T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

YÜZ GİZLİLİĞİ İÇİN GÖRSEL KRİPTOGRAFİ

(Yüksek Lisans Tezi)

Tezi Hazırlayan :

Engin AKÇIL

İstanbul, 2014

T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

YÜZ GİZLİLİĞİ İÇİN GÖRSEL KRİPTOGRAFİ

(Yüksek Lisans Tezi)

Tezi Hazırlayan :

Engin AKÇIL

Öğrenci No:

110820019

Danışman:

Yrd. Doç. Dr. Ediz ŞAYKOL

İstanbul, 2014

YEMİN METNİ

Yüksek lisans tezi olarak sunduğum **"Yüz Gizliliği İçin Görsel Kriptografi "** başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım**.** 13/ 06/ 2014

Aday: **Engin AKÇIL**

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi 121082019 no'lu Engin AKGÜL 'in 13/06/2014 tarihinde yapılan tez savunma sınavı[1] sonucunda 50 dakika süreyle sunduğu ve savunduğu tezi hakkında[2] oybirliğiyle, KABUL kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : Bilgisayar Mühendisliği
Programı : Bilgisayar Mühendisliği
Tez Başlığı[3] : YÜZ GİZLİLİĞİ İÇİN GÖRSEL KRİPTOGRAFİ

Tez Sınav Jürisi	Öğretim Üyesi	İmza
Danışman :	Yrd. Doç. Dr. Ediz Saykol	
Üye :	Doç. Dr. Gökhan Silahtaroğlu	
Üye :	Doç. Dr. Kazım Sarı	

---

[1] Jüri üyeleri söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez savunma sınavına alır. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda jüri en geç onbeş gün içinde toplanarak adayı tez savunma sınavına alır. Tez savunma sınav süresi en az 45 dakikadır. Yüksek lisans tez savunma sınavı, tez çalışmasının sunulması ve bunu izleyen soru-yanıt bölümlerinden oluşur ve dinleyiciye açıktır. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-3)

[2] Tez sınavının tamamlanmasından sonra jüri, tez hakkında "kabul", "düzeltme" veya "red" kararı verir. Jüri başkanı, jüri üyelerince imzalanmış sınav tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi başarısız bulunan öğrencinin Enstitü ile ilişiği kesilir. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve yönetmelikte belirtilen usullere uygun olarak tezini aynı jüri önünde yeniden savunur. Bu savunma sınavında da tezi kabul edilmeyen öğrencinin enstitü ile ilişiği kesilir. (Beykent Lisansüstü eğitim ve Öğretim Yönetmeliği-Madde30-4)

[3] İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

Adı ve Soyadı : Engin AKÇIL
Danışmanı : Yrd. Doç. Dr. Ediz ŞAYKOL
Türü ve Tarihi : Yüksek Lisans / Proje, 2014
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Görsel Kriptografi, Özel Hayatın Gizliliği, Yüz Tespiti

ÖZ

YÜZ GİZLİLİĞİ İÇİN GÖRSEL KRİPTOGRAFİ

Günümüzde halka açık alanlarda yer alan güvenlik kameraları suçu önleme veya suçlunun tespiti konularında başarısını kanıtlamış olsa da özel hayatın gizliliği konusunda tartışma yaratmaktadır.

Bu çalışmada; halka açık alanlarda özel hayatın gizliliği ve suçla mücadele arasında bir denge kurmak amacıyla, insan yüzünün monitör görüntüsünden gizlenmesi ve orijinal görüntünün şifrelenerek saklanması ve daha sonra yalnızca yetkili kişiler tarafından açılabilmesi için bir Java uygulaması geliştirilmiştir.

Name and Surname : Engin AKÇIL
Supervizor : Asst. Prof. Ediz ŞAYKOL
Degree and Date : Master, 2014
Major : Computer Engıneerıng
Key Words : Visual Cryptography, Privacy Protection, Face Detection

ABSTRACT

VISUAL CRYPTOGRAPHY FOR FACE PRIVACY

The security cameras in public areas are successful in preventing crimes and detection the offenders, though nowadays, there is a serious concern about the confidentiality of private life.

In this study a Java application has been developed for hiding the human face from the monitor display, storage the orijinal image by encrypting and later being able to be opened by only an authorized person in order to provide a balance between the privacy in public areas and the fight against crime.

İÇİNDEKİLER

ŞEKİLLER LİSTESİ

Sayfa No.

1. Giriş

Bu tezin amacı halka açık alanlarda özel hayatın gizliliği ve suçla mücadele arasında bir denge kurmak amacıyla insan yüzünün monitör görüntüsünden gizlenmesi için çalışan bir uygulama yapmaktır.

Günümüzde kamera ve kayıt teknolojilerinin gelişmesi ve fiyatlarının düşmesi ile birlikte halka açık alanlarda kayıt yapan güvenlik kameralarının sayısı oldukça artmıştır. Bu güvenlik kameralarının gerek suç işlendikten sonra suçlunun tespiti gerekse de suçu önleme konusundaki başarısından dolayı sayıları her geçen gün artmaktadır.

Buna paralel olarak; hem kamu kurumlarının hem ticari işletmelerin ve hatta kişisel mülklerin halka açık alanlarda yaptıkları kayıt işlemlerinin herhangi bir yasal düzenlemeye tabi olmaması, özel hayatın gizliliğinin ihlal edilmesi konusunda kamuoyunda giderek artan bir endişeye sebebiyet vermektedir.

Ayrıca halka açık alanlardan kaydedilen görüntülerin saklanması konusunda da kanuni bir yaptırım bulunmamaktadır.

Anayasımızın 20. maddesine göre herkes, 'özel hayatına saygı gösterilmesini isteme hakkına sahiptir, özel hayatın gizliliğine dokunulamaz ve kendisiyle ilgili kişisel verilerin korunmasını isteme hakkına sahiptir'. Kişisel veriler, ancak kanunda öngörülen hallerde veya kişinin açık rızasıyla işlenebilir. Kişisel verilerin korunmasına ilişkin esas ve usuller kanunla düzenlenir [1].

Türk ceza kanunun 134. maddesine göre kişilerin özel hayatının gizliliğini ihlâl eden kimse, altı aydan iki yıla kadar hapis veya adlî para cezası ile cezalandırılır. Gizliliğin görüntü veya seslerin kayda alınması suretiyle ihlâl edilmesi hâlinde, cezanın alt sınırı bir yıldan az olamaz. Ayrıca aynı kanunun 135. maddesine göre hukuka aykırı olarak kişisel verileri kaydeden kimseye altı aydan üç yıla kadar hapis cezası verilir [2].

İstanbul Valiliği İl İnsan Hakları Kurulu Başkanlığı 08.03.2010 tarihli kararında "kişinin yüz ve çehresinin kişiye sıkı sıkıya bağlı olduğu ve vücut

bütünlüğünün ayrılmasız bir parçası olması ve kişisel bilgi niteliği taşıması " kararı alınmıştır ancak bu karar yalnızca bir tavsiye niteliği taşımaktadır [3].

Öte yandan insan yüzünün kişisel veri olması konusunda kesim hüküm veren bir kanun veya yönetmelik bulunmamaktadır. Bu durum halka açık mekanlardan toplanan görüntülerin ve özelde insan yüzlerinin kaydedilmesi ve paylaşılması konusunda hukuki bir boşluk yaratmaktadır.

Günümüzde Başbakanlık'ta bekleyen Kişisel Verilerin Korunması Kanunu Taslağı'nın 3. Maddesinin Ç bendine göre kişisel veri, "Belirli veya kimliği belirlenebilir gerçek ve tüzel kişilere ilişkin bütün bilgiler" olarak tanımlanmıştır [4]. Bu taslağın yasalaşması durumunda TCK'nın 134. maddesine göre kişilerin halka açık alanlarda kamera kayıtlarını toplamak ve kaydetmek suç niteliği taşıyacaktır.

Ancak halka açık alanlarda kamera kayıtlarının yapılmaması suçlunun tespiti ve suçun önlenmesi hususlarında zorluk yaratacaktır.

