



Ambalaj Yüzeyindeki Hataların Görüntü İşleme Tekniği ile Tespiti

  
Gültekin Çağıl*, Fatma Okcu¹, Nazlı Hilal Güngör²

*,^{1,2}Sakarya Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü

Öz

Bu çalışma; gıda ambalajı üretimi yapan bir firmada, ambalaj ürünlerinin yüzeylerindeki hataların Görüntü İşleme yöntemi ile tespiti üzerinedir. Firmadan alınan ürünlerin görüntülerinde, Pycharm Platformunda Python Programlama Dilinde OpenCV Kütüphanesi desteği ile; boya, sinyal, ayarsızlık, bıçak çizgisi gibi hata türlerinin tespiti yapılmaktadır. OpenCv Kütüphanesi zamandan kazanç sağlamakta ve sistemin performansını artırmaktadır. Çalışmada, ilk olarak renk maskeleye yapılarak, mavi rengin “dusuk” ve “yüksek” adlı iki dizi oluşturulmaktadır. Görüntü HSV formatına çevrilmektedir, çünkü maskeleye işlemi yapılabilmesi için görüntünün HSV formatına ihtiyaç duyulmaktadır. Maskeleye işleminden sonra mavi olan arka plan, beyaz ve diğer tüm renkler siyah olmaktadır. Bu işlemden sonra beyaz renk olan pikseller orijinal değerlerine dönüştürülerek siyah renk olan piksellerin değerleri değişmemesi sağlanmaktadır. Kenar Algılama işlemi yapabilmek için görüntü gri tona çevrilmekte böylece görüntüde istenmeyen gürültüler elde edilebilmektedir. Daha sonra hatalı kısımların görüntüleri alınarak benzerlik uygulaması ile gerçek görüntüdeki hatalı kısımları bulunması sağlanmaktadır. Son olarak hatalı kısımlar sarı dörtgen içerisinde belirtilmektedir. Firmada operatörler hata tespitini göz ile muayene ederek yapmaktadırlar. Bu sistemde operatörün hatayı tespit edemediği durumlarda, sarımının yapıldığı bobinlerde hatalı ambalajlar üretilmeye devam etmektedir. Bu durum karşısında ortaya çıkan maliyetleri en aza indirmek amacıyla Görüntü İşleme yöntemi kullanılmaya karar verilmiş ve uygun sonuçlar alınmaktadır.

Anahtar kelimeler: Görüntü İşleme, Python Programlama Dili, OpenCv, Gıda Ambalajı, Kalite Kontrol.

Determination Of Errors On The Surface Of The Packaging By Image Processing Technique

Abstract

This work; The aim of this study is to detect the defects on the surfaces of packaging products by means of Image Processing in a company producing food packaging. In the Images of products received from the company, with the support of OpenCV Library in Python Programming Language in Pycharm Platform; types of defects such as paint, signal, misalignment, knife line are detected. The OpenCv Library saves time and improves system performance. In this study, firstly by color masking, two sets of blue color which are “dusuk” and “yüksek” are created. The image is converted to HSV format, because the masking process requires the HSV format of the image. After masking, the blue background becomes white and all other colors become black. After this process, the pixels with white color are converted to their original values so that the values of the pixels with black color do not change. In order to perform Edge Detection, the image is converted to gray tone so that unwanted noise is obtained in the image. Then, the images of the faulty parts are taken and similarities are applied to find the faulty parts in the real image. Finally, the defective parts are indicated in the yellow rectangle. Operators in the company make the error detection by visual inspection. In this system, when the operator cannot detect the error, the wrong packages continue to be produced in the coils where the wrapping is made. In order to minimize the costs incurred in this situation, it was decided to use the Image Processing method and appropriate results were obtained.

Keywords: Image Processing, Python Programming Language, OpenCV, Food Packaging, Quality Control.

* Sorumlu yazar. Gültekin Çağıl
E-posta adresi: cagil@sakarya.edu.tr

Alındı : 20 Eki 2019
Revizyon : 22 Eki 2019
Kabul : 27 Eki 2019

1. Giriş (Introduction)

Son yıllarda üretim yapan kuruluşlar arasında rekabet ortamı artmaktadır. Artan rekabetle birlikte her kuruluş daha fazla üretim yapmak ve tüketiciye daha kaliteli ürünler sunmak istemektedir. Bu koşullarda, olabilecek en hatasız üretim sisteminin oluşturulması gerekmektedir. Üretim sisteminde üretilen her hatalı veya kalitesiz ürün üreticiye zaman, malzeme ve enerji kaybı olarak geri yansımaktadır. Bu durum firmada maliyet kaybı oluşturduğu gibi müşteri güveni açısından da kayıplara yol açmaktadır. Bu sebepten dolayı üretilen ürünlerin üretim aşaması sırasında kontrolü ve ayıklanması üretim sisteminin sağlıklı bir şekilde yürütülmesi için oldukça önem arz etmektedir. Bu kontrol sisteminde hem hatayı hem de insan gücü maliyetini minimize etmek için Yapay Zekâ seçeneklerinden Görüntü İşleme Yönetiminin kullanılması uygun görülmüştür.

