논문 요약.

논문명.[[1]](#footnote-1)

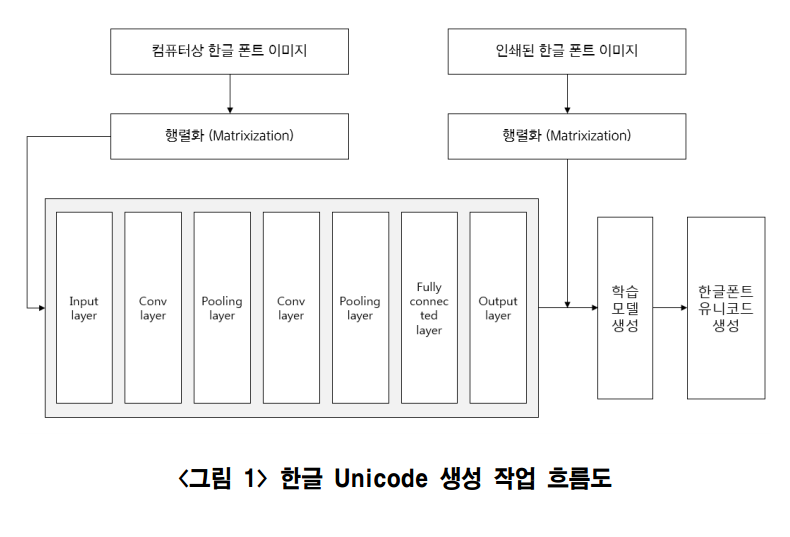
1. 텐서플로우(Tensorflow)를 사용한 한글 폰트의 특징 검출과 Unicode 자동 생성을 위한 알고리즘
2. 텐서플로를 이용한 OCR 시스템 개발
3. 온라인 쇼핑몰에서 상품 설명 이미지 내의 키워드인식을 위한 딥러닝 훈련 데이터 자동 생성 방안
4. 다양한 딥러닝 알고리즘과 활용.
5. 영상처리 기법을 이용한 문자인식 시스템 개발
6. 모바일 기반의 문자인식 시스템.
7. 딥러닝의 모형과 응용사례
8. 합성곱 신경망과 극한 학습기계의 융합기반의 딥러닝 신경망을 이용한 차선 검출 성능 향상
9. 텐서플로우(Tensorflow)를 사용한 한글 폰트의 특징 검출과 Unicode 자동 생성을 위한 알고리즘

딥 러닝 중 CNN[[2]](#footnote-2)사용.

한글 폰트의 특징 : 형태는 글자의 크기, 장평, 굵기 등에 영항을 받는다.

일반적인 폰트는 9~12 포인트 사이이며 6포인트 이하의 글자는 알아보기 힘들다.

문서용 한글 폰트는 종이에 출력된 상태로 감식되기 때문에 사용한 프린터나 종이의 상태에 따라 형태에 왜곡이 올 수 있으므로 이를 고려한다.



위의 작업 흐름도를 바탕으로 알고리즘 만들기.

CNN( 이하 설명.)

input layer와 output layer 사이의 hidden layer에 convolutional layer와 pooling layer를 배치한 것으로 이 두 layer에서는 이미지의 해상도를 낮추거나 샘플링하는 처리를 계속 반복한다 Convolutional layer는 입력 이미지의 일부에 가중치 필터(Filter)를 적용해가며 분류에 도움이 될 만한 특징들을 추출하여 특징맵(Feature map)을 만든다. Pooling layer는 convolutionallayer에서 얻은 특징맵에서 가장 핵심적인 부분을 추출(Sub sampling)하여특징맵을 축소한다. 특징을 유지한 상태로 축소하므로 위치 변경으로 인한 결과 변화를 막아주고 데이터 사이즈의 축소를 통해 신경망의 성능을 높여준다. 이렇게 추출된 특징들을 기반으로 분류를 한다.

CNN 사용 예시.

1. 폰트 원본에서 폰트 이미지를 추출한다.
2. 추출한 이미지를 신경망에 넣기 위해 이미지 매트릭스로 변환한 후 그림 2와 같이 정규화를 시킨다.
3. 변환된 이미지 매트릭스를 CNN 입력 데이터로 넣은 뒤 두 번의 convolutional layer와 pooling layer 단계를 거쳐 여러 64개의 특징맵을 찾는다.
4. 구한 특징맵을 이용하여 Fully connected layer에 넣어서 hypothesis를 세우고 학습모델을 만들어, 이후 인쇄된 전자 문서의 한글 폰트 이미지가 입력으로 들어왔을 때 해당 폰트가 어떤 폰트인지 예측할 수 있도록 한다.

아래는 코드의 일부.

.

<텐서플로우 코드 일부>

x = tf.placeholder(tf.float32, [None, 784])

ximg = tf.reshape(x, [-1, 28, 28, 1])

y = tf.placeholder(tf.float32, [None, n]) # n is number of font class

# convolutional layer phase1

w1 = tf.Variable(tf.random\_normal([3,3,1,32], stddev=0.01))

l1 = tf.nn.conv2d(ximg, w1, strides=[1, 1, 1, 1], padding=‘SAME’)

l1 = tf.nn.relu(l1)

l1 = tf.nn.max\_pool(L1, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1],

padding=’SAME’)

# convolutional layer phase2....

# fully connected layer

w3 = tf.get\_variable(“w3”, shape=[7\*7\*64, n],

initializer=tf.contrib.layers.xavier\_initializer)

b = tf.Variable(tf.random\_normal([n]))

h = tf.matmul(l2, w3) + b

cost = tf.reduce\_mean(tf.nn.softmax\_cross\_entropy\_with\_logits(

logits=h, labels=y)

opt = tf.train.AdamOptimizer(learning\_rate=learning\_rate)