Kişisel verileri toplamadan suçlunun tespiti, halka açık alanlarda gerçekleşen kamera kayıtlarının şifrelenerek yapılması veya görüntülerin yerelde hiç kaydedilmeden güvenli bir şekilde ilgili birimlere gönderilmesi ve kanunla izin verilen kurumlarca kaydedilmesi ile sağlanabilir. Ancak gerçek zamanlı görüntülerin güvenlik birimlerince izlenememesi suçun önlenmesi husunda zorluk teşkil edecektir.

Örneğin çalıştığı kurumun dış mekan kamera kayıtlarını izleyen bir güvenlik görevlisi veya kolluk kuvvetinin elinde silah ile yaklaşan kahverengi ceketli bir insan gördüğünde çeşitli şekillerde duruma müdahalesi son derece önemlidir. Bu kamera kaydının halka açık alanda gerçekleşmesi nedeniyle gerçek zamanlı olarak güvenlik görevlisince izlenememesi suça müdahalede gecikmeye neden olabilir. Öte yandan bu şekilde bir saldırı olma ihtimaline karşı kurumun dış mekanından geçen herkesin özel hayatının gizliliğinin güvenlik birimlerince ihlal edilmesine de gerek yoktur. Bu örnekte güvenlik birimlerince önemli olan elinde silahla kimin yaklaştığı değil; elinde silahla kahverengi montlu bir saldırganın yaklaştığıdır.

Bu nedenle kamera görüntülerinin gösterildiği monitörlerden çeşitli tekniklerle insan yüzünün gizlenmesi ve açık kayıdın daha sonra yetkili kişilerce izlenebilmesi,

hem insanların özel hayatının gizliliği hem de suçun önlenmesi veya suçlunun tespiti konularında bir çözüm getirecektir.

Bu çalışmanın kapsamı; mevcut cisim tanımlama algoritmaları ile tespit edilen özelleştirilmiş bir cismin (bu projede insan yüzü), boyama yöntemi ile gizlenerek gösterilmesi ve orjinal video'nun seçilecek olan en uygun algoritmayla şifrelenerek kaydedilmesi ve gerektiğinde yetkilendirilmiş kişi/ kurum tarafından açılabilmesi (deşifre edilmesi) için anahtar üretim yollarının araştırılmasıdır.

2. Görsel Kriptografi

2.1. Kriptografi

Kriptografi, haberleşen iki veya daha fazla tarafın bilgi alışverişini emniyetli olarak yapmasını sağlayan veya gizli bilgiyi istenmeyen kişilerin anlayamayacağı hale getirerek korumayı esas alan, temeli matematiksel yöntemlere dayanan tekniklerin ve uygulamaların bütünüdür [5].

2.2. Kriptografi Teknikleri

2.2.1. Simetrik Anahtar Kriptografisi

Verinin şifrelenmesinde ve şifrelenmiş verinin deşifre edilmesinde aynı anahtarın kullanıldığı ve temelde şifreleme için yapılmış olan adımların tam tersinin deşifreleme esnasında yapıldığı matematiksel işlemlerdir. Çok eski tarihlerden beri kullanılmakta olan bu yöntem günümüzde de modern versiyonlarıyla kullanılmaktadır. Günümüzde kullanılan en popüler yöntemler MD4, MD5, SHA, DES ve AES'tir [6].

2.2.2. Asimetrik Anahtar Kriptografisi

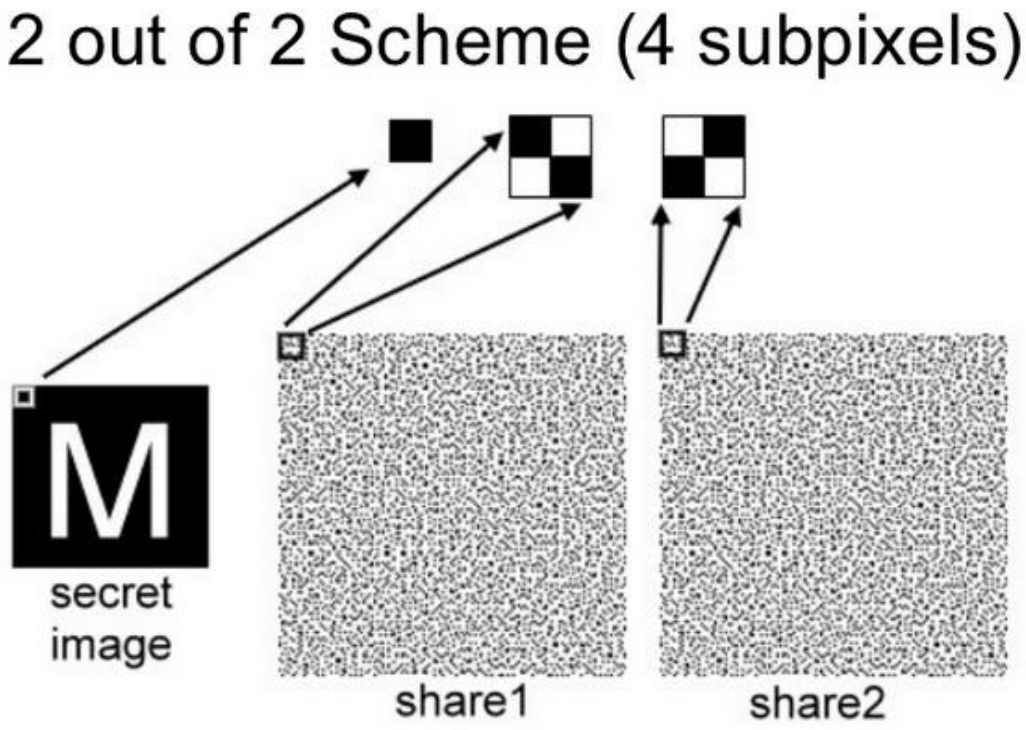
Şifreleme ve deşifreleme işlemlerinde farklı methotların ve farklı anahtarın kullanıldığı şifreleme tekniğidir. Simetrik şifrelemeye göre en önemli avantajı 2'den fazla merkez arasında kullanılması gereken şifreli paylaşımlarda daha güvenli olmasını sağlayan açık anahtar ve gizli anahtar yapısıdır, en önemli dezavantajı ise asimetrik yöntemlerle çalışan uygulamaların simetrik yöntemlere kıyasla daha düşük performans göstermesidir. Bu nedenle asimetrik kriptografi genelde simetrik kriptografi yöntemlerinde anahtar değişimi için kullanılmaktadır [6].

2.3. Görsel Kriptografi Kavramı

Görsel kriptografi, gizlenmek istenen resmin çeşitli tekniklerle gizli parçalara ayrılması ve birden fazla kanaldan iletilmek istenen yere ulaştırılması veya resmin iletilmeden önce insan gözünün karmaşık hesaplar yapmadan ayırt edemeceği bir hale getirilmesi tekniğidir [7].

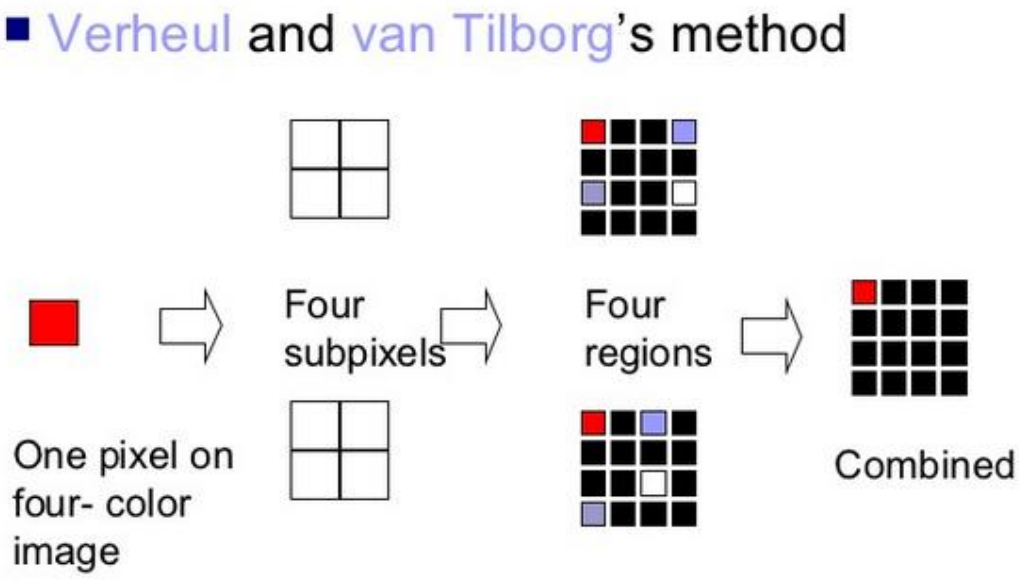
2.4. Yüz Gizliliği Üzerine Literatür Taraması

Görsel kriptografi terimi ilk olarak 1994 yılında Naor ve Shamir tarafından ortaya atılmıştır. Kullandıkları teknik herhangi bir kriptografik hesaplama gerektirmeden son derece güvenli bir şekilde görsel veri paylaşımına olanak sağlamıştır. Kullandıkları yöntem siyah-beyaz bir imajın piksel değerlerini belirli kurallar ile birden fazla resme dağıtmak ve farklı kanallardan iletmek, alıcı tarafından ise bu resimlerin üst üste konması tekniği ile orjinal resme ulaşmaktır.



