

Programın Genel Çalışma Yapısı ve Kütüphaneler:

Bu program, görüntü işleme tekniklerini kullanarak, baskı devre (PCB) üzerindeki kusurları tespit eden bir görüntü işleme uygulamasıdır. Genel olarak, kullanılan kütüphaneler ve programın çalışma mantığı aşağıda açıklanmıştır.

Kullanılan Kütüphaneler:

1. **OpenCV (cv2):** Görüntü işleme ve video okuma/yazma işlemleri için kullanılır. OpenCV, bilgisayarla görme görevlerinde yaygın olarak kullanılan bir kütüphanedir.
2. **Numpy:** Sayısal işlemler ve matris hesaplamaları için kullanılır. Numpy, çok boyutlu diziler ve matrislerle çalışmak için temel araçlar sunar.
3. **Matplotlib (plt):** Python'da statik, animasyonlu ve etkileşimli görselleştirmeler oluşturmak için kullanılan bir kütüphanedir. Burada, görüntülerin çeşitli işleme aşamalarında gösterilmesi için kullanılır.

Kullanıcı Girdisi, Kullanıcıdan hata türünü (Open Circuit, Missing Hole, Mouse Bite) ve belirtilen aralıktaki bir görüntü numarasını seçmesi istenir. Kullanıcının girdisine göre test görüntüleri için dosya yolları ayarlanır.

Görüntüleri Okuma ve Gösterme, Referans görüntü (PCB) ve test görüntüsü hem renkli hem de gri tonlamalı formatlarda okunur. Görüntüler, Matplotlib kullanılarak görselleştirilir.

Görüntü Ön İşleme, Gri tonlamalı referans görüntü, gürültüyü azaltmak için Gaussian bulanıklaştırma kullanılarak bulanıklaştırılır. Bulanıklaştırılmış referans görüntüye adaptif eşikleme uygulanarak ikili (binary) bir görüntü oluşturulur. Bu, PCB'nin özelliklerini vurgulamaya yardımcı olur.

Özellik Tespiti ve Eşleştirme, ORB (Oriented FAST and Rotated BRIEF) özellik tespit edici kullanılarak hem referans hem de test görüntülerinde anahtar noktalar ve tanımlayıcılar tespit edilir. Tanımlayıcılar, Hamming mesafesi ile brute-force eşleyici kullanılarak eşleştirilir. En iyi eşleşmeler, daha fazla işlenmek üzere seçilir.

Görüntü Hizalama, Eşleştirilmiş anahtar noktalar kullanılarak homografi hesaplanır ve test görüntüsü referans görüntü ile hizalanır. Hizalanmış test görüntüsü, referans görüntünün perspektifine uyacak şekilde dönüştürülür.

Hizalama Sonrası İşleme, Hizalanmış test görüntüsü, referans görüntüdeki ile aynı ön işleme adımlarından geçirilir (Gaussian bulanıklaştırma ve adaptif eşikleme). Referans ve test görüntülerinin ikili görüntüleri çıkarılarak farklılıklar vurgulanır, bu farklılıklar kusurları temsil eder.

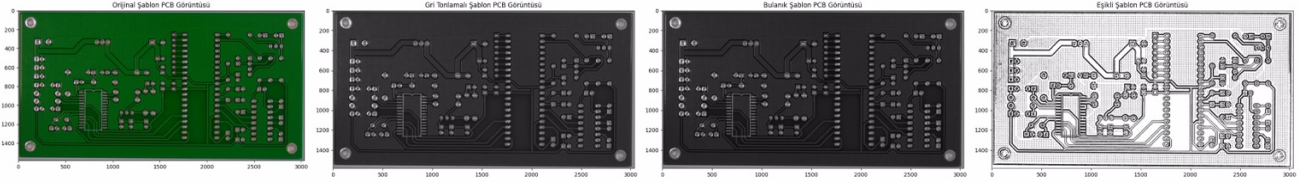
Son Kusur Tespiti, Fark görüntüsü, gürültüyü azaltmak ve kusurları daha net vurgulamak için median bulanıklaştırma ile işlenir. Ortaya çıkan görüntü, PCB üzerindeki tespit edilen kusurları gösterir.

