****

**KASTAMONU ÜNİVERSİTESİ**

**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**GÖRÜNTÜ İŞLEME DERSİ**

**ENDÜSTRİDE HATALI ÜRÜN DENETİMİ**

|  |  |
| --- | --- |
| **144410003**  **144410016** | **Cihangir YAZICI**  **Suha KARAAHMETOĞLU** |

İÇİNDEKİLER

[GİRİŞ 1](#_Toc469866598)

[YAPAY GÖRME NEDİR 1](#_Toc469866599)

[YAPAY GÖRME SİSTEMLERİNİN UYGULAMA ALANLARI 2](#_Toc469866600)

[MİMARİ YAPI 3](#_Toc469866601)

[SİSTEMİN AVANTAJLARI 3](#_Toc469866602)

[UYGULAMA 3](#_Toc469866603)

[MATLAB KODLARI 4](#_Toc469866604)

# GİRİŞ

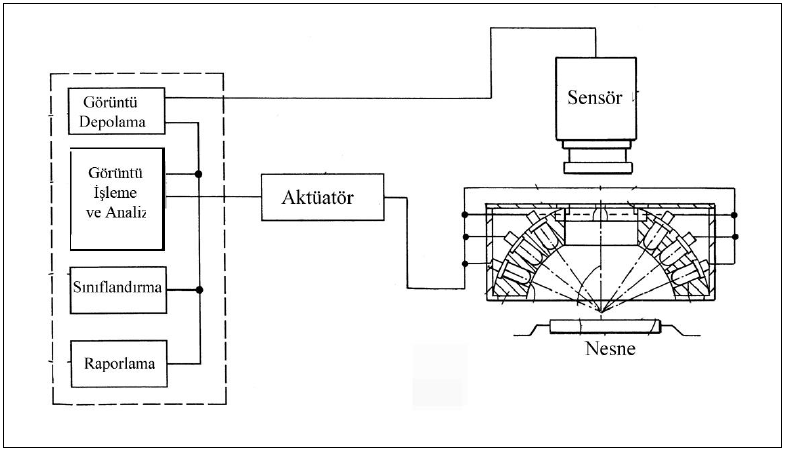
Hata tespit sistemleri üzerine çalışmalar 1980’li yıllarda başlamış, günümüze kadar gelmiş ve hemen hemen her yıl üzerinde çeşitli araştırmalar yapılmıştır. Hata tespit sistemleri, başta insan odaklı çalışan sistemler olarak sanayide yerini alsa da gelişen teknoloji ile birlikte yerini makinalara bırakmıştır. Hata tespit sistemlerinde insan odaklı uygulamalardan makina odaklı uygulamalara geçilmesinde yapay görme çalışmalarının katkısı çok büyüktür. Yapay görme teknolojilerinin gelişmesi ile hata tespit çalışmaları, çok daha hızlı ve güvenilir bir hal almıştır. Hata tespit sistemlerinde makina odaklı yani bilgisayar odaklı sistemlerin daha fazla kullanılıyor olmasının nedenleri ise; bilgisayar ve yapay görme sistemlerinin hata toleransının insanlara göre çok daha düşük olması, uzun süreler boyunca hiç durmadan çalışabilmesi ve hata tespiti için personele ihtiyaç duyulmadan tüm işlemleri eksiksiz ve hatasız gerçekleştirebilmesidir.

# YAPAY GÖRME NEDİR

Yapay görme, biyolojik insan göz yapısından yola çıkılarak geliştirilmiş bir bilgisayar sistemidir. İnsanlardaki göz yapısı oldukça karmaşık bir yapıya sahiptir. Bir saniyenin onda biri kadar bir zamanda birey; büyük boyutlarda veriyi algılayabilmekte, değerlendirebilmekte ve cisimleri farklı yönlerden ve hatta çok az görüş alanına sahip olduğu alanlarda bile tanımlayabilmektedir. İnsan retinasının, saniyede yaklaşık olarak on milyar işlem gerçekleştirdiği ve beyindeki korteks yapısının ise bu hızdan daha yüksek bir hıza sahip olduğu bilinmektedir.

Yapay görme sistemlerinin kullanıldığı ilk zamanlarda, sistemlerin karmaşıklığı ve mevcut bilgi işleme gücünün sınırlılıkları bulunmaktaydı. Günümüz teknolojileri sayesinde artık makina görmesi; tıp alanında mikro cerrahi işlemlerinden, robotik alanındaki üç boyutlu modellemelere ve hatta uzay araştırmalarına kadar uzanan çok geniş bir alanda kullanılan başarılı uygulamalara sahip bir teknoloji olma konumuna gelmiştir.

