2014년 8월
석사학위논문

로봇환경에서 3차원 콘볼루션 신경회로망을 이용한 비디오기반 얼굴인식

조 선 대 학 교 대 학 원

제 어 계 측 공 학 과

변 영 현

로봇환경에서 3차원 콘볼루션 신경회로망을 이용한 비디오기반 얼굴인식

A Video-based Face Recognition Using 3D Convolutional Neural Networks Under Robot Environments

2014년 8월 25일

조 선 대 학 교 대 학 원

제 어 계 측 공 학 과

변 영 현

로봇환경에서 3차원 콘볼루션 신경회로망을 이용한 비디오기반 얼굴인식

지도교수 곽 근 창

이 논문을 석사학위신청 논문으로 제출함.

2014년 4월

조 선 대 학 교 대 학 원

제어계측공학과

변 영 현

변영현의 공학석사학위논문을 인준함

위원장 조 선 대 학 교 교수 장 순 석 인

위 원 조 선 대 학 교 교수 반 성 범 인

위 원 조 선 대 학 교 교수 곽 근 창 인

2014년 5월

조선대학교 대학원

목 차

표 목차

그림 목차

ABSTRACT

A Video-based Face Recognition Using 3D Convolutional Neural Networks Under Robot Environments

Byeon, Yeong Hyeon
Advisor : Prof. Kwak, Keun Chang, Ph. D.
Dept. of Control and Instrumentation Eng.,
Graduate School of Chosun University

A HRI(Human-Robot Interaction) is a system for robot to interact with human by channels of communication. Natural interaction between robot and human is needed to provide human services. One of the HRI's components is a vision of which role is eye in human. So, a face detection and a face recognition are essential. There are convention techniques for face recognition such as PCA(Principal Component Analysis), LDA(Linear Discriminant Analysis) and 2D-CNN(Convolutional Neural Network). These methods are normally process images one by one. In this paper, we experiment video-based face recognition using 3D CNN on the ETRI face video. Training data is 50pictures captured from 1m away, and checking data are 50, 200 pictures captured from 1m, 2m away and The rates of true recognition are 100% and 88% as maximum by varying the number of feature maps.

제1장 서론

인간-로봇 상호작용(HRI: Human-Robot Interaction)기술은 인간과 로봇이 다양한 의사소통 채널을 통해 인지적/정서적 상호작용을 할 수 있도록 로봇 시스템 및 상호작용 환경을 디자인, 구현 및 평가하는 기술로써 지능형 로봇의 핵심기술이며, 로봇이 인간과의 자연스러운 의사소통 및 상호협력을 위해 사용자의 의도를 종합적으로 판단하고, 그에 맞는 반응 및 행동을 하기 위한 기술이다[1]-[5].

이러한 HRI 기술은 기존 인간-컴퓨터 상호작용(HCI: Human-Computer Interaction)과 로봇이 가지는 자율성, 상호작용의 양방향성, 상호작용 또는 제어 수준의 다양성 등에서 근본적인 차이점을 가지고 있다 [6][7].

지능형 서비스 로봇이 일상 환경에서 인간과 함께 생활하면서 필요한 서비스를 제공하기 위해서는 인간과 동일한 방식으로 인간과 교류하는 능력이 필수적이다. 이렇듯이 효과적인 인간과 로봇 간 효과적인 상호작용을 위해서는 인간 및 로봇상호간의 편리성 (Convenience), 협력성 (Cooperativeness), 친밀성(Closeness)을 구현하는 모듈 구조를 가진 C3 패러다임 시스템이 요구된다. 또한, 인간과 로봇의 효과적인 상호작용을 위해서는 다양한 분야의 기술이 융합되어야 한다. 예를 들면, 시각, 청각, 촉각, 매개 인터페이스 등 다양한 의사전달 매체를 지원할 수 있는 멀티모달 상호작용기법과 다양한 상호작용 채널을 통해 들어오는 정보를 통합하는 멀티모달 통합기술이 필요하다. 게다가 사용자 및 상황적 요구에 적절한 과제를 수행하기 위해서는 상황인식, 추론, 의사결정, 계획 등의 일련의 인지적 과정과 이러한 인지적 과정이 필요하다. 또한, 음성, 표정 등의 다양한 인간의 정서적 반응을 인식하여 로봇의 성격 모형과 상황에 맞는 적절한 정서적 행동을 생성하는 기술이 요구된다[6]. HRI의 요소기술은 시각, 청각, 자율주행, 부품 및 센서, 매개 인터페이스, 지능, 매니퓰레이터, 디자인으로 구성되며[8], 그중에 시각 및 청각은 사람의 눈과 귀에 해당하며 로봇의 카메라와 마이크로폰에 의해 얻어진 영상 및 음성정보로부터 인간과 로봇 간 자연스럽게 상호작용을 할 수 있다. 기존의 시청각 기반 HRI기술은 관련 컴포넌트 기술들을 로봇자체에서 실행되기 때문에 컴퓨팅 파워에 많은 부하가 생기고, 독립로봇에만 한정해서 HRI관련 서비스를 수행할 수 있다. 그러나, u-로봇환경에서의 시청각기반 HRI기술은 서버-클라이언트 구조를 갖기 때문에 로봇카메라와 마이크로폰으로부터 얻어진 영상 및 음성 정보

들을 서버로 전송하고 관련 컴포넌트 기술들을 서버에서 처리하게 된다. 이렇게 함으로써 Planet과 HRI서버 연동을 통한 복수가구 지원이 가능해지고, 하나의 서버에서 많은 로봇들이 동시에 HRI관련 서비스를 수행할 수 있는 장점을 가지고 있다.

얼굴검출 및 인식기술은 다양한 지능형 로봇들에 탑재될 필수 핵심 컴포넌트 기술로써 로봇의 카메라로부터 입력된 영상으로부터 존재하는 사람의 얼굴을 검출하여 그 사람의 신원을 부여하고 인증하는 기술이 필요하다. 특히 지정된 장소에서 지정된 작업을 반복하는 산업용 로봇과는 달리 가정환경에서 사용자와 함께 생활하면서 다양한 서비스를 제공하는 것을 목적으로 하는 지능형 서비스 로봇에서는 얼굴검출 및 인식기능은 필수적이다. 얼굴검출을 통해 근거리에서 로봇과 음성대화하는 동안 로봇은 계속적으로 사용자의 얼굴을 추적하면서 시선을 맞추고, 인간과 자연스럽게 상호작용할 수 있다. 또한, 얼굴인식을 통해 사용자의 기호, 습관, 의도 등에 최적화된 맞춤형 서비스와 같은 고품질 서비스가 제공될 수 있다. 얼굴검증의 경우에는 등록된 가족 이외의 사용자는 서비스를 거부함으로써 가족의 정보보안뿐만 아니라 도둑의 가정 내 침입 등의 정보를 가족에게 메시지를 보낼 수 있다[9][10][11][12].

기존 얼굴인식분야에서 일반적으로 사용되고 있는 특징추출 방법으로 주성분분석기법(PCA: Principal Component Analysis) [13] 와 LDA(Linear Discriminant Analysis) [14] 그리고 2D CNN(Convolutional Neural Network) [15] 등이 있다. 이러한 경우는 일반적으로 얼굴인식을 한 장씩 처리하게 되는데, 로봇환경에서는 사용자가 협조적이지 않고 비협조적이므로 한 장씩 처리하는 것보다 여러 장을 한 번에 처리하여 인식하는 것이 인식성능면에서 효율적이다. 그래서 여러 장을 처리하기 위해 3차원 신경회로망이 연구되고 있으며 최근에 3차원 콘볼루션 신경회로망 (CNN: Convolutional Neural Networks)을 이용한 행동인식[16]과 사람추적[17] 등에 성공적으로 적용되고 있다[18].

본 논문에서는 사용자가 비협조적인 서비스 로봇환경에 적합하도록 기존 2차원 콘볼루션 신경회로망(2D-CNN)을 3차원 콘볼루션 신경회로망(3D-CNN)으로 개선하여 비디오기반 얼굴인식을 수행한다. 3D-CNN구조에서 어느 정도 이동과 크기 그리고 방향에 불변성을 갖고 있고[15], 또한 사용자가 협조적이지 않고 비협조적인 로봇환경에서 한 장씩 처리하는 것보다 여러 장을 한 번에 처리하여 시간에 따른 정보의 손실, 떨림이 어느 정도 보완되어 더 효율적이다. 한국전자통신연구원(ETRI)의

로봇 테스트베드(test bed)환경에서 구축된 얼굴영상 데이터베이스에 실험한 결과 제안된 3D-CNN은 기존의 얼굴인식방법들인 PCA와 LDA 그리고 2D-CNN보다 좋은 인식성능을 얻었다. 2장에서는 기존 얼굴인식방법들인 PCA와 LDA 그리고 2D-CNN에 대해서 설명하고, 3장에서는 제안된 3D-CNN에 대해서 기술하고, 4장에서는 실험에 사용된 데이터베이스와 실험결과를 보여주고 있다. 마지막으로 5장에서 결론을 맺는다.

제2장 기존 얼굴인식방법

제1절 주성분분석기법

주성분 분석 방법(PCA: Principal Component Analysis)은 분산까지의 통계적 성질을 이용한 2차 통계적 방법으로, 고차원의 입력 데이터의 차원을 효율적으로 축소하는 데에 주로 사용된다. 주성분 분석 방법을 요약 하자면, 전체 영상의 데이터를 가지고 그것들의 분산이 큰 몇 개의 고유 방향에 대한 축으로 선형 투영시켜 차원을 줄이는 방법을 말한다[14][19].

그림 2.1은 두 클래스로 이루어진 2차원 벡터들을 주성분 분석 방법에 의해 생성된 벡터 축으로 투영된 것을 나타내고 있다.

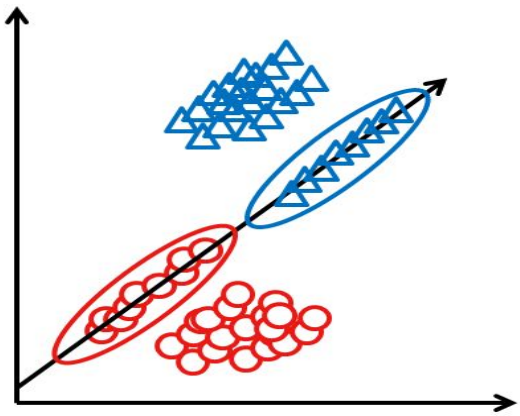


그림 2.1 주성분 분석 방법에 의한 투영

주성분 분석 방법으로 1차원으로 축소되었지만 클래스간의 분리가 가능하다는 것을 나타내고 있다. 즉, 입력 벡터의 차원이 감소하더라도 입력 데이터의 분포에 대한 정보 유지, 계산상의 부하 감소, 노이즈 제거, 데이터 압축과 같은 효과를 나타내는 장점을 갖는다. 주성분 분석 방법의 학습 과정을 간단히 살펴보면, 아래 표 2-1과 같다.

표 2-1. 주성분 분석 방법 학습

1. P개의 학습 영상 벡터 정의

$$X = [x^1|x^2|...x^P] \tag{2-1}$$

2. 각 영상 벡터와 평균 영상 벡터의 차

$$\overline{x^i} = x^i - mean \quad , mean = \frac{1}{P}\sum_{i=1}^{p} x^i \tag{2-2}$$

3. P개의 $\overline{x^i}$벡터를 이용한 $N \times N$ 공분산 행렬

$$\Omega = \overline{X}\overline{X}^T \ , \overline{X} = [\overline{x^1}|\overline{x^2}|...\overline{x^i}] \tag{2-3}$$

4. 공분산 행렬에 대해서 고유치와 고유벡터 정의

$$\Omega v_i = \lambda v_i \tag{2-4}$$

5. 학습 영상에 대한 특징 벡터 정의

$$\widetilde{x^i} = V^T\overline{X^i} \ , \ \overline{V} = [\overline{v^1}|\overline{v^2}|...\overline{v^P}] \tag{2-5}$$

여기서 공분산 행렬에 의해 얻어지는 고유치는 분산을 최대로 하는 방향을 나타내고 이 고유치에 대응하는 고유 벡터는 특정 방향의 변동성을 나타낸다. 이 고유벡터가 고유 얼굴(Eigenfaces)이다. 그림 2.2는 고유 얼굴과 특징벡터의 선형결합에 의한 얼굴 영상 표현을 나타낸다. 주성분 분석 방법의 검증 과정은 다음 표 2-2와 같다.

