CNN 기반 한국어 유아 손글씨 OCR 모델 연구

CUAI 5기 DA T2 (매의 눈)
곽수민(응용통계학과), 김우엽(기계공학부)

[요약] 본 연구는 차세대 에듀테크 서비스의 핵심 기술로 자리 잡은 OCR 기술을 유아 손글씨 인식에 특화하여 연구하였다. 구현한 CNN 기반 모델은 Test Accuracy 85.55%의 성능을 기록하였으며 차후 교육 업계와 연동한다면 아이들의 효율적인 학습을 위한 교두보의 역할을 수행할 수 있을 것이다.

1. 서 론

컴퓨터의 발달 이전에는 대부분 종이를 통해 수기로 기록해 문서를 저장하였다. 컴퓨터의 발달로 대부분 전자문서로 저장하는 지금, 과거의 기록을 전자문서로 변환하기 위해선 광학문자인식(이하 OCR) 기술이 필수적이다. 단순히 종이 문서를 스캔해서 이미지로 저장한다면 저장의 기능만 전자로 대체할 뿐 검색이나 자유로운 수정 등 전자문서가 가지고 있는 이점은 전혀 활용할 수 없기 때문이다. 이와 같은 시대적 흐름 속에서 OCR 기술은 굉장한 발전을 이뤄왔다.

또한 최근 교육·인공지능(AI) 스타트업 업계에서는 OCR이 핵심기술로 떠오르면서 소비자들에게 더욱 편리한 서비스를 제공하기 위해 OCR을 속속 도입하고 있다. 쌍방향 디지털 교육 환경이 중요해진 교육 업계에서는 OCR이 차세대 에듀테크 서비스의 핵심 기술로 자리 잡았다.
교육업계에 따르면, 천재교육과 콴다, 프리딕션 등은 OCR 관련 분야를 미래 교육 기술의 주요 기반 중 하나로 보고 그동안 축적해 온 학습 데이터와 교육교재 제작 기술을 첨단기술에 접목하고 있다.
업체들은 기술 개발을 통해 필기 인식, OCR 기술 도입에 집중하고 있다. 빅데이터 분석, 자연어 처리, 이미지 인식(OCR, 필기체, 수식, 이미지 등), 머신러닝(딥러닝 포함), AI 수학, 학습분석 관련 6가지 분야를 미래 교육 기술의 기반으로 삼고 온라인 기반의 스마트러닝 시장을 열어가고 있다.[7]
이러한 사회적 흐름에서 본 연구는 OCR 기술이 유아 교육에 활용될 수 있도록 유아의 손글씨 데이터 인식에 최적화된 OCR 모델을 개발하고자 시작하였다. 데이터는 교원그룹에서 제공한 유아 손글씨 데이터를 활용하였다.

OCR 모델의 경우 크게는 주어진 사진에서 글자 영역을 인식하는 Scene Text Detection 모델과 글자 영역에서 글자를 읽어오는 Scene Text Recognition 모델로 나누어진다. Scene Text Detection의 경우 CRAFT라고 불리는 모델[2]이 대표적으로 사용되며, Scene Text Recognition의 경우 컨볼루션 신경망에 순환 신경망을 덧붙여 만든 RCNN [1]이 가장 흔하게 사용되고 있고 이외에 VGG[3], ResNet[4] 등 여러 신경망을 활용한 기법들이 사용되었다. 여기에 덧붙여 인식된 글자의 순서 정보를 활용하여 예측하기 위해 BiLSTM[15] 모델이 사용되기도 하며, Unsegmented Data가 많은 OCR의 특성에 맞추어 CTC[6]를 활용하기도 한다.

