

Görüntü İşlemede Yeni Bir Soluk, OPENCV

Ezgi Erişti¹

¹ İstanbul Ticaret Üniversitesi, Bilgisayar Mühendisliği Bölümü, İstanbul

eeristi@iticu.edu.tr

Özet: OpenCV, Intel tarafından geliştirilerek BSD lisansı ile lisanslanmış, “Bilgisayarla Göru/Görme” kütüphanesidir. Özellikle gerçek zamanlı uygulamalar hedef alınarak geliştirilmiş olması, ticari kullanımı dahil ücretsiz olması ve Windows, Linux, MacOS X gibi farklı platformlarda kullanılabilmesi bu kütüphaneyi diğer görüntü işleme araçlarından bir adım öne çıkarmaktadır.

Anahtar Sözcükler: OpenCV, Görüntü İşleme, Bilgisayarla Göru/Görme.

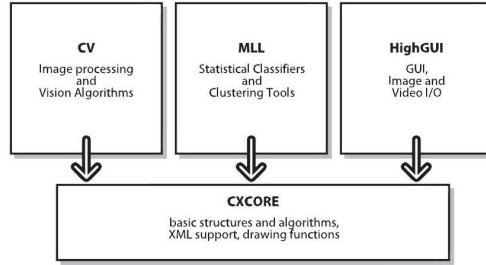
A Novel Refreshment for Image Processing, OPENCV

Abstract: OpenCV is a computer vision library which was developed by INTEL and released under a BSD license. Furthermore it was developed for real time computer vision applications and its free of charge usage is aimed for both academic and commercial clients. Featuring portability for different operating systems like Windows, Linux, MacOS X, OpenCV brings forward this library from other image processing tools.

Keywords: OpenCV, Image Processing, Computer Vision.

1. Giriş

OpenCV, bir resim ya da video içindeki anlamlı bilgileri çıkarıp işleyebilmek için INTEL tarafından C ve C++ dilleri kullanılarak geliştirilmiş, açık kaynak kodlu bir “Bilgisayarla Görme” kütüphanesidir.



Şekil 1. OpenCV Bileşenleri[1]

OpenCV kütüphanesi, beş temel bileşenden oluşmaktadır. Bu bileşenlerin dört tanesi Şekil 1’de görülmektedir [1].

Computer Vision (Bilgisayarla Göru/Görme) kelimesinin baş harfleri kullanılarak isimlendirilen **CV bileşeni**, temel resim işleme fonksiyonları ve Bilgisayarla Göru/Görme için kullanılan yüksek seviyeli algoritmaları bünyesinde barındıran beş temel kütüphaneden biridir. **Machine Learning Library** kelimesinin baş harfleri kullanılarak isimlendirilen **MLL bileşeni**, adından da anlaşılacağı üzere Makina Öğrenmesi dalı için gerekli istatistiksel verilere ulaşmak, mevcut verileri sınıflandırmak için kullanılan fonksiyonları/araçları içeren diğer bir kütüphanedir. **HighGUI bileşeni**, slider, form gibi OpenCV kütüphanesi içerisinde tanımlanmış pek çok nesneyi yaratabilmemizi sağlayan bir grafik arabirimi olmakla beraber, resim ve videoları kaydetmek, yüklemek, hafızadan silmek için gerekli giriş/çıkış (I/O) fonksiyonlarını da içerir [1].

CXCore bileşeni, OpenCV'ye ait IplImage, cvPoint, cvSize, cvMat, cvHistogram...vs gibi veri yapılarını bünyesinde barındıran, XML desteği de sağlayan bir kütüphanedir. Son olarak **CvAux bileşeni**, şablon eşleştirme (template-matching), şekil eşleştirme (shape matching), bir objenin ana hatlarını bulma (finding skeletons), yüz tanıma (face-recognition), ağız hareketleri izleme (mouth-tracking), vücut hareketlerini tanıma (gesture recognition) ve kamera kalibrasyonu gibi daha pek çok deneysel algoritmaları bünyesinde barındıran kütüphanedir [1].

