BLM 102- Algoritmalar ve Programlama I
2019-2020 Bahar Dönemi

Görüntü İşleme Yöntemleri ile Üretim Hattı Analizi

Proje Teslim Raporu
10 Haziran 2020

Ömer Sait Yorulmaz, Ahmet Hazar Haspolat, Mustafa Berk Taşkın,
Alp Yılmaz

Öğretim Elemanı: Dr. Vecdi Emre Levent, emre.levent@fbu.edu.tr,

İÇİNDEKİLER

1 GİRİŞ 1
1.1 Projenin Amacı 1
1.2 Proje Ekibi 1
2 GELİŞTİRİLEN UYGULAMA 2
2.1 Kullanılan Araçlar 2
2.1.1 Visual Studio Community 2
2.2 Tasarım 2
2.2.1 İş Akışı 2
3 SONUÇLAR 4

1 GİRİŞ

Bu dokümanda ‘Analysis of Production Line’ isimli projenin amacı, kapsamı, yapım aşamaları ve elde edilen sonuçlar detaylandırılmıştır.

1.1 Projenin Amacı

Fabrikalarda üretim hatlarından çıkan ürünlerin analizi; yüksek çözünürlüklü ve yüksek FPS (Frame per Second) özellikli kameralar ile yapılmaktadır. Bir üretim hattından çıkan üç farklı ürünün, video kaydı boyunca kaçar adet üretildiğinin tespit edileceği bir sistem geliştirilecektir.

1.2 Proje Ekibi

ÖMER SAİT YORULMAZ – 190301025

Fenerbahçe Üniversitesi Bilgisayar Mühendişliği öğrencisi olan Yorulmaz, aynı zamanda 2 yıldır özel bir şirkette software specialist olarak görev yapmaktadır.

AHMET HAZAR HASPOLAT – 190301012

Çamlıca Doğa kolejinde lise eğitimi tamamlayan Haspolat, daha sonrasında Fenerbahçe Üniversitesi Bilgisayar Mühendisliğine girmeye hak kazanarak, kariyerine bilgisayar bilimlerinde devam etmeyi planlamaktadır.

MUSTAFA BERK TAŞKIN – 190301021

Bursa Doğa kolejinden mezun olan Taşkın, Fenerbahçe Üniversitesi Bilgisayar Mühendisliğinde eğitim hayatına devam etmektedir.

ALP YILMAZ – 190301017

Celal Aras Anadolu lisesinde lise eğitimi tamamlayan Yılmaz, daha sonrasında Fenerbahçe Üniversitesi Bilgisayar Mühendisliğine girmeye hak kazanarak, kariyerine bilgisayar bilimlerinde devam etmeyi planlamaktadır.

2 GELİŞTİRİLEN UYGULAMA

2.1 Kullanılan Araçlar

Bu tasarım geliştirilirken Microsoft'un derleyicisi olan Visual Studio Community aracı kullanılmıştır.
Proje geliştirilirken bilgisayar görüşüne yönelik programlama fonksiyonları kütüphanesi olan OpenCv kullanılmıştır.

2.1.1 Visual Studio Community

Visual Studio, birçok programlama dilini kullanarak program, uygulama ya da web sitesi yapılabilen bir IDE yani entegre geliştirme ortamıdır. Microsoft Windows için bilgisayar programları, web siteleri, web uygulamaları, web hizmetleri ve mobil uygulamalar geliştirmek için kullanılır.

Visual Studio, Windows API, Windows Forms, Windows Presentation Foundation, Windows Store ve Microsoft Silverlight gibi Microsoft yazılım geliştirme platformlarını kullanır. Hem yerel kod hem de yönetilen kod üretebilir.

Visual Studio, IntelliSense'i (kod tamamlama bileşeni) ve kod yeniden düzenleme işlemini destekleyen bir kod düzenleyici içerir. Entegre hata ayıklayıcı; hem kaynak düzeyinde hata ayıklayıcı hem de makine düzeyinde hata ayıklayıcı olarak çalışır. Diğer yerleşik araçlar arasında bir kod profili oluşturucu, GUI uygulamaları oluşturmak için form tasarımcısı, web tasarımcısı, sınıf tasarımcısı ve veritabanı şeması tasarımcısı bulunur. Neredeyse her düzeyde işlevselliği artıran eklentileri kabul eder.

2.2 Tasarım

Projenin yapımında baz alınan veri videosunda, bir üretim hattından geçen ürünleri algılamak, sınıflandırmak ve saymak gerekiyor. Bu doğrultuda OpenCV kütüphanesinin akış boyunca detaylandırılacak bazı fonksiyonları ve algoritmaları kullanıldı.

2.2.1 İş Akışı

- Başlangıçta baz alınan video, framelere bölünerek bu frameler üzerinden aşağıdaki adımlar uygulandı.

- İlk olarak *cvtColor* isimli fonksiyon kullanılarak her bir frame'in renk tonlaması gri tabanlı olacak şekilde dönüştürüldü.

  (Cvtcolor: OpenCV'de birçok renk uzayı desteklenmektedir ve bunlar arasında dönüşüm yapılabilmektedir.)