.minimize(cost)

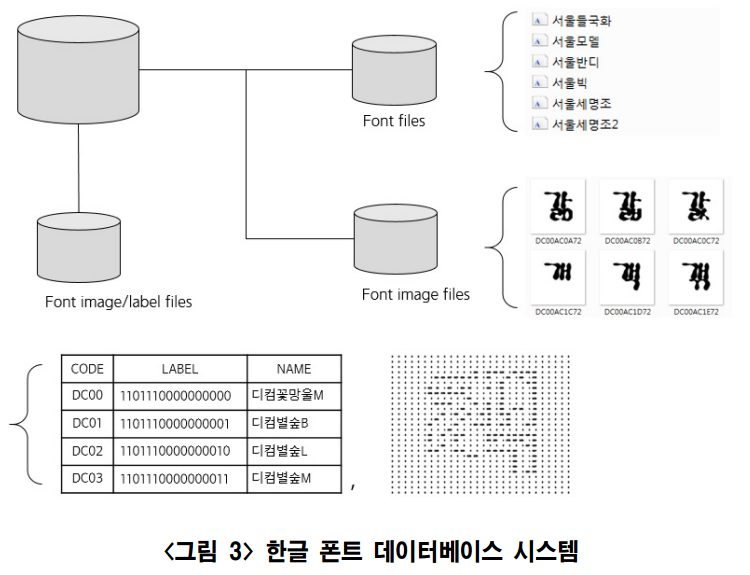
#train model....

#test model....

.

.

본 논문에서는 한글 폰트의 사용에 있어서는 유니코드를 사용한다 단 본래의 유니코드가 아닌, 한글 폰트의 종류에 따라 각 폰트마다 0000~FFFF사이의 임의의 유니코드 값을 할당한 뒤 그를 통해 폰트 이미지의 라벨을 관리해서 CNN의 분류 기준으로 삼았다.



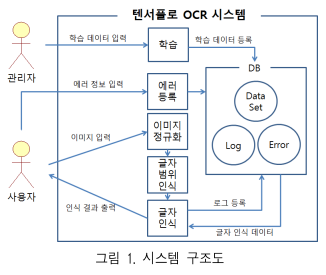
설명.

각 폰트 TTF 파일을 보관하는 Font files, 각 폰트의 한글 문자별 이미지를 보관하는 Font image files, 그리고 폰트 이름과 그 폰트에 할당된 16진법과 2진법의 유니코드 값을 맵핑한 테이블, label 및 이미지 매트릭스 파일을 보관하는 Font image/label files의

데이터베이스를 갖는다. 알고리즘에 실제로 이용할 데이터는 Font image/label files의 데이터이다.

1. 텐서플로를 이용한 OCR 시스템 개발

2.1 시스템 구조도.

해당 그림1에서는 서버는 사용자가 입력한 한글 영상의 정규화 과정을 거친 이후 해당 문장을 한 글자 단위로 자른 후 각 글자를 인식한다. 인식이 잘못되면 DB에 오류 내용을 등록할 수 있다.

2.2 Adaptive Threshold

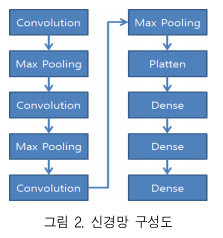
미리 정해둔 임계값에 대하여 전체 영상을 이진화하는 것이 아닌, 픽셀 주변 영역의 밝기 평균에 일정한 상수를 빼서 결정된 값을 임계값으로 사용하는 방법이다.

2.3 컨볼루션 신경망

전체 영상은 바로 글자로 인식할 수 없기 때문에 영상 특징 추출, Max Pooling이 필요하다. 영상 특징 추출은 영상에 Filter를 적용하여 Feature를 형성하는 과정이다. Max Pooling은 Feature에 혼재한 Noise의 영향력을 줄이기 위해 Feature의 최댓값을

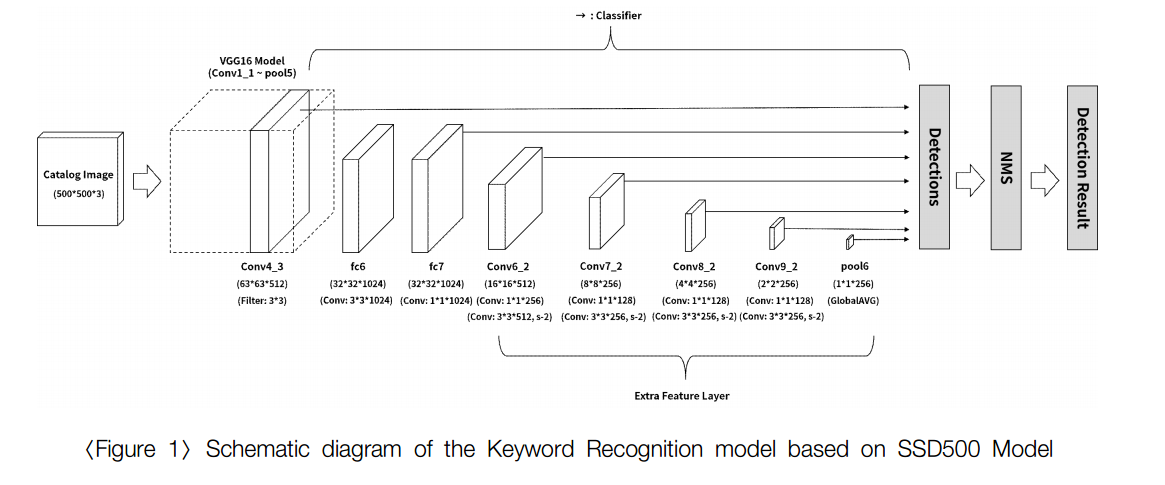
추출하여 보다 작은 규모의 Feature로 축소하는 과정이다[1]. Filter의 임계값은 Convolution 네트워크 학습에서 자동으로 최적화된다.

