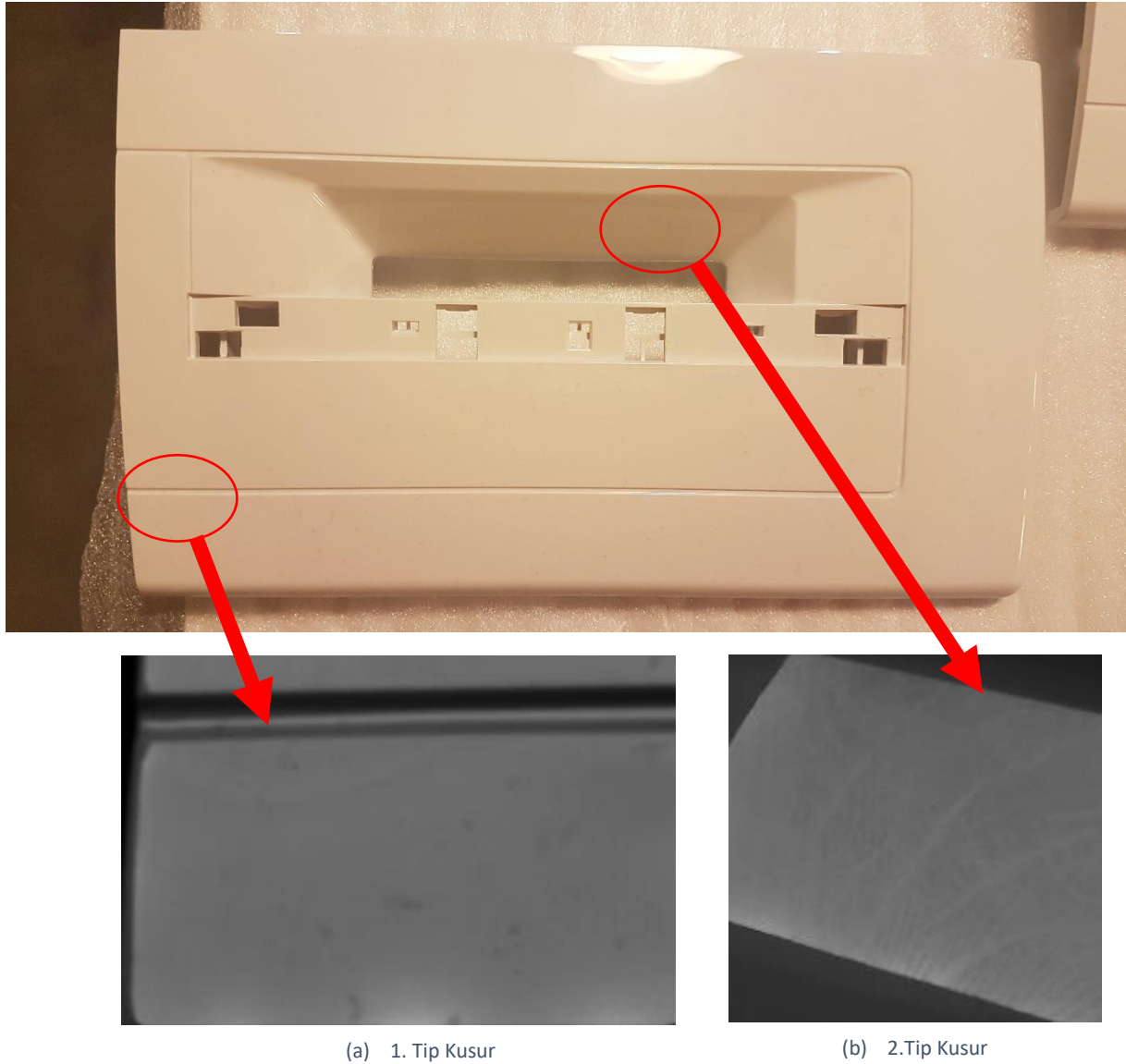


1. Giriş

Bu raporda, FAREL Plastik tarafından sağlanan parçalarda, belirli alanlarda oluşan yüzeysel kusurlar, görüntü işleme ve bilgisayar görüşü algoritmaları ile tanımlama çalışmasının metodu ve sonuçları verilmiştir. Şekil-1’de yaklaşık 120 adet örnek parçadan bir tanesi gösterilmiş ve kusurların konumları birer kamera görüntüsü ile belirtilmiştir.