OpenCV kütüphanesi, BSD lisansı ile lisanslanmıştır. Özgür lisanslar içinde en özgürü olarak bilinen bu lisansla kodu alan kişi, istediği gibi kullanma özgürlüğüne sahiptir [2]. Akademik ve ticari kullanımı ücretsiz olan bu kütüphane Windows, Linux, MacOS X gibi farklı platformlarda kullanılabilir [3].

Intel'in görüntü işleme laboratuvarlarında geliştirilen ve hız açısından optimize edilen OpenCV kütüphanesi, gerçek zamanlı uygulamalar hedef alınarak geliştirilmiştir. USB 2.0 teknolojisi ile birlikte artık standart bir bilgisayarda bile gerçek zamanlı uygulamalar çalıştırılabilir. Tüm bu gelişmeler oyuncak yapımından endüstriyel üretime kadar pek çok alanda bu kütüphanenin kullanılmasına yol açmıştır [4].

2. OpenCV ile Temel Uygulamalar

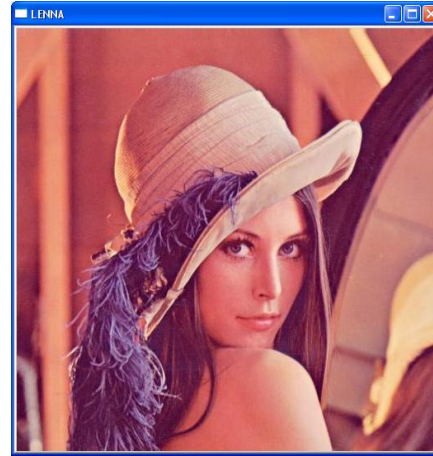
Bu bölümde OpenCV kütüphanesini etkin bir şekilde kullanabilmek için basit örnekler yer verilmiştir. Tüm örnekler Windows XP işletim sistemi üzerinde OpenCV_1.1pre1a sürümü ve Visual Studio 2005 Yazılım Geliştirme Ortamı (IDE) kullanılarak çalıştırılmıştır. Özellikle Windows işletim sistemi ile OpenCV kütüphanesini kullanırken pek çok sorunla karşılaşmıştır. Bu yazıda karşılaşılan problemlerin neler olduğuna ve çözümlerine de yer verilmiştir.

2.1 Bir Resmin Görüntülenmesi

Aşağıdaki program dosya yolu verilen bir resmin OpenCV kütüphanesi ile nasıl açılacağını göstermektedir [1].

```
//PROGRAM 1
#include <highgui.h>
int main()
{
    IplImage* img;
    img = cvLoadImage("C:\\\\Lenna.jpg");
    cvNamedWindow("LENNA", 1);
    cvShowImage("LENNA", img);
    cvWaitKey(0);
    cvReleaseImage(&img);
    cvDestroyWindow("LENNA");
    return 0;
}
```

IplImage veri yapısı, çeşitli resim dosyalarını hafızaya alabilmek için oluşturulmuş özel bir veri yapısıdır. Program 1'de istenen resim IplImage tipinde bir değişkene cvLoadImage fonksiyonu yardımıyla atanır. Bu fonksiyon, BMP, DIB, JPEG, JPG, JPE, PNG, PBM, PGM, PPM, SR, RAS, TIFF ve TIF uzantılı resim dosyalarını okuyabilir [5]. Okunabilen resim dosyaları cvShowImage Fonksiyonu ile form üzerinde gösterilir. İşlem bittikten sonra yaratılan bütün nesneler hafızadan silinir.



Şekil 2. Lenna – Standart Test İmaji

Şekil 2'de Program 1 çalıştırıldığında elde edilen sonuç görülmektedir.

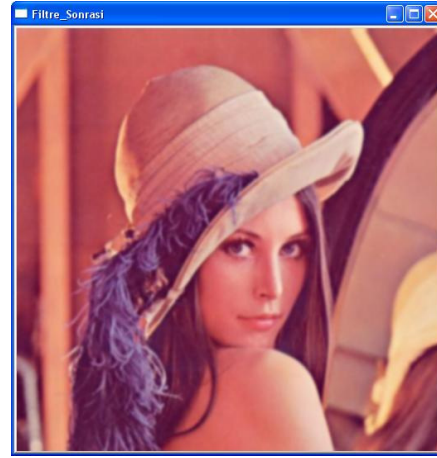
Belirtilen resim dosyası okunamadığı durumlarda aşağıdaki kontroller ve işlemler yapılmalıdır.

