

신경망을 이용한 차량의 주행방향과 장애물 인식에 관한 연구

Recognition of Driving Direction & Obstacles Using Neural Network

김명수*, 양성훈(부산대 생산기계공학과 대학원), 이석(부산대 기계공학부)

ABSTRACT

In this paper, an algorithm is presented to recognize the driving direction of a vehicle and obstacles in front of it based on highway road image. The algorithm employs a neural network with 27 sub sets obtained from the road image as its input. The outputs include the direction of the vehicle movement and presence or absence of obstacles. The road image, obtained by a video camera, was digitized and processed by a personal computer equipped with an image processing board.

Keyword : Neural Network, Back-Propagation, Driving Direction, Obstacles

1. 서론

최근에 열악한 도로 사정과 차량의 급증으로 인해 차광사고가 증가 추세에 있다. 이러한 사고의 예방과 방지를 위해 차량의 추돌 방지 시스템에 관한 연구가 활발히 진행되고 있다. 또한 보다 안전한 운전 환경을 제공하기 위해 자율 주행 시스템도 각 국에서 연구가 활발히 진행 중이다. 이러한 시스템의 구축을 위해서는 주행방향과 지역에 관한 정확하고 많은 정보를 필요로 한다. 대부분의 연구에서 레이저, 혹은 레이더 센서를 중심으로 연구가 진행중이다. 그러나 이러한 센서들이 받아들일 수 있는 정보는 한계가 있다. 그러므로 다양한 정보를 추출할 수 있는 비전(vision) 시스템에 관한 연구가 주목을 받고 있다.

비전 시스템에서 주행 환경 중 받아들여야 할 정보로는 도로의 상태와 형상(직선, 곡선도로), 그리고 장애물 인식 등이다. 이러한 목적들을 달성하기 위해 주로 허프변환(Hough transform)이나 최소 자승법과 같은 방법이 많이 쓰인다[1-3]. 이러한 방법들은 복잡한 수학적인 모델링이 필요하고 많은 처리시간을 요한다. 또한 다양한 도로의 상황을 처리하기가 매우 어렵다. 이 문제를 해결하기 위해 패턴인식에 많은 장점을 지닌 신경회로망을 사용한 시스템을 구축하였다. 신경망을 이용한 영상인식은 전통적인 패턴인식의 통계적 혹은 수학적인 처리 과정을 필요로 하지 않는다. 반면 학습을 수행하는데 많은 시간을 요한다[4].

본 연구의 전체 알고리즘은 그림 1과 같이 구성하였고 알고리즘 내에서는 두 개의 서브 윈도우(sub window)를 설정한다. 하나는 영상 전처리 과정들을 위한 윈도우(640×290), 다른 하나는 신경망 입력을 위한 윈도우(450×150)이다. 윈도우의 크기는 카메라의 영향을 받는다. 입력 영상은 고속도로, 국도를 기준으로 하고 도로에 문자나 기호가 있는 영상은 고려하지 않는다.

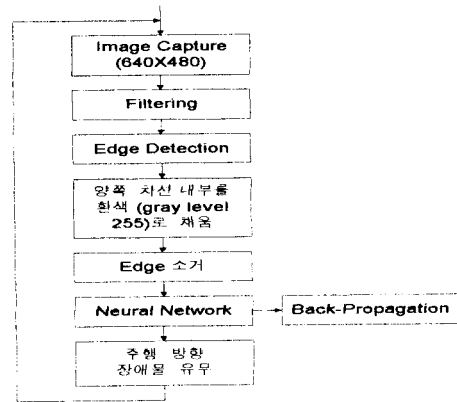


그림 1. 전체 알고리즘 흐름도

2 영상 전처리

전처리 과정은 다음과 같은 순서로 진행하고 서브 윈도우(640×290)내에서 이루어진다.

step 1. 잡음 제거(filtering)

잡음 제거과정(filtering)의 필터는 mean 필터를 사용한다[5].

step 2. 윤곽선 검출(Edge Detection)

윤곽선 검출 과정은 Sobel operator를 사용한다[5].

step 3. 양쪽 차선 내부를 흰색(gray level 255)으로 채움.