Şekil.1. İlk Görsel Kriptografi

İlk renkli görsel kriptografi tekniği 1997 yılında Verheul ve Van Tilborg tarafından geliştirilmiştir. Bu teknikde renkli imajdaki her piksel, m adet alt piksellere ayrılarak farklı imajlar içerisinde gizlenerek alıcıya ulaştırılır. [7].



Şekil.2. İlk Renkli Görsel Kriptografi

AES (Advanced Encryption Standart; Gelişmiş Şifreleme Standardı) günümüzde en yaygın olarak kullanılan şifreleme yöntemlerinden birisidir ve görsel kriptografi alanında bu üstünlüğü elinde tutmaktadır. AES ile tanımlanan şifreleme algoritması; hem şifreleme hem de şifreli metni çözmede kullanılan anahtarların birbiriyle ilişkili olduğu, simetrik-anahtarlı bir algoritmadır. AES için şifreleme ve şifre çözme anahtarları aynıdır. AES'in hem yazılım hem de donanım performansı yüksektir. 128-bit girdi bloğu, 128,192 ve 256 bit anahtar uzunluğuna sahiptir. AES'in temel alındığı Rijndael ise 128-256 bit arasında 32'nin katı olan girdi blok uzunluklarını ve 128 bitten uzun anahtar uzunluklarını desteklemektedir. Dolayısıyla, standartlaşma sürecinde anahtar ve girdi blok uzunluklarında kısıtlamaya gidilmiştir.Algoritma belirli sayıda tekrar eden girdi açık metni, çıktı şifreli metne dönüştüren özdeş dönüşüm çevirimlerinden (round) oluşmaktadır. Her çevirim, son çevirim hariç, dört adımdan oluşmaktadır. Şifreli metni çözmek için bu çevirimler

ters sıra ile uygulanır. Çevirimlerin tekrar sayıları 128-bit, 192-bit ve 256-bit anahtar uzunlukları için sırası ile 10, 12 ve 14'tür [8].

Arun Ross ve Asem A. Othman'ın "Visual Cryptography for Face Privacy" adlı çalışmasında, yüz tanımlama projelerinde kullanılan merkezi bir veri tabanında saklanan insan yüzlerinin şifrelenerek saklanması ve gerektiğinde deşifre edilmesi için bir yöntem anlatılmıştır. Bu yöntem esas olarak Naor ve Shamir tarafından bulunan yönteme oldukça benzemektedir. İlk olarak saklanan resimler çeşitli teknikler ile her bir piksel ancak tek bir resimde yer alacak şekilde 2 farklı resme ayrılmıştır ve tek başlarına orjinal resme ulaşmak mümkün olmamaktadır [9].

Karl Martin ve Konstantinos N. "Plataniotis'ın Privacy Protected Surveillance Using Secure Visual Object Coding" adlı çalışmasında daha önce tanımlanmış her hangi bir nesnenin piksel yer değiştirme yöntemi ile şifrelenmesi ve gerektiğinde deşifre edilmesi ile tüm resmin şifrelenmesine göre %5 daha az işlem yapılabildiği anlatılmıştır [10].

3. MODELLEME VE KONFİGÜRASYON

3.1. Sistemin Çalışma Prensibi

Video'dan yakalanan her bir resim, öncelikle şifrelenerek saklanmak üzere orjinal olarak tutulacak ve tespit edilen yüzlerin her birisinin gizlenerek gerçek zamanlı olarak gösterimi için bir kopyası alınacaktır.

Kopyası alınan bu resim öncelikle üzerindeki yüzlerin tespiti için yüz bulma sınıfına gönderilecektir.

Seçilen yüz bulma algoritması yüzün eğik açılarda testpitini yapamadığı için resim seçilecek olan bir açıyla sağa ve sola döndürülerek aynı sınıfa tekrar gönderilebilmesi için ayrı bir sınıf daha yazılmıştır.

Ardından tespit edilen her yüz bölgesi için piksel değerleri sabit bir renk ile boyanarak yüzün tanınmaması sağlanacaktır. Sabit renk ile boyanan yüz monitörden gösterilecek ve hafızada tutulmadan silinecektir.

Orijnal resim ise gerçek zamanlı olarak AES algoritması ile şifrelenmek üzere şifreleme sınıfına gönderilecektir, şifrelemede kullanılacak anahtar yine AES algoritmasıyla random olarak üretilerek resimle birlikte şifreleme sınıfına yollanacaktır.

Deşifre işlemi için ana sınıftan bağımsız olarak deşifre sınıfı çalıştırılacaktır, çalıştırmadan önce str1 değişkenine atanacaktır.

3.2. Kullanılan Teknolojiler

Bu çalışmada bilgisayarla görü işlemleri için opencv kütüphanesi , programlama dili olarak java, işletim sistemi olarak 64 bit MS windows 8.0,geliştirme platformu olarak eclipse, opencv kütüphanesini java ile kullanmak için javacv kütüphanesi kullanılmıştır.

Java ilk olarak Sun Microsystems tarafından 1995 yılında piyasaya sürülen bir programlama dili ve bilgi işlem platformudur .Windows işletim sistemi üzerinde Java ile çalışılabilmesi için öncelikle java sdk'sının yüklenmesi gerekmektedir [11].

Geliştirme ortamı olarak açık kaynak kodlu olan eclipse geliştirme platformu kullanılmıştır, işletim sistemine uygun olan versiyon ücretsiz olarak indirilebilir [12].

OpenCv hem akademik hem ticari olarak kullanılabilecek açık kaynak kodlu bir bilgisayarlı görü kütüphanesidir; C, C++, Python ve Java dilleri için geliştirme arayüzleri vardır ve Windows, Linux, Mac OS, iOS and Android platformlarını destekler [13].

JavaCv kütüphanesi Java sanal makinesinin opencv kütüphanesine ulaşmasını sağlayan bir arabirimdir.

Bu çalışmada OpenCv 2.4.4 ve JavaCv 2.4.4 sürümleri kullanılmıştır.

Yüz tanımlamak için haarcascade kullanılmıştır. Haarcascadeintel tarafından geliştirilen cisim tanımla kütüphanesidir. Yüz bulma için Rainer Lienhart tarafından geliştirilen haarcascade_frontalface_default.xml dosyası referans olarak gösterilmiştir [14].