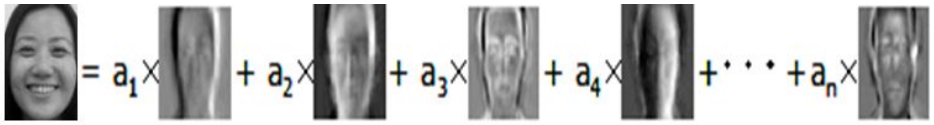


그림 2.2 고유 얼굴과 특징벡터의 선형결합에 의한 얼굴 영상 표현

표 2-2. 주성분 분석 방법 검증

1. P개의 검증 영상 벡터 정의

$$Y = [y^1|y^2|...y^P] \tag{2-6}$$

2. 각 영상 벡터와 평균 영상 벡터의 차

$$\overline{y^i} = y^i - mean \quad , mean = \frac{1}{P}\sum_{i=1}^{p} y^i \tag{2-7}$$

3. 고유벡터 V를 이용한 검증 영상에 대한 특징 벡터 정의

$$\tilde{y^i} = V^T \overline{y^i} \tag{2-8}$$

위에서 구해진 검증 영상의 특징 벡터와 학습 영상의 특징 벡터들 간의 유사도를 측정하여 가장 유사한 특징 벡터 영상은 인식 결과 영상으로 사용된다. 유사도 측정 방법은 4장 실험 부분에서 자세히 설명한다[20].

제 2절 선형판별분석기법

선형 판별 분석 방법(LDA: Linear Discriminant Analysis) 은 클래스내의 분산을 나타내는 행렬(within-class scatter matrix: S_W)과 클래스 간 분산을 나타내는 행렬(between-class scatter matrix: S_B)의 비율이 최대가 되는 선형 변환법으로 데이터에 대한 특징벡터의 차원을 축소하는 방법이다[21-23]. 그림 2.3은 주성분 분석 방법과 선형 판별분석 방법의 비교를 나타낸 것이다.

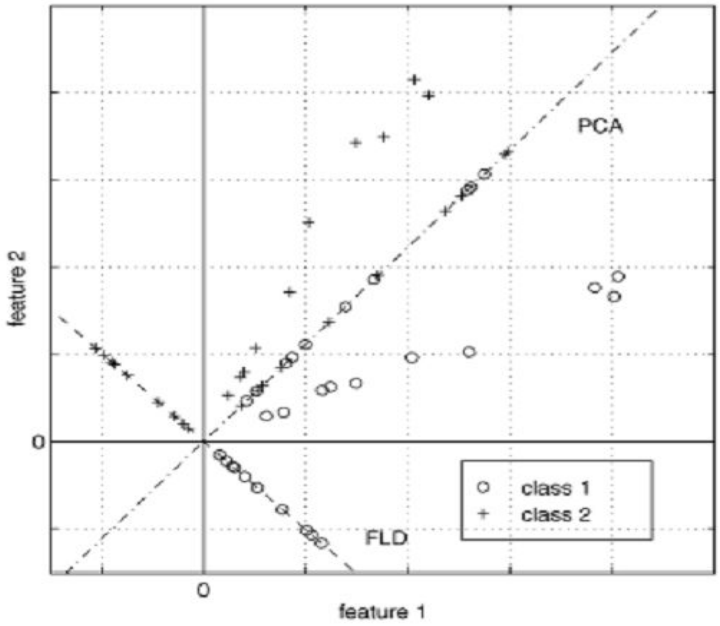


그림 2.3 주성분 분석 방법과 선형 판별분석 방법의 비교

이 변환 방법은 클래스간의 편차는 최대로 해주면서, 집단 내 편차를 최소로 하여, 데이터를 쉽게 나눌 수 있고, 집단 또한 쉽게 분리 할 수 있다. 학습과정을 간단히 살펴보면 아래와 같다.

표 2-3. 선형 판별 분석 방법의 학습 과정

1. P개의 학습 영상 벡터 정의

$$X = [x^1|x^2|x^3|...x^P] \tag{2-9}$$

2. i번째 클래스 내의 분산을 나타내는 행렬 정의

$$S_i = \sum_{x \in X_i} (x - mean_i)(x - mean_i)^T \quad , mean = \frac{1}{P_i} \sum_{x \in X_i} x \tag{2-10}$$

3. 전체 클래스 내의 분산을 나타내는 행렬 S_W정의

$$S_W = \sum_{i=1}^{C} S_i \tag{2-11}$$

4. 클래스 간 분산을 나타내는 행렬 S_b정의

$$S_B = \sum_{i=1}^{C} n_i (Mean_i - Mean)(Mean_i - Mean)^T \ , Mean = \frac{1}{P}\sum_{i=1}^{P} x^i \qquad (2\text{-}12)$$

5. S_W와S_b의 비율이 최대가 되는 행렬 정의

$$W_{opt} = \arg max \frac{|W^T S_B W|}{|W^T S_W W|} \qquad (2\text{-}13)$$

$$= [w_1, w_2 ... w_m]$$

$$S_B w_i = \lambda_i S_W w_{i,\ i=1,2,...m} \qquad (2\text{-}14)$$

- 여기에서, X_i는 클래스, C는 클래스의 수, n_i는 각 클래스에 속하는 샘플의 수, P는 모든 표본 집합의 개수, m은 얻고자 하는 고유 벡터의 수를 나타낸다.

위의 과정을 거치면 C-1개의 영이 아닌 고유치가 존재하며 따라서, m의 상한 값은 C-1개가 존재한다. 이 고유벡터가 Fisherfaces이다. 그림 2.4는 Fisherfaces와 특징벡터의 선형결합에 의한 얼굴영상 표현 나타낸다.

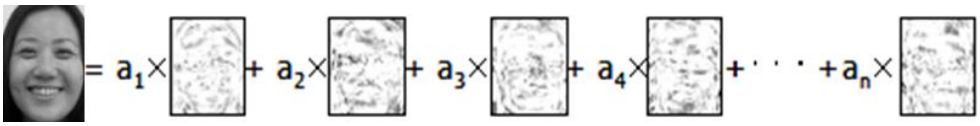


그림 2.4 Fisherfaces와 특징벡터의 선형결합에 의한 얼굴 영상 표현

선형 판별 분석방법의 특징 추출 과정은 다음 표 2-4와 같다.

표 2-4. 선형 판별 분석 방법의 특징 추출

1. P개의 검증 영상 벡터 정의

$$Y = [y^1 | y^2 | ... y^P] \qquad (2\text{-}15)$$

2. 각 영상 벡터와 평균 영상 벡터의 차

$$\overline{y^i} = y^i - mean \quad , mean = \frac{1}{P}\sum_{i=1}^{p} y^i \tag{2-16}$$

3. W_{opt}를 이용한 검증영상에 대한 특징 벡터 정의

$$\tilde{y^i} = W_{opt}\overline{y^i} \tag{2-17}$$

위에서 구해진 검증 영상의 특징 벡터와 학습 영상의 특징 벡터들 간의 유사도를 측정하여 가장 유사한 특징 벡터 영상을 인식 결과 영상으로 사용된다. 선형 판별 분석방법은 Eigenfaces방법보다 조명이나 표정변화가 있는 얼굴영상에 대해 우수한 인식 성능을 나타낸다. 그러나 선형 판별 분석방법은 클래스내의 분산이 비정칙(Singular)이 되는 문제가 있다. 클래스내의 분산이 정칙이 되도록 하기 위해서는 주성분 분석 방법을 이용하여 영상집합을 저차원 공간으로 투영함으로써 해결할 수 있다[20].

제 3절 2차원 콘볼루션 신경회로망(2D-CNN)

영상에 콘볼루션(Convolution)을 수행하는 것은 영상의 특징을 추출하는 과정이다. 영상에서 2D-CNN은 홀수 배 크기의 커널(Kernel)의 중심이 영상의 픽셀에 놓인 상태에서 영상과 커널이 겹쳐진 부분들만 곱해서 더하여 출력 영상의 픽셀에 넣는 것이다. 이러한 과정은 영상의 전체 픽셀에 대해서 수행이 된다. 예를 들어 그림 2.5과 같이 7x6 사이즈의 영상과 3x3 커널이 있다.

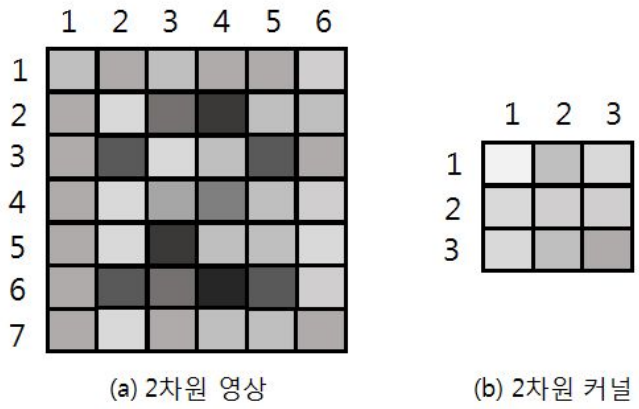


그림 2.5 콘볼루션에 대한 영상과 커널 예시(1)

그림 2.5의 (a)영상과 (b)커널에 대해 콘볼루션을 수행하게 하려면, 그림 2.6와 같이 콘볼루션을 계산할 (a)영상 위의 한 픽셀에 (b)커널의 중심을 일치시킨다. 그러면 (a)영상의 1행1열부터 3행3열까지 겹치는 것을 볼 수 있다. 겹치는 영역은 파란 선으로 나타내었고, 겹치는 영역의 중심 픽셀은 빨간 선으로 나타내었다.

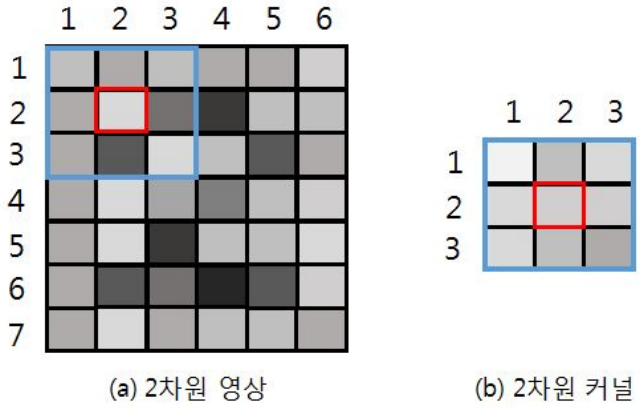


그림 2.6 콘볼루션에 대한 영상과 커널 예시(2)

그림 2.6에서 영상과 커널의 콘볼루션을 계산하는 방법은 식(2-18)과 같다. 커널과 영상의 겹치는 픽셀 각각 곱하고 전체를 더한다. o는 출력을 의미한다.

$$o_{(1,1)} = a_{(1,1)}b_{(1,1)} + a_{(1,2)}b_{(1,2)} + a_{(1,3)}b_{(1,3)} + a_{(2,1)}b_{(2,1)} + a_{(2,2)}b_{(2,2)} + a_{(2,3)}b_{(2,3)} + a_{(3,1)}b_{(3,1)} + a_{(3,2)}b_{(3,2)} + a_{(3,3)}b_{(3,3)} \quad (2\text{-}18)$$

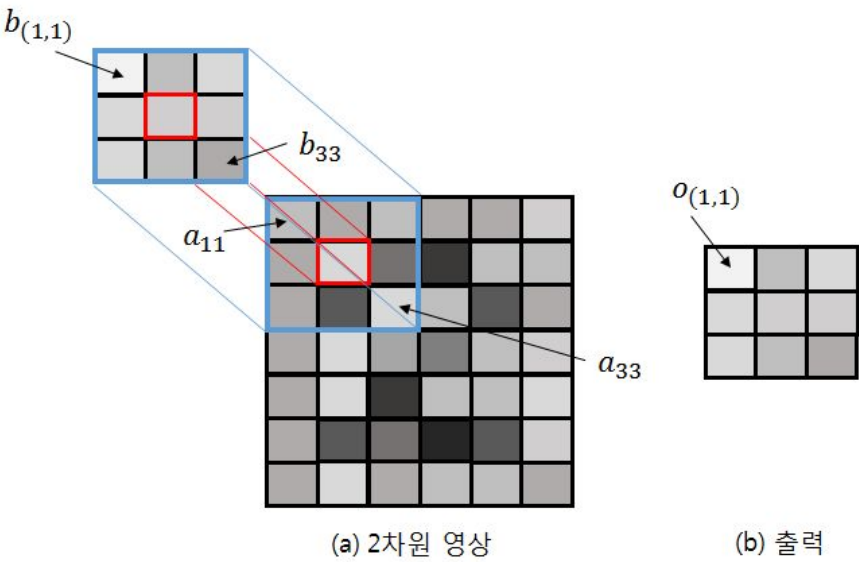


그림 2.7 콘볼루션에 대한 영상과 커널의 예시(3)

그림 2.7의 (a)영상에서 $a_{(2,2)}$에 있는 커널의 중심은 $a_{(2,3)}$으로 이동하여 콘볼루션을 수행하면 $o_{(1,2)}$가 구해진다. 커널의 중심이 영상의 끝 픽셀까지 이동하여 모두 구하면 콘볼루션이 완료된다.

