Information Systems Review
Vol. 25, No. 1 February 2023 http://dx.doi.org/10.14329/isr.2023.25.1.129

인코더와 디코더에 기반한 합성곱 신경망과 순환 신경망의 새로운 하이브리드 접근법

New Hybrid Approach of CNN and RNN based on Encoder and Decoder

우 종 우 (Jongwoo Woo) ㈜데이터월드 차장
김 건 우 (Gunwoo Kim) 국립한밭대학교 융합경영학과 교수
최 근 호 (Keunho Choi) 국립한밭대학교 융합경영학과 부교수, 교신저자

요 약

빅데이터 시대를 맞이하여 인공지능 분야는 괄목할만한 성장을 보이고 있으며 특히 딥러닝에 의한 이미지 분류 학습방법이 중요한 영역으로 자리하고 있다. 이미지 분류에서 많이 사용되어 온 CNN의 성능을 더욱 개선하기 위해 다양한 연구가 활발하게 진행되었는데, 이 중에서 대표적인 방법이 CRNN(Convolutional Recurrent Neural Network) 알고리즘이다. CRNN 알고리즘은 이미지 분류를 위한 CNN과 시계열적 요소를 인식하기 위한 RNN의 조합으로 구성되는데, CRNN의 RNN영역에서 사용하는 입력값은 학습 대상의 이미지를 합성곱과 풀링 기법을 적용하여 추출된 결과물을 flatten한 값이고, 이 입력값들은 이미지 내 동일 위상에 있는 픽셀값들이 서로 다른 순서로 나타나기 때문에, RNN에서 의도한 이미지 내 배열 순서를 제대로 학습하기 어렵다는 한계점을 지닌다. 따라서 본 연구는 인코더와 디코더의 개념을 응용한 CNN과 RNN의 새로운 하이브리드 방법을 제안하여, 이미지 분류 성능을 향상시키는 것을 목적으로 하였다. 본 연구에서는 다양한 알고리즘 비교 실험을 통해, 새로운 하이브리드 방법의 효과성을 검증하였다. 본 연구는 인코더와 디코더 개념의 적용 가능성을 넓히고, 제안한 방법이 기존 하이브리드 방법에 비해, 복잡도가 크게 증가하지 않아 모델 학습 시간과 인프라 구축 비용 측면에서 이점을 있다는 점에서 학문적 시사점을 가진다. 또한, 정확한 이미지 분류가 필요한 다양한 분야에서 제공되는 서비스의 품질을 높일 수 있는 가능성을 제시하였다는 점에서 실무적 시사점을 가진다.

키워드 : 딥러닝, RNN, LSTM, BiLSTM, CRNN

Ⅰ. 서 론

인간은 끊임없이 과학기술을 발전시켜 왔다. 그 중에서도 데이터를 활용한 기술은 인공지능 분야에 커다란 진전을 보여 왔고 앞으로 4차 산업혁명 시대를 선도하는 중요한 분야가 될 것이다. 특히 컴퓨터 비전(Computer Vision) 기술의 발전으로 인해 이미지와 영상 영역에서 점증적인 발전을 거

듭하여 현재는 빅데이터와 인공지능이 결합한 형태로 인간의 인식능력 범위까지 확대되고 있다(박경철, 2019; 허인성, 2015).

컴퓨터 비전의 분야 중 하나는 이미지 분류(Image Classification)이다. 이미지 분류란 이미지가 주어졌을 때 해당 이미지가 속하는 클래스를 찾아주는 것으로 카테고리(Class)별로 분류된 이미지를 가지고 모델을 훈련하고 이미지가 어느 카테고리에 속하는지 예측하는 방식이다. 특정한 입력 이미지가 정해져 있는 label 중 어떤 label에 해당하는지 구별하는 분야이다. 즉 이미지 분류는 한 장의 이미지를 알고리즘에 입력해주면, 그 이미지가 어떤 클래스의 label에 속하는지 알려준다. 예를 들면 이미지 인식용 알고리즘이 볼펜, 노트, 가방과 같이 3개 중 하나의 클래스 label을 가진 이미지들로 훈련되었다면 알고리즘은 이미지를 입력 받을 때마다 그 이미지가 세 개 중 어떤 클래스의 label에 속하는가를 분류해낼 수 있다(김윤진, 2017; 박경철, 2019).

