T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

GÖRÜNTÜLERDE İSTENMEYEN ŞİŞE BULANIKLAŞTIRMA ALGORİTMASI

Yüksek Lisans Tezi

Tezi Hazırlayan:

Selim Can TEMELLİ

İSTANBUL, 2014

T.C.

BEYKENT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

GÖRÜNTÜLERDE İSTENMEYEN ŞİŞE BULANIKLAŞTIRMA ALGORİTMASI

Yüksek Lisans Tezi

Tezi Hazırlayan:

Selim Can TEMELLİ

Öğrenci No:

120820005

Danışman:

Yrd. Doç. Dr. Ediz ŞAYKOL

İSTANBUL, 2014

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum "Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması" 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. 04/09/2014

Selim Can TEMELLİ

İmza:

Adı ve Soyadı : Selim Can TEMELLİ

Danışmanı : Yrd. Doç. Dr. Ediz ŞAYKOL

Türü ve Tarihi : Yüksek Lisans, 2014

Alanı : Bilgisayar Mühendisliği

Anahtar Kelimeler : Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması, OpenCV, C++, Cascade Sınıflayıcı Alıştırması, Haar Alıştırması.

ÖZET

GÖRÜNTÜLERDE İSTENMEYEN ŞİŞE BULANIKLAŞTIRMA ALGORİTMASI

Bu çalışmada, C++ yazılım dili kullanılarak OpenCV kütüphanesiyle birlikte "Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması" tezi gerçekleştirilmiştir. Tezin amacı resim üzerinde istenilen şişenin belirlenmesi ve sansürlenmesidir. Tez kapsamında sansürlenecek nesne olarak bir içki şişesi kullanılmıştır. Microsoft Visual Studio 2010 derleyicisi kullanılarak kodlama işlemleri yapılmıştır. OpenCV kütüphaneleri projeye entegre edilmiştir. Modelleme için nesnenin pozitif ve negatif resimleri çekilmiştir. Haar alıştırması metodu kullanılarak belirlenen şişenin xml dosyası oluşturulmuştur. Oluşturulan xml dosyası belirtilen konumdan çağrılarak projeye eklenmiş ve daha sonra Cascade sınıflayıcı alıştırması yardımıyla kullanılmıştır. Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması uygulamasında görüntü arayüzden seçilmektedir. Seçilen görüntü için sansür butonuna basılır ve yazılımın içinde bulunan xml dosyasındaki nesneyle eşleşmesi sonucunda tespit edilerek sansürlenme işlemi gerçekleştirilmektedir. Nesnenin sansürlenmiş hali ekranda gösterilmektedir. Ayrıca arayüzden histogram eşitleme işlemi de yapılabilmektedir. Farklı resimler kullanılarak sansürleme işlemi tekrar tekrar uygulanmış olup ve aynı sonuca ulaşılmıştır. Tez ile alakalı çeşitli test ve deneyler yapılarak sorunsuz halde çalışması sağlanmıştır. Çalışmanın sonucu olarak "Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması" ortaya çıkarılmıştır.

Name and Surname : Selim Can TEMELLİ

Supervisor : Yrd. Doç. Dr. Ediz ŞAYKOL

Degree and Date : Master, 2014

Major : Computer Engineering

Key Words : Unwanted Bottle Blurring Algorithm On Images, OpenCV, C++, Cascade Classifier Training, Haar Training.

ABSTRACT

UNWANTED BOTTLE BLURRING ALGORITHM ON IMAGES

In this work, while using C++ programming language with OpenCV library "Unwanted Bottle Blurring Algorithm On Images" thesis practiced. Objective of the thesis is to determine the bottle which we want to cencor on the image and cencor it. Within thesis, an alcohol bottle used as test subject. Coding done with Microsoft Visual Studio 2010 compiler. OpenCV libraries are integrated to the project. As for modeling, the object photographed in negative and positive copies. With using Haar training, determined bottles xml file has created. Created xml file added to project with calling from designated folder and then used with help of the Cascade classifier practice. On Unwanted Bottle Blurring Algorithm On Images application, the images are chosen from interface. For the chosen image, hitting on the cencor button matches the object from the xml file and the process finishes. Then the cencored version of the image shows on screen. Also a histogram equalization process can be done from interface. Same process has been done multiple times with different images and resulted same. Different tests and experiments about the thesis are finished and reached an error free version. As for conclusion, "Unwanted Bottle Blurring Algorithm On Images" revealed.

İÇİNDEKİLER

Sayfa No:

ŞEKİLLER LİSTESİ

Sayfa No:

KISALTMALAR

Bat : Dos tabanlı toplu işlem dosyası

BSD : Berkeley Software Distribution

BÜ : Beykent Üniversitesi

BM : Bilgisayar Mühendisliği

Bmp : Bitmap resim dosyası

CV : Computer Vision

C++ : C with classes

Doc : Document (Belge)

DVD : Digital Versatile Disc (Çok Amaçlı Sayısal Disk)

Exe : Executable file

GİŞBA : Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması

(I/O) : Giriş/çıkış

MLL : Machine Learning Library

Mvs : Microsoft Visual Studio

OpenCV : Open Source Computer Vision Library

Pdf : Portable Document Format (Taşınabilir Belge Biçimi)

Pps : Power Point Script

Txt : Text, yazı dosyası

Xml : eXtensible Markup Language

GİRİŞ

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması OpenCV kütüphanesi yardımıyla C++ yazılım dili kullanılarak resimlerde istenilmeyen cisimleri sansürlemek amaçlı geliştirilmiş bir tezdir.

Görüntü işleme günümüzde hayatımızın çeşitli alanlarında karşımıza çıkmaktadır. Özellikle insan odaklı yapılması gereken birçok işlemin görüntü işleme ile birlikte makinalar tarafından yapılmasına olanak sağlamaktadır. Günlük hayatımızdan örnekler vermek gerekirse, otoyol ve köprülerde kullanılan plaka tanıma sistemleri, güvenlik kameralarda ki yüz tanıma sistemleri, parmak izi okuyucuları gibi birçok örnek sayılabilmektedir. Geliştirilmiş olan projelerin amacı yapay zekâ yardımıyla hareketli ve ya hareketsiz görüntüleri anlamlandırıp işlemektir. Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması tezi de aynı amaca hizmet etmektedir. Önceden belirlenmiş olan bir cisim, görüntülerden tespit edilerek sansürlenmektir. İnsan odaklı bir tarama yerine, görüntü işlemeden faydalanılarak tespit ve sansürleme işlemi yapılabilmektedir. Bununla beraber ciddi bir iş yükü ortadan kalkacaktır.

"Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması" çalışması hakkında bu tezde bilgiler verilecek ve nasıl gerçekleştirildiği anlatılacaktır. Tezin birinci bölümünde görüntü işlemeden faydalanarak görüntüler üzerinde ki nesneleri anlamlandırmaya yönelik daha önceden yapılmış benzer tezler hakkında bilgiler verilecektir. İkinci bölümde de tez ile alakalı, amaçlar, uygulanan yöntemler, kullanılan araçlar, kısıtlar ve tezi gerçekleştiren program hakkında bilgiler verilecektir. Üçüncü bölümde tez ile alakalı yapılmış deneysel çalışmaların sonuçları belirtilecektir. Ayrıca uygulamanın çalışmasıyla alakalı anlatımlarda bulunulacaktır.

1. LİTERATÜRDE YAPILMIŞ BENZER TEZLER

Görüntü işleme teknikleri kullanılarak görüntülerde bulunan nesnelerin anlamlandırılmasına bağlı olarak üzerinde işlemler yapılabilen literatüre geçmiş benzer tezlerden bahsedilmektedir.

Akciğer tomografileri kullanılarak yapay zeka ve görüntü işleme tekniklerine dayalı otomatik nodül bölge tespit yöntemi geliştirilmesi tezinde; akciğer BT (Bilgisayarlı Tomografi) kesit görüntüleri üzerinden juxtapleural nodül bölgelerinin otomatik tespiti ve tanılamasını sağlayan bir yöntem ve sistem geliştirilmiştir. **[1]**

Otomatik kan hücrelerinin tanınması ve sınıflandırılmasında değişmez momentlere dayalı görüntü işleme yöntemlerinin kullanılması tezinde; mikroskobik görüntülerindeki kan hücrelerinin tanınması ve sınıflandırılması için değişmez moment ve çoklu sınıf destek vektör makinelerine (ÇSDVM) dayalı bir yöntem önerilmiştir. **[2]**

Görüntü işleme teknikleri kullanarak optik karakter tanımlama tezi; Türkiye Cumhuriyeti (TC) kimlik numaralarının kamerayla çok kısa zamanda tespiti ve veri tabanından kişi bilgilerinin çağrılması gerçek zamanlı olarak amaçlanmıştır. Tez kapsamında, görüntü işleme teknikleri kullanarak kameradan alınan kimlik görüntülerinden kimlik bilgisinin otomatik olarak doğru tanınması önerilmiştir. **[3]**

Sabit görüntülerde görüntü işleme teknikleri ile orman yangını tespiti tezi; fotoğraf makinaları, kameralar ve insansız hava araçları (İHA) gibi görüntü sağlayıcılarından elde edilen veriler ışığında görüntü işleme teknikleri kullanılarak orman yangınlarının tespitini sağlayacak bir sistem üzerinde çalışılmıştır. **[4]**

Görüntü işleme teknikleri ve yapay zekâ yöntemleri kullanarak görüntü içinde görüntü arama tezi; mevcut bir görüntü içerisinde belirlenecek bir görüntünün görüntü işleme teknikleri ve yapay zekâ yardımıyla tespit edilmesine olanak sağlayacak bir uygulamanın tasarlanması için çalışmalar yapılmıştır. **[5]**

Görüntü işleme teknikleri kullanarak elma tasnifleme tezi; görüntü işleme teknikleri kullanarak kameradan elde edilen elma görüntülerinden kalitelerine ve özelliklerine göre sınıflandırılmasına olanak sağlayacak bir uygulama tasarlanmıştır. Uygulamanın görüntülerden elde ettiği veriler ile birlikte ciddi bir insan gücü ve zamandan tasarruf sağlanmıştır. **[6]**