앞서, 사이즈가 7x6인 영상과 3x3인 커널을 콘볼루션 연산을 하였다. 그러면, 콘볼루션 연산 결과로 출력된 영상의 크기는 콘볼루션 옵션에 따라 보통 3가지로 변경할 수 있다. 그림 2.8의 (a)와 같이 커널과 영상이 겹치는 픽셀이 1개일 때도 콘볼루션을 수행을 하게 되면, 원래 7x6인 영상은 3x3 커널의 콘볼루션 연산 결과로 그림 2.8의 (b)와 같이 9x8 사이즈가 나온다. 이때 사이즈를 계산하는 방법은 식(2-19)와 같다.

결과 영상의 크기 = 원래 영상의 크기 + (커널의 크기 - 1) (2-19)

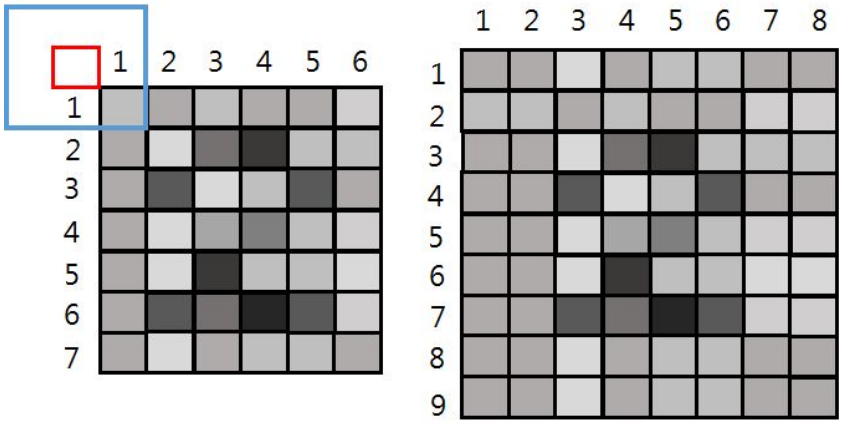


그림 2.8 콘볼루션 연산 후 영상의 크기가 증가

다음은 그림 2.9의 (a)와 같이 커널의 중심을 영상의 데이터가 존재하는 곳만을 일치시켜 콘볼루션 연산한 경우이고, 원래 7x6인 영상이 그림 2.9의 (b)와 같이 동일한 7x6 사이즈가 나온다. 이때 사이즈를 계산하는 방법은 식(2-20)와 같다.

결과 영상의 크기 = 원래 영상의 크기 (2-20)

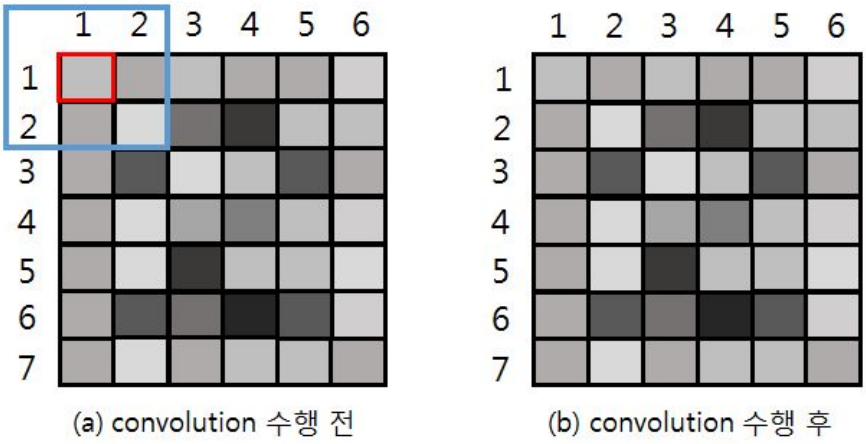


그림 2.9 콘볼루션 연산 후 영상의 크기가 동일

다음은 그림 2.10의 (a)와 같이 커널이 영상의 데이터가 존재하는 부분 전체와 겹칠 경우만 콘볼루션 연산을 한 경우이다. 7x6인 영상과 3x3 커널에 대해 콘볼루션 연산을 한 결과로 그림 2.10의 (b)와 같은 5x4 사이즈가 나온다. 이때 사이즈를 계산하는 방법은 식(2-21)와 같다.

결과 영상의 크기 = 원래 영상의 크기 - (커널의 크기 - 1) (2-21)

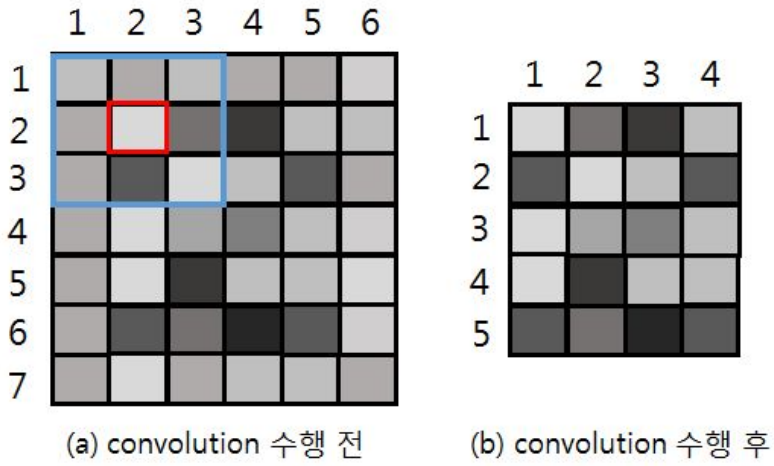


그림 2.10 콘볼루션 연산 후 영상의 크기가 감소

CNN에서 콘볼루션 층의 연산은 3번째 경우로서, 커널이 영상과 완전히 포개어 수행한다.

영상에서 서브샘플링(Subsampling)을 수행하는 것은 영상의 이동, 회전, 크기 변화에 불변성을 갖도록 하고, 영상의 사이즈를 자연수 배로 축소시킨다. 그 방법은 영상을 축소 배수와 동일한 사이즈의 커널로 평균 콘볼루션을 연산하고, 그 결과 영상에서 축소 배수만큼 건너 띄면서 픽셀을 취득하여 감소된 영상을 얻는다. 예로 그림 2.11의 (a)와 같이 사이즈가 8x6인 영상이 있고, 서브샘플링을 통해 2배 축소하여 4x3 영상을 얻으려고 한다. 그러기 위해서 그림 2.11의 (b)와 같이 축소 배수의 크기인 2x2로 커널을 정의한다.

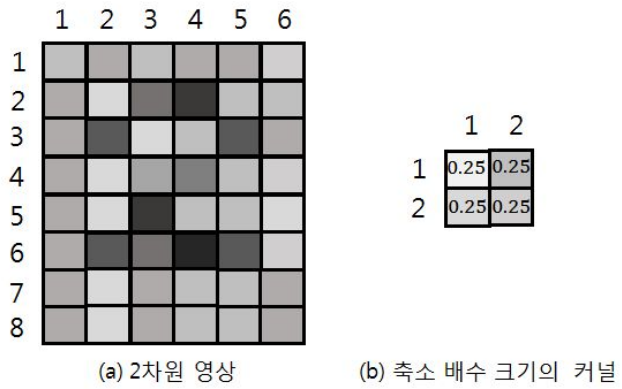


그림 2.11 서브샘플링에 대한 영상과 커널 예시

그 다음, 평균을 구하기 위해 커널의 값을 초기화하는데, 그 값을 구하는 방법은 식 (2-22)와 같다.

커널의 픽셀 값 = 1 / 축소 배수의 제곱 (2-22)

식 (2-22)를 통해서 커널의 모든 픽셀에 대입될 값은 그림 2.11의 (b)와 같이 $\frac{1}{2^2}=\frac{1}{4}=0.25$이다.

그림 2.12의 (a)에서처럼 영상과 평균 커널간에 커널과 영상이 완전히 겹친 콘볼루션 연산을 수행한 결과로 그림 2.12의 (b)와 같이 원본 영상보다 사이즈가 감소한 7x5 사이즈를 얻는다. 그 다음, 축소 배수씩 건너서 샘플링을 하게 되면, 그림 2.12의 (b)처럼 x 표시의 픽셀정보가 지워지고, 최종적으로 그림 2.12의 (c)와 같은 4x3 사이즈의 결과가 얻어진다.

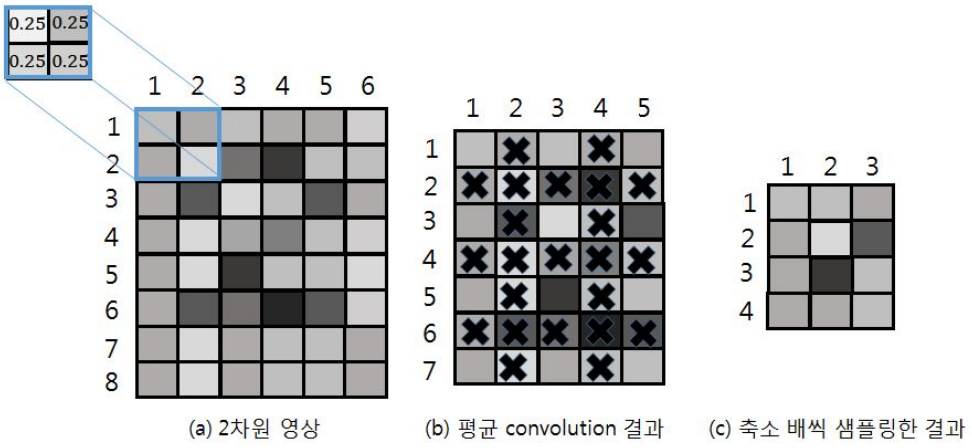


그림 2.12 서브샘플링(subsampling) 과정

CNN의 구조는 다양하게 설계가 가능하다. 예를 들어, 콘볼루션 층 또는 서브샘플링 층의 수를 늘리거나 줄일 수 있고, 콘볼루션 층의 특징 맵의 개수도 변경이 가능하다. 여기에서 설명하는 구조는 GitHub에서 다운로드가 가능한 deep learning 툴박스에 포함된 CNN 구조를 기반으로 한다.

그림 2.13는 CNN의 구조를 나타내기 전에 콘볼루션 층만 간단한 모델로 보인 것이다. 하나의 음영 사각형은 한 맵을 의미하고, 'k-1'은 'k'보다 이전 층을 의미한다. 'k'층에서 입력맵의 개수는 'k-1'층의 맵수인 n개이고, 출력맵의 개수는 'k'층의 맵수인 m개 이다. 'k'층에서 한 맵이 출력되기 위해서는 입력층의 모든 맵마다 각각 정의된 커널과 콘볼루션이 수행되어 합쳐진다.

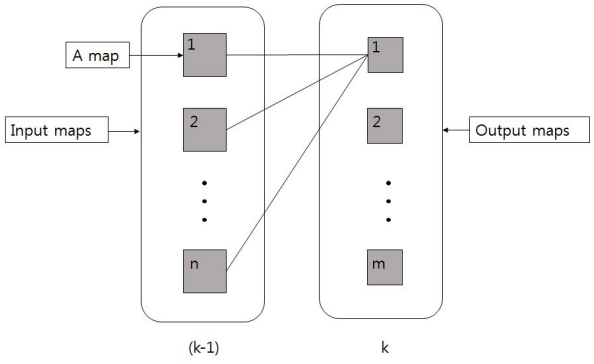


그림 2.13 콘볼루션 층의 구조

그림 2.14은 CNN의 구조를 나타내기 전에 서브샘플링 층만 간단한 모델로 보인 것이다. 하나의 음영 사각형은 한 맵을 의미하고, 'k-1'은 'k'보다 이전 층을 의미한다. 서브샘플링은 입력맵의 개수가 출력맵의 개수와 같다. 이전 층과 동일한 순서로 영상의 사이즈가 줄어든다.

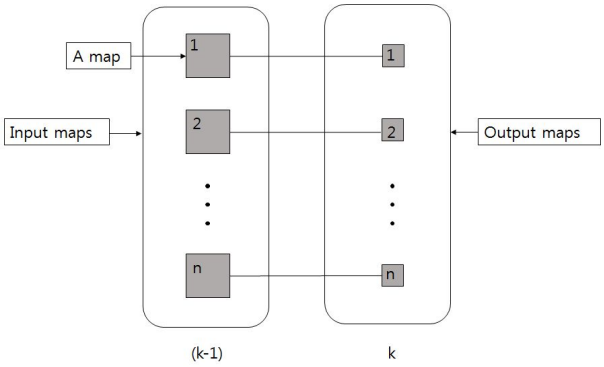


그림 2.14 서브샘플링 층의 구조

그림 2.15와 그림 2.16은 필기체 숫자인식 예제의 CNN 구조를 보여주고 있다. 그곳에서 CNN은 5층으로 설계가 되어있다. 1층은 입력단계이고, 2층은 1차 콘볼루션단계, 3층은 1차 서브샘플링단계, 4층은 2차 콘볼루션단계, 5층은 2차 서브샘플링단계의 구조를 가지고 있다. 초기 커널 값들은 특정영역의 임의 값으로 설정되며, 1층에서 한 장의 영상이 입력으로 들어가고, 그 영상으로부터 2층에서 6개 맵으로 콘볼루션 되어 특징이 추출되고, 3층에서 서브샘플링을 거쳐 영상의 사이즈를 줄이고, 다시 6개의 입력맵으부터 4층에서 12개 맵으로 콘볼루션 되어 특징이 추출되고, 5층에서 서브샘플링을 거쳐 영상의 사이즈를 줄이고, 마지막 층에서 영상을 단일 벡터로 구성하고, 12개의 맵을 모두 이어 붙이면 한 입력에 대한 특징 벡터가 구해진다.