2.4 학습 모델 구성.

학습 모델은 총 10개의 층으로 구성한다. 시스템에 입력한 영상은 정규화 과정을 거친 후 모델로 입력한다. 모델 입력 영상은 그림 2에 해당하는 층을 거쳐 출력 값 중 하나를 선정하게 된다. 각 층은 ReLU 활성화 함수를 사용하고 출력 값을 선택하는 최종 Desne 레이어만 Softmax 활성화 함수를 사용한다. 출력 값은 KS1001에서 사용하는 한글 2350자와 숫자 10개를 합친 2360자 중 1글자로 한다. 모델 구성은 KERAS을 사용하였다.

1. 온라인 쇼핑몰에서 상품 설명 이미지 내의 키워드 인식을 위한 딥러닝 훈련 데이터 자동 생성 방안.

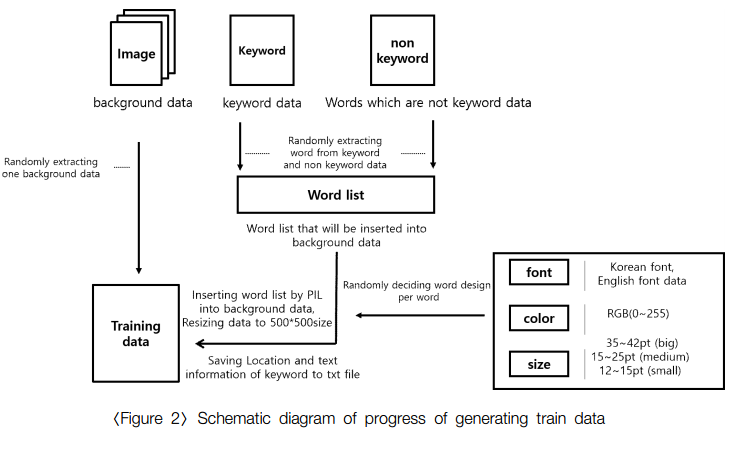
이 논문에서는 딥러닝 훈련 부분만 요약.



요약 : 시작은 미시적인 특징을, 끝은 이미지 천체를 인지하고 학습하게 만들었다. 검출은 7개의 층에서 구한 특징 지도들의 특징들을 학습하는 합성곱 층과 미리 산정한 다수의 Default Box들의 정보들로 이루어져 있는데, 이를 이용하여 이미지 내에서 키워드가 들어있을 영역들의 후보를 구한다. 사물 대비 글자가 보통은 작은 사이즈이기 때문에 Default Box의 영역은 작게 만들었다. 검출 영역에서 나온 결과물은 보통 한 키워드 주변에 중복되어 많은 Default Box들이 나타나게 되는데, NMS 부분에서는 이 중복되어 겹치는 박스들을 확률이 가장 높은 하나의 박스로 통합해 주는 역할을 담당하고 있다.(위 실험은 45% 기준) 마지막으로 출력 부분에서는 NMS 부분에 의해 정리된 값들을 받아, 키워드의 종류와 확률, 상대 좌표들로 구성되어 있는 결과값들을 키워드와 실제 이미지 상의 박스 좌표 값들을 산정하는 역할을 담당한다.

