딥러닝을 이용한 일반 영상에서의 문자 인식

정규환 · 김현준 · 이예하 (VUNO Inc.)

목차

1. 서 론

인류가 디지털 시대로 진입하면서 기존의 신문, 책, 서류 등의 인쇄물을 디지털화하여 보관 기간을 영구화 하거나 검색과 분류, 편집이 가능하도록 하고, 시각 장애인에게 시각적 정보에 접근이 가능하게 하는 문자 인식 기술은 지난 수십 년간 수많은 연구자들에 의해 개발이 진행되어 왔다[1-4]. 특히 스캐너에 의해 촬영된 정형화된 문서 내 문자를 인식하는 OCR(Optical Character Recognition)[1,2]기술은 다양한 상용 및 오픈 소스 소프트웨어로 개발되어 인간의 인식 능력에 준하는 매우 높은 인식 정확도를 보이고 있다[3].

한편, 모바일 기기에 내장된 카메라 해상도 증가와 및 통신 속도의 향상, 클라우드 저장 시스템의 발전에 따라 다양한 환경에서 촬영된 일반 영상의 수가 기하급수적으로 증가하고 있고, 이러한 영상을 이용한 다양한 응용서비스에 개발의 중요성이 증대되고 있다. 특히, 일반 영상으로부터 활자 정보들을 실시간으로 정확하게 인식할 수 있게 된다면, 자동 주행 시스템, 시각 장애인 지원 시스템, 전시관 내 정보 제공 시스템, 외국인을 위한 번역 안내 시스템 등 다양한 영역에 활용이 가능하므로 상업적, 공공적 가치가 큰 기술로 주목받고 있다. 하지만 기존의 OCR 방법들이 다루는 정형적인 문서가 아닌 제한이 없는 일반적인 영상 내의 문자에는 배경, 질감, 서식, 조명 조건 등 다양한 변화들이 존재하기 때문에 영상 내 문자들의 위치를 탐색하고 이를 정확하게 인식하는 것은 높은 복잡도와 난이도를 가진 도전적인 주제이다. 이를 위해 컴퓨터 비전분야에서 널리 쓰여 온 다양한 수동적인(Hand-crafted) 특징 추출 방법을 조합하거나, 각각의 영상 촬영 환경에 특화된 특징을 선택하는 방법들이 제안되어 왔으나, 영상별로 문자 인식 성능 편차가 크고, 복잡하고 내재된 문자의 특징을 추출해 내지 못해 인식률이 떨

어지는 등의 한계를 나타내었다[4].

이러한 가운데, 최근 깊은 구조의 인공신경망을 통해 주어진 다량의 데이터로부터 영상 내 문자의 특징을 자동적이고 계층적으로 학습하는 방법이 그 대안으로 주목받고 있다. 딥러닝(Deep Learning)이라 불리는 이러한 깊은 구조의 인공신경망은, 출력값 없이도 입력 데이터의 비선형적 변환을 반복하면서 하위 층의 단순한 특징들로부터 상위 층의 보다 복잡하고 구조적인 형태의 특징들까지를 학습해 내는 비지도학습(Unsupervised Learning)이 가능하다. 이렇게 학습된 특징을 기존의 다양한 기계학습 분류 모델의 입력값으로 사용할 때, 인식 성능이 기존의 방법에 비해 대폭 향상됨이 다양한 분야에서 확인되고 있다[5-8].

따라서 본 논문에서는 딥러닝 기반의 특징 학습 방법에 대해 설명하고, 이를 일반 영상 내 문자 인식에 적용한 다양한 연구 사례를 살펴보고자 한다.

2. 연구 배경

영상 내에서 문자인식 기술은 다시 영상 내 문자 검출, 문자 인식의 세부 단계로 나누어지며, 각각의 기능은 다음과 같다.

2.1 영상 내 문자 검출

영상 내 문자 탐지의 목표는 주어진 영상 내에 문자가 존재하는지를 판별하고 문자가 존재한다고 판단될 경우, 문자의 후보 위치를 문자를 포함하는 사각형 형태로 포착해내는 것이다. 기존의 문자 탐지 방법은 사용되는 화소의 특징에 따라 크게 영역기반(Region-based) 방법과 질감기반(Texture-based) 방법이 있는데, 영역 기반 방법은 경우 문자들이 배경과 다르게 가지는 색상 특성이 있다는 가정을 바탕으로 같은 색상 특성을 가진 영역을 점차적으로 연결함으로써 문자를 찾는 연결 요소 (Connected Component) 방법과, 문자와 배경의 경계에 높은 대비(Contrast)가 있다는 가정을 바탕으로 문자의 모서리를 찾아내는 모서리 기반(Edge-based) 방법으로 나누어진다[1]. 질감 기반 방법은 문자가 배경과 구분되는 질감적인 특성을 가진다는 가정을 바탕으로 하고 있으며 전통적인 컴퓨터 비전 분야에서 영상 분석에 사용되는 가버 필터(Gabor Filter), SIFT (Scale-invariant Feature Transform), HoG (Histogram of Oriented Gradient), FFT(Fast Fourier Transfrom), SWT(Stroke Width Transform), 공간적 변화도(Spatial Variance)등의 특성을 활용한다[8]. 하지만 이런 전통적인 문자

(Fig. 1) Examples of natural scene text detection

탐지 방법은 각각 문자의 속성 및 환경에 대한 특정한 가정을 바탕으로 하고 있으며, 따라서 이러한 가정을 만족하지 않는 다양한 일반 영상에서는 탐지 성능이 떨어지는 한계를 가지고 있다.