윤곽선을 검출한 후 차차가 속한 차선의 안쪽을 흰색(gray level 255)을 채운다. 이 과정은 다음과 같은 형식을 취하였다. 양쪽 차선의 중간의 화소(X,Y)에서부터 x축 양쪽 방향으로 gray level 255를 채워 나간다. 윤곽선 추출 후 차선의 윤곽선에 해당하는 화소는 통상 70이상의 값을 가지기 때문에 70이상

의 화소가 검출 될 때까지 수행한다. 음의 방향에서 멈춘 x값을 X_L 이라 두고 양의 방향에서 멈춘 x값을 X_R 이라 둔다. 그리고 X_C 를 식(2.a)와 같이 계산한다.

$$X_C = (X_L + X_R) / 2 \quad (2.a)$$

양방향으로 흰색을 채우는 과정이 끝나면 y축 방향으로 1을 감하여 즉 좌표($X_C, Y-1$)의 화소에서 다시 x축 양 방향으로 gray level 255를 채워나간다. 이와 같은 방법을 계속적으로 반복한다. X_L 과 X_R 사이에 화소가 특정 개수(본 논문에서는 15개) 이하가 되면, 이 과정을 멈춘다. 그러면 양쪽 차선이 만나는 지점이나, 혹은 장애물이 검출된 곳에서 멈춘다. 차선이 불연속인 부분은 영상면에서 도로의 특성을 이용한다. 직선도로는 물론이고 곡선도로 일지라도 영상면에서 특정 거리(본 연구에서는 y축을 기준으로 480에서 280화소들 까지)까지는 차선이 중간으로 모이는 형상이다. 따라서 특정 거리까지는 이전의 X_L , X_R 까지 윤곽선이 나타나지 않으면 흰색을 채우는 과정을 멈추고 X_L 과 X_R 을 1씩 감하고, y축에서 상위의 화소들 즉 ($X_C, Y-1$)화소에서 다시 흰색을 채우는 과정에 수행한다. 특정 거리 이상의 부분은 X_L 과 X_R 을 일정 개수(본 연구는 3개)만큼 벗어나는 화소까지 허용한다. 이것은 곡선도로를 위한 것이다.

step 4. 윤곽선(Edge) 소거

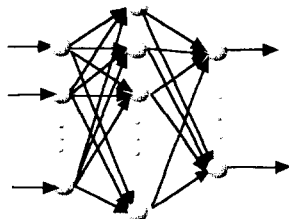
step 3의 과정을 수행후 차선을 비롯한 차선 주위의 윤곽선을 소거한다. 이 과정은 다음과 같이 수행하였다. 우선 3×3의 마스크를 설정한다. 그런 다음 마스크에 속한 화소들의 값을 더한다.

$$G = f(x-1,y-1) + f(x,y-1) + f(x+1,y-1) + f(x-1,y) + f(x,y) + f(x+1,y) + f(x-1,y+1) + f(x,y+1) + f(x+1,y+1) \quad (2.b)$$

G가 T(=9×255) 미만이면 해당 화소의 값을 0으로 하여 윤곽선(edge)을 소거한다.

3. 신경망 구조와 학습

신경망은 간단한 계산요소(computational element)를 밀도 있게 연결하므로써 좋은 성능을 얻고자 하는데 그 목적이 있다. 그 연결방식에 따라 여러 가지 모델이 제시될 수 있다. 본 연구에서는 그림 2와 같은 구조로 신경망을 설계하였다. 입력층의 노드의 개수는 27개, 은닉층은 30개, 출력층은 18개로 구성된다.



입력층(27) 은닉층(30) 출력층(18)

그림 2. 신경망의 구조

신경망 각 노드의 활성화 함수는 단극성 시그모이드 함수 사용한다[6-7].

$$f(net) = \frac{1}{1 + \exp(-net)} \quad (3.a)$$

net는 입력 가중 합이다.