- Programda resim dosyasına ait dosya yolunun (path) doğru şekilde belirtildiğinden emin olunuz.
- Visual Studio 2005 yazılım geliştirme ortamı ile çalışıyorsanız “Visual Studio 2005 Service Pack 1” yükleyiniz.
- Seçtiğiniz yazılım geliştirme ortamı ile OpenCV kütüphanesini çalıştırabilmek için gerekli ayarları kontrol ediniz.

2.2 Resimlere Filtre Uygulama

Aşağıdaki programda, seçilen bir resme Gaussian Filtresi uygulanmıştır.

```
//PROGRAM 2
#include "cv.h"
#include "highgui.h"
int main()
{
    IplImage* img;
    img = cvLoadImage("C:\\\\Lenna.jpg");
    cvNamedWindow("Filtre_Oncesi",1);
    cvShowImage("Filtre_Oncesi",img);
    cvNamedWindow("Filtre_Sonrasi",1);
    IplImage* img2;
    CvSize boyut;
    boyut=cvGetSize(img);
    img2 = cvCreateImage(boyut,8,3);
    cvSmooth(img,img2,CV_GAUSSIAN,5,5);
    cvSmooth(img2,img2,CV_GAUSSIAN,5,5);
    //Filtre Seçenekleri
    //0 CV_BLUR_NO_SCALE
    //1 CV_BLUR
    //2 CV_GAUSSIAN
    //3 CV_MEDIAN
    //4 CV_BILATERAL
    cvShowImage("Filtre_Sonrasi",img2);
    cvWaitKey(0);
    cvReleaseImage(&img);
    cvReleaseImage(&img2);
    cvDestroyWindow("Filtre_Oncesi");
    cvDestroyWindow("Filtre_Sonrasi");
    return 0;
}
```



Şekil 3. Gaussian Filtresi Uygulanmış Resim

Yukarıdaki programda Şekil 2’deki resme ard arda 2 defa Gaussian Filtresi uygulanarak Şekil 3’deki resim elde edilmiştir.

Programda “cvsmooth” fonksiyonunun parametreleri değiştirilerek resme farklı filtreler de uygulanabilir.

2.3 Kenar Bulma Uygulamaları

Bütün kenar bulma işlemlerinde verilen resim önce Gri-Tonlu (Gray-Scale) resme çevrilir daha sonra istenen alt ve üst eşik değerlerine göre resim üzerindeki kenar noktaları tespit edilir. Kenar bulma yöntemlerinden bazıları Canny, Sobel ve Laplace Kenar Bulma (Edge Detector) yöntemleridir.

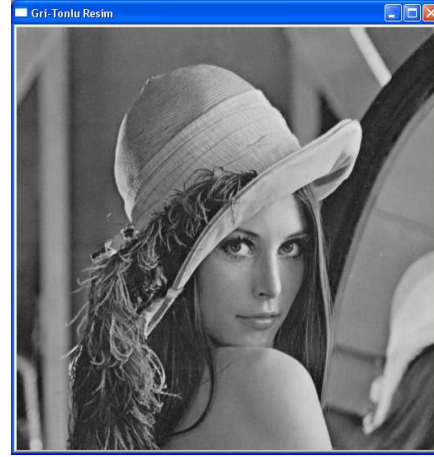
OpenCV kütüphanesi, Canny kenar bulma yöntemi için, ‘cvCanny’, Sobel kenar bulma yöntemi için ‘cvSobel’ ve Laplace kenar bulma yöntemi için, ‘cvLaplace’ isimli fonksiyonları içermektedir. Bu fonksiyonlara gerekli parametreler verilerek, resimdeki kenarlar tespit edilebilir.

Program 3’te bu fonksiyonların kullanımı gösterilmiştir.