Şekil-1: Örnek parçada kusurların kamera görüntüleri ve konumları

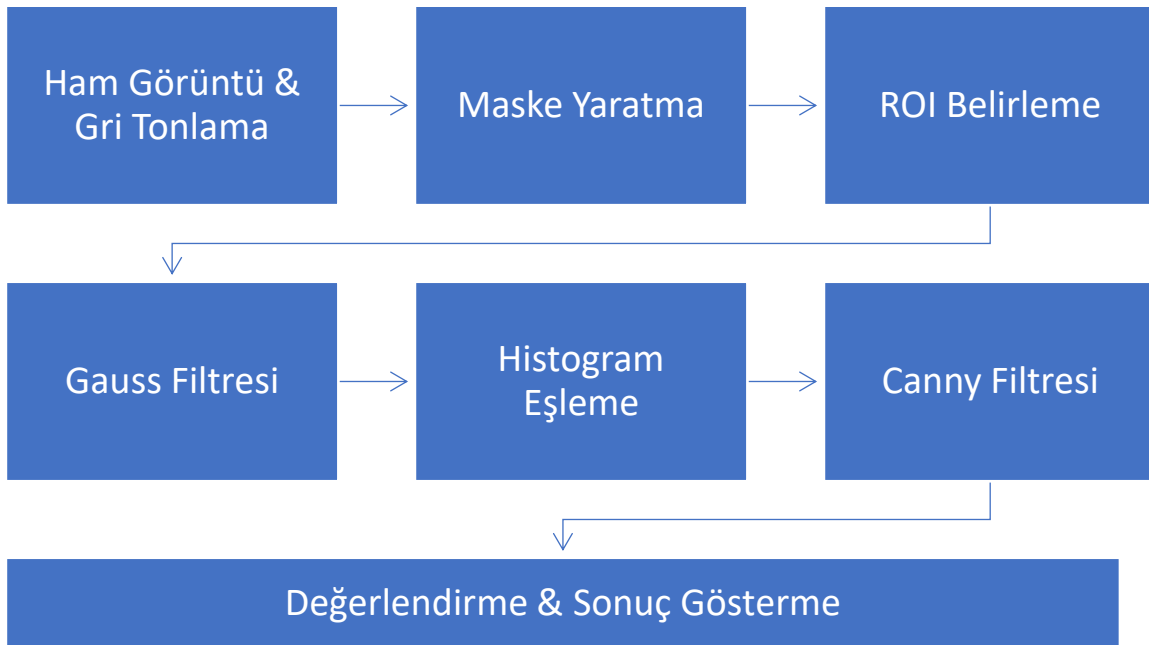
Şekilde görüldüğü gibi kusurların gözle tespiti zor olmaktadır. Boya işlemine girmemiş bu parçalar, parmak izi gösterme, kirlenme ve çizilme gibi yanıltıcı etkilere karşı oldukça yatkındır. Projenin amacı, yüksek doğruluk ve hızda (1 saniyenin altında) tanımlanan kusurların bilgisayar görüşü ile bulunmasının gerçekliğini ölçmektir.

Raporun ikinci bölümünde algoritma adımları tartışılmıştır. Sonraki bölümde, düzenek elemanları ve sonuçlar sunulmuştur. Son bölümde, gelecek çalışmalar ile sonuçlandırılmıştır.