Algoritma:

- 1) Başla.
- 2) Kullanıcıdan bir test PCB görüntüsü için dosya adresi al.
- 3) Referans görüntüsü gri ve grafikte ölçeklendir.
- 4) Referans görüntüsü gri, bulanık, eşikli şablon olmak üzere işle.
- 5) Alınan test PCB görüntüsü grafiğe ölçeklendir.
- 6) Bu test görüntüsü hizala; gri, bulanık, eşikli şablon olmak üzere işle.
- 7) Referans ve test görüntüleri eşikli şablonlar üstünde çakıştır.
- 8) Test PCB üzerindeki kusurları tespit et ve grafikte noktalar şeklinde göster.
- 9) Bu grafikteki noktaları "Annotations" içindeki verilerle karşılaştır ve yorumla.
- 10) Bitir.

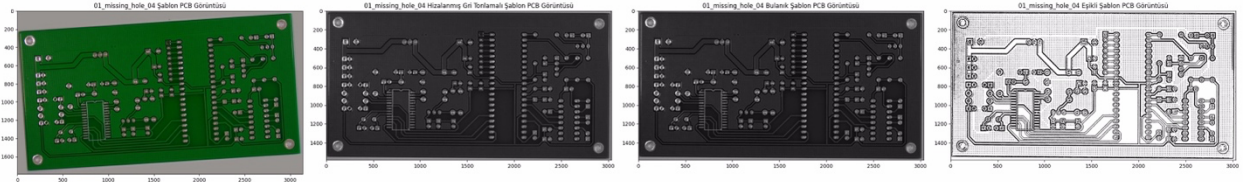
Programın Çalışması:

Program, önce Referans klasörü içindeki 01.jpg dosyasını referans PCB görüntüsü olarak alıp, bu görüntüyü gri tonlamalı, bulanık ve eşikli şablon olacak şekilde işler (Görüntü 1.0). Amaç, test görüntüsünü referans görüntüsü ile eşikli şablon halinde karşılaştırıp hataları yakalamaktır.



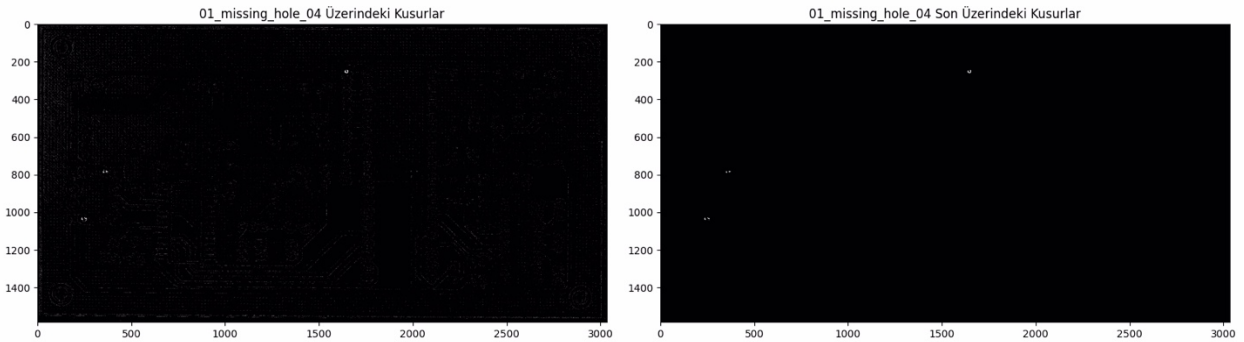
Görüntü 1.0 (Referans PCB'nin işlenmesi)

Daha sonra, kullanıcıdan alınan test PCB görüntüsü (rotation klasörü içindeki herhangi bir hata) hizalanır ve bu hizalanmış test PCB görüntüsü gri tonlamalı, bulanık ve eşikli şablon olacak şekilde işlenir (Görüntü 2.0).



Görüntü 2.0 (Test PCB'nin işlenmesi)

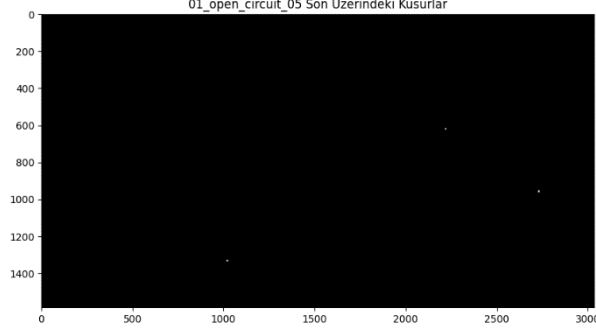
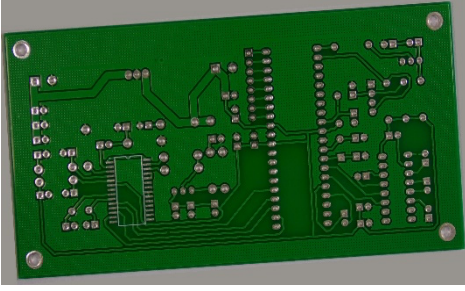
Referans ve test görüntüleri, en son işlenmiş halleri olan eşikli şablonlar üzerinden karşılaştırılır. Çakışma sonucunda, test PCB üzerindeki kusurlar tespit edilir ve grafikte noktalar şeklinde gösterilir (Görüntü 3.0).



