olarak "tuba" değerleri girilerek bir sonraki kurulum adımına geçilir.

5.2 Kurulum Sonrası Yapılması Gereken Ayarlar

5.2.1 Debian GNU/Linux'ta çekirdek derlenmesi

Çekirdek derlemesi için çekirdek kaynak kodları ve derleme işleminde kullanılan yardımcı araçlar ana makinede kurulu olmalıdır. Yardımcı araçlardan önce "*kernel-package*", "*libncurses5-dev*" ve "*libc6-dev*" paketleri kurulur. Çekirdeğin dağıtım adresinden "*linux-2.6.28.tar.bz2*" dosyası "*/tmp*" dizinine indirilir ve sırasıyla aşağıdaki komutlar girilir [19]:

```
apt-get install build-essential libncurses5-dev
apt-get install libc6-dev gcc make libncurses5-dev kernel-package apache2
apt-cache search kernel-source
apt-get install kernel-source-2.6.28
cd /root/calisma/kernel/
mv /tmp/linux-2.6.28.tar.bz2 .
tar jxf linux-2.6.28.tar.bz2
ln -sf kernel-source-2.6.28 linux
cd /usr/src/linux
cp /boot/CONFIG-2.6 linux-2.6.28/.config
```

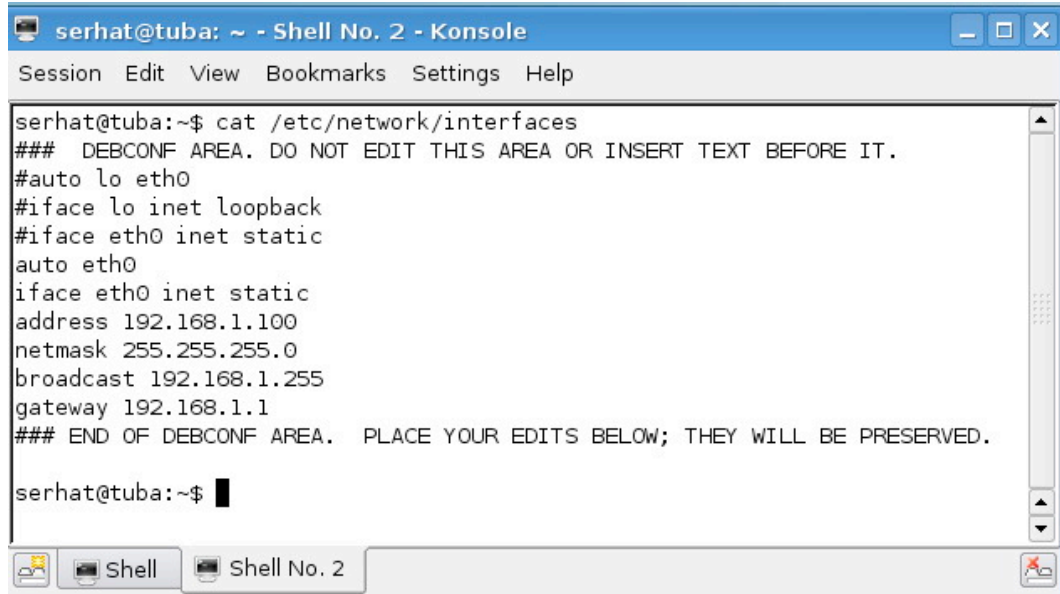
"*CONFIG-2.6*" dosyası mevcut çekirdeğe entegre olan özelliklerin, yeni derlenecek çekirdeğe taşınmasını sağlayan dosyadır. Ön hazırlıkların tamamlanmasının ardından aşağıdaki komutlarla çekirdek derlenerek ana makineye kurulur:

```
make oldconfig
make menuconfig
make-kpkg clean
make-kpkg kernel-image --initrd
cd ..
dpkg -i kernel-image-[yeni_oluşan_çekirdek_görüntü_dosyasının_tam_ismi]
```

Kurulumdan sonra önyükleyici yöneticisi içerisine uygun kayıt otomatikman girilir ve ana makine yeniden başlatılır [25, 58].

5.2.2 Kurulum sonrası ağ yapılandırılması

Debian'da ağ bağlantılarına ilişkin temel ayarlar `/etc/network/interfaces` dosyasında yer alır. IP adresinin sabit olarak elle verilmesinin gerektiği durumlarda `/etc/network/interfaces` dosyasına aşağıdaki gibi bilgiler girilir ve değerlerin aktif hale getirilmesi için `/etc/init.d/networking restart` komutu verilir.



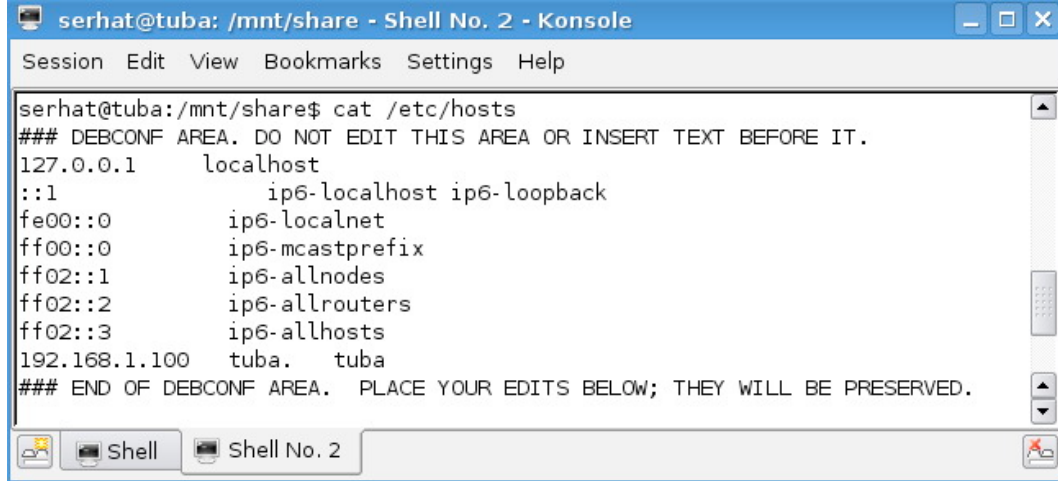
```
serhat@tuba: ~ - Shell No. 2 - Konsolle
Session Edit View Bookmarks Settings Help

serhat@tuba:~$ cat /etc/network/interfaces
### DEBCONF AREA. DO NOT EDIT THIS AREA OR INSERT TEXT BEFORE IT.
#auto lo eth0
#iface lo inet loopback
#iface eth0 inet static
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.1
### END OF DEBCONF AREA. PLACE YOUR EDITS BELOW; THEY WILL BE PRESERVED.

serhat@tuba:~$
```

Şekil 5-3: `/etc/network/interfaces` dosyası

`/etc/hosts` dosyasında ana makine ismi ile ilgili bir giriş olmalıdır. Bilgisayar isminin `tuba` olduğu ve sabit IP adresine sahip bir ana makinede aşağıdaki gibi bir kayıt girişi yapılır:



```
serhat@tuba: /mnt/share - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
serhat@tuba:/mnt/share$ cat /etc/hosts
### DEBCONF AREA. DO NOT EDIT THIS AREA OR INSERT TEXT BEFORE IT.
127.0.0.1    localhost
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
ff02::3     ip6-allhosts
192.168.1.100 tuba. tuba
### END OF DEBCONF AREA. PLACE YOUR EDITS BELOW; THEY WILL BE PRESERVED.
```

Şekil 5-4: /etc/hosts dosyası

“/etc/nsswitch.conf” dosyasında aşağıdaki gibi bir kayıt bulunmalıdır:

hosts: files dns

5.2.3 USB cihaz ayarlamaları

Linux işletim sistemi USB cihazın, USB 2.0 veya USB 1.1 standardında olması durumlarında farklı konfigüre edilir. USB 2.0 bir cihaz için ana makinede “EHCI” ile “UHCI” veya “OHCI” sürücülerinden birisine ihtiyaç vardır. UHCI, Intel tarafından geliştirilen Intel yongalı cihazlarda kullanılan aygıt kontrol arabirimidir. OHCI ise çeşitli firmaların katılımı ile ortaya konulan USB HCI spesifikasyonudur. İki farklı kontrol birimi kullanıldığından dolayı kullanılacak arabirimin hangisi olduğu ilgili Internet adresinden öğrenilir [26]. OHCI sürücüsü “ohci-hcd”, UHCI sürücüsü “uhci-hcd” ve EHCI ise “ehci-hcd” olarak adlandırılır [27].

Bir USB cihazın çalışması için gereken modüller aşağıdaki gibidir:

- Yukarıdaki açıklamalara uygun modüller (USB 1.1 için UHCI veya OHCI USB 2.0 için EHCI vs.)
- Cihazın çekirdek ile irtibatını sağlayan usbcore modülü (usbcore.o)

- Kullanılacak aygıtın türüne uygun modül (videodev, usb_storage, sg, sd_mod, scsi_mod vb.)

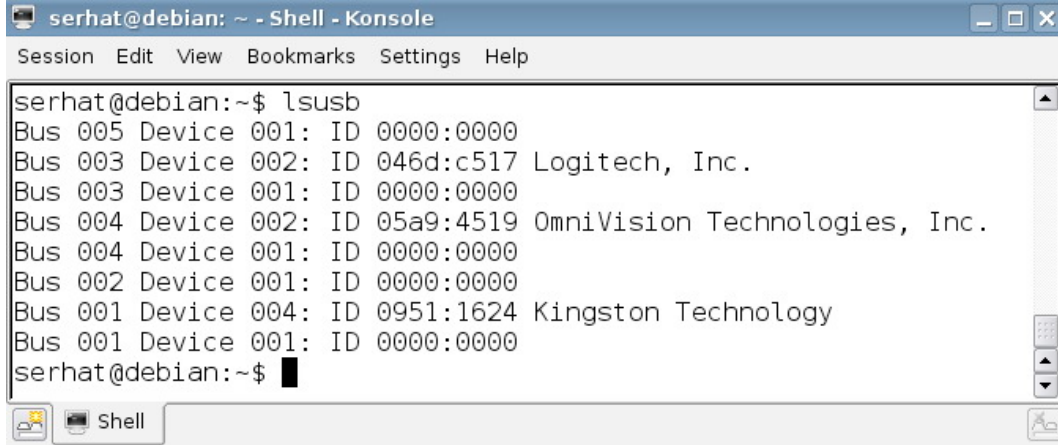
Ana makinede "*lsmod*" komutuyla yüklü modüllerin bir çıktısı alınır ve eksik modül varsa kurulumu yapılır:

Örnek: Ana makinedeki USB modüller

tuba:/home/serhat# *lsmod*

Module	Size	Used by
ov51x_jpeg	135736	0
usb_storage	72000	0
ohci_hcd	18276	0
usbvideo	24388	0
compat_ioctl32	1472	1 usbvideo
videodev	21120	2 ov51x_jpeg,usbvideo
v4l1_compat	12036	1 videodev
v4l2_common	20448	1 videodev
ovcamchip	18120	0
i2c_core	19680	2 ovcamchip,i2c_viapro
pci_hotplug	28704	2 pciehp,shpchp
usbhid	37248	0
ehci_hcd	28136	0
uhci_hcd	21164	0
usbcore	112644	10 ov51x_jpeg, usb_storage, ohci_hcd, usbvideo, usbhid,ehci_hcd,uhci_hcd

"*lsusb*" komutu ile ana makinede takılı USB aygıtların bir listesi elde edilir [29].



```
serhat@debian:~$ lsusb
Bus 005 Device 001: ID 0000:0000
Bus 003 Device 002: ID 046d:c517 Logitech, Inc.
Bus 003 Device 001: ID 0000:0000
Bus 004 Device 002: ID 05a9:4519 OmniVision Technologies, Inc.
Bus 004 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 004: ID 0951:1624 Kingston Technology
Bus 001 Device 001: ID 0000:0000
serhat@debian:~$
```

Şekil 5-5: Ana Makinedeki USB aygıtlar hakkında bilgi edinilmesi

Yukarıdaki listede "*Logitech, Inc.*" satırı USB fare ve klavyesine, "*Kingston Technology*" satırı USB belleğe, "*OmniVision Technologies, Inc.*" satırı ise Internet kamerasına ait bilgileri içerir.

5.2.4 OV51X-JPEG sürücü kurulumu

"*lsusb*" çıktısında yer alan "*05a9:4519*" ID'sine sahip Internet kamerası "*ov51x-jpeg*" modülüne ihtiyaç duyar [30, 31]. Modül kurulumundan önce ana makinede olması gereken özellikler kontrol edilir ve gerekli bileşenler Internet'ten sağlanır [31, 32].