<전문>

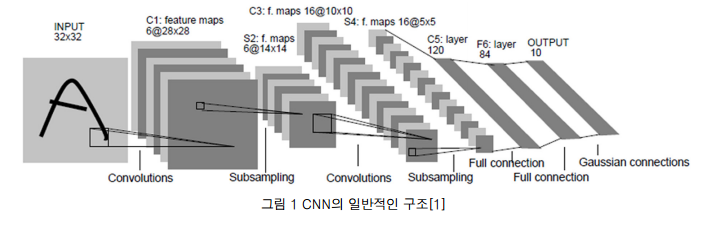
훈련 모델의 프로토타입에 대한 자세한 설명은 다음과 같다. 우선 입력 부분의 경우 일반적인 SSD 모형이 인식하는 사물들에 비해 본 키워드 인식기가 인지할 키워드들이 대부분 상대적으로 크기가 작다는 것을 고려하여, 상품 카탈로그 이미지를 입력할 때 500\*500 픽셀로 변환한 이미지를 입력 받도록 설계하였다. 특징 추출 부분에서는 입력 받은 이미지들이 다수의 합성곱 층과 최대 풀링(Max Pooling) 층을 통과하면서 각 키워드들의 특징들을 차례대로 학습하게 구성되어 있다. 즉, 특징 추출 부분의 제일 앞 단에 있는 ‘Conv1\_1’ 층에서는 키워드가 가지고 있는 직선이나 직각, 곡선과 같은 미시적인 특징들을 학습하며, 그보다 뒤에 있는 ‘Conv4\_3’ 층에서는 앞 층에서 학습한 특징들을 기반으로 일반적인 키워드들의 전반적인 특징들을 학습하게 된다. 그리고 그보다 뒤에 있는 ‘Conv6\_2’, ‘Conv7\_2’, ‘Conv8\_2’, ‘Conv9\_2’ 등의 층에서는 이전 층들보다 크기가 큰 텍스트들의 특징들을 학습하도록 구성하였다. 그리고 마지막의 ‘pool6’ 층의 경우 이미지 전체를 인지하도록 설계된 층으로, 합성곱 층 대신 특징 지도(Feature Map)들의 값들의 평균을 취하는 ‘Global Average Pooling’ 층으로 구성되어 있다. 기존 SSD모형에서는 이 특징 추출 부분 중 VGG16 모형 부분에 해당하는 검출 부분을 구성할 때 이미 학습된 가중치를 불러온 뒤 훈련이 되지 않도록 설정하였지만, 본 연구에서는 사물과 글자의 특징이 다른 만큼 훈련이 이루어지도록 변경함과 동시에 학습률을 약 3배 정도 높은 0.001로 설정하였다. 다음으로 검출 부분에서는 앞서 설명한 ‘Conv4\_3’, ‘fc7’, ‘Conv6\_2’, ‘Conv7\_2’, ‘Conv8\_2’,‘Conv9\_2’, ‘pool6’ 등 총 7개의 층에서 구한 특징 지도들의 특징들을 학습하는 합성곱 층과 미리 산정한 다수의 Default Box들의 정보들로 구성되어 있다. 이 부분을 통해 여러 위치에 존재하는 Default Box들이 어떠한 키워드인지, 얼마나 큰 텍스트인지 등을 산출하여 이미지 상에 키워드가 있을 영역들의 후보를 구하게 된다. 본 연구에서는 50% 이상의 확률을 가지는 영역들만을 검출하도록 설정하였다. 또한 사물 대비 텍스트들의 크기가 보통 작다는 것을 고려하여 Default Box들의 크기들을 기존 연구들보다 작게 설정하였다. 검출 영역에서 나온 결과물은 보통 한 키워드 주변에 중복되어 많은 Default Box들이 나타나게 되는데, NMS 부분에서는 이 중복되어 겹치는 박스들을 확률이 가장 높은 하나의 박스로 통합해 주는 역할을 담당하고 있다. 본 연구에서는 기연구된 모형과 같이 45% 이상 겹치는 부분이 존재하면 이를 통합하도록 하였다. 마지막으로 출력 부분에서는 NMS 부분에 의해 정리된 값들을 받아, 키워드의 종류와 확률, 상대 좌표들로 구성되어 있는 결과값들을 키워드와 실제 이미지 상의 박스 좌표 값들을 산정하는 역할을 담당한다. 본 연구에서 사용된 키워드 인식기의 프로토타입에서는 널리 알려져 있는 SSD 모형과 유사한 설정들을 활용하였으나, 특징 추출 부분의 VGG16 모형을 다른 모형으로 변경하거나 검출 부분의 Default Box 및 필터 사이즈 등을 텍스트의 특징에 맞게 세부 조절함으로써 키워드 인식 성능을 향상시킬 수 있을 것이다.



훈련용 데이터 자동 생성기: 대다수 상품 카탈로그 이미지는 컴퓨터 환경 하에서 제작되기에 학습데이터를 프로그램으로 생성해도 유의적인 차이가 없다는 점에서 착안한 생성기.

자동 생성기는 Python 3.5 버전으로 개발되었으며 주요 사용 라이브러리는 ‘PIL(Python Image Library)’을 기반으로 한 ‘Pillow’와 내장 모듈인 ‘Random’ 등이 있다.

1. 다양한 딥러닝 알고리즘과 활용.
2. CNN(Convolutional Neural Network)



이미지 데이터 처리에 적합한 구조를 가진 신경망. 주로 필기/숫자인식, 얼굴 인식 등의 제한적인 분야에서 사용.

CNN은 그림 1과 같이 크게 convolution-pooling layer와 fully connected layer의 두 부분으로 구성되어 있다. 전자는 입력된 이미지로부터 계층적 구조의 feature를 추출하는 역할을, 후자는 추출된 feature를 입력 받아 타겟 클래스로 분류하는 역할을 담당한다. CNN layer는 아래와 같은 두 가지 특징을 통해 이미지 데이터의 특성을 반영하는 동시에 모델의 복잡

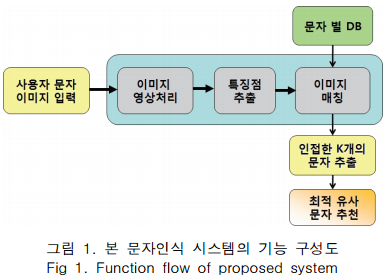
도를 크게 단순화시킨다.

-Local connectivity: 가장 일반적인 형태인 fully connected layer와 달리, 해당 convolution 필터의 크기인 NxN window 내의 인접한 뉴런들에만 연결이 되어있다. 이는 인접한 픽셀들끼리는 상관관계가 높지만 멀리 떨어진 픽셀들은 그렇지 않은 이미지의 특성(locality)을 반영한 것이다.

-Shared weights: convolution 필터들은 적용되는 위치가 달라도 같은 weight값을 공유한다. 이는 픽셀값의 통계적 특성이 이미지 상의 좌표와 무관하다는 이미지의 특성(stationarity)을 반영한 것이다

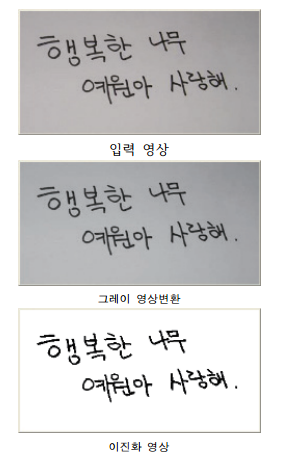
- Pooling layer는 NxN 윈도 내의 입력값들을 그 최대값 또는 평균값으로 매핑함으로써 이미지의 크기를 점차적으로 줄인다. 그 결과 상위 layer로 올라갈수록 같은 크기의 convolution 필터가 상대적으로 넓은 영역을 처리하게 되어, 상위 layer에서는 하위 layer의 저차원 feature를 조합한 고차원 feature를 학습하는 효과를 가져온다.