영상의 사이즈 변화는 1층의 입력영상이 28x28 사이즈이고, 2층의 콘볼루션 층에서 커널사이즈가 5x5인 콘볼루션을 수행하게 되는데 여기에서 콘볼루션은 커널과 영상이 완전히 포개어지는 것을 사용하기 때문에 결과 영상은 '원본영상의 크기 - (커널의 사이즈 - 1)' 이므로 24x24 사이즈의 영상이 출력된다. 그 다음, 3층에서 이동과 회전, 크기 변화에 불변성을 갖도록 하면서 영상의 사이즈를 줄이는 서브샘플링을 수행한다. 이때 커널의 사이즈는 2x2이므로 영상의 크기는 절반

으로 줄어들어 12x12가 된다. 그 다음, 4층에서는 커널사이즈가 5x5인 콘볼루션을 수행하여 8x8 사이즈의 영상이 출력된다. 5층에서 다시 2x2 커널 사이즈의 서브샘플링을 거치면서 영상 크기가 절반 줄어 출력맵의 사이즈가 4x4가 된다. 이 영상을 단일 벡터로 구성하면 16차원 벡터가 된다. 즉, 한 맵당 16개의 특징값을 가진다. 그래서 마지막 층의 맵 수가 모두 12개인 것을 고려하면 최종 특징벡터는 192(16x12)차원이다. 이렇게 구한 특징벡터는 MLP(Multi-Layer Perceptron)분류기를 이용하여, 가중치, 바이어스, 커널을 학습시켜 인식한다. 예제의 실험은 60,000장의 영상을 한 번(epoch) 학습시키고 10,000장의 영상으로 검증한 결과 11% 에러를 보였으며, 같은 영상으로 학습만 백 번(epoch)으로 늘려 검증한 결과 1.2% 에러를 보인다[15][24].

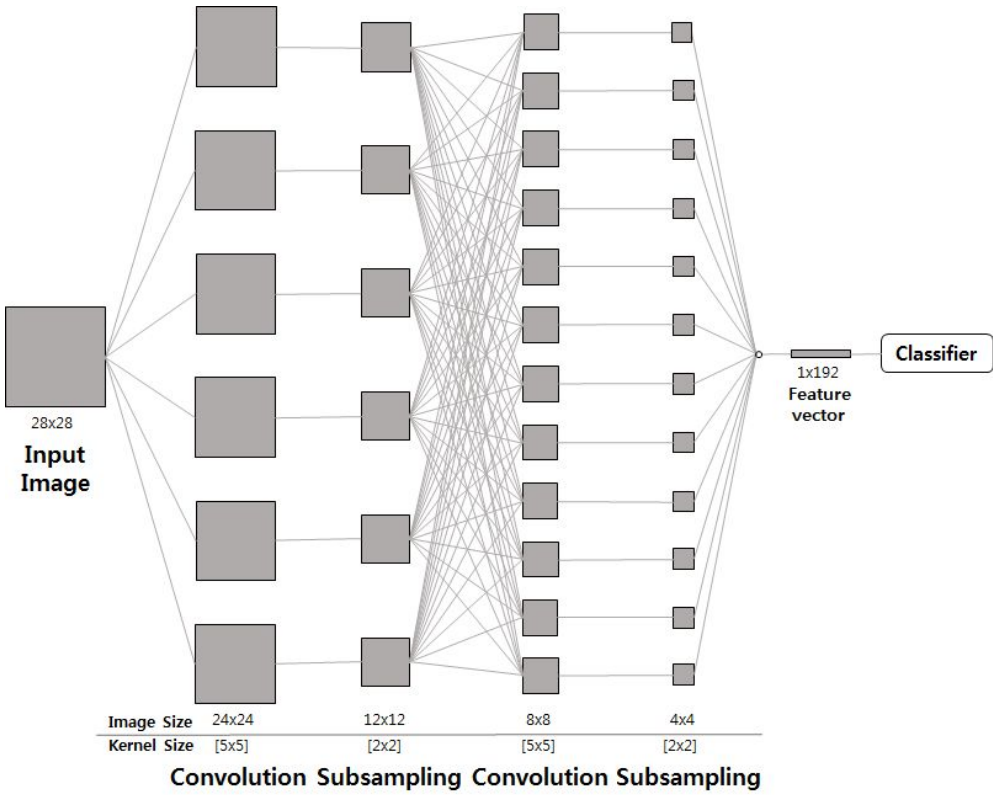


그림 2.15 2D-CNN의 일반적인 구조

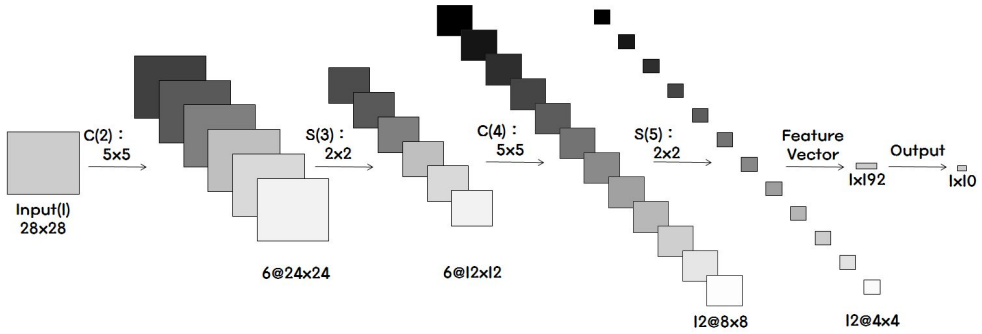


그림 2.16 2D-CNN(필기체 숫자인식 예)

그림 2.17은 필기체 숫자인식 예제의 중간 과정의 이미지를 보여주고 있다. 그림을 출력하는 방법은 한 영상의 픽셀 값들에 대해서 그 중 최대값으로 나누고, 히스토그램 평활화를 거쳤다.

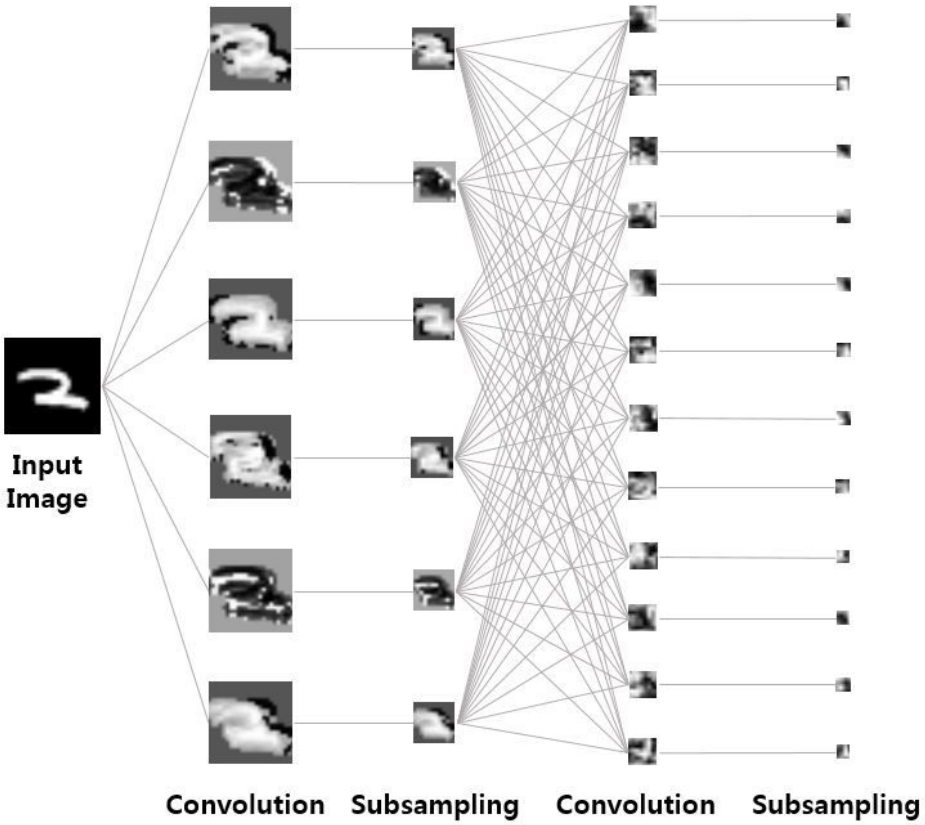


그림 2.17 2D-CNN에 의한 필기체 숫자인식 예제의 중간 과정 이미지

기존 방법들은 주로 한 장의 영상으로부터 특징값을 추출하였다. 그러나 네트워크기반 서비스 로봇환경에서는 영상을 서버로 전송해야 하기 때문에 여러 장의 시퀀스 영상을 가지고 있는 비디오에는 적절하지 않다. 사용자가 비협조적인 로봇환경에서는 한 장씩 인식하는 것보다는 여러 장을 한 번에 처리하는 것이 효율적이다[25].

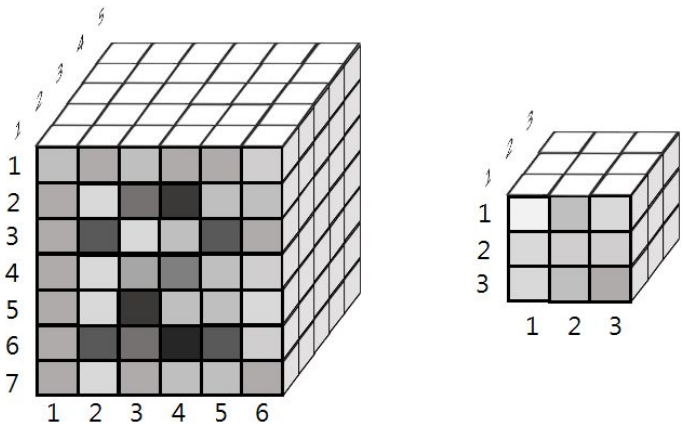
제3장 제안된 3D-CNN에 의한 얼굴인식

얼굴 인식 방법은 컴퓨터 비전을 이용하여 다양한 연구들이 진행되고 있다. 이 연구들을 큰 맥락에서 보면 얼굴 검출, 특징 추출, 얼굴 분류의 세 단계를 거쳐서 이루어진다[26].

일반적으로 CNN의 구조는 입력층, 은닉층, 출력층으로 구성되어 있으며, 은닉층의 개수, 커널의 크기, 특징 맵의 개수 등에 따라 다양한 구조 설계가 가능하다. 중요한 것은 은닉층에서 1차와 2차 콘볼루션층과 서브샘플링층이 반복되는 것이다[24].

제1절 3D 콘볼루션

영상에 콘볼루션을 수행하는 것은 영상의 특징을 추출하는 과정이다. 영상에서 3차원 콘볼루션은 홀수 배 크기의 커널의 중심이 영상의 픽셀에 놓인 상태에서 영상과 커널이 겹쳐진 부분들만 곱해서 더하여 출력 영상의 픽셀에 넣는 것이다. 이러한 과정은 영상의 전체 픽셀에 대해서 수행이 된다. 예를 들어 그림 3.1과 같이 7x6x5 사이즈의 영상과 3x3x3 커널이 있다.



(a) 3차원 영상　　(b) 3차원 커널

그림 3.1 콘볼루션에 대한 영상과 커널 예시(1)

그림 3.1의 (a)영상과 (b)커널에 대해 콘볼루션을 수행하게 하려면, 그림 3.2와 같이 콘볼루션을 계산할 (a)영상 위의 한 픽셀에 (b)커널의 중심을 일치시킨다. 그러면 (a)영상의 (1,1,1)부터 (3,3,3)까지 겹치는 것을 볼 수 있다. 겹치는 영역은 파란 선으로 나타내었고, 겹치는 영역의 중심 픽셀은 빨간 선으로 나타내었다.