2. Algoritma

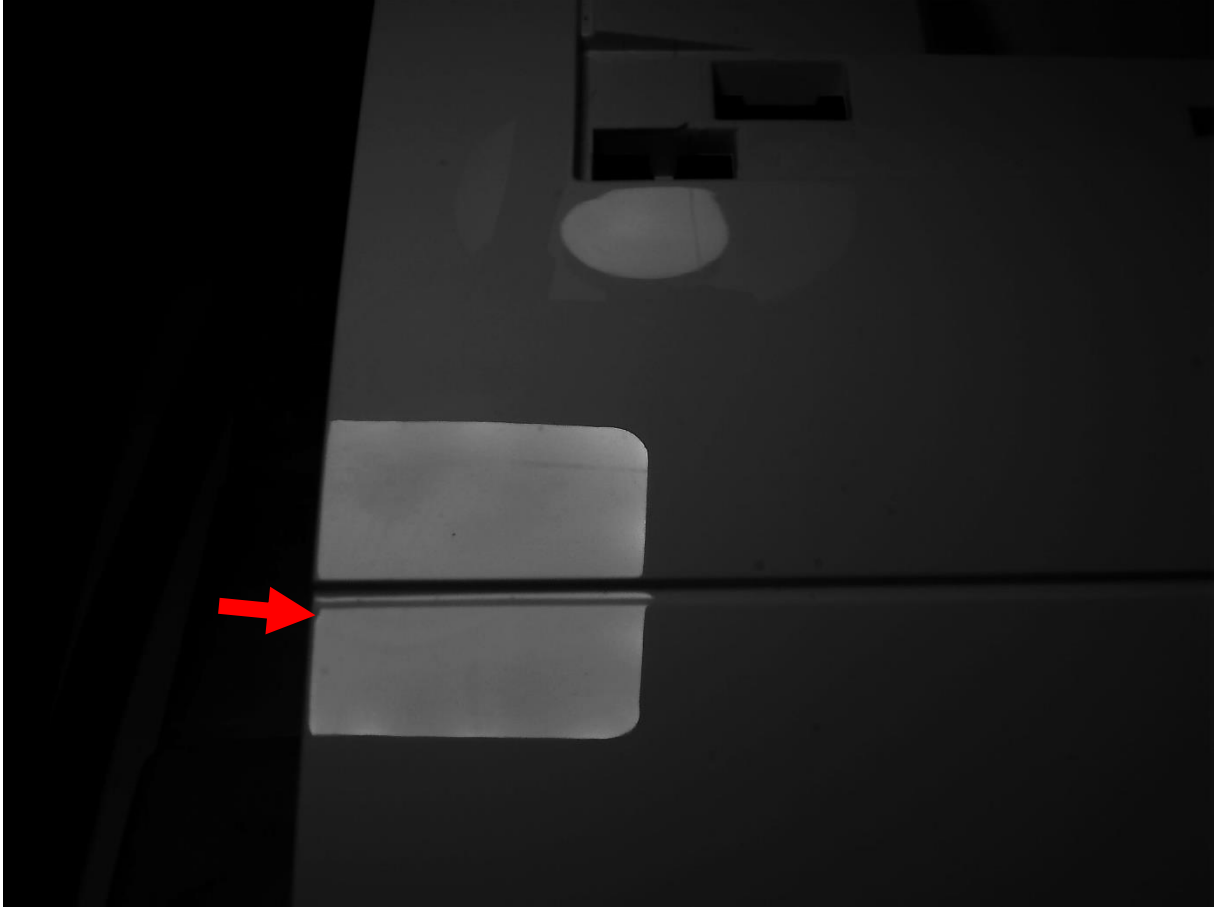
Bu raporda açıklanan tüm yöntem ve sonuçlar MATLAB programlama dili ve MATLAB R2020b platformu kullanılarak gerçekleştirilmiştir. Ara yüz tasarımı ve son ürün C++ ortamında planlanmakta, ancak modelleme kolaylığı sebebiyle algoritma sonuçları MATLAB platformunda sunulmuştur.

Tamamen otomatik olan algoritmanın temelinde, maskeleme ve kenar bulma metodu kullanılmıştır. Bu metod, sadece 1.tip kusura yönelik geliştirilmiştir. Diğer yüzey kusurunun, kamera ile yeterli belirginlikte görüntülenmesi henüz gerçekleştirilemedi. Kusurun arandığı bölgeye odaklanılmış, başarı ve hızın artırılması hedeflenmiştir. Genel algoritma akışı Şekil-2’de verilmiştir.



Şekil-2: Algoritma akış şeması

Monokrom kameradan (tek renk kanallı, renksiz görüntü) alınan görüntü gri tonlamalı şekilde her seferinde bir kare işleme girecek şekilde alınır. Bu görüntü üzerinde ışıklandırmanın görüntüsü aranmaktadır, çünkü tanımlanan kusurlar bu alanın için girdiğinde görünür olmaktadır. Ham görüntünün tamamı Şekil-3’de verilmiştir, 1.tip kusur kırmızı ok ile belirtilmiştir.



Şekil-3: Gri tonlamalı kamera görüntüsü

Bu parlak alanı belirlemek için öncelikle tüm görsele özel bir eşik değeri uygulanmış ve parlak-karanlık alanlar ayrılmıştır. Bu değer Otsu Eşik Yöntemi [1] ile belirlenir. Bu eşik değeri ile ayrılan görselde lambadan yansıyan ve istenmeyen bir parlak alan daha görülebilir (Şekil-4a). Arzulanan bölgeye ulaşmak için boşluklar morfolojik işlemler ile doldurulup (Şekil-4b) en büyük alan otomatik olarak seçilir (Şekil-4c). Gerçek maskeye ulaşmak ve aradaki çizgiyi elemek adına 4a'daki görsel ile 4c'deki görseller kesiştirilir. Sonuç parlak, kusurun aranacağı maskedir (Şekil-4d).