Haber videolarının görüntü işleme yöntemleri ile haberlere bölütlenmesi tezi; video görüntülerini değerlendirerek haber programlarının haber geçişlerine göre bölütlenmesini sağlayan bir sistem tasarlanmış ve gerçekleştirilmiştir. **[7]**

Görüntü işleme esaslı parmak izi doğrulama; Yapay Sinir Ağı (YSA) tabanlı bir parmak izi tanıma sistemi geliştirilerek, parmak izinin daha geniş alanlarda kullanılmasına olanak sağlanmak amaçlanmıştır. **[8]**

Kamera kullanılarak görüntü işleme yoluyla gerçek zamanlı güvenlik uygulaması; hareketli görüntü sağlayıcılarından faydalanarak elde edilen görüntülerden, güvenlik ile alakalı durumları görüntü işleme teknikleri yardımıyla tespit edilmesini amaçlayan bir tez ortaya konulmuştur. **[9]**

İşaret dili harflerinin görüntü işleme yöntemleri ile tanınması için bir uygulama; sunulan bu çalışma da işaret dili ile iletişim kurulurken kullanılan el işaretlerinin görüntü işleme teknikleri kullanılarak anlaşılmasını amaçlayan bir tez ortaya konulmuştur. **[10]**

Mekatronik teknolojisinde görüntü işleme tekniklerine dayalı yüz tanıma sistemi geliştirilmesi; bu tezde, yüz tanıma problemi üzerinde araştırma yapılarak, kullanılan temel yöntemler incelenmiştir. Yapılan incelemelere paralel olarak da bir yüz tanıma uygulaması geliştirilmiştir. **[11]**

Görüntü işleme ile yüzey pürüzlülüğü ölçümü ve analizi tezi; işlenen yüzeylerde oluşan yüzey pürüzlülüğünün, görüntüleme cihazları kullanarak alınan yüzey görüntüsü üzerinden, görüntü işleme teknikleri ile ölçme işlemi ve değerlendirmesi yapılmıştır. Bu tez çalışmasında alınan görüntüler yüzey pürüzlülüğü açısından analiz edilerek izleyici uçlu yöntemle karşılaştırılmıştır. **[12]**

2. GÖRÜNTÜLERDE İSTENMEYEN ŞİŞE BULANIKLAŞTIRMA ALGORİTMASI

2.1. Amaç

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tezi; görüntü işleme teknikleri yardımıyla belirlenen nesnelerin istenilen görüntülerde taranıp tespit edilerek sansürlenmelerini amaçlamaktadır.

Tezde nesnelerin görüntülerden tespit edilerek üzerinde işlemler yapılmasına olanak sağlanması istenmektedir. Bununla birlikte öncelikle görüntüyü anlamlandırabilecek kalitede bir xml dosyası oluşturulması amaçlanmıştır. Daha sonra ki adım olarak da bu xml dosyasının kullanılarak görüntü tespitlerinin yapılacağı ve sansürleme işlemlerinin gerçekleşeceği yazılım meydana getirilmesi üzerine çalışılacaktır. Geliştirilen uygulama üzerinde testler yapılarak uygun bir şekilde çalışması istenilmektedir.

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tezi ile görüntü işleme teknikleriyle birlikte ciddi bir zaman ve insan gücü tasarrufunda bulunulması amaçlanmıştır. Bununla beraber sansürleme işleminin oldukça basit bir hale getirilmesi üzerine çalışılacaktır.

2.2. Yöntem ve Araçlar

2.2.1. OpenCV Nedir

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

OpenCV kütüphanesi, beş temel bileşenden oluşmaktadır. Bu bileşenlerin dört tanesi **Şekil 1**'de görülmektedir.

Computer Vision (Bilgisayarla Görü/Görme) kelimesinin baş harfleri kullanılarak isimlendirilen CV bileşeni, temel resim işleme fonksiyonları ve Bilgisayarla Görü/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 Makine 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.

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.

OpenCV kütüphanesi, BSD lisansı ile lisanslanmıştır. Özgür lisanslar içinde en özgürü olarak bilinen bu lisansta kodu alan kişi, istediği gibi kullanma özgürlüğüne sahiptir. Akademik ve ticari kullanımı ücretsiz olan bu kütüphane Windows, Linux, MacOS X gibi farklı platformlarda kullanılabilir. **[13]**

2.2.2. C++ Programlama Dili

C++, Bell Laboratuvarlarından Bjarne Stroustrup tarafından 1979 yılından itibaren geliştirilmeye başlanmış, C'yi kapsayan ve çok paradigmalı, yaygın olarak kullanılan, genel amaçlı bir programlama dilidir. İlk olarak C With Classes (Sınıflarla C) olarak adlandırılmış, 1983 yılında ismi C++ olarak değiştirilmiştir.

Genel olarak her C programı aynı zamanda bir C++ programıdır, ancak her C++ programı bir C programı değildir. Bu durumun bazı istisnaları mevcuttur. C++'ı C'den ayıran özellikler C++'ın nesne paradigması kullanılarak programlamaya olanak tanıyan özelliklerdir. Sınıflar sayesinde yeni veri türleri yaratılabilir veya var olan türlerden yenileri türetilebilir. Ayrıca çok biçimlilik sayesinde bir sınıf tanımıyla yazılmış kod, o sınıf türünden türetilmiş yeni sınıflarla da çalışabilir. **[14]**

2.2.3. Histogram Eşitleme

Histogram, sayısal bir resim içerisinde her renk değerinden kaç adet olduğunu gösteren grafiktir. Bu grafiğe bakılarak resmin parlaklık durumu ya da tonları hakkında bilgi sahibi olunabilmektedir.

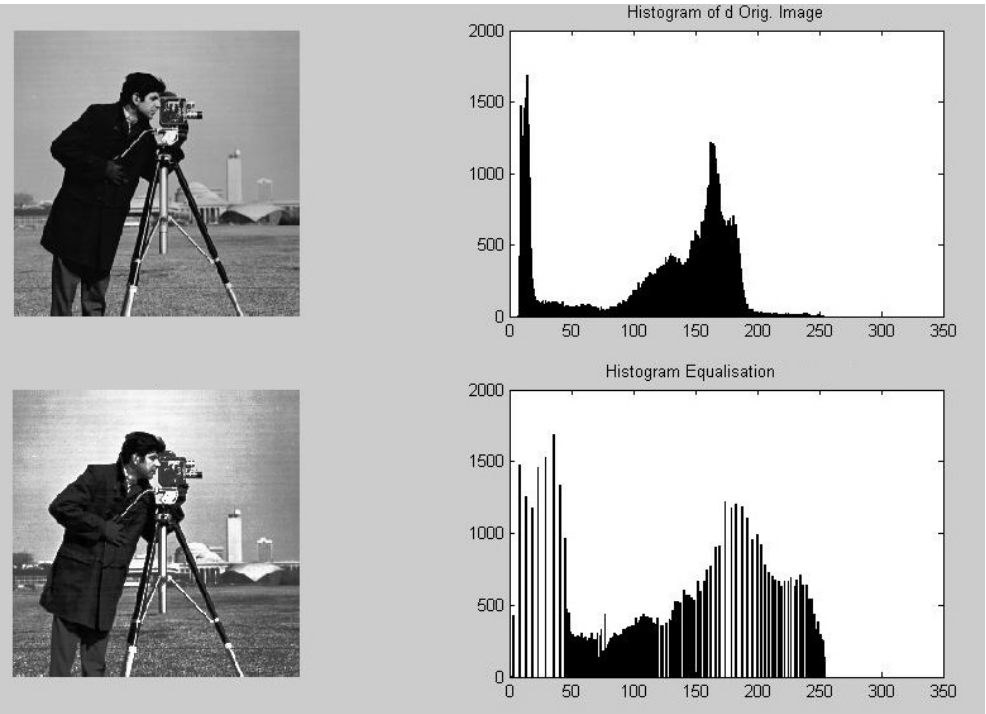
Resimler günün değişik zamanlarında çekildiğinde farklı gri parlaklık/kontrast değerlerine sahip olurlar. Kapalı havalarda veya ışığın kısıtlı olduğu ortamlarda çekilen resimler koyu gri tonlarda iken güneş ışığına maruz kalan

resimler yüksek parlaklığa sahiptir. Özellikle geceleri güvenlik kameralarından alınan görüntülerde histogram eğrisi karanlık ortamların çokluğu nedeniyle düzgün olmayacaktır ve bu görüntüde yer alan ayrıntılar görünemeyecektir. Bunu engellemek için histogram üzerine uygulanabilecek iki temel işlem vardır. Bunlar histogram eşitleme ve histogram germedir.

Histogram eşitleme; histogram eşitleme ile belirli bir ton etrafında toplanan histogram eğrisi (0-255) tonları arasına düzgün bir şekilde dağıtılır böylece resmin renk dağılımının homojen olarak yapılandırılması sağlanır.

Histogram eşitleme işleminde, resmin kümülatif renk seviyeleri dağılımı üzerinde normal dağılım uygulanmaktır. Bu yeniden dağılım, renk seviyeleri dağılımında dengeleme sağlamaktadır. (**Şekil 2)** Histogram eşitleme için geliştirilen algoritma adımları aşağıdaki şekildedir.