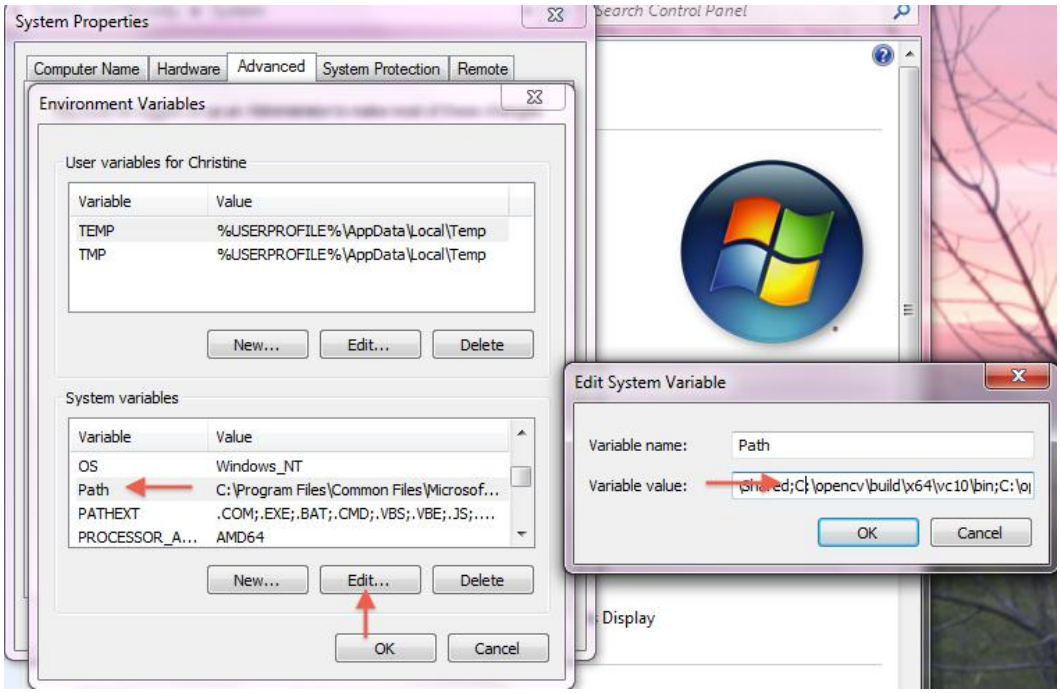
3.3. Konfigürasyon

Öncelikle OpenCV ile çalışmak için Visiual C++ runtime Componenti'nin kurulması gerekmektedir. Microsoft.com'dan işletim sistemine uygun olanı indirilmeli ve kurulmalıdır.

OpenCv ve JavaCv kütüphanelerinin aynı sürümleri indirilmeli ve indirme esnasında Windows ile uyumluluğu kontol edilmelidir, 32 bit ve 64 bit Windows işletim sistemi için ayrı sürümleri bulunmaktadır. İndirilen OpenCv.rar dosyası C:\opencv klasörü yaratarak bu klasörün altına ekstrakte edilmelidir.

Eclipse'in opencv kütüphanesine ulaşabilmesi için windows üzerinde son bir konfigürasyon daha yapmak gerekmektedir;

Bilgisayarım'a sağ tıklayarak açılan ekranda "Properties" seçeneğine tıklanmalı, Ardından sırasıyla "Advanced system settings" sekmesi ,"Advanced"tabından "Environment Variables.." seçeneğine tıklanmalıdır. "System variables" panelinden "Path" seçeneği seçilmeli ve edit butonuna tıklanmalıdır.



Şekil.3. Windows Konfigürasyon

Variable value alanının sonuna 32 bit işletim sistemi için C:\opencv\build\x86\vc10\bin;C:\opencv\build\common\tbb\ia32\vc10 , 64 bit işletim sistemi için C:\opencv\build\x64\vc10\bin;C:\opencv\build\common\tbb\intel64\vc10 eklenip tamam butonuna tıklanmalıdır.

eclipse.exe Dosyasını çalıştırarak açılan uygumada tez isminde bir "project" oluşturulmuş ve bu projenin altında oluşturulan "facePrivacy" "package"inde Main.java, FindFace.java, SkewGrayImage.java, TransposeImage.java, PaintFace.java, Encrypter.java ve Decrypter.java sınıfları oluşturulmuştur.

Son olarak indirilen JavaCv.rar herhangi bir klasöre extract edilerek bu adres oluşturulan tez projesine eklenmelidir. Bununn için Tez projesine sağ tıklayarak açılan ekranda Project | Properties | Java Build Path | Librariesekranından "Add External JARs..." seçeneği seçilerek ekstrakte edilen dosyalar seçilmelidir.

4. GÖRSEL KRİPTOGRAFİ UGULAMASI

4.1. Ana sınıf

Uygulamanın çalışmasının sağlayan void main sınıfını içeren ve diğer sınıfların çalışmasını kontrol eden sınıftır, öncelikle web kamerasından görüntü alınm alınmasını sağlayan openCv kütüphanesinin CvCapture sınıfına ait olan cvCreateCameraCapture methodu kullanılmıştır.

Alınan görüntülerin işlendikten sonra gizlenen yüzü içeren vidoenun monitör'den gösterimi için javaCv kütüphanesinden CanvasFrame sınıfı kullanılmıştır.

cvCreateCameraCapture Methoduyla yakalanan resim öncelikle olası yüzlerin tespiti içn yazılan FindFace sınıfına gönderilmektedir.

Find Face sınıfından kullanılan haarcascade implimantasyonu insan yüzlerini eğik açılarda yakalayamadığı için orjinal resim ve tranzpozesi resim döndürme sınıfına yollanarak 30 derece ile eğilerek birer kez daha yüz tespiti classına gönderilmektedir.

Tespit edilen yüzlerin gizlenmesi için yakalanan resimler yüz bilgilerini de içeren "sign" objesi ile birlikte PaintFace sınıfına gönderilmektedir.

Ardından yüzlerin gizli olarak oluşturulan resim monitörden gösterilerek hafızdan silinmekte ve kayıt işlemi yapılmamaktadır.

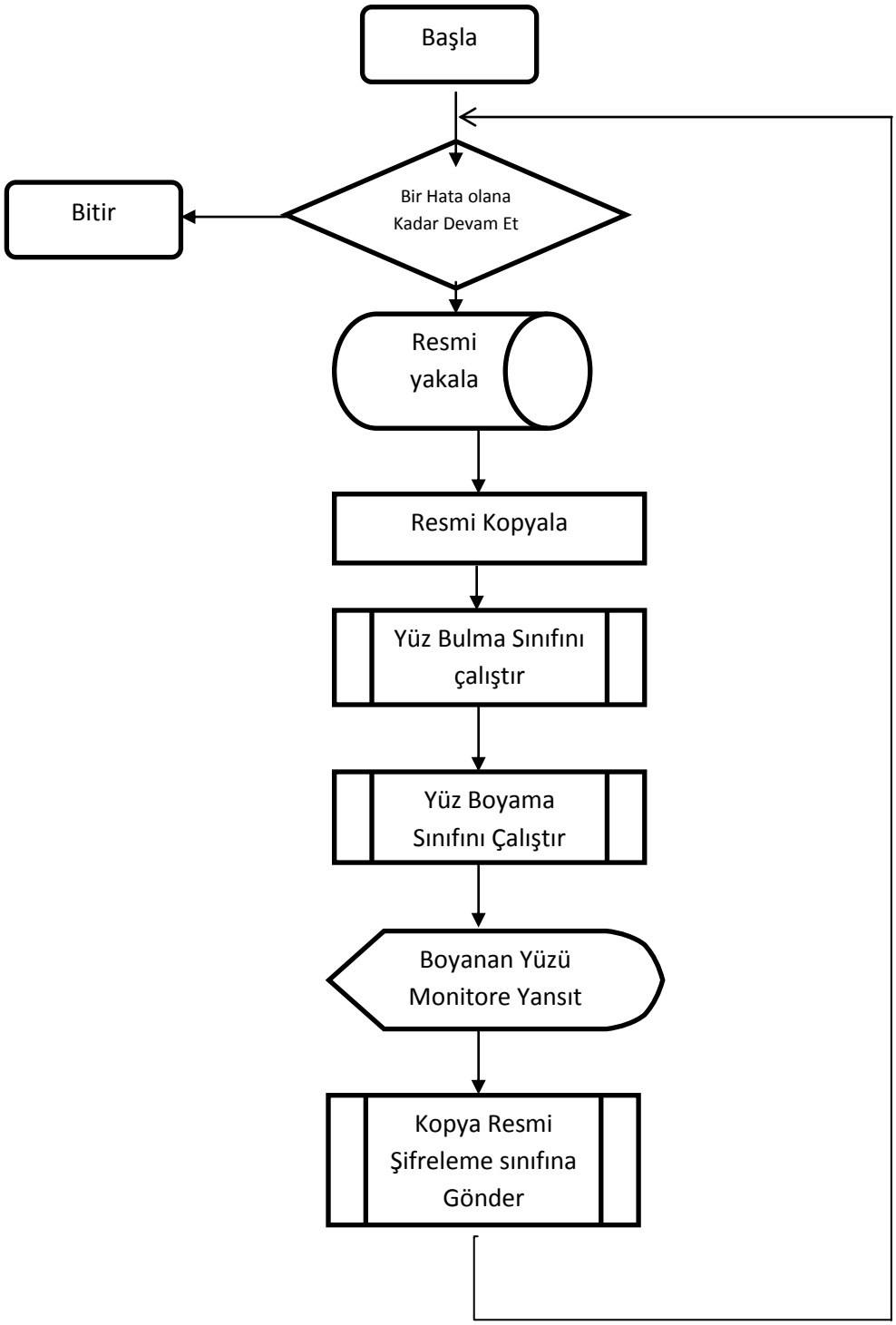
Orjinal olarak yakalanan ve üzerinde hiç işlem yapılmayan orjinal kayıt ise şifreleme sınıfına yollanarak şifrelenmekte ve bu şekilde kaydedilmektedir.