(a) Tüm parlak bölgeler

(b) Doldurulmuş parlak bölgeler



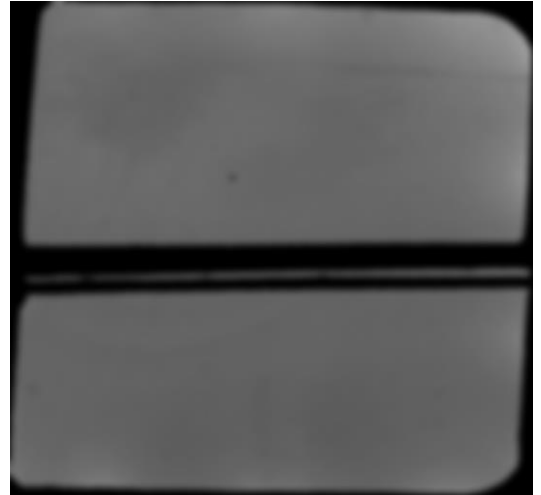
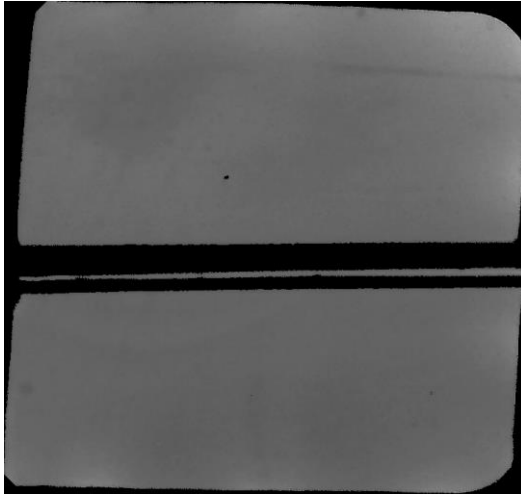
(c) En büyük parlak bölge

(d) Maske

Şekil-4: Maske aşamaları

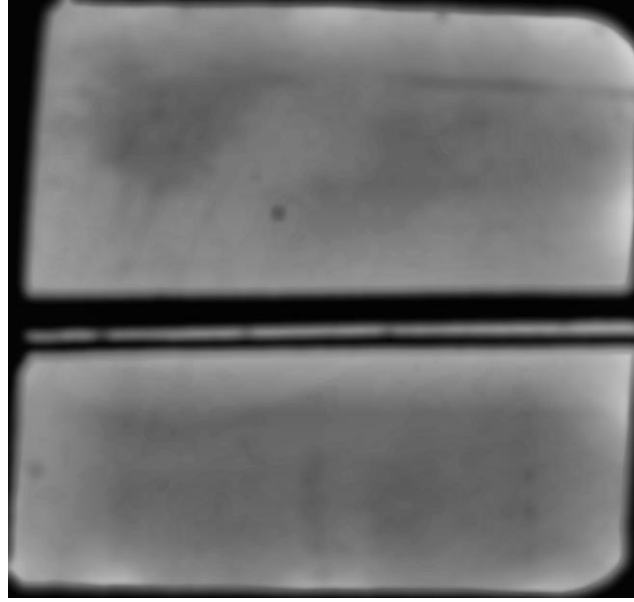
Maskenin yaratılması ardından, maskenin tüm görsel içindeki konumu ve büyüklüğü bulunur. Ayrıca maskenin ortasından geçen çizginin de konumu hesaplanır, çünkü 1.tip kusur genellikle bu çizginin alt ve sol tarafında gözlemlenmiştir. Bu şekilde, belli bir konuma odaklanmak için ROI yani “*Region of Interest*” koordinatları çıkarılmış olur.

Odak bölgeleri belirlemek, hatalı sonuçları azaltmakta ve yüksek çözünürlüklü tüm görsel yerine belli bir alana indirgeme yoluyla işlem yükünü azaltmaktadır. Maskelemenin ardından, kamera görüntüsündeki gürültünün azaltılması için çan eğrisi olarak da bilinen 2 boyutlu bir Gauss Filtresi uygulanmıştır (Şekil-5). Bu çok sayıdaki gürültüyü azaltırken, kusurların belirginliğini de kaybettirmektedir. Bu sebeple optimum bir değerde kullanılır.