- "*05a9:4519*" ID'li Internet kamerası takıldığında çekirdek otomatik olarak "*ov511*" modülünü yüklemeye çalışacağı için, "*rmmmod ov511*" komutuyla "*ov511/ov518*" Internet kamerasına ait modül ana makineden çıkarılır [33].
- Ana makineye bir derleyici ve mevcut çekirdeğe ait çekirdek kaynağı kurulur [34, 35]:

```
apt-get install libc6-dev gcc make libncurses5-dev kernel-package
apt-cache search kernel-source
apt-get install kernel-source-2.6.28
```

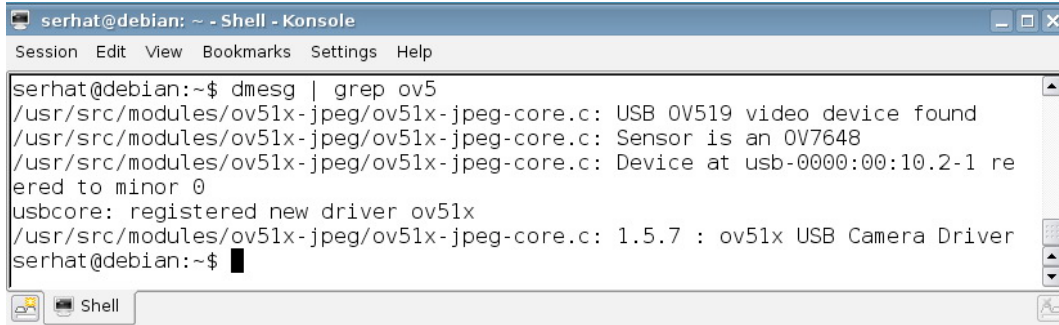
- "ov51x-jpeg-source_1.5.8-1_all.deb" veya daha güncel bir sürücü dosyası İnternette indirilir ve dosyanın bulunduğu dizinde aşağıdaki komutla kurulur:

```
tuba:/home/serhat# dpkg -i ov51x-jpeg-source_1.5.8-1_all.deb
(Reading database ... 85760 files and directories currently installed.)
Preparing to replace ov51x-jpeg-source 1.5.8-1 (using ov51x-jpeg-source_1.5.8-1_all.deb) ...
Unpacking replacement ov51x-jpeg-source ...
Setting up ov51x-jpeg-source (1.5.8-1) ...
```

- Sırasıyla aşağıdaki komutlar verilerek ilgili modül yüklenir:

```
apt-get update
apt-get install ov51x-jpeg-source module-assistant
module-assistant a-i ov51x-jpeg
```

- Sürücünün çalıştığını kontrol etmek için "dmesg" komutu ile OV51X-JPEG sürücüsüne ait mesajlardan İnternet kamerasının sahip olduğu köprü ve sensör chip bilgileri görüntülenir:



```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
serhat@debian:~$ dmesg | grep ov5
/usr/src/modules/ov51x-jpeg/ov51x-jpeg-core.c: USB OV519 video device found
/usr/src/modules/ov51x-jpeg/ov51x-jpeg-core.c: Sensor is an OV7648
/usr/src/modules/ov51x-jpeg/ov51x-jpeg-core.c: Device at usb-0000:00:10.2-1 re
ered to minor 0
usbcore: registered new driver ov51x
/usr/src/modules/ov51x-jpeg/ov51x-jpeg-core.c: 1.5.7 : ov51x USB Camera Driver
serhat@debian:~$ █
```

Şekil 5-6: Sürücü yüklendikten sonraki ov51x-jpeg mesajı

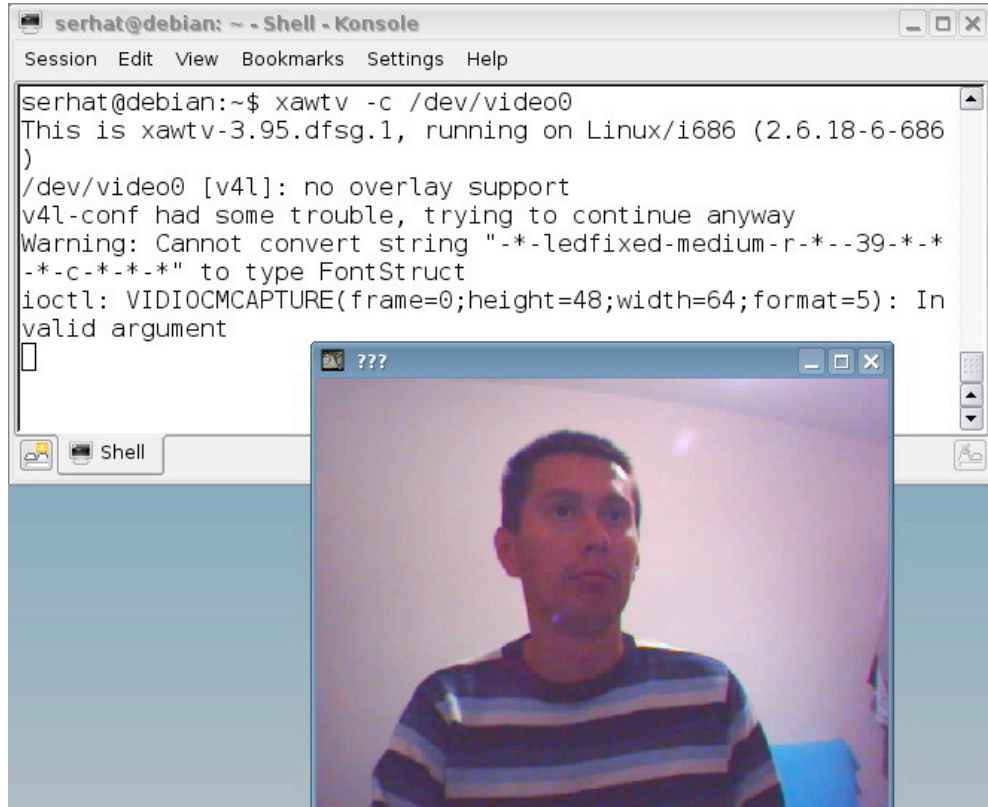
- Sürücü kurulumundan sonra "/dev/video0" düğümünün kurulu olduğu kontrol edilir. Eğer belirtilen düğüm yoksa, "mknod /dev/video0 c 81 0" komutuyla oluşturulur.



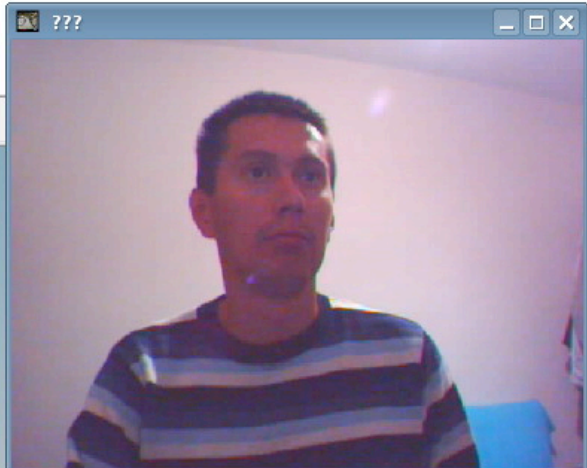
```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
serhat@debian:~$ ls -l /dev/video*
crw-rw---- 1 root video 81, 0 2009-11-11 21:51 /dev/video0
serhat@debian:~$
```

Şekil 5-7: /dev/video içeriği

- Kullanıcıların ve root'un kameraya erişmeleri için "`chmod 666 /dev/video0`" komutu ile okuma ve yazma hakları verilir.
- Kamerayı kullanabilecek herhangi bir uygulama yazılımı (gnomemeeting [38] - xawtv [37]) çalıştırılarak tüm adımların eksiksiz tamamlandığı test edilir. Örneğin, "`xawtv -c /dev/video0`" komutuyla Internet kamerasından alınan görüntü ekranda görülür (Şekil 5-8).



```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
serhat@debian:~$ xawtv -c /dev/video0
This is xawtv-3.95.dfsg.1, running on Linux/i686 (2.6.18-6-686
)
/dev/video0 [v4l]: no overlay support
v4l-conf had some trouble, trying to continue anyway
Warning: Cannot convert string "-*-ledfixed-medium-r-*--39-*-*
-*-c-*-*-*" to type FontStruct
ioctl: VIDIOCMACTURE(frame=0;height=48;width=64;format=5): In
valid argument
[]
```



Şekil 5-8: xawtv uygulaması ile modül kurulumunun test edilmesi

5.2.5 OpenCV kütüphanesi kurulumu

OpenCV kütüphanesinin 1.1 versiyonuna ait sıkıştırılmış kaynak kod dosyası İnternette indirilip açıldıktan sonra aşağıdaki komutlarla kurulur:

```
cd opencv-1.1.0/  
./configure --prefix=/opt/opencv --without-quicktime --without-zlib --without-ffmpeg  
make  
make install
```

Kurulumun ardından çevresel değişkenler ile ilgili eklentiler yapılır: "`nano /etc/ld.so.conf.d/opencv.conf`" komutu ile bir dosya yaratılır ve dosyaya aşağıdaki satırlar eklenir:

Örnek: /etc/ld.so.conf.d/opencv.conf dosyası

```
/lib  
/usr/lib  
/opt/opencv/lib
```

"`opencv.conf`" dosyasına eklenen kütüphane yolları ile `pkg-config` ayarlarının sisteme kaydedilmesi için aşağıdaki komutlar girilir:

```
ldconfig  
cd /kaynak_kodunun_yer_aldigi_dizin  
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/opt/opencv/lib/pkgconfig  
export PKG_CONFIG_PATH
```

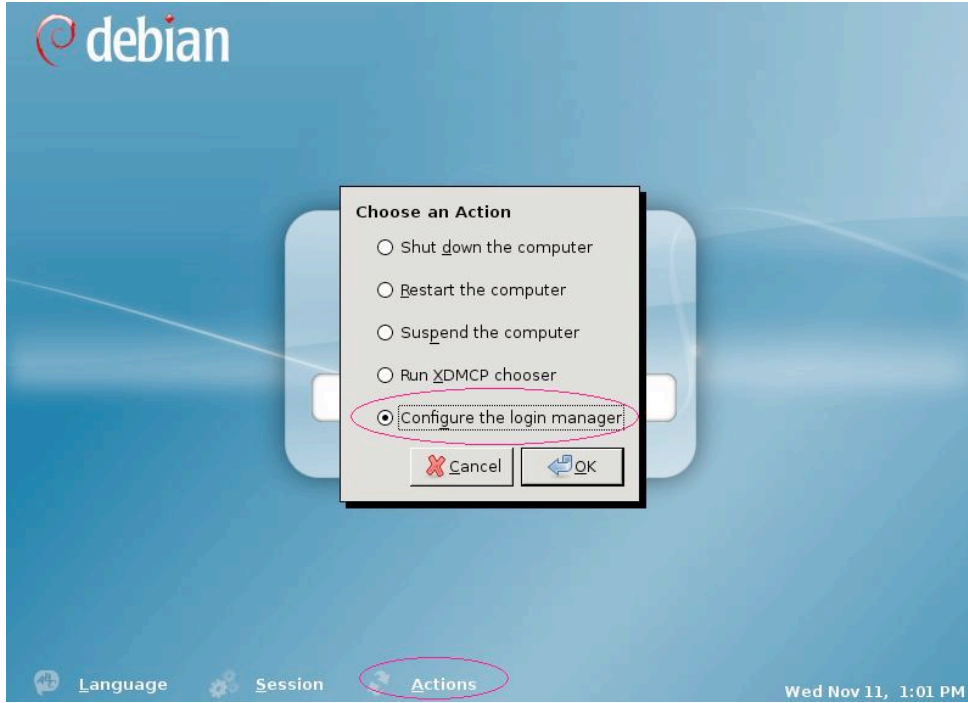
`pkg-config` için girilen değerlerin kontrolü aşağıda belirtildiği şekilde yapılır:

```
tuba:/home/serhat# pkg-config --cflags opencv  
-I/opt/opencv/include/opencv  
tuba:/home/serhat# pkg-config --libs opencv  
-L/opt/opencv/lib -lxcvcore -lev -lhighgui -lcvaux -lml
```

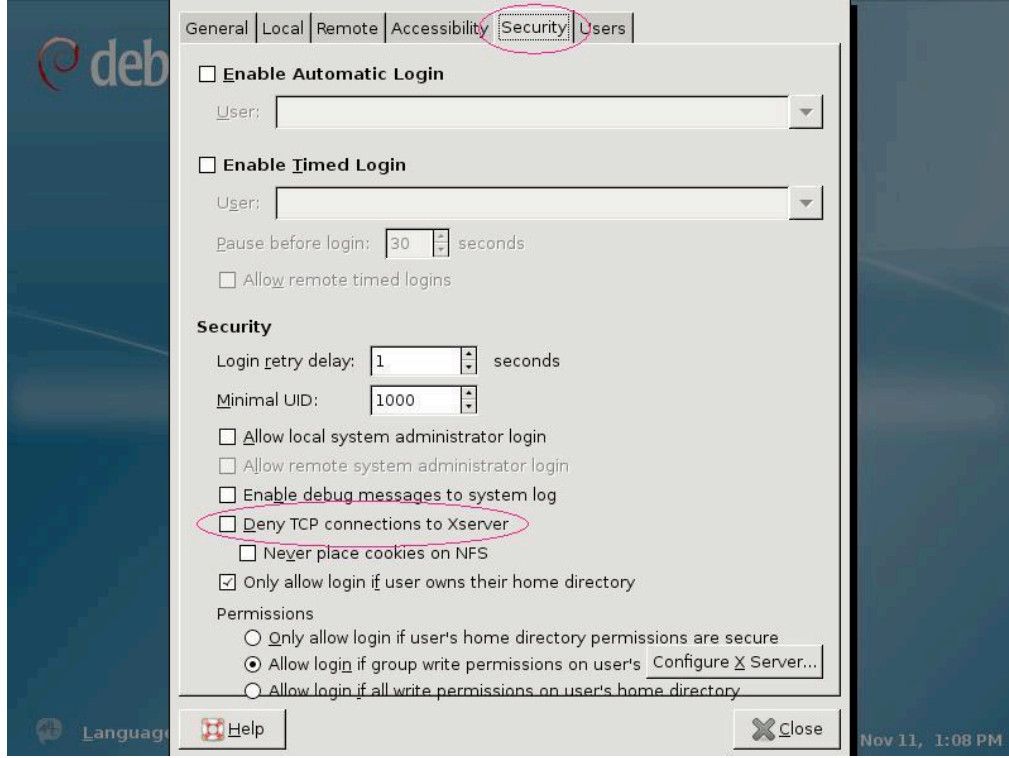
Kaynak kodunu derlemek ve çalıştırılabilir bir dosya yaratmak için "`gcc`
`pkg-config --cflags opencv` `pkg-config --libs opencv`
-o YuzTakip main.c`" komutu verilir.