- Histogramı çıkar.
- Kümülatif histogramı hesapla ve normalize et.
- Normalize kümülatif histogram eğrisini kullanarak pikselleri yeniden hesapla. **[15]**



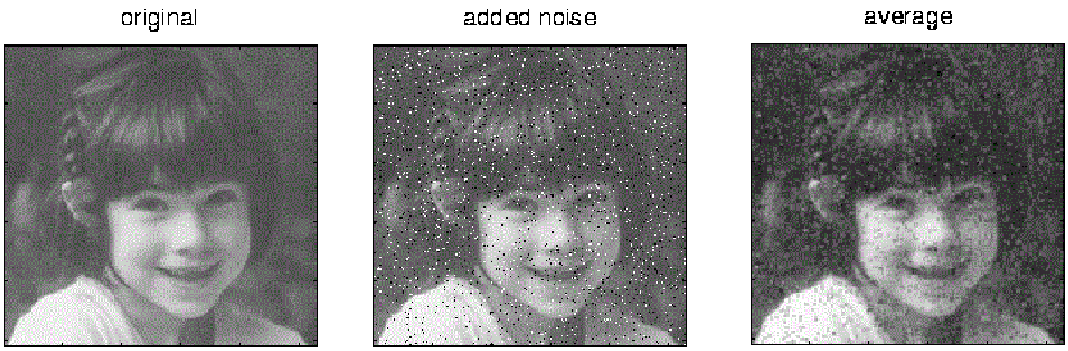
Şekil 2. Histogram eşitleme

2.2.4. Görüntü Filtreleme

Filtreleme resmin üzerinde bir filtre varmış gibi düşünüp her piksel değerinin yeniden hesaplanmasıdır. Filtreleme sayesinde görüntü üzerinde netleştirme, belirli ayrıntıları ortaya çıkarma, görüntüyü yumuşatma, kenar keskinleştirme veya kenar bulma gibi işlemler gerçekleştirilir. Filtreler genelde 3×3 lük matrislerdir. Fakat boyutları 5×5, 7×7, 9×9, 11×11 şeklinde olabilir.

2.2.4.1. Ortalama (Average) Filtresi

Resimdeki her piksel yerine komşuları ile beraber ortalaması alınarak yeniden hesaplanır. Resimdeki gri düzeyler arasında keskin geçişler azalır; daha yumuşak geçişler söz konusudur. Resim üzerindeki kenarlarda bulanıklaşmaya (blur) yol açarlar. Örnek **Şekil 3**'de görülmektedir.

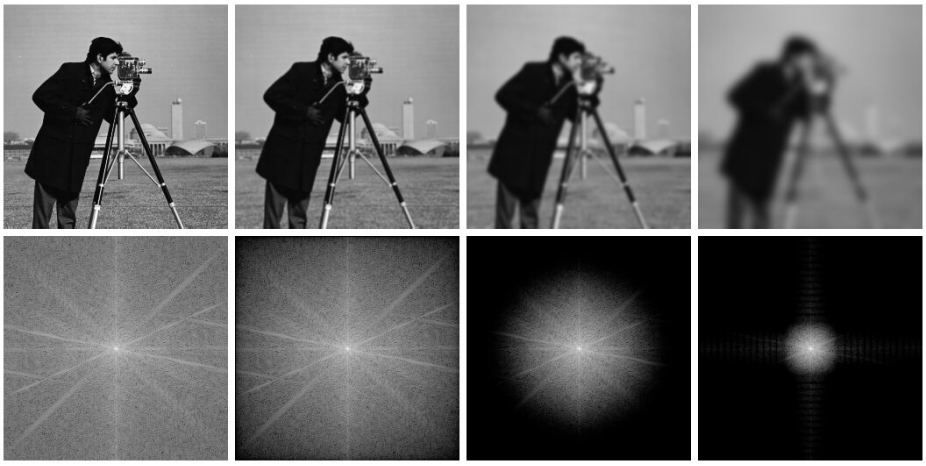


Şekil 3. Ortalama (Average) filtresi

2.2.4.2. Gaussian Filtresi

Ortalama filtrenin Gaussian dağılımını kullanarak biraz daha değiştirilmiş hali Gaussian filtresi olarak bilinir. Gaussian filtreleme aynı zamanda bir fourier dönüşümüdür. Gauss filtre ile sonsuz bir transfer fonksiyonuna karşılık mekânsal alanda sonlu bir pencerede (tarama penceresi) filtreleme yapılabilmektedir. Bu da filtrelemenin temel problemini daha kolay çözülebilir hale getirir.

Gauss Yumuşatmasının Avantajları: Filtreleme önce yatay ardından çıkan sonuçla düşey eksende gerçekleştirilebilir. (**Şekil 4**)

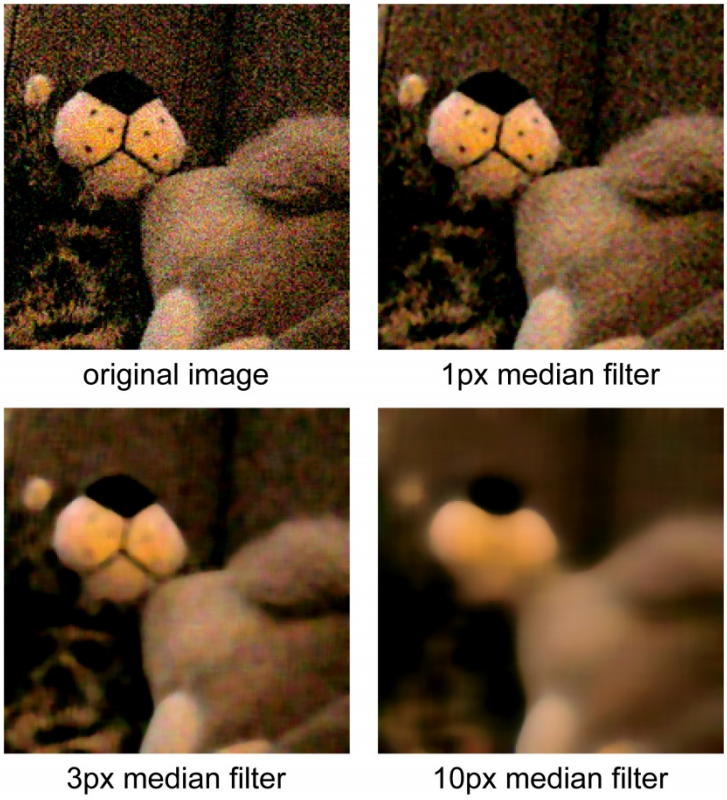


Şekil 4. Gaussian filtresi

2.2.4.3. Medyan Filtresi

Bu filtreleme yönteminde, orijinal sıralanmış piksel komşularının arasındaki ortanca değer ile değiştirilir. Bunun ağırlıklı ortalama filtrelerinden farkı şudur: Ağırlıklı ortalama filtrelerinde, komşuların ağırlıklı ortalaması alınır, hesaplanan bu değer orijinal piksel ile yeniden ortalanarak sonuç bulunur. Ortanca filtresinde ise, komşuluk değerleri önce sıraya konulur, sonra ortadaki değer alınır. Bu değer doğrudan sonuç kabul edilir. Ortanca değeri net elde edebilmek için genellikle tek sayıda komşu seçilir. Eğer hesaplamada çift sayıda komşu kullanılırsa, bu durumda ortada kalan iki pikselin aritmetik ortalaması kullanılır.

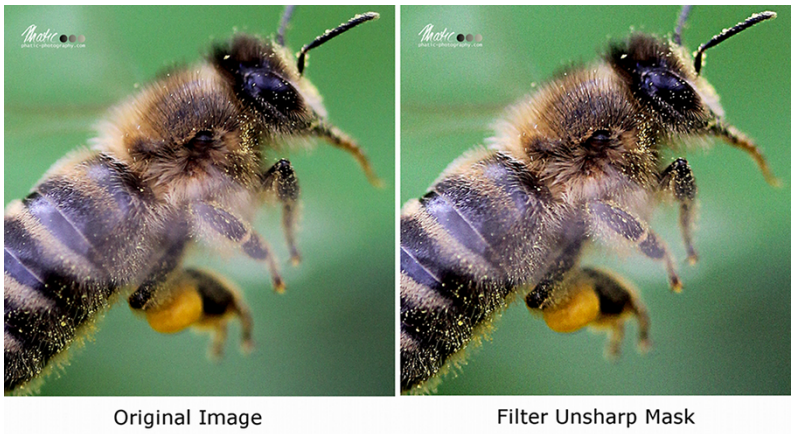
Ortanca filtre; Uzaysal çözünürlüğü bozmadan, kopuk (bağımsız) nokta veya çizgi gürültülerini temizlemek için kullanışlıdır. Bu nedenle ikili (binary) gürültülerde başarılı olmasına rağmen Gaussian gürültüsünde kullanışlı değildir. Gürültü piksellerinin sayısı komşu piksellerin yarısına eşit veya daha fazla ise bu filtre pek başarılı çalışmaz. Medyan filtresinin uygulanışı **Şekil 5**'te gösterilmektedir.



Şekil 5. Medyan filtresi

2.2.4.4. Unsharp Filtresi

Resimdeki ayrıntıları, keskin geçişleri belirginleştirmek, bulanıklaştırılmış görüntülerdeki ayrıntıları yeniden ortaya çıkarmak için kullanılır. Endüstriyel ve askeri alanda, tıbbi çalışmalarda ve diğer birçok alanda yararlıdır. (**Şekil 6**)



Şekil 6. Unsharp filtresi

2.2.4.5. Motion Filtresi

Cismin hareketini görebilmemize yarar. Görüntüyü hareket esnasında çekilmiş gibi algılanmasına sebep olur. Açı (angle) değeri bulanıklığın yönüdür. +90 ve -90 arasında değişebilir. Mesafe (distance) değeri ise bulanıklığın yoğunluğu olup 1 ile 999 arasında değer alabilir. Motion filtresi **Şekil 7**'de görülmektedir.

Şekil 7. Motion filtresi

2.2.4.6. Blurring Filtresi

Blur, fotoğraflarda kontrastı azaltmak ve renk geçişlerinde oluşan parazitleri ortadan kaldırmak için kullanılır. Hafif bir bulanıklık efekti verir. Bilgisayarımızda oluşturduğumuz grafik çalışmalarımıza farklı bir yapı kazandırmak için de kullanılmaktadır. Blurring filtresi örneği **Şekil 8**'de ki gibidir. **[16]**

Şekil 8. Blurring filtresi

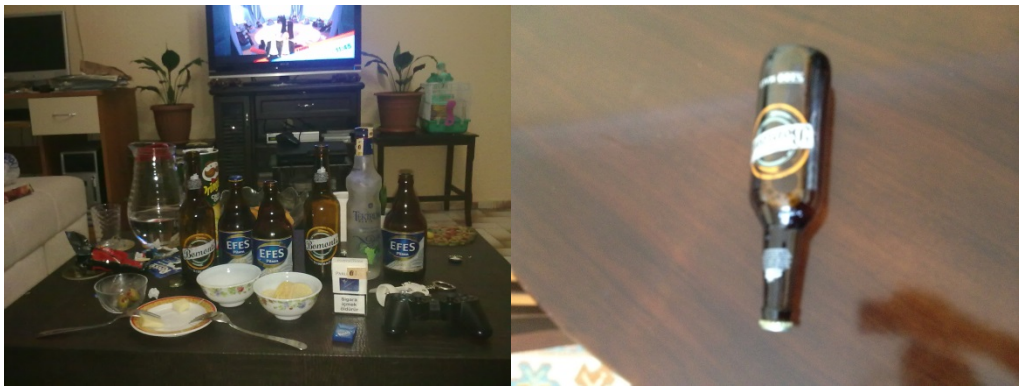
2.3. Kısıtlar

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inin sorunsuz halde işleyebilmesi için yapılan testler sonucunda bazı kısıtlar ortaya çıkmıştır. Bu kısıtlar üç nedenden dolayı meydana gelmektedir. Lokasyon, açı ve boyut olarak kısıtları isimlendirebiliriz.