Şifreleme sınıfı çalıştıtılırken parola ve tuz değerleri de oluşturulup gönderilmektedir.

Gerektiğinde şifrenin çözülmesi ve orjinal videonun yüzlerin açık bir şekilde izlenebilmesi için deşifreleme sınıfının ayrıca çalıştırılması ve çalışma öncesinde doğru anahtarın girilmesi gerekmektedir.

Bu çalışmada anahtar rastal olarak oluşturulmakta ve konsola string değeri yazılmaktadır. Bu string değer deşifre sınıfını çalıştırırken kullanılmalıdır.

4.1.1. Akış Diagramı



Şekil.4. Ana Sınıf Diagramı

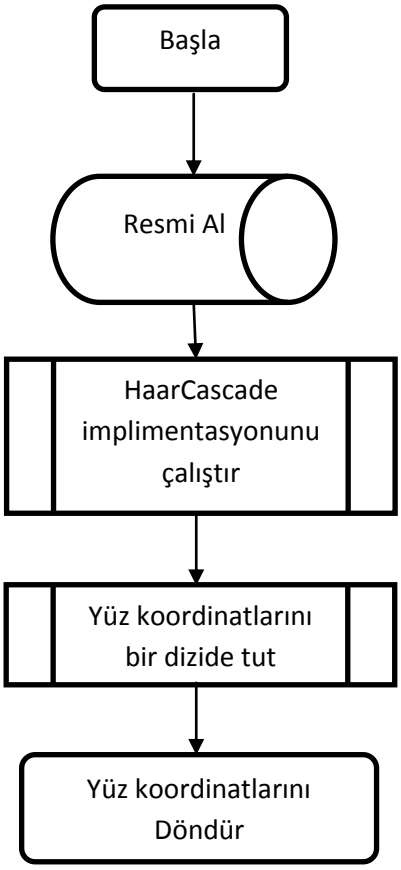
4.2. Yüz Tespiti

Ana sınıftan gönderilen resimlerden yüzleri tespit eden ve yüzlerin koordinat ve en-boy bilgisini içieren sign objesini döndüren FindFace sınıfı yazılmıştır.

Bu sınıf yüzleri tanımlayabilmek için haarcascade implimantasyonu kullanmıştır.

Şekil.5. Yüz Tespiti

4.2.1. Akış Diagramı



Şekil.6. Yüz Tespiti Diagramı

4.3. Resim Döndürme

Resmin verilen açıyla eğilip geri döndürülmesi için SkewGrayImage sınıfı yazılmıştır.

Alınan orjinal resmin üzerinde oynama yapabilmek ve piksel değerlerini başka piksellere atayabilmek için openCv kütüphanesinin CvMat sınıfından faydalanılarak oluşturulan objeden faydalanılmıştır.

Bu obje her bir piksel değeri için 0'dan 255'e kadar değer içeren verileri barındırmaktadır. Siyah için 0, beyaz için 255 olmak üzere renkleri barındırmayan gri bir resim yaratır.

Döndürülen resmi içeren yeni bir resim yaratılıp yüz bulma sınıfında kullanılmak üzere döndürülecek yeni bir resim yaratılcağı için bu yeni resmin boyutlarını doğru tespit etmek geekmektedir.

Q derece eğilecek olan yeni oluşacak resmin boyutları aşağıdaki şekilde hesaplanır;

DstCols yeni resmin eni, DstRowsyeni resmin boyu, SrcCols döndürülecek resmin eni, SrcRowsdöndürülecek resmin eni olmak üzere;

DstCols =Cos(Q)*SrcCols+Sin(Q)*SrcRows;
DstRows = Cos(Q)**SrcRows+ Sin(Q)*SrcCols;

Dönme işlemi için öncelikle yeni oluşturulan resmin ağırlık merkezi orjin (0,0) kabul edilerek eski resmin orjini ile çakışacak şekilde öteleme işlemi yapılır,

X ve Y eksenlerinde yapılacak öteleme işlemi için her iki resmin en ve boyları arasındaki fark değerlerinin yarısı kullanılır.

h ve k ötelenecek değerler olmak üzere;

$T(h,k) : R^2 \rightarrow R^2$

$x,y \rightarrow x' , y' = x - h , y - k$

hesaplaması ile resim ötelenir.

Döndürme işlemi için döndürülecek olan resmin bütün piksel değerleri alınarak yeni sisteme göre koordinatları hesaplandıktan sonra aşağıdaki formülle dönme işlemi gerçekleşdikten sonraki koordinat değerlerine aynı piksel değerleri yazılır.

(x,y) dönmeden önceki koordinatlar, (x’,y’) döndükten sonraki koordinatlar olmak üzere;