Üretilen ürünlerde zaman zaman makinelerden sebep yüzey hataları meydana gelmektedir. Bu hataların, insan tarafından kontrol edilerek hatasız olan diğer ürünler arasından ayrılması gerekir. Burada, üretim sahasında kontrol olarak çalışan bir kişinin iş tecrübesi, o anki fiziksel ve ruhsal durumu çok fazla ön plana çıkmaktadır. Ayrıca, kalite kontrolü için ayrılan zaman artmakta ve kontrol objektif bir şekilde yapılmayabilmektedir. Büyük çapta üretim yapan endüstriyel sistemlerde görüntü işleme tabanlı kalite kontrolü bir gereklilik haline gelmektedir. Bilgisayar yazılım ve donanımlarındaki büyük teknolojik gelişmeye bağlı olarak görüntü işlemeye dayalı uygulamalar hızla yaygınlaşmaktadır. Görüntü işlemede asıl amaç, gerçek ortamdan alınan dijital görüntülerin, görüntünün en küçük parçası olan piksel değerlerinin, çeşitli algoritmalar ile istenilen şekle getirilmesidir (Zeki, 2011).

Hata tespiti için çoğunlukla insanlar kullanılmakla birlikte son yıllarda makinalar tercih edilmektedir. Makinaların tercih edilme sebepleri ise yorulmama, sürekli çalışma, uzun vadede maliyetinin düşük olması, kararlı çalışma gibi avantajlarının olmasıdır. Bu çalışmanın yapıldığı ürünün üretiminde kullanılan baskı makinesinin görüntüsü Şekil 1' de verilmektedir.



Şekil 1. Baskı Makine Görüntüsü (Printing Machine Image)

Çalışmanın yapıldığı işletmede, üretim programına göre makineler iş taksimatı gerçekleştirilmektedir. Silindir deposundan alınan silindirler, baskı makinelerinin her silindir sırasına uygun renkteki ünitelere getirilerek montajı yapılmaktadır. Basılacak malzemeye göre uygun kauçuklar makinelerin uygun bölümüne takılmaktadır. Hazne yıkama bölümünden getirilen temiz boya hazneleri yukarıya alınarak aparatları takılıp, boya sıyırma bıçakları da takılarak ayar yapılmaktadır. PVC yarı mamul ve hammadde deposundan alınan basılacak işin malzeme bobini makineye ayarları yapılarak takılmaktadır. Malzeme; ayırıcıya bağlanır, boyahane deposundan alınan baskı mürekkepleri süzülerek haznelere aktarılır, makineye start (başlat) verilerek baskı ve renk ayarları yapıldıktan sonra baskı işlemine geçilerek üretim yapılmaktadır. Üretim bittikten sonra basılan malzemeye göre parafin, lamine yapılmakta veya kesme bölümüne gönderilmektedir. Üretimde hata tespiti işleminde, bobine hata türleri tanımlı etiketler yapıştırılmaktadır. Hata türleri etiketlerin bobine yapıştırılmış durumu Şekil 2'de gösterilmektedir.



Şekil 2. Bobine yapıştırılan etiket (Label affixed to coil)

Çalışmanın yapıldığı Amcor Flexible fabrikasında gıda ambalajı üretimi yapılmaktadır. Bu süreçte ambalajın basım işleminde oldukça fazla fire verilmektedir. Fireler; boya, sinyal, ayarsızlık, silindirde meydana gelen vuruşlar ve rakle çiziklerinde kaynaklanıyor olabilmektedir.

Baskı makinesindeki ayarsızlık, boya, sinyal, silindirde meydana gelen vuruşlar ve rakle (bıçak) çizigi gibi sebeplerden kaynaklanan hataların, operatör tarafından gözle muayenesi yapılmaktadır. Fark edilen hatalara göre ürünlere etiket yapıştırılmaktadır. Üründeki hatalar operatörün dikkatsizliğinden dolayı fark edilmez ve etiket yapıştırılmaz ise bobinin geri kalan kısmı hatalı olarak üretilerek fire miktarının artmasına sebep olmaktadır. Etiket yapıştırılmayan hatalı ürünler kesildikten sonra, müşteriye sevkiyat işlemi gerçekleştirilmektedir. Bu hatalı ürünler müşteri tarafından geri gönderilmektedir. Bu durum firmada maliyet ve güven açısından kayıplara yol açmaktadır. Bu fire miktarlarında ki artış, hem enerji hem maliyet hem de zaman kısıtlarında kayıp oluşturmaktadır. Bu kayıpları minimuma indirgeyebilecek yöntem olarak Görüntü İşleme düşünülmüştür.

Çalışmada ürünün görüntülerinde oluşan bir veri tabanı oluşturulmuştur. Bu veri tabanında; boya, bıçak çizigi, ayarsızlık ve parlama hata türlerine sahip ürün görüntüleri mevcuttur. Bu hataların tespiti, Pycharm Platformunda Python Programlama Dilinde OpenCV Kütüphanesi desteği ile yapılmaktadır.