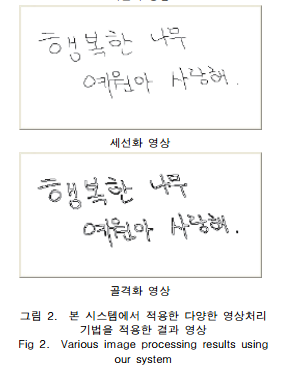
이하 내용은 멀티모달 학습에 관한 내용이라 생략.

1. 영상처리 기법을 이용한 문자인식 시스템 개발

입력된 이미지를 가지고 정보를 추출하기 위해 본 논문에서는 흑백 영상변환, 이진화, 세션화. 골격화 알고리즘을 적용.

입력 영상을 흑백 영상으로 변환할 때는 RGB 색상 공간상에서 각 화소의 RGB값의 평균값을 구하여 사용함. 이후의 모든 기법들은 흑백 영상을 기반으로 알고리즘 적용.

1. 입력 영상 변환.

그림1

좌측의 그림은 흑백 영상변환, 이진화, 세션화, 골격화 순으로 영상처리를 하는 과정을 보여준다.

이와 같은 과정을 거친 이미지를 기반으로 다음 과정을 진행한다.

1. 특징 점 추출.

위의 영상처리 기법을 적용한 결과들을 바탕으로 영상의 특징점을 추출한다. 다만 입력된 이미지의 크기와 위치가 저마다 다르므로 이를 정규화 시키는 작업이 요구된다.

과정

<1> 입력 받은 이미지 중에서 글자가 있는 영역을 찾아낸다.

<2> 찾아낸 영역을 균일한 크기로 조정한다.

<3> 문자를 인식하는 것을 방해나는 노이즈를 제거한다.

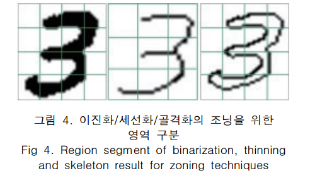
<3-1> 본 논문에서는 노이즈의 기준을 체인코드를 사용하여, 체인코드가 일정 길이 미만일 경우 노이즈로 간주하였다.

<4> 정규화 된 영상을 기반으로 특징 점을 추출한다. 여기서는 영역 구분을 통해 각 영역들의 정보를 분석하는 조닝(zoning) 기법과 전체 영상의 히스토그램 투영, 가로 세로의 비율을 사용했다.

<5> 1)의 과정을 거친 영상을 가로/세로 각 4등분해서 총 16개의 공간으로 만든다. 전체 이미지의 검은 화소의 수를 대상으로 각 영역의 검은 화소수를 비율로 나타낸다. 전체 영역의 비율을 다 더하면 1이다.

<6> 이진화 조닝(세션화 조닝도 동일)을 적용하므로 16개의 특징점이 추출된다. 단 세선화 조닝(3\*3등분)도 이진화 조닝과 비슷한 모양으로 변하기 때문에 같이 적용한다. 골격화 조닝 역시 영역별 비율을 사용한다.

<7> 추출된 특징점은 각각 4개의 방향값들을 가지게 되는데, 4등분 한 조닝에서는 총 64개의 특징점을 갖게 된다.

그림 2

각각 이진화 세선화 골격화의 조닝 예시.

<8> 가로 세로의 비율 : 문자마다 가로 세로의 비율이 다른데, 이것은 문자의 고유한 특성으로 볼 수 있다. 따라서 이 또한 하나의 기준점이 된다.

<9> 히스토그램: 정규화된 영상을 기반으로 가로/세로/소지한 화소의 수를 따져본다. 그림 2 같은 경우에는 왼편과 오른편에는 화소가 존재하지 않다가 글자가 존재하는 부분에는 많은 화소가 분포하는 것이 보이며, 세로로는 길게 분포하는 것을 보인다, 본 논문에서는 이를 이용, 60\*60 화소의 각 화소의 열과 행을 하나의 피처로 사용해 120개의 특징점을 추출했다.

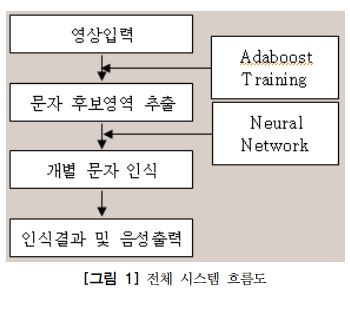
<10> 종합적으로 보자면, 해당 논문의 시스템에서는 이진화 조닝(16) + 세선화 조닝(9) + 가로/세로의 길이비(1) + 컨투어 조닝(64)+히스토그램 프로잭션(120)개로 총합 210개를 사용했다.

1. 문자 DB 데이터 구축.

위의 특징점 정보를 각 문자별 다양한 샘플 영상에 적용하여 특징점 DB 테이터를 구축한다.

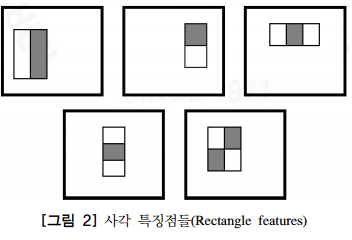
1. 문자 DB로부터 최적 문자 매칭 및 추천.

해당 DB를 완성 시키면 입력 받은 영상과의 비교를 통해 학습하고 관리한다. 이때의 알고리즘을 본 논문에서는 K-최 근접 이웃 알고리즘(K-NN)을 사용했다. (이때의 K는 5를 사용) 이는 특징점을 분석한 결과와 유사한 후보군 5개를 추천해서 그 중 가장 추천을 많이 받은 것을 정답으로 고른다.