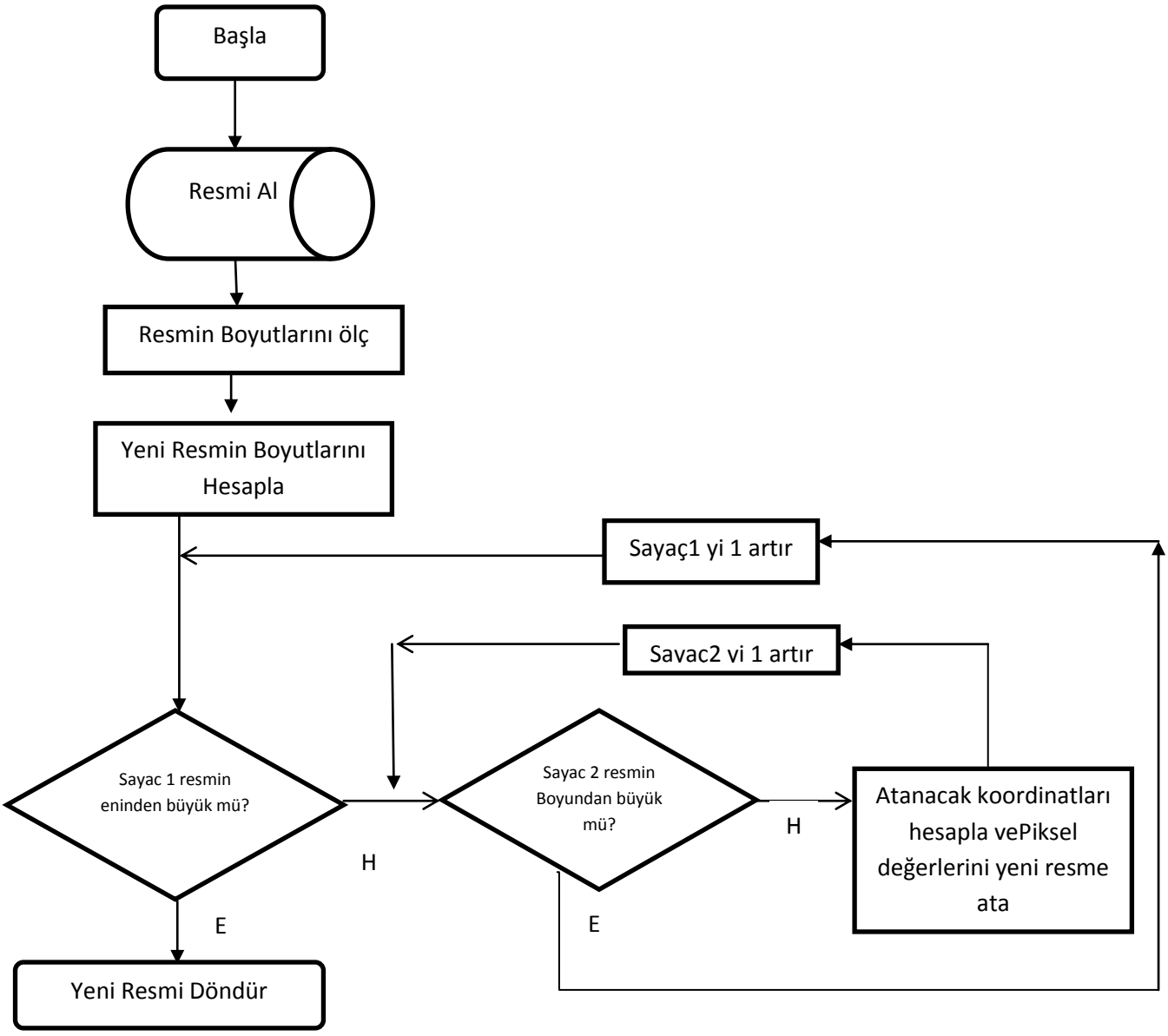
$x = x' \cos \theta - y' \sin \theta$

$y = x' \sin \theta + y' \cos \theta$

Yeni oluşan resim SkewGrayImage sınıfını çağıran ana sınıfa döndürülür.

Şekil.7. Yüz Döndürme

4.3.1. Akış Diagramı



Şekil.8. Yüz Döndürme Diagramı

4.4. Tespit Edilen Yüzlerin Gizlenmesi

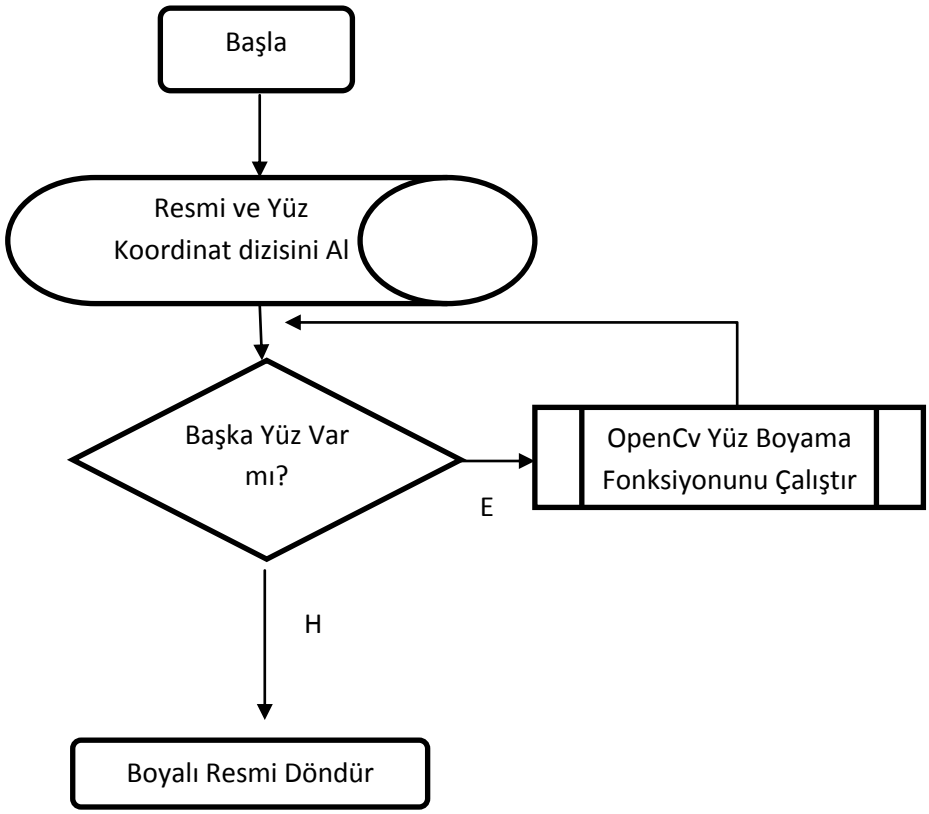
Tespit edilen yüzlerin gizlenmesi için PaintFace sınıfı yazılmıştır, bu sınıf aldığı resim ve yüz bilgilerini içieren sign objesinden faydalanarak tespit edilen yüzleri içi dolu bir dikdörtgen ile doldurarak yüzleri gizler ve yüzler gizlenmiş şekilde yeni bir resim oluşturur ve bu resmi döndürür.

Eğik olarak gelen resimlerden alınan sign objesi orjinal resimden daha büyük olduğu için yüzlerin koordinatlarını orjinal resimdeki yüzler ile aynı koordinatlarda bulunmadığı için eğik olarak gelen resmin öteleme ve dönme bilgilerini de alarak doğru bölgeyi boyar.

Ayrıca tranzpozesi alınarak eğilen yüzün de doğru işaretlenmesi bu alınan öteleme ve dönme bilgilerini kullanır.

Şekil.9. Tespit Edilen Yüzlerin Gizlenmesi

4.4.1. Akış Diagramı

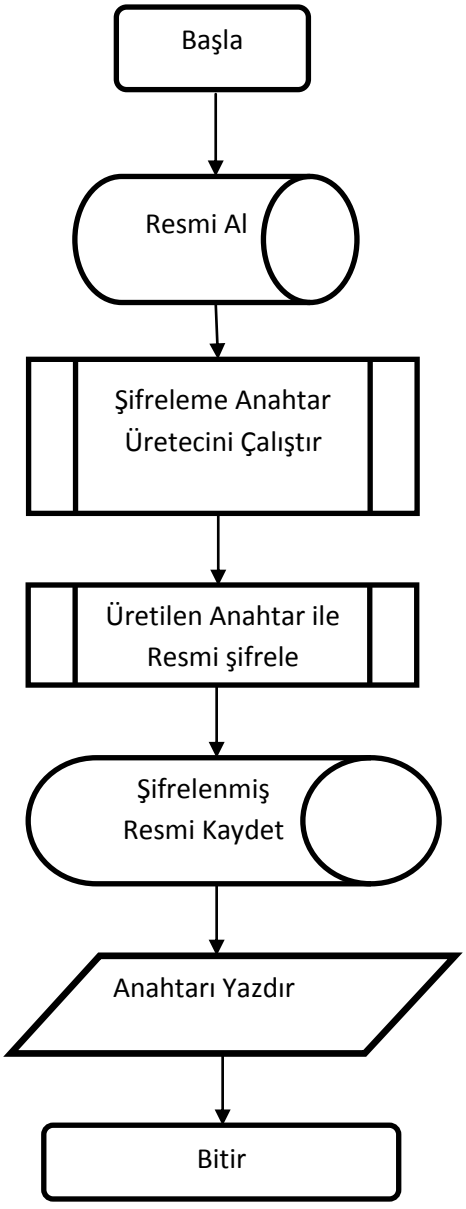


Şekil.10. Yüzlerin Gizlenmesi Diagramı

4.5. Gerçek Zamanlı Şifreleme

Şifreleme işlemi ve şifrelemede kullanılacak gizli anahtar üretimi için AES algoritması kullanılmıştır, bunun için javax.crypto.cipher ve javax.crypto.KeyGenerator kütüphanelerinden yararlanılmıştır.