Bilgisayar görmesi olarak da geçen yapay görme terimi, çeşitli kamera ya da sensörler aracılığı ile alınan verilerin, anlamlı hale getirilerek bilgisayarlar tarafından yorumlanması işlemidir. Yapay görme sistemi; ışıklandırma, görüntü alma ve görüntü işleme bileşenlerinden oluşmaktadır. Şekil 1’de yapay görme sisteminin işleyişi blok diyagram ile gösterilmiştir.



Şekil 1 – Yapay görme sistemi işleyiş diyagramı

Endüstri alanında hata tespit sistemleri genel olarak; kameralar aracılığı ile renkli cisimleri ayırma, kumaş ve benzeri malzemelerdeki hata tespit işlemlerini gerçekleştirme, farklı boyuttaki nesneleri ayırma ya da barkoda sahip kutuların bilgilerini okuma gibi işlemlerde kullanılmaktadır. Kameradan alınan veriler, görüntü işlem ve analiz merkezine getirilir. Burada elde edilen görüntülerin kayıtları tutulur ve sınıflandırılması yapılır. Sınıflandırılma işlemi sonucunda, nesne üzerinde bir rapor oluşturulur ve kullanıcı için gerekli bilgilendirme işlemleri yapılır.

# YAPAY GÖRME SİSTEMLERİNİN UYGULAMA ALANLARI

Yapay görme teknikleri ve görüntü işleme yöntemleri teknolojinin gelişmesiyle birlikte yıllar içerisinde farklılıklar göstermiş ve farklı endüstrilerde kullanım alanı bulmuştur. Yapay görme ile hata tespit sistemleri günümüzde; otomatik PCB (baskılı devre tasarımlarının) denetimi, mobilya endüstrisinde ahşap kalite kontrolü, ambalaj sanayiinde paketleme denetimi, ürün boyut ve etiket kontrolü, plastik film, kumaş, film, pet, kağıt ve metal ürünlerin yüzey kontrolü, güneş panellerindeki hücrelerin denetimi, tıbbi şişe denetimi, temassız ısı ölçümü, dolum seviyesi kontrolü, üç boyutlu yüzey ölçümü denetimi ve modellenmesi gibi pek çok işlemde kullanılmaktadır.

Yapay görme sistemi tarafından elde edilen görüntüler, görüntü işleme algoritmaları ile işlenerek değerlendirilmektedir. Bu işlem ile gerekli malzemelerin sınıflandırılması yapılmaktadır.

# MİMARİ YAPI

Görüntü işlemede genelde 2 tip mimari yapı uygulanmaktadır. Bunlardan biri merkezi analize dayalı diğeri de dağıtık analize dönük yapıdır. Merkezi analize dayalı yapıda kameradan alınan görüntü verilerinin analiz edilmeden önce merkez sunucuda toplanması söz konusudur. Oysa dağıtık analiz yapısında kullanılan kameralarda görüntünün analiz edilerek merkez sunucuda toplanması gerçekleşir. Bu koşullar dikkate alındığında, dağıtık analiz sistemlerinin daha avantajlı olduğunu söyleyebiliriz. Özellikle, dağıtık sistemlerde alınan görüntü anında analiz edildiği için tespit edilmesi istenen olaylar daha erken belirlenebilmektedirler. Ancak dağıtık mimarilerde üzerinde görüntü işleme rutinlerini çalıştırabilecek kadar işlemci gücüne sahip olan kameralar kullanılmalı ve bu tip kameralar için gömülü yazılımlar geliştirilmelidir.

# SİSTEMİN AVANTAJLARI

* Kalite kontrolün standart hale gelmesi, üretilen tüm ürünlerin kontrolden geçmesinin sağlanması
* Kalitenin tutarlılığı ve tahmin edilebilirliğinin arttırılması
* Kontrol sisteminde otomasyona gidilerek kontrolün hızlandırılması ile birlikte üretim maliyetinin azaltılması
* Hataların istatistiklerinin tutulması ve böylece hataların kaynağının tespit edilerek önleyici bir yaklaşımının benimsenmesi ile sıfır hata idealine yaklaşılması
* Kontrolün 7 gün 24 saat devam edebilirliğinin sağlanması

# UYGULAMA

Gerçekleştirdiğimiz Matlab uygulamasında, bir fabrikanın üretim bandından geçen ürünleri elimizde bulunan hatasız orijinal ürünle bire bir karşılaştırarak, görüntülerindeki fiziksel farklılıkları ve toplamdaki fark/hata oranlarını elde edeceğiz.