2.3.1. Lokasyon

Görüntü işleme projelerinde lokasyon çok önemli bir kısıttır. Görüntülerde lokasyona bağlı olarak değişiklikler meydana gelebilmektedir. Örneğin; çok parlak bir resimde ya da gürültünün (noise) çok bulunduğu bir resimde ne kadar isteseniz de istediğiniz nesneyi tespit etmeniz çok zorlaşacaktır.

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inde kullanılan içki şişesinin bulunduğu ortama bağlı olmak kaydı ile tespit işlemi yapılamadığı görülmektedir. Örnek vermek gerekirse eğer şişe çok kalabalık bir ortamda bulunuyorsa ve şişe başka nesneler tarafından kesiliyorsa algılama işlemi yapılamadığı görülmektedir. Diğer bir durum olarak da görüntünün çekildiği lokasyon da ortamın aldığı ışıktan kaynaklı parlamalar ve renk bozulmalarının olmasından dolayı problemler meydana gelebilmektedir. Bu tarz durumlarda tezin işleyebilmesi için ön işleme tekniklerinin uygulanması gerekmektedir. Görüntüler üzerinde düzenleme yapılarak işleme hazır hale getirilmelidir. (**Şekil 9**)



Şekil 9. Lokasyon kısıtı

2.3.2. Açı

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inde açıya bağlı tespit işlemi yapılamadığı durumlar gözlenmiştir. Bu durumların başlıca sebebi uygulamanın nesne tespiti yapmasını sağlayan xml dosyasının modellemesiyle alakalı olduğu tespit edilmiştir. Modellenecek nesne olarak bir içki şişesi seçildiği için problem meydana gelmektedir. Şişe homojen ölçülere ait bir yapıya sahip değildir, taban kısmı oval ve geniş bir yapıya sahipken, üst kısmı dar bir yapıdadır. Bununla beraber nesne tespiti, şişenin boyun bölgesinden yapılmaktadır. Modelleme sırasında boyun kısmına ait ölçüler alınmaktadır ve o şekilde modellenmektedir.

Şişenin düz veya ters şekilde durduğu resimlerde herhangi bir nesne tespitiyle alakalı problem yaşanmazken, şişe yan veya çapraz konumlara geçtiğinde nesne tespiti yapılamamaktadır. Eğer yatay konumda ki şişe resmi 90 derecelik açıyla dik konuma getirildiğinde nesne tespiti yapılabilmektedir. Bu durumun en önemli sebebi modelleme işleminin haar sınıflayıcısı metodu kullanılarak yapılmasıdır. Çünkü modelleme işleminde koordinat mantığı kullanılmaktadır. Ön işleme teknikleriyle birlikte açıların değiştirilmesi gerekmektedir. (**Şekil 10**)

Şekil 10. Açı kısıtı

2.3.3. Boyut

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inde uygulama testlerinde diğer bir kısıt olarak görüntüde tespit edilmek istenilen nesnenin boyutuyla alakalı problemler olduğu görülmüştür. Nesne boyutunun bütün görüntü boyutunun yüzde 70'ini kaplaması durumunda sorunlar meydana geldiği görülmektedir. Bu durumda nesne görüntüde tespit edilememektedir. (**Şekil 11**)

Şekil 11. Boyut kısıtı

2.4. Program

2.4.1. Yazılımın Oluşturulması

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tezi C++ yazılım dili kullanılarak Microsoft Visual Studio derleyicisinde yazılmıştır. Mevcut bir görüntüden objeler işleneceği için OpenCV kütüphanesine ihtiyaç duyulmuş olup, projeye entegre edilerek kullanılmıştır.

OpenCV kütüphanesi proje entegre edildikten sonra tezin amaçları kapsamında konuyla alakalı araştırmalar yapılarak gerekli metotlar saptanmıştır.

Yapılan araştırmalar sonucunda tezde kullanılmak üzere en uygun olan metot Cascade sınıflayıcı olarak tespit edildikten sonra program için gerekli olan kod tasarlanmıştır.

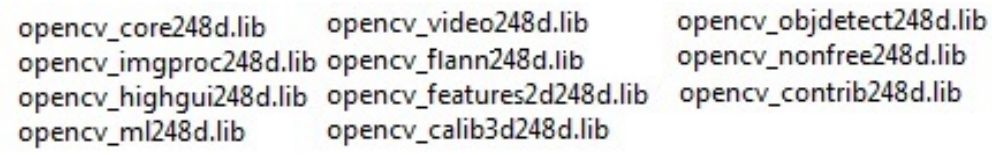
Şişenin saptanması için programda xml dosyasına ihtiyaç duyulmaktadır. Tespit işlemi için yüklenmiş olan resimle xml dosyasında tanımlanmış olan nesne birbirleriyle eşleşmek zorundadır, bundan dolayı proje kapsamında daha önceden belirlenen içki şişesine ait bir xml dosyası oluşturulmuştur.

Kod derlenmiştir ve ortaya çıkan hatalar giderildikten sonra çalıştırılmıştır ve amacına uygun sonuç alınmıştır.

2.4.2. OpenCv İçin Gerekli Ayarların Yapılması

Projenin çalışması için OpenCV kütüphanesinin entegre edilmesi gerekmektedir. Daha önceden projenin tasarlandığı bilgisayara OpenCV 2.4.8 versiyonu yüklenmiştir.

OpenCV 2.4.8 in kütüphane dosyaları hem projeye entegre edilmiştir hem de bilgisayarın system32 klasörüne eklenmiştir. Eklenen kütüphane dosyalarının listesi **Şekil 12**'de gösterilmektedir.



Şekil 12. OpenCV 2.4.8 kütüphane dosyaları

Gerekli kütüphane dosyaları projeye eklendikten sonra OpenCV, Microsoft Visual Studio derleyicisinde çalışır duruma gelmiştir.

2.4.3. Yazılım Kodunun Açıklanması

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inde kullanılan kodların işlevleri bu bölümde açıklanmaktadır. Yazılım kodu dört bölümde anlatılacaktır.

Birinci bölümde; programda kullanılacak olan xml'in dosyası belirtilmiş olan konumdan çağrılarak entegre edilmiştir. Daha sonra resmin eklenebilmesi için depolama alanı yaratılmıştır. Yaratılan depolama alanına belirtilmiş olan konumdan üzerinde nesnelerin tespit edileceği resim eklenmiştir. [**Şekil 13]**

```
int main( int argc, char** argv )
        {

            string RootCascade = "C:/Program Files/opencv/sources/data/haarcascades/";
            string Extension = ".xml";
            string filename = RootCascade + "bottleneck" + Extension;
            cascade = ( CvHaarClassifierCascade* )cvLoad( filename.c_str());

            storage = cvCreateMemStorage(0) ;

            IplImage* inImage = cvLoadImage("C:/Users/NightDriveN/Documents/Visual Studio 2010/Projects/opencv_test/MyImages/x001.jpg",1);
```

Şekil 13. Xml ve resmin eklenmesi

İkinci bölümde; xml ve resim yükleme işlemleri tamamlandıktan sonra programda kontrol işlemleri yapılmaktadır. **Şekil 14** 'te görüntülenen kod parçasında öncelikle Cascade yani xml dosyası, devamında depolama kısmı ve son kısmında da resmin yüklenip yüklenmediği kontrol edilmektedir.

```
if( !cascade  || !storage || !inImage)
    {
        printf("Initialization Failed: %s\n",
            (!cascade)? " Cascade file not found !\n":
            (!storage)?  " Not memmory allocated or not enough memory !\n":
            (!inImage)? " The input file can not be found !\n": "");

        system("pause");

        return 0;
```

Şekil 14. Kontrol bölümü

Üçüncü bölümde; sansürlenmek istenilen nesnenin resimden tespit edilmesi ve ekranda gösterilmesi sağlanmaktadır. [**Şekil 15]**

```
        detectbottles( inImage );
        cvShowImage("Detection",inImage);
        cvReleaseImage( &inImage);
        cvWaitKey(0);

        cvReleaseHaarClassifierCascade( &cascade );
        cvReleaseMemStorage( &storage );

        return 0;
    }
```

Şekil 15. Nesnenin resimden tespit edilmesi ve ekranda gösterilmesi

Dördüncü bölümde; programda belirlenmiş olan nesnenin koordinatlarının tespit edilmesi ve istenilen düzeyde sansürleme işlemi uygulanmıştır. Gaussian filtresi kullanılarak sansürün derecesi ayarlanmıştır. Kodun bu parçasında nesnenin sağlıklı olarak tespitini kontrol etmek için pasif halde bir kısım daha bulunmaktadır ve bu kodla birlikte nesnenin çevresinin çizimle gösterilmesine olanak sağlanmıştır. [**Şekil 16]**

```
void detectbottles( IplImage *newframe)
        {
            CvSeq *bottles = cvHaarDetectObjects( newframe, cascade, storage,
                                                  1.15, 5,
                                                  0,
                                                  cvSize( 30, 30 ) );

            for( int i = 0 ; i < ( bottles ? bottles->total : 0 ) ; i++ )
                {
                    CvRect *r = ( CvRect *)cvGetSeqElem( bottles, i );

                    /*cvRectangle(newframe,
                                cvPoint( r->x, r->y ),
                                cvPoint( r->x + r->width, r->y + r->height ),
                                CV_RGB( 0, 255, 0 ), 2, 8, 0 );*/

                                Mat mat(newframe);
                                Mat rio(mat, *r);
                                Mat out;
                                blur(rio, rio, Size(45, 45));

                }
        }
```

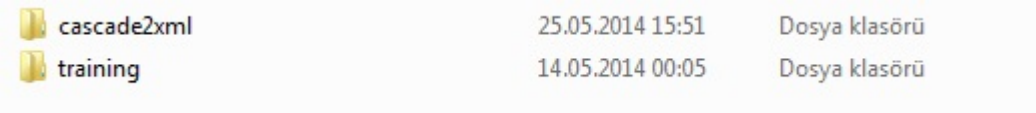
Şekil 16. Nesnelerin tanımlanması ve sansürlenmesi

2.4.4. XML Dosyasının Oluşturulması

Tezde nesnenin tespit edilebilmesi için xml dosyasına ihtiyaç duyulmaktadır. Xml dosyası projeye eklenen resimle Cascade sınıflayıcısı ile eşleşerek nesneyi tespit etmektedir.