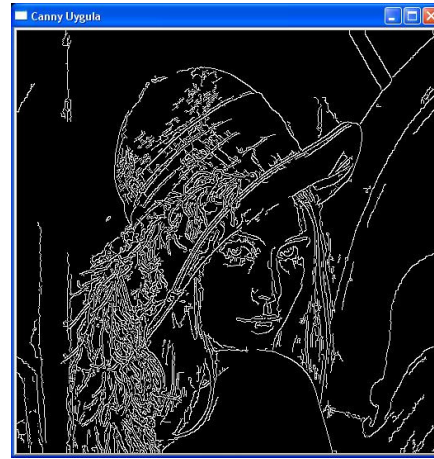
```

//PROGRAM 3
#include <cv.h>
#include <highgui.h>
#include <stdio.h>
int main( )
{
    int sec;
    IplImage *rgb, *gry;
    rgb=cvLoadImage( "C:\\\\Lenna.jpg");
    cvNamedWindow("Kaynak Resim",1);
    cvShowImage("Kaynak Resim",rgb);
    CvSize boyut=cvGetSize(rgb);
    gry = cvCreateImage(boyut,8,1);
    cvCvtColor(rgb,gry,CV_RGB2GRAY);
    cvNamedWindow("Gri-Tonlu Resim",1);
    cvShowImage("Gri-Tonlu Resim",gry);
    cvWaitKey(0);
    printf("1-Canny,2-Sobel,3-Laplace");
    printf("\nYontem Seciniz...\n");
    scanf("%d",&sec);
    if (sec==1){ //Canny
        IplImage* cny;
        cny = cvCreateImage(boyut,8,1);
        cvNamedWindow("Canny Uygula",1);
        cvCanny(gry,cny,45,120,3);
        cvShowImage("Canny Uygula",cny );
        cvWaitKey(0);
        cvReleaseImage(&cny);
        cvDestroyWindow("Canny Uygula");}
    else if (sec==2){ //Sobel
        IplImage* sbl;
        rgb=cvCloneImage(gry);
        sbl =
        cvCreateImage(boyut,IPL_DEPTH_16S,1);
        cvSobel (rgb, sbl, 1, 0, -1);
        cvConvertScaleAbs (sbl, gry);
        cvNamedWindow("Sobel Uygula",1);
        cvShowImage("Sobel Uygula",gry);
        cvWaitKey(0);
        cvReleaseImage(&sbl);
        cvDestroyWindow("Sobel Uygula");}
    else if (sec==3){ //Laplace
        IplImage* lplc;
        lplc =
        cvCreateImage(boyut,IPL_DEPTH_16S,1);
        cvLaplace (gry, lplc, 3);
        cvConvertScaleAbs (lplc, gry);
        cvNamedWindow("Laplace Uygula",1);
        cvShowImage("Laplace Uygula",gry);
        cvWaitKey(0);
        cvReleaseImage(&lplc);
        cvDestroyWindow("Laplace Uygula");}
    else printf("Yanlis Girdiniz.");
    cvWaitKey(0);
    cvReleaseImage(&rgb);
    cvReleaseImage(&gry);
    cvDestroyWindow("Kaynak Resim");
    cvDestroyWindow("Gri-Tonlu Resim");
    return 0;
}