2.2 영상 내 문자 인식

영상 내에서 문자의 위치가 탐지가 되면 그 결과로 문자가 존재하는 것으로 판단되는 사각형 형태의 경계가 주어진다. 가장 단순한 방법은 문자가 포함된 사각형 경계를 기존의 OCR 모듈에 그대로 입력하여 그 결과를 활용하는 방법이지만 그 인식 성능이 낮아 활용도가 떨어진다[9]. 따라서 사각형 내에서 개별 문자의 영역을 추출하고 이를 개별 문자로 인식한 후, 이를 연결하여 단어로 인식해내는 다양한 기술이 개발되어 왔는데 대부분은 단어 내의 문자들의 간격의 일정하다는 특성이나 문자들이 일정 선상에 위치하는 특성, 혹은 문자들의 크기가 균일하다는 특성 등의 몇 가지 가정을 바탕으로 하고 있다. 일반적으로 주어진 사각형 내에서 작은 창을 가장 왼쪽에서부터 가장 오른쪽으로 이동하며 해당 창 내에 존재하는 내용을 문자 분류기에 입력으로 사용하고 해당 분류기의 결과를 조합하여 최종 문자와 단어를 인식하는 이동 창(Sliding Window) 방법이 사용된다. 이때 주로 사용되는 문자 분류기는 영상에서 추출한 문자 영역과 실제 정답에 해당되는 문자의 쌍으로 구성된 데이터를 이용하여 학습 시킨 모델로, SVM(Support Vector Machine), 인공신경망(Artificial Neural Network), 로지스틱 회귀분석(Logistic Regression)등이 있다[10].

최근에는 입력 영상으로부터 최종 인식 결과까지를 하나의 시스템으로 통합하는 종단간(End-to-end) 문자 인식 시스템에 대한 연구가 활발한데, 이는 문자 탐색 모델과 문자 인식 모델을 동시에 학습시키거나 서로 상호작용하는 형태로 구성하여 정확도와 효율성을 향상 시킬 수 있는 것으로 나타나고 있다[11-13].

3. 딥러닝을 이용한 특징 학습 방법

3.1 딥러닝 방법의 등장 배경

인간 두뇌의 뉴런(Neuron)간의 연결 구조를 모방하여 만든 인공신경망 모델은 1980년대 다층 신경망(Multi-layer Perceptron)의 학습 방법인 오류 역전파(Back Propagation)알고리즘이 개발되어 복잡한 분류 문제들을 높은 성능으로 성공적으로 수행할 수 있음이 밝혀지며 주목을 받았다. 하지만 모델의 크기가 커질수록 학습 속도가 느려지고 학습 결과가 국소 최적해(Local Optima)에 빠지는 단점이 있어 활용 범위가 제한되어왔다[14].

한편, SVM이나 로지스틱 회귀분석과 같이 입력 데이터가 최종 분류값으로 변환되는 과정이 짧은 얕은 구조(Shallow Architecture)와 달리, 인공 신경망의 은닉층의 수가 두 개 이상인 딥러닝 모델들은 여러 층의 비선형 변환과정을 거치는 깊은 구조(Deep Architecture)를 가짐으로써 동일한 수의 연결 가중치 값으로 보다 복잡하고 표현력 높은 모델을 구축할 수 있다[5]. 반면, 딥러닝 모델은 다층 구조로 인해 기존의 역전파 알고리즘으로 효율적인 학습이 불가능하고 네트워크를 학습하기 위한 계산량이 많은 한계로 인해 역사가 오래되었음에도 불구하고 활용범위가 제한적이었다. 하지만 2000년대 중반 제안된 제한적 볼츠만 머신(Restricted Boltzmann Machine)을 활용한 사전 학습(Pre-training)을 통한 초기해 탐색 방법[15], 확률적 앙상블 기법인 드롭아웃(Dropout)[16]과 같은 과적합 방지 방법 등의 방법론적 진보와

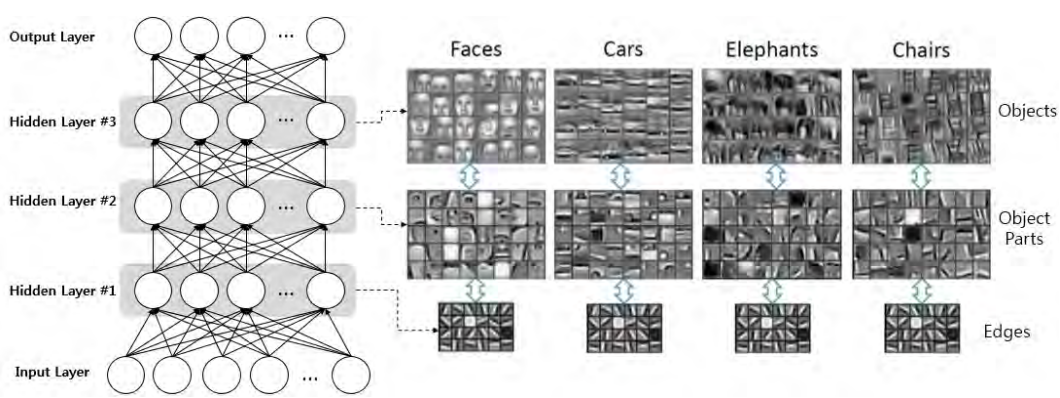
더불어 GPGPU(General-purpose Computing on Graphics Processing Units)를 이용한 컴퓨팅 성능의 비약적 향상으로 인해 기존의 한계점들이 극복됨으로써 음성인식, 얼굴인식, 물체 인식, 자연어 처리 등 다양한 분야에서 기존 방법론들의 성능을 뛰어 넘는 획기적인 결과들이 발표되고 있다.