5.2.6 Sistemden görüntünün alınması için gerekli ayarlamalar

SBC'lerde grafik ekranlı bir işletim sistemi kurmak yerine, sistemin görüntüsü ethernet üzerinden ana makinedeki bir pencereye yönlendirilir. Ethernet üzerinden veri alınması için ana makinede bazı ayarlar yapılır.



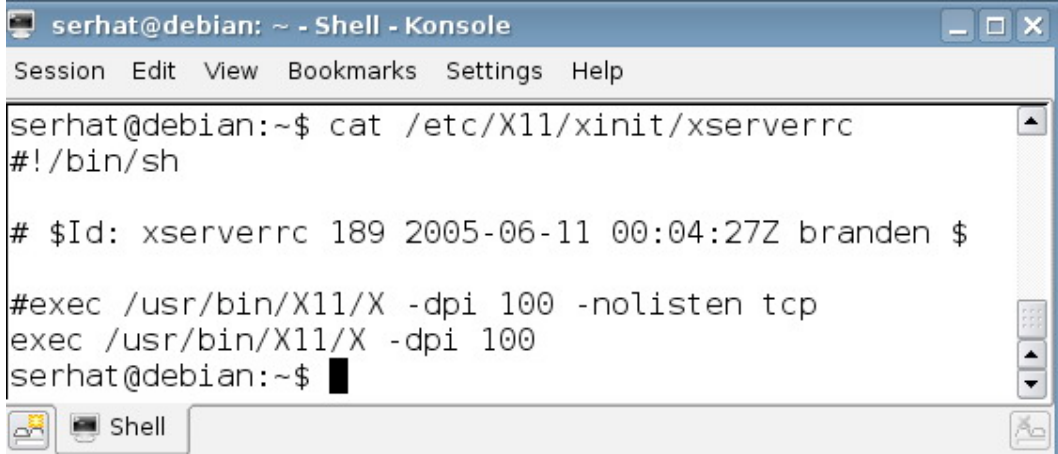
Şekil 5-9: Grafiksel karşılayıcı ayar penceresi



Şekil 5-10: “Deny TCP connections to Xserver” işareti

Ana makinede güvenlik gereği “Deny TCP connections to Xserver” isimli seçenek öntanımlı olarak seçili geldiğinden, ana makinedeki X Window TCP portunu dinlemez ve sistemden gelecek görüntü bilgisinin alınmasına engel olur. Dinlemeyi sağlayabilmek için eğer gdm kurulu değilse “`apt-get install gdm`” komutu ile ana makineye kurulur ve “Ön Tanımlı Grafiksel Karşılılayıcı” olarak gdm seçilir. Ana makine yeniden başlatıldığında grafik ekran “gdm” grafiksel yöneticisi ile başlar. Kullanıcı adı girmeden önce altta yer alan “Actions” seçeneğine tıklanır (Şekil 5-9). Ekranı gelen pencerede “Security” bölümüne girilerek “Deny TCP connections to Xserver” isimli seçenekteki işaret kaldırılır (Şekil 5-10).

Ana makinedeki X Window’un TCP portunu dinlemesini sağlamak için bir konsol ekranı açılır ve “`nano /etc/X11/xinit/xserverrc`” komutu verilir. “xserverrc” dosyasındaki “`-nolisten tcp`” ibaresi kaldırılır:



```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

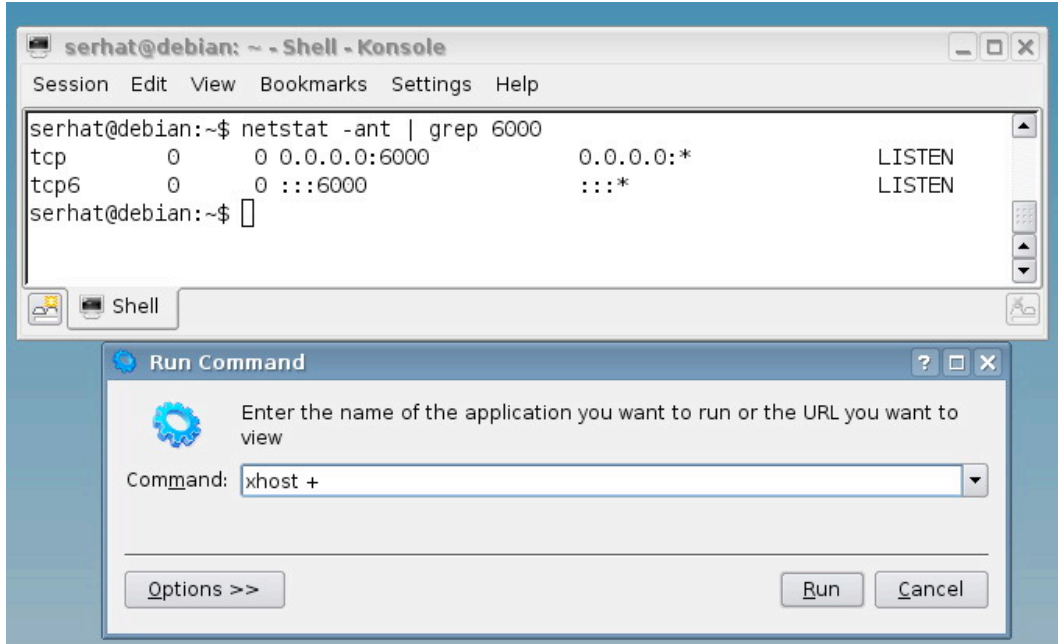
serhat@debian:~$ cat /etc/X11/xinit/xserverrc
#!/bin/sh

# $Id: xserverrc 189 2005-06-11 00:04:27Z branden $

#exec /usr/bin/X11/X -dpi 100 -nolisten tcp
exec /usr/bin/X11/X -dpi 100
serhat@debian:~$
```

Şekil 5-11: /etc/X11/xinit/xserverrc dosyası

Yapılan ayarların kontrolü için "*netstat -ant | grep 6000*" komutu verilir. Eğer Şekil 5-12' deki çıktı alınırsa, sistemin ana makinedeki X Window'a erişimi için gerekli tüm ayarlar tamamlanmış demektir. Görüntü alabilmek için "*ALT+F2*" ile komut çalıştır ekranına gelinir ve "*xhost +*" yazılarak her kullanıcıya mevcut X Window oturumu için erişim kontrol hakkı verilir (Şekil 5-12).



```
serhat@debian:~$ netstat -ant | grep 6000
tcp        0      0 0.0.0.0:6000          0.0.0.0:*            LISTEN
tcp6       0      0 :::6000              :::*                  LISTEN
serhat@debian:~$
```

Run Command

Enter the name of the application you want to run or the URL you want to view

Command: xhost +

Options >> Run Cancel

Şekil 5-12: X Window erişim hakkının verilmesi

5.2.7 Sistemde çalışacak çekirdeğin derlenmesi

Sistemde çalışacak çekirdeği derlemek için aşağıdaki komutlar girilir:

```
cd /root/calisma/  
mkdir kernel  
cd kernel/  
mv /tmp/linux-2.6.28.tar.bz2 .  
tar jxf linux-2.6.28.tar.bz2  
cp /home/serhat/.CONFIG-2.4.27 CONFIG-2.6  
cp CONFIG-2.6 linux-2.6.28/.config  
cd linux-2.6.28  
make oldconfig  
make menuconfig  
make bzImage  
make modules  
make modules_install
```

Yukarıdaki komutlarla, SBC için derlenecek çekirdeğe ait kaynak dosyası "*calisma*" dizini altında oluşturulan kernel dizinine kopyalanarak "*tar jxf*" komutu ile açılır. Ardından daha önceden elimizde olan 2.4.27 çekirdeğine ait çekirdek konfigürasyon dosyası kernel dizinine "*CONFIG-2.6*" adıyla kopyalanır. Daha sonra ise CONFIG-2.6 dosyası yeni oluşan linux-2.6.28 dizini altına "*.config*" adı ile kopyalanır. Burada amaçlanan çekirdek derleme esnasında oluşabilecek sorunlarda "*.config*" dosyasının orijinal halinin korunmasıdır. "*linux-2.6.28*" dizinine geçerek çekirdek derleme komutları girilir. Yapılandırmanın tamamlanmasının ardından sıkıştırılmış bir çekirdek görüntüsüne sahip olabilmek için "*make bzImage*" komutu verilir. Çekirdekte video ile ilgili kısımlar modül olarak konfigüre edildiği için, "*make modules*" ve "*make modules_install*" komutları icra edilerek modüller eklenir.

5.2.8 Sistemde çalışacak çekirdeğe Internet kamerasına ait modülün eklenmesi

Kameraya ait modül sistemde çalışacak çekirdeğe aşağıdaki komutlarla eklenir:

```
cd /root/calisma/kernel
wget http://www.rastageeks.org/downloads/ov51x-jpeg/ov51x-jpeg-1.5.9.tar.gz
tar zxvf ov51x-jpeg-1.5.9.tar.gz
cd ov51x-jpeg-1.5.9
vi Makefile
make
make install
cd ..
cd linux-2.6.28
cp arch/i386/boot/bzImage /var/www/
```

Internet kamerasına ait sürücü dosyası Internetten kernel dizinine indirilir ve "*tar zxvf*" komutu ile açılır. Sürücü makefile dosyaları, ön tanımlı olarak modülü ana makinede çalışan çekirdek dizinine eklemeye çalıştığından, makefile'da "*KDIR := /lib/modules/\$(shell uname -r)/build*" satırının hemen ardından "*KDIR := /lib/modules/2.6.28/build*" diye yeni bir satır yazarak modülün sistem için hazırlanan çekirdeğe ekleneceği belirtilir. "*make*" ve "*make install*" komutları ile sürücü dosyası oluşturulur. Yeni çekirdekle sistemi açabilmek için, "*arch/i386/boot/bzImage*" isimli çekirdek görüntüsü sistem tarafından alınmak üzere "*/var/www/*" dizinine kopyalanır.