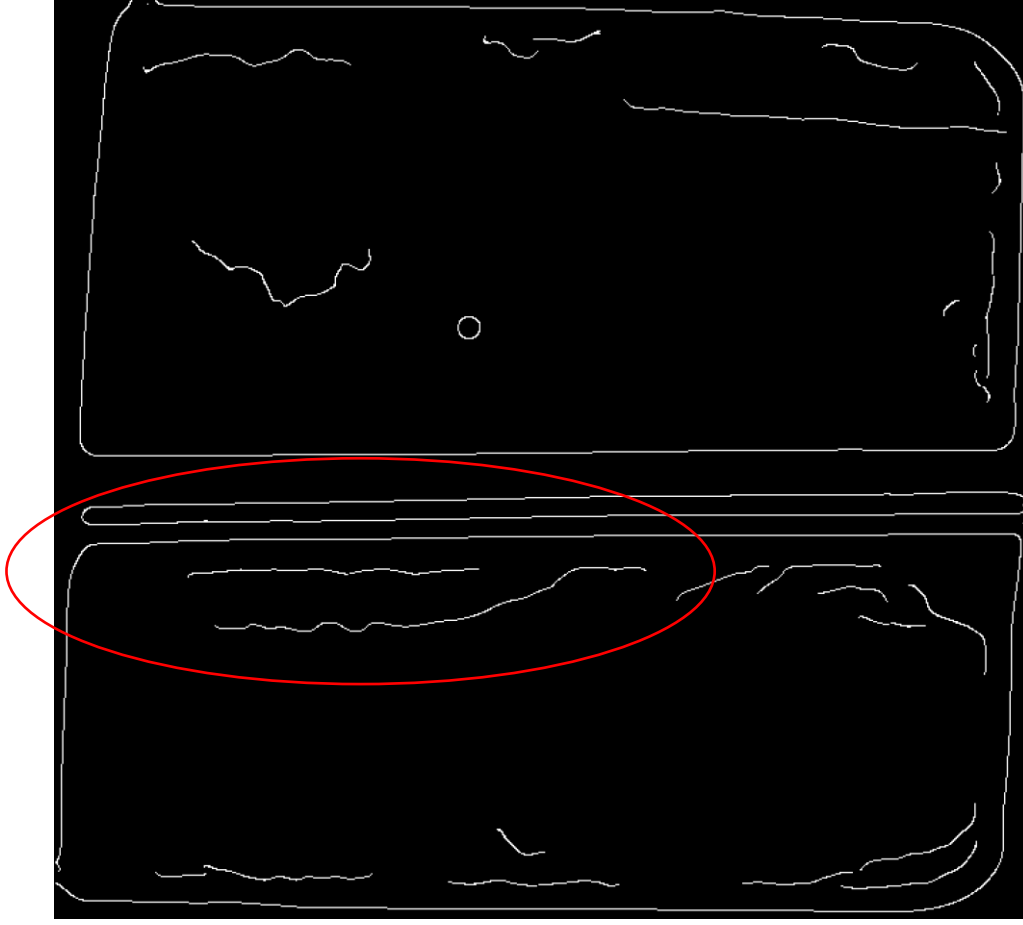
Şekil-5: Maskelenmiş görsel (solda) ve Gauss Filtre ile yumuşatılmış görsel (sağda)

Gürültüsü azaltılmış görselde, istenen objelerin (kusurlar) belirginliğini arttırmak amacıyla piksel değerlerinin farkı keskinleştirilmesi amaçlanır. Bu sebeple CLAHE [2] histogram eşleme algoritması uygulanmıştır. Yerel görsel parçalarında gri değerlerinin farkını açmaya yarar, sonuç olarak desenler daha açık hale gelir (Şekil-6).



Şekil-6: Histogram eşleme

Yeterli miktarda belirgin hale gelen desenleri, kenar bulma yöntemleri ile tanımlama aşamasında Canny Filtre [3] kullanılmıştır. Bu güçlü algoritma, görseldeki tüm kenar farklarını ortaya çıkaracak, kusur dışında birçok deseni kenar olarak tanımlayacaktır, bkz. Şekil-7. Bu aşamada filtre büyüklüğü ve Canny eşik değerlerinin doğru kullanılması önemlidir. Şekilde, kusurun sebep olduğu kenar çıktısı kırmızı ile işaretlenmiştir.