3.2 비지도 사전 학습

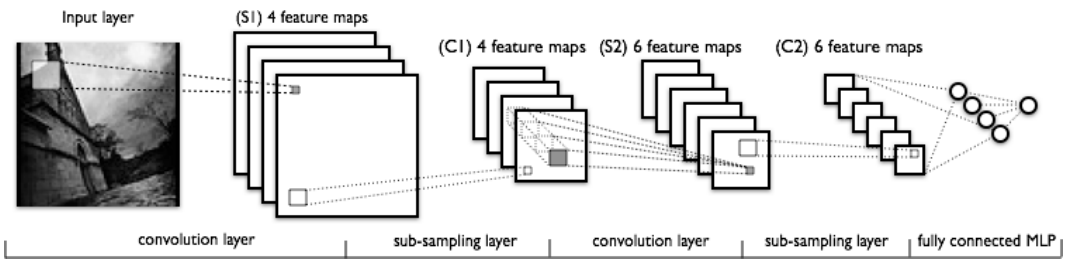
영상 분석은 영상으로부터 영상의 내용과 특성을 표현하는 방식인 특징(Feature)을 추출하는 방법에 따라 분석의 성능이 크게 변화한다. 따라서 영상 내 문자의 특성 대한 여러 가정을 바탕으로 고정적인 특징 추출 방법을 사용할 경우, 조명, 질감, 서식, 배경, 해상도 등의 복잡한 환경에 따라 문자 탐색 및 인식 성능이 크게 저하될 수 있다. 따라서 이러한 가정을 배제하고 자동적으로 데이터로부터 인식 대상의 특성을 가장 잘 표현하는 특징을 추출할 수 있다면 다양한 환경에도 안정적이고 높은 성능을 보이는 인식 모델을 구축할 수 있으며 따라서 이에 적합한 방법으로 비지도 사전학습(Unsupervised Pre-training)이 주목받고 있다[15]. 여기서 비지도(Unsupervised)란 데이터가 학습을 하기 위해서 주어지는 데이터에 대한 정답이 없이 스스로 데이터 자체의 특징을 학습함을 의미하며, 기존에 모델 학습을 위해 데이터와 함께 사람이 직접 생성하여야 하는 정답값이 필요한 지도(Supervised) 학습 방법에 비해 데이터 획득에 드는 비용이 획기적으로 감소되어 영상과 같은 대용량 데이터 분석에 적합하다. 또한 사전 학습(Pre-training)이란 데이터를 직접 인식 모델에 사용하기 전에 미리 데이터의 특징을 자동적으로 추출하는 모델을 학습함으로서 분류기의 성능을 높이는 전처리 과정임을 의미한다.

비지도 사전학습을 위해 제한적 볼츠만 머신(Restricted Boltzmann Machine)[15]이나 Sparse Coding[17] 등의 방법들이 대표적으로 사용되는데, 입력데이터와 첫 번째 은닉층 사이의 변환을 비지도 학습 방법으로 학습하고, 첫 번째 은닉층에 의해 변환된 값을 입력값으로 다시 첫 번째 은닉층과 두 번째 은닉층 사이의 변환을 학습하는 과정을 반복하여 최종적으로 마지막 은닉층까지의 변환을 학습하게 된다.

깊은 구조는 은닉층이 많아짐에 따라, 층별로 서로 계층적(Hierarchial)인 구조와 의미를 갖게 되는데 이는 모델의 복잡도(Complexity)와 표현력이 높여 입력데이터에 대한 식별능력 향상을 가져오게 된다. 실제로 Fig. 2와 같이 각 은닉층별로 학습된 특



(Fig. 2) Hierarchical Feature Learning using Deep Neural Network



(Fig. 3) Structure of Convolutional Neural Network

징을 시각화하면, 하위 층에서 상위 층으로 올라갈수록 보다 추상화되고 고차원적인 특징이 학습되었음을 관찰할 수 있는데, 맨 하위 층은 다양한 각도와 형태를 갖는 선이, 중간층은 객체를 구성하는 각 부분들이, 그리고 최상위 층은 객체단위의 특징들이 추출됨을 확인할 수 있다.

3.3 콘볼루셔날 신경망

딥러닝의 여러 모델중 CNN(Convolutional Neural Network)은 영상과 같은 2차원 구조를 가진 데이터 분석에 적합한 모델으로, 영상의 각 영역에 대해 복수의 필터를 적용하여 특징 지도(Feature Map)을 만들어 내는 콘볼루션 층(Convolution Layer)과 특징 지도를 공간적으로 통합함으로써 크기를 줄여 위치나 회전의 변화에 불변하는 특징을 추출할 수 있도록 하는 통합 층(Pooling Layer)을 번갈아 수차례 반복하는 구조로 구성되어 있다[18]. 이를 통해 점, 선, 면 등의 저수준의 특징에서부터 복잡하고 의미있는 고수준의 특징까지 다양한 수준의 특징을 추출해내고, 이를 최종 단계의 분류 모델의 입력값으로 사용함으로써 기존의 모델에 비해 높은 분류 성능을 나타냄으로써, 다양한 영상 분석 문제에 적용되고 있다[6-8].

콘볼루션 층은 입력 영상의 각 패치에 대하여 필터와 국지 수용장(Local Receptive Field)의 내적에 비선형 활성 함수(Activation Function)을 취함으로서 특징지도(Feature Map)을 구하게 되는데, 다른 네트워크 구조와 비교하여, CNN은 희소 연결성(Sparse Connectivity)과 공유 가중치(Shared Weights)를 가진 필터를 사용하는 특징이 있다. 이러한 연결구조는 학습할 모수의 개수를 줄여주고, 역전파 알고리즘을 통한 학습을 가능하게 만든다.