합성곱 신경망(Convolution Neural Network: CNN)은 이미지 분류에 특화된 딥러닝 알고리즘으로, 이미지를 분석하여 사전에 정의된 클래스로 분류 및 예측하는 알고리즘이다. 동물들이 다른 대상을 구분할 때, 대상의 전체 모습이 아니라 특정 부분을 민감하게 받아들여 대상을 구분하는 것에서 아이디어를 얻었다. 이러한 뇌의 활동에 힌트를 얻어 CNN이라는 딥러닝 알고리즘이 발표되었고 현재 이미지 분류 분야에서 많이 활용되고 있다(성상하, 2019).

순환 신경망(Recurrent Neural Network: RNN)은 순환신경망의 구조를 통해 시간에 따라 변화하는 데이터의 상호관계를 반영할 수 있다. RNN 알고리즘의 경우 문장의 길이가 길어지면 문장의 상호관계를 잘 반영하지 못하는 장기 의존성 문제가 존재하고, 기울기 소실과 같은 문제를 나타낸다는 단점이 존재한다. RNN의 장기 의존성 문제를 해결하기 위해 RNN의 개선된 구조로 LSTM(Long Short-Term Memory)이 제안되었다. RNN의 경우 모든 입력의 값을 기억하는 hidden state에 반영하였는데, LSTM은 정보의 전달량을 조절하는 게이트(gate)를 추가하여 장기의존성 문제와 기울기 소실 문제를 해결할 수 있게 되었다(이동엽 등, 2017).

CNN의 성능을 더욱 개선하기 위해 다양한 연구가 활발하게 진행되었는데, 이 중에서 대표적인 방법이 CRNN(Convolutional Recurrent Neural Network) 알고리즘이다. CRNN 알고리즘은 이미지 분류를 위한 CNN과 시계열적 요소를 인식하기 위한 RNN의 조합으로 구성된다. 분류 과정의 첫 번째 단계에서는 CNN을 활용하여 입력된 이미지로부터 Feature Sequence를 추출한다. 두 번째 단계에서는 추출한 Feature Sequence들을 RNN 계열의 신경망 알고리즘으로 처리하여 이미지 내 픽셀의 배열 순서를 누적하여 학습하고, 마지막으로 누적된 픽셀의 배열 순서의 학습결과를 바탕으로 이미지를 분류하게 된다(성상하, 2019).

하지만, CRNN의 RNN영역에서 사용하는 입력값은 학습 대상의 이미지를 합성곱과 풀링 기법을 적용하여 추출된 결과물을 flatten한 값이다. 이 입력값들은 이미지 내 동일 위상에 있는 픽셀값들이 서로 다른 순서로 나타나기 때문에, RNN에서 의도한 이미지 내 배열 순서를 제대로 학습하기 어렵다는 한계점을 지닌다. 따라서 본 연구는 기존에 제안된 CRNN 알고리즘의 한계점을 개선함으로써 이미지 분류 성능을 향상시킬 수 있는 새로운 아이디어를 제안하는 것을 목적으로 하였다. 이를 위해, 인코더와 디코더에 기반한 CNN과 RNN의 새로운 하이브리드 방식을 제안하였고, 새로운 접근법의 성능평가를 위해 기존의 CNN 및 CRNN과 비교하는 실험을 진행하였다.

본 논문의 구성은 다음과 같다. 제Ⅱ장에서는 딥러닝, CNN, RNN, CRNN의 선행연구들을 살펴보고 본 연구에서 제안한 CRNN의 개선 방향에 대해 알아본다. 제Ⅲ장에서는 본 연구에서 사용한 데이터와 새롭게 제안하는 접근법을 설명하고, 제Ⅳ장에서는 연구의 실험결과에 대해 살펴본다. 마

지막 제Ⅴ장에서는 본 연구를 요약하고 시사점과 한계점, 그리고 향후 연구 방향을 제시한다.