Çelikdemir(2015)'de yüksek sıcaklığa maruz kalmış betonlarda meydana gelen çatlakların görüntü işleme tekniği ile tespit edilmesiyle ilgili bir sistem geliştirmiştir. Yapılan bu çalışmada özel kamera kullanılarak çekilen beton numune görüntüleri Matlab bilgisayar programına aktarılmıştır. Burada görüntü işleme tekniklerine tabi tutulan numune görüntülerde öncelikle beton yüzeyinde belirlenen bir çatlakın bulunduğu alan 'imtool' komutu kullanılarak croplanmıştır. Daha sonra bu çalışmada da önerilen yöntem olan gri düzeyli görüntüye dönüştürülerek 'thresholding_level' komutu ile eşik değeri atanmıştır. Eşik değeri seçimi, çatlakın bulunduğu bölgenin piksel değerlerine bakılarak uygun olanın belirlenmesi ile yapılmıştır. Eşik değeri ataması sonucu bu değeri altında kalan değerler 0 değerine, üstünde kalan değerler 255 değerine dönüştürülmüştür. Son olarak Dijital Mikroskop kullanılarak beton numunelerinde belirlenen çatlak boyutlarının gerçek uzunlukları ölçülmüştür. Daha sonra Matlab bilgisayar programı kullanılarak ölçülen sonuçlar ile karşılaştırılmıştır. Bu çalışmada görüntü işleme tekniğinin başarısı ortamın ışık şiddeti, görüntü alınan mesafe, fotoğraf makinesinin belirlenen zoom ayarı, görüntü üzerine düşen gölge miktarı, görüntü alma süre aralığı ve yüksek sıcaklığın maruziyet derecesine göre değişim gösterdiği belirtilmiştir. Bu çalışmada önerilen yöntemin Çelikdemir'in çalışmasından farklı olarak gerçek zamanlı

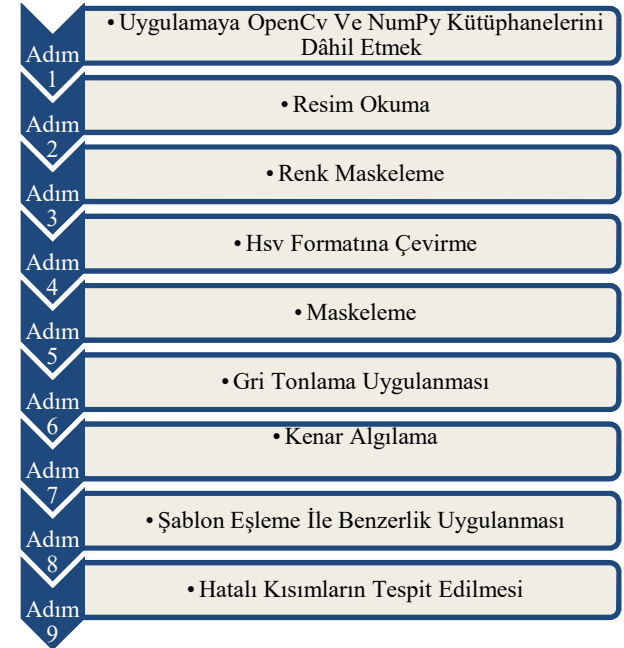
yapılmamasıdır. Önerilen yöntemde uygun ışık sağlanan ortamda ürünlerin görüntüleri alınmıştır.

Şaban(2015)'de cam üretim hatalarının görüntü işleme tabanlı bulunması ile ilgili bir sistem geliştirmiştir. İlk olarak camın sahip olduğu özellikler göz önünde bulundurularak bir prototip ölçüm ortamı tasarlanmıştır. Bu çalışmada ki önerilen çözüm yönetiminde, görüntüler alınırken ışık girilmeyen bir ortamdan alınmıştır. Aynı yöntem cam üretimindeki hata tespitinde donanım kısmında sisteme dışarıdan ışık girmesi engellenmiştir ve bu sayede ışık miktarı LED ışık kaynakları kullanılarak kendileri tarafından kontrol edilebilir hale gelmiştir. Bu çalışmada denen algoritmalar Canny ve LoG kenar belirleme algoritması, dalgacık dönüşümü, Gabor bankası ve hücresel sinir ağı modelleridir. Yapılan çalışmada Gabor Bankası yöntemi başarılı bulunmuş ve çalışmaya özel bir Gabor Bankası oluşturulmuştur. Bu çalışmadaki önerilen yöntemde de kenar bulmada Canny Algoritması kullanılmıştır.

2. Materyal ve Yöntem (Material and Method)

Bu çalışma, gıda ürünlerine ambalaj üretimi yapan bir firmada gerçekleştirilmektedir. Baskı makinelerinde basılan ambalajların yüzeyindeki hatalar Görüntü İşleme yöntemiyle tespit edilmektedir. Yukarıda bahsedilen problemin çözümü için Tablo 1'deki adımlar takip edilerek çözüme ulaşılmaktadır.

Şekil 3. Problemin Çözümü İçin Takip Edilecek Aşamalar (Stages for the solution of the problem)



OpenCV, bir resim ya da video görüntüsü içindeki anlamlı bilgilerin işleyebilmesi için çıkaran INTEL tarafından geliştirilerek BSD lisansı ile lisanslanan, C ve C++ Programlama dilleri kullanılarak geliştirilen, açık kaynak kodlu bir “Bilgisayarla Görü/ Görme” kütüphanesidir. Özellikle gerçek zamanlı çalışmalar hedef alınarak geliştirilen olması, ticari kullanımı ücretsiz olması ve Windows, Linux, MacOS X gibi farklı platformlarda kullanılabilmesi bu kütüphaneyi diğer Görüntü İşleme araçlarından bir adım daha öne çıkarmaktadır (Abduloğlu, 2017).

NumPy, Sayısal Python Programlama Dilinin kısaltılan halidir. Özellikle veri merkezli olan çalışmaların odağında bulunan bir kütüphanedir. Bu Kütüphanenin en önemli nesnelere şunlardır:

- Hızlı çalışan çok boyutlu diziler,
- Diziler veya dizi ile çalışan matematiksel işlemler,
- Dizi tabanlı veri setlerinin oluşturulması, okunması ve yazılması,
- Lineer Cebir işlemleri, rastgele sayı üretimi,
- C, C++ gibi programlama dilleri ile uyum sağlayan kodlardır (Akdoğan, 2018).

