텐서플로우(Tensorflow)를 사용한 한글 폰트의 특징 검출과 Unicode 자동 생성을 위한 알고리즘

Tensorflow-based Korean Font Feature Classification and Intelligent UNICODE Generation of Printed Korean Font

저자
(Authors)

강유현, 고운, 조동섭
Youhyun Kang, Wun Ko, Dongsub Cho

출처
(Source)

정보 및 제어 논문집 , 2017.10, 162-163 (2 pages)
INFORMATION AND CONTROL SYMPOSIUM , 2017.10, 162-163 (2 pages)

발행처
(Publisher)

대한전기학회
The Korean Institute of Electrical Engineers

URL

http://www.dbpia.co.kr/Article/NODE07261370

APA Style

강유현, 고운, 조동섭 (2017). 텐서플로우(Tensorflow)를 사용한 한글 폰트의 특징 검출과 Unicode 자동 생성을 위한 알고리즘. 정보 및 제어 논문집, 162-163.

텐서플로우(Tensorflow)를 사용한 한글 폰트의 특징 검출과 Unicode 자동 생성을 위한 알고리즘

강유현, 고운, 조동섭
이화여자대학교 컴퓨터공학과

Tensorflow-based Korean Font Feature Classification and Intelligent UNICODE Generation of Printed Korean Font

Youhyun Kang, Wun Ko, and Dongsub Cho
Department of Computer Science and Engineering, Ewha Womans University

Abstract - 전자 문서를 이용한 디지털 범죄의 증가에 따라 한글 전자 문서에 대한 법적 효력과 증거물 인정의 체계적인 정립이 필요성이 제기되고 있다. 출력된 한글 문서의 위조, 변조 등의 작업을 이해하고 한글 폰트의 재구성을 통해 원본 확인 및 법적 증거물의 신뢰를 높일 수 있다. 본 논문에서는 한글 폰트의 다양한 출력 방식과 출력 형태를 분석한 후 Tensorflow를 이용하여 폰트의 특징을 추출하고 해당되는 한글 유니코드를 생성시키는 알고리즘을 제안한다. 또한 한글의 구조적인 특징과 서체 디자인의 유형을 데이터베이스화하여 디지털 포렌식에 활용할 수 있도록 한다.

1. 서 론

IT 기술의 발전 및 정보의 디지털화로 인해 컴퓨터 관련 범죄뿐만 아니라 일반 범죄에서도 중요 증거 또는 단서를 문서 파일, 인터넷 로그, 사진, 동영상 파일 등에서 찾는 전문적인 디지털 포렌식 기술이 요구되고 있다. 특히 전자 문서를 이용한 행정업무의 증가와 함께 한글 전자 문서의 위조, 변조, 조작 등의 범죄가 늘어나고 있다. 컴퓨터 저장장치에 보관된 문서 파일 등은 전자적 증거로 취급하기 편리하나 인쇄된 전자 문서에 대한 증거 분석은 그 작업이 어렵다. 따라서 문서에 사용된 한글 폰트에 대한 분석 및 인식을 통해 한글 전자 문서의 법적 효력과 증거물 인정의 체계적인 정립이 필요하다. 이와 관련하여 한글 문자 인식과 관련된 연구로는 LHS를 통한 폰트 유사도 판정 연구[1], 자소 분할 문자 인식 연구[2] 등의 연구가 이루어졌다. 하지만 각 한글 폰트는 11,172자의 문자로 구성되고 한글 자소 조합이 복잡하며, 폰트의 종류가 수 없이 많고 매년 새로운 폰트가 제작되므로 대량의 폰트를 연구하고 정확한 한글 문자 인식 시스템을 만들기에 어려운 실정이다[3].

본 논문에서는 한글 폰트의 다양한 출력 방식과 출력 형태를 분석하고, 딥러닝(Deep Learning)의 한 종류인 CNN(Convolution Neural Network)을 기반으로 하여 폰트의 특징을 추출한 뒤 해당하는 한글 유니코드(Unicode)를 자동으로 생성하는 알고리즘을 제안한다. 제안하는 알고리즘은 기계학습과 딥러닝을 만들어진 오픈소스 Tensorflow를 사용한다.