Tezde sansürlenecek resim olarak bir içki şişesi kullanılmasına karar verilmiştir. Belirtilen şişenin modellenmesine ihtiyaç vardır. Xml dosyası oluşturmak için Haar sınıflayıcısı aracı kullanılmıştır. **[18]**

Öncelikle araçta sınıflama kısmında işlemler yapılmıştır. [**Şekil 17]**

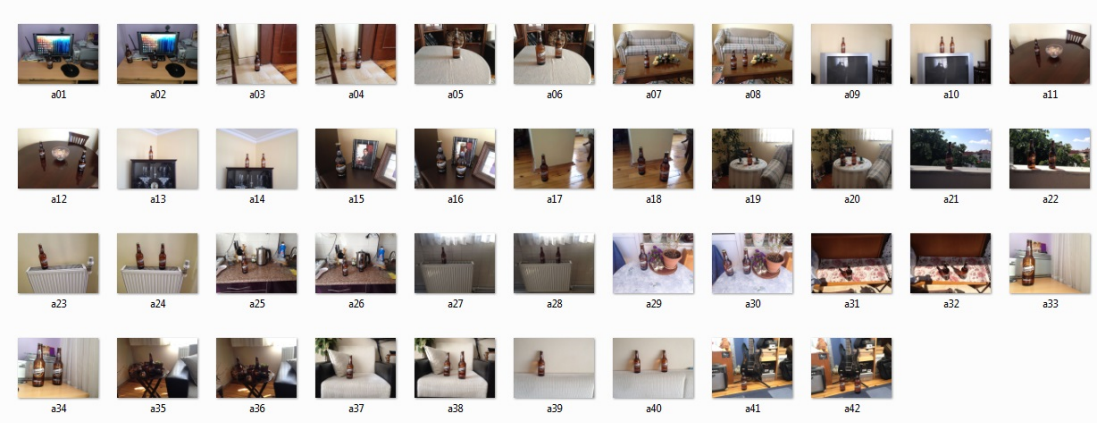


Şekil 17. Haar sınıflayıcı aracı

Sınıflama kısmında ilk olarak negatif klasörüne daha önceden oluşturulmuş olan resimler eklenmiştir, resimler xml dosyası oluşturulurken arka plan ile alakalı görüntü örneği olması için eklenmiştir. Kopyalama işleminin ardından create_list.bat dosyası çalıştırılarak bg.txt dosyasına kayıt edilmiştir. [**Şekil 18]**

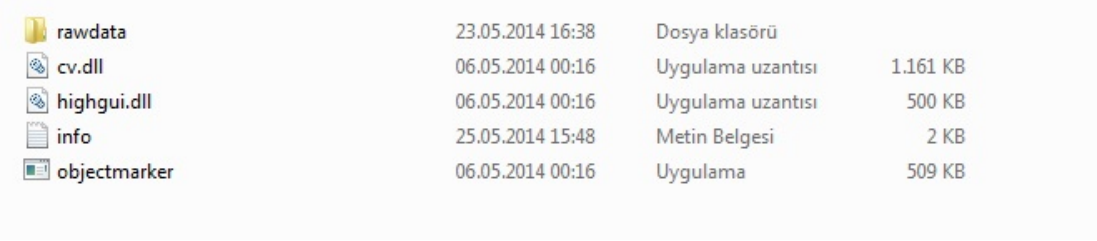
Şekil 18. Negatif resimler ve dosya görünümü

Arka plan işleminden sonra modellemenin yapılacağı olan kısım olan pozitif klasörünün içinde bulunan rawdata klasörüne bmp formatlı resimler eklenmiştir. [**Şekil 19]**



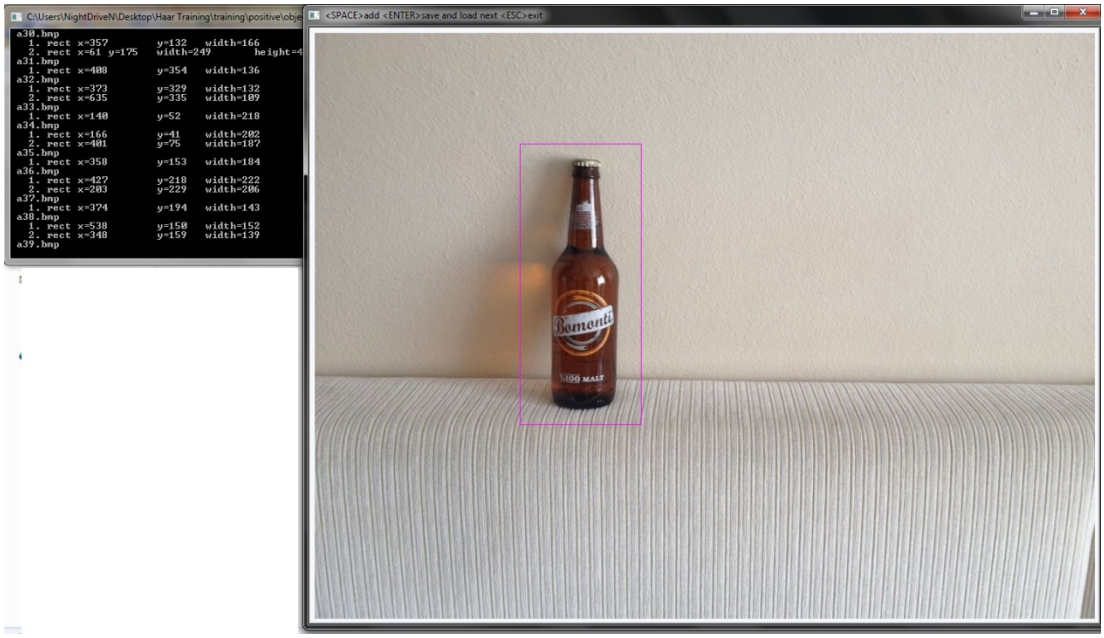
Şekil 19. Pozitif resimler

Resimler eklendikten sonra objectmarker.exe dosyası çalıştırılarak modelleme işlemi yapılmaya başlanmıştır.



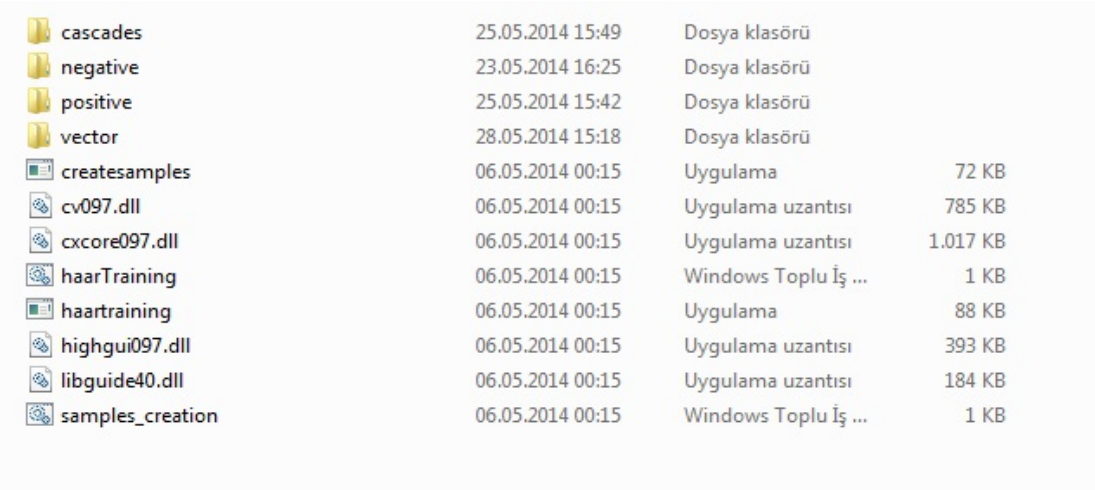
Şekil 20. Pozitif klasörü

Objectmaker.exe dosyası ile daha önceden rawdata klasörüne eklenen dosyalar tek tek ekrana gelmektedir ve ekranda modellenmesi istenilen cisim üzerinde işaretleme yapılmaktadır. Araçta her noktalar belirlendikten sonra escape tuşuna basılmıştır cisminle alakalı bilgiler ekranın sol tarafındaki pencereye eklendikten sonra enter tuşuna basılarak sıradaki resme geçilmiştir. Resimler bittikten sonra bilgiler klasör içerisinde bulunan info.txt dosyasına kayıt edilmiştir. [**Şekil 21]**