```

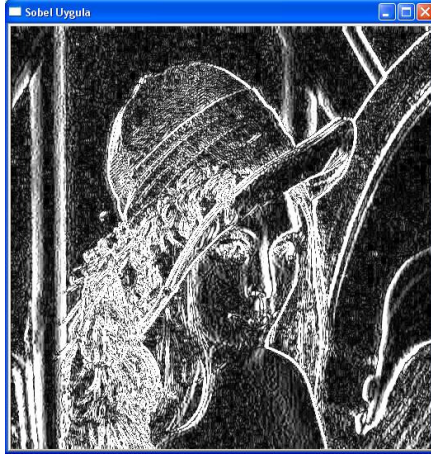


Şekil 4. Gri Tonlu Resim

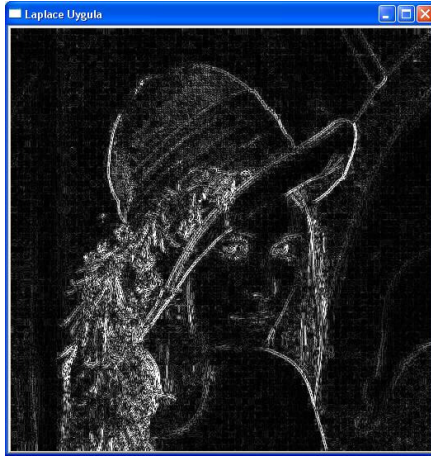


Şekil 5. cvCanny Fonksiyonu Çıktısı

Program 3, Şekil 2'deki standart test imajı (Lenna.jpg) ile çalıştırıldığında, Şekil 4'teki Gri tonlu resim ile program akışındaki seçime bağlı olarak bu üç kenar bulma yöntemlerinden biri uygulanmış resim elde edilir. Şekil 5'te Canny kenar bulma yöntemi ile kenarları tespit edilen resim, Şekil 6'da Sobel kenar bulma yöntemi ile kenarları tespit edilen resim, Şekil 7'de ise Laplace kenar bulma yöntemi ile kenarları tespit edilen resim görülmektedir.



Şekil 6. cvSobel Fonksiyonu Çıktısı



Şekil 7. cvLaplace Fonksiyonu Çıktısı

2.4 Kamera ile Gerçek Zamanlı Görüntü Yakalama

OpenCV kütüphanesi ile USB ya da dahili web kamerasından görüntü yakalayarak gerçek-zamanlı uygulamalar geliştirilebilir. Sistemde birden fazla kamera kullanılması durumunda gerekli kamera ID'si belirlenip ilgili metoda parametre olarak girilmelidir.

Program 4'te herhangi bir kameradan görüntü yakalayabilmek için gereken temel kodlara yer verilmiştir.

“cvCaptureFromCAM” metodu kullanılan kameranın ID'sini parametre olarak alır. Sistemde tek bir USB kamera kullanılıyorsa parametre olarak 0 (CV_CAP_ANY), birden fazla kamera kullanılıyorsa 100 (CV_CAP_MIL), 200 (CV_CAP_VFW) ya da 300 (CV_CAP_FIREWIRE, CV_CAP_IEEE1394, ..vs) değerlerinden biri parametre olarak kullanılır. Sistemde kullanılan kameranın çeşidine ve sayısına göre, deneme yanılma yöntemiyle gereken parametre seçilir.

Program 4'te “cvCaptureFromCAM” metodu ile “CvCapture” tipinden bir değişkene görüntü gelmesi sağlandıktan sonra sonsuz bir döngü yardımıyla yakalanan görüntünün içerisindeki çerçeveler/resimler (frame) “cvQueryFrame” metodu ile tek tek sorgulanıp okutularak ekranda gösterilir. ESC'a basıldığında görüntü yakalama işlemi sona erer.

```
//PROGRAM 4
#include "highgui.h"
#include "stdio.h"
int main( )
{
    cvNamedWindow( "GORUNTU",1);
    CvCapture* video=cvCaptureFromCAM(0);
    if (video==NULL)
    {
        printf("Dosya okunamadi..\n");
        return 0;
    }
    IplImage* frame;
    while(1)
    {
        frame=cvQueryFrame(video);
        if ( !(frame) ) break;
        cvShowImage( "GORUNTU", frame);
        char c = cvWaitKey(30);
        if ( c == 27 ) break;
    }
    printf("Okuma islemi bitmistir.\n");
    cvReleaseCapture( &video );
    cvDestroyWindow( "GORUNTU" );
    cvWaitKey(0);
    return 0;
}
```

Bu kodlar ile bir TV kartı aracılığıyla elde edilen görüntü de yakalanabilir.

2.5 Bir Video Dosyasının Oynatılması

OpenCV kütüphanesi ile bir video dosyasını okumak, bir resim dosyasını okumak kadar kolaydır. Aşağıdaki program ile avi uzantılı dosyaları açabilirsiniz.