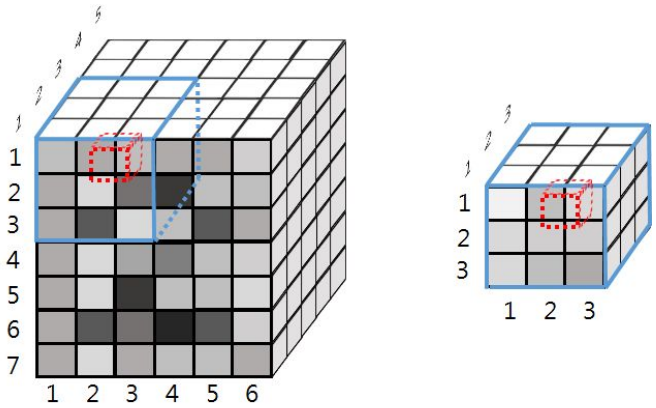


그림 3.2 콘볼루션에 대한 영상과 커널의 예시(2)

그림 3.3에서 영상과 커널의 콘볼루션을 계산하는 방법은 식(3-1)과 같다. 커널과 영상의 겹치는 픽셀 각각 곱하고 전체를 더한다. o는 출력을 의미한다.

$$\begin{aligned} o_{(1,1,1)} = {} & a_{(1,1,1)}b_{(1,1,1)} + a_{(1,2,1)}b_{(1,2,1)} + a_{(1,3,1)}b_{(1,3,1)} + a_{(2,1,1)}b_{(2,1,1)} + \\ & a_{(2,2,1)}b_{(2,2,1)} + a_{(2,3,1)}b_{(2,3,1)} + a_{(3,1,1)}b_{(3,1,1)} + a_{(3,2,1)}b_{(3,2,1)} + a_{(3,3,1)}b_{(3,3,1)} + \\ & a_{(1,1,2)}b_{(1,1,2)} + a_{(1,2,2)}b_{(1,2,2)} + a_{(1,3,2)}b_{(1,3,2)} + a_{(2,1,2)}b_{(2,1,2)} + \\ & a_{(2,2,2)}b_{(2,2,2)} + a_{(2,3,2)}b_{(2,3,2)} + a_{(3,1,2)}b_{(3,1,2)} + a_{(3,2,2)}b_{(3,2,2)} + a_{(3,3,2)}b_{(3,3,2)} + \\ & a_{(1,1,3)}b_{(1,1,3)} + a_{(1,2,3)}b_{(1,2,3)} + a_{(1,3,3)}b_{(1,3,3)} + a_{(2,1,3)}b_{(2,1,3)} + \\ & a_{(2,2,3)}b_{(2,2,3)} + a_{(2,3,3)}b_{(2,3,3)} + a_{(3,1,3)}b_{(3,1,3)} + a_{(3,2,3)}b_{(3,2,3)} + a_{(3,3,3)}b_{(3,3,3)} \end{aligned} \tag{3-1}$$

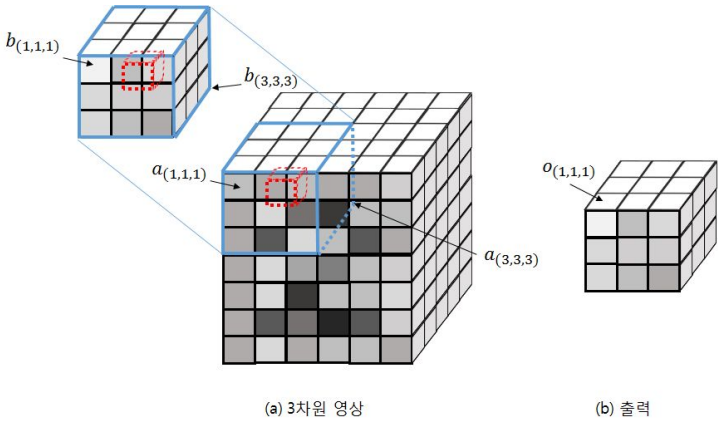
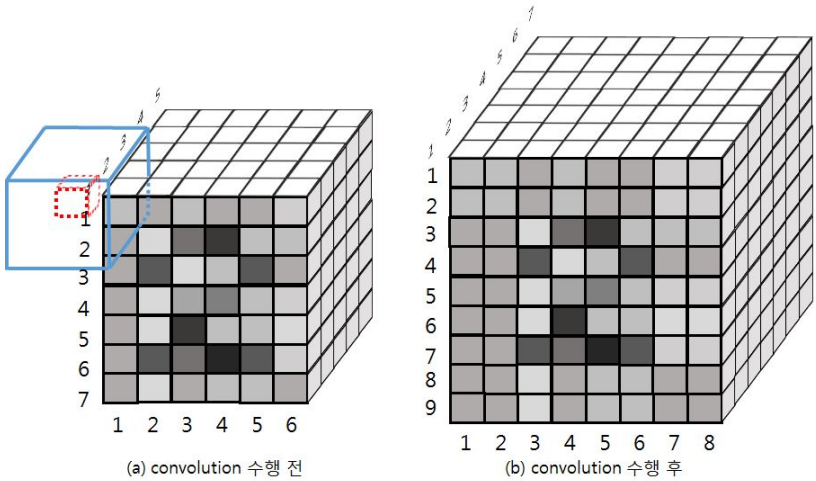


그림 3.3 콘볼루션에 대한 영상과 커널의 예시(3)

그림 3.3의 (a)영상에서 $a_{(2,2,2)}$에 있는 커널의 중심은 $a_{(2,2,3)}$으로 이동하여 콘볼루션을 수행하면 $o_{(1,1,2)}$가 구해진다. 커널의 중심이 영상의 끝 픽셀까지 이동하여 모두 구하면 콘볼루션이 완료된다.

앞서, 사이즈가 7x6x5인 영상과 3x3x3인 커널을 콘볼루션 연산을 하였다. 그러면, 콘볼루션 연산 결과로 출력된 영상의 크기는 콘볼루션 옵션에 따라 보통 3가지로 변경할 수 있다. 그림 3.4의 (a)와 같이 커널과 영상이 겹치는 픽셀이 1개일 때도 콘볼루션을 수행을 하게 되면, 원래 7x6x5인 영상은 3x3x3 커널의 콘볼루션 연산 결과로 그림 3.4의 (b)와 같이 9x8x7 사이즈가 나온다. 이때 사이즈를 계산하는 방법은 식(3-2)와 같다.

결과 영상의 크기 = 원래 영상의 크기 + (커널의 크기 - 1) (3-2)

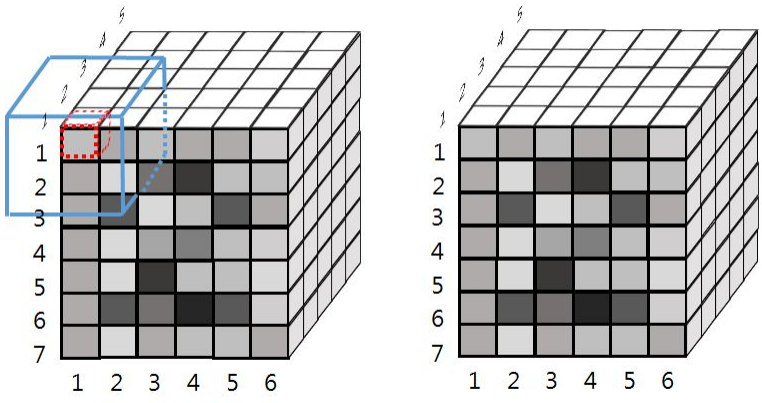


(a) convolution 수행 전 (b) convolution 수행 후

그림 3.4 콘볼루션 연산 후 영상의 크기가 증가

다음은 그림 3.5의 (a)와 같이 커널의 중심을 영상의 데이터가 존재하는 곳만을 일치시켜 콘볼루션 연산한 경우이고, 원래 7x6x5인 영상이 그림 3.5의 (b)와 같이 동일한 7x6x5 사이즈가 나온다. 이때 사이즈를 계산하는 방법은 식(3-3)와 같다.

결과 영상의 크기 = 원래 영상의 크기 (3-3)



(a) convolution 수행 전 (b) convolution 수행 후

그림 3.5 콘볼루션 연산 후 영상의 크기가 동일

다음은 그림 3.6의 (a)와 같이 커널이 영상의 데이터가 존재하는 부분 전체와 겹칠 경우만 콘볼루션 연산을 한 경우이다. 7x6x5인 영상과 3x3x3 커널에 대해 콘볼루션 연산을 한 결과로 그림 3.6의 (b)와 같은 5x4x3 사이즈가 나온다. 이때 사이즈를 계산하는 방법은 식(3-4)와 같다.

결과 영상의 크기 = 원래 영상의 크기 - (커널의 크기 - 1) (3-4)

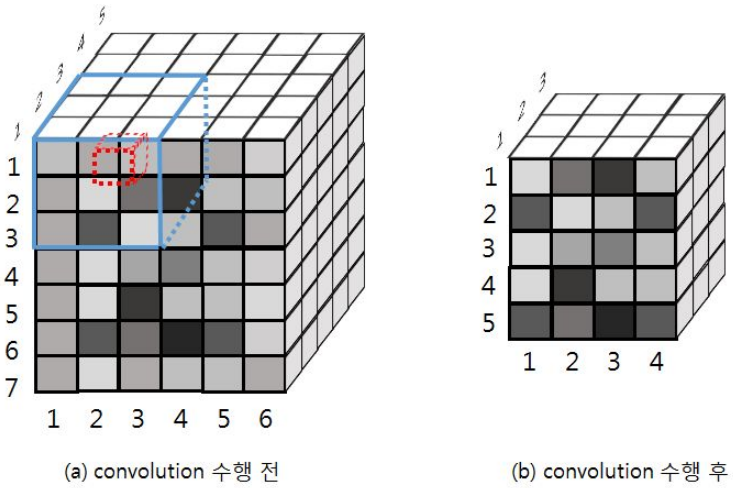


그림 3.6 콘볼루션 연산 후 영상의 크기가 감소

CNN에서 콘볼루션 층의 연산은 3번째 경우로서, 커널이 영상과 완전히 포개어 수행한다.

제 2절 서브샘플링

영상에서 서브샘플링을 수행하는 것은 영상의 이동, 회전, 크기 변화에 불변성을 갖도록 하고, 영상의 사이즈를 자연수 배로 축소시킨다. 그 방법은 영상을 축소 배수와 동일한 사이즈의 커널로 평균 콘볼루션을 연산하고, 그 결과 영상에서 축소 배수만큼 건너 띄면서 픽셀을 취득하여 감소된 영상을 얻는다. 예로 그림

3.7의 (a)와 같이 사이즈가 8x6인 영상이 있고, 서브샘플링을 통해 2배 축소하여 4x3 영상을 얻으려고 한다. 그러기 위해서 그림 3.7의 (b)와 같이 축소 배수의 크기인 2x2로 커널을 정의한다.

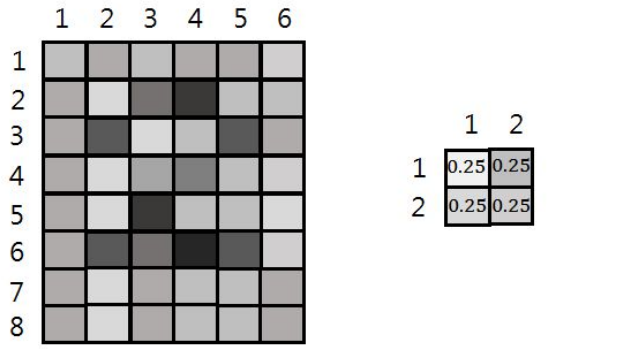


그림 3.7 서브샘플링에 대한 영상과 커널 예시

그 다음, 평균을 구하기 위해 커널의 값을 초기화하는데, 그 값을 구하는 방법은 식 (3-5)와 같다.

커널의 픽셀 값 = 1 / 축소 배수의 제곱 (3-5)

식 (3-5)를 통해서 커널의 모든 픽셀에 대입될 값은 그림 3.7의 (b)와 같이 $\frac{1}{2^2}=\frac{1}{4}=0.25$이다.

그림 3.8의 (a)에서처럼 영상과 평균 커널간에 커널과 영상이 완전히 겹친 콘볼루션 연산을 수행한 결과로 그림 3.8의 (b)와 같이 원본 영상보다 사이즈가 감소한 7x5 사이즈를 얻는다. 그 다음, 축소 배수씩 건너서 샘플링을 하게 되면, 그림 3.8의 (b)처럼 x 표시의 픽셀정보가 지워지고, 최종적으로 그림 3.8의 (c)와 같은 4x3 사이즈의 결과가 얻어진다.