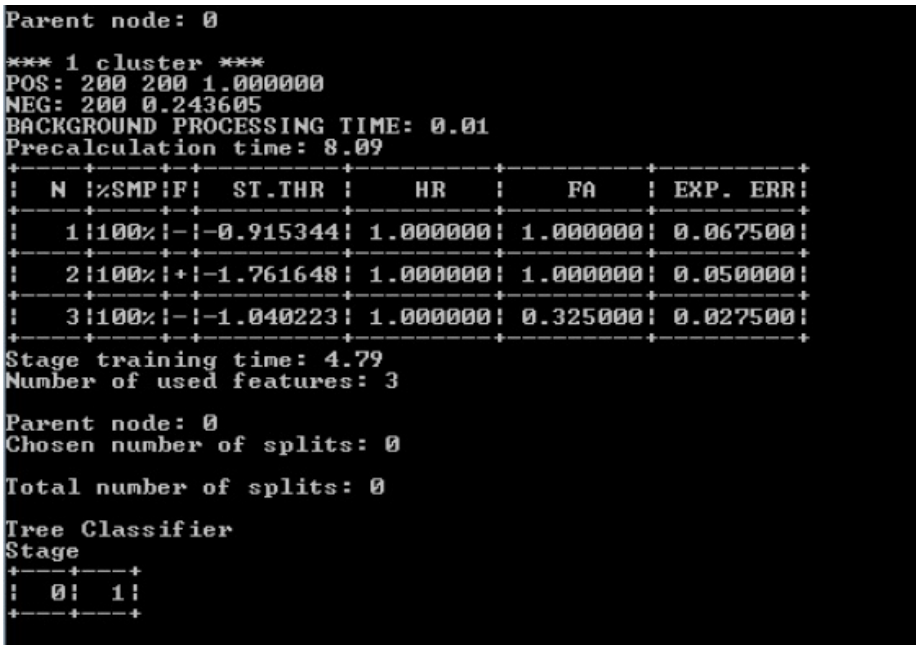
Şekil 21. Objectmaker.exe çalışması

Pozitif işlemlerinden sonra samples_creation.bat dosyası çalıştırılarak vektör klasörü altında bottlevector isimli dosya oluşturulmuştur. Daha önceden oluşturulan bilgiler tek dosya haline getirilmiştir. [**Şekil 22]**



Şekil 22. Sınıflayıcı klasörü görümü

Vektör işleminin ardından haartraining.bat dosyası çalıştırılmıştır ve Haar sınıflayıcısına uygun veriler cascades klasöründe elde edilir. [**Şekil 23]**

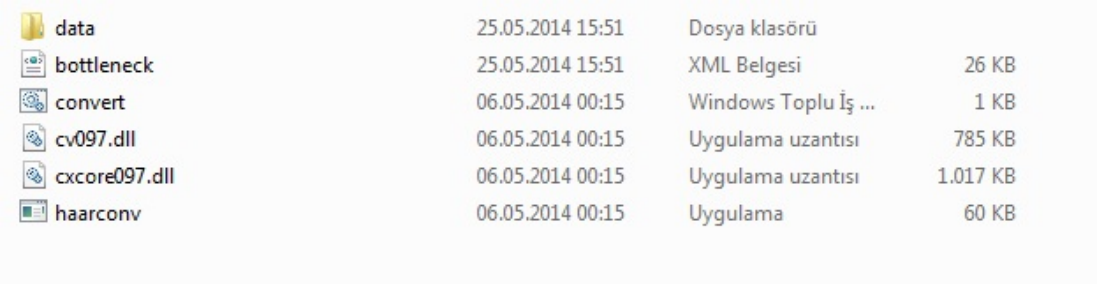


Şekil 23. Haartraining.bat çalışması

Cascades klasörü içerisinde elde edilmiş olan veriler kopyalanarak cascade2xml isimli klasörün altında bulunan data klasörüne yapıştırılmıştır. [**Şekil 24]**

Şekil 24. Cascades veri dosyaları

Dosyalar belirtilen dosyaya yapıştırıldıktan sonra convert.bat dosyası çalıştırılmıştır ve oluşturulmak istenen xml dosyası bottleneck ismiyle oluşturulmuştur. [**Şekil 25]**



Şekil 25. Cascade2xml klasörü görünümü ve bottleneck.xml dosyası

2.4.5. Arayüz

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inin yazılımı istenilen düzeyde çalışır hale getirildikten sonra uygulamanın bir arayüz ihtiyacı oluşmuştur. Arayüz; tez yazılımının kodlarla uğraşılarak çalıştırılması zorunluluğunu ortadan kaldırmak amaçlı ve kullanıcı dostu olması için olabilecek en basit şekilde yazılmıştır.

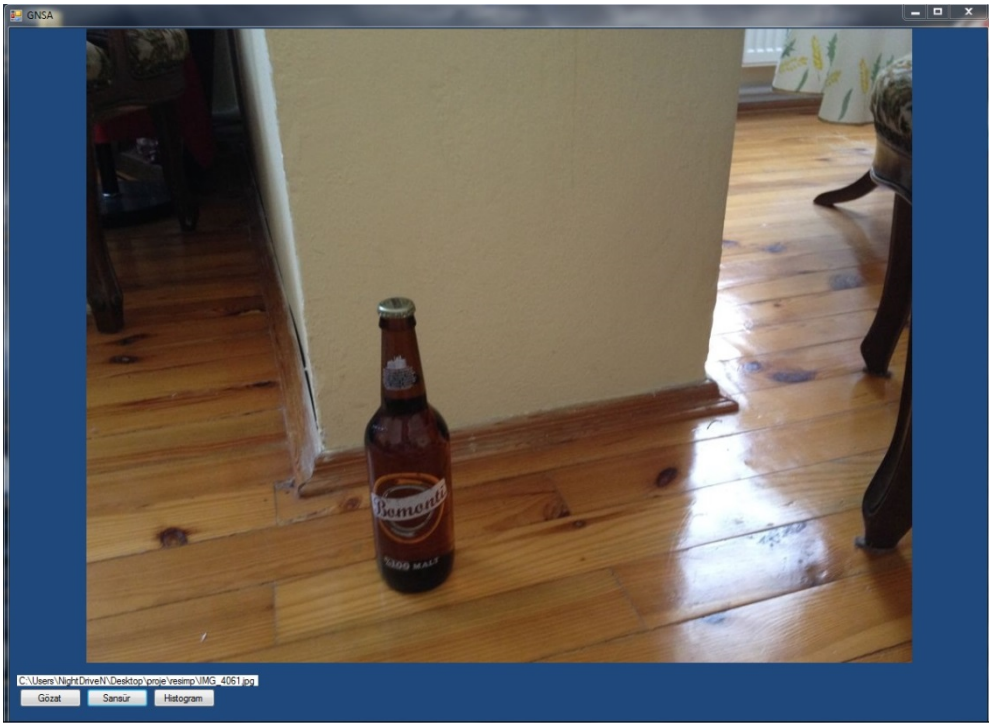
GİŞBA uygulaması tıklandıktan sonra karşımıza **Şekil 26**'da göründüğü gibi açılmaktadır.

Şekil 26. GİŞBA arayüzü açılış görüntüsü

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inin uygulaması üç farklı işlevden meydana gelmektedir. Gözat tuşu tıklanarak bilgisayardan üzerinde işlem yapılacak görüntü seçilebilmektedir. **Şekil 27**'de gösterilmektedir.

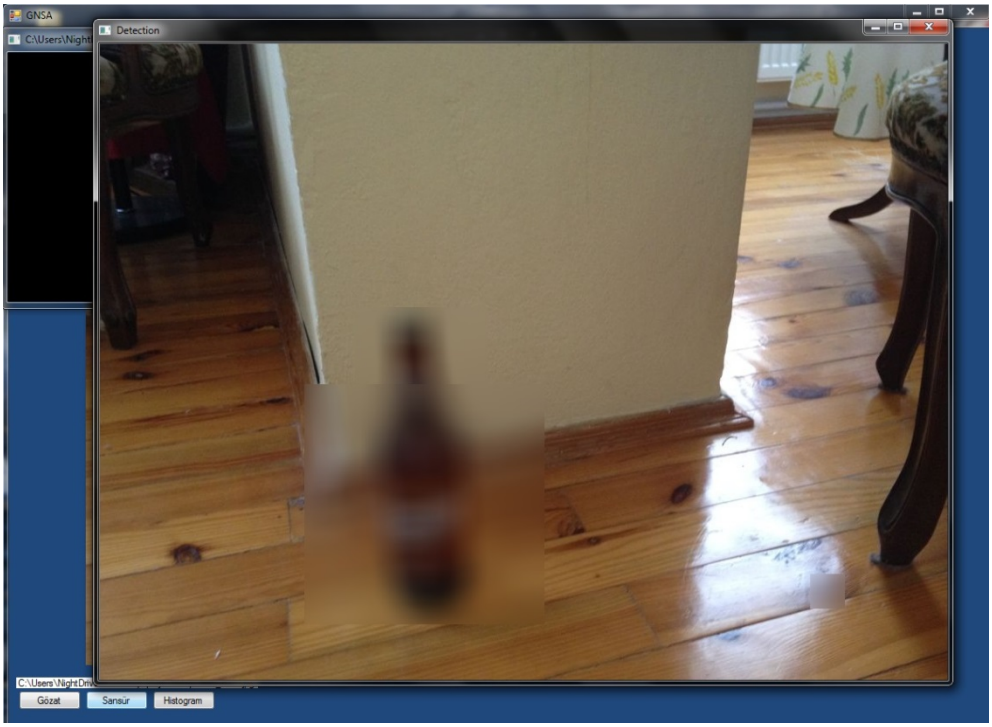
Şekil 27. GİŞBA gözat penceresi

Görüntü seçildikten sonra arayüz üzerinde gösterilmektedir. (**Şekil 28**)