통합 층(Pooling Layer 또는 Sub-sampling Layer)는 이전 콘볼루션 층에서 구해진 특징 지도의 지역 정보를 활용하여 새로운 특징 지도를 생성한다. 일반적으로 통합 층에 의해 새로 생성된 특징지도는 원래의 특징 지도보다 작은 크기로 줄어드는데, 대표적인 통합 방법으로는 특징 지도 내 해당 영역의 최대값을 선택하는 최대 통합(Max Pooling)과 특징 지도 내 해당 영역의 평균값을 구하는 평균 통합(Average Pooling) 등이 있다. 통합 층의 특징지도는 일반적으로 이전 층의 특징 지도보다 입력 영상에 존재하는 임의의 구조나 패턴의 위치에 영향을 적게 받는다. 즉, 통합 층은 입력 영상 혹은 이전 특징 지도에서의 노이즈나 왜곡과 같은 지역적 변화에 보다 강인한 특징을 추출할 수 있게 되고, 이러한 특징은 분류 성능에 중요한 역할을 한다. 또 다른 통합 층의 역할은 깊은 구조상에서 상위의 학습 층으로 올라갈수록 더 넓은 영역의 특징을 반영할 수 있게 한다. 이러한 특징으로 CNN이 층이 쌓이면서, 아래쪽 특징은 지역적인 특징을 반영하고 상위로 올라갈수록 보다 추상적인 전체 영상의 특징을 반영하

는 특징 생성할 수 있다[18].

이와 같이 콘볼루션 층과 통합 층의 반복을 통해 최종적으로 추출된 특징은 MLP나 SVM, Softmax 층과 같은 완전 연결 층(Fully-connected Layer)과 연결되어 분류 모델 학습 및 예측에 사용된다.

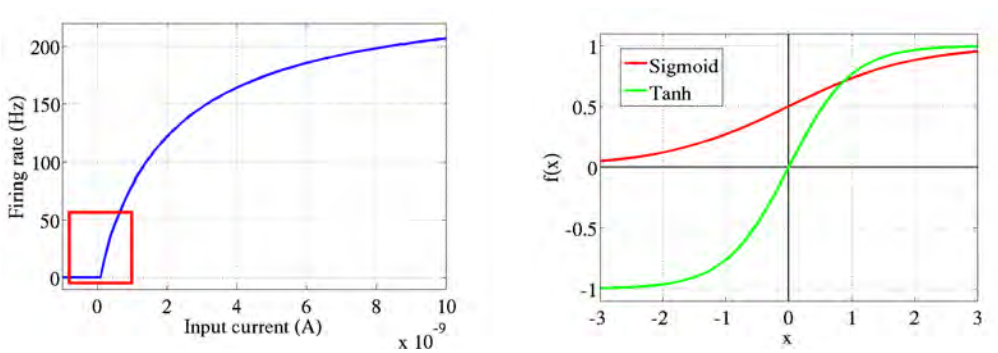
3.4 비선형 활성 함수

CNN을 포함한 대부분의 인공신경망 모델은 표현력(Expressive Power)를 높이기 위해 특징 지도의 각각의 원소에 비선형 함수를 적용한다. 일반적으로 신경망 논문들에서 사용되는 활성 함수(Activation Function)으로는 시그모이드(Sigmoid) 혹은 하이퍼볼릭 탄젠트(Hyperbolic Tangents)가 있다. 하지만 Fig 4.에서 볼 수 있듯이, 이러한 함수들은 생물학적 반응과 특성이 달라 인간의 뇌가 학습하는 방법을 모방하려는 인공신경망의 목표에 부합되지 않는다[19].

이러한 문제를 해결하기 위해 정류 선형 유닛(Rectified Linear Units; ReLU)이 제안되었다[19,20]. 정류 함수(Rectifier Function)인 $\max(0,x)$)은 기존의 비선형 함수보다 생물학적 모델의 특성과 유사한 반응 형태를 가지고 있으며, 계산 복잡도가 낮아 다양한 딥러닝 모델에서 사용되고 있다. 또한 ReLU는 네트워크 학습에서 중요한 몇 가지 장점을 가지고 있는데, 첫 번째로, ReLU는 네트워크의 희소 표현을 가능하게 만든다. 만약 가중치가 균등 분포(Uniform Distribution)로 초기화 되어있다면, 약 50%의 은닉 유닛(Hidden Units)이 0 값을 가지게 되며, 이러한 희소 표현은 네트워크를 학습하는데 있어 중요한 특징으로 작용한다. 다음으로, ReLU는 네트워크의 일부 경로만 활성화(즉 0의 값을 가지지 않음) 시키며, 이러한 하위 집합에서 네트워크는 선형적인 특징을 가지게 된다. 선형성으로 인해, 딥러닝 학습의 어려움중의 하나인 기울기 소실(Vanishing Gradient)문제를 해결하여 학습을 가능하게 만든다.

3.5 정규화

하나 이상의 은닉 층을 가진 다층 퍼셉트론은 이론적으로 임의의 함수를 근사(Approximation)할 수 있는데, 이 때문에 깊은 구조는 학습데이터에 대해서는 매우 낮은 오류를 가지나 실제 테스트 데이터에서는 높은 오류율을 보이는 과적합 문제에 빠지게 쉽다. 이러한 문제를 해결하기 위해 기존의 기계학습의 정규화 방법들이 적용되었으나 깊은 구조 학습에는 효과가 크지 않음이 밝혀진 가운데, 최근



(a) (b)

(Fig. 4) (a) Neural Activation Function from Biological Data (b) Common Activation Function used by Artificial Neural Networks

