

**SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ**



**GÖRÜNTÜ İŞLEME FİNAL PROJE ÖDEVİ**

**AD SOYAD : Nefise Aybar      NO: B210109036**

**DERS : Görüntü İşleme**

**DERS EĞİTMEN: Dr. Öğr. Üyesi Muhammed Ali Nur Öz**

**KONU: Görüntü işleme tekniklerini kullanarak, baskı devre (PCB) üzerindeki kusurları tespit eden bir algoritma geliştirmek**

# PCB Kusur Tespiti Algoritması Raporu

## 1. Giriş

PCB (Printed Circuit Board) üzerindeki kusurlar, elektronik devrelerin verimliliğini ve dayanıklılığını doğrudan etkileyen kritik problemlerdir. Bu projede, PCB üzerindeki kusurları otomatik olarak tespit eden bir algoritma geliştirdim. Algoritma, referans bir PCB görüntüsünü test görüntüleri ile karşılaştırarak kusurları tespit etmeyi amaçlıyor

Algoritma üç temel kusur türü için optimize edildi:

- Missing Hole
- Mouse Bite
- Open Circuit

## 2. Yöntem

Geliştirdiğim algoritma, görüntü işleme teknikleri ve temel istatistiksel analiz yöntemleri üzerine kuruludur. Projemde sırasıyla şu yolları izledim:

### 2.1. Görüntü ve Anotasyonların Yüklenmesi

- Test görüntülerini ve referans görüntüsünü belirlenen dosya yollarından yüklenir.
- Kusur anotasyonları XML dosyalarından parse edilerek kusur bölgeleri elde edilir.

### 2.2. Görüntü Hizalama (Image Registration)

- Test ve referans görüntülerini arasında hizalama işlemi yapılır.
- `cv2.findTransformECC` yöntemi kullanılarak, test görüntüsünü referans görüntüsüne hizalanır.
- Bu işlem, referans ve test görüntüsünü arasında ölçek, döndürme ve kaydırma farklarını düzeltir.

### 2.3. Fark Analizi

- Hizalanmış test görüntüsünü referans görüntüsünü arasındaki farklar `cv2.absdiff` kullanılarak hesaplanır.
- Farklar, bir eşikleme işlemi ile ikili (binary) bir görüntüye dönüştürülür. Böylece yalnızca anlamlı farklar öne çıkar.

### 2.4. Kontur Tespiti ve Kusur İşaretleme

- İkili fark görüntüsünde konturlar tespit edilir.
- Tespit edilen konturlar minimum alan filtresi, kenar filtresi ve şekil filtresi ile optimize edilir.
- Her kontur bir mavi kutu ile işaretlenirken, anotasyon kutuları yeşil olarak görselleştirilir.

## 2.5. Doğruluk Analizi

- Tespit edilen kutular ve anotasyonlar arasındaki örtüşme oranları (IoU - Intersection over Union) hesaplanır.
- Doğruluk metrikleri (Precision, Recall, Accuracy) şu formüller ile hesaplanır:
  - **Precision:**  $Precision = \frac{TP}{TP + FP}$
  - **Recall:**  $Recall = \frac{TP}{TP + FN}$
  - **Accuracy:**  $Accuracy = \frac{TP + TN}{TP + FP + FN}$

## 3. Algoritma Akışı

1. Referans ve test görüntüleri yüklenir.
2. Anotasyonlar yüklenir.
3. Test görüntüsü referans görüntüsüne hizalanır.
4. Fark analizi yapılır.
5. Tespit edilen kusurlar ve anotasyonlar karşılaştırılır.
6. Precision, Recall ve Accuracy değerleri hesaplanır.
7. Sonuçlar görselleştirilir ve kaydedilir

# Algoritmanın Başarısı ve Kullanılabilirliği

## Algoritmanın Başarıları

### 1. Doğru Hizalama:

- Referans ve test görüntüleri, `cv2.findTransformECC` yöntemi kullanılarak başarılı bir şekilde hizalandı
- Histogram eşitleme ve bulanıklaştırma adımları hizalama doğruluğunu arttırdı

### 2. Fark Analizi ve Kusur Tespiti:

- Görüntüler arasındaki farklar, `cv2.absdiff` yöntemi ile doğru bir şekilde tespit edildi
- Eşikleme ve kontur tespiti adımları kullanılarak kusurların konumları belirlendi

### 3. Doğruluk Analizi

- Tespit edilen kusurlar, anotasyonlarla karşılaştırılarak performans metrikleri hesaplanır:
  - **Precision:** Gereksiz kutuları minimum seviyeye indirmiştir.
  - **Recall:** Çoğu gerçek kusur başarıyla tespit edilmiştir.
  - **Accuracy:** Precision ve recall arasında dengeli bir performans sunmuştur.

### 4. Esneklik:

- Kod, farklı kusur türleri (Missing Hole, Mouse Bite, Open Circuit) için uygulanabilir durumda
- Test veri setindeki farklı boyut ve şekillerdeki kusurları başarılı bir şekilde analiz edildi.

## 2. Algoritmanın Kullanılabilirliği

### 1. Genel Kullanılabilirlik:

- Algoritma, farklı PCB veri setlerine kolayca uyarlanabilir.
- Veri yollarını ve parametreleri ayarlayarak diğer test senaryolarına hızla uygulanabilir.

### 2. Modüler Tasarım:

- Algoritmanın her bir fonksiyonu (örneğin, hizalama, fark analizi, doğruluk değerlendirme) bağımsız olarak çalışabilir ve gerektiğinde genişletilebilir.

### 3. Performans:

- İşlem süresi ve doğruluk dengesi göz önünde bulundurularak optimize edilmiştir.
- Minimum alan ve şekil filtreleri gereksiz konturları azaltarak işlem süresini iyileştirmiştir.

#### **4.Hata Yönetimi:**

- Kod, hizalama hataları durumunda test görüntüsünü doğrudan analiz etmeye devam edecek şekilde tasarlanmıştır.