Şekil 28. Arayüzden seçilen resim görünümü

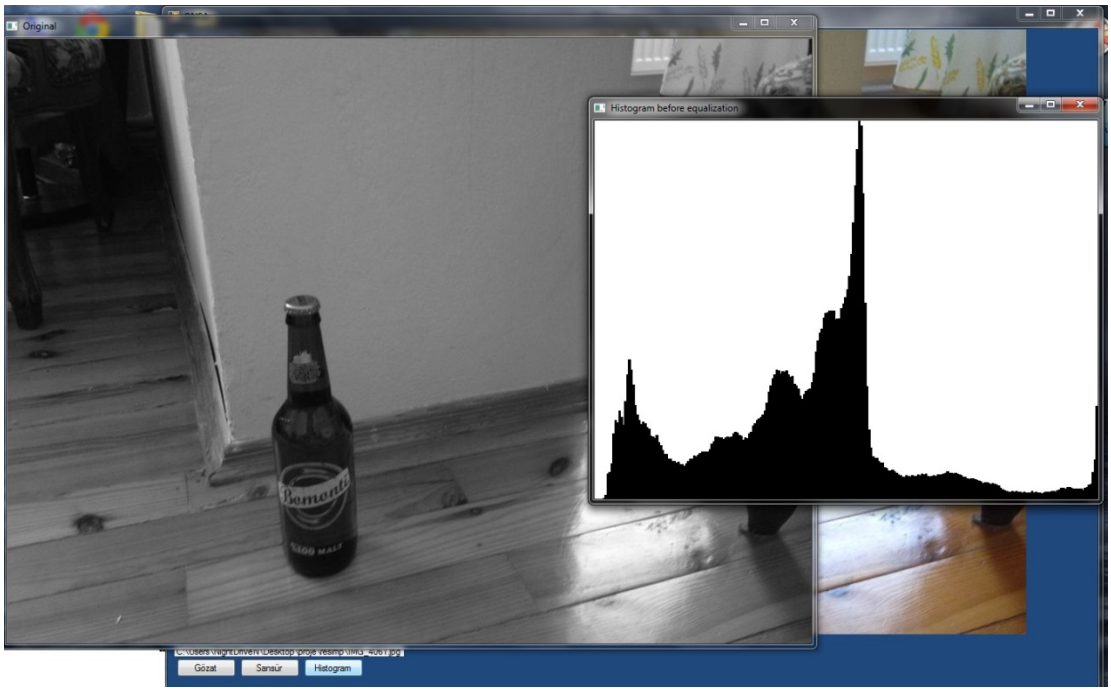
Arayüzden üzerinde işlem yapmak istenilen görüntü seçildikten sonra eğer sansürleme işlemi yapılacak ise, sansür tuşuna basılmaktadır. Ve ardından **Şekil 29**'da görüldüğü gibi görüntünün sansürleme işlemi gerçekleşmiş halinin olduğu pencere ekranda açılmaktadır.



Şekil 29. GİŞBA sansür işlemi

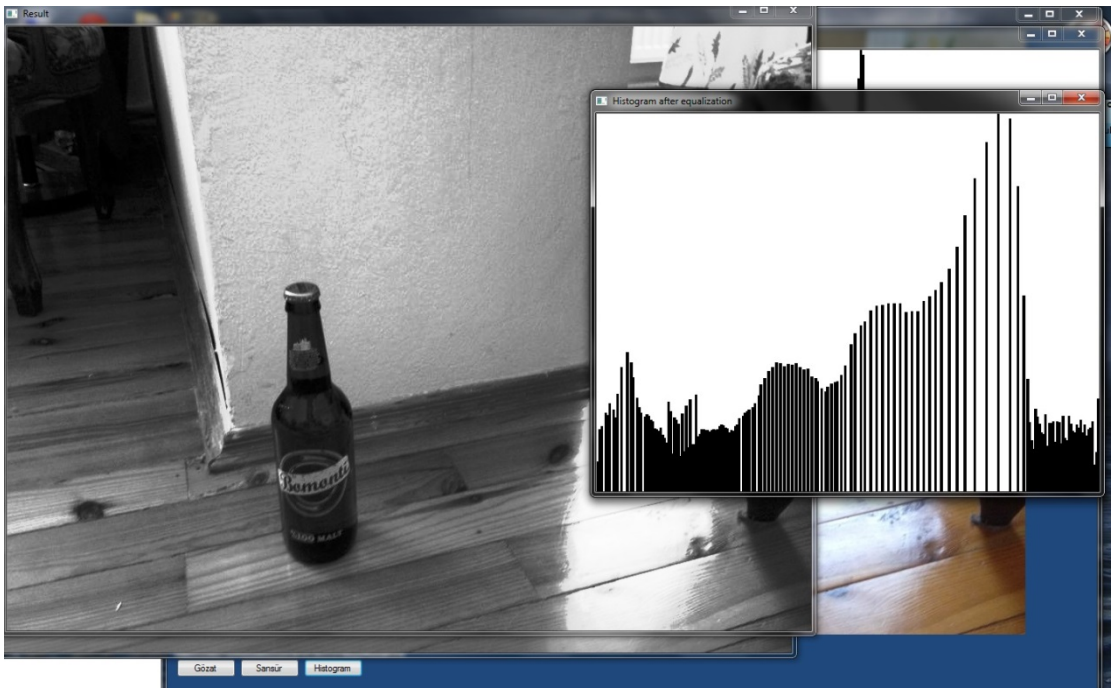
Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inin uygulamasının diğer bir işlevi olan histogram eşitleme işlemini yapmak için; arayüzden işlem yapılmak istenilen resim seçilir ve daha sonra histogram tuşuna basılmaktadır.

Histogram tuşuna basıldıktan sonra öncelikle resmin orijinal gri hali ekranda açılan pencerede gösterilmektedir ve orijinal duruma ait histogram grafiği ekranda gösterilmektedir. (**Şekil 30**)



Şekil 30. Orijinal gri görüntü

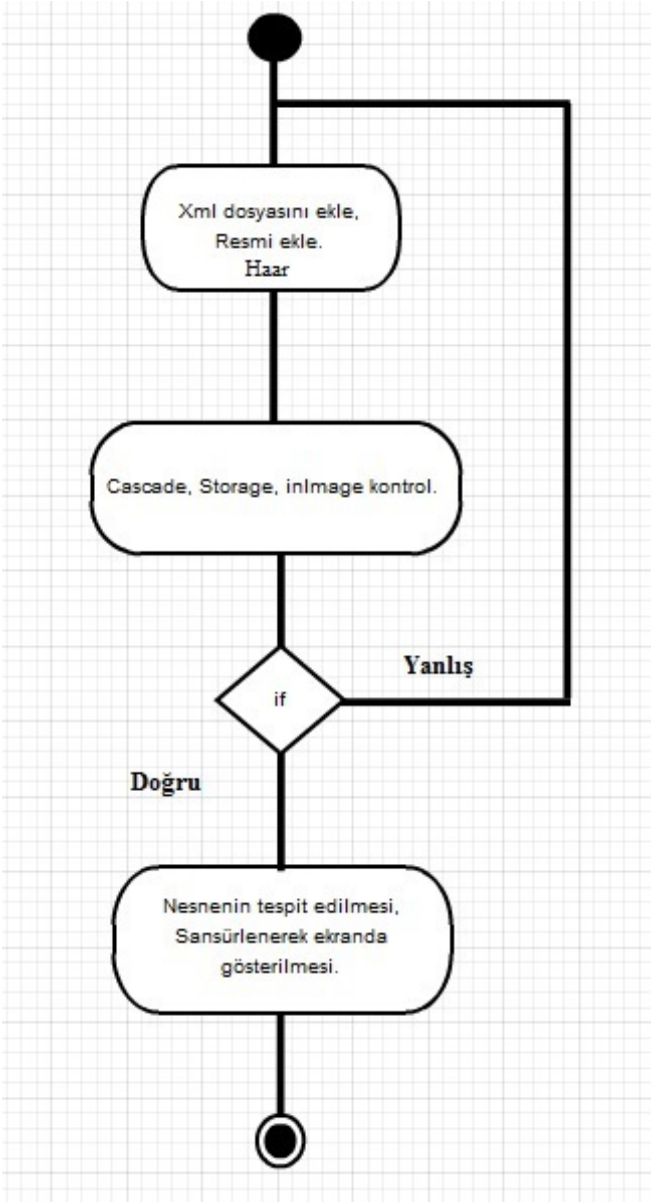
Bu işlemlere paralel olarak histogram eşitleme işleminin gerçekleştiği görüntü ve grafiğin olduğu pencerelerde ekranda açılarak gösterilmektedir.**(Şekil 31**)



Şekil 31. Histogram eşitleme işlemi

2.4.6. Aktivite Diyagramı

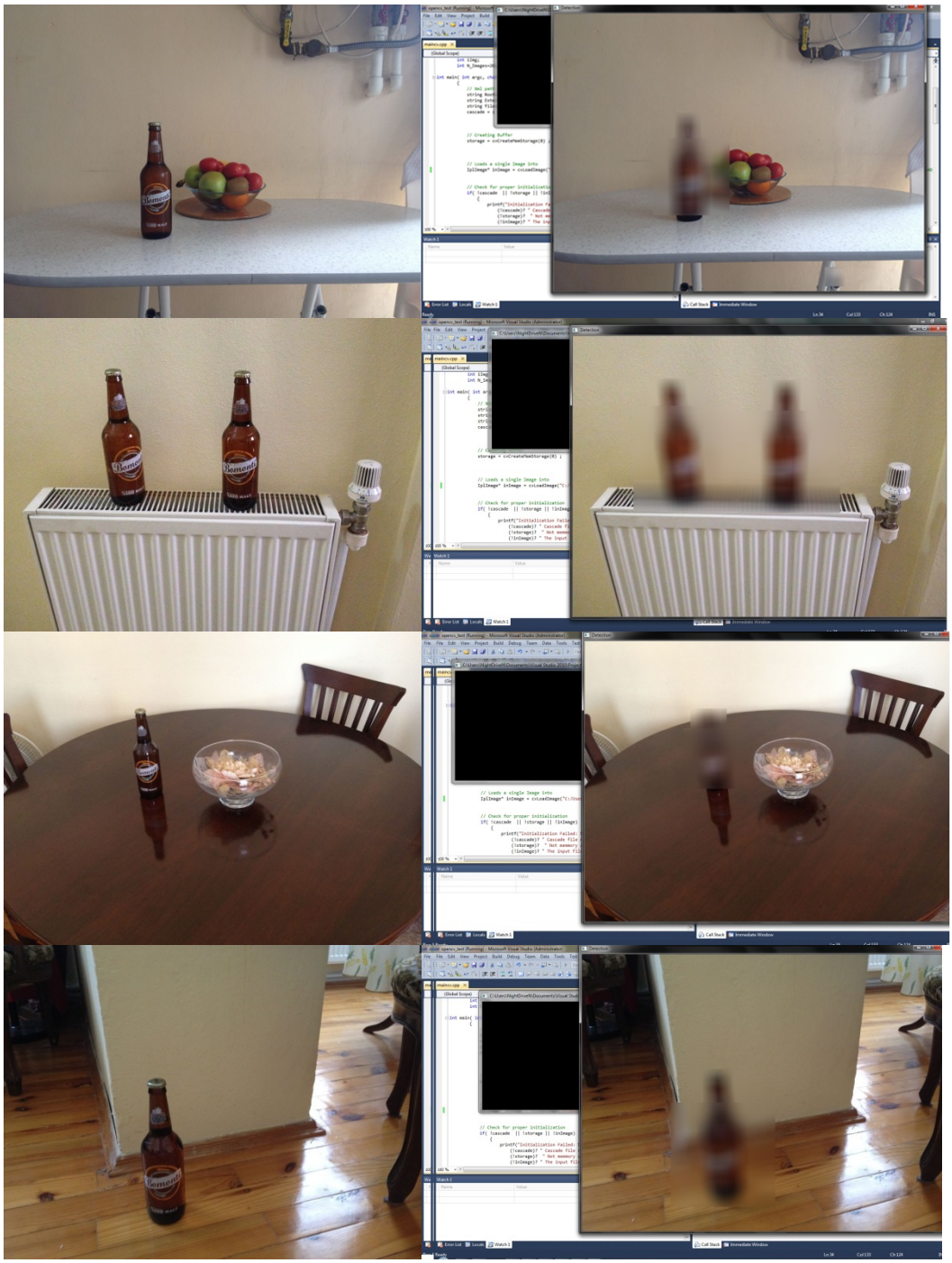
Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inin aktivite diyagramı **Şekil 32** 'da göründüğü gibidir.