신경망의 입력은 450×150 크기의 서브 윈도우 내의 값들을 취한다(그림 3). 도로의 형상이 이 서브 윈도우에서 직선 혹은 곡선 도로형상이 결정된다. 서브 윈도우는 27개의 소집합으로 구성된다. 하나의 소집합은 가로 50화소, 세로 50화소로 구성되어 있으므로 총 2500화소가 된다. 신경망 입력층의 i 번째 노드의 입력 값은 식(3.b)와 같다

$$(Input)_i = \sum x_j / T \quad (3.b)$$

x_j 는 소집합에 속한 화소들의 값이고, T는 637500(2500×255)이다. $\sum x_j$ 는 소집합들에 속한 화소들의 합이므로 이 화소들의 특성을 반영한다. 식 3.b는 시그모이드 함수가 지수함수로 구성되어 있으므로 0과 1사이의 값으로 정규화 하기 위한 것이다.

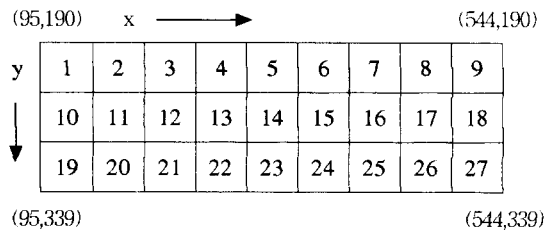


그림 3. 신경망의 입력을 위한 서브 윈도우(sub window)

신경망의 출력 형태는 그림 4와 같이 모두 18개로 구성된다. 1-3번 노드에서 최고 값을 출력되면 도로가 왼쪽으로 급하게 굽은 곡선도로가 되고, 4-6번 노드는 왼쪽으로 약간 굽은 곡선도로, 7-9번 노드는 직선도로, 10-12번 노드는 오른쪽으로 약간 굽은 곡선도로, 13-15번 노드는 오른쪽으로 급하게 굽은 곡선도로이다. 즉 도로의 형상이 주행 방향을 결정한다. 마지막으로 16-18번 노드에서 최고 값을 출력하면 전방에 장애물이 있는 경우이다. 출력 값을 3개씩 설정한 이유는 학습되지 않은 영상이 입력으로 사용될 경우 출력 값에 유연성을 주기 위해서이다.

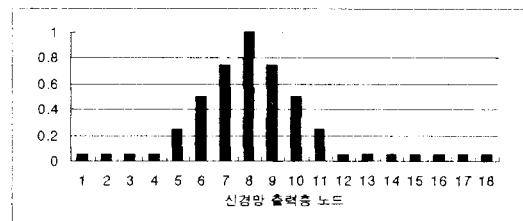


그림 4. 신경망의 출력 형태

신경망에서는 최적의 시스템을 획득하기 위해서는 학습과정이 필요하다. 본 연구에서는 Back-Propagation 학습방법을 택하였다. Back-Propagation 학습법은 현재 가장 널리 쓰이는 학습법으로 델타 학습법의 일종이다. 델타 학습법의 기본은 현재 주어진 연결 강도로 생성되는 오차를 구하여 이를 감소시키는 방

향으로 연결 강도의 값을 조정하는 것이다.

학습방법의 기본 원리는 다음과 같다. 입력층의 각 유니트에 입력패턴을 주면, 이 신호는 각 유니트에서 변환되어 중간층에 전달되고 최후에 출력층에서 신호를 출력하게 된다. 이 출력값과 목표 값을 비교하여 차이를 줄여 나가는 방향으로 연결강도를 조절하고, 상위층에서 역전파하여 하위층에서는 이를 근거로 다시 자기층의 연결강도를 조정해 나간다. 즉 j 노드에서 i 노드로의 연결 강도의 변화는 식 (3c)와 같다.

$$\begin{aligned} \Delta w_{ji} &= -\eta \frac{\partial E}{\partial w_{ji}} = \eta e_i o_j, \\ &= \eta (d_i - o_i) f(\text{net}_i) (1 - f(\text{net}_i)) o_j, \end{aligned} \quad (3c)$$

Δw_{ji} 는 가중치 변화량, η 는 학습률, e_i 는 오차, o_j 는 j 노드의 출력, d_i 는 목표 출력 값이다[6-7]. 본 실험에서 목표 출력값은 사용자가 지정해주는 방식을 취하였다. 신경망의 학습은 목표 출력값과 신경망 출력값의 오차를 역전파 함으로써 이루어진다.