본 연구에서 사용한 모델은 크게 네 가지 모듈로 구성되었는데, 글자 영역을 인식해 이미지를 표준화할 수 있는 Transformation 모듈, 이미지에 대한 정보를 수집하는 Feature Extraction 모듈, 수집된 정보로부터 글자의 순서 정보를 활용하는 Sequence Modeling 모듈, 마지막으로 이러한 계산값을 바탕으로 글자를 예측하는 Prediction 모듈로 구성하였다. 위 모듈들은 네이버 Clova AI 팀에서 벤치마크하기 위해 사용했던 모듈 구조를 바탕으로 하였다. [6]

2. 본 론

1) 데이터 소개 및 EDA

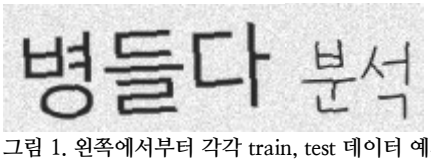

그림 1. 왼쪽에서부터 각각 train, test 데이터 예시

데이터는 폰트 손글씨 학습 이미지 데이터 76,888개, 폰트 손글씨 평가 이미지 데이터 74,121개와 샘플 고유 id, 샘플 이미지 파일 경로, 샘플 이미지에 해당하는 텍스트 레이블값이 포함된 train csv파일, 그리고 샘플 고유 id, 샘플 이미지 파일 경로가 포함된 test csv파일로 이루어져 있다.

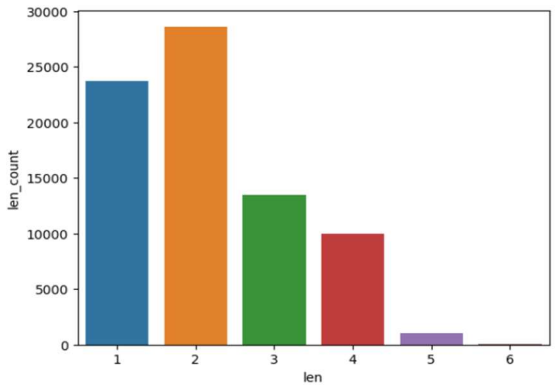

그림 2. 글자 길이당 개수를 나타내는 그래프. 글자 길이가 길어질수록 대체로 개수가 적어지는 불균형을 확인할 수 있다.

학습 데이터의 1글자 샘플의 경우, 1글자 단위의 모든 글자가 데이터에 존재하였다. 또한 모든 레이블에 해당하는 텍스트에는 '공백'이 존재하지 않았다.
하지만 글자 길이가 2인 데이터의 개수는 28,631개인 반면 6인 데이터의 개수는 26으로 글자 길이의 불균형이 존재하였다. 글자 길이가 3 이상인 데이터부터 표본의 수가 적은 만큼, 이러한 3글자 이상의 데이터를 예측하는 데에 어려움이 있을 것이라고 예상했다. 그래서 두 가지 글자를 하나의 행으로 합쳐 글자 수를 늘리는 augmentation 기법인 Cut-Mix 기법을 사용하였다.

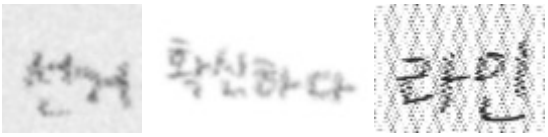

그림 3. 불규칙한 이미지 데이터 예시

또한, 데이터는 앞선 그림1에서 제시한 올곧은 글자 이미지만 존재하지 않고, 그림3과 같이 불규칙한 이미지가 존재하였다. 순서대로 대조, 밝기, 픽셀 및 잡음 등의 불규칙성이 추가된 형식으로, 예시와 같은 이미지들을 보다 잘 예측할 수 있도록 augmentation 기법을 추가하여 적용하였다.

2) 데이터 augmentation

데이터 augmentation은 Cut-Mix와 STRaug[8] 방법을 적용하였다.