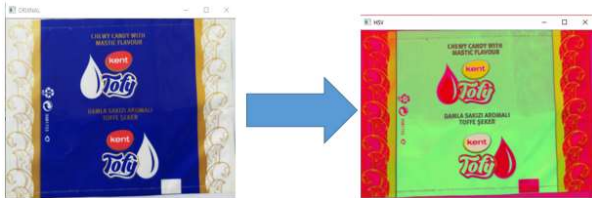
3. Önerilen Yöntem (Proposed Method)

Bu şablonla verilen bilgilerin makalenizi hazırlamada yararlı olacağını ümit ediyoruz.

3.1. Uygulamaya OpenCv Ve NumPy Kütüphanelerini Dâhil Etmek (Including OpenCv and NumPy Libraries)

Bu çalışmanın uygulama kısmında birçok işleve destek olacak OpenCV ve NumPy Kütüphaneleri dahil

Çalışmada ki mavi renk, “dusuk” ve “yüksek” olarak tanımlanan dizideki renklerin ve görüntünün HSV formatına dönüşmüş hali ile filtrelenmektedir. HSV, renk temelli ayırma işlemlerinde sıklıkla tercih edilmektedir. Görüntüyü HSV formatına çevirerek, piksellerin doygunluğuna, parlaklığına ve renk tonuna erişim sağlanmaktadır.



Şekil 4. Orijinal Görüntünün HSV Formatına Dönüştürülmüş Hali (Original Image Converted to HSV Format)

Şekil 4’de çalışmanın yapıldığı ürünün orijinal RGB (Kırmızı, Yeşil, Mavi) formatlı hali, HSV (Ton, Doymunluk, Değer) formatlı haline dönüştürülmüştür.

edilmektedir. OpenCV, tüm Görüntü İşleme fonksiyonlarında kullanılan ve bu alanın öncüsü bir kütüphanedir. OpenCV kütüphanesinin en büyük avantajlarından biri de açık kodlu bir kütüphane olmasıdır. Bu özelliğinden dolayı algoritmalar üzerinde değişiklikler yapılarak geliştirilebilmektedirler.

3.2. Resim Okuma (Picture Reading)

Uygulamada, imread() fonksiyonu resim nesnesini okumak için kullanılmaktadır. Fonksiyonun içine okunmak istenen resmin adı yazılmaktadır. Bu çalışmanın yapıldığı ürün için 100 tane farklı görüntüden oluşan veri tabanı oluşturulmaktadır. Bu veri tabanında ki görüntülerden herhangi birinin adı fonksiyon içine yazıldığında, o görüntü uygulamaya bütünleşmiş olmaktadır.

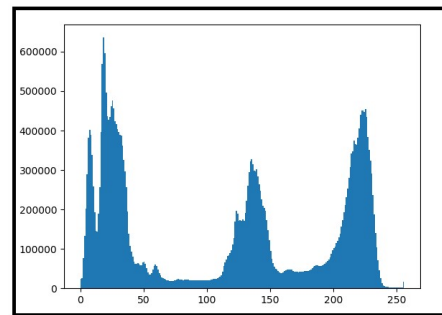
3.3. Renk Maskeleye (Color Masking)

Görüntüde zemin mavi renk olduğu için, maskeleye işleminde mavi rengin yüksek ve düşük olarak iki değeri belirlenmektedir. Bu değerler NumPy kütüphanesinin fonksiyonu olan dizilere atanmaktadır.

3.4. HSV (Ton, Doymunluk, Değer) Formatına Çevirme (Converting to HSV (Hue, Saturation, Value) Format)

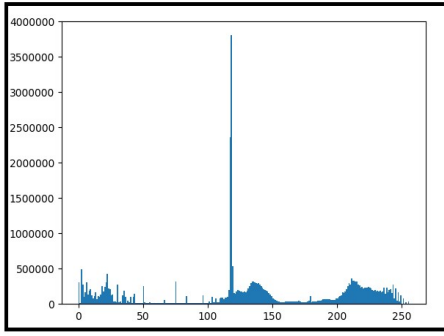
HSV (Ton, Doymunluk, Değer) renk uzayının renk bileşenlerinden renk özü, doymunluk ve parlaklık olarak tarif edilmektedir. İnsan gözüne benzer bir renk algılama olarak ifade etmek için tanımlanmaktadır. HSV, aynı zamanda renk tanıma işlemi için en uygun renk uzaylarından biri olmaktadır (Akdoğan, 2018).

Bu durumdaki görüntülerin histogram grafikleri çizilerek aradaki fark daha iyi anlaşılır olmaktadır.



Şekil 5. Orijinal Görüntünün HSV Formatına Dönüştürülmüş Hali (Histogram of the Image in RGB Format)

Şekil 5. ‘de RGB formatta olan görüntü için Histogram grafiği gösterilmektedir.