4 실험 결과

도로의 영상은 비디오 카메라로 촬영하고 비전 보드는 Oculus Tci-Vga Ultra II를 사용한다. 컴퓨터는 펜티엄 133MHz를 사용한다. 비전 보드는 프레임 그래버(frame grabber) 기능만 수행하고 나머지 모든 영상처리는 컴퓨터(PC)를 통해 이루어진다. 신경망 학습 데이터 영상은 6가지의 신경망 출력력을 기준으로 6 프레임의 영상을 사용하였다. 실험 결과는 다음과 같다.

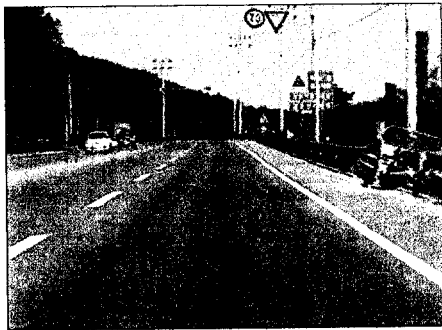


그림 5. 직선도로

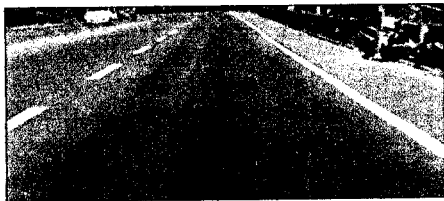


그림 6. 잡음을 제거한 영상

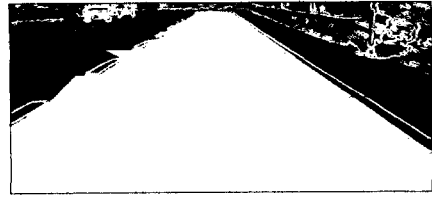


그림 7. 흰색(gray level 255)을 채운 영상

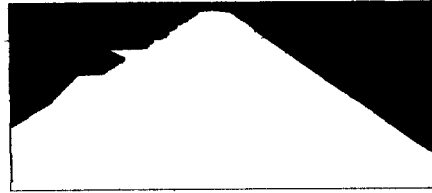


그림 8. 윤곽선을 제거한 영상

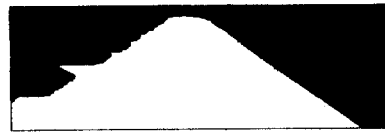


그림 9. 신경망 입력영상

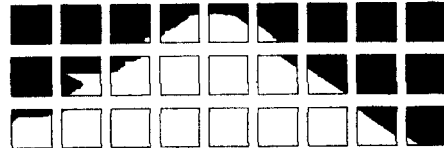


그림 10. 신경망 입력을 위한 27개의 소집합

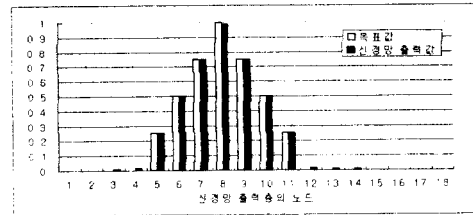


그림 11. 그림 6의 출력 값

그림 11은 그림 5의 출력 값이다. 직선도로이므로 8번 노드에서 최고 값이 출력된다. 그림 11은 학습 데이터로 사용한 영상이다.



그림 12. 신경망 입력영상(곡선도로 1)



그림 13. 신경망 입력영상(곡선도로 2)

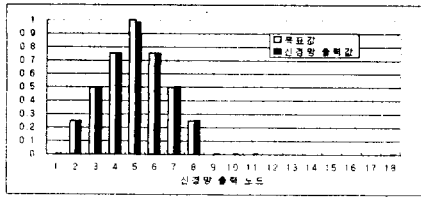


그림 14. 그림12의 출력값

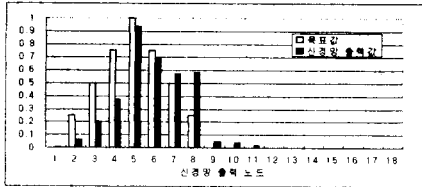


그림 15. 그림 13의 출력값

그림 12는 학습 데이터로 사용한 영상이고 그림 13은 학습 데이터로 사용하지 않은 영상이다. 그림 13은 그림 12와 비슷한 유형이나 도로의 형상이 약간 오른쪽으로 치우쳤다. 그러므로 출력값도 오른쪽으로 치우쳐 나타난다.



