

Emgu CV를 이용한 자동차 번호판 자동 인식 프로그램 구현에 관한 연구

김남우* · 허창우*

*목원대학교

Study on the panorama image processing
using the SURF feature detector and technicians.

Nam-woo Kim* · Chang-Wu Hur**

*MOKWON University

E-mail : gotree94@gmail.com

요 약

자동차 번호판 인식은 대중적인 감시 기술 중의 한 종류로서, 주어진 비디오나 영상 내 광학문자 인식을 수반한다. 고속도로나 국도 상에 과속 단속 시스템, 재형 건물이나 유통센터 및 주차장 등에서 주차 정산 시스템, 고속도로 톨 게이트에서 hi-pass 에러 및 불법 도주 차량 단속 시스템, 전국 주요 도로 불법 주 정차 단속 시스템, 공공기관, 기업 출퇴근 시간 확인 및 외부 차량 안내 시스템 등의 지능형 교통 시스템(ITS)이나 국도 상에 범위 차량 검거 시스템, 사건 발생 시 주요 도로상에 설치된 CCTV를 통해 용의 차량 이동 추적 시스템, 이동식 범죄 차량 조회, 버스에 탑재된 버스 전용차선 위반 단속들의 지능형 방법 시스템 등에 활용하고 있다. 번호판 인식은 자동차 번호판 국부화, 번호판의 크기, 차원, 명암대비, 밝기를 조정하는 정규화, 개별문자를 얻어내는 문자 분할, 문자를 인식하는 광학 문자 인식, 번호판의 형태, 크기, 위치 등이 연도별, 지역별로 차이가 있는 번호판들의 데이터베이스를 비교하여 구문 분석을 하는 절차를 거친다. 본 논문에서는 EmguCV를 이용하여 구현한 번호판 감지를 수행하여 위치를 찾아내고, 오픈 소스 광학 문자 인식 엔진으로 잘 알려져 있는 테서렉트 OCR을 이용하여 번호판의 문자를 인식하는 자동 인식 프로그램을 구현하고 기술하였다.

키워드

EmguCV, OCR, Tesseract OCR, LPR

1. 서론

Emgu CV는 OpenCV의 닷넷 래퍼로, 교차 플랫폼 영상처리 라이브러리이다. 즉 Emgu CV는 닷넷에서 쓸 수 있게 한 OpenCV라고 할 수 있다. c#, 비주얼베이직, 아이언파이썬,VC++ 등 몇몇 언어를 지원한다. 모노로 Emgu CV를 컴파일할 수 있으며 윈도우, 맥 OS X 뿐만 아니라 안드로이드 디바이스 아이폰, 아이패드등의 모바일 플랫폼에서도 실행할 수 있다. [1].

Emgu CV의 목표중의 하나는 닷넷 개발자가 사용이 간단한 컴퓨터 비전 기반을 제공함에 있

으며, 상당히 정교한 비전 애플리케이션을 빠르게 구축할 때 도움을 주는데 있다. Emgu CV라이브러리는 비전 관련한 영역 중 다수를 포괄하는데, 공장 생산품 검사, 의학 영상, 사용자 인터페이스, 카메라 교정, 스테레오 비전, 로봇공학도 이 영역에 포함된다. 컴퓨터 비전과 기계 학습은 종종 밀접한 관련이 있는데, Emgu CV에도 OpenCV 영상처리 라이브러리의 범용 기계 학습 라이브러리가 모두 담겨 있다.

Emgu CV는 제품이나 어플리케이션을 오픈소스로 하되, 무료로 배포해야 하는 GPL라이선스와 비공개 소스로 배포할 수 있는 상업용 라이선스

가 있다.

문서나 예제, 자료들은 OpenCV에 비해 많지 않는 부분이 있지만, Emgu CV가 OpenCV의 래퍼라는 부분에 있어서 OpenCV를 이해한다면 라이브러리 사용에 있어서 큰 도움이 될 것이다. 이런 부분의 단점에도 불구하고 성능 면에서는 OpenCV와 동일성을 유지하면서 라이브러리 사용에 있어서는 닷넷의 편의성을 사용할 수 있는 장점이 있다.

자동차 번호판 인식은 감지보다 더 많이 복잡한 작업으로, 영상 안에서 자동차 번호판 위치를 찾은 후에 자동차 번호판 영상을 텍스트로 변환한다. 이 논문에서는 광학문자 인식관련 기술 이용 및 자동 인식 프로젝트를 구현하였다.

II. 본론

자동차 번호판 인식(License Plate Recognition)은 대중적인 감지 기술 중의 한 종류로서, 주어진 비디오나 영상 내 OCR(Optical Character Recognition)을 수반한다. LPR시스템은 낮과 밤, 날씨에 되도록 영향을 덜 받을 수 있도록 카메라는 적외선이나 플래시를 켜서 자동차 번호판 좋은 품질의 영상을 얻을 수 있도록 구성해야하며, 각 지역 및 발행 시기에 따라서 번호판이 다른 불확실성이 잔존하기 때문에 이에 맞는 번호판 처리 및 인식을 위한 알고리즘을 설계해야 한다. 일반적으로 다음과 같은 일반적인 알고리즘을 사용한다.

2.1 LPR 알고리즘

자동차 번호판 인식의 핵심 아이디어는 다음 두 단계로 나눌 수 있다.

첫째, 자동차 번호판 감지를 수행한다.

둘째, 번호판 문자와 숫자를 가져오기 위해 광학 문자 인식을 영역에 사용한다.

더 구체적으로, 완전한 LPR 시스템에서는 자동차 번호판을 인식하기 위해 다음 다섯 가지 기본 절차를 따라야 한다.

자동차 번호판 국부화 : 주어진 영상에 있는 번호판을 감지하고 분리한다.

정규화 : 번호판의 크기, 차원, 명암대비, 밝기를 조정한다.

문자 분할 : 영역 내 각 개별 문자를 얻는다.

광학 문자 인식 : 문자를 인식한다.

구문 분석 : 번호판 규칙을 비교한다.

[2][3]



그림 1. 정규화, 문자분할, OCR 과정

2.2 OCR

광학 문자 인식으로 영상에 포함된 글자를 문자로 변환한다. 보통 스캔한 영상의 글자에는 인쇄한 글자, 손으로 쓴 글자 등이 포함된다. 글자를 스캔해 영상을 분석한 다음에 문자를 얻는다는 과정을 거치게 된다. 인쇄한 글자나 손으로 쓴 글자를 디지털로 처리한 다음에 컴퓨터에 저장하고, 검색하고, 띄우는 일은 일상적인 컴퓨터 비전과 패턴 인식 연구 분야이다.

초창기 광학 문자 인식 시스템에서는 단어 중 한 글자씩 차례로 인식하는 방법을 썼었고, 문자 인식 처리 과정에서 글꼴도 한 종류만 쓸 수 있었다. 이제는 대다수 OCR 시스템이 거의 모든 글꼴을 아주 정확히 처리한다. 패턴 인식 기술의 발달로 인부 선도적인 OCR 시스템은 잘 정돈된 글들 속에 있는, 글자가 아닌 요소도 인식 처리한다. 글자가 아닌 요소로는 문단, 다단, 영상 등이 있다.

2.3 테서렉스 OCR

테서렉트는 유명한 OCR 엔진으로서, 레이 스미스가 HP 연구소에서 1985년부터 1995년까지 개발했다.[4] 테서렉트 OCR 엔진은 60 이상의 언어와 서로 다른 영상 포맷을 지원한다. 1995년 UNLV 정확도 테스트에서 상위 3위 안에 드는 OCR 엔진 중 하나였다. 그 후 약간 개선을 하였고, 2006년부터는 구글이 대폭적으로 개선했다. 이제는 가장 정확한 오픈소스 광학 문자 인식 엔진이 되었다.

테서렉트는 윈도우, 맥 OS X, 리눅스, 안드로이드, iOS에서 작동할 수 있다. 하지만 엔진의 경우에는 명령행 도구만 제공한다. EmguCV는 프로젝트에서 EmguCV 라이브러리를 통해 테서렉트를 사용할 수 있도록 OCR 엔진인 테서렉트의 닷넷 래퍼인 tessnet2를 제공한다.[5] 다만 이 OCR 래퍼는 윈도우에서만 실행할 수 있으며, 리눅스를 비롯한 다른 운영체제와 호환하지 않는다.

III. 결론

본 논문에서는 유럽 자동차 번호판을 기준으로 가정한다. 이런 경우 (3,10)의 범위에 있는 종횡비와 문자가 A부터 Z까지와 0부터 9까지만 있는 직사각형만 고려한다. 종횡비가 특정 범위에 있지 않으면 자동차 번호판이 아니라고 판단한다.



그림 2. 자동차 번호판 샘플

더 잘 적응하게 하려는 데 있다. 배경이 흰 번호판에서만 작동하는데 경계값을 설정했기 때문이다.

```
private static Image<Gray, Byte> GetWhitePixelMask(Image<Bgr, byte> image)
{
    using (Image<Hsv, Byte> hsv = image.Convert<Hsv, Byte>())
    {
        Image<Gray, Byte>[] channels = hsv.Split();
        try
        {
            //channels[1] is the mask for saturation less than 40,
            this is the mask for either white or black pixels
            channels[1].ThresholdBinaryInv(new Gray(40), new Gray(255));

            //channels[2] is the mask for bright pixels
            channels[2].ThresholdBinary(new Gray(200), new Gray(255));

            CvInvoke.cvAnd(channels[1], channels[2], channels[0],
                IntPtr.Zero);
        }
        finally
        {
            channels[1].Dispose();
            channels[2].Dispose();
        }
        return channels[0];
    }
}
```

```
public LicensePlateDetector()
{
    //create OCR engine
    _ocr = new Tesseract(@"C:\Emgu\emgu-cv-windows-universal-gpu2.4.9.1847\Emgu.CV.OCR\", "eng", Tesseract.OcrEngineMode.OEM_TESSERACT_CUBE_COMBINED);
    _ocr.SetVariable("tessedit_char_whitelist", "ABCDEFGHIJKLMNOPQRSTUVWXYZ-1234567890");
}
```

3.1 GetWhitePixelMask

이 메소드는 주어진 영상에 대한 흰색 화소 마스크를 계산한다. 이 메소드의 목표는 컬러 영상을 그레이스케일 영상으로 변환함으로써 변화에

3.2 DetectLicensePlate

감지방식으로는 캐니 감지기를 사용한다.

```
public List<String> DetectLicensePlate(
    Image<Bgr, byte> img,
    List<Image<Gray, Byte>> licensePlateImagesList,
    List<Image<Gray, Byte>> filteredLicensePlateImagesList,
    List<MCvBox2D>
    detectedLicensePlateRegionList)
{
    List<String> licenses = new List<String>();
    using (Image<Gray, byte> gray = img.Convert<Gray,
```

```

Byte>())
//using (Image<Gray, byte> gray =
GetWhitePixelMask(img))
using (Image<Gray, Byte> canny = new
Image<Gray, byte>(gray.Size))
using (MemStorage stor = new MemStorage())
{
    CvInvoke.cvCanny(gray, canny, 100, 50, 3);
    Contour<Point> contours = canny.FindContours(

Emgu.CV.CvEnum.CHAIN_APPROX_METHOD.CV_CHAI
N_APPROX_SIMPLE,

Emgu.CV.CvEnum.RETR_TYPE.CV_RETR_TREE,
        stor);
        FindLicensePlate(contours, gray, canny,
licensePlateImagesList, filteredLicensePlateImagesList,
detectedLicensePlateRegionList, licenses);
    }
    return licenses;
}
    
```

3.3 FindLicensePlate

이 메소드는 주어진 영상 내 자동차 번호판 영역을 찾을 때 사용한다. 사각형 간지 코드와 거의 같다.

◎ 외곽선 영역을 계산한다. 값이 너무 작으면 생략한다. 남겨진 영역이 자동차 번호판 영역일 가능성이 있다고 간주한다.

◎ 외곽선의 점을 합해 계산한다. 값이 너무 작으면 생략한다. 남겨진 문자만 다음 단계로 갈 수 있도록 자동차 번호판은 최소한 6문자 혹은 7문자를 가져야 한다.

◎ 차후 계산을 위해 남은 영역을 정규화한다.

◎ 적절한 값의 범위안에 있는 종횡비를 갖는 직사각형만 고려한다.

프로그램을 실행한 결과 다음 그림과 같이 오른쪽 패널에서 OCR 결과를 볼 수 있고, 자동차 번호판 영역과 필터 영역을 볼 수 있다.



그림 3. 자동차 번호판 광학문자 인식 결과

본 논문에서는 Emgu CV를 이용해 자동차 번호판 인식과 광학문자 인식을 하는 방법을 구현하였다.

참고문헌

- [1] Shin shi, "Emgu CV와 테서렉트 OCR로 하는 컴퓨터 비전 프로그래밍", 에이콘출판사
- [2] M. Ko, "Effective License Plate Character Recognition Based on Geometric Invariant Features," Ph. D. dissertation, Kyungpook National University, 2004.
- [3] B. Enyedi, L. Konyha, and K. Fazekas, "Real Time Number Plate Localization Algorithms," J. of Electrical Engineering, Vol.57, No.2, pp.69-77, 2006.
- [4] <http://code.google.com/p/tesseract-ocr>
- [5] <http://www.pixel-technology.com/freeware/tssnet2>