5.2.9 Code::Blocks Kurulumu

Code::Blocks açık kaynak kodlu C/C++ dilleri için çapraz-platform bir "*entegre geliştirme ortamıdır (integrated development environment - IDE)*" [6, 13]. Code::Blocks IDE'si projenin ihtiyaç duyduğu derleyici ve hata ayıklayıcıyı içerir. Kurulum dosyaları Internetten

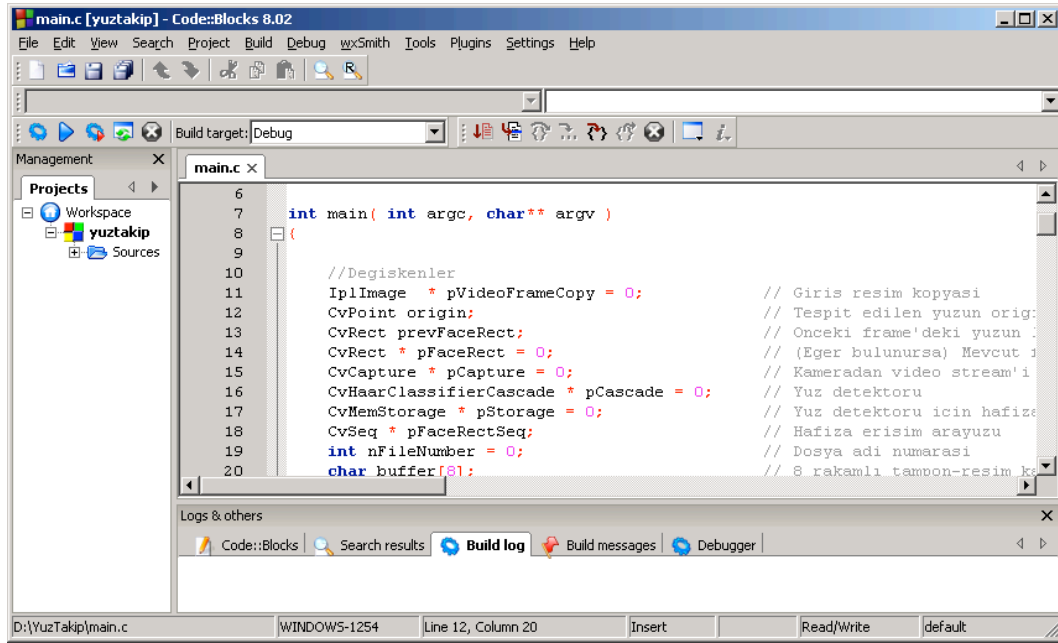
indirilip, codeblocks isimli dizine açıldıktan sonra aşağıdaki komutlarla kurulur [13]:

apt-get install libwxbase2.8-0

apt-get install libwxgtk2.8-0

(codeblocksa ait deb uzantılı dosyaların olduğu dizinin bir üstünde aşağıdaki komut verilir)

dpkg -i -R codeblocks/



Şekil 5-13: Code::Blocks

6 SBC’NİN HAZIRLANMASI

SBC’nin ařađıda belirtilen donanımsal bađlantılarının yapıldığı kontrol edilir (řekil 4-5):

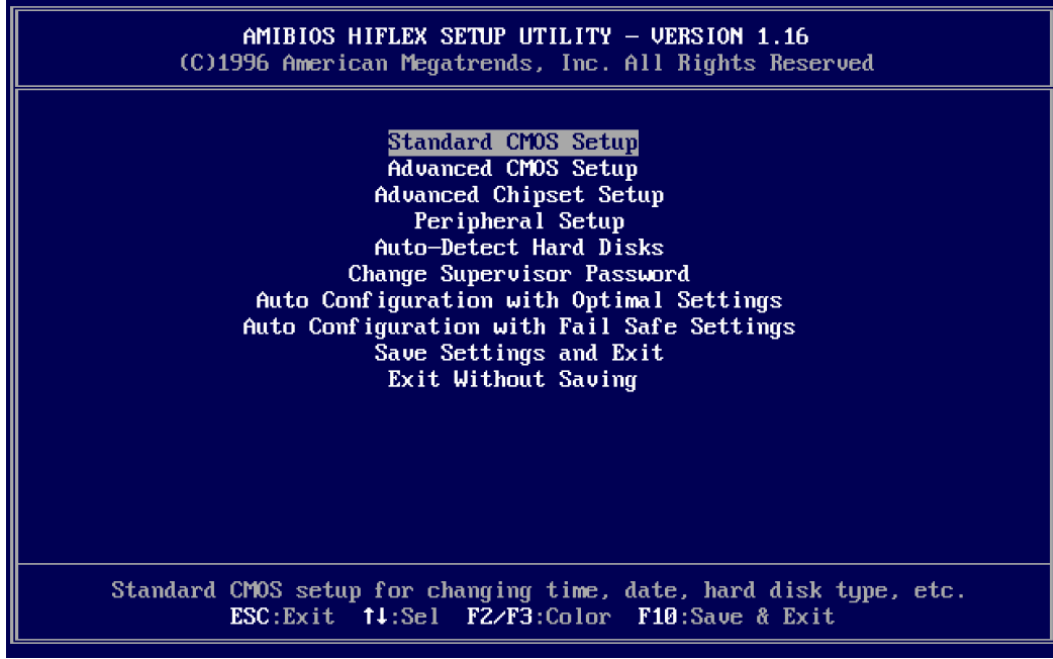
- VGA monitör
- Klavye
- Güç bađlantısı
- SBC’nin önyüklenebilmesi için bir disket sürücü
- Internet kamerası
- Ethernet çıkışına bir çapraz kablo
- DOM

6.1 SBC’yi Disketten Başlatma

Ana makinede Debian GNU/Linux’a ait “*rescue.bin*” ve “*root.bin*” açılıř disket görüntüleri ařađıdaki komutlarla disketlere yazdırılır:

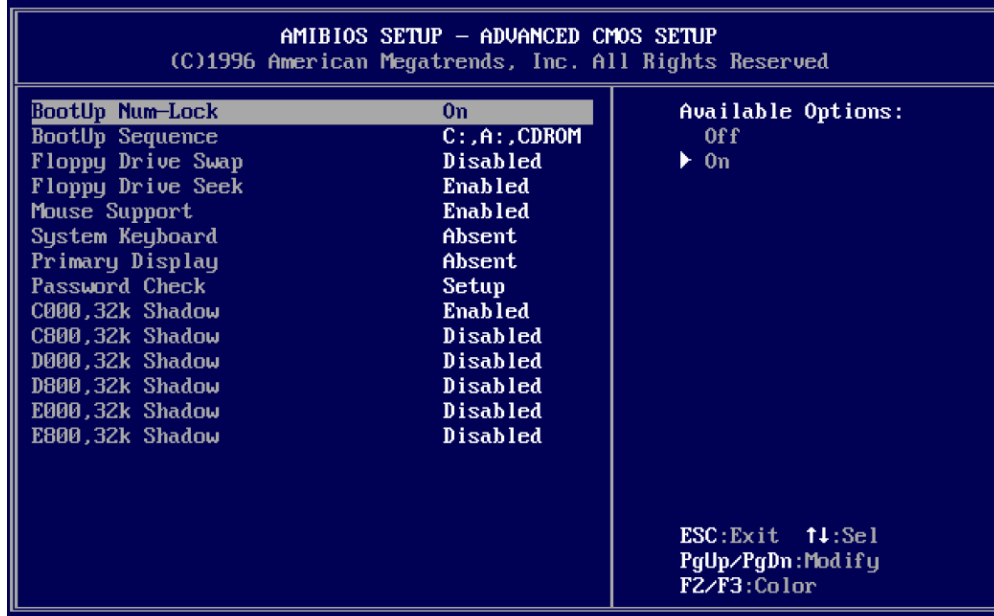
Örnek – Açılıř Disketleri
dd if=rescue.bin of=/dev/fd0
dd if=root.bin of=/dev/fd0

SBC’yi disket sürücüsünden başlatmak için, açılıř esnasında “*DEL*” tuřu ile AMI BIOS ekranına geçilir ve “*Advanced CMOS Setup*” seçeneđine girilir (řekil 6-1).



Şekil 6-1: Sistem BIOS Ekranı

"*BootUp Sequence*" seçeneği sistem disket sürücüsünden başlatılacak şekilde ayarlanır (Şekil 6-2).



Şekil 6-2: Advanced CMOS SETUP Ekranı

SBC, BIOS ayarlarının ardından sırasıyla rescue ve root disketleri takılarak çalıştırılır. Önyükleme sonrasında Debian GNU/Linux kurulum programı ekrana geldiğinde (Şekil 5-1), "Alt + F2" tuşlarına basılarak ikinci konsol ekranına geçilir.

6.2 DOM'un Hazırlanması

Sistem başlatıldıktan sonra DOM bölümlendirilir ve bir disk bölümünün maksimum 64 MB olabildiği minix dosya sistemine göre biçimlendirilir [39]. İkinci konsolda aşağıdaki komutla DOM'un "/dev/hda" ve 64 MB olarak tanıdığı kontrol edilir:

```
Örnek - Disk Kontrolü  
#dmesg | grep -i disk  
RAM disk driver initialized:  
16 RAM disks of 16384K size  
hda:  
64MB ATA Flash Disk, ATA DISK driver  
...
```

DOM üzerinde 64 MB'lik bir Linux disk bölümü yaratmak için aşağıdaki komutlar icra ettirilir:

```
Örnek - Disk Bölümleme  
# fdisk /dev/hda  
Command (m for help): n  
Command action e extended p primary partition (1-4) p  
Partition number (1-4): 1  
First cylinder (1-489, default 1): 1  
Last cylinder or +size or +sizeM or +sizeK (1-489, default 489): 489  
Command (m for help): a  
Partition number (1-4): 1  
Command (m for help): w  
The partition table has been altered!  
Calling ioctl() to re-read partition table.  
Syncing disks.
```

#

Ana makinedeki "*mkfs.minix*" dosyası bir diskete kopyalandıktan sonra disket SBC'ye bağlı disket sürücüyeye takılır. Aşağıdaki komutlarla sistemde minix dosya sistemi yaratılır:

Örnek – Minix Dosya Sisteminin Yaratılması

```
# mount /dev/fd0 /floppy
# ls /floppy
mkfs~1.min
# cp /floppy/mkfs~1.min /tmp/mkfs.minix
# umount /floppy
# /tmp/mkfs.minix /dev/hda1
20864 inodes
62576 blocks
Firstdatazone=665 (665)
Zonesize=1024
Maxsize=268966912
```

Kullanılan disket MS-DOS formatlıdır. Sistem "*mkfs.minix*" dosyasını "*mkfs~1.min*" olarak gördüğünden, sisteme "*cp /floppy/mkfs~1.min /tmp/mkfs.minix*" komutuyla taşınır. Yukarıdaki son komutun icrasıyla da DOM "*mkfs.minix*" dosya sistemine dönüştürülür.

"/yeni" isminde bir dizin açılır ve DOM bu dizine bağlanarak üzerinde çalışılır hale getirilir [60].

Örnek - /yeni dizini açılması ve DOM'un dizine bağlanması

```
# mkdir /yeni
# mount /dev/hda1 /yeni
# mount
/dev/ram0 on / type ext2 (rw)
/proc on /proc type proc (rw)
/dev/hda1 on /yeni type minix (rw)
#
```

6.3 Temel Linux Ayarları

6.3.1 Temel Dizinlerin Oluşturulması

Aşağıdaki komutlarla temel Linux dizinleri DOM üzerinde oluşturulur [40].

```
# cd /yeni
#mkdir -p bin boot etc floppy home root lib proc sbin tmp usr \
var usr/bin usr/lib usr/sbin var/lib var/lock var/log var/run var/tmp
```

6.3.2 Dizinlere Erişimlerin Düzenlenmesi

"/tmp" dizini geçici işler için sistemdeki tüm kullanıcılar tarafından yazılabilir bir alan olmalı ve sticky bit'i ile verilirken, "/root" dizinine de sadece root kullanıcısı girmelidir [40]. Belirtilen erişim hakları aşağıdaki şekilde düzenlenir:

```
# chmod -R 1777 tmp
# ls -lad tmp
drwxrwxrwt 2 root root 64 Aug 25 19:53 tmp
# chmod 700 root
```

6.3.3 /dev Dizini Altında Aygıtların Oluşturulması

Aygıtlara erişmekte kullanılan özel dosyalar "/dev" dizini altında bulunur [40]. Sistem için aygıt dosyaları açılış disketlerinin belleğe yüklediği "/dev" dizininden "cp -a /dev ." komutuyla oluşturulur. Bellekte yer almayan aygıt dosyaları "mknod" komutu ile majör, minor numaraları, karakter veya blok veri transferi özelliklerine göre sonradan elle oluşturulur. Çalışır durumdaki SBC' nin aygıt listesi aşağıdaki gibidir:

```
/proc # cat devices
Character devices:
```

1 mem
2 pty
3 ttyp
4 ttyS
5 cua
7 vcs
10 misc
81 video_capture
128 ptm
136 pts
162 raw
180 usb

Block devices:

2 fd
3 ide0

6.3.4 Kütüphanelerin Taşınması

Sistemde çalışacak her uygulamanın bağlı olduğu kütüphane dosyaları ana makineden alınarak, sistemde olması gereken dizinlere disket yardımıyla birebir kopyalanır. MS-DOS formatlı disket ana makinede "*mkfs.minix /dev/fd0*" komutuyla minix formatında biçimlendirilir. Aşağıda adları geçen temel kütüphane dosyaları ana makinedeki "*/lib*" dizininden diskete kopyalanır:

- libc.so.6
- ld-linux.so.2
- libdl.so.2
- libnsl.so.1
- libm.so.6
- libcrypt.so.1

Disket sisteme bağlanarak aşağıdaki komutlarla `"/lib"` dizinine kopyalanır ve `"libc.so.6"` ve `"ld-linux.so.2"` kütüphanelerine çalıştırılabilir dosya erişimi verilir:

```
# mount /dev/fd0 /floppy
# cp /floppy/* /yeni/lib
# chmod 755 /yeni/lib/libc.so.6
# chmod 755 /yeni/lib/ld-linux.so.2
```

6.3.5 Temel Uygulamaların Busybox İle Sisteme Kurulması

Linux'ta yer alan temel uygulamaların gömülü sistemlerde kullanımı için içerdikleri fonksiyonlar azaltılarak uygulama ebatlarının düşürüldüğü ve tek bir uygulama haline getirilerek busybox isimli bir uygulama oluşturulmuştur [41]. Busybox'ın içerdiği fonksiyonlar aşağıdaki şekilde listelenir:

```
tuba:/home/serhat/busybox-1.15# ./busybox
Using fallback suid method
BusyBox v1.15 (2009.06.30-16:59+0000) multi-call binary
```

```
Usage: busybox [function] [arguments]...
or: [function] [arguments]...
```

BusyBox is a multi-call binary that combines many common Unix utilities into a single executable. Most people will create a link to busybox for each function they wish to use, and BusyBox will act like whatever it was invoked as.