그림 16. 전방에 장애물이 있는 영상

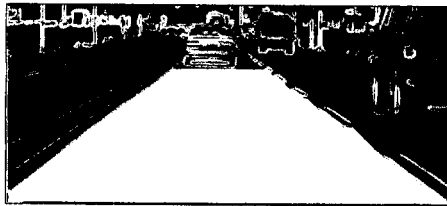


그림 17. 윤곽선 검출후 흰색을 채운 영상



그림 18. 신경망 입력 영상

장애물이 있는 도로는 서브 윈도우내에 0의 값을 가진 화소들이 많이 나타난다. 장애물이 가까이 있는 경우는 윈도우내의 모든 화소들의 값이 0으로 나타날 수도 있다. 그림 19에서 출력값은 17번 노드에서 최고 값을 출력한다. 그림 16은 학습 데이터 영상이다.

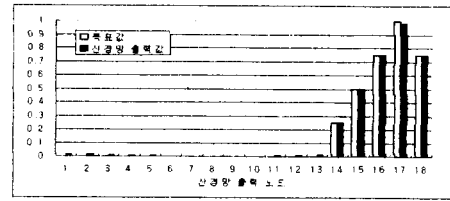


그림 19. 그림 16의 신경망의 출력값

5. 결론

본 실험에서는 도로의 형상을 결정하는 부분을 27개의 소집합을 기초로 하여 차량의 주행방향과 장애물을 인식하는 시스템을 구축하였다. 실험 결과 신경망 출력 값이 목표 값에 잘 추종한다. 이 실험은 기존의 연구들보다 입력으로 사용하는 도로 영상의 범위를 넓혔다.

신경망의 입력 값을 처리속도를 상승시키기 위해 소집합을 25개의 샘플(sample) 화소로 구성할 수 있다. 즉 10화소씩 건너서 샘플 화소를 취하는 방법이다. 결과는 2500개를 취한 경우와 거의 유사하게 목표 값에 잘 추종한다. 또한 출력 값을 차량의 주행방향 이외 다른 값들을 사용할 수 있다. 예를 들어 도로의 형상으로 할 수도 있고, 핸들의 방향과 각도로 잡을 수도 있다. 핸들의 방향과 각도를 출력으로 사용할 경우는 차량의 무게와 속도에 관한 파라메타들이 고려되어야 한다. 그리고 그림 14, 15에서 알 수 있듯이 학습 데이터를 다양하게, 주행 방향 출력을 5가지 이상 더 많은 경우로 구성하면 보다 나은 결과를 얻을 수 있다.

본 실험의 목표 출력 값은 사용자가 지정해주는 방식을 취하였다. 목표 출력 값을 시스템 내에서 자동으로 설정해주는 것이 과제로 남아있다.

참고문헌

- [1] 정준의, 최성구, 노도환, "직선 Edge 추출에 의한 주행방향 및 장애물 검출에 관한 연구," KACC, October 1996
- [2] 문영수, 정상철, 이준용, 강동중, 권인소, "교통 정보 검지기 및 지능형 자동차 개발을 위한 영상처리 알고리즘," ICASE, 1996년 11월
- [3] Toshiaki Kakinami, Jun Sato, Mitsuyoshi Saiki, and Kunihiko Soshi, "Autonomous Vehicle Control System Using an Image Processing Sensor", SAE, 1995
- [4] Shumeet Baluja, "Evolution of an Artificial Neural Network Based Autonomous Land Vehicle Controller," IEEE Transactions on Systems, Man, and Cybernetics - PART B: Cybernetics, Vol. 26, No.3, June 1996
- [5] 이문호, 염재훈, "C언어를 이용한 영상신호처리," 대영사
- [6] 오창석, "뉴로 컴퓨터," 지성출판사
- [7] 이성환, "패턴인식의 원리 II권," 홍릉과학 출판사