1. 모바일 기반의 문자인식 시스템.

카메라로부터 들어오는 입력 이미지에서 텍스트 후보 영역을 정확하게 검출하기 위해서 제안하는 문자인식 시스템에서는 간단한 형태를 가지는 특징점들의 값을 바탕으로 텍스트 유무를 판단한다. 이미지의 픽셀 값을 직접 이용하는 픽셀 기반의 시스템 보다 특징점 기반의 시스템이 알고리즘의 수행 속도면에서 훨씬 빠르다는 장점이 있다.

본 논문에서는 다섯 종류의 특징점을 사용한다.

특징점.

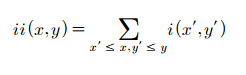
두개의 사각형 영역내의 밝기 값들의 합의

차이이다. 각각의 영역은 동일한 크기와 모양을 가진다.

세개의 사각형으로 구성된 특징점의 값은 바깥쪽 두 개의 사각형 영역내의 밝기 값의 합에서 가운데 사각형 영역의 밝기 값의 합을 뺀 값이다. 끝으로 네개의 사각형으로 구성된 특징점의 값은 대각선 방향의 두 쌍의 사각형 영역의 밝기 값의 합의 차이이다. 간단히 말해서 그림 2에서 보여지는 각각의 특징점의 값은 흰색 영역의 밝기 값의 합에서 회색 영역의 밝기 값의 합을 뺀 값을 의미한다. 제안하는 문자인식 시스템에서는 그림 2에 보이는 다섯 가지의 특징점을 이용하여 텍스트 후보영역을 검출하고자 한다.

적분영상.

앞서 제시한 특징점의 값은 적분 영상이라 불리는 이미지 표현 방식으로 계산하는 것이 효율적이다.

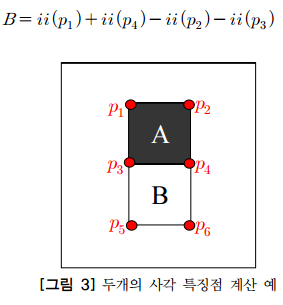
이미지의 위치 (x,y)에서의 적분 영상의 값은 좌측의 식과 같이 좌표의 왼쪽과 위쪽에 위치한 픽셀들의 밝기 값의 합을 나타낸다.

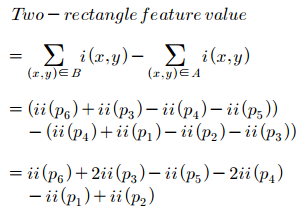
여기서 는 적분 영상을 나타낸다. 그리고 i(x,y)는 원래의 입력 이미지를 뜻한다.

위의 식은 순환식을 사용하여 구하면 쉽다.

여기서 s(x,y)는 열 방향의 픽셀 값들의 합이고 s(x,-1)=0, ii(-1,y)=0이다.

적분 영상을 이용하면 사각 특징점들의 픽셀 값을 빠르게 구한다.





인접한 두 개의 사각형 각각의 합 사이의 차이 값은 적분 영상 상의 6개의 점을 참조함으로써 좌측의 식과 같이 간단하게 구할 수 있다.

결과적으로 말하면 앞 절에서 제시한 두개의 사각형으로 구성된 특징점 값을 구하기위해서는 적분 영상 상의 여섯개의 점을 참조함으로써 구할 수 있고 3개의 사각형으로 구성된 특징점 값은 여덟 개의 점을 참조하면 구할 수 있다. 또한 4개의 사각형으로 구성된 특징점 값은 아홉 개의 점을 참조함으로써 쉽고 빠르게 구할 수 있다.

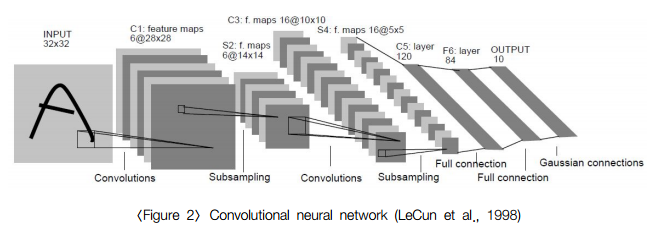
1. 딥러닝의 모형과 응용사례

다양한 예시 중에서 CNN 부분과 역전파 알고리즘만 정리.

1. 오류 역전파 알고리즘

논문 참조.

1. 합성곱 신경망(CNN, convolutional neural network)은 생물의 시각처리과정을 모방하여 패턴의 크기나 위치가 바뀌어도 인식을 할 수 있다는 장점이 있는 모형으로서 그 뿌리는 Neocognitron(Fukushima, 1980)이라는 것이다. 이 모형이 발전되어서 LeCun et al.(1998)을 통하여 현재 사용되고 있는 합성곱 신경망이 널리 쓰이는 계기가 되었다. LeCun et al.(1998)에 의하면 합성곱 신경망은 세가지 특징을 가지고 있는데, 그것은 local receptive field, shared weight, 그리고 sub-sampling이다.