2. 본 론

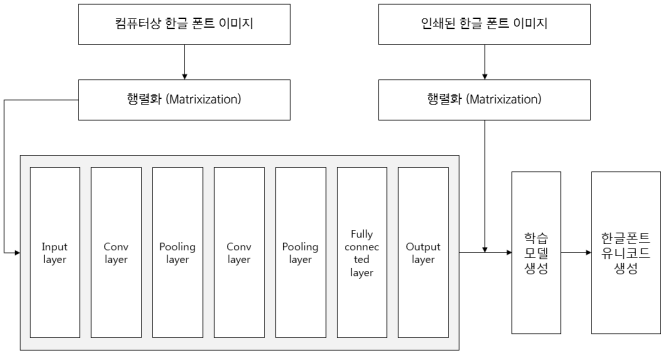
2.1 한글 폰트의 출력 방식 및 출력 형태 분석

컴퓨터 폰트의 방식은 크게 비트맵 폰트(Bitmap font), 윤곽선 폰트(Outline fone), 그리고 구조적 폰트(Structured font)로 나눌 수 있다. 비트맵 폰트는 스크린과 프린터 등의 출력 장치에 점들을 찍음으로써 글자를 출력한다. 가장 간단한 표현 방식으로, 저장 공간을 적게 차지하나 축소, 확대 등의 출력이 어려운 단점이 있다. 반면에 윤곽선 폰트는 벡터의 형태로 표현된 폰트로 글자의 윤곽을 직선, 자유 곡선 등으로 나타낸다. 확대나 각도 조절에 자유롭고 컴퓨터의 발전으로 속도도 향상되어 오늘날 대부분의 시스템에서 사용되는 폰트로써, 특히 마이크로소프트사와 애플사가 만든 트루타입(TrueType) 폰트의 경우, 힌팅(Hinting)이라는 기술을 사용하여 확대, 축소, 변형 처리에 대한 자유도가 높고 저해상도 스크린과 프린터에서도 글꼴을 보존할 수 있다[4]. 구조적 폰트는 글자 획의 중심선을 이용하여 글자의 모양을 표현한다. 구조적 폰트를 사용하면 데이터의 양이 크게 줄고 설계 및 변형이 용이하지만 글자 생성 속도가 느려 일부 분야에서만 응용된다[5].

폰트의 형태는 글자의 크기, 장평, 굵기 등에 영향을 받는다. 일반적으로 폰트의 크기는 9에서 12사이의 포인트가 사용되며 6포인트 이하의 글자는 매우 알아보기 힘들고, 장평 값에 따라 원래의 형태와 전혀 다르게 넓적하거나 길쭉한 형태의 글자가 되기도 한다. 따라서 이러한 요소에 주의하여 한글 폰트의 형태적 특징을 분석해야한다. 특히 문서용 한글 폰트는 종이에 출력된 상태로 감식되기 때문에 프린터 기기나 종이의 상태에 따라 형태에 왜곡이 있을 수 있으므로 문서에 사용된 폰트의 종류와 크기, 굵기에 따라 출력 정보를 체계화할 필요가 있다.

2.2 한글 폰트의 특징 검출과 Unicode 자동 생성 알고리즘

본 논문에서 제안하는 알고리즘은 한글 폰트 문자 인식을 위해 이미지 처리에 높은 성능을 보이는 CNN을 기반으로 하여 인쇄된 전자 문서의 한글 폰트의 특징을 검출한다. 그리고 검출한 특징을 바탕으로 폰트를 식별하기 위한 유니코드를 생성한다. 제안하는 알고리즘은 Tensorflow를 사용하여 구현한다[6]. <그림 1>은 제안하는 알고리즘을 나타낸 것이다.



<그림 1> 한글 Unicode 생성 작업 흐름도

2.2.1 CNN

CNN(Convolution Neural Network)은 자연어 처리, 이미지 인식 등에다 양한 분야에서 혁신적 성과를 얻고 있는 딥러닝의 한 종류이다. CNN은 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 convolutional layer에서 얻은 특징맵에서 가장 핵심적인 부분을 추출(Sub sampling)하여 특징맵을 축소한다. 특징을 유지한 상태로 축소하므로 위치 변경으로 인한 결과 변화를 막아주고 데이터 사이즈의 축소를 통해 신경망의 성능을 높여준다. 이렇게 추출된 특징들을 기반으로 분류를 한다[7].

<그림 2> 한글 폰트 이미지 행렬화

본 논문에서 제안하는 알고리즘은 CNN을 통해 한글 폰트 이미지의 특징을 검출한다. 우선 컴퓨터 시스템 내 폰트 원본으로부터 28×28 크기의 폰