# MATLAB KODLARI

function ff = kontrol(ResimOrijinal, ResimGelen);

% parametre ile dışarıdan iki adet görüntü dosyası ismi alınıyor

Rorj = imread(ResimOrijinal);

% Rorj (orijinal ürün) değişkenine 1.parametredeki ismin görüntüsü aktarılıyor

Rgel = imread(ResimGelen);

% Rgel (kontrol edilecek ürün) değişkenine 2.parametredeki ismin görüntüsü aktarılıyor

[y,g,r] = size(Rorj);

% Orijinal ürüne ait yükseklik, genişlik ve kullanılan renk katmanı değerleri alınıyor

ToplamPiksel = y\*g\*r;

% Görüntüye hata oranını hesaplayabilmek için gereken ToplamPiksel bilgisi hesaplanıyor

HataliPiksel = 0;

% HatalıPiksel bilgili başlangıçta sıfıra eşitleniyor

Rson = Rgel;

% Rson (sonuç görüntüsü) Rgel'den aktarılıyor

subplot(1,3,1); imshow(Rorj);title('Orijinal Ürün');

% Orijinal ürün görüntüleniyor

xlabel(ResimOrijinal);

% Orijinal ürüne ait görüntü dosyasının ismi görüntüleniyor

subplot(1,3,2); imshow(Rgel);title('Kontrol Edilen Ürün');

% Kontrol edilecek ürün görüntüleniyor

xlabel(ResimGelen);

% Kontrol edilecek ürüne ait görüntü dosyasının ismi görüntüleniyor

% İki ürün arasındaki her bir piksel karşılıklı olarak eşitlik kontrolü yapılacak

for satir=1:y

% satir değişkeni 1'den y (yükseklik) değerine kadar döngüye sokuluyor

for sutun=1:g

% sutun değişkeni 1'den g (genişlik) değerine kadar döngüye sokuluyor

PikselHatali = false;

% Başlangıç olarak kontrol edilecek pikselde hata olmadığı farzediliyor

for renk=1:r

% renk değişkeni 1'den r (renk katmanı) değerine kadar döngüye sokuluyor

if Rorj(satir,sutun,renk) ~= Rgel(satir,sutun,renk)

% Orijinal görüntü ile kontrol edilecek görüntünün aynı koordinatlarının birbirinden farklı olması durumunda

PikselHatali = true;

% pikselin hatalı olduğu anlaşılıyor

end

end

if PikselHatali == false

% Pikselde hata bulunmamışsa

for renk=1:r

% işaretleme yapmak için renk döngümüz tekrar oluşturuluyor

Rson(satir,sutun,renk) = 0;

% kontrol edilen koordinat değeri sonuç görüntüsünde 0 (siyah)'a eşitleniyor

end

else

HataliPiksel = HataliPiksel+r;

% Pikselde hata bulunmuşsa HataliPiksel değişkenimize renk katmanı değeri kadar değer ekliyoruz

end

end

end

HataOrani = HataliPiksel / ToplamPiksel \* 100;

% Kontrolümüz bittiği için iki görüntü arasında oluşan HataOrani'ni hesaplıyoruz

str\_hataorani = sprintf('Hata Oranı : %% %2.2f \n',HataOrani);

% str\_hataorani değişkenine istediğimiz formatta bilgi aktarıyoruz

subplot(1,3,3); imshow(Rson);title('Sonuç');

% Kontrol sonucunda iki görüntü arasındaki farklardan oluşan sonuç görüntüsü görüntüleniyor

xlabel(str\_hataorani);

% Görüntünün altına hata oranımızı yazdırıyoruz

fprintf('Toplam Piksel : %7d \n',ToplamPiksel);

% Komut ekranına ToplamPiksel yazdırılıyor

fprintf('Hatalı Piksel : %7d \n',HataliPiksel);

% Komut ekranına HataliPiksel yazdırılıyor

fprintf('Hata Oranı : %% %2.2f \n',HataOrani);

% Komut ekranına HataOrani yazdırılıyor