4.5.1. Akış Diagramı

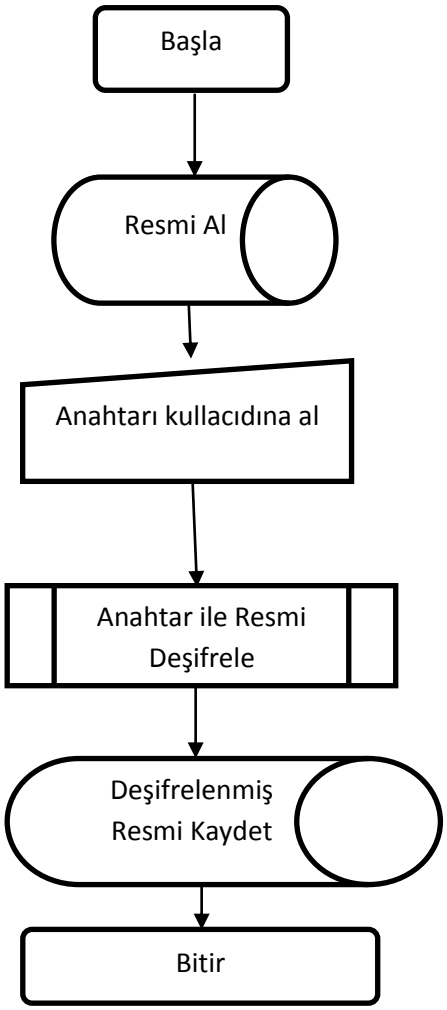


Şekil.11. Şifreleme Diagramı

4.6. Deşifreleme

Şifreleme işleminde olduğu gibi deşifreleme işlemi için de AES algoritması kullanılmıştır, deşifre işleminde kullanılacak olan anahtar, şifreleme yapılırken üretilen anahtar olmalıdır. Bu anahtar string olarak şifreleme esnasında konsola yazdırılmıştır.

4.6.1. Akış Diagramı



Şekil.12. Deşifreleme Diagramı

5. GENEL DEĞERLENDİRME VE SONUÇ

5.1. Sistemin Açıkları

Sistemin verimli çalışabilmesi, kullanılan haarcascade implimentasyonunun başarısına bağlıdır, profil başarısı zayıf olduğu için yüzler tespit edilememektedir.

Hareket takibi kullanılmadığı için 1 karede bile tespit edilemeyen yüz gizlenememektedir, bu durum da monitöre verilen görüntüden kötü niyetli kişilerce kaydedilmesi durumunda yüzün tanınabilmesine olanak sağlamaktadır.

Bu sistem demo prototip olarak üretildiği için şifreleme ve deşifreleme işlemlerinde kullanılacak olan anahtarların üretimi ve gizliliği konularında gerekli hassasiyet gösterilmemiştir.

5.2. Sistemin Geliştirilebilmesi İçin Olası Yöntemler

Yüz tanımlama sınıfının geliştirilmesi veya daha iyi çalışan yüz tanımlama algoritmalarının sisteme entegre edilmesi, özellikle profilden alınan yüz verilerinde sistemin daha sorunsuz çalışmasını sağlayacaktır.

Hareket takibi ile açı nedeniyle bir kaç kare yakalanamayan yüzlerin hafızada tutulması ve takip edilen objenin hızı hesaplanarak tespit edilen yüzün bir süre daha gizlenmesi sağlanabilir.

Resmi verilen açıyla eğmek için yazılan sınıf uygulamaya entegre edilerek daha mevcut algoritmayla daha iyi bir sonuç elde edilebilir.

Şifrelemede kullanılan anahtar şifreleme esnasında üretilmekte ve deşifre işleminde kullanılmak üzere konsola yazdırılmaktadır. Bu oluşturulan anahtar deşifre işlemini yapmaya yetkili kişi veya kurumlara ayrıca şifrelenerek yollanabilir veya bunun için hazır anahtar değişim protokolleri kullanılabilir. Farklı bir yöntem olarak da bu anahtar yetkili kişilerce üretilip belirli dönemlerde bir web servisi aracılığı ile uygulamanın çalıştığı sisteme gönderilebilir.

5.3. Sonuç

Sonuç olarak gerçek zamanlı olarak alınan kamera görüntülerinde tespit edilen insan yüzünün gizlenerek monitöre verilmesi ile özel hayatın gizliliği, eş zamanlı olarak orijinal görüntünün daha sonra yalnızca yetkili kişi ya da kurumların açabileceği şekilde şifrelenerek saklanması ile suçun önlenmesi ve suçlunun tespitine olanak sağlanması amacına ulaşılmıştır.

Sistemin açıkları ve bu açıkların giderilmesi için olası çözüm yolları araştırılmış ve anlatılmıştır.

KAYNAKLAR

1. Türkiye Cumhuriyeti Anayasası. (1982). http://www.tbmm.gov.tr/anayasa/anayasa_2011.pdf.

2. Türk Ceza Kanunu. (2004). http://www.tbmm.gov.tr/kanunlar/k5237.html

3. T.C. İstanbul Valiliği İl insan Hakları Kurulu Başkanlığı. (2010). Kameralı Takibi Kararı, http://www.istanbul.gov.tr/Portals/InsanHaklari/docs/kamera%20takip.pdf

4. T.C. Adalet Bakanlığı Kanunlar Genel Müdürlüğü. (2012). Kişisel Verilerin Korunması Kanunu Tasarısı Taslağı, http://www.kgm.adalet.gov.tr/Tasariasamalari/Basbakanlik/Basbakanlik.html

5. Yalman, Y ve Ertürk, İ. (2008). *Sayısal Ses İçerisinde Gizli Veri Transferinin Kablosuz Ortamda Gerçekleştirilmesi*. Politeknik Dergisi, 11(4), 319–327.

6. Bhirud P. P. ve Prabhu N. (2013). *Secured Biometric Authentication using Visual Cryptography and Transforms*. International Journal of Computer Application, 77(8), 23-28.

7. Deendayal T. ve Sunitha C. (2012). *Review of Various Visual Cryptography Techniques for Color Images*. International Journal of Computer Application, 2(2), 127-132.

8. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

9. Ross A. ve Othman A. A. (2010). *Visual Cryptography for Face Privacy*. Proc. of SPIE Conference on Biometric Technology for Human Identification VII. Orlando, USA.

10. Martin K. ve Plataniotis K. N. (2008). *Privacy Protected Surveillance Using Secure Visual Object Coding*. Multimedia Lab Technical Report 2008.01.

11. http://www.java.com/tr/

12. http://www.eclipse.org/downloads/

13. Intel Corporation. (2001).*Open Computer Vision Library Reference Manual*. USA.

14. http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html

EKLER

Ek-1: Main.java

```
package facePrivacy;

public class Main {
  private static final String ALGO_SECRET_KEY_GENERATOR = "AES";
  public static void main(String[] args) throws IOException{
      CanvasFrame canvas = new CanvasFrame("VideoCanvas");
canvas.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
      CvCapture capture1 = cvCreateCameraCapture(CV_CAP_ANY);
      FrameRecorder recorder1 = new
OpenCVFrameRecorder("RecordVid.avi",640,480);
                recorder1.setVideoCodec(CV_FOURCC('M','J','P','G'));
                recorder1.setFrameRate(6);
                recorder1.setPixelFormat(1);
                File inFile = new File("RecordVid.avi");
        File outFile = new File("enc_video.avi");
      try {
            SecretKey key =
KeyGenerator.getInstance(ALGO_SECRET_KEY_GENERATOR).generateKey();
          byte[] keyData = key.getEncoded();
          String str1 = new String(keyData);
          System.out.println("password "+str1);
          byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C,
                0x07, 0x72, 0x6F, 0x5A, (byte) 0x8E, 0x12, 0x39, (byte) 0x9C,
                 0x07, 0x72, 0x6F, 0x5A };
                AlgorithmParameterSpec paramSpec = new
IvParameterSpec(iv);


          IplImage originalImage;
          IplImage paintedImage;
```

```
IplImage reverseImage;
        IplImage imageLeft;
        IplImage imageRight;
        CvMat LeMat ;
        CvMat RiMat ;
        CvSeq sign;
        recorder1.start();
            while (true) {
          originalImage = cvQueryFrame(capture1);
          paintedImage=cvCloneImage(originalImage);
          CvScalar colour;
             double angle=0.5235;
             LeMat= paintedImage.asCvMat();
                  reverseImage=cvCloneImage(paintedImage);
 everseImage=TransposeImage.getTransposedImage(reverseImage);
                  RiMat= reverseImage.asCvMat();

                  sign = FindFace.getFoundFace(paintedImage);
                  colour=CvScalar.GRAY;
                  PaintFace.setColour(paintedImage,sign,colour);
    imageLeft=SkewGrayImage.getSkewedImage(LeMat,angle);
                  sign = FindFace.getFoundFace(imageLeft);
                  colour=CvScalar.RED;
                  PaintFace.setColour(paintedImage,sign,colour);

imageRight=SkewGrayImage.getSkewedImage(RiMat,angle);
                  sign = FindFace.getFoundFace(imageRight);
                  colour=CvScalar.YELLOW;
                  PaintFace.setColour(paintedImage,sign,colour);

                    if (paintedImage != null) {

                     canvas.showImage(paintedImage);
                     recorder1.record(originalImage);
```

```
                                        Encrypter.encrypt(key, paramSpec, new
FileInputStream(inFile), new FileOutputStream(outFile));
                                        }
                        }
        }
        catch (Exception e)
        {   System.out.println("bir hata olustu");
        }
            cvDestroyWindow("LiveVid");
            cvReleaseCapture(capture1);
  }

}
```