Ⅱ. 관련 연구

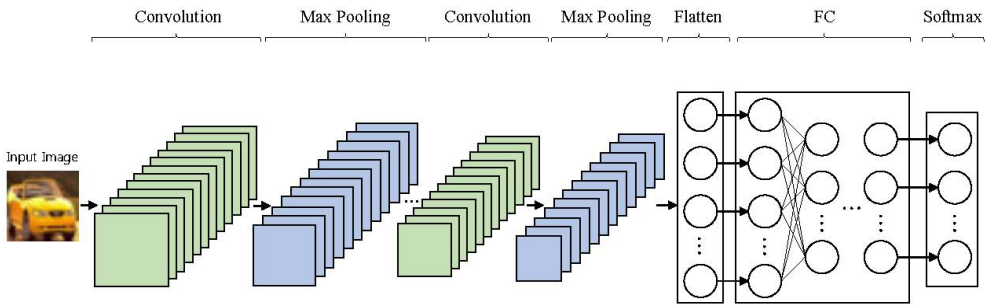
2.1 딥러닝

딥러닝은 신경망이 확장된 개념이다. 딥러닝 알고리즘에는 대표적으로 심층 신경망(Deep Neural Network, DNN)과 합성곱 신경망(Convolution Neural Network, CNN), 순환 신경망(Recurrent Neural Network, RNN) 등이 있다. 딥러닝 알고리즘은 데이터의 특성과 종류, 목표에 따라 사용되는 알고리즘이 달라진다(김윤진, 2017). 딥러닝은 AND, OR, XOR 문제 중에서 XOR가 두 개의 클래스로 구분하는 선을 찾을 수 없다는 점을 해결하기 위해 시작되었다. MIT의 Marvin Minsky 교수는 XOR 문제는 선형 회귀분석으로는 풀 수 없다는 사실을 수학적으로 증명하였고, XOR 문제를 풀기 위해서는 퍼셉트론을 사용한 MultiLayer Perceptron으로 신경망을 구성해야 한다고 제시하였다.

신경망은 분석 데이터로부터 반복적인 학습 과정을 거쳐 패턴을 찾아내고 이를 일반화하여 예측하는 문제에 있어서 유용하게 이용되는 기법으로, 인간이 의사결정을 내릴 때 뇌 속 신경망 사이에서 일어나는 메커니즘을 참고하여 만들어졌으므로 신경망이라는 이름으로 부르게 되었다. 신경망은 크게 입력층(input layer), 은닉층(hidden layer), 그리고 출력층(output layer)으로 구성된다. 각 층은 여러 노드로 구성되는데, 입력층은 독립변수의 값을 받아들이는 역할을 하고, 은닉층은 독립변수의 값을 이용하여 복잡한 수많은 계산을 수행하며, 출력층은 분석의 결과값을 출력해주는 역할을 한다(박경철, 2019; 허인성, 2015).

2.2 CNN

CNN은 이미지 분류에 특화된 딥러닝 알고리즘으로, 이미지를 분석하여 사전에 정의된 클래스로 분류 및 예측하는 작업이다. 이미지는 [넓이×높이×깊이]로 표현되는데, CNN은 이러한 이미지의 모든 정보를 한 번에 입력받아 분석하지 않고, 이미지의 각 부분을 따로 학습하고 이를 종합하여 전체 이미지의 정보를 파악하는 방법이다. CNN은 이와 유사한 원리로 이미지를 분류할 때에도 이미지 전체가 아닌 이미지의 각 부분을 학습하여 종합적으로 분석함으로써 분류의 정확도를 높이고자 한 알고리즘이다. 아래의 <그림 1>에서 CNN에서 수행하는 합성곱 과정을 보면, 32×32×3의 픽셀 크기를 갖는 이미지가 있다. CNN은 이미지의 각 부분을 학습하는 과정에서 filter를 이용한다. 5×5×3의 픽셀 크기를 가진 filter는 각 픽셀마다 임의의 가중치를 포함하고 있는데, 이 filter를 이미지에 씌운 후, 이미지의 각 픽셀에 있는 값과 filter의 대응되는 부분에 있는 가중치를 곱하여 가중합 한 하나의 값을 해당 영역으로부터 도출하게



〈그림 1〉 CNN 아키텍처

된다. 이 과정을 합성곱이라 한다(허인성, 2015).