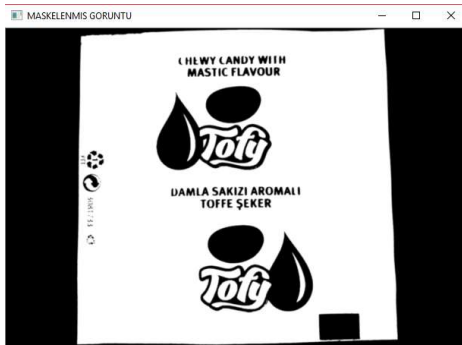


Şekil 6. Orijinal Görüntünün HSV Formatına Dönüştürülmüş Hali (Histogram of the Image in RGB Format)

Şekil 6'de HSV formatta olan görüntü için Histogram grafiği bulunmaktadır. RGB formatlı görüntünün Histogramında ki değerler eşit seviyelere göre HSV formatlı görüntüden daha düşüktür. Örneğin, 100 ile 150 seviye arasında RGB formatta en fazla 300000 değeri görülmüş ama HSV formatta bu değer 3700000'e kadar çıkmaktadır. Aynı şekilde RGB Histogramı 0 ve 600000 arası değişmekte, HSV formatı 0 ve 4000000 arası değişmektedir.

3.5. Maskeleye (Masking)

Bu çalışmada ambalaj zeminin renginden dolayı mavi alanlar maskelenmektedir. Bu maskeleye işlemini yapabilmek için, mavinin düşük ve yüksek olarak iki değerine ve görüntünün HSV formatına ihtiyaç duyulmaktadır.



Şekil 7. Maskeleye Uygulanmış Görüntü (Masked Image)

Maskeleye uygulanan görüntü Şekil 7'de mevcuttur. Bu görüntüde beyaz alanlar, orijinal görüntüde mavi rengin olduğu alanlardır. Siyah alanlar ise, mavi renk dışında kalan alanlardır. Maskeleye fonksiyonunun amacı, maskelenmek istenen rengi beyaz şekilde belirgin hale getirip, geri kalan tüm renkleri

siyah renk haline getirerek maskeleye çalışmasıdır. Maskeleye sonucu oluşan beyaz alanda hata tespiti işlemi yapılmaktadır. Maskeleye işleminden sonra, oluşan görüntüde istenilen mavi alanı orijinal haline dönüştürülmektedir.



Şekil 8. Orijinal Haldeki Maskelenmiş Görüntü (Original Masked Image)

Maskeleye işleminden sonra, beyaz renk değeri atanan pikseller eski orijinal RGB değerlerine dönüştürülerek, sadece mavi renkteki alanlar ortaya çıkmaktadır. Şekil 8'de olduğu gibi maskeleye ile RGB değeri beyaz renk (255, 255, 255) olan pikseller orijinal RGB değerlerine dönüştürülerek, görüntünün orijinal yüzeyinde ki hatalar, uygulanan filtreler sonucu görünür hale getirilmektedir.

3.6. Gri Tonlama Uygulanması (Applying Grayscale)

Maskeleye işlemi uygulanan görüntü yüzeyindeki hataları tespit etmek için kenar bulma işlemi yapılmaktadır. Bu kenar bulma işlemi daha başarılı yapabilmek için görüntüye gri tonlama uygulanmaktadır.



Şekil 9. Gri Tonlama Uygulanmış Görüntü (Grayscale Image)

Maskeleme işleminden sonra görüntüye gri tonlama uygulanan durumu Şekil 9’de verilmektedir. Çalışmada ki ürüne mavi maskeleme uygulandığı için gri tonlamada, mavi dışındaki tüm alanlar siyah renkte olmaktadır. Bu durum çalışmada hata tespiti için kolaylık sağlamaktadır.

3.7. Kenar Algılama (Edge Detection)

Bu çalışmada, ambalaj yüzeyinde olan çizik, boya, ayarsızlık gibi hatalar kenar bulma işlemi ile belirginleştirilmektedir. Kenar algılama algoritmaları, görüntü üzerindeki piksellerin renk değerlerinin bir birlerinden farklılaşması ile belirlenmektedirler. Şekil 9’de ki Gri Tonlama uygulanan görüntüde piksellerin renk değerlerindeki farklılaşmanın olduğu yerler kenar olarak algılanmakta ve OpenCv Kütüphanesi tarafından keskin bir renk geçişi uygulanmaktadır. Bu renk geçişi (gürültü olmadığı takdirde) iki farklı nesnenin başlangıç ve bitiş çizgilerini belirtmektedir. Bu geçişler kenar çizgilerini ifade etmektedir.



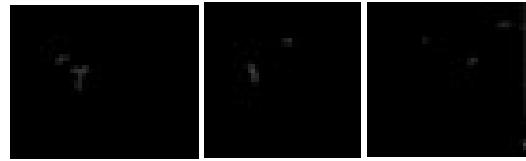
Şekil 10. Kenar Algılama Uygulanmış Görüntü (Edge Detection Applied Image)

Kenar algılama uygulanan görüntü Şekil 10’de verilmektedir. Uygulanan Kenar algılama algoritması çok hassas çalıştığından görüntüdeki istenmeyen gürültüleri de kenar olarak algılayabilmektedir. Gri (Gray) renk uzayına sahip, matris elemanları yani pikseller 0-255 arasında renk değerlerine sahiptir.

Gri Tondaki görüntü siyah-beyaz renk uzayına çevrilip 0-1 aralığında renk kodları alması uygulamada ki benzerlik işlemini kolaylaştırmaktadır.

3.8. Şablon Eşleme İle Benzerlik Uygulanması (Applying Similarity with Template Mapping)

Kenar algılama algoritması ile Şekil 10’deki görüntünün yüzeyinde ki hatalar meydana çıkmaktadır. Bu hatalara şablon eşleme yöntemiyle benzerlik uygulanarak, tespit etme işlemi uygulanmaktadır. Şablon Eşleştirme, daha büyük bir resimde bir şablon görüntünün konumunu aramak ve bulmak için kullanılan bir yöntemdir.