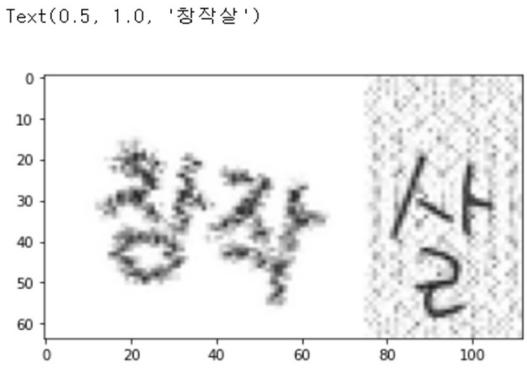

그림 4. Cut-Mix 기법

Cut-Mix 기법은 앞선 설명처럼 두가지 이미지를 횡축으로 합쳐 글자 길이가 긴 데이터를 증식시키는 augmentation 기법이다. 예를 들어 '창작' 이라는 단어가 적힌 이미지와 '살' 이라는 단어가 적힌 이미지가 있을 때, 이 두 이미지를 다음 그림과 같이 옆으로 이어붙여 '창작살' 이라는 3글자로 된 가상의 단어를 만들어 사용한다.
이와같은 Cut-Mix 기법은 한자리 또는 두자리로 된 단어의 이미지가 많은 불균형한 데이터셋에서 세자리 이상의 길이를 가진 단어도 잘 예측할 수 있도록 하기 위해 사용하였다.

STRaug는 8가지 범주로 나누어진 36가지의 STR 특화 augmentation 모음이다. 이 중 GaussianNoise, Contrast, Brightness, Pixelate, GaussianBlur 기법을 적용하였다.

그림 5. 왼쪽 위부터 시계방향으로 각각 원본 그림, GaussianNoise, Brightness, GaussianBlur, Pixelate, Contrast 를 적용한 모습[8]

GaussianNoise는 이미지에 노이즈를 적용한 것이고 GaussianBlur는 카메라의 불완전한 센서나 오염된 렌즈 때문에 발생하는 현상을 표현한다. Brightness와 Pixelate, Contrast 또한 카메라 센서의 불완전성에 의한 현상을 적용한 것으로 각각 밝게 하기, 픽셀화, 대조를 표현하였다.

3) 모델링

모델은 크게 4가지 모듈로 구성되어 있다. 첫 번째 모듈은 Transformation 모듈로 일종의 전처리 모듈인데, 불규칙한 텍스트 이미지에서 텍스트 영역만 인식해 반듯한 이미지로 변형시켜주는 모듈이다. 두 번째는 Feature Extraction 모듈로 가장 메인이 되는 모듈인데, 이름 그대

로 입력된 이미지에서 특징이 되는 Feature를 추출해 연산하는 모듈이다. 세번째로는 Sequence Modeling 모듈인데, Feature Extraction 영역에서 인식된 Feature 값을 바탕으로 Sequence 정보를 이용해 이전 글자를 바탕으로 다음 글자를, 다음 글자를 바탕으로 이전 글자를 예측하는 모듈이다. 마지막은 최종적으로 예측을 해주는 Prediction 모듈이다.

위 모듈 구조는 네이버 Clova AI 팀에서 여러 모듈을 벤치마크하기 위해 사용했던 모듈 구조[6]를 바탕으로 하였다. Transformation 모듈로는 없음(None)과 TPS [9] 모델이, Feature Extraction 모듈로는 VGG[10], RCNN[11], ResNet[12] 모델이, Sequence Modeling 모듈은 없음(None)과 BiLSTM[15] 모델이, Prediction 모듈은 CTC[16]와 Attn[13, 14] 모듈의 선택지가 존재하였다.
본 연구에서는 각 모듈의 선택 가능한 조합별로 가장 성능이 좋은 모듈조합을 확인하고자 하였고, 그 조합은 바로 TPS-VGG-BiLSTM-CTC 모델이었다.

3-1) Transformation : TPS

현장에서 수집한 텍스트 이미지 데이터는 카메라 각도, 촬영자의 위치 등의 문제로 불규칙한 텍스트(irregular text) 데이터가 많이 존재한다. 하지만 일반적인 OCR 모델은 글씨가 올곧게 되어있는 규칙적 문자(regular text)를 입력했을 때 가장 잘 작동하기 때문에, 텍스트 이미지를 모델에게 보다 더 잘 인식할 수 있도록 하기 위해 변환을 진행하였다.