Dropout 기법이 제안되었다[16,21]. 학습 단계에서 Dropout은 임의로 확률(예를 들어, 0.5)로 은닉 유닛을 각 은닉 층에서 제거하는데, 이것은 특징 탐지기(Feature Detector) 사이의 지나친 동시적응(Co-adaptation)을 막고, 임의로 선택된 특징 집합들이 올바른 정답을 찾도록 유도한다. 실제로 Dropout 기법은 추계적 기울기 하강법(Stochastic Gradient Descent) 학습 기법에서 정규화 항으로 해석될 수 있으며, CNN에서 Dropout은 일반적으로 마지막 단계인 완전 연결층(Fully Connected Layer)에 적용된다. Dropout의 또 다른 역할은 테스트 단계에서 찾아 볼 수 있는데, 테스트 오류를 줄이는 좋은 방법 가운데 하나는 무수히 많은 서로 다른 네트워크로부터의 결과를 평균하는 것이다. 하지만 모든 네트워크들을 일일이 학습하고 테스트하는 것은 실제로 불가능한데, Dropout은 매 학습 단계마다 임의의 은닉 노드를 제거함으로써 새로운 네트워크를 생성하고 학습하는 효과를 나타낸다. 즉, 서로 다른 네트워크들이 가중치를 공유하며 학습되는 것인데, 이것은 Dropout을 통하여 학습된 모델들이 테스트 단계에서 가중치를 일정 비율로 나누어 주는 것으로서, 가능한 모든 하위 모델들의 기하평균 연산을 수행(또는 근사화)하는 것과 동일한 역할을 한다.

4. 딥러닝 기반 문자 인식

영상으로부터 문자를 인식하기 위해서는 영상 내에서 문자의 후보 위치를 탐색하는 문자 검출, 검출된 문자 위치에서 문자 혹은 단어를 추출하는 문자 인식의 두 단계를 거쳐야 한다. 본 장에서는 딥러닝 모델을 이용하여 일반 영상 내에서 문자를 탐색, 인식하는 방법에 대해 설명하고 관련된 최신 연구 사례를 살펴본다.

4.1 문자 검출

영상 내에서 문자의 후보 위치를 찾기 위해서는 영상을 보다 작은 패치로 쪼개고, 각 패치별로 이동창을 통해 각 영역별 문자 포함 여부를 판단하여야 한다. 주어진 영역에서 문자 포함 여부를 판단하기 위해서는 우선 문자를 나타내는 특징을 학습하여야 하며, [12, 22, 23]에서는 대량의 영상으로부터 문자의 특징을 학습하기 위해 비교적 간단하고 속도가 빠른 K-means 군집 방법의 변형을 다음과 같이 사용하였다.

(1) 주어진 영상 내의 다양한 부분에서 원래 이미지의 크기보다 작은 영상 패치를 생성한다. 예를 들어 8 x 8 크기의 영상 패치를 m개 생성할 경우, 해당 영상은 64차원의 벡터 m개로 표현될 수 있다.

(2) 각 영상 패치 벡터를 밝기와 대비에 따라 정규화 한다. 즉, 각 영상 패치 벡터를 $\tilde{x}^{(i)}$라고 하고 해당 벡터의 평균을 $\mu_{\tilde{x}_i}$, 분산을 $\sigma_{\tilde{x}_i}$라고 하면, 정규화된 영상 패치 벡터 $\hat{x}^{(i)}$는 다음과 같이 정의된다.

$$\hat{x}^{(i)} = \frac{\tilde{x}^{(i)} - \mu_{\tilde{x}_i}}{\sigma_{\tilde{x}_i}}$$

(3) 앞서 얻어진 m개의 $\hat{x}^{(i)}$들에 대해 ZCA(Zero Component Analysis)를 이용한 백색화를 한다. 즉, 얻어진 $\hat{x}^{(1)}, \ldots, \hat{x}^{(m)}$각 요소에 대해 평균 벡터 *M*과 공분산 행렬 Σ를 구하고, 공분산 행렬 Σ에 대한 고유벡터 분해(Eigenvector Decomposition)을 통해 구해진 고유행렬 벡터 *V*와 고유값 행렬 *D*를 이용해 다음과 같이 백색화된 영상 패치 벡터를

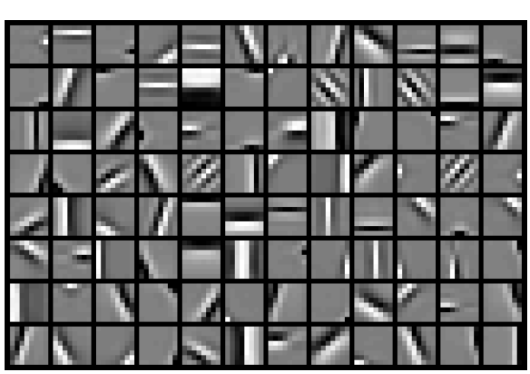
얻는다.

$$x^{(i)} = VD^{-1/2}V^T\left(\hat{x}^{(i)} - M\right) = P\left(\hat{x}^{(i)} - M\right)$$

(4) 앞서 얻어진 백색화된 영상 패치 벡터 $x^{(i)}$로부터 문자 탐지 모델에 사용될 특징 $z^{(i)}$를 자동적으로 추출하기 위한 변환 필터를 학습 시킨다. 즉, 만약 $x^{(i)}$가 64차원 벡터이고 d개의 특징을 추출하고자 한다면 d개의 64차원 필터들의 행렬인 $D \in \mathbb{R}^{64\times d}$를 다음과 같은 최적화 문제의 해를 찾아 구한다.