Currently defined functions:

[, ash, basename, bunzip2, busybox, bzip, cat, chgrp, chmod, chown, chroot, chvt, clear, cmp, cp, cut, date, dd, deallocvt, df, dirname, dmesg, du, echo, egrep, env, expr, false, fdisk, fgrep, find, free, grep, gunzip, gzip, halt, head, hexdump, hostname, id, ifconfig, inetd, init, install, kill, killall, klogd, linuxrc, ln, logger, login, ls, mkdir, mkfs.minix, mknod, mktemp, more, mount, mv, openvt, pidof, ping, pivot_root, poweroff, ps, pwd, readlink, reboot, reset, rm, rmdir, route, sed, sh, sleep, sort, strings, swapoff, swapon, sync, syslogd, tail, tar, tee, telnet, telnetd, test, time, touch, tr, true, tty, umount, uname, uniq, unzip, uptime, usleep, vi, wc, wget, which, whoami, xargs, yes, zcat

Busybox'ı kurmak için kaynak kodları Internet adresinden indirilir, açılır ve kaynak kodunun bulunduğu dizine geçilir [42]. "make" ve "make install" komutları verildiğinde "_install" adında bir alt dizin otomatik olarak oluşur. Oluşan dizin tgz arşivi haline getirilerek bir disket yardımıyla SBC sisteme taşınır.

```
tuba:/home/serhat/busybox-1.15# cd _install/  
tuba:/home/serhat/busybox-1.15/_install# tar cvfz /tmp/busy-bin.tgz .  
./  
./bin/  
./bin/busybox  
./bin/mkdir  
....
```

(Yukarıda adı geçen busybox içindeki tüm uygulamalar tam path'leriyle listelenecektir.)

"busy-bin.tgz" isimli arşiv dosyası disket yardımıyla SBC'ye kopyalanır ve açılır:

```
# cd /yeni/bin  
# cp /floppy/busy-bin.tgz .  
# tar zxvf busy-bin.tgz
```

6.4 Sistemin Kendi Başına Yüklenebilmesi İçin Gerekli Ayarlamalar

6.4.1 SBC'ye Özgün Çekirdeğin Oluşturulması

SBC sistemi üzerinde çalışacak Linux çekirdeğinin derlenmesi ana makinede "5.2.7 Sistemde Çalışacak Çekirdeğin Derlenmesi" kısmında anlatıldığı gibi yapılır. Bir disket yardımı ile derlenen çekirdek, SBC üzerindeki "/yeni/boot" dizini altına "linux" adıyla kopyalanır.

6.4.2 Sistemin Başlatılmasından Sonra Kullanılacak Programların Ayarlanması

"*/etc/init.d/rcS*" dosyasına sistem açılırken yapılması istenen işlemler ve zorunlu adımlar eklenir. Aşağıdaki komutlarla sistemde bir *init.d* dizini ve *rcS* dosyası yaratılarak dosyaya gerekli yetkilendirme yapılır:

```
# mkdir /yeni/etc/init.d  
# vi /yeni/etc/init.d/rcS  
# chmod 755 /yeni/etc/init.d/rcS
```

```
serhat@tuba: ~ - Shell No. 3 - Konsole
tuba:/home/serhat# cat /etc/init.d/rcS
#!/bin/sh

# proc dosya sisteminin bağlanması
mount -t proc proc /proc

# root dosya sisteminin okuma-yazma modunda tekrar bağlanması
mount -n -o remount,rw /

# devpts dosya sistemini mount et
mount -t devpts devpts /dev/pts

# Network Ayarlarını yap
IP=192.168.1.160
NETMASK=255.255.255.0
GW=191.168.1.1

# Loopback interface
echo -n "Loopback aktifleştiriliyor..."
ifconfig lo 127.0.0.1 netmask 255.0.0.0 up
echo " Tamam."

echo -n "Ağ ayarları yapılıyor..."
ifconfig eth0 $IP netmask=$NETMASK up
route add default gw $GW
echo " Tamam."

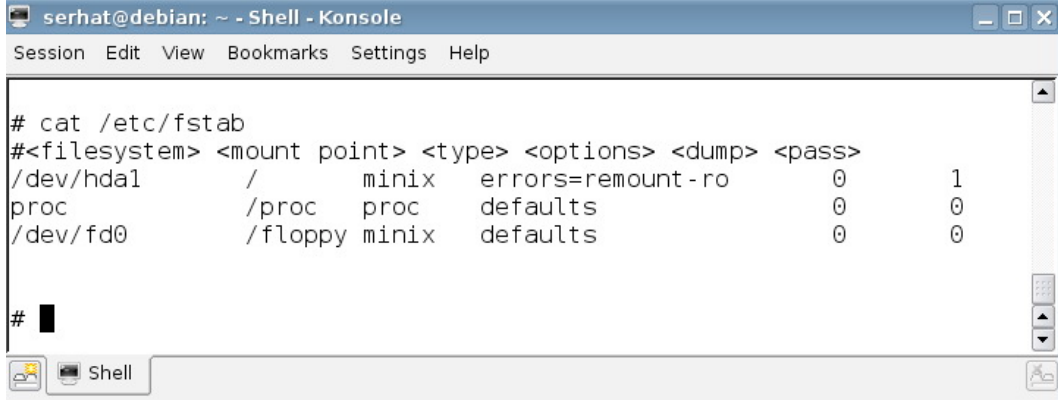
echo -n "Inetd servisi başlatılıyor..."
inetd
echo " Tamam."

echo -n "Telnet servisi başlatılıyor..."
telnetd
echo " Tamam."

echo -n "Modüller yükleniyor..."
modprobe ehci_hcd
modprobe ohci_hcd
modprobe ov51x_jpeg
echo " Tamam."
```

Şekil 6-3: /etc/init.d/rcS dosyası

Sistemin bağlanacağı aygıtlar `"/etc/fstab"` dosyasında tanımlanır:



```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

# cat /etc/fstab
#<filesystem> <mount point> <type> <options> <dump> <pass>
/dev/hda1      /          minix  errors=remount-ro    0      1
proc           /proc     proc   defaults              0      0
/dev/fd0       /floppy   minix  defaults              0      0

#
```

Şekil 6-4: /etc/fstab dosyası

6.4.3 Önyükleyicinin Oluşturulması

Ana makinedeki "lilo" ön yükleyicisi bir disket ile sistemde "/yeni/sbin" dizini altına, ön yükleyicinin bağımlı olduğu aşağıdaki dosyalar ise "/yeni/boot" dizini altına kopyalanır.

- /boot/boot-menu.b
- /boot/boot.b
- /boot/chain.b
- /boot/map

Bellekte yer alan kök dizini "/" yerine, "chroot" komutuyla "/yeni" dizini kök dizini olarak atanır:

```
#chroot /yeni
BusyBox v1.00 (2004.11.27-10:14+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.
/
```

"vi /etc/lilo.conf" komutuyla sistemin açılış sırasında ihtiyaç duyduğu bilgilerin sisteme tanıtıldığı bir ayar dosyası aşağıdaki gibi oluşturulur:

```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

# cat /etc/lilo.conf
lba32
boot=/dev/hda
install=/boot/boot-menu.b
map=/boot/map
prompt
timeout=20
vga=normal
default=Linux-2.6.28

image=/boot/vmlinuz-2.6.28
# Asagidaki append satiri bu cekirdek versiyonunun
# daha sessiz boot etmesini sagliyor.
#append="quiet"
root=/dev/hda1
label=Linux-2.6.28
read-only

image=/boot/vmlinuz-2.6.30.5
# Asagidaki append satiri bu cekirdek versiyonunun
# daha sessiz boot etmesini sagliyor.
#append="quiet"
root=/dev/hda1
label=Linux-2.6.30.5
read-only

# █
```

Şekil 6-5: /etc/lilo.conf dosyası

"Ana Ön Yükleme Kaydı (Master Boot Record - MBR)" bölümünün açılışa uygun hale getirilmesi için aşağıdaki komut girilir:

```
/sbin # lilo -v
```

```
LILO version 22.2, Copyright (C) 1992-1998 Werner Almesberger  
Development beyond version 21 Copyright (C) 1999-2001 John Coffman  
Released 05-Feb-2002 and compiled at 20:57:26 on Apr 13 2002.  
MAX_IMAGES = 27
```

```
Reading boot sector from /dev/hda  
Merging with /boot/boot-menu.b  
Boot image: /boot/vmlinuz-2.6.30  
Added Linux-2.6.30 *
```

```
Boot image: /boot/vmlinuz-2.6.28
```

Added Linux-2.6.28

/boot/boot.0300 exists - no backup copy made.

Writing boot sector.

İşlemleri aktif etmek için sistem yeniden başlatılır ve sistem kendi başına çalışabilir duruma getirilir:

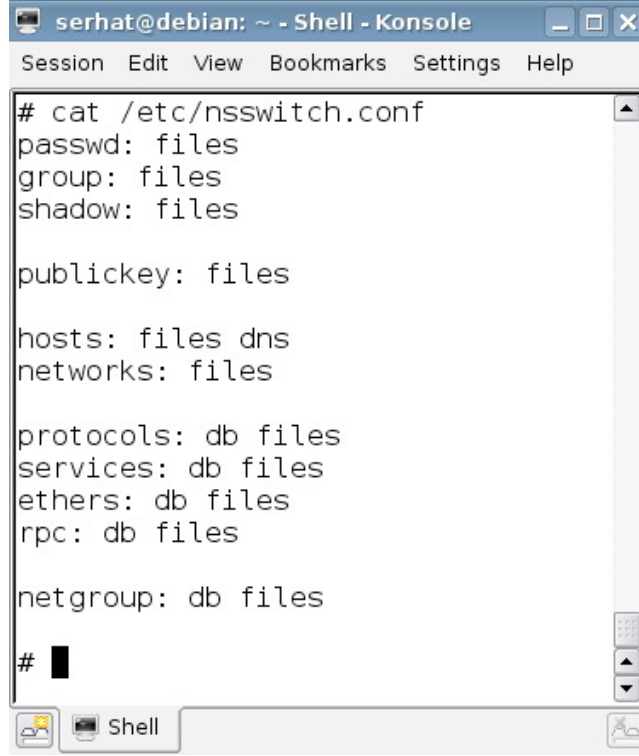
/ # exit

reboot

6.5 Sistem Kendi Başına Yüklendikten Sonra Yapılacak Ayarlamalar

6.5.1 Sistem Kullanıcıları ve Gruplar

- "`vi /etc/passwd`" komutuyla yaratılan dosyaya "`root::0:0:root:/root:/bin/sh`" satırı girilip kaydedilir
- "`vi /etc/group`" komutu ile root kullanıcısının grubu "`root:x:0:`" satırı ile oluşturulur
- Kullanıcıları sistemdeki uygulamalarla ilişkilendiren "`/lib/libnss_files.so.2`" ana makineden sisteme birebir kopyalanır
- "`vi /etc/nsswitch.conf`" komutuyla aşağıdaki ayar dosyası oluşturulur:



```
serhat@debian: ~ - Shell - Konsolle
Session Edit View Bookmarks Settings Help
# cat /etc/nsswitch.conf
passwd: files
group: files
shadow: files

publickey: files

hosts: files dns
networks: files

protocols: db files
services: db files
ethers: db files
rpc: db files

netgroup: db files

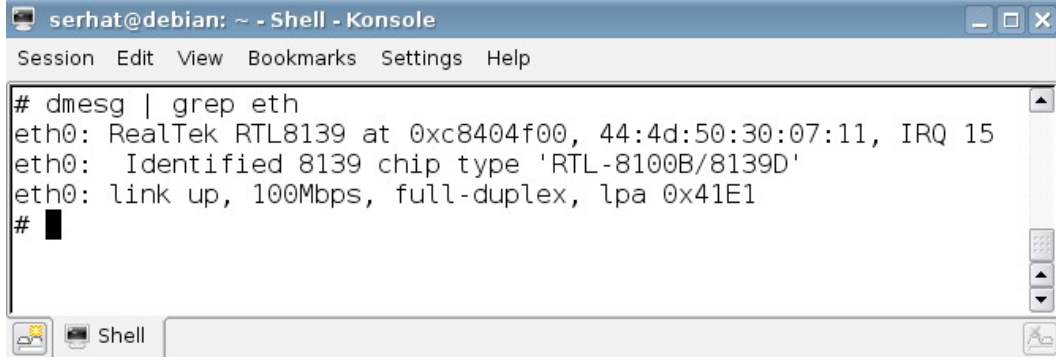
#
```

Şekil 6-6: /etc/nsswitch.conf dosyası

- "serhat" kullanıcıını users grubuna eklemek için "/etc/group" dosyasına "echo "users:x:100:" >> /etc/group" şeklinde bir kayıt eklenir
- "/etc/passwd" dosyasına "serhat::1000:100:Serhat Tatlidil:/home/serhat:/bin/sh" şeklinde bir kayıt girilerek serhat kullanıcıına üçüncü konsoldan sisteme giriş yapma yetkisi verilir.