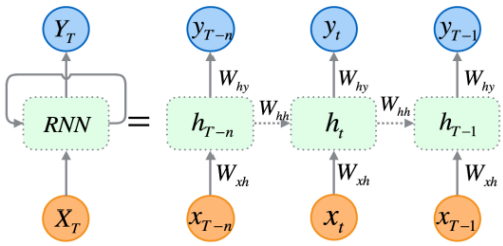
이러한 구조를 반복적으로 수행하며 최종적으로 완전연결계층을 생성하여 분류를 수행한다(허인성, 2015). 채널(컬러, 흑백)에서 3채널인 컬러는 1:1로 합성곱 한 후 3개의 값을 합산한다. 합성곱의 결과로 생성되는 이미지의 크기는 원본 이미지보다 작아진다. 합성곱이 여러 번 진행되어 이미지의 크기는 더욱 작아지게 되는데, 이미지의 크기가 너무 작아지게 되면 유용한 정보의 손실이 발생할 수 있으므로 이미지 크기가 작아지는 것을 방지하고, 이미지의 모서리 부분임을 알려주기 위해 Padding 방법을 사용한다(허인성, 2015).

풀링(Pooling)은 합성곱을 통해 생성된 이미지와 관련해서 과잉 적합을 완화하고, 왜곡된 이미지 보정을 해주며, 일반화 가능성을 높이기 위해 크기를 재조정(resize)하는 작업이다. 가장 대표적인 풀링 방법은 max pooling으로 특정 크기의 filter를 변환하려는 이미지에 씌우고, 해당 영역에서 가장 큰 값 중 하나를 선택하여 해당 영역의 대표값으로 만들어주는 방법이다.

Flattening은 합성곱과 풀링을 여러 차례 반복적으로 수행하면서 최종적으로 여러 개의 결과 이미지를 생성하고, 이 최종 결과 이미지는 flattening 작업을 거쳐 신경망 모델의 입력값으로 사용하기 적합한 형태로 변환된다. Flattening은 여러 이미지 내에 있는 각 픽셀값을 하나의 독립변수로 만들어주는 것으로 예를 들어, 5×5×1 크기의 최종 이미지가 100개 생성되었을 경우, 독립변수의 개수는 5×5×1×100 = 2,500개이다. 독립변수들은 신경망에 전달되어 각 노드와 학습한 후 클래스별로 분류된다(허인성, 2015).

2.3 RNN

RNN은 현재 상태에 대한 분석 결과를 도출하기 위해 현재 시점에서의 x값뿐만 아니라 이전 상태에 관한 정보도 함께 이용한다. 벡터 x0의 값이 입력되면 이를 이용하여 신경망에서와 같은 학습이 이루어지게 된다. 다음은 두 번째 벡터 x1의 값이 입력되면 x1값뿐만 아니라 이전 단계에서 학습된 분석값(hidden state)도 함께 이용하여 학습이 이루어지게 된다(이동엽 등, 2017). RNN은 새로운 x벡터가 들어왔을 때 현 x벡터의 값만 이용하여 예측값을 추정하는 것이 아니라 이전 x벡터들의 결과값을 함께 분석에 활용한다는 점에서 일반적인 신경망이나 CNN 모델과는 다른 특징을 갖는다. 경사하강법과 chain rule을 이용한 역전파 방식으로 가중치를 업데이트한다. 하지만 RNN은 입력되는 x값이 많은 경우엔 기울기 소실(Gradient vanishing)이 일어난다. 기울기 소실은 모든 기울기가 1보다 작은 경우에 발생하며, 이 경우 가중치의 업데이트가 잘 이루어지지 않는다. 이것에 대한 해결방법으로 LSTM 알고리즘을 사용하고 있다(이동엽, 2017).