Şekil 32. Aktivite diyagramı

2.4.7. Çalıştırılması ve Test Edilmesi

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inde değerlendirilmesi istenilen resmin bulunduğu dosyanın konumu belirtilmiştir. Proje derlenip çalıştırılmıştır. Farklı resimler kullanılarak proje test edilmiştir. [**Şekil 33]**



Şekil 33. Farklı resimlerle projenin test edilmesi

3. DENEYSEL ANALİZ

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inin uygulamasında içinde çeşitli resimler barındıran bir veri seti kullanılarak deneysel analiz işlemi yapılmıştır.

Şişe Durumu	**Adet**	**Doğru Tanıma Oranı**	**Bulanıklaştırma Oranı**	**Sonuç**
Bomonti Şişe	50	%98	%91	Olumlu
Kalabalık Ortamda Bomonti Şişe	50	%73	%70	Olumlu
Bomonti Şişe Farklı Nesneli Ortamda	50	%85	%82	Olumlu
Bomonti Şişe Dik Açı	50	%98	%96	Olumlu
Bomonti Şişe Ters Açı	50	%98	%96	Olumlu
Bomonti Şişe Yan Açı	50	%5	%2	Olumsuz
Farklı Marka Bira Şişesi	50	%20	%16	Olumsuz
Farklı İçki Şişesi	50	%10	%8	Olumsuz
Meşrubat Şişesi	50	%12	%9	Olumsuz
Şişesiz Ortam	50	%0	%0	Olumsuz
Farklı Nesneli Ortam	50	%2	%1	Olumsuz
Bomonti Şişe Net Olmayan Görüntülerde	50	%8	%4	Olumsuz
Toplam	**600**	**%42**	**%39**	**5/10**

Şekil 34. Deneysel analiz tablosu

SONUÇ

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tezi gerekli kodlama işlemleri ve xml tanımlamaları yapıldıktan sonra derlenip, çalıştırılmıştır. Tezin uygulamasında meydana gelen hata ve problemler testler sonucu ortaya çıkarılarak giderilmiştir. Ayrıca rahat kullanılabilmesi için bir arayüz tasarlanmıştır.

Uygulama da bazı problemler görülmüştür, bu problemlerin lokasyon, açı ve boyut kaynaklı olduğu tespit edilmiştir. Nesnenin ortamdan kaynaklı bir şekilde tespit edilememesi durumu, kullanılan nesnenin duruş açısından dolayı tespit edilmemesi ve seçilen görüntüde nesnenin bütüne oranla belirli bir yüzdeden daha büyük olma durumundan kaynaklı tespit edilememe durumları oluşmuştur. Problemlerin birçoğu ön işleme yapılarak çözülmeye çalışmıştır. Gerekli düzenlemeler yapılıp hata payının en az seviyeye indirilmesi amaçlanmıştır.

Xml dosyasında mevcut olarak tanımlanan içki şişesinin sorunsuz olarak tespit edildiği görülmüştür. Yapılan koordinat belirleme işleminden sonra elde edilen noktalardan itibaren sansürleme işlemi yapıldığı görülmüştür. Sansürlemenin derecesini belirlemek için Gaussian filtreleme kullanılmıştır.

Uygulamanın test safhasında; çeşitli mekânlarda ve durumlarda çekilmiş resimler kullanılarak nesnenin tespit edilmesi sınanmıştır. Sonuç olarak mekâna ve duruma fazla bir bağlılığı olmadığı ortaya konmuştur.

Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tez'inde bulunan xml dosyasının içindeki nesne türünün çeşitlendirilmesiyle birlikte tespit edilme kapsamı artırılmasına olanak sağlamaktadır. Yazılım üzerinde çeşitli değişiklikler yapılarak video gibi hareketli ortamlarda da tespit işlemi yapılabilmesi amaçlanabilmektedir. Görüntülerde İstenmeyen Şişe Bulanıklaştırma Algoritması Tezi geliştirilmeye açık bir tezdir.

KAYNAKÇA

[1] Arif Erdal Taşçı, Akciğer tomografileri kullanılarak yapay zeka ve görüntü işleme tekniklerine dayalı otomatik nodül bölge tespit yöntemi geliştirilmesi, Ege Üniversitesi, 2013.

[2] Muammer Türkoğlu, Otomatik kan hücrelerinin tanınması ve sınıflandırılmasında değişmez momentlere dayalı görüntü işleme yöntemlerinin kullanılması, Fırat Üniversitesi, 2013.

[3] Halit Çetiner, Görüntü işleme teknikleri kullanarak optik karakter tanımlama, Süleyman Demirel Üniversitesi, 2012.

[4] Veli Burak Çelen, Sabit görüntülerde görüntü işleme teknikleri ile orman yangını tespiti, TOBB Ekonomi ve Teknoloji Üniversitesi, 2012.

[5] Mehmet Karakoç, Görüntü işleme teknikleri ve yapay zeka yöntemleri kullanarak görüntü içinde görüntü arama, Pamukkale Üniversitesi, 2011.

[6] Orhan Er, Görüntü işleme teknikleri kullanarak elma tasnifleme, Süleyman Demirel Üniversitesi, 2011.

[7] Rajab Davudov, Haber videolarının görüntü işleme yöntemleri ile haberlere bölütlenmesi, Yıldız Teknik Üniversitesi, 2009.

[8] Murat Alçın, Görüntü işleme esaslı parmak izi doğrulama, Marmara Üniversitesi, 2009.

[9] Atınç Yılmaz, Kamera kullanılarak görüntü işleme yoluyla gerçek zamanlı güvenlik uygulaması, Haliç Üniversitesi, 2007.

[10] Hasan Yakut, İşaret dili harflerinin görüntü işleme yöntemleri ile tanınması için bir uygulama, Fırat Üniversitesi, 2013.

[11] Mustafa Filizci, Mekatronik teknolojisinde görüntü işleme tekniklerine dayalı yüz tanıma sistemi geliştirilmesi, Marmara Üniversitesi, 2011.

[12] Erkan Horozoğlu, Görüntü işleme ile yüzey pürüzlülüğü ölçümü ve analizi, Selçuk Üniversitesi, 2013.

[13] Erişti Ezgi, Görüntü İşlemede Yeni Bir Soluk, OPENCV, Akademik Bilişim'10 - XII. Akademik Bilişim Konferansı Bildirileri, Muğla Üniversitesi, 10 - 12 Şubat 2010.

[14] Vikipedi Özgür Ansiklopedi, C++ Programlama Dili, 2014, http://tr.wikipedia.org/wiki/C%2B%2B, (Erişim Tarihi: 14 Ağustos 2014)

[15] Bahri Abacı, Histogram Eşitleme, 25 Haziran 2012,

http://www.cescript.com/2012/06/histogram-saysal-bir-resim-icerisinde.html, (Erişim Tarihi:10 Ağustos 2014).

[16] Bülent Siyah, Görüntü Filtreleme, 6 Mart 2012, http://www.bulentsiyah.com/goruntu-filtreleme-uygulamalari-ve-amaclari-matlab/, (Erişim Tarihi: 15 Temmeuz 2014).

[17] OpenCV, Open Source Computer Vision, http://www.opencv.org

[18] Rezaei Mahdi, Creating a Cascade of Haar-Like Classifiers, University of Auckland, 2014.

[19] Bradski Gary, Kaehler Adrian, Learning OpenCV, O'Reilly Media, Sebastopol CA, September 2008.

EKLER

a) Projede kullanılan yazılım programları:

- Microsoft Visual Studio 2010 (Visual C++)
- İrfanview (Resim ölçeklendirme ve renklendirme)
- Haar Training Tool (Xml dosyası oluşturma)
- OpenCV 2.4.8 (Open source computer vision library)

b) Proje kitapçığıyla birlikte teslim edilecek DVD'nin içeriği:

- Bitirme projesi kitapçığı (doc ve pdf formatlarında)
- Proje sunum dosyası (pps ve pdf formatlarında)
- Proje yazılım dosyaları
- Proje resimleri ve videoları

ÖZGEÇMİŞ

Selim Can TEMELLİ

14 Nisan 1988 tarihi, İstanbul ili Kadıköy ilçesi doğumluyum. İlköğretim ve liseyi aynı ilçede tamamladıktan sonra, Beykent Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümüne kayıt oldum. İngilizce dil eğitimi aldım. Stajlarımı Türk Telekomünikasyon A.Ş. ve İnnova Bilişim Çözümleri A.Ş yaptım. Bu bölümden 2012 yılında mezun oldum. Lisans eğitimimi tamamladıktan sonra 2012 yılının Eylül ayında Beykent Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümüne kayıt olarak Yüksek Lisans eğitimime başladım. Halen eğitimimi sürdürmekle beraber son sınıf öğrencisiyim.