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} \left\| Ds^{(i)} - x^{(i)} \right\|_2^2 \\ \text{subject to} \quad & \left\| s^{(i)} \right\|_1 = \left\| s^{(i)} \right\|_\infty, \; i = 1, \ldots, m \\ & \left\| D^{(j)} \right\|_2 = 1, \; j = 1, \ldots, d. \end{aligned}$$

여기서 $s^{(i)}$는 $x^{(i)}$를 D의 필터들의 조합으로 표현하기 위한 표현 벡터이며, 위 최적화 문제의 첫 번째 제약식에 의해 벡터의 요소 중 하나만 0이 아닌 값을 가지게 된다. 위의 과정을 거쳐 주어진 영상 패치 벡터 $\tilde{x}^{(i)}$를 d개의 특징 벡터로 변환하는 필터 행렬 D가 학습되며, 이 때, 각 필터를 시각화 하면 아래 그림과 같다.



(Fig. 5) Feature Transform Filter from Unsupervised Pre-training

따라서 새로운 영상 패치 벡터 $\tilde{x}$가 주어지면, 앞서 행했던 정규화와 백색화를 거쳐 x를 생성하고, 학습된 필터 행렬 D를 이용하여 특징 벡터 $z' \in \mathbb{R}^d$를 $z' = D^T x$와 같이 추출하게 된다.

위와 같이 영상 패치 내 문자에 대한 특징이 학습되면, 이를 이용하여 새롭게 주어진 영상 조각의 문자 여부를 판별하는 모델이 필요한데, 이를 학습 시키기 위해서 CNN을 사용한다. CNN의 콘볼루션 층은 사전 학습으로 구해진 필터 행렬 D를 이용하여 변환된 특징 벡터를 추출하게 되는데, 이때 사용되는 변환 함수는 다음과 같다.

$$z = h\left(D^T x\right) = \max\left\{0, \left|D^T x\right| - \alpha\right\}$$

이어지는 통합층과, 반복되는 콘볼루션-통합층을 통해 보다 고차원의 특징을 추출하고 이러한 특징들이 문자를 나타내는지 판별하게 되는데, [11] [12]에서는 이를 위해 SVM 이진 분류기를 사용하였다. 따라서 새로운 이미지 영역이 주어지면, 사전 학습된 필터를 통해 해당 이미지 영역으로부터 특징을 추출하고, CNN을 모델을 통해 이를 보다 고차원의 특징으로 변환한 이후에, 최종적으로 학습된 SVM 모델을 통해 해당 이미지 영역이 문자인지 아닌지를 판별하게 되는 것이다.

4.2 문자 인식

주어진 영상에 대해서 특정 크기의 이동 창(Sliding Window)을 생성하고, 해당 영역에 앞서 구축된 문자 검출 모델을 적용하여 해당 창 내에 문자가 존재할 확률을 점수화 할 수 있다. 실제 영상 내에는 다양한 크기의 문자가 존재하므로 문자 검출 성능이 이동 창의 크기에 따라 달라질

수 있어 원본 영상을 확대하거나 축소하여 문자 검출 모델을 적용하는 다중규모 이동 창(Multiscale Sliding Window) 인식기법을 사용하는 것이 중요하다[12].

일반적으로 영상 내의 문자는 특정 영역 내에 밀집하여 정렬이 되어 있는데 각 문자로부터 단어를 인식하기 위해서는 문자들이 정렬된 영역과 해당 영역 내에서 각 문자들의 위치를 정확하게 추출하는 것이 필요하다. 문자 검출 모델은 해당 이동 창 내에 문자가 정확하게 위치해 있을수록 높은 점수를 나타내는데, 이러한 특성을 이용하면 특정 영역 내에서 각 문자들의 위치 및 단어 구분 지점을 파악해 낼 수 있다. 즉, 원본 이미지에서 문자가 검출된 영역 주변에 대한 문자 검출 모델의 결과값을 관찰하면 문자가 정확하게 위치한 지점에서 가까울수록 양(Positive)의 값을 가지고, 문자가 아니거나 문자 중간에서는 음(Negative)값을 가지게 되므로, 문자 탐색 모델의 결과를 $R(x)$라고 하면, 다음과 같은 비최대지점 억제(Non-maximal Suppression : NMS)[24] 식에 의해 문자 존재 위치를 파악해 낼 수 있다.

$$R'(x) = \begin{cases} R(x) & R(x) \geq R(y), \forall y : |x-y| < \delta \\ 0 & \text{otherwise} \end{cases}$$

즉, 해당 영역에 문자와 단어가 정확하게 위치하면 양의 값의 비율이 높고, 제대로 정렬되지 못한 경우에는 음의 값의 비율이 높다. 따라서 NMS를 통해 추출된 해당 영역의 정점의 값의 평균을 $\bar{R}_{\text{peak}}$ 라고 하면, 이것이 미리 정해진 역치값 τ보다 큰 경우 해당 영역에 문자/단어가 잘 포함되어 있다고 판단하고, 그렇지 않은 경우 해당 영역을 문자/단어 인식 대상에서 제외 한다. 만약 해당 영역이 문자/단어가 정확하게 위치되어 있을 것이라고 판단된 인식 후보라면 정점 값들의 위치에 대한 분포를 이용하여 추가적인 분석을 하여 문자/단어의 정렬을 할 수 있는데, 일반적으로 단어를 이루는 문자들 사이의 간격이 일정하므로, 정점간의 간격에 대한 분포가 매우 큰 경우, 이는 실제로 문자/단어들이 위치한 영역이 아님에도 불구하고 탐색기가 문자로 인식한 영역으로 판별하여 인식 후보 영역에서 제외한다. 또한, 해당 영역의 정점간 거리의 중간값(Median)의 수배 이상 클 경우, 해당 영역은 동일한 영역으로 볼 수 없으므로 해당 정점의 사이를 기준으로 두 개의 영역으로 분리한다.