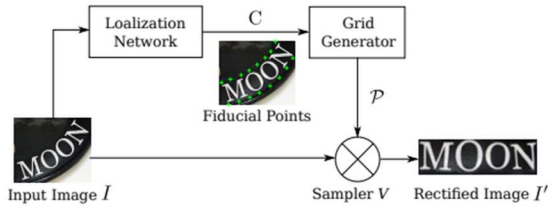

그림 6. TPS 모델의 구조[9]

TPS 모델의 구조는 다음과 같다. 먼저 원본 이미지를 입력하면 Localization Network를 통과해 글씨 영역을 인식하여 $C = [c_1, \dots, c_K] \in R^{2\times K}$, $c_k = [x_k, y_k]^T$로 구성된 기준점의 좌표를 계산한다. 이 Localization Network는 풀링층과 완전연결층으로 구성된 컨볼루션 신경망이다.

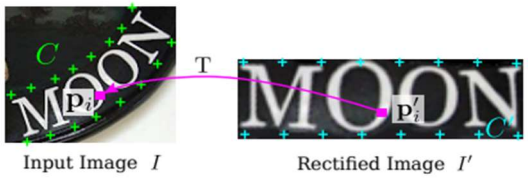

그림 7. Grid Generator[9]

그 후 Grid Generator를 통과하여 TPS Transformation parameter를 추정한 뒤 Sampling Grid를 생성한다. Sampling Grid는 원본 이미지에 있는 글씨를 올곧은 글씨로 변환할 때 활용하기 위해 사용하는데, 기존 이미지의 글씨 영역에서 각 픽셀이 어느 위치로 수정되어야 하는지를 나타내는 좌표 행렬이다.

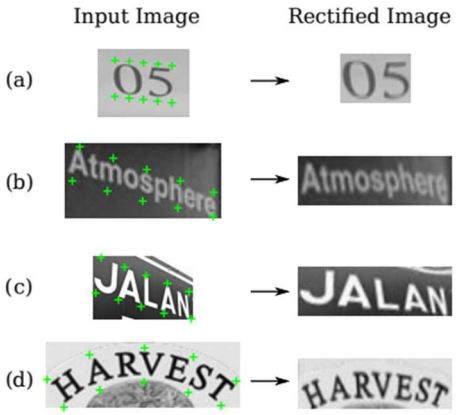

그림 8. (a) 배경이 많은 텍스트(loosely-bounded text), (b) 카메라를 수평으로 두지 않은 텍스트(multi-oriented text), (c) 정면에서 찍지 않은 텍스트(perspective text), (d) 구부러진 텍스트(Curved text) 등에 대해 TPS Transformation을 적용한 예시 [9]

최종적으로 양선형(bilinear) 함수인 Sampler를 활용해, 원본 이미지의 기준점 좌표와 수정되어야 할 각 픽셀의 좌표(Sampling Grid) 값을 바탕으로 데이터의 변환이 진행된다. 또한 앞선 사진으로 보여주었던 구부러진 텍스트(Curved text)뿐만 아니라 다른 여러 가지 원인에 의해 발생한 불규칙 텍스트(irregular text)에 대해서도 TPS Transformation은 유용한 성능을 발휘할 수 있다.

3-2) Feature Extraction : VGG

VGGNet(이하 VGG)은 옥스포드 대학의 연구팀 VGG에 의해 개발된 모델로, 2014년 ImageNet Challenge에서 준우승을 거두며 컴퓨터비전 분야의 대표적인 모델로 자리매김한 모델 중 하나다.
VGG는 당시 대다수의 컨볼루션 신경망 구조와는 다르게 신경망의 깊이에 좀 더 초점을 두었는데, 기존 모델보다 더 깊은 신경망을 구성할 수 있었던 이유는 아주 작은 컨볼루션 필터(3 x 3 사이즈, 1 스트라이드)를 사용했기 때문이다.