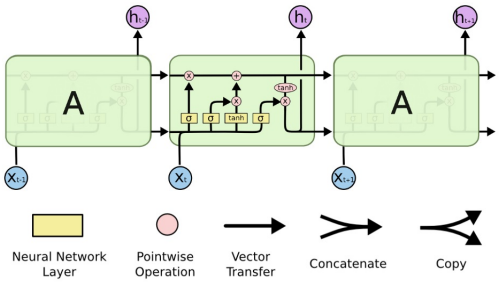


〈그림 2〉 RNN 아키텍처

2.3.1 LSTM

LSTM은 입력되는 x값의 시퀀스(sequence)가 길어질 경우 발생하게 되는 기울기 소실 문제에 대한 해결책으로 제안된 알고리즘이다. LSTM은 RNN 알고리즘에 메모리 셀이라는 개념을 추가하였다(이동엽 등, 2017). RNN은 현 상태에 대한 학습 시 이전 단계에서 학습된 결과를 함께 반영하게 되는데, 너무 오래된 학습결과까지 반영하게 되면 잘못된 예측값을 줄 수 있고, 입력되는 x에 따라 이전 단계에서 학습된 결과를 반영하지 않는 것이 더 좋은 결과를 보여줄 때도 있다. 따라서 LSTM은 메모리 셀을 이용하여 현 단계에서의 모

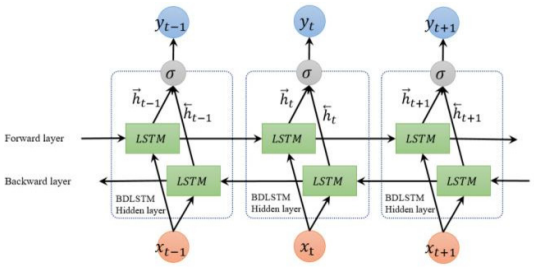
델 학습 시 이전 단계에서 학습된 결과를 반영하는 양을 조절하여 위에서 언급된 문제들을 해결하고자 하였다. 결국, 메모리 셀은 과거의 경험을 얼마만큼 반영할지를 결정하는 값이라고 할 수 있다(이동엽 등, 2017).



〈그림 3〉 LSTM 아키텍처

2.3.2 BiLSTM

RNN이나 LSTM은 입력값을 시간 순서대로 입력하기 때문에 결과물이 직전 패턴을 기반으로 수렴하는 경향을 보인다는 한계가 있다. 이 단점을 해결하는 목적으로 양방향 순환신경망(Bi-RNN)이 제안되었다(이동엽 등, 2017). Bi-RNN은 기존의 순방향에 역방향을 은닉층에 추가하여 성능을 향상 시켰다. 그러나 데이터 길이가 길고 층이 깊으면, 과거의 정보가 손실되는 단점이 있다. 이를 극복하기 위해 제안된 알고리즘이 양방향 LSTM이다. 양방향 LSTM은 최근 머신러닝 분야에서 좋은 성과에 적용된 모델일 정도로 높은 성능의 알고리즘 중 하나이다. 출력값에 대한 손실을 최소화하는 과정에서 모든 파라미터가 동시에 학습되는 종단 간 학습이 가능하다. 단어와 구(Phrase)간 유사성을 입력 벡터에 내재화하여 성능을 개선하고, 데이터 길이가 길어도 성능이 저하되지 않는 것이 LSTM의 기본 성능이다. 2개의 LSTM 계층을 사용하면 계층의 단어 순서를 조정할 수 있다. 첫 번째 LSTM 계층은 기존과 동일하게 입력 문장을 왼쪽에서 오른쪽으로 처리를 한다. 추가된 두 번째 LSTM 계층은 입력 문장의 단어 순서를 반대로 처리한다. 예를 들면 A-B-C의 순서로 처리되는 과정이 C-B-A의 역순서로 처리된다. 단계마다 두 모델에서 나온 2개의 Hidden Vector는 학습된 가중치를 통해 하나의 Hidden Vector로 만들어지게 된다(성상하, 2019).



〈그림 4〉 BiLSTM 아키텍처

2.4 CNN-RNN 결합 연구

CNN과 RNN의 결합을 시도한 선행연구들을 살펴보면, CNN과 RNN을 순차적으로 결합한 연구와 병렬적으로 결합한 연구들로 분류할 수 있다.

