

Görüntü İşleme Yöntemleri ile Üretim Hattı Analizi

Arda ALHAN, Evrim Arda KALAFAT

Proje Tanımı

- Fabrikalarda üretim hatlarından çıkan ürünlerin analizi yüksek çözünürlüklü, yüksek FPS (Frame per Second) özellikli kameralar ile yapılmaktadır.
- Bir üretim hattından çıkan üç farklı tür ürünün, video kayıtlı boyunca kaçar adet üretildiğinin tespit edileceği bir sistem geliştirilecektir. OpenCV (Open Computer Vision) kütüphanesi kullanılacaktır.

```
eTransaction = new Structure  
...  
AssistanceContDBenefit, InSDM  
...  
Assistance = new  
...  
Assistance
```

Kullanılan Araçlar

- Microsoft Visual Studio Community

Kullanılan Kütüphane

- OpenCV

Gerçekleme Yöntemi:

```
#include <opencv2/opencv.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/video/background_segm.hpp>
#include <opencv2/video/tracking.hpp>
#include <opencv2/video/video.hpp>
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

using namespace std;
using namespace cv;

//-----RENKLER-----//
const Scalar COLOR_BLACK = Scalar(0.0, 0.0, 0.0);
const Scalar COLOR_WHITE = Scalar(255.0, 255.0, 255.0);
const Scalar COLOR_BLUE = Scalar(255.0, 0.0, 0.0);
const Scalar COLOR_GREEN = Scalar(0.0, 200.0, 0.0);
const Scalar COLOR_RED = Scalar(0.0, 0.0, 255.0);
//-----RENKLER-----//
```

- İlk olarak kütüphanelerimizi ve renklerimizi tanımladık.

```
Transaction = new Structure  
AssistanceContDBenefit, InSD  
entager, $00$1  
Assistance = new  
AssistanceContDBenefit, InSubsid  
$00$1  
Assistance
```

```
int main() {  
  
    VideoCapture capture("C:\\Users\\evrim\\Desktop\\uretimHatti.mp4");  
  
    Mat Frame1, Background, foreground;  
  
    int nesnesayisi = 0, küçük = 0, büyük = 0, daire = 0, test = 0;  
  
    if (!capture.isOpened()) {  
        cout << "Error: Video acilamadi" << endl;  
        return -1;  
    }  
  
    capture.read(Background);
```

- Videoyu açtık ve arka planı için ilk karesini okuttuk. Ayrıca daha sonra kullanmak için birkaç tanımlamayı ayarladık.

```
Transaction = new Structure  
AssistanceContDBenefit, InSD  
centage, $00$1  
Assistance = new  
AssistanceContDBenefit, InSubsid  
er, $00$1  
Assistance
```

```
//Frame1 ve arkaplan kopyalama  
Mat Frame1Copy = Frame1.clone();  
Mat BackgroundCopy = Background.clone();  
  
// Grilestirme  
cvtColor(Frame1Copy, Frame1Copy, CV_BGR2GRAY);  
cvtColor(BackgroundCopy, BackgroundCopy, CV_BGR2GRAY);  
// imshow("CVTCOLOR", Frame1Copy);  
  
//Blurlama  
GaussianBlur(Frame1Copy, Frame1Copy, Size(9, 9), 0);  
GaussianBlur(BackgroundCopy, BackgroundCopy, Size(9, 9), 0);  
// imshow("BLUR", Frame1Copy);
```

- Frame1 ve Background'ın kopyalarını oluşturduk. Sonra ikisini de grileştirip bulanıklaştırdık.

```
Transaction = new Structure  
AssistanceContDBEREET, InSD  
entager; $0051  
Assistance = new  
Assistance(SNIDBereET, InSubsid  
$0051:  
Assistance;
```

```
// fark  
Mat imgDifference;  
absdiff(Frame1Copy, BackgroundCopy, imgDifference);  
// imshow("DIFFERENCE", imgDifference);  
  
// Threshold  
Mat imgThresh;  
threshold(imgDifference, imgThresh, 64, 255.0, CV_THRESH_BINARY);  
// imshow("THRESHOLD", imgThresh);  
  
// Genisletme  
Mat structuringElement9x9 = getStructuringElement(MORPH_RECT, Size(9, 9));  
dilate(imgThresh, imgThresh, structuringElement9x9);  
  
// BackgroundSubtractorMOG2  
BackgroundSubtractorMOG2 mog;  
mog(imgThresh, foreground);  
// imshow("Fore", foreground);
```