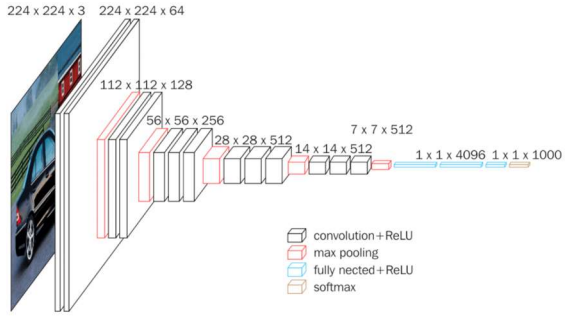

그림 9. VGG-16 모델의 구조[20]

다음은 VGG 모델의 구조이다. 먼저 입력 이미지는 224 x 224 RGB 이미지로 통일하였다. 데이터가 입력되면 3x3 크기의 1 스트라이드 필터를 사용한 컨볼루션 신경망 스택을 통과하고, 2x2 크기를 가진 2 스트라이드의 최대풀링층(MaxPooling Layer)을 통과하며 마지막엔 3개의 완전연결층으로 연결되는 구조를 택하였다.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

그림 10. 신경망 층의 개수에 따른 VGG 구조[10]

흔히 VGG-16, VGG-19라고 부를 때 뒤에 붙이는 숫자는 신경망 층의 개수를 의미한다. 11개의 층으로 구성되면 VGG-11, 13개의 층으로 구성되면 VGG-13인 것이다. 본 연구에선 VGG-13 모델을 사용하였다.

또한 VGG는 다른 컨볼루션 신경망 구조와는 다르게, 3x3 사이즈의 아주 작은 필터를 사용했다. 3x3 필터 3개를 중첩하면 7x7 필터 하나를 사용하는 것에 대응하는데, 7x7 필터 하나가 아니라 3x3 필터를 중첩한 이유는 두 가지가 있다. 첫째는 하나의 필터보단 여러 겹의 필터를 사용하면 결정함수의 비선형성이 더욱 증가하여 Feature의 식별성이 증가하기 때문이다. 다른 하나는 7x7 하나의 필터보다 3x3 3겹의 필터의 파라미터의 수가 더 적기 때문이다. 하나의 7x7 필터는 $49C^2$ 만큼의 파라미터를 가지는데(C는 클래스 개수), 3겹의 3x3 필터는 $27C^2$ 만큼의 파라미터를 가진다. 파라미터의 수가 줄어들었기 때문에 보다 더 효율적인 학습 및 적용을 할 수 있었다.

3-3) Sequence Modeling : BiLSTM

BiLSTM 모델, 즉 양방향 LSTM은 기존 LSTM의 한계인 앞의 문장의 정보만 참고하는 것을 보완하기 위해 이용하는 모델이다. LSTM (Long Short-Term Memory: 장단기 메모리)의 원리를 간단히 살펴보면 다음과 같다.

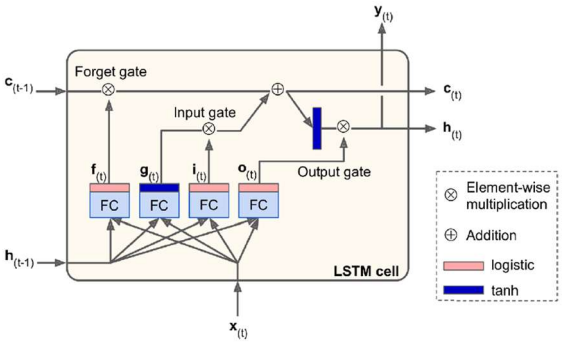

그림 11. LSTM cell의 구조[17]

LSTM 셀은 중요한 입력을 인식하고 (입력 게이트의 역할), 자기 상태에 저장하고, 필요한 기간 동안 이를 보존하고 (삭제 게이트의 역할), 필요할 때마다 이를 추출하기 위해 학습한다. 위 그림에서 $h_{(t)}, c_{(t)}, x_{(t)}, y_{(t)}$는 각각 단기 상태 (short-term state), 장기 상태 (long-term state), 현재 입력 벡터, 현재 출력 벡터이다.