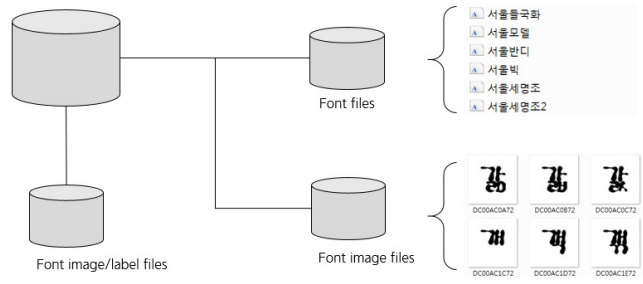
트 이미지를 추출한다. 추출한 이미지를 신경망에 넣을 수 있도록 이미지 매트릭스로 변환한다. <그림 2>와 같이 한글 폰트 이미지의 각 픽셀 색상을 추출하여 0(백색)부터 1(흑색)사이의 값으로 정규화 한다. 변환된 이미지 매트릭스를 CNN에 입력 데이터로 넣은 뒤 두 번의 convolutional layer와 pooling layer 단계를 거쳐 여러 64개의 특징맵을 찾는다. 구한 특징맵을 이용하여 Fully connected layer에 넣어서 hypothesis를 세우고 학습모델을 만들어, 이후 인쇄된 전자 문서의 한글 폰트 이미지가 입력으로 들어왔을 때 해당 폰트가 어떤 폰트인지 예측할 수 있도록 한다. 아래 <그림 4>의 코드는 알고리즘의 CNN을 Tensorflow에서 구현한 코드의 일부이다.

```
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....
```

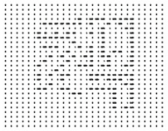
<그림 4> 한글 폰트 특징 검출에 대한 Tensorflow 코드

2.2.2 Unicode 생성

한글 폰트의 체계를 정립함에 있어 유니코드의 사용을 제안한다. 본래의 유니코드는 각 나라별 언어를 모두 표현하기 위해 나온 코드 체계이나 본 논문에서는 한글 폰트의 종류에 따라 각 폰트마다 0000부터 FFFF사이의 임의의 유니코드 값을 할당하고, 폰트 이미지의 label로 관리함으로써 CNN의 분류기준으로 활용한다. 예를 들어 한글 폰트 중 하나인 새굴림체는 AA32의 유니코드로 표현할 수 있고, 새궁서체는 AA33, 새돋움체는 AA34 등으로 표현한다. <그림 3>은 한글 폰트와 디자인 유형을 데이터베이스화한 데이터베이스 시스템 구조를 나타낸다. 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.



CODE	LABEL	NAME
DC00	1101110000000000	디컴꽃망울M
DC01	1101110000000001	디컴별숲B
DC02	1101110000000010	디컴별숲L
DC03	1101110000000011	디컴별숲M



<그림 3> 한글 폰트 데이터베이스 시스템

3. 결 론

본 논문에서 제안한 알고리즘은 전자 문서의 한글 폰트를 이미지로 추출하여 CNN을 통해 폰트를 식별하고 한글 유니코드를 자동 생산하는 알고리즘이다. MNIST 손글씨 데이터 분류 문제를 토대로 하여 한글 이미지에 대해 특징을 검출하는 부분에 관해 구현을 하였고, 한글 폰트에 대한 유니코드를 생성하는 데이터베이스를 구축하였다. 그러나 아직 딥러닝 알고리즘을 구현하기에 미흡한 점이 많아 추가적인 연구가 필요하다. 다음 연구로는 CNN을 더 심층적으로 분석하고, 폰트 이미지 매트릭스 데이터를 확보하여 제안한 알고리즘의 성능을 테스트할 예정이다. 또한 추가적으로 폰트 종류에 대한 유니코드 label 외에 한글 글자에 대한 기존 유니코드를 label을 더하여 한글 폰트 및 글자 식별에 대한 연구를 진행할 것이다.

ACKNOWLEDGMENT

본 연구는 대검찰청 용역연구개발사업과제 [WORD문자 감정을 위한 지능형 폰트 분류 시스템 구축] 의 연구결과로 수행되었음.

[참 고 문 헌]

[1] 이하경, 조동섭, "컴퓨터 문서 감정용 FONT 이미지의 유사도 판정 시스템 설계." 정보 및 제어 논문집, pp. 217-218, 2012
[2] 이진수, 권오준, 방승양. "개선된 자소 인식 방법을 통한 고인식률 인쇄체 한글 인식." 정보과학회논문지(B), 23(8), pp. 841-851, 1996
[3] 이준구, 정윤수, 김두식. "딥러닝을 이용한 대량의 한글 폰트 인식 방법." 한국통신학회 학술대회논문집, pp. 154-155, 2017
[4] Laurence Penny, "A history of TrueType", TrueType Typography Technical Report, TrueType Development Team at Apple, 1999
[5] Richard Rubinstein, "Digital Typhography," Addison-Wesley, pp. 140-147, 1998.
[6] Tensorflow, https://www.tensorflow.org/
[7] Ciresan, Dan Claudiu, et al. "Convolutional neural network committees for handwritten character classification." Document Analysis and Recognition (ICDAR), 2011 International Conference on. IEEE, 2011.