- Arka planla o anki karenin farkını aldık. Bir Threshold yerleştirdik. Aralarındaki farkı(nesneyi) genişlettik. Son olarak MOG2 filtresini uyguladık.

```
Transaction = new Structure  
AssistanceContDBenefit, InSDS  
centage, 150051  
Assistance = new  
AssistanceContDBenefit, InSubsid  
150051  
Assistance
```

```
//-----PARAMETRELER-----//  
SimpleBlobDetector::Params params;  
  
params.minThreshold = 10;  
params.maxThreshold = 500;  
  
params.filterByArea = true;  
params.minArea = 200;  
  
params.filterByCircularity = false;  
params.minCircularity = 0.7;  
  
params.filterByConvexity = false;  
params.minConvexity = 0.87;  
  
params.filterByInertia = false;  
params.minInertiaRatio = 0.01;  
//-----PARAMETRELER-----//
```

- SimpleBlobDetector için parametreleri hazırladık.


```
transaction = new Structure
AssistanceContDBREMIT, InSD
ptager, $500$1
Assistance + new
Assistance (SimpleBlobDetector, InSubsid
Assistance
Assistance
```

```
//-----SimpleBlobDetection-----//
SimpleBlobDetector detector(params);
vector<KeyPoint> keypoints;

detector.detect(foreground, keypoints);

for (int i = 0; i < keypoints.size(); i++)
{
    test++;
    if (((15 < floor(keypoints[i].size)) && (floor(keypoints[i].size < 19))) && ((32 < floor(keypoints[i].pt.x)) && (floor(keypoints[i].pt.x) < 36)) &&
        ((216 < floor(keypoints[i].pt.y)) && (floor(keypoints[i].pt.y) < 220)))
    {
        nesnesayisi++;
        küçük++;
    }
    else if (((29 < floor(keypoints[i].size)) && (floor(keypoints[i].size < 33))) && ((50 < floor(keypoints[i].pt.x)) && (floor(keypoints[i].pt.x) < 53))
        && ((199 < floor(keypoints[i].pt.y)) && (floor(keypoints[i].pt.y) < 203)))
    {
        nesnesayisi++;
        büyük++;
    }
    else if (((36 < floor(keypoints[i].size)) && (floor(keypoints[i].size < 40))) && ((96 < floor(keypoints[i].pt.x)) && (floor(keypoints[i].pt.x) < 100))
        && ((145 < floor(keypoints[i].pt.y)) && (floor(keypoints[i].pt.y) < 149)))
    {
        nesnesayisi++;
        daire++;
    }
    std::cout << "Sekilin " << " Kordinat X: " << keypoints[i].pt.x << " Y: " << keypoints[i].pt.y << " Boyut:" << keypoints[i].size << "\n";
}

Mat im_with_keypoints;
drawKeypoints(imgThresh, keypoints, im_with_keypoints, COLOR_RED, DrawMatchesFlags::DRAW_RICH_KEYPOINTS);
imshow("keypoints", im_with_keypoints);
//-----SimpleBlobDetection-----//
```

- SimpleBlobDetection sınıfını kullanarak nesnelere koordinatlarını bulduk ve koordinatlarına göre kategorize ederek üretim bandından geçen nesnelere saydırttık.

```
Transaction = new Structure  
AssistanceContDBenefit, InSD  
entager, $00$1  
Assistance = new  
AssistanceContDBenefit, InSubsid  
$00$1:  
Assistance
```

```
//-----Bitis-----//  
char c = (char)waitKey(10);  
if (c == 27) break;  
//-----Bitis-----//  
}  
  
cout << endl << "Toplam nesne sayisi: " << nesnesayisi << endl;  
  
cout << "Kucuk nesne sayisi: " << küçük << endl;  
cout << "Sari nesne sayisi: " << büyük << endl;  
cout << "Yuvarlak nesne sayisi: " << daire << endl;  
  
capture.release();  
return 0;
```

- Video yakalamayı bitirdik.
SimpleBlobDetection'ı kullanarak saydığımız nesne sayısını ve türünü ekrana bastırdık.

```
Transaction = new Structure  
AssistanceContBenefit, InSD  
centage, $500$1  
Assistance = new  
AssistanceContBenefit, InSubsid  
er, $500$1  
Assistance
```

Teşekkürler.