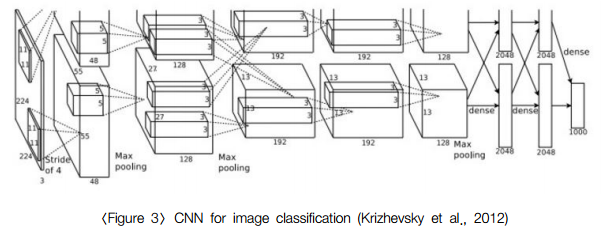


local receptive field는 <Figure 1>에서와 같은 기존의 모형이 하위계층의 각 노드가 모든 상위계층의 노드와 연결되어 있는 것과 달리, 상위계층의 일부의 노드에만 가중치로 연결되는 것을 의미한다. <Figure 2>에서 가장 왼쪽에 있는 것이 입력패턴이며, 입력패턴의 오른쪽 하단에 있는 작은 사각형이 local receptive field가 되며 이것이 우측에 있는 다음계층(C1)의 한 개의 노드와 가중치(필터라고도 부른다. 필터의 크기가 5Х5라면 편향을 더하여 총 26개의 가중치가 있다)로 연결되어 있는 것을 볼 수 있다. 이러한 필터를 입력패턴의 모든 영역(왼쪽상단에서 아래쪽 하단으로)에 적용하여(가중치를 해당 영역에 곱하여 합하는 연산을 함) C1에 있는 패널의 노드로 전달하는 것을 합성곱(convolution)이라고 한다. 한편, shared weight는 가중치를 공유한다는 뜻이다. <Figure 2>의 C1에는 여러 개의 패널이 있는 것을 볼 수 있는데, 한 개의 패널에 있는 노드와 입력패턴(혹은 하위계층)에 적용되는 필터는 동일하다는 의미이다. <Figure 2>의 C1에는 6개의 패널이 있으므로 이 경우에는 전부 6개의 서로 다른 필터가 존재한다. 한 개의 필터는 한 개의 특징(혹은 작은 패턴)을 인식할 수 있으므로 C1에 있는 한 개의 패널은 입력패턴의 특정 위치에 존재하는 해당 특징을 발견할 수 있게 된다. sub-sampling은 <Figure 2>의 왼쪽에서 두 번째 계층(C1)과 세 번째 계층(S2)에서 일어나는 과정으로서 C1에 있는 패널을 압축하는 것이며 pooling이라고도 부른다. 예를 들면 C1에 있는 2Х2영역을 S2의 하나의 노드로 만드는 것으로서, 축소하는 방법은 다양하다. LeCun et al.(1998)에서는 2Х2영역의 값을 평균하여 가중치를 곱한 뒤 S자 모양의 활성화 함수를 적용하는 방법 average pooling을 사용하였으나 최근에는 2Х2영역의 네 값 중에서 가장 큰 값을 사용하는 max-pooling 방법을 많이 사용한다. sub-sampling은 입력패턴의 크기를 줄여주는 기능과 발견한 특징에 대한 위치의 변동에 덜 민감하게 하는 기능을 한다.

이와 같은 convolution과 sub-sampling과정은 응용사례에 따라서 여러 번 반복되며 <Figure 2>에서는 2번이 반복되는 것을 볼 수 있다. sub-sampling이 종료된 후에는 <Figure 1>과 같은 형태의 신경망이 연결된다. <Figure 2>에서는 우측에 두 개의 내부계층(C5와 F6)과 출력계층이 있는 것을 볼 수 있다. 출력계층에는 10개의 노드가 있는데 이는 숫자(0~9)를 인식하는 모형이기 때문이다.

합성곱 신경망의 가중치는 어떤 방법으로 학습시키는가? 합성곱 신경망은 내부계층이 매우 많으므로 따라서 노드의 수가 많아지고 노드를 연결하는 가중치도 많아지게 된다. 일반적으로 모형에서 가중치의 수가 많아지면 앞에서 설명한 오류 역전파 알고리즘을 적용할 때 일차편도함수를 반복적으로 계산하는 과정에서 그 크기가 매우 커지거나 반대로 0이 되는 문제가 발생하게 된다. 하지만 합성곱 신경망의 경우에는 shared weight라는 특징이 있어서 가중치의 수가 그런 문제가 발생할만큼 많지 않아서 오류 역전파 알고리즘을 무리 없이 적용할 수 있다. 다만 shared weight란 특징과 sub-sampling이란 과정이 일반적인 신경망 모형과 다르기 때문에 알고리즘을 조금 수정할 필요가 있다. 우선 shared weight에 대해서는 가중치가 서로 다른 경우와 동일한 방법으로 일차편도함수를 계산한 후에 동일한 가중치에 대하여는 일차편도함수의 값을 모두 더해주는 과정을 적용하면 된다. sub-sampling을 통하여 패턴이 압축되는 경우에는 오류를 역전파 할 때 반대방향의 연산을 해주어야 하므로 up-sampling을 통하여 확장해주는 과정이 필요하다. 그런데 sub-sampling의 방식이 응용사례에 따라 다르므로 경우에 따라 다르게 적용할 필요가 있다. 예를 들어 LeCun et al. (1998)에서 적용된 sub-sampling에서는 가중치와 활성화 함수를 사용하므로 일반적인 오류역전파 알고리즘을 적용하여 가중치의 일차편도함수를 계산하고 그때 계산한 를 up-sampling을 통하여

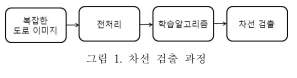
아래로 전달하면 된다. 가중치와 활성화 함수가 없이 max-pooling을 했다면 상위노드에서 계산한 를 max-pooling단계에서 가장 큰 값이었던 노드로 전달하여 하위노드의 를 계산한다.이러한 합성곱신경망은 문자인식과 다양한 형태의 영상인식 등에 성공적으로 적용되어 왔다. 특히 음성인식, 문서읽기, 그리고 필기체문자인식 등의 분야에서는 1990년대 초반부터 적용되어 왔으며, LeCun et al.(1998)의 모형은 MNIST (H1)에 대하여 매우 낮은 오류율을 보인바 있다. 그러나 합성곱신경망이 본격적으로 학계의 주요 관심사가 된 것은 2012년에 발표된 영상인식모형(Krizhevsky et al., 2012)이 놀라운 성과를 거두면서부터이다

.