Görüntü 3.0 (PCB üzerindeki kusurların tespit edilmesi)

Ekran Görüntüleri:

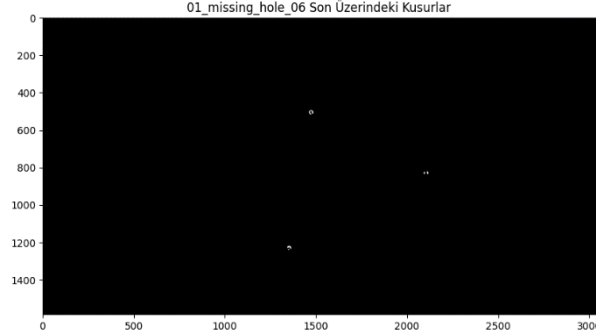
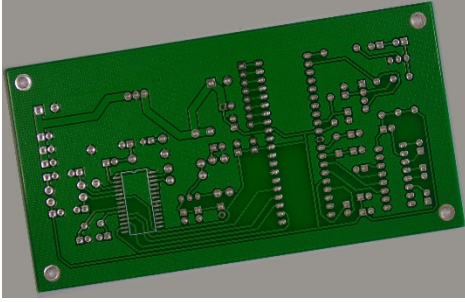
Open Circuit_05 PCB



Open Circuit_05 Annotation Tablosu

xmin	ymin	xmax	ymax
2204	604	2236	641
1015	1312	1050	1349
2713	946	2746	979

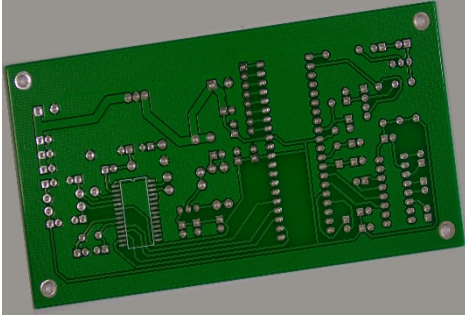
Missing Hole Circuit_06 PCB



Missing Hole Circuit_06 Annotation Tablosu

xmin	ymin	xmax	ymax
1446	480	1504	542
1326	1203	1386	1269
2070	800	2140	860

Mouse Bite Circuit_07 PCB



Mouse Bite Circuit_07 Annotation Tablosu

xmin	ymin	xmax	ymax
1471	403	1508	461
441	916	476	947
663	1302	702	1343

Sonuç, Yorumlar ve İyileştirmeler:

Veri setinde verilen test verilerinin programa verilmesiyle, ekran görüntülerindeki örnekler gibi tutarlı sonuçlar elde edilmiştir. Noktaların yanlarındaki tablolarda belirtilen minimum ve maksimum yatay ve dikey değerlerin incelenmesi sonucunda, programın çıkardığı koordinatların veri setindeki koordinatlarla uyduğu gözlemlenmiştir. Bu da programımızın istenilen şekilde çalıştığını göstermektedir.

Programın tutarlı çalışmasına rağmen çıkan görüntülerde hataların gösterildiği noktalarda bazı görsel kusurlar vardır. Mesela ekran görüntüleri kısmındaki “01_missing_hole_06” grafiğindeki noktalardan bir tanesi iki tane olarak algılanmış. Bu sorunu düzeltmek için test ve referans görüntüleri ve grafikleri yeniden boyutlandırılabilir. Daha küçük boyutlandırma yapıldığında görüntüler daha keskin ve seçilebilir olacaktır. Mesela (750,450) koordinatları baz alınarak yeniden boyutlandırılması bu hatanın düzeltilmesinde kullanılabilir. Ancak bu yöntem kullanıldığında Annotation tablolarındaki değerlerle uyuma olmayacağından tablolardaki değerlerin ya da program çıktılarının da aynı oranda güncellenmesi gerekir.

Akış Şeması:

