

# OpenCV 와 Convolutional neural network를 이용한 눈동자

## 모션인식 시스템 구현

이승준 · 허승원 · 이희빈 · 유운섭  
한경대학교 전기전자제어공학과, IITC

### Implementation to eye motion tracking system using OpenCV and convolutional neural network

Seung Jun Lee · Seung Won Heo · Hee Bin Lee · Yun Seop Yu  
Dept. of Electrical, Electronic, Control Eng. and IITC, Hankyong National University  
E-mail : tmdwns\_\_4534@naver.com

#### 요 약

본 논문에서는 이전에 발표한 “Convolutional neural network를 이용한 눈동자 모션인식 시스템 구현”을 OpenCV의 눈 영역 검출을 이용해 보완하고 신경망 구성 및 연산에 Numpy를 이용한 눈동자 모션인식 시스템에 대해 소개한다. Numpy를 이용해 신경망을 구성하고 OpenCV를 이용해 얼굴영역과 눈영역을 검출해 내고 눈영역 이미지를 이용해 신경망을 학습하고 학습된 신경망으로 눈동자의 움직임을 인식한다. 이 시스템은 SOC 보드인 DE1-SOC 보드 위에 구현했다.

#### ABSTRACT

Previously presented “Implementation to pupil motion recognition system using convolution neural network” is improved. Using OpenCV, face and eye areas are detected, and then configure the neural network using Numpy. This pupil motion recognition system is based on the Numpy for configuring and calculating the neural network. This system is implemented on DE1-SOC.

#### 키워드

인공신경망, OpenCV, CNN, 눈동자 추적

#### 1. 서 론

인공 신경망이란 인간의 신경망을 참고해서 개발된 머신러닝의 기법이다[1]. OpenCV 는 주로 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리이고, Numpy는 행렬이나 대규모 다차원 배열을 쉽게 처리 할수 있게 해주는 파이썬 라이브러리 이다. 이 신경망을 이용한 한 예로써 루게릭병 환자와 같이 몸을 움직이지 못하는 환자를 위해 눈동자를 추적하여 의사소통 시스템에 대한 연구가 활발히 진행되고 있다[2,3]. 이전에 64비트 Linux OS 환경에서 Tensorflow를 이용해 신경망 생성 및 학습하고 학습된 신경망을 통하

여 눈동자의 위치를 파악하는 시스템을 제안했다 [4]. 이 시스템은 32비트 OS 환경을 대개 채택하는 임베디드시스템에는 적용할 수 없기 때문에 32비트 OS 환경에서 적용 가능한 시스템이 요구 된다.

본 논문에서는 32비트 Linux OS환경에서 OpenCV를 사용해 눈영역을 검출하고, Numpy를 사용해 신경망 구성 및 연산을 하여 눈동자 움직임을 추적할 수 있는 시스템 구현에 대해서 소개 한다.

## II. 본론

### 1) OpenCV를 이용한 눈동자 검출.

본 논문에서는 OpenCV를 이용해 검출된 눈영역을 학습 및 입력 데이터로 사용한다. OpenCV는 많은 영상처리에 유용한 Library를 제공하는데 그중 harr cascade를 사용했다. 우리가 필요한 부분은 환자의 눈동자이므로 우선 프레임에서 환자의 얼굴을 검출하고 그 얼굴 안에서 눈을 검출해 낸다. 움직이지 못하는 환자를 위한 연구이고 매프레임 마다 눈동자를 검출한다면 비효율을 적이기 때문에 눈 검출을 매프레임이 아닌 한번만 진행 하도록 했다.

### 2) 입력 데이터

신경망으로 입력, 학습 되는 데이터들은 OpenCV를 이용해 검출된 눈영역이다. 환자마다 눈의 크기가 같지도 않고 같은 환자이더라도 때때로 다른 크기로 눈동자 영역이 검출될 수도 있기 때문에 패딩 과정을 거쳐야 하는데 본 논문에서는 입력과 학습되는 데이터가 30x30의 크기를 같도록 패딩과정을 거쳤다.

### 3) 학습 데이터

학습 데이터로 쓰이는 데이터들은 그림과 같이 눈동자가 오른쪽, 가운데, 왼쪽 각각 100 개씩 총 300개의 이미지를 학습데이터로 사용하였고 패딩 후에 이미지의 크기는 30x30이다. 또한 학습 알고리즘으로 ADAM 최적화 알고리즘을 사용하였다.

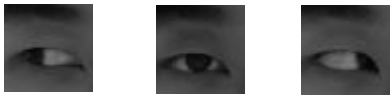


그림 1. 눈동자 추적 사진(오른쪽, 가운데, 왼쪽)

### 4) 신경망 구성

이 시스템에서 신경망은 합성곱 계층 2단계, 완전연결 계층 2단계를 사용한 컨볼루션 신경망(convolutional neural network; CNN)이 사용된다. 첫 번째 CNN계층의 필터의 크기는 5x5이고 활성화함수로 Relu함수를 사용한 후에 2x2 Maxpooling 을 거쳐 다음 계층으로 입력된다. 두 번째 CNN계층의 필터의 크기는 5x5이고 활성화함수와 풀링방식은 앞 계층과 같다. 두 번째 CNN계층의 출력이 다음 완전연결계층의 입력으로 들어가기 위해서 형변환 과정을 거친다. 첫 번째 완전연결계층의 가중치의 크기는 10816x1024 이고 활성화함수로 Relu함수를 사용한다. 마지막 완전연결계층에서 가중치의 크기는 1024x3 이고 활성화함수로 Softmax함수를 사용해 최종 출력을 얻는다.

## III. 결과

눈영역에 대한 이미지 300개를 10번 학습해서 얻어진 인공신경망에 30개의 이미지를 테스트한 결과는 표1과 같다. 신경망의 출력이 각 위치에 대해 97%이상의 확률로 각 위치에 있다고 인식했다.

표 1. 눈동자 위치 추적 정확도

	왼쪽	가운데	오른쪽	총합
샘플갯수	10	10	10	30
검출갯수	10	10	10	30
확률	98.01	97.78	99.5	98.43

## IV. 결 론

본 논문에서 눈동자의 위치를 파악해주는 컨볼루션계층이 2단계, 완전연결계층이 2단계로 구성된 인공신경망을 소개하였다. 신경망에 입력되는 눈영역을 추출하기 위해서 OpenCV의 harr cascade를 사용했고 추출된 눈영역을 30x30으로 패딩했다. 또한 30x30의 그레이 변환된 이미지 300개를 총 10회 학습을 하였고 학습 알고리즘으로는 ADAM 최적화 알고리즘을 사용하였다. 본 논문에서 소개한 인공신경망으로 눈동자 위치를 인식한 결과를 보면, 30 개 이미지를 테스트했을 때 거의 정확하게 검출되는 성능을 확인했다.

## References

- [1] A. K. Jain, J. Mao, and K. M. Mohiuddin, *Computer*, vol. 29, no. 3, pp. 31-44, Mar 1996.
- [2] S. Dongre and S. Patil, *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 7, pp. 156-160, 2015.
- [3] C. Gou, Y. Wu K. Wang, F.-Y. Wang, and Q. Ji, *Pattern Recognition*, vol. 67, pp. 23-31, 2017.
- [4] 이승준, 허승원, 이희빈, 유윤섭, "Convolutional neural network를 이용한 눈동자 모션인식 시스템 구현," 한국정보통신학회 2018년 춘계학술대회, pp.703-704, 2018.