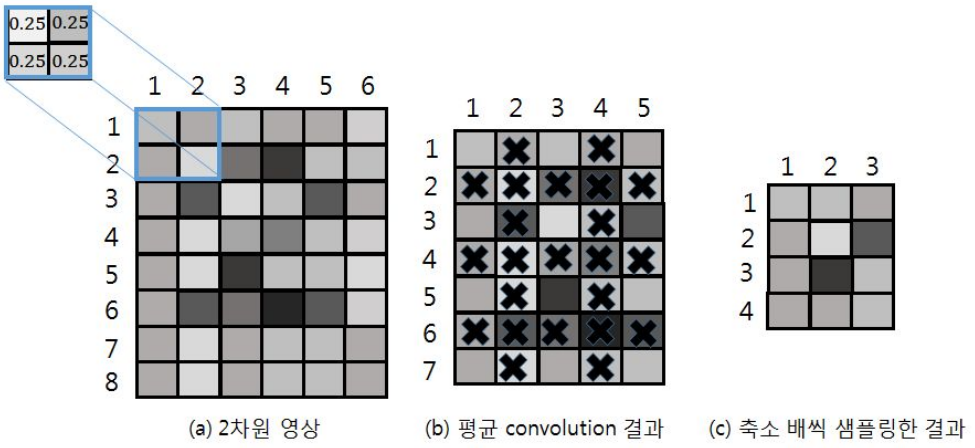


그림 3.8 서브샘플링 과정

3차원 CNN 구조에서 서브샘플링을 위해 커널을 2x2x2로 정의한다면 시간축도 절반으로 축소가 가능할 것이다. 하지만 본 논문에서는 콘볼루션 층에서만 시간축의 사이즈 감소를 하고 있다. 여기에서 적용한 서브샘플링 방법은 시간축의 사이즈가 5라면 각 영상 5장 개별적으로 2차원 서브샘플링을 수행하는 것이다[15][16].

제 3절 3차원 CNN의 구조 및 분류기

CNN의 구조는 다양하게 설계가 가능하다. 예를 들어, 콘볼루션 층 또는 서브샘플링 층의 수를 늘리거나 줄일 수 있고, 콘볼루션 층의 특징 맵의 개수도 변경이 가능하다. 여기에서 설명하는 구조는 본 논문의 실험에 적용된 3차원 CNN 구조를 기반으로 한다.

그림 3.9는 CNN의 구조를 나타내기 전에 콘볼루션 층만 간단한 모델로 보인 것이다. 하나의 음영 사각형은 한 맵을 의미하고, 'k-1'은 'k'보다 이전 층을 의미한다. 'k'층에서 입력맵의 개수는 'k-1'층의 맵수인 n개이고, 출력맵의 개수는 'k'층의 맵수인 m개 이다. 'k'층에서 한 맵이 출력되기 위해서는 입력층의 모든 맵마다 각각 정의된 커널과 콘볼루션이 수행되어 합쳐진다.

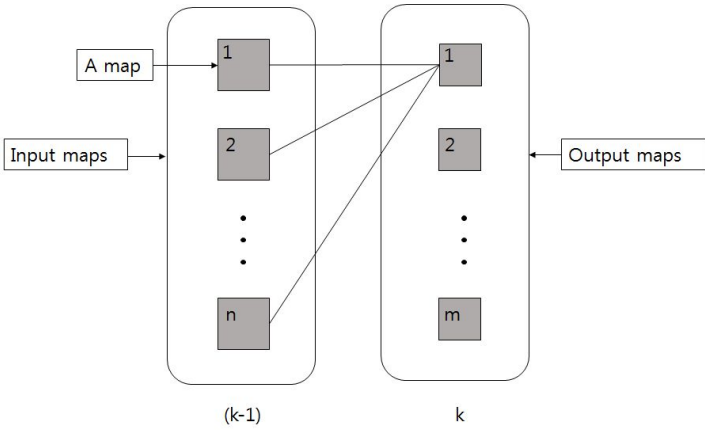


그림 3.9 콘볼루션 층의 구조

그림 3.10은 CNN의 구조를 나타내기 전에 서브샘플링 층만 간단한 모델로 보인 것이다. 하나의 음영 사각형은 한 맵을 의미하고, 'k-1'은 'k'보다 이전 층을 의미한다. 서브샘플링 층은 입력맵의 개수가 출력맵의 개수와 같다. 이전 층과 동일한 순서로 영상의 사이즈가 줄어든다.

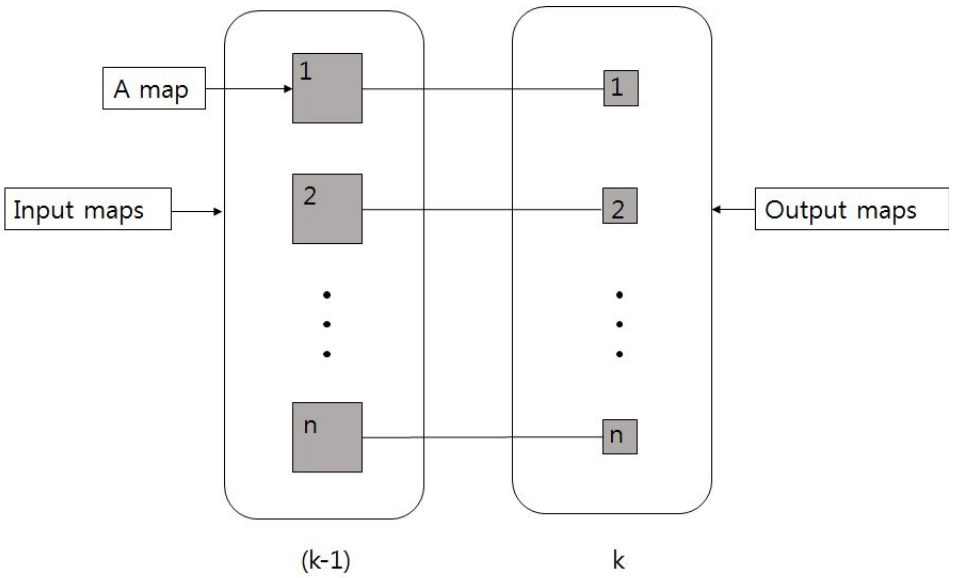


그림 3.10 서브샘플링 층의 구조

그림 3.11은 3차원 CNN을 이용한 비디오기반 얼굴인식의 CNN 구조를 보여주고 있다. 여기서 3차원 CNN은 5층으로 설계가 되어있다. 1층은 입력층이고, 2층은 콘볼루션 층이고, 3층은 서브샘플링 층이고 4층은 콘볼루션 층이고, 5층은 서브샘플링 층이다. 초기 커널 값들은 특정영역의 임의 값으로 설정되며, 1층에서 10 장으로 이뤄진 1개의 3차원 데이터가 입력으로 들어가고, 그 영상으로부터 2층에서 1개 맵으로 콘볼루션 되어 특징이 추출되고, 3층에서 서브샘플링을 거쳐 영상의 사이즈를 줄이고, 다시 1개의 입력맵으부터 4층에서 13개 맵으로 콘볼루션 되어 특징이 추출되고, 5층에서 서브샘플링을 거쳐 영상의 사이즈를 줄이고, 마지막 층에서 영상을 단일 벡터로 구성하고, 13개의 맵을 모두 이어 붙이면 한 입력에 대한 특징벡터가 구해진다.

입력으로 사용된 데이터는 44x40 사이즈의 얼굴영상이 10프레임 겹쳐진 비디오기반 얼굴데이터이다. 영상의 사이즈의 변화는 44x40x10 사이즈의 입력 영상이 입력층에 들어오고 2층의 커널사이즈가 5x5x5인 콘볼루션을 수행하게 된다. 여기에서 콘볼루션은 커널과 영상이 완전히 포개어지는 것을 사용하기 때문에 결과 영상은 '원본영상의 크기 - (커널의 사이즈 - 1)'이므로 40x36x6 사이즈의 영상이 출력된다. 그 다음, 3층에서 이동과 회전, 크기 변화에 불변성을 갖도록 하면서 영상의 사이즈를 줄이는 서브샘플링을 수행한다. 이때 커널의 사이즈는 2x2이므로 영상의 크기는 절반으로 줄어들어 20x18x6이 된다. 그 다음, 커널사이즈가 5x5x6인 콘볼루션이 수행되어 16x14x1 사이즈의 영상이 출력된다. 5층에서 다시 2x2 커널 사이즈의 서브샘플링을 수행함으로서 절반 줄어 사이즈가 8x7x1인 영상이 출력된다. 이 영상을 단일 벡터로 구성하면 56차원 벡터가 된다. 즉, 한 맵당 56개의 특징값을 가진다. 그래서 마지막 층의 맵 수가 모두 13개인 것을 고려하면 최종 특징벡터는 728(56x13)차원이다.

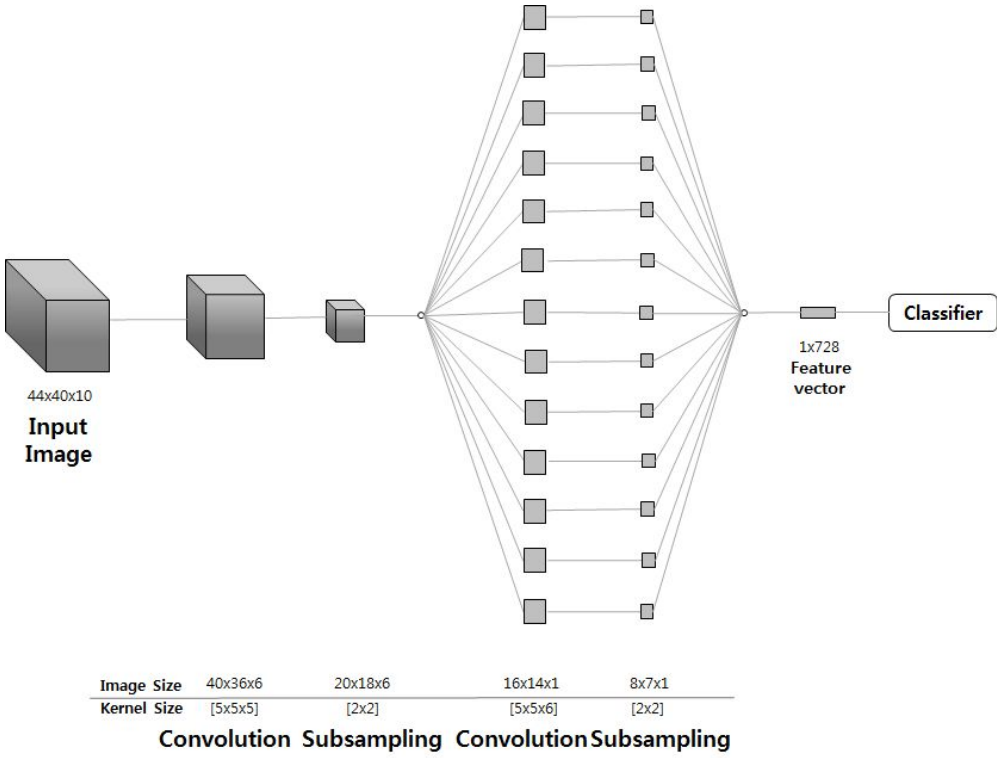


그림 3.11 제안된 3D-CNN의 구조

그림 3.12은 3차원 CNN을 이용한 비디오기반 얼굴인식의 중간과정의 이미지를 보여주고 있다. 그림을 출력하는 방법은 한 영상의 픽셀 값들에 대해서 그 중 최대값으로 나누고, 히스토그램 평활화를 거쳤다[15][16].