Şekil 11. Örnek Hata Türleri (Example Error Types)

Şekil 11’da yüzeydeki oluşan hataların maskeleme uygulanıp, gri tonlama yapıldıktan sonra kenarların algılanan halleri mevcuttur. Bu hata türlerinin görüntüleri ile ürün yüzeyinin bütününde benzerlik arama işlemi yapılmaktadır.

3.9. Hatalı Kısımların Tespit Edilmesi (Detection of Defective Areas)

NumPy Kütüphanesinin desteklediği np.where() fonksiyon ile benzerliği arama işlemi uygulanmaktadır. Benzerliği uygulanacak görüntülerin tanımlandığı “res” değişkeni ve benzerlik oranı olan eşik değeri; “threshold”, np.where() fonksiyonunda karşılaştırmaya tabi tutulmuştur. Bu karşılaştırma “res” değişkeninin “threshold” dan büyük olduğu durumları aramaktadır. Karşılaştırma sonucu bulunan durumlar “loc” adlı değişkene atanmaktadır.

Bu değişkenlerin sayısı döngüye parametre olarak atanarak, orijinal görüntüye değişken sayısı kadar kare çizdirilmektedir.

3.9. Hatalı Kısımların Tespit Edilmesi (Detection of Defective Areas)

NumPy Kütüphanesinin desteklediği np.where() fonksiyon ile benzerliği arama işlemi uygulanmaktadır. Benzerliği uygulanacak görüntülerin tanımlandığı “res” değişkeni ve benzerlik oranı olan eşik değeri; “threshold”, np.where() fonksiyonunda karşılaştırmaya tabi tutulmuştur. Bu karşılaştırma “res” değişkeninin “threshold” dan büyük olduğu durumları aramaktadır.



Şekil 12. Görüntüde Hataların Tespiti (Detecting Image Defects)

Bulunan hatalar makine operatörünün de çok kolay görebilmesi sarı dörtgen içerisinde çizdirilmektedir.

4. Araştırma Bulguları (Research Results)

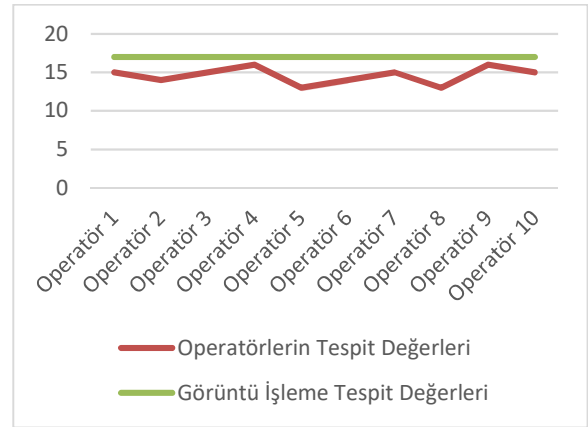
4.1. Deney Numunelerine, Operatörler Tarafından Yapılan Hata Tespiti ile Uygulamanın Yaptığı Hata Tespitinin Karşılaştırılması (Test Samples, Comparison of Error Detection by Operators and Error Detection by Application)

Çalışmanın yapıldığı ürüne ait deney numunelerinin, firmada çalışan operatörler tarafından gözle muayenesi yaptırılarak uygulama ile karşılaştırılması yapılmaktadır. Deney numunelerine ait 100 tane görüntüden oluşan bir veri tabanı oluşturulmaktadır. Bu

Karşılaştırma sonucu bulunan durumlar “loc” adlı değişkene atanmaktadır.

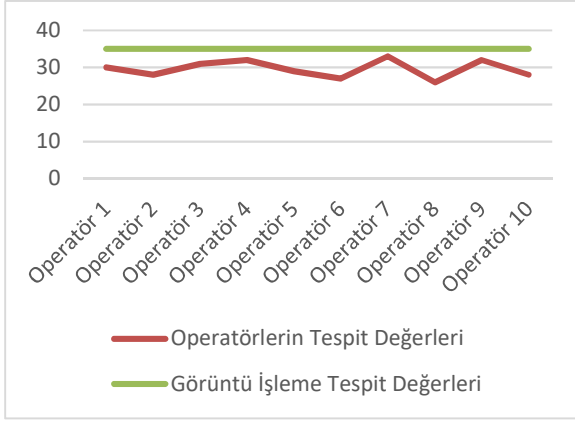
Bu değişkenlerin sayısı döngüye parametre olarak atanarak, orijinal görüntüye değişken sayısı kadar kare çizdirilmektedir.

veri tabanında 15 tane hatasız ürün, 85 tane hatalı ürün mevcut durumdadır. Bu veri tabanındaki hatalardan 17 tanesi boya, 38 tanesi ayarsızlık, 10 tanesi bıçak çizdiği, 20 tanesi parlamadan oluşmaktadır. Bu deney numuneleri 10 operatör tarafından gözle muayene edilmektedir.