Şekil-7: Canny Filtresi sonucu

Hatalı sonuçları elemek için maskeleme ve ROI koordinatlarından yardım alınır. Parlak bölgenin kenarları en güçlü kenarlardır ve maskelenmesi gerekir. Ayrıca kusurun olası bölgesine odaklanarak diğer alanlarda oluşan çıktılar göz ardı edilmiştir. Şekil-8’de bu işlemler sonrası oluşan çıktı verilmiştir. Kullanıcıya rahatlık sunmak adına sonuçlar kalınlaştırılmıştır.



Şekil-8: Elenen kenarların sunumu ve kusur

Daha iyi bir algı yaratmak için aşağıdaki Şekil-9’da kamera görüntü üzerine sonuçlar basılarak kullanıcıya sunulur. Bu şekilde, diğerlerinin aksine kusur çevresindeki yeşil renkli kutu yazılım tarafından çizilmiştir. Kusur bulunamayan durumlarda ise kırmızı kutu kullanılır, bir sonraki kare kusur bulunmayan bir yüzeyin çıktısıdır (Şekil-10).



Şekil-9: Kusurun tespit edildiği sonuç görseli



Şekil-10: Kusur bulunmayan başka bir parçanın sonucu

3. Düzenek ve Deney Sonuçları

Algoritmayı geliştirme öncesi video çekme ve gerçek zamanlı testler için aşağıdaki teknik ekipman kullanılmıştır.

Lens : Fujinon HF8XA-5M

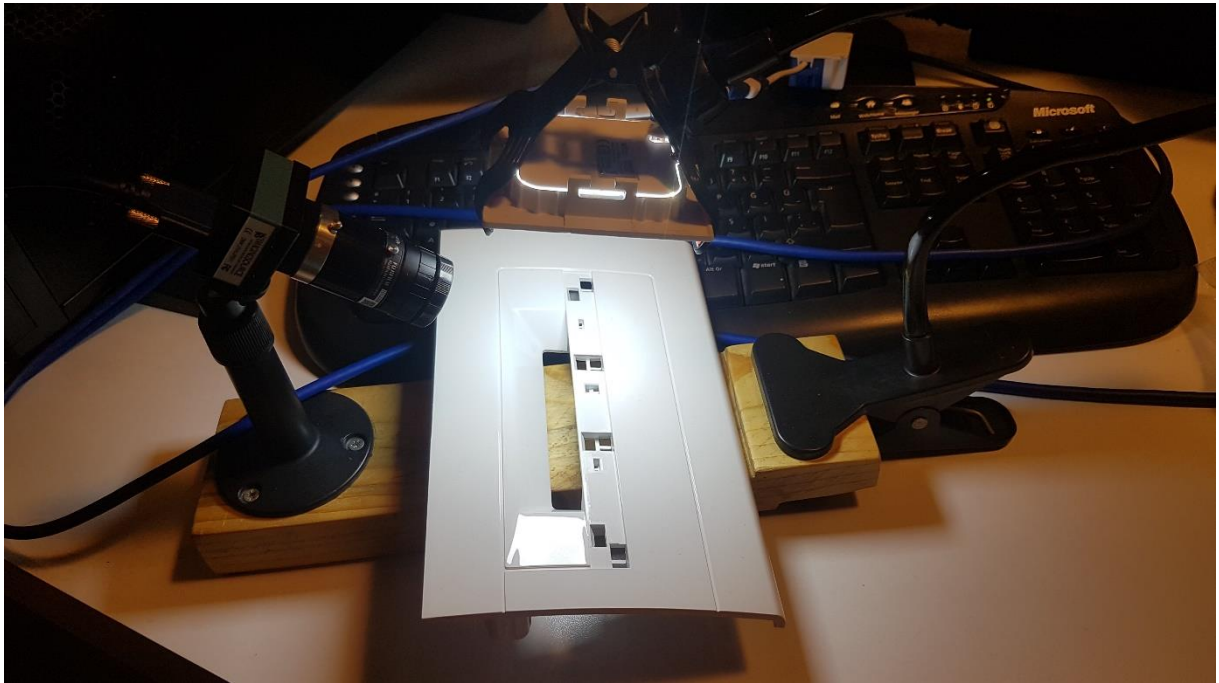
Kamera : TIS DMK 27AUP031

Sensör : CMOS Aptina MT9P031

Bağlantı: USB 3.0 SuperSpeed

Bu elemanlar Şekil-11'deki basit düzeneğe sabitlenmiştir. Kamera saniyede 15 kare kapasitesi olan, tek renkli (monokrom, gri tonlamalı) bir modeldir. Testlerde, maksimum çözünürlük olan 2592x1944 (5 Megapixel) girdi olarak kullanılmıştır.