이 모형은 백만 개 정도의 컬러영상을 천 개의 클래스로 분류하는 일을 하는 것인데, 전체적인 모형은 <Figure 3>과 같다. <Figure 3>에서 보듯이 이 모형은 두 개의 GPU를 통하여 구현되었기 때문에 입력계층과 출력계층을 제외한 내부계층은 두 부분으로 구분되어 구성되어 있다. 컬러영상을 인식해야 하므로 입력패널은 3개이고, <Figure 3>에서 입력계층 오른쪽에 있는 첫 번째 내부계층은 convolution과 max-pooling을 수행하는 것으로 총 96개의 패널로 이루어져 있다.

그 오른쪽의 두 번째 내부계층은 마찬가지로 convolution과 max-pooling을 수행하며 총 256개의 패널로 구성된다. 그 다음에 있는 두 개의 내부계층은 max-pooling이 없는 convolution을 하고 다섯 번째 내부계층은 convolution과 max-pooling을 수행하는 256개의 패널로 구성된다. 여섯 번째와 일곱 번째 내부계층은 하위계층의 노드와 전부 연결되어 있는 구조이고 마지막 출력계층은 1000개의 노드로 구성되어 있다. 이 모형은 대략 6천만개의 가충치가 있으며 학습시간으로 GTX 580 3GB GPU를 사용하여 6~7일 정도가 소요되었다. 이 모형은 활성화 함수로 ReLU (max(0,x))를 사용하여 학습시간을 단축시키고, dropout이라는 방식으로 과적합을 방지하고, 또 <Figure 4> Recurrent neural network학습영상을 변형하여 입력패턴의 수를 늘리는 방법으로 모형의 인식률을 높였다. 그 결과 당시의 다른 모형의 인식 오류율의 절반에 가까운 오류율을 달성하는 성과를 얻었다. 그러한 이유로 현재의 많은 합성곱신경망 모형은 10~20개의 ReLU 계층과 수십억 개의 노드간 연결이 있고 수억 개의 가중치를 학습시킨다. 물론 학습알고리즘의 발전과 GPU기술의 발전으로 학습속도는 수시간단위로 단축이 되었다.

1. 합성곱 신경망과 극한 학습기계의 융합기반의 딥러닝 신경망을 이용한 차선 검출 성능 향상



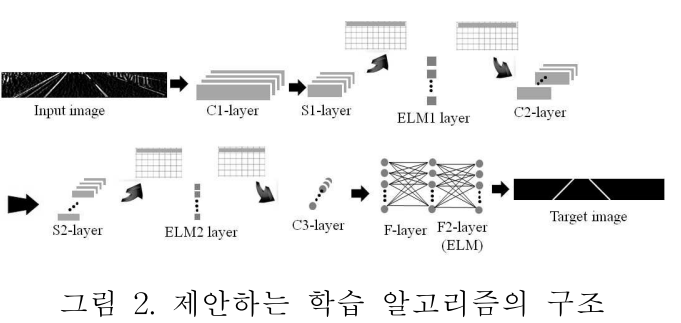
1. 전처리.

이 부분에서는 기본적으로 차선을 선명하게 만드는 작업을 한다. 노이즈 제거를 위한 가우시안 필터링과 색이 다른 차선을 위한 윤곽선 검출을 사용했다.

1. 합성곱 신경망(CNN)과 극한 학습기계의 융합기반의 딥러닝 신경망

CNN은 현재 가장 각광 받는 딥러닝 알고리즘이다. 이 알고리즘은 인간이 사물을 볼 때의 일어나는 일을 알고리즘화 시킨 것이다.

극한 학습기계의 경우 기존의 신경망에서 역전파 방식을 통한 에러 조절 대신에 고안된 학습 방법이다. 의사역행렬을 사용하여 최적의 출력 가중치를 한 번의 연산으로 계산하는 방식으로, 학습속도가 매우 빠르다.

CNN은 속도가 느리다는 단점이 있지만 극한 학습기계를 이용하여 새로운 알고리즘을 짜는 것을 제안한다. 

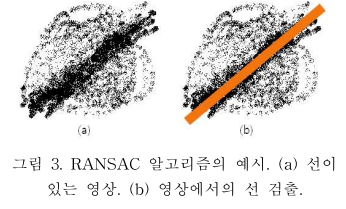
기존의 CNN에서 극학 학습기계인 층을 중간중간에 넣었다. . C층과 S층의 경우 기존의 역전파 방식을 그대로 사용해서 학습한다. C와 S사이 ELM층은 다음 수식을 이용하여 목표 값을 구해 계산한다



은 이미지가 들어갔을 때 저장된 C층의 값이며, 은 역전파 방식에서 구한 에러, a는 0~1사이 파라미터이다. 마지막 F층은 차선만 있는 목표이미지를 직접 만들어서 사용한다. 그렇게 F층 ELM을 통해 결과를 얻게 된다.

1. 차선 검출

차선 검출에서는 RANSAC[6]을 사용해서 결과를 얻었다. RANSAC 알고리즘은 어떤 현상에서 노이즈나 예측을 방해하는 데이터(outliner)가 있을 경우 예상되는 최적의 모델을 선택하는 절차적 방법이다. 그림 3의 경우 RANSAC 알고리즘으로 선을 검출할 시 나오는 결과를 나타내고 있다.



1. 출처 dbpia [↑](#footnote-ref-1)
2. CNN(Convolution Neural Network)은 자연어 처리, 이미지 인식 등 다양한 분야에서 혁신적 성과를 얻고 있는 딥러닝의 한 종류 [↑](#footnote-ref-2)