Ek-2: FindFace.java

```
package facePrivacy;
public class FindFace {
  public static final String XML_FILE =
                "resources/haarcascade_frontalface_default.xml";
  public static CvSeq getFoundFace(IplImage src){
            CvHaarClassifierCascade cascade = new
                        CvHaarClassifierCascade(cvLoad(XML_FILE));
            CvMemStorage storage = CvMemStorage.create();
            CvSeq sign = cvHaarDetectObjects(src,cascade,storage,1.5,3,
                        CV_HAAR_DO_CANNY_PRUNING);
            cvClearMemStorage(storage);
            return sign;
  }
}
```

Ek-3: SkewGrayImage.java

```
package facePrivacy;
public class SkewGrayImage {
   public static IplImage getSkewedImage(CvMat SrcMat, double angle)
    {
        double sin = - Math.sin(angle);
        double AbsSin = Math.abs(sin);
        double cos = - Math.cos(angle);
        double AbsCos = Math.abs(cos);
        int SrcCols = SrcMat.cols();
        int SrcRows = SrcMat.rows();
        int DstCols = (int) ( AbsCos*SrcCols+AbsSin*SrcRows);
        int DstRows = (int) ( AbsCos*SrcRows+AbsSin*SrcCols );
        CvMat DstMat = cvCreateMat(DstRows, DstCols, CV_8UC1);
        int shiftCs= (int)(SrcCols/2);
        int shiftRs= (int)(SrcRows/2);
        int shiftCd= (int)(DstCols/2);
        int shiftRd= (int)(DstRows/2);
        int srow;
        int scol;
        int drow;
        int dcol;
```

```
            for(int irow = 0; irow < SrcRows; irow++){
        srow=irow-shiftRs;
        for(int icol = 0; icol < SrcCols; icol++){
                scol=icol-shiftCs;
                drow=(int)(srow*AbsCos+scol*AbsSin);;
                dcol=(int)(-srow*AbsSin+scol*AbsCos);
                DstMat.put(drow+shiftRd, dcol+shiftCd, (int) SrcMat.get(irow,icol));
         }
            }
            IplImage Dst = cvCreateImage(cvSize(DstCols, DstRows),
IPL_DEPTH_8U, 1);
            Dst = DstMat.asIplImage();
            return Dst;
          }
       }
```

Ek-4: PaintFace.java

```
package facePrivacy;

public class PaintFace {

public static  void setColour(final IplImage src,final CvSeq sign,final CvScalar colour ){

        int total_Faces = sign.total();

        for(int i = 0; i < total_Faces; i++){

            CvRect r = new CvRect(cvGetSeqElem(sign, i));

            cvRectangle (src,cvPoint(r.x(), r.y()),
        cvPoint(r.width() + r.x(), r.height() + r.y()), colour, -1,CV_AA,0);
        }

  }

}
```

Ek-5: Encrypter.java

```
package facePrivacy;

public class Encrypter {

   private final static int DEFAULT_READ_WRITE_BLOCK_BUFFER_SIZE = 1024;

   private final static String ALGO_VIDEO_ENCRYPTOR = "AES/CBC/PKCS5Padding";
```

```
@SuppressWarnings("resource")
public static void encrypt(SecretKey key, AlgorithmParameterSpec paramSpec, InputStream in, OutputStream out)
        throws NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException,
        InvalidAlgorithmParameterException, IOException {
    try {
        Cipher c = Cipher.getInstance(ALGO_VIDEO_ENCRYPTOR);
        c.init(Cipher.ENCRYPT_MODE, key, paramSpec);
        out = new CipherOutputStream(out, c);
        int count = 0;
        byte[] buffer = new byte[DEFAULT_READ_WRITE_BLOCK_BUFFER_SIZE];
        while ((count = in.read(buffer)) >= 0) {
            out.write(buffer, 0, count);
        }
    } finally {
        out.close();
    }
}
```

Ek-6: Decrypter.java

```
package facePrivacy;

public class Decrypter {

private final static int DEFAULT_READ_WRITE_BLOCK_BUFFER_SIZE = 1024;
private final static String ALGO_SECRET_KEY_GENERATOR = "AES";
  private final static String ALGO_VIDEO_ENCRYPTOR = "AES/CBC/PKCS5Padding";
  @SuppressWarnings("resource")
  public static void decrypt(SecretKey key, AlgorithmParameterSpec paramSpec, InputStream in, OutputStream out)
throws NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException,
        InvalidAlgorithmParameterException, IOException {
  try {
        byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C,
        0x07, 0x72, 0x6F, 0x5A, (byte) 0x8E, 0x12, 0x39, (byte) 0x9C,
         0x07, 0x72, 0x6F, 0x5A };

        paramSpec = new IvParameterSpec(iv);
        Cipher c = Cipher.getInstance(ALGO_VIDEO_ENCRYPTOR);
        c.init(Cipher.DECRYPT_MODE, key, paramSpec);
        out = new CipherOutputStream(out, c);
        int count = 0;
        byte[] buffer = new byte[DEFAULT_READ_WRITE_BLOCK_BUFFER_SIZE];
        while ((count = in.read(buffer)) >= 0) {
           out.write(buffer, 0, count);
        }
     } finally {
```

```
            out.close();
        }
    }

    public static void main(String[] args) {

        File outFile = new File("enc_video.avi");
        File outFile_dec = new File("dec_video.avi");

        try {
            SecretKey key = KeyGenerator.getInstance(ALGO_SECRET_KEY_GENERATOR).generateKey();
            //parolayi asagidaki stringe atayin
        String str1 = "ÙVo[————————————————Wi|ÀŸAoW‘";
                    byte[] keyData  = str1.getBytes();;
             SecretKey key2 = new SecretKeySpec(keyData, 0, keyData.length, ALGO_SECRET_KEY_GENERATOR);
            byte[] iv = new byte[] { (byte) 0x8E, 0x12, 0x39, (byte) 0x9C,
                0x07, 0x72, 0x6F, 0x5A, (byte) 0x8E, 0x12, 0x39, (byte) 0x9C,
                 0x07, 0x72, 0x6F, 0x5A };
                AlgorithmParameterSpec paramSpec = new IvParameterSpec(iv);

            Decrypter.decrypt(key2, paramSpec, new FileInputStream(outFile), new FileOutputStream(outFile_dec));


        } catch (Exception e) {
            e.printStackTrace();
        }

    }

}
```

ÖZGEÇMİŞ

8 kasım 1982 tarihi, Niğde ili Bahçeli kasabası doğumluyum. Liseyi Mersin ilinde, İçel Anadolu Lisesinde 2000 yılında tamamladıktan sonra aynı yıl Yıldız Teknik Üniversitesi Fen-Edebiyat Fakültesi Matematik Bölümü'ne kaydoldum. Lisans eğitimini 2009 yılında Ege Üniversitesi Fen Fakültesi Fakültesi Matematik Bölümü'nde tamamladım. 2012 Yılında Beykent Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı Yüksek Lisans programına kaydoldum. Askerliğimi 2012 yılı içerisinde tamamladım. 2006 Yılından beri Şans oyunları sektöründe çeşitli özel şirketlerde üst düzey yönetici olarak çalıştım ve hala aynı sektörde çalışma hayatımı sürdürmekteyim.

İlgi alanlarım şans oyunları ve kriptolojidir. 2009 yılında aldığım "Tuş takımı barındıran USB kimlik doğrulama aparatı" adında bir patentim ve şans oyunları sektöründe geliştirdiğim çeşitli oyun türleri bulunmaktadır.

Yabancı dilim İngilizcedir

Engin AKÇIL