Işık kaynağı 7000° Kelvin sıcaklığında (gün ışığı) seçilmiştir.



Şekil-11: Test düzeneği

Test sırasında parça hareketli kullanılmış ve farklı açılarda görüntü alınmıştır. Fakat, hareket modeli belirli olursa ve uygun parametreler ayarlanırsa daha yüksek başarılı sonuçlar alınacaktır. Ani hareketlerde, kamera düşük kare hızı (15 FPS) sebebiyle keskin görüntüler yakalamakta zorlanmaktadır. Bu da algoritma çıktısına olumsuz yansımaktadır.

Benzer şekilde, ortamın parlaklık gibi fotometrik değerlerinin sabit olması doğru parametre seçimini etkileyecektir. Testler sırasında fazla ve az parlaklığı örneklemek için lensin diyafram ayarı, manuel olarak değiştirilmiştir. Sonuçlara göre, fazla parlak ışıkta maske bulmanın imkânsız hale gelmesi ve

desenlerin kaybolması sebebiyle algoritma başarısız olmaktadır. Aksine karanlık ortamda, kameranın yakalayabildiği desenler başarıyla ile çıkartılmaya devam etmektedir.

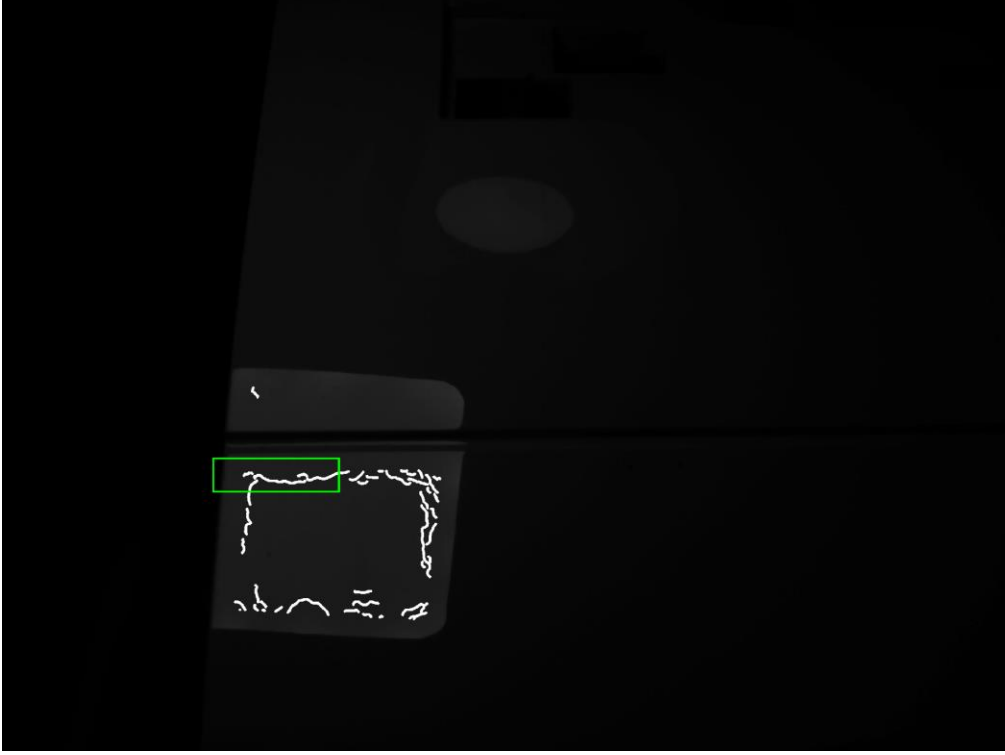
Tüm bu durumların bulunduğu 523 karelik test aşamasında algoritma 105 kere hatalı sonuç vermiş, %80'lik bir başarı yakalamıştır. Bu karelerden yaklaşık 20 tanesi yüksek parlaklık, 30 tanesi karanlık ve yaklaşık 20 tanesi ani hareket içermektedir. Sabit koşullar altında algoritmanın doğruluğunun %90'ın üzerinde olması öngörülmektedir.

MATLAB ortamında yapılan testlerde kare başına ortalama 0.4 saniyelik bir işlem süresi gözlemlenmiştir. Gerçek zamanlı uygulamada ve C++ dili kullanılarak yapılan yazılımın 5 FPS hızında çalışacağı sonucuna varılmıştır. Bu değer kamera limitleri sebebiyle maksimum 15 olabilmektedir.