최종적으로 이동 창의 크기와 이동 범위에 따라 같은 문자/단어에 대해 중첩된 인식 영역이 발생될 수 있으므로 두 영역의 겹치는 범위가 큰 경우 정점의 값의 평균값이 큰 영역을 최종 문자/단어 인식 대상으로 결정하고 나머지 영역을 인식 대상에서 제외한다. 즉, 두 영역 B_1과 B_2각각의 정점 값의 평균을 각각 s_1과 s_2라고 하고 $|B_1|$, $|B_2|$를 두 영역의 넓이, $|B_1 \cap B_2|$를 두 영역이 겹치는 영역의 넓이라고 할 때, 아래의 식을 만족 시키면 정점값 평균이 작은 영역을 문자/단어 인식 대상에서 제외한다.

$$\frac{|B_1 \cap B_2|}{\min\{|B_1|, |B_2|\}} > \frac{1}{2}$$

앞선 일련의 과정을 거쳐 주어진 문자/단어 인식 대상 영역에 대하여 다음의 과정을 거쳐 해당 영역 내의 문자와 단어를 인식해 낸다.

(1) 단어와 단어간 구분을 위해서 앞에서 문자 탐지기와 NMS 기법을 이용한다. 문자 탐지기는 문자가 정확하게 위치하는 경우 큰 양의 값을 나타내는 특성을 반대로 이용하면, 문자

사이, 특히 단어와 단어 사이의 공간에서는 큰 음의 값을 가지게 되므로 이를 반전하면, 문자와 문자, 단어와 단어 사이에서 정점이 발견된다. 다양한 환경과 서식에 따라 문자 사이의 공간과 단어 사이의 공간을 분리하는 것이 어려울 수 있는데, 음의 문자 탐색기 값에 적용한 NMS를 통해 발견된 각 정점들의 값 $\tilde{R}'(x)$에 미리 정해진 역치값 τ를 이용하여 $R'(x) = \max\left\{\tilde{R}'(x) - \tau, 0\right\}$로 변환을 하여 남은 정점들을 단어 간의 경계로 판별한다. $R'(x)$의 크기는 이러한 경계에 대한 확실성에 대한 척도로 사용할 수 있다.

(2) 앞서 파악된 단어간 경계를 이용하여 개별 단어 영역을 추출한 후, 해당 영역에서 이동 창(Sliding Window)방식으로 사전에 학습시킨 문자 분류기를 적용한다. 예를 들어 현재 인식 하려는 문자가 영어 대문자(26개), 소문자(26개)와 숫자(10개)일 경우 총 62가지의 문자 중 하나로 인식해야 하며, 이는 62개의 분류가 있는 분류 문제가 된다. 해당 단어 인식 영역에서 총 N개의 이동 창을 생성하였다면 결과적으로 62 x N 크기의 분류 결과 행렬 M이 생성된다. 즉, 높은 $M(i,j)$값은 j위치의 이동 창이 i번째 글자를 포함하고 있을 가능성이 높다는 의미이다. 이 때, 각 이동 창의 위치 j에서 가장 $M^{(j)}$의 값이 큰 값과 작은 값의 차이를 c_j를 구하면 이는 해당 위치에서 가장 값이 큰 문자의 상대적인 확신도(Confidence)라고 할 수 있다. 따라서 영역 내 모든 이동 창에 대해서 확신도 벡터 $c = (c_1, c_2, \cdots c_N)$를 구한다.

(3) 앞서 구한 확신도 벡터에 NMS를 적용하면 다음과 같이 적용하면 정점에서 각 문자의 위치를 정확하게 파악해낼 수 있다.

$$c'_j = \begin{cases} c_j & c_j \geq c_i, \forall i : |j-i| \leq \delta \\ 0 & \text{otherwise} \end{cases}$$

이 때, 주어진 단어 사전 $\mathcal{L}$ 내의 각 후보 단어 w에 대해서 문자 정렬 점수 S_M^w를 다음과 같이 구한다.

$$S_M^w = \max_{\ell^w \in L^w} \left(\sum_{k=1}^{|w|} M\left(w_k, \ell_k^w\right) \right)$$

즉, 각 후보 단어 w마다 단어 내 문자들에 위치에 따른 문자 분류기 점수의 합 S_M^w가 최대가 되는 문자 정렬 벡터 ℓ^w를 찾는 것이며, 단어 내 문자간의 간격이 유사하다는 조건을 추가하여 문자간 간격의 분산을 최소화 하도록 하는 수정 정렬 점수 $\hat{S}_M^w$를 Viterbi 알고리즘과 같은 동적 계획법을 통해 효율적으로 구할 수 있다. 최종적으로 주어진 사전 내의 모든 단어에 대해 위의 과정을 수행하고, 아래 식과 같이 그 중 가장 점수가 높은 단어를 인식된 단어로 결정한다.

$$w^\star = \arg\max_{w \in \mathcal{L}} \hat{S}_M^w$$

위의 과정을 영상 내의 복수 문자 후보 영역에 대해 반복함으로써, 주어진 영상 내에서 검출된 모든 단어들에 대한 인식이 완료 된다.