$c_{(t-1)}$는 네트워크를 왼쪽에서 오른쪽으로 관통하면서, 삭제 게이트(Forget gate)를 지나 일부 기억을 잃고, 덧셈 연산으로 새로운 기억 일부(입력 게이트(Input gate)에서 선택한 기억)를 추가한다. 만들어진 $c_{(t)}$는 다른 추가 변환 없이 바로 출력으로 보내진다. 덧셈 연산 후 이 장기(long-term) 상태가 복사되어 tanh 함수로 전달되고, 이 결과는 출력 게이트(Output gate)에 의해 걸러진다. 이는 $h_{(t)}$를 만든다. ($y_{(t)}$와 동일하다.)
$x_{(t)}$와 $h_{(t-1)}$는 네 개의 다른 완전 연결 층에 주입된다. 이 층들은 주 층 $g_{(t)}$와 세 개의 게이트 제어기(gate controller) 로 구성된다. 게이트 제어기는 로지스틱 활성화 함수를 사용하기 때문에 출력이 원소별 곱셈 연산으로 주입되어 0을 출력하면 게이트를 닫고 1을 출력하면 게이트를 여는 식으로 제어한다. $f_{(t)}, i_{(t)}, o_{(t)}$는 각각 삭제 게이트, 입력 게이트, 출력 게이트를 제어한다.

양방향 LSTM(이하 BiLSTM)은 기존의 LSTM 계층에 역방향으로 처리하는 LSTM 계층을 추가한다. 최종 hidden state는 두 LSTM 계층의 hidden state를 연결한 벡터를 출력한다.

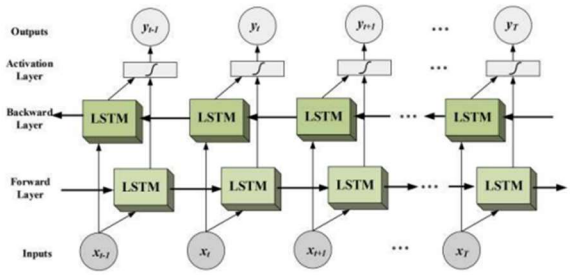

그림 12. BiLSTM의 구조[18]

BiLSTM은 두 개의 LSTM 계층을 사용하여 계층의 단어 순서를 조정한다. 첫 번째 LSTM 계층은 기존과 동일하게 입력 문장을 왼쪽에서 오른쪽으로 처리한다. 추가된 두 번째 LSTM 계층은 입력 문장의 단어 순서를 반대로 처리한다. 그리고 두 LSTM 계층의 출력을 연결하여 하나의 출력을 만든다. 이와 같은 방법으로 BiLSTM 모듈은 모델이 이전의 정보뿐만 아니라 이후의 정보 또한 이용할 수 있게 해준다.

3-4) Prediction : CTC

CRNN은 Step마다 완전연결층의 logit을 Softmax 함수에 넣어줌으로써 어떤 문자일 확률이 높은지 판단한다. 그러나 다음 그림과 같이 모델 출력의 Sequence가 이미지에 걸쳐지는 문제로 기대한 것과 다른 결과가 나올 때가 있다.

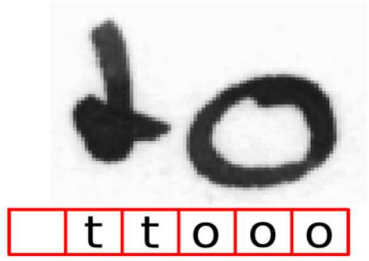

그림 13. 여러 horizontal position에 걸쳐진 이미지 [19]

CTC(Connectionist Temporal Classification)는 그림 12와 같은 Unsegmented data(분리가 어려워 segmentation이 되어있지 않은 데이터) 가 Input과 Output이 서로 다른 길이의 Sequence를 가질 때 활용하는 기법이다.