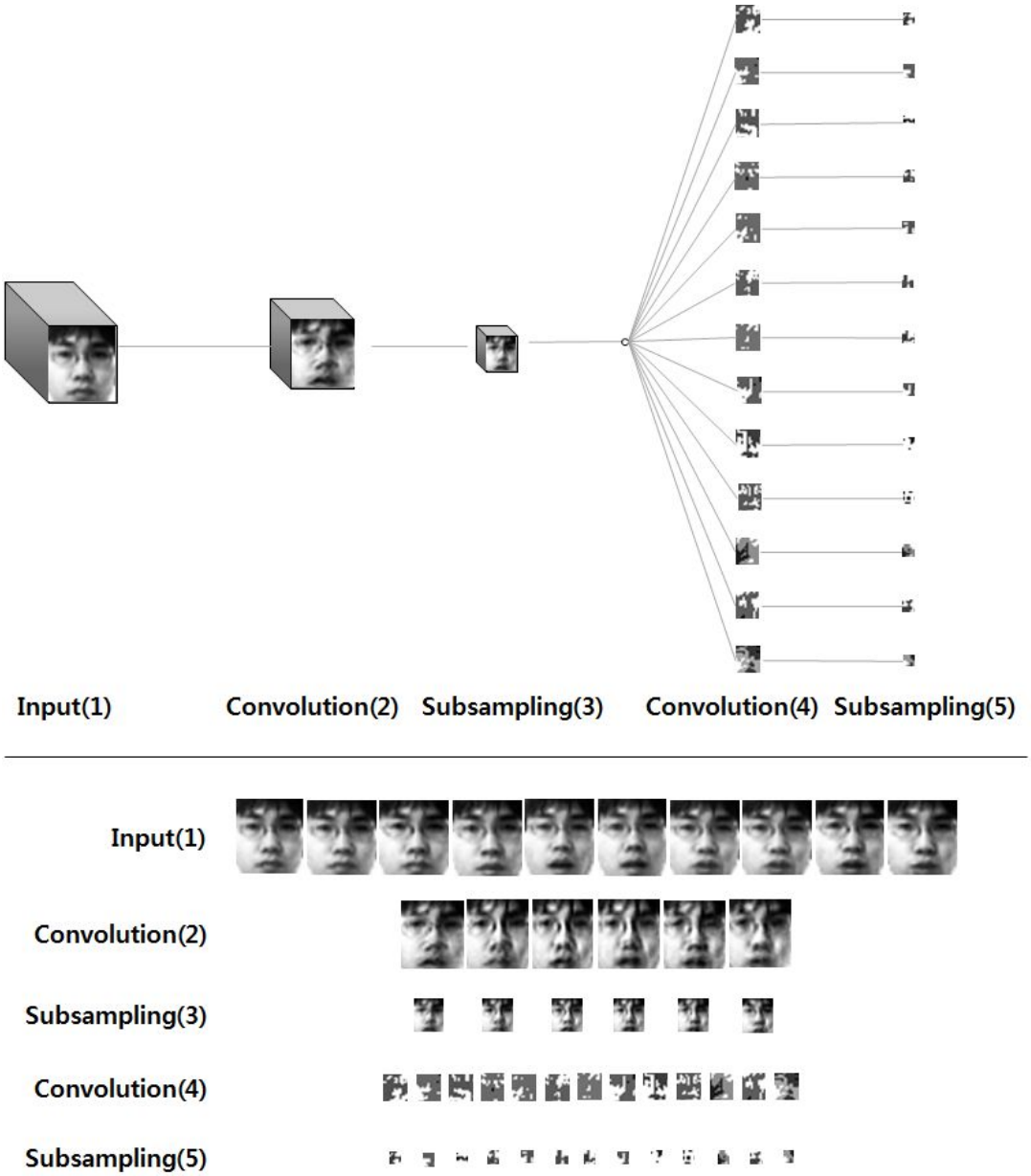


그림 3.12 3D-CNN을 이용한 비디오기반 얼굴인식의 중간 과정 이미지

분류방법은 CNN을 통해 얻어진 학습과 검증의 특징벡터간의 유클리디언 거리를 계산하여 가까운 것으로 인식한다. 식(4-1)은 두 특징벡터간의 유클리디언 거리를 구하는 식이다[20].

$$d(X,Y)=\sum_{i=1}^{n}|x_i-y_i| \tag{4-1}$$

제4장 실험 및 결과 분석

제 1절 얼굴인식 데이터베이스 설명

본 논문에서 제안한 3D-CNN을 이용하여 특징벡터를 추출하고 학습과 검증의 특징벡터 사이의 거리가 가까운 것으로 인식하는 방법의 성능을 평가하기 위해, ETRI에서 만든 Face Video 데이터베이스를 사용하였다. 이 데이터베이스는 웨버2(Wever-2) 로봇에 장착된 카메라로부터 일반 가정집 환경과 비슷한 u-로봇 테스트베드환경에서 구축하였다. 그림 4.1은 웨버2 로봇을 그림 4.2은 u-로봇 테스트베드 환경을 보여주고 있다.

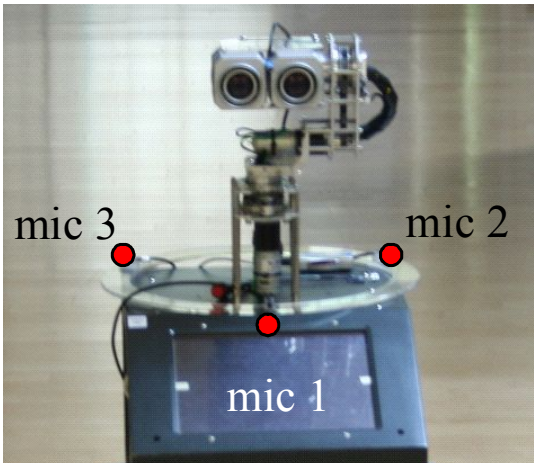


그림 4.1 웨버2(Wever-2)로봇

그림 4.2 u-robot 테스트 방의 환경

카메라로부터 획득된 영상은 영상 자체의 2차원에 연속된 프레임을 나타내는 시간축이 더해져 3차원 데이터가 된다. 그리고 다양한 환경을 고려해서 카메라로부터 1m, 2m 거리에 따라 획득되었다. 그림 4.3는 거리에 따른 얼굴 비디오 예를 보여주고 있다.

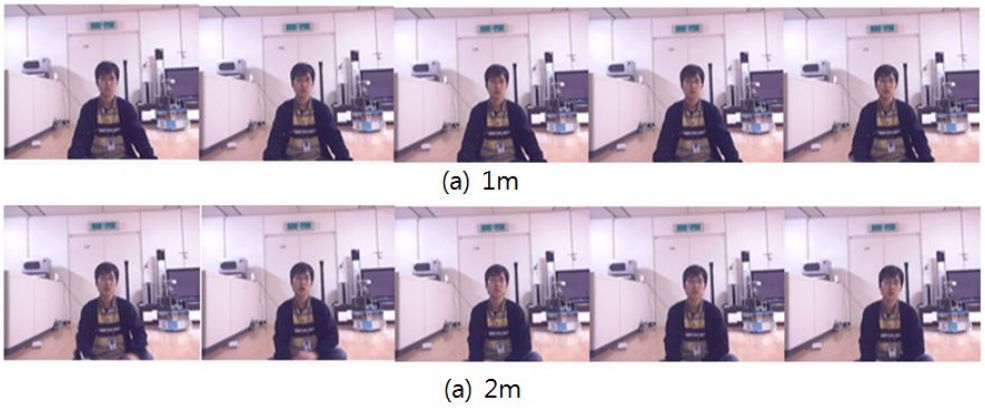


그림 4.3 거리에 변화를 가진 연속적인 영상들의 예

우선적으로, 우리는 아다부스팅(Adaboosting)과 RMCT(Revised Modified Census Transform) 기반 얼굴 검출을 수행했다. 이것은 3단계로 구성된다. 첫번째 단계에서, 원치 않는 변수들을 보상하기 위해 RMCT를 이용한 전처리(Preprocessing)를

수행한다. 두번째 단계에서, 2D-LS(Logarithmic Search)와 결합한 피라미드(Pyramid) 영상차가 수행되어 얼굴 후보 영역을 추출한다. 그런 뒤, 아다부스팅 알고리즘을 이용한 얼굴검출이 수행된다. 끝으로, 얼굴의 크기, 위치, 회전과 신뢰도를 포함한 얼굴정보를 획득하기 위해 FCM(Face Certainty Map)을 사용했다[27].

이러한 얼굴검출 알고리즘을 통해 획득된 얼굴영상들은 그레이스케일과 히스토그램 평활화를 거쳐 얼굴인식에 좋도록 처리를 한다. 그림 4.4은 거리에 따른 얼굴인식에 사용될 최종 영상들의 예를 보여주고 있다.

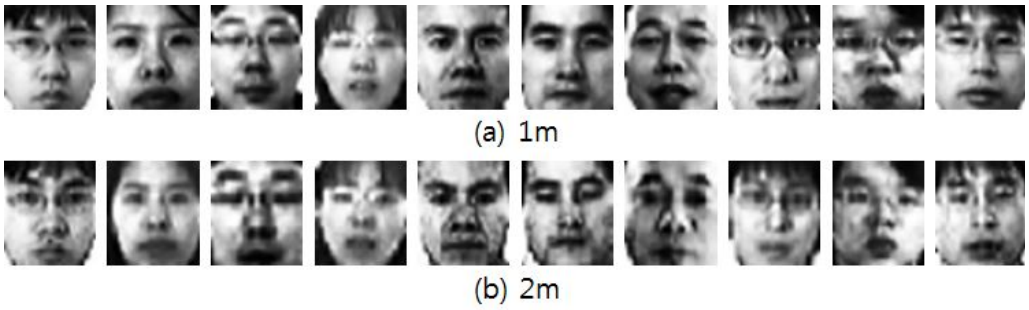


그림 4.4 얼굴검출된 얼굴영상들의 예

ETRI Face Video 데이터베이스에는 10명에 대한 얼굴 데이터가 있다. 모든 단일 얼굴 영상의 사이즈는 45x40이다. 이러한 단일 영상을 연속하는 10프레임씩 묶음으로서 사이즈가 45x40x10인 하나의 얼굴 비디오 데이터가 된다. 그림 4.5는 얼굴 비디오 데이터의 3차원 예시를 보여주고 있다. 실험에서 3차원 CNN의 구조에 맞추기 위해 영상의 크기인 45x40x10을 44x40x10으로 변경한다.

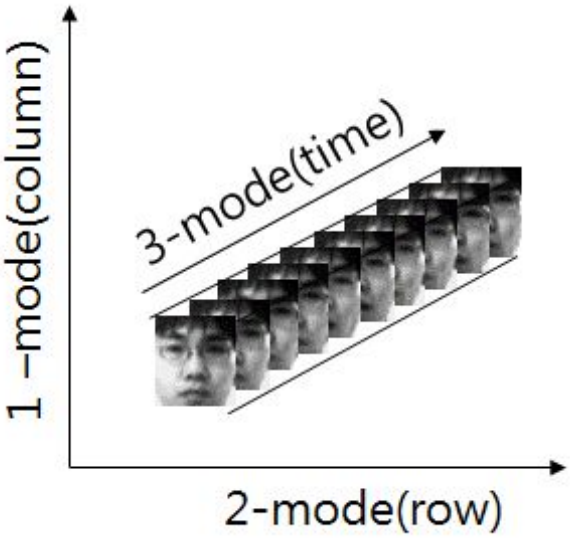


그림 4.5 연속영상을 갖는 얼굴영상들의 예

제 2절 실험결과

4장 1절에서 설명한 얼굴 비디오 데이터베이스를 활용하여 3차원 CNN의 특징값을 이용하는 얼굴인식방법의 우수성과 효율성을 증명한다.

성능 분석을 위해 사용된 컴퓨터의 하드웨어 사양은 CPU 2.80GHz, Intel(R) Core(TM) i7, 메모리 4GB이며, 소프트웨어는 MATLAB R2013A버전을 사용하였다.

실험에서 학습데이터는 로봇으로부터 1m거리에서 획득한 45x40x10의 3차원 데이터이며, 10명에 대해 5개씩 50개로 구성하여 학습데이터의 사이즈는 45x40x10x50이다. 검증 데이터는 로봇으로부터 1m거리에서 획득한 다른 45x40x10의 3차원 데이터이며 10명에 대해 5개씩 50개로 구성하여 첫 번째 검증 데이터의 사이즈는 45x40x10x50이다. 두 번째 검증 데이터는 2m거리에서 획득한 45x40x10의 3차원 데이터이며 10명에 대해 20개씩 200개로 구성하여 두 번째 검증데이터의 사이즈는 45x40x10x200이다. 이러한 데이터들은 CNN 구조에 맞추기 위해 영상의 사이즈가 44x40로 변경되도록 하였다.

실험에서 사용한 3차원 CNN의 구조는 표 4-1과 같다.

표 4-1. 3차원 CNN 구조

Layer	type	kernel size
1	Input	
2	Convolution	5x5x5
3	Subsampling	2x2
4	Convolution	5x5x6
5	Subsampling	2x2

표 4-1과 같은 3차원 CNN의 구조에 모든 학습데이터 50개에 대해 특징 벡터를 구한다. 이어서 첫 번째 검증데이터 50개에 대한 특징 벡터도 구한다. 그리고 첫 번째 검증데이터의 첫 번째 특징벡터와 학습데이터 50개에 대해 각각 거리를 계산한다. 그 다음 거리가 가장 가까운 최소값을 찾아 그것의 해와 동일한 것으로 인식한다.

여기서 학습데이터는 교사 데이터이므로 해를 알고 있고, 검증데이터도 교사데이터이지만 인식과정에서는 비교사로 간주한다. 검증데이터의 해는 인식 성공률을 구하는 과정에서만 사용한다. 이러한 과정은 검증데이터의 특징벡터들 전반에 걸쳐 수행됨으로서 첫 번째 검증데이터 50개에 대한 인식이 수행된다.

마찬가지로 두 번째 검증데이터 200개에 대해서도 동일한 과정을 거친다. 그 후, 인식 성공한 데이터 수를 전체 데이터 수로 나누어 인식률을 계산한다.

다양한 실험을 위해 2층과 4층의 특징 맵 개수에 변화를 주었다. 2층의 특징 맵 개수는 1부터 30까지, 4층의 특징 맵 개수는 1부터 60까지 변화를 주었다. 그리고 표 4-2는 모든 검증 데이터의 실험 결과에 대한 정리를 나타낸다. 최고의 인식률과 그 때의 2, 4층의 맵 수를 보여준다.