5. 결 론

본 논문에서는 일반적인 영상에서 문자 정보를 추출하기 위한 딥러닝 기반 문자 인식 방법을 살펴보았다. 문자 인식을 위해서는 다양하고 복잡

한 환경에서 영상 내 문자들이 가지는 특징을 효과적이고 유연하게 추출해내는 방법이 필요한데, 본 논문에서는 기존의 고정적인 특징 추출 방법의 한계를 지적하고, 이에 대한 대안으로 활발히 연구되고 있는 다층 신경망을 통한 계층적이고 자동화된 특징 학습 방법에 관해 설명하였다. 딥러닝 방법은 학습 영상들로부터 비지도 학습을 통해 다양한 환경에서 문자를 가장 잘 표현하는 특징을 스스로 학습하고, 이를 CNN과 같은 다층 구조의 신경망을 이용해 보다 고차원적인 특징으로 변환하며, 이를 문자 여부를 판별하는 문자 검출 모델의 입력값으로 사용하므로 보다 정확하고 성능이 높은 문자 검출 및 인식 모델을 구축할 수 있다. 향후 영상 내 문자 인식을 위한 보다 최적화된 네트워크 구조와 학습 방법이 개발된다면 보다 다양하고 복잡한 환경에서 안정적으로 실시간 문자 인식 및 추출이 가능하여 다양한 상업적, 공공적 서비스 개발에 널리 활용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] K. Jung, K. I. Kim and A. K. Jain, "Text information extraction in images and video : a survey", Pattern Recognition, vol. 37, no. 5, pp. 977-997, 2004

[2] S. Singh, "Optical character recognition techniques : a survey", Journal of Emerging Trends in Computing and Information Sciences, vol. 4, no. 6, pp. 545-550, 2013

[3] C. Patel, A. Patel and D. Patel, "Optical chracter recognition by open source OCR tool Tesseract : a case study", International Journal of Computer Applications, vol. 55, no. 10, pp. 50-56, 2012

[4] C. Yao, X. Bai and W. Liu, "A unified framework for Multioriented text detection and recognition", IEEE Transactions on Image Processing, vol. 23, no. 11, pp. 4737-4749, 2014

[5] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI", Large-scale Kernel Machines 34, pp. 1-41, 2007

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Advances in Neural Information Processing Systems 25, 2012, pp. 1097-1105

[7] I. J. Goodfellow, Y. Bulatov, J. Ibraz, S. Arnoud and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks", arXiv:1312.6082, 2014

[8] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", arXiv:1311.2524, 2013.

[9] A. Bissacco, M. Cummins, Y. Netzer and H. Neven, "PhotoOCR : reading text in uncontrolled conditions", in Proceedings of the IEEE Conference on Computer Vision, 2013, pp. 785-792

[10] K. Koray, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu and Y. LeCun, "Learning convolutional feature hierarchies for visual recognition", Advances in Neural Information Processing Systems 23, 2010, pp. 1090-1098

[11] K. Wang, B. Babenko and S. Belongie, "End-to-end scene text recognition", in Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 1457-1464

[12] T. Wang, D. J. Wu, A. Coates and A. Y. Ng, "End-to-end text recognition with convolutional neural networks", in Proceedings of the International Conference

on Pattern Recognition, 2012, pp. 3304-3308
[13] O. Alsharif and J. Pineau, "End-to-end text recognition with hybrid HMM maxout models", arXiv:1310.1811, 2013
[14] D. E. Rumelhart and J. L. McClelland, “Parallel distributed processing: explorations in the microstructure of cognition”, Cambridge: MIT Press, 1986
[15] 15. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks", Science, vol. 313, no. 5786, pp. 504-507, 2006
[16] P. Baldi and P. J. Sadowski "Understanding dropout", Advances in Neural Information Processing Systems 26, 2013, pp. 2814-2822
[17] H. Lee, A. Battle, R. Raina and A. Y. Ng, "Efficient sparse coding algorithms", Advances in Neural Information Processing Systems 19, 2007, pp. 584-592
[18] C. Szegedy, W. Liu., Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions", arXiv:1409.4842, 2014
[19] X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks", in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315-323
[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines", in Proceedings of International Conference on Machine Learning, 2010, pp. 807-814
[21] G. E. Hinton, N. Srivastava. A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", arXiv:1207.0580, 2012
[22] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu and A. Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning", in Proceedings of the International Conference on Document Analysis and Recognition, 2011, pp. 440-445
[23] A. Coates, H. Lee and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning", AISTATS, 2011
[24] A. Neubeck, L. V. Gool, "Efficient non-maximal suppression", in Proceedings of the International Conference on Pattern Recognition, 2006, pp. 850-855

저 자 약 력

정 규 환

이메일 : khwan.jung@vuno.co

- 2005년 포항공과대학교 산업경영공학과 (학사)
- 2010년 포항공과대학교 산업경영공학과 (박사)
- 2010년 포항공과대학교 미래형기계기술사업단 / 박사 후연구원
- 2011년 SK텔레콤 플랫폼기술원 / 매니저
- 2011년~2014년 SK플래닛 플랫폼기술원 / 매니저
- 2014년 삼성전자 종합기술원 / 전문연구원
- 2015년~현재 VUNO Inc. / CTO
- 관심분야 : 머신러닝, 딥러닝, 추천시스템, 컴퓨터비전

김 현 준

이메일 : dannis@vuno.co

- 2003년 인하대학교 컴퓨터공학과 (학사)
- 2005년 인하대학교 컴퓨터공학과 (석사)
- 2009년 인하대학교 컴퓨터공학과 (박사수료)
- 2005년~2014년 삼성전자 종합기술원 / 전문연구원
- 2014년~현재 VUNO Inc. / CSO
- 관심분야 : 머신러닝, 딥러닝, 컴퓨터비전

이 예 하

이메일 : yeha.lee@vuno.co

- 2006년 포항공과대학교 컴퓨터공학과 (학사)
- 2012년 포항공과대학교 컴퓨터공학과 (박사)
- 2012년~2014년 삼성전자 종합기술원 / 전문연구원
- 2014년~현재 VUNO Inc. / CEO
- 관심분야 : 머신러닝, 딥러닝, 컴퓨터비젼, 음성인식, 정보검색