CTC는 Encoding, Loss calculation, Decoding의 세 가지 과정을 거친다. Encoding 과정에서는 중복 문자 문제를 해결하기 위해 '-'로 표시하는 blank를 사용한다. 위의 경우는 't-o-'가 된다.

Loss calculation 과정에서는 출력 값의 모든 경우의 수에 대한 가능성을 계산하고 음의 로그 값을 취해 loss값을 구한다.

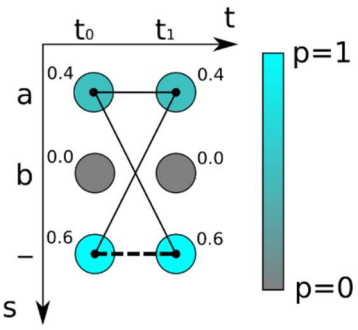

그림 14. Loss calculation 과정 설명을 위한 타임 스텝과 문자에 따른 뉴럴 네트워크의 출력 행렬[19]

그림 14와 같이 두 번의 타임 스텝(t_0, t_1)과 3개의 문자('a', 'b', '-')가 존재한다고 가정하자. 뉴럴 네트워크는 매 타임 스텝마다 각각의 문자의 점수를 출력하였을 것이다. 타임 스텝마다 문자 점수들의 합은 1이다. 하나의 path에 대한 점수는 그 길에 일치하는 문자 점수들의 곱으로 계산된다. 예를 들어 그림 14에서 'aa'라는 path에 대한 점수는 0.4*0.4 = 0.16이다. 반면, '-a'에 대한 점수는 0.6*0.4 = 0.24이다. 정답 문자에 대한 점수를 구하기 위해서는 이 문자와 일치하는 모든 path에 대한 점수를 합해야 한다. 정답 문자가 'a'라고 가정하면 모든 가능한 path는 'aa', '-a', 'a-'이다. 각각의 점수를 계산하여 합하면 0.64임을 알 수 있다. 만약 정답 문자가 ''라고 가정하면 오직 하나의 path만 ('--')존재할 것이고, 그 path의 점수는 0.36이다. 이 점수에 음의 로그 값을 취하면 loss 값이 된다.

Decoding은 타임 스텝마다 가장 가능성이 높은 문자를 채택함으로써 best path를 계산하고, 그 path에 있는 모든 blank를 지우는 과정으로 진행된다.

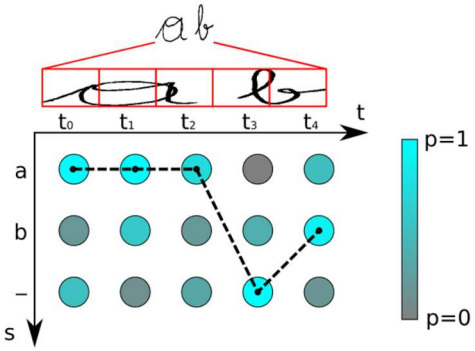

그림 15. Decoding 과정 설명을 위한 타임 스텝과 문자에 따른 뉴럴 네트워크의 출력 행렬[19]

예를 들어, 그림 15에서 각각의 타임 스텝마다 가능성이 높은 문자를 선택하면 best path는 'aaa-b'임을 알 수 있다. 이 경로에서 중복된 문자를 제거하면 'a-b'가 남게 되고 남아있는 경로에서 blank를 제거하면 'ab'이라는 결과를 출력한다.

3. 결 론

표 1. Augmentation을 적용하지 않은 여러 가지 모듈 조합별 성능. 순서대로 ① TPS-VGG-BiLSTM-CTC ② TPS-RCNN-BiLSTM-CTC ③ TPS-ResNet-BiLSTM-CTC ④ TPS-ResNet-None-CTC ⑤ None-VGG-BiLSTM-CTC

	Valid accuracy	Valid Loss	Test Accuracy
①	0.9697	0.0988	0.8186
②	0.9775	0.0741	0.8198
③	0.9804	0.0713	0.7962
④	0.9799	0.0783	0.6159
⑤	0.9239	0.2732	0.7601

어그멘테이션을 진행한 데이터로 모듈 조합을 테스트하려면 학습 시간의 문제로 많은 조합을 테스트하기에 한계가 있었다. 그래서 기본 데이터로 여러 가지 모듈을 테스트한 후 최적의 성능을 나타낸 2가지 모델을 선정했다. 바로 ① TPS-VGG-BiLSTM-CTC 모델과 ② TPS-RCNN-BiLSTM-CTC 모델이었다.