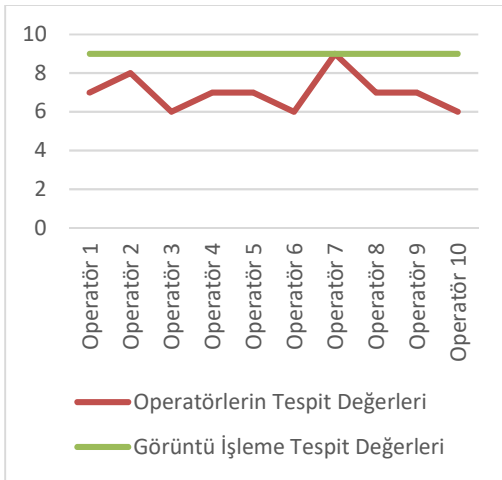
Şekil 13. Tespit Edilen Ayarsız Hata Sayısı (Number of Detected Paint Defects)

Operatörler tarafından yapılan gözle muayene sonucunun da boya hata türü için tespit değerleri Şekil 13'deki grafikte gösterilmektedir. Grafikte operatörlerin yaptığı doğru tespit değerleri ve Görüntü İşlemenin yaptığı doğru tespit değerleri verilmektedir. Operatörler boya hatasını tespit etmekte oldukça başarılı olmalarına karşın, Görüntü İşleme boya hatasında %100 başarı sağlamaktadır. Görüntü İşlemenin, boya hata tespitinde operatörlerden daha başarılı olduğu kanıtlanmaktadır.



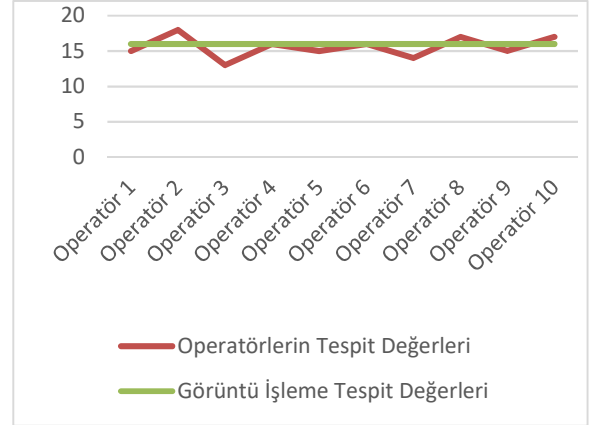
Şekil 14. Tespit Edilen Ayarsızlık Hata Sayısı
(Number of Detected Maladjustment Defects)

Operatörler tarafından yapılan gözle muayene sonucun da ayarsız hata türü için tespit değerleri Şekil 14'deki grafikte gösterilmektedir. Grafikte operatörlerin yaptığı doğru tespit değerleri ve Görüntü İşlemenin yaptığı doğru tespit değerleri verilmektedir. Ayarsız hata tespitinde operatörlerin çok dikkatli olması gerekmektedir. Görüntü İşlemenin, ayarsız hata tespitinde operatörlerden daha başarılı olduğu kanıtlanmaktadır.



Şekil 15. Tespit Edilen Bıçak Çizdiği Hata Sayısı
(Number of Detected Knife Scratch Defects)

Operatörler tarafından yapılan gözle muayene sonucun da bıçak çizdiği hata türü için tespit değerleri Şekil 15'deki grafikte gösterilmektedir. Grafikte operatörlerin yaptığı doğru tespit değerleri ve Görüntü İşlemenin yaptığı doğru tespit değerleri verilmektedir. Bıçak Çizdiği hata tespitinde operatörlerden yalnızca bir tanesi Görüntü İşleme sisteminin başarısını yakalayabilmektedir. Görüntü İşlemenin, bıçak çizdiği hata tespitinde operatörlerden daha başarılı olduğu kanıtlanmaktadır.



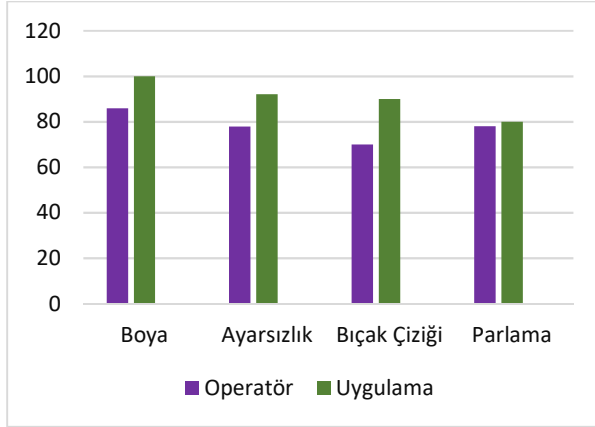
Şekil 16. Tespit Edilen Parlama Hata Sayısı
(Number of Detected Flare Defects)

Operatörler tarafından yapılan gözle muayene sonucun da parlama hata türü için tespit değerleri Şekil 16'deki grafikte gösterilmektedir. Grafikte operatörlerin yaptığı doğru tespit değerleri ve Görüntü İşlemenin yaptığı doğru tespit değerleri verilmektedir. 3 tane operatör parlama hata tespiti işleminde Görüntü İşleme sisteminin daha başarılı olmaktadır. 2 tane operatör ise Görüntü İşleme sisteminin başarısını yakalayabilmektedir. Görüntü İşlemenin, parlama hata tespiti değeri 16 iken operatörlerin ise hata tespiti ortalama değeri 15,6 olmaktadır. Görüntü İşlemenin, parlama hata tespitinde operatörlerden daha başarılı olduğu kanıtlanmaktadır.