6.5.2 Sistemin Ağ'a Bağlanması

SBC üzerindeki Realtek 8139 ethernet arayüzü çekirdek derleme esnasında aktif hale getirilir. Ethernet yapılandırması "dmesg | grep eth" komutuyla kontrol edilir.



```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
# dmesg | grep eth
eth0: RealTek RTL8139 at 0xc8404f00, 44:4d:50:30:07:11, IRQ 15
eth0: Identified 8139 chip type 'RTL-8100B/8139D'
eth0: link up, 100Mbps, full-duplex, lpa 0x41E1
#
```

Şekil 6-7: Ethernet yapılandırma kontrolü

Ağ arayüzüne ilişkin IP ve yönlendirme ayarları `"/etc/init.d/rcS"` dosyasına eklenir:

```
# Network Ayarlari
IP=192.168.1.160
NETMASK=255.255.255.0
GATEWAY=192.168.1.1

# Loopback interface
echo -n "Loopback aktiflestiriliyor..."
ifconfig lo 127.0.0.1 netmask 255.0.0.0 up
echo " Tamam."

echo -n "Ag ayarlari yapiliyor..."
ifconfig eth0 $IP netmask=$NETMASK up
route add default gw $GW
echo " Tamam."
```

`"/etc/init.d/rcS"` dosyası çalıştırılarak ağ arayüzü ile ilgili satırlara ait bilgiler, `"ifconfig eth0"` komutu ile de değerlerin düzgün algılanıp algılanmadığı kontrol edilir. Son olarak, ana makineye ping atarak arada bağlantı olup olmadığına bakılır:

```
/ # /etc/init.d/rcS
mount: Mounting proc on /proc failed: Device or resource busy
mount: Mounting devpts on /dev/pts failed: Device or resource busy
Loopback aktiflestiriliyor... Tamam.
```

Ag ayarlari yapiliyor...route: SIOC[ADD|DEL]RT: File exists

Tamam.

Inetd servisi baslatiliyor... Tamam.

Telnet servisi baslatiliyor...telnetd: bind: Address already in use

Tamam.

Moduller yukleniyor... Tamam.

/ # ifconfig eth0

eth0 Link encap:Ethernet HWaddr 44:4D:50:30:07:11

inet addr:192.168.1.160 Bcast:192.168.1.255 Mask:255.255.255.0

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:104 errors:0 dropped:0 overruns:0 frame:0

TX packets:55 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:8593 (8.3 KiB) TX bytes:3863 (3.7 KiB)

Interrupt:15 Base address:0xf00

/ # ping 192.168.1.100

PING 192.168.1.100 (192.168.1.100): 56 data bytes

64 bytes from 192.168.1.100: icmp_seq=0 ttl=64 time=0.5 ms

64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=0.3 ms

64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=0.3 ms

6.5.3 Sisteme Bir Internet Daemon (inetd) Eklenmesi

Sisteme ethernet üzerinden dosya göndermek ve sisteme bağlanmak için ftp ile telnet servislerinden yararlanır. Ana makinedeki "*in.ftpd (ftp daemon)*" ve "*telnetd (telnet daemon)*" uygulamaları sisteme eklenir. Ana makineden "*/usr/sbin/inetd*" ve "*/usr/sbin/tcpd*" uygulamaları ile her iki uygulamanın ihtiyaç duyduğu kütüphane dosyaları "*ldd*" komutuyla öğrenilerek sisteme aktarım yapılır:

tuba:/home/serhat# ldd /usr/sbin/tcpd

libwrap.so.0 => /lib/libwrap.so.0 (0xb7fd3000)

libc.so.6 => /lib/tls/libc.so.6 (0xb7e9d000)

libnsl.so.1 => /lib/tls/libnsl.so.1 (0xb7e89000)

`/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0xb7fea000)`

inetd dinleyeceği portları ve çalıştıracağı uygulamaları `"/etc/inetd.conf"` ayar dosyasından okur:

Örnek - /etc/inetd.conf dosya örneği

```
etc # cat inetd.conf
ftp  stream tcp  nowait root  /usr/sbin/tcpd  /usr/sbin/in.ftpd
```

`"/etc/services"` dosyası aşağıdaki kayıtları içerecek şekilde oluşturulur:

Örnek - /etc/services dosya örneği

```
/etc # cat services
ftp      21/tcp
ftp-data 20/tcp
telnet   23/tcp
```

`"/etc/init.d/rcS"` dosyası içerisine `"inetd"` şeklinde bir kayıt girilerek sistemin her açılışında uygulamanın çalıştırılması sağlanır.

6.5.4 Sisteme Ftp Sunucu Eklenmesi

Sistemde `"bsd-ftpd"` programını ftp sunucu olarak kullanmak için kaynak dosyası İnternette indirilir ve açılır. Makefile dosyası bir editörle açılarak aşağıdaki gibi değiştirilir [44]:

Orijinal Satırlar

```
#CFLAGS = $(OPT_CFLAGS) $(EXTRA_CFLAGS) -DTCPPWRAPPERS -DUSE_PAM -
          DAUTO_UNCOMPRESS -DINTERNAL_LS
#LIBS = $(EXTRA_LIBS) -lutil -lwrap -lnsl -lcrypt -lpam -ldl
```

Yeni Eklenen Satırlar

```
CFLAGS = $(OPT_CFLAGS) $(EXTRA_CFLAGS) -DTCPPWRAPPERS -DUSE_SHADOW
          -DAUTO_UNCOMPRESS -DINTERNAL_LS
```

LIBS = \$(EXTRA_LIBS) -lutil -lwrap -lnsl -lcrypt

"make" ve "strip ftpd" komutları ile ftpd uygulaması elde edilerek sisteme "/usr/sbin/in.ftpd" adıyla taşınır. Ana makineden sistemdeki ftp sunucusu aşağıdaki şekilde bağlanır:

Örnek - "YuzTakip" dosyasının sisteme "/bin/YuzTakip" dosyası olacak şekilde ftp üzerinden gönderilmesi

serhat@tuba:~/home/serhat\$ ncftp -u root 192.168.1.160
NcFTP 3.1.8 (Jul 27, 2004) by Mike Gleason (<http://www.NcFTP.com/contact/>).

Connecting to 192.168.1.160...
(none) FTP server (Version 6.4/OpenBSD/Linux-ftpd-0.17) ready.

Logging in...
Password requested by 192.168.1.160 for user "root".

Password required for root.

Password: *****

User root logged in.
Logged in to 192.168.1.160.

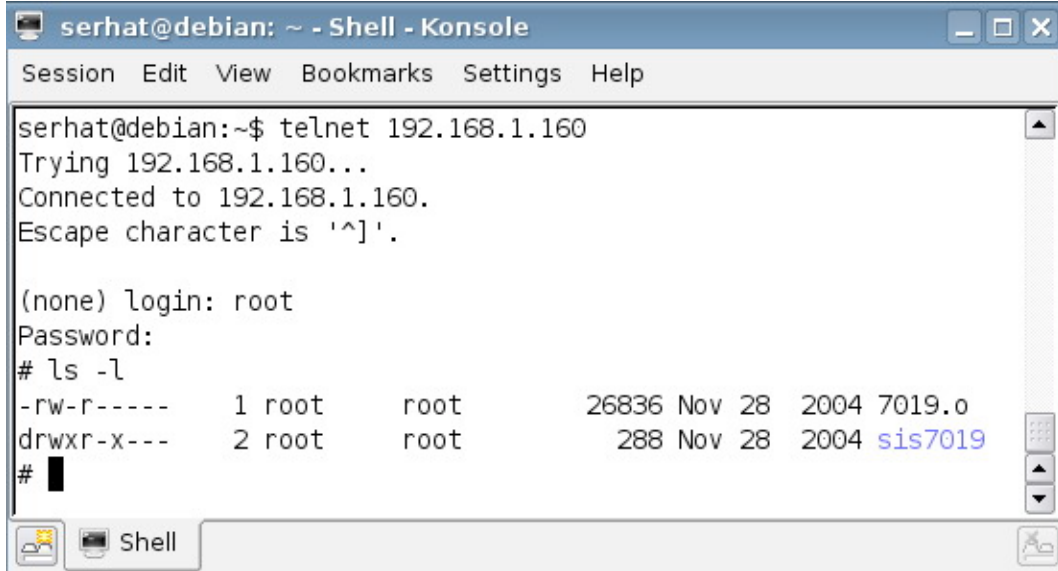
ncftp /root > cd /bin
ncftp /bin > put YuzTakip
ftp> bye

221 Goodbye.

serhat@tuba:~/home/serhat\$

6.5.5 Sisteme Telnet Sunucu Eklenmesi

"/usr/sbin/telnetd" ve "/usr/lib/telnetlogin" uygulamaları ile bağımlı kütüphane dosyaları sisteme aktarılır. "chmod 755 /usr/sbin/telnetd /usr/lib/telnetlogin" komutuyla erişim hakları düzenlenerek aşağıdaki şekilde telnet sunucusu test edilir:



```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

serhat@debian:~$ telnet 192.168.1.160
Trying 192.168.1.160...
Connected to 192.168.1.160.
Escape character is '^]'.

(none) login: root
Password:
# ls -l
-rw-r----- 1 root  root  26836 Nov 28 2004 7019.o
drwxr-x---  2 root  root   288 Nov 28 2004 sis7019
# █
```

Şekil 6-8: Telnet ile ana makineden sisteme bağlanma

7 YÜZ TESPİT VE TAKİP UYGULAMASI

7.1 Uygulamanın Derlenmesi ve Çalıştırılması

“5.2.5 OpenCV Kütüphanesi Kurulumu” bölümünde OpenCV kütüphanesi ana makinede “/opt/opencv” dizinine kurulmuştu. Belirtilen dizin birebir kopyalama yöntemiyle sisteme aktarıldıktan sonra kütüphane kurulumu esnasında ana makinede verilen tüm komutlar sistemde de girilir.

Ana makinede proje uygulamasına ait kaynak kodunu içeren “/var/www/chroot/serhat/main.c” dosyası aşağıdaki şekilde derlenir:

```
chroot /var/www/chroot
cd serhat
gcc `pkg-config --flags opencv` `pkg-config --libs opencv` -o YuzTakip main.c
```

Derlemenin ardından başka bir konsol penceresinde

“cp /var/www/chroot/serhat/YuzTakip /var/www” komutu verilerek, uygulama ana makinenin “/var/www” dizinine kopyalanır. Böylece “apache sunucusu” uygulamayı ağdan kopyalanmak üzere yayımlamaya başlar.

Telnet ile sisteme bağlanılır ve sırasıyla aşağıdaki komutlar verilir:

```
# cd /bin
# wget http://192.198.1.100/YuzTakip
# chmod 755 YuzTakip
# export DISPLAY=192.168.1.100:0.0
# ./YuzTakip
```

Yukarıdaki komutlarla önce “/bin” dizinine geçilir. Ana makinenin IP’si kullanılarak apache sunucusu tarafından yayımlanan “YuzTakip” dosyası BusyBox içinde bulunan “wget” komutuyla sisteme kopyalanır ve çalışabilir dosya haline getirilir. Program çıktısının ethernet üzerinden ana makineye gönderilmesi için “export” komutu ana makinenin IP’sini içerecek şekilde

verildikten sonra program çalıştırılır. Bir yüz tespit edildikten sonra ana makinede otomatik olarak bir pencere açılır ve sistem tarafından takip edilen yüze ait video görüntüsü alınır.

Bu kısımda anlatılan komutları ve örnek bir yüz tespit ve takip işlemini içeren ekran görüntüsüne ait bir kayıt dokümanla birlikte sağlanan CD içinde "*uygulama.ogv*" ismiyle yer almaktadır.