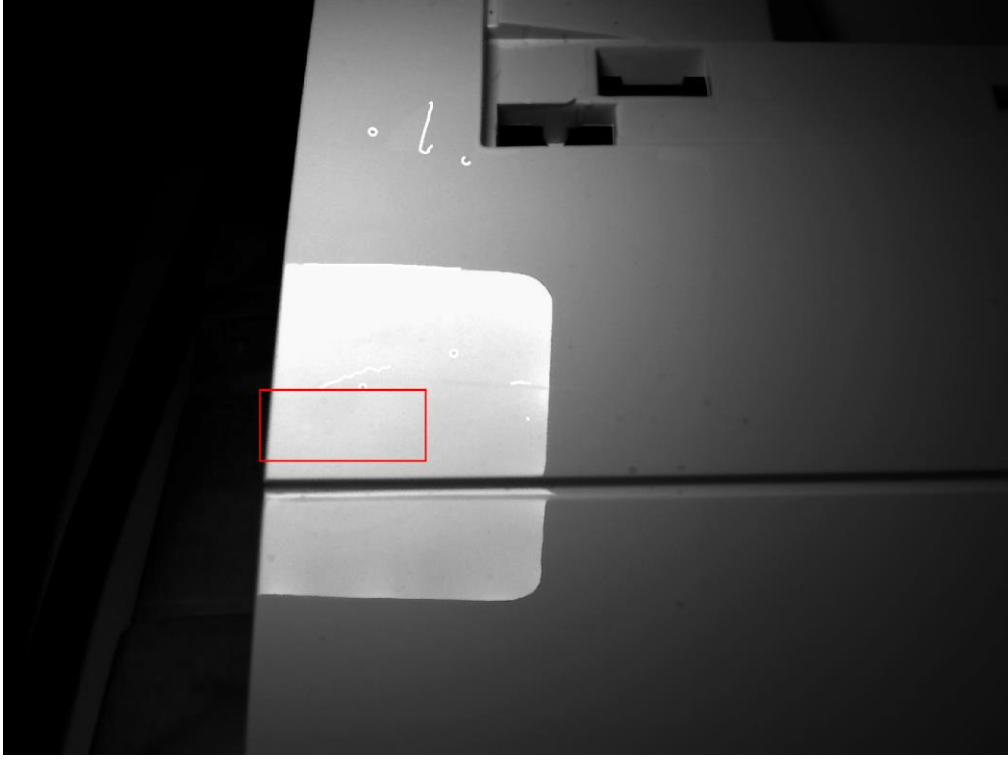
Aşağıdaki şekillerde, bazı başarılı sonuçlar (Şekil-12) ve başarısızlık yaratan durumlara birer örnek (Şekil-13) verilmiştir.







Şekil-12: Başarılı sonuçlar



Şekil-12: Fazla parlak ve ani hareket durumları

4. Sonuç

Sonraki adımda, C++ dilinde tasarlanan kullanıcı ara yüzüne algoritmanın eklenmesi ve gerekli fonksiyonların kullanıcıya sunulması gerçekleştirilecektir. İkinci tipteki kusur ile ilgili çalışmalar devam etmektedir, henüz algoritma aşamasına geçmeden, kamera ile yeterli belirginlikte görseller alınamamıştır.

Yöntem olarak, ön çalışma raporunda Laplacian gibi filtreleme metodları, ardından proje süresince makine öğrenmesi modelleri denenmiştir. Ancak hız ve doğruluk bir arada düşünüldüğünde kenar bulma algoritmaları temelindeki yöntem seçilmiştir. Kullanılan yöntem başarılı olsa da doğru parametreler ortamın fotometrik değerlerine ve ekipmanlara bağlıdır.

Raporun test kısmında belirtildiği gibi, algoritmanın parametrik olması farklı ekipman ve şartlarda optimizasyon gereğini ortaya çıkarmaktadır. Daha yüksek teknoloji sunan kamera-lens elemanları, parçaların modellenmiş robotik hareketleri ve tutarlı ışık koşulları başarının artması için olumlu olacaktır.

Teknik Referanslar

- [1] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms." *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 9, No. 1, 1979, pp. 62–66.
- [2] Zuiderveld, Karel. "Contrast Limited Adaptive Histogram Equalization." *Graphic Gems IV*. San Diego: Academic Press Professional, 1994. 474–485.
- [3] Canny, John, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679-698.