표 4-2. 실험결과

	맵의 수		인식률
	2층	4층	
1m 검증	1	13	100%
2m 검증	28	3	88%

아래 그림 4.6은 사람 1의 3차원 CNN의 중간과정 영상들이고, 그림 4.7은 사람 2의 3차원 CNN의 중간과정 영상들이다.

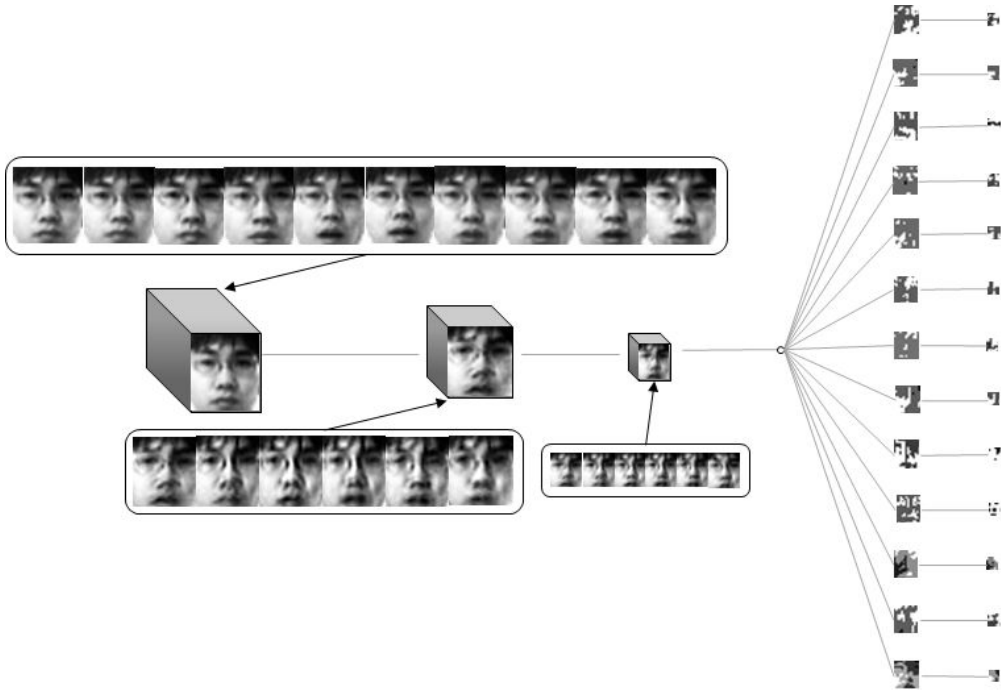


그림 4.6 2층 1개, 4층 13개 일 때 1번 얼굴의 중간과정

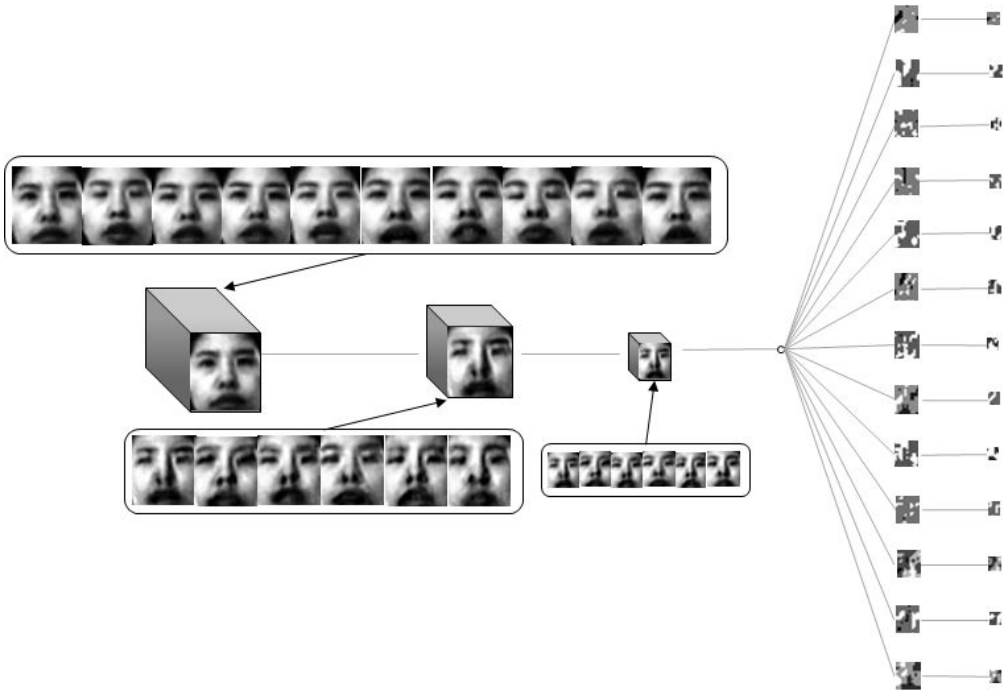


그림 4.7 2층 1개, 4층 13개 일 때 2번 얼굴의 중간과정

본 논문에서 나타낸 3차원 CNN을 이용한 인식결과와 주성분분석기법(Principle Component Analysis)과 2차원 콘볼루션 신경회로망(CNN: Convolutional Neural Network)을 이용한 인식결과들의 비교를 위해 표 4-3에 각 인식률 나타내었다.

표 4-3. 3차원 CNN과 주성분 분석의 비교

	1m	2m
PCA[13]	99.6%	78.3%
2D-CNN[15]	100%	83%
제안된 3D-CNN	100%	88%

3차원 콘볼루션 신경회로망을 이용하여 얼굴비디오 인식을 한 결과와 주성분

분석기법과 2차원 콘볼루션 신경회로망을 이용하여 얼굴비디오 인식을 한 결과를 서로 비교해보면 3차원 콘볼루션 신경회로망을 이용하여 얼굴비디오 인식을 한 경우가 2m에서 5% 향상된 더 좋은 인식률을 보여주었다.

5장 결론

본 논문에서는 로봇환경에서 쓰일 수 있는 비디오 기반 얼굴인식을 위해 3차원 콘볼루션 신경회로망(CNN: Convolutional Neural Network) 구조의 특징벡터를 사용하였다. 비디오 얼굴 데이터는 연속된 프레임을 갖기 때문에 단일 프레임과 다른 장점이 있다. 이러한 장점을 극대화 시키는데 3차원 콘볼루션 신경회로망 구조가 이전 프레임과의 특징도 뽑기 때문에 적당하다. 3차원 콘볼루션 신경회로망의 특징벡터를 이용해서 학습데이터와 검증데이터의 거리가 가까운 것으로 인식하였다. 학습데이터는 1m에서 획득한 50개를 사용하고, 검증데이터는 다른 1m에서 획득한 50개와 2m에서 획득한 200개이다. 실험 결과 1m에서 맵 수가 2층이 1이고 4층이 13일 때 인식률이 100%가 나왔다. 이 경우 말고도 100%는 많이 나왔고, 그 중에 하나를 나타낸 것이다. 2m에서는 맵 수가 2층이 28이고 4층이 3일 때, 최대 88%가 나왔다. 이 최대치가 다른 경우에서 중복되어 나오지는 않았다.

본 논문에 나타낸 3차원 콘볼루션 신경회로망을 이용한 인식 결과를 기존 얼굴인식방법들인 주성분분석기법(PCA: Principal Component Analysis) 와 2차원 콘볼루션 신경회로망을 이용한 인식 결과들과 비교하였고, 그 결과 3차원 콘볼루션 신경회로망을 이용한 인식 결과가 더 좋음을 확인 할 수 있었다.

본 논문의 3차원 콘볼루션 신경회로망은 콘볼루션 층에서만 3차원을 적용하였다. 다음에 서브샘플링 층에도 3차원을 적용하여 연구할 것이다.

3차원 콘볼루션 신경회로망을 이용한 인식 결과는 맵 수에 따라 다양한 인식률을 얻었다. 분류 알고리즘을 보통 퍼셉트론을 사용하는데, 그래픽 연산 장치(GPU: Graphics Processing Unit)를 사용하지 못해 실행시간이 오래 걸려 충분한 실험을 하지 못하였다. 향후, 그래픽 연산 장치를 이용하는 프로그래밍을 통해 여러 가지 분류 알고리즘을 비롯한 다양한 실험을 통해 인식률을 높일 것이다.

참고문헌

[1] A. S. Sekmen, M. Wilkes, K. Kawamura, "An application of passive human-robot interaction: human tracking based on attention distraction", IEEE Trans. on Systems, Man, and Cybernetics-Part A, vol. 32, no. 2, pp. 248-259, 2002.

[2] X. Yin, M. Xie, "Finger identification and hand posture recognition for human-robot interaction", Image and Vision Computing, vol. 25, pp.1291-1300, 2007.

[3] Y. Sugimoto, Y. Yoshitomi, S. Tomita, "A method for detecting transitions of emotional states using a thermal facial image based on a synthesis of facial expressions", Robotics and Autonomous Systems, vol. 31, pp. 147-160. 2000.

[4] J. J. Lien, T. Kanade, J. F. Cohn, C. C. Li, "Detection, tracking, and classification of action units in facial expression", Robotics and Autonomous Systems, vol. 31, pp. 131-146. 2000.

[5] G. Medioni, A. R. J. Francois, M. Siddiqui, K. Kim, H. Yoon, "Robust real-time vision for a personal service robot", Computer Vision and Image Understanding, vol. 108, pp. 196-203, 2007.

[6] 특허청, 인간-로봇상호작용(HRI) 특허출원동향, 2005.

[7] 정보통신연구진흥원, 인간-로봇상호작용 기술(지능형 서비스 로봇분야), 2006.

[8] 한국전자통신연구원, HRI 로드맵, 2007

[9] K. C. Kwak, D. H. Kim, B. Y. Song, D. H. Lee, S. Y. Chi, and Y. J. Cho, "Vision-based Human-Robot Interaction Components for URC intelligent service robots," IROS'06, video session, Beijing, 2006.

[10] 반규대, 곽근창, 지수영, 정연구, “로봇환경의 템플릿 기반 얼굴인식 알고리즘 성능비교”, 한국로봇공학논문지, vol. 2, no. 3, 2007, pp. 270-274,

[11] D. H. Kim, J. Lee, H. S. Yoon, E. Y. Cha, "A non-cooperative user

authentication system in robot environments", IEEE Trans. on Consumer Electronics, vol. 53, no. 2, 2007, pp. 804-811.

[12] W. H. Yun, D. H. Kim, H. S. Yoon, "Fast group verification system for intelligent service", IEEE Trans. on Consumer Electronics, vol. 53, no. 4, pp. 1731-1735, 2007.

[13] M. Turk and A. Pentland, "Face recognition using eigenfaces," Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, pp. 586-591, 1991.

[14] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 711-720, 1997.

[15] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural-Network Approach", IEEE Trans. on Neural Networks, Vol. 8, No. 1, pp. 98-113, 1997.

[16] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3D Convolutional Neural Networks for Human Action Recognition", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 35, No. 1, pp. 221-231, 2013.

[17] Jialue Fan, Wei Xu, Ying Wu, and Yihong Gong, "Human Tracking Using Convolutional Neural Networks", IEEE Trans. on Neural Networks, Vol. 21, No. 10, pp. 1610-1623, 2010.

[18] 곽근창, 윤호섭, "u-로봇을 위한 인간-로봇상호작용기술의 연구동향 및 발전전망", SK Telecom Telecommunications Review, Vol. 18, No. 3, pp.385-402, 2008.

[19] 신윤희, 주진선, 김은이, T. Kurata, A. K. Jain, S. Park, K. Jung, "HCI를 위한 트리 구조 기반의 자동 얼굴 표정 인식," 한국산업정보학회논문지, vol. 12, no. 3, pp. 60-68, 2007.

[20] 이명원, "동영상으로부터 텐서표현을 통한 얼굴 표정 인식", 조선대학교, 2012.

[21] K. Etemad and R.Chellappa, "Discriminant analysis for recognition

of human face images," *J. Optical Society of America*, vol. 14, pp. 1724-1733, 1997.

[22] J. Yang, D. Zhang, A.F. Frangi, and J.Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131-137, 2004.

[23] 최종무, "얼굴 인식을 위한 저차원 영상표현," 성균관대학교 박사 학위 논문, 2003.

[24] http://www.mathworks.com/matlabcentral/fileexchange/38310-deep-learning-toolbox

[25] Y. H. Han, and K. C. Kwak, "Face representation and recognition using third-order Tensor-based MPCA method", Journal of Korean Institute of Information Technology, Vol. 9, No. 6, pp.147-154, 2011.

[26] M. Pantie and L. J. M Rothkrantz, "Automatic Analysis of Facial Expressions: the State of the Art," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp 1424-1445, 2000.

[27] W. H. Yun, D. H. Kim, and H. S. Yoon, "Fast Group verification system for intelligent robot service," IEEE Trans. on Consumer Electronics, Vol. 53, No. 4, pp. 1731-1735, 2007.