4. Sonuçlar (Conclusions)

Bu çalışma, Pycharm Platformunda Python Programlama Dilinde OpenCV Kütüphanesi desteği ile yapılmaktadır. Çalışmada, ilk olarak renk maskeleme yapılarak mavi rengin "dusuk" ve "yukse" adlı iki dizisi oluşturulmaktadır. Görüntü HSV formatına çevrilmektedir çünkü Maskeleme işlemi yapılabilmesi için "dusuk" ve "yukse" olarak tanımlanan dizideki renk kodları ve görüntünün HSV formatına ihtiyaç olmaktadır. Maskeleme işleminden sonra mavi olan arka plan, beyaz ve diğer tüm renkler siyah olmaktadır. Maskeleme işleminden sonra beyaz renk olan pikseller orijinal RGB değerlerine dönüştürülmekte ama siyah renk olan piksellerin değerleri değişmemektedir. Kenar Algılama işlemi yapabilmek için görüntü gri tona çevrilmektedir. Kenar Algılama işlemi yapılarak görüntüde istenmeyen gürültüler bulunmaktadır. Görüntüdeki hatalı kısımların görüntüleri alınmaktadır. Bu görüntüler şablon eşleme ve benzerlik uygulaması kullanılarak benzerlik yardımıyla gerçek görüntüdeki hatalı kısımları bulunmasını sağlamaktadır. Son olarak hatalı kısımlar sarı dörtgen içerisinde belirtilmektedir. Çalışmada ürünün görüntülerinde oluşan bir veri tabanı oluşturulmaktadır. Bu veri tabanında; boya, bıçak çizdiği, ayarsızlık ve parlama hata türlerine sahip ürün

görüntüleri mevcut olmaktadır. Veri tabanında oluşturulan görüntüler, Görüntü İşleme yöntemini kullanmayarak sadece operatörlerin göz ile muayenesinden, ne oranda hata tespit edebildiklerini ölçmek amacıyla oluşturulmaktadır. Bu amaç doğrultusunda veri tabanındaki görüntüler 10 operatör tarafından göz ile hata tespiti yaptırılarak deney sonuçları kaydedilmektedir.



Şekil 17. Deneysel Sonuçları Grafiği
(Graph of Experimental Results)

Yapılan deney sonucu operatörlerin hata tespiti başarısı %78,58' tir. Operatörler sırasıyla boyadan %85.88, ayarsızlıktan %77.89, bıçak çiziziğinden %70, parlamadan %78 başarı sağlamaktadır. Bu çalışmada yapılan Görüntü İşleme yöntemiyle yapılan hata tespiti başarısı ise %90,58'dir. Elde edilen veriler doğrultusunda Görüntü İşleme yönteminin Baskı makinasına bütünleştirilmesi ile birlikte; fire miktarlarında azalma, müşteri şikâyetleri ile geri iadelerin minimum seviyeye inme, gereksiz enerji, malzeme ve işgücü harcamalarında azalma ve daha kaliteli ürünler üreterek firmaya kar sağlama gibi faydalı etkiler oluşturacağı düşünülmektedir.

Kaynaklar (References)

- Abduloğlu, Ö., 2017. "Görüntüde Logo Algılama Ve Analiz." Yüksek Lisans Tezi. Beykent Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Akdoğan, M., 2018. "Karmaşık Şekli Mamullerin Görüntü İşleme Teknikleri Kullanılarak Gerçek Zamanlı Boyut Kontrolü." Doktora Tezi. Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Çelikkemir, M., 2015. "Yüksek Sıcaklığa Maruz Kalmış Betonlarda Meydana Gelmiş Çatlakların Görüntü İşleme Yöntemi İle Tespiti." Yüksek Lisans Tezi. Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
- Demirbaş, M., 2015. "Günışığına Bağlı Aydınlatma Kontrolü İle İç Aydınlatmada Enerji Tasarrufunun Görüntü İşleme Yöntemleri Kullanılarak Gerçekleştirilmesi." Doktora Tezi. Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Sakarya.
- Eren, M., 2008. "Dokuma Kumaş Sıklıklarının Görüntü İşleme Yöntemi İle Tespitini Etkileyen Parametrelerin İncelenmesi." Yüksek Lisans Tezi. Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Tekstil Mühendisliği Ana Bilim Dalı, Kayseri.
- Hanbay, K., 2016. "Yuvarlak Örgü Makineleri İçin Görüntü İşleme Tabanlı Kumaş Hatası Tespiti Sistemi". Yüksek Lisans Tezi. İnönü Üniversitesi Fen Bilimleri Enstitüsü, Malatya.
- Köksal, K., 2017. "İnsansız Hava Araçlarında Görüntü İşleme İle Rota Tespiti Ve Otonom Takibi." Yüksek Lisans Tezi. Bülent Ecevit Üniversitesi Fen Bilimleri Enstitüsü, Zonguldak.
- Kuncan, M., 2013. "Endüstriyel Bir Makinada Görüntü İşleme Yöntemiyle Zeytinlerin Renklerine Göre Ayırt Edilmesi." Yüksek Lisans Tezi. Kocaeli Üniversitesi Fen Bilimleri Endüstrisi, Kocaeli.
- Taşçı, İ., 2015. "Yonga Levha Üretim Sistemlerinde Görüntü İşleme Tabanlı Hata Denetimi." Yüksek Lisans Tezi. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta.
- Uçar, K., 2018. "Silindirik Cisimlerin Görüntü İşleme İle Hata Tespiti." Yüksek Lisans Tezi. Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya.
- Zeki, İ., 2011. "Ürün Yüzey Hatalarının Yayınım Tabanlı Görüntü İşleme Algoritmaları İle Tespiti." Yüksek Lisans Tezi. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.