이후 위 두 모델에 대해 원본 훈련셋의 약 3배가량 데이터를 증식시킨 데이터셋으로 훈련을 진행했고, 최종적으로 ① TPS-VGG-BiLSTM-CTC 모델이 0.8555의 Test Accuracy를 기록하며 가장 높은 성능을 기록하였다.

본 연구의 한계점은 다음과 같다. 우선 제한적인 시간과 학습 여건 등의 문제로 조합 가능한 24개의 모듈 조합을 전부 비교하지 못했다는 점이다. 또한 augmentation이 진행된 데이터로 학습을 진행해 비교할 때 학습 시간이 오래 걸리기 때문에 실제 최종 학습에 사용한 데이터가 아닌 기본 데이터로 학습한 결과를 비교하였다. 그 때문에 비교 척도가 다소 달라 최적의 모델을 찾는 방법에 한계가 있었다.

차후 연구에서는 augmentation을 진행한 데이터로 모든 모듈 조합을 비교해보고, 아직 사용해보지 않은 TrOCR 등 다른 OCR 모델을 활용해 성능을 확인해보며 비교할 계획이다. 그리고 이러한 손글씨 인식 OCR 모델을 유아 교육 학습지 등의 사업과 연계한다면 양방향 교육이 중요해진 지금, 차세대 에듀테크의 교두보가 되어 아이들이 훨씬 효율적인 학습을 하도록 도움을 줄 수 있을 것이다.

참고 문헌

[1] Chen-Yu Lee and Simon Osindero. 2016. Recursive recurrent nets with attention modeling for ocr in the wild. In CVPR.

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In CVPR.

[3] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In ICLR.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In CVPR.

[5] Alex Graves, Santiago Fernandez, Faustino Gomez, and ´ Jurgen Schmidhuber. 2006. Connectionist temporal classification: ¨ labelling unsegmented sequence data with recurrent neural networks. In ICML.

[6] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. 2019. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. In ICCV.

[7] "'손글씨' AI 인식 …교육업계, 'OCR'로 똑똑해진다 - 신아일보" 신아일보, 2022년 6월 10일 수정, 2023년 1월 28일 접속, http://www.shinailbo.co.kr/news/articleView.html?idxno=1561408

[8] Rowel Atienza. 2021. Data Augmentation for Scene Text Recognition. In ICCV

[9] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xinag Bai. 2016. Robust Scene Text Recognition with Automatic Rectification. In CVPR.

[10] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR.

[11] Chen-Yu Lee and Simon Osindero. 2016. Recursive recurrent nets with attention modeling for ocr in the wild. In CVPR.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and

Jian Sun. 2016. Deep residual learning for image recognition. In CVPR.

[13] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2016. Robust scene text recognition with automatic rectification. In CVPR.

[14] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. 2017. Focusing attention: Towards accurate text recognition in natural images. In ICCV.

[15] Baoguang Shi, Xiang Bai, and Cong Yao. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. In IEEE

[16] Alex Graves, Santiago Fern ´ andez, Faustino Gomez, and J ¨ urgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In ICML

[17] Aurelien Geron. 2020. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3nd Edition

[18] 고상준, 윤호영 and 신동명. (2018). 양방향 LSTM 을 활용한 전력수요 데이터 예측 기법 연구. 한국소프트웨어감정평가학회 논문지, 14(1), 33-40.

[19]
CTC 그림
https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c

[20]
VGG-16 구조 그림
https://neurohive.io/en/popular-networks/vgg16/