- Sonrasında OpenCv'nin *GaussianBlur* fonskiyonu kullanılarak gri ile tonlanan frameler için blurlama işlemi yapıldı; frame'deki görüntünün gürültüsü azaltılıp keskin kısımlar yumuşaklaştırıldı.

  (GaussianBlur: Genellikle görüntü parazitini ve ayrıntıları azaltmak için grafik yazılımında yaygın olarak kullanılan bir efekttir.)

- Grileştirme ve blurlama işlemlerinden sonra frame içerisinde değişen kısımlar (üretim bandında kayan objeler) ve değişmeyen kısımlar (hareketsiz arka plan) *absdiff* fonksiyonu kullanılarak ayrıştırıldı.

  (absdiff: OpenCV de arka plan temizleme işlemini bu metot yapmaktadır. Verilen iki matris arasında çıkarma işlemi yapar bu çıkarma işlemi sonucunda değişen kısımlar yani hareketli kısımlar gösterilir. Örnek olarak şöyle özetleyebiliriz; kaynak olarak bir manzara resmi var, bu manzara resminden ikinci bir kare aldık ve bu ikinci karede bir kuş belirdi. Bizim amacımız arka planı temizlemek ise iki resim arasında çıkarma işlemi yapıyoruz. Kuş dışında kalan görüntü aynı olduğu için bu matris içerisinde ekstra olarak kuşun renkleri olacaktır ve bu bize sonucu verecektir.)

- Bu ayrıştırma sonucunda elde edilen gri tabanlı frame üzerinde *Thresholding* fonksiyonu kullanılarak objelerin görünürlüğü arttırıldı.

  (Thresholding: Giriş olarak verilen görüntüyü ikili görüntüye çevirmek için kullanılan bir yöntemdir. İkili görüntü (binary), görüntünün siyah ve beyaz olarak tanımlanmasıdır. Morfolojik operatörler gibi görüntü üzerindeki gürültüleri azaltmak veya nesne belirlemek gibi farklı amaçlar için kullanılır. Giriş olarak verilen görüntü üzerinde uygulanan thresholding tipine bağlı olarak, pikselleri verilen eşik değerine göre siyah ya da beyaz olarak günceller.)

- Daha sonra bazı objelerin görünürlüğünün arttırılması için *Dilation* fonksiyonu ile objelerin boyutları genişletildi. Bu sayede saptama (detection) işleminin gerçekleştirilmesi kolaylaştırıldı.

  (Dilation: Bu operatör giriş olarak verilen görüntü üzerinde parametreler ile verilen alan içerisindeki sınırları genişletmektedir, bu genişletme sayesinde piksel gurupları büyür ve pikseller arası boşluklar küçülür.)

- Objelerin boyutları genişletildikten sonra bir *Background Subtraction* algoritması olan *BackgroundSubtractorMOG2* kullanılarak hareketsiz kısım (arka plan) frame'den çıkarıldı ve hareket eden nesneler için frame tekrar oluşturuldu. Her piksel için uygun sayıda Gauss dağılımını seçmesi sebebiyle bu algoritma kullanıldı.

- Frameler üzerindeki manipülasyon bittiğinde arka plan kaldırılmış ve frame üzerindeki objelerin boyutları büyütülmüş oldu. Bu işlemler sonrasında *SimpleBlobDetector* kullanılarak dairesellik veya keskinlik gibi özellikler üzerinden obje saptama işlemi yapıldı.

  (*SimpleBlobDetector: Bir frame içerisinde ortak özellikleri paylaşan bağlı piksel grupları için belirtilen özelliklere göre sınıflandırma, saptama ve filtreleme işlemi yapar.)*

- Son olarak her bir frame üzerinden saptanan objeler, frame üzerindeki koordinatları ve boyutları üzerinden sınıflandırıldı. Belirli bir x-y koordinatında ve belirli bir boyutta olan cisim bilgisi alındığında saptanan objenin türü belirlendi ve video boyunca bu objelerden kaçar adet görüntülendiği hesaplandı.

- Detaylı bilgi için sunum videosunu inceleyebilir ve/ya kaynak koda bakabilirsiniz.

3 SONUÇLAR

Bu projenin tasarımı ve geliştirilmesi aşamasında öncelikle açık kaynaklı bir kütüphane ile çalışmanın gereksinimleri keşfedildi ve sonrasında bu gereksinimler üzerinden araştırma, deneme-yanılma, uygulama konusunda ilerleme kaydedildi.

İkincil olarak projenin temel konusu olan ‘Görüntü İşleme’nin yazılımsal temelleri, hareketli ve sabit görüntülerin nasıl manipüle edilebileceği ve elde edilen veriler üzerinden görüntünün nasıl işleneceği hususunda kazanım sağlandı. Bu bilgiler ışığında yazılım alanında farklı bir dala dair vizyon kazanıldı.

Hazırlanan sunum videosunun adresi https://youtu.be/SSqTy6ABNVA
Proje dosyalarının github adresi: https://github.com/fbuni/analysis-of-production-line