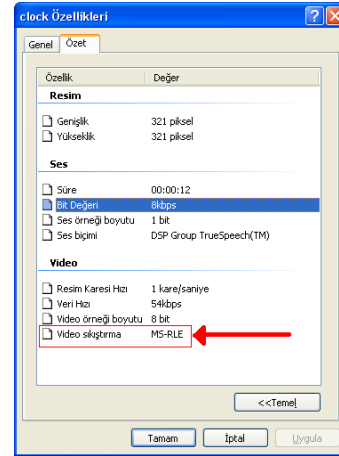
```
//PROGRAM 5
#include "highgui.h"
#include "stdio.h"
int main( )
{
    cvNamedWindow("Avi Dosyası",1);
    CvCapture* video;
    video=cvCaptureFromFile("A.avi" );
    //video=cvCaptureFromAVI("A.avi" );
    int zaman, fps;
    fps=cvGetCaptureProperty(video,5);
    zaman=1000/fps;
    if (video==NULL)
        printf("Dosya okunamadı..\n");
    IplImage* frame;
    while(1)
    {
        frame=cvQueryFrame(video);
        if (!(frame)) break;
        cvShowImage("Avi Dosyası", frame);
        char c = cvWaitKey(zaman);
        if ( c == 27 ) break;
    }
    cvWaitKey(0);
    cvReleaseCapture(&video);
    cvDestroyWindow("Avi Dosyası");
    return 0;
}
```



Şekil 8. Video dosyası okunamadığı durumda açılan pencere

Belirtilen video dosyası okunamadığı durumlarda “video” değişkeninin içeriği null gelmekte ve Şekil 8.’deki program çıktısı elde edilmektedir. Bu gibi durumlarda aşağıdaki kontroller ve işlemler yapılmalıdır.

- Programda video dosyasına ait dosya yolu (path) belirtilmemişse okunmak istenen video dosyası, proje dosyasının içerisindeki ilgili klasöre taşınmalı ya da dosya yolunun doğru şekilde programda belirtildiğinden emin olunmalıdır.
- “Cinepak Codec” gibi bazı codec’ler ile hazırlanmış video dosyaları Opencv ile okunabilir ancak değişik codec’ler ile kodlanmış avi dosyaları çoğu zaman (Windows üzerine kurulmuş) OpenCV kütüphanesi ile açılmamaktadır. Bu gibi durumlarda dosya özelliklerinden codec kontrolü yapılmalıdır (Şekil 9).



Şekil 9. Video dosyası özellik penceresi

- Eğer video herhangi bir codec ile sıkıştırılmış ise ‘VirtualDub’ [6] gibi bir aracı program ile dosyanın codec’ini değiştirilmeli ya da video dosyası sıkıştırılmamış avi dosyası haline getirilmelidir.

3. Sonular

OpenCV kütüphanesi, ücretsiz olması, işletim sistemi bağımsız olması, gerçek zamanlı uygulamalarda kullanılabilir olması ve 500’ün üzerinde fonksiyon desteğı sayesinde görüntü işleme ile ilgilenen kişiler için hızlı ve rahat bir geliştirme süreci sağlar. Ayrıca eğitim kurumlarında açık kaynak kodlu ders aracı olarak kullanılabilir.

4. Kaynaklar

[1] Bradski, G. and Kaehler, A., “Learning OpenCV: Computer Vision with the OpenCV Library”, **O’Reilly Media**, Amerika Birleşik Devletleri, 16-17 (2008).

[2] <http://www.bilisim-kulubu.com/sozluk/>

[3] <http://opencv.willowgarage.com/wiki/>

[4] INTEL CORPORATION: Intel researchers teach computers to ‘read lips’ to improve accuracy of speech recognition software. M2 Presswire, Coventry, Apr 28,2003, pg1.

[5] OpenCV Reference Manuals - HighGUI Reference Manual

[6] <http://www.virtualdub.org/>

[7] OpenCV Reference Manuals - CV Reference Manual

[8] OpenCV Reference Manuals - CXCORE Reference Manual