7.2 Uygulama Performansı

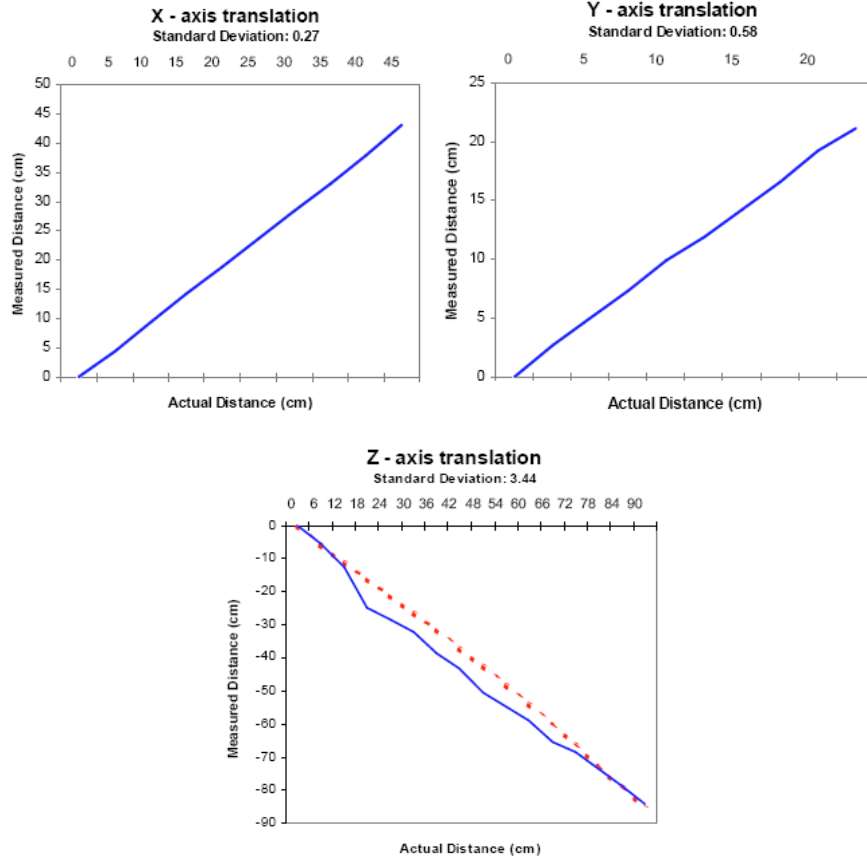
Viola ve Jones yüz tespit çalışmasında kullanılan yüz detektörü, 4916 adet yüz barındıran ve 9544 adet yüz içermeyen resimden elde edilen 6000 öznitelik ve 38 aşamadan oluşmaktadır [2]. Tez çalışmasında ise 24 adım ve 2963 öznitelikten oluşan bir dosya kullanılarak %98'lik bir oranla cepheden çekilen tek bir yüz tespit edilmektedir [72, 80]. Uygulamanın her adımı yüz olmayan kısımları %50 gibi yüksek bir oranla yüz olarak tespit etse de, 24 adım sonunda bu tespit oranı %0.0001'e düşmektedir. Uygulamada alınan resimde herhangi bir yüz barındırmayan kısımlar ilk iki adım sonucunda % 70 oranında elenmektedir [72]. Viola ve Jones [2] tarafından belirlenen özniteliklere eklenen genişletilmiş özniteliklerin kullanımı ile yanlış tespit hatası %10 düşürülmüştür [80].

Yüz takip işlemi sırasında boyun bölgesi de yüzü çevreleyen elipsin içinde yer almaktadır (Şekil 7-1a ve b). Uygulama yüzün önüne gelen el çeldiricisinden minimum seviyede etkilenirken, bir başka yüzün ortamda bulunmasından ise etkilenmemektedir (Şekil 7-1c ve d). CAMSHIFT algoritmasının başlangıcında alınan histogram değerinin kırmızı / turuncu renkli kıyafetlerden etkilendiği ve yüz yerine kırmızı / turuncu nesneyi takip ettiği görülmüştür (Şekil 7-1e ve n). Takip edilen yüz kameradan yakınlaştığında veya uzaklaştığında (Z ekseninde bir değişim olduğunda) takip işlemine devam edilmektedir (Şekil 7-1f ve g). Sistem ortamdaki ışıktan etkilenmektedir (Şekil 7-1h, i ve j). Sistem profil görüntülerini takip edebilmektedir (Şekil 7-1k ve l). Yüzün başka nesnelere arkasında kaldığı durumlarda takip işlemine devam edilmektedir (Şekil 7-1m).



Şekil 7-1: Örnek resimler

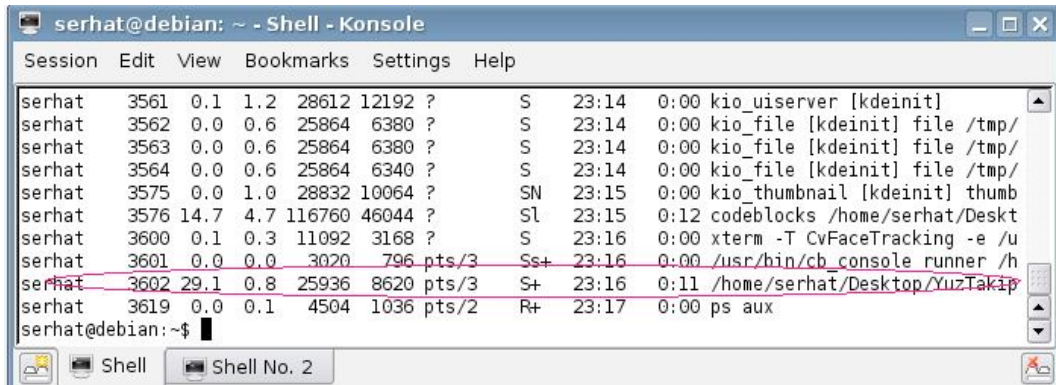
Yüz takip performansına ait istatistiki bilgiler Şekil 7-2’de verilmiştir. Z eksenindeki değişimler, Polhemus manyetik sensörüne ait performans değerleri ile karşılaştırılmıştır [63].



Şekil 7-2: Yüz takip performansı

Uygulama yazılımı kameradan alınacak resimlerin çözünürlüğü 320x240 olacak şekilde ayarlanmıştır.

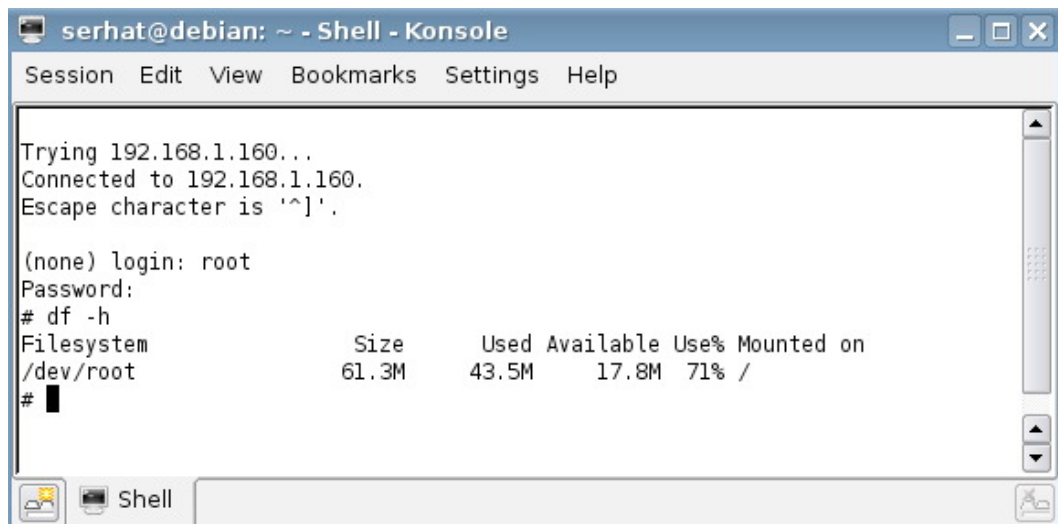
Hazırlanan uygulama dosyası ana makinede %29.1 CPU kaynağı ve 26 MB hafıza kullanmaktadır (Şekil 7-3).



Şekil 7-3: Sistem CPU ve hafıza kullanımı

8 SONUÇLAR ve ÖNERİLER

Çalışmanın ilk kısmında ana makine ile gömülü sistemden meydana gelen bir geliştirme ortamı oluşturulmuştur. Ana makineye Debian GNU/Linux işletim sisteminin kurulumunun ardından, bilgisayardan gömülü sistem görevi yapan SBC'ye aktarımlar yapılarak karta özel bir Linux işletim sistemi hazırlanmıştır. Hazırlanan işletim sistemi ve özel uygulama için toplam 43.5 MB disk alanı kullanılmıştır (Şekil 8-1).



```
serhat@debian: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Trying 192.168.1.160...
Connected to 192.168.1.160.
Escape character is '^]'.

(none) login: root
Password:
# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/root        61.3M    43.5M    17.8M   71% /
#
```

Şekil 8-1: Sistem disk kullanım miktarı

Çalışmanın ikinci kısmında ise Viola ve Jones [2] tarafından geliştirilen nesne tespit algoritmasının bir gerçekleştirilmesi yapılarak Internet kamerasından gerçek zamanlı alınan resimlerden otomatik olarak bir yüz tespit edilmekte ve tespit edilen yüzün CAMSHIFT algoritması ile takip edildiği bir uygulama SBC'de çalıştırılmak üzere C programlama dilinde kodlanmıştır. Oluşturulan sistemin ön cepheden alınan resimlerdeki yüz tespit oranı %98'dir. Yüz takip aşaması ise aşağıda belirtilen birkaç konu haricinde istendiği gibi çalışmaktadır:

- Kullanılan CAMSHIFT algoritmasının çalışma mantığı nedeniyle boyun bölgesi de yüz bölgesi içine dahil edilmektedir.
- CAMSHIFT algoritması diğer takip yöntemleri gibi ortamdaki ışıktan etkilenir bir yapıdadır. Bu yüzden, uygulama minimum ve maksimum

HSV verilerine ait eşik değerleri sistemin yerleştirileceği her ortam için değiştirilecek şekilde konfigüre edilmiştir.

- Sistem ortamdaki ten rengine yakın tonlu nesnelere etkilenebilir.

Çalışma sonucunda 43.5 MB disk alanı ve basit bir Internet kamerası kullanılarak yüz tespit ve takip işlemlerinin otomatik ve gerçek zamanlı olarak gerçekleştirildiği bir gömülü sistem başarı ile oluşturulmuştur.

İlerideki çalışmalarda önerilen sistem ve uygulama pek çok yönden geliştirilebilir:

- Sistemde birden fazla yüzün tespit ve takip edilmesi için iyileştirmeler yapılabilir
- İhtiyaca göre sisteme eklenecek bir ekran yardımı ile sistem tespit ve takip görüntülerini direk ekrana yansıtabilir
- Kaliteli bir kamera ve daha yüksek işlemci frekansına sahip bir kart kullanımı sayesinde uygulama güvenlik amaçlı görevlerde kullanılacak şekilde düzenlenebilir
- Yüz tespitinin ardından yüz tanınması yapacak şekilde uygulama genişletilebilir
- Yüz takip yönteminin ışıktan etkilenmesi nedeniyle ortamdaki ışığı algılayıp HSV değerlerinin otomatik olarak ayarlanması sağlanabilir.

KAYNAKLAR

- [1] Bradski G. R., *Computer Vision Face Tracking For Use in a Perceptual User Interface*, 1998.
<http://download.intel.com/technology/itj/q21998/pdf/camshift.pdf>
- [2] Viola P. ve Jones M.J., “Robust Real-Time Face Detection”, *International Journal of Computer Vision*, **57 (2)**, 137 – 154, Mayıs 2004.
- [3] Anonim, *FPGA - Field programmable gata arrays*, 2009.
<http://tr.wikipedia.org/wiki/FPGA>
- [4] Maurer, T. ve von der Malsburg, C., *Tracking and learning graphs and pose on image sequences of faces*, *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 1996.
- [5] Shen, S. ve Liu, Y., *Efficient multiple faces tracking based on Relevance Vector Machine and Boosting learning*, *Journal of Visual Communication and Image Representation archive*, 382-391, 2008.
- [6] Freund, Y, ve Schapire, R.E., *A decision-theoretic generalization of on-line learning and an application to boosting*, *Journal of Computer and System Sciences*, 119–139, 1997.
- [7] Podolsky A. ve Frolov V., *Face tracking, In the context of Saya, the android secretary*, 2008.
<http://www.cs.bgu.ac.il/~orlovm/teaching/saya/reports/saya-tracking-report.pdf>
- [8] Sun Y., Xie L., Wang Z. ve An Y., “An Embedded System of Face Recognition Based on ARM and HMM”, *Lecture Notes in Computer Science Series Entertainment Computing – ICEC 2007*, Springer Berlin / Heidelberg, 389-394, 2007.
- [9] Patil R., Rybski P.E., Kanade T. ve Veloso M.M., “People Detection and Tracking in High Resolution Panoramic Video Mosaic”, *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

- [10] Schneiderman H., “*Feature-centric evaluation for cascaded object detection*,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.
- [11] Comaniciu D., Ramesh V. ve Meer P., “*Real-time tracking of non-rigid objects using mean shift*,” *Proc. IEEE Conf. On Computer Vision and Pattern Recognition*, 142–151, 2000.
- [12] Li Q. ve Ji H., “*Face detection in complex background based on Gaussian models and neural networks*”, *Signal Processing, 2006 8th International Conference*, 2006.
- [13] Anonim, *Code::Blocks IDE dağıtım sayfası*, 2009.
<http://sourceforge.net/projects/codeblocks/>
- [14] Chu S., Yeh M. ve Cheng K., “*A Real-time, Embedded Face Annotation System*” , *International Multimedia Conference, Proceeding of the 16th ACM international conference on Multimedia*, Vancouver, British Columbia, Canada, 989-990, 2008.
- [15] Beekmans G., *Linux From Scratch Book V. 6.5*, 2009.
<http://www.linuxfromscratch.org/lfs/downloads/stable/LFS-BOOK-6.5.pdf>
- [16] Yaghmour K., Masters J., Ben-Yossef G. ve Gerum P., *Building Embedded Linux Systems Second Edition*, O'Reilly Media, 2008.
- [17] Anonim, *Icop Technology firmasının ana sayfası*, 2009.
<http://www.icoptech.com>
- [18] Anonim, *Kullanılan SBC'ye ait Icop Technology ürün sayfası*, 2009.
<http://www.icoptech.com/pddetail.aspx?id=11&pid=4>
- [19] Anonim, *Linux çekirdek dağıtım adresi*, 2009. <http://www.kernel.org/>.
- [20] Anonim, *What Does Free Mean? or What do you mean by Free Software?*, 2009. <http://www.debian.org/intro/free>
- [21] Anonim, *Tez sırasında kullanılan Debian GNU/Linux işletim sistemi dağıtımının ana sayfası*, 2009. www.debian.org
- [22] Anonim, *GNU Ana Sayfası*, 2009. <http://www.gnu.org/>
- [23] Anonim, *Kullanılan Debian GNU/Linux işletim sistemi kurulum görüntü dosyası*, 2009. <http://cdimage.debian.org/debian-cd/5.0.3/i386/iso-dvd/debian-503-i386-DVD-1.iso>

- [24] Anonim, *Resmi Debian kurulum rehberi*, 2009.
<http://www.debian.org/releases/stable/installmanual>
- [25] McKinley K., *Creating custom kernels with Debian's kernel-package system*, 2003. <http://newbiedoc.sourceforge.net/system/kernel-pkg.html>
- [26] Anonim, *Linux USB Overview*, 2009.
<http://www.qbik.ch/usb/devices/controller.php>
- [27] Rubini A., *Usb Device Drivers*, 2009.
<http://www.linux.it/~rubini/docs/usb/usb.html>
- [28] Reuvers, M., *Face Detection On The INCA*, Yüksek Lisans Tezi, University of Amsterdam, Informatics Institute, Faculty of Science, Amsterdam Hollanda, 2004.
- [29] Hards B., *The Linux USB sub-system*, 2009.
<http://linuxusbguide.sourceforge.net/USB-guide-1.0.9/book1.html>
- [30] Anonim, *Kullanılan kamera sürücüsünün desteklediği tüm Internet kameralar*, 2009.
http://www.rastageeks.org/ov51x-jpeg/index.php/Working_Webcams
- [31] Shane H., *The Webcam HOWTO*, 2005. <http://tldp.org/HOWTO/Webcam-HOWTO>
- [32] Anonim, *Debian Linux Web Cam Server Configuration*, 2009.
<http://www.aboutdebian.com/webcam.htm>
- [33] Anonim, *ov51x-jpeg sürücü kurulumu*, 2009.
<http://www.rastageeks.org/ov51x-jpeg/index.php/Ov51xJpegHackedInstall>
- [34] Anonim, *GNU derleyicisi olan GCC'nin ana sayfası*, 2009.
<http://gcc.gnu.org/>
- [35] Anonim, *Gömülü Sistemlerde kullanılan mikroişlemciler için bir derleyici*, 2009. <http://sdcc.sourceforge.net/>
- [36] Anonim, *ov51x-jpeg sürücüsü*, 2009.
<http://www.rastageeks.org/downloads/ov51x-jpeg/ov51x-jpeg-1.5.9.tar.gz>
- [37] Anonim, *İçinde Internet kamerası kayıt ve gösterim özellikleri de taşıyan xawtv uygulamasının ana sayfası*, 2009. <http://linux.bytesex.org/xawtv>
- [38] Anonim, *gnomemeeting isimli Internet kamerası uygulamasının ana sayfası*, 2009. www.gnomemeeting.org

- [39] Anonim, *Making and Managing Linux Filesystems*, 2009.
http://www.comptechdoc.org/os/linux/usersguide/linux_ugmakingfs.html
- [40] Nguyen B., *Linux Filesystem Hierarchy*, 2004.
<http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/Linux-Filesystem-Hierarchy.html>
- [41] Anonim, *Busybox uygulamasının ana sayfası*, 2009. www.busybox.net
- [42] Anonim, *Busybox uygulamasının kurulum dosyalarının yer aldığı sayfa*, 2009. <http://www.busybox.net/downloads/>
- [43] Birchfield, S., *Elliptical head tracking using intensity gradients and color histograms*, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santra Barbara, ABD*, 232–237, 1998.
- [44] Madore D., *ftpd-BSD uygulamasının sayfası*, 2009.
http://www.madore.org/~david/programs/#prog_ftpd-BSD
- [45] Fieguth P. ve Terzopoulos D., “Color based tracking of heads and other mobile objects at video frame rates,” *In Proc. Of IEEE CVPR*, 21-27, 1997.
- [46] Wren C., Azarbayejani A., Darrell T., Pentland A., “Pfinder: Real-Time Tracking of the Human Body” *SPIE* **2615**, 1995.
- [47] Hunke M. ve Waibel A., “Face locating and tracking for human-computer interaction” *Proc. Of the 28th Asilomar Conf. On Signals, Sys. and Comp.*, 1277-1281, 1994.
- [48] Sobottka K. ve Pitas I., “Segmentation and tracking of faces in color images” *Proc. Of the Second Intl. Conf. On Auto. Face and Gesture Recognition*, 236-241, 1996.
- [49] Swain M. ve Ballard D., “Color indexing” *Intl. J. of Computer Vision*, **7(1)**, 11-32, 1991.
- [50] Fukunaga K., “*Introduction to Statistical Pattern Recognition*” Academic Press, Boston, 1990.
- [51] Comaniciu D. ve Meer P., “Robust Analysis of Feature Spaces: Color Image Segmentation” *Computer Vision and Pattern Recognition - CVPR '97*, 750-755, 1997.
- [52] Smith A.R., “Color Gamut Transform Pairs” *SIGGRAPH* **78**, 12-19, 1978.

- [53] Foley J.D., van Dam A., Feiner S.K. ve Hughes J.F., “*Computer graphics principles and practice*” Addison-Wesley, 590-591, 1995.
- [54] Williams, O., Blake, A. ve Cipolla, R., *Sparse bayesian learning for efficient visual tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1292–1304, 2005.
- [55] Anonim, *MCIMX25LPDK Linux Product Development Kit for i.MX25 isimli ürünün tanıtım sayfası*, 2009.
<http://www.denx.de/en/Hardware/FSLiMX25LPDK>
- [56] Anonim, *MCIMX25LPDK Linux Product Development Kit for i.MX25 isimli ürünün tanıtım sayfası*, 2009.
<http://www.denx.de/wiki/view/DULG/Manual>
- [57] Anonim, *Digital Signal Processor - DSP*, 2009.
http://en.wikipedia.org/wiki/Digital_signal_processor
- [58] Lowe K., *Kernel Rebuild Guide*, 2004.
<http://www.digitalhermit.com/linux/Kernel-Build-HOWTO.html>
- [59] Anonim, *Open Computer Vision Library - OpenCV ana sayfası*, 2009.
<http://sourceforge.net/projects/opencvlibrary/>
- [60] Wirzenius L., Oja J. ve Stafford S., *The Linux System Administrator's Guide*, 2009. http://www.faqs.org/docs/linux_admin/x1298.html
- [61] Matthews, I. ve Baker, S., *Active appearance models revisited, International Journal of Computer Vision*, 135–164, 2004.
- [62] Rybski P.E. ve Veloso M.M., *Robust Real-Time Human Activity Recognition from Tracked Face Displacements, In Proceedings of the 12th Portuguese Conference on Artificial Intelligence, EPIA 2005*, 87–98, Portekiz, 2005.
- [63] Anonim, *Polhemus ana sayfası*, 2009. <http://www.polhemus.com/>
- [64] Anonim, *OKI firması ana sayfası*, 2009. <http://www.oki.com/en/>
- [65] Bigdeli, A., Sim, C., Biglari-Abhari, M. ve Lovell, B. C., *Face Detection on Embedded Systems, Proceedings of the 3rd International Conference on Embedded Software and Systems: Lecture Notes in Computer*, 295-308, 2007.

- [66] McCready, R., *Real-Time Face Detection on a Configurable Hardware Platform*, Yüksek Lisans Tezi, University of Toronto, Graduate Department of Electrical and Computer Engineering, Toronto Kanada, 2000.
- [67] Pun, K. H., Moon, Y.S., Tsang, C. C., Chow, C. T. ve Chan, S. M., *A face recognition embedded system*, *Proceedings of SPIE 5779, Biometric Technology for Human Identification II*, Orlando, ABD, 2005.
- [68] Anonim, *Single-Board Computer*, 2009. http://en.wikipedia.org/wiki/Single-board_computer.
- [69] Su, S. ve Yin, B., *A Robust Face Detection Method*, *Proceedings of the Third International Conference on Image and Graphics (ICIG'04)*, 2004.
- [70] Rowley, H.A., Baluja, S. ve Kanade, T., *Neural Network-Based Face Detection*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.
- [71] Lee Y., Moon, S. ve Kim, Y., "Face and Facial Expression Recognition with an Embedded System for Human-Robot Interaction", *Lecture Notes in Computer Science, Affective Computing and Intelligent Interaction, Springer Berlin / Heidelberg*, 271-278, 2005.
- [72] Bradski, G. ve Kaehler, A., "Viola-Jones Classifier Theory", *Learning OpenCV*, O'Reilly Media, ABD, 508-509, 2008.
- [73] Anonim, *Single Board Computers & Embedded Systems*, 2009. <http://www.embeddedplanet.com/>
- [74] Anonim, *Red Hat işletim sisteminin ana sayfası*, 2009. <http://www.redhat.com/>
- [75] Anonim, *VxWorks işletim sisteminin ana sayfası*, 2009. <http://www.windriver.com/>
- [76] Anonim, *Integrity işletim sisteminin ana sayfası*, 2009. <http://www.ghs.com/>
- [77] Anonim, *Timesys Linux işletim sisteminin ana sayfası*, 2009. <https://linuxlink.timesys.com/3/Home>
- [78] Anonim, *MQX işletim sisteminin ana sayfası*, 2009. <http://www.arc.com/>
- [79] Anonim, *QNX işletim sisteminin ana sayfası*, 2009. <http://www.qnx.com/>
- [80] Lienhart, R., ve Maydt, J., *An Extended Set of Haar-like Features for Rapid Object Detection*. *IEEE ICIP 2002, 1*, 900-903, 2002.

- [81] Yang, M., Kriegman, D.J. ve Ahuja, N., *Detecting Faces in Images: A Survey*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.
- [82] Fisher, J.A., Faraboschi, P. ve Young, C., *Embedded Computing*, Elsevier Inc. ABD, 1-5, 2005.
- [83] Anonim, *Meeting Minimum Hardware Requirements*, 2008.
<http://www.debian.org/releases/stable/i386/ch03s04.html.en>

EK: CD İÇERİĞİ

GNU/Linux standartlarına uygun bir uygulama yayımlamak için belirli bir düzen olması gerekmektedir [22]. CD'deki "*Kaynak Kod*" dizininde bu biçeme göre hazırlanan dosyalar ile proje dosyaları yer almaktadır:

- **Kopyalama ve kullanım hakları için : COPYING dosyası**
- **Kaynak program kodunun yer aldığı : main.c dosyası**
- **Kurulum için : MAKEFILE Dosyası**
- **Kullanıcılara bilgi vermek için: README Dosyası**
- **Kameradan alınan resimlerde bir yüz tespit edildiği takdirde otomatikman resim kaydetmek üzere oluşturulmuş ResimKaydet.c Dosyası**

"*Yüz Tespit ve Takip Uygulamasının Çalıştırılması*" bölümünde sisteme verilen komutlar ve örnek bir yüz tespit ve takip ekran görüntüsü kaydı "*uygulama.ogv*" ismiyle CD'ye konulmuştur.