먼저, CNN과 RNN을 순차적으로 결합한 연구들을 보면, 이영욱 등(2022)은 CNN과 RNN(LSTM)을 결합하여 낙상을 감지하는 시스템 모델을 구현하였다. 적외선 열상 카메라로 수집한 비디오 데이터 셋을 이용하였으며, CNN의 출력값을 LSTM의 입력값으로 사용하였다. 실험을 통해, CNN-LSTM과 ResNetCNN-LSTM 모델을 비교하였는데, CNN-LSTM 모델이 더 좋은 성능을 보였다. 박호연, 김경재(2019)는 CNN과 LSTM을 결합한 모델을 이용하여 감성분석의 분류 정확도를 개선하고자 하였다. 영화 리뷰 데이터 셋인 IMDB의 리뷰 데이터 셋을 이용하였는데, CNN과 LSTM을 직접 순차적으로 연결하였으며, 비교 실험 결과 각각의 기법보다 두 기법을 결합한 모델의 성능이 더 좋게 나타났다. 홍창우, 허견(2021)은 실제 선박의 항해 시뮬레이션 데이터를 이용하여, 선박의 전력 부하를 예측하는 모델을 개발하였다. CNN의 출력값을 RNN의 입력값으로 사용하는 결합방법을 이용하였는데, 다른

기법들과의 성능 비교는 하지 않고, 제안한 모델의 레이어 깊이에 따른 성능 비교만 진행하였다. 임근영, 조영복(2019)은 Microsoft Malware Classification Challenge에서 제공하는 데이터 셋을 이용해 임의의 길이 입력 데이터에 적용할 수 있는 멀웨어 분류 모델을 제안하였다. 이 연구에서는 멀웨어 데이터를 이미지화 시킨 후, 생성된 이미지를 RNN으로 학습하고 그 출력값을 CNN으로 다시 학습시켜 최종적으로 멀웨어를 분류하는 모델을 개발하였는데, 다른 기법과의 비교 실험은 진행하지 않았지만, 96%의 정확도 성능을 보였다. 다만, 타겟 클래스 간 데이터 건수가 달라 데이터 건수가 적은 클래스의 경우 정확도가 낮게 나타나는 모습을 보였다.

다음으로, CNN과 RNN을 병렬적으로 결합한 연구를 보면, 이정민, 이현(2022)은 단백질 서열을 이용한 기능과 구조 예측 분야에서 CNN과 RNN을 결합한 효소 기능 예측 모델을 설계하였다. 이 연구에서는 입력 데이터를 임베딩한 후, CNN, LSTM, GRU에 각각 입력하고, 각 알고리즘에서 나온 출력값들을 concatenation하여 완전 연결층에 입력함으로써 예측을 수행하였다. 비교 실험 결과, 선행연구에서 제안한 모델 보다 대부분의 실험에서 좋은 성능을 보였다.

2.5 CRNN

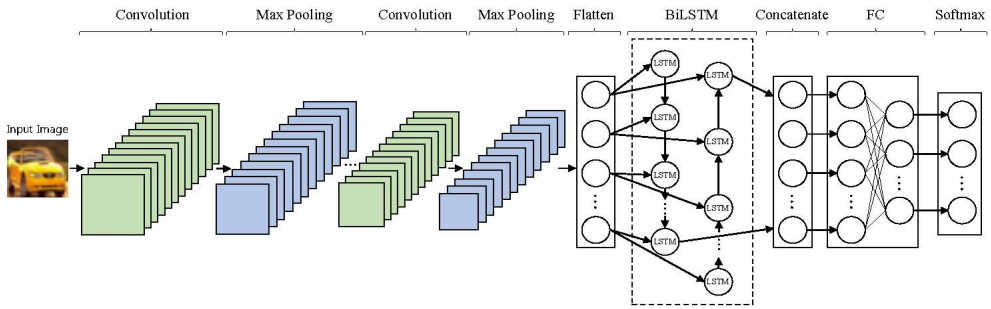
CRNN은 앞 절에서 언급한 CNN-RNN 결합 방법 중 많은 연구에서 사용되어 온 CNN과 RNN을 순차적으로 결합한 방법의 하나로, CRNN은 이미지 내 특징 추출을 위한 CNN과 시계열적 요소를 인식하기 위한 RNN의 조합으로 구성된다. CRNN은 다양한 크기의 이미지를 입력 받아 특징을 추출한 뒤, RNN을 활용해 예측된 분류값을 출력한다(성상하, 2019). 즉, <그림 5>와 같이, CRNN은 CNN과 RNN(LSTM)을 결합한 모형으로서 이미지 기반 시계열 인식 모델이라고 할 수 있으며, 특수한 시계열을 해결하기 위해서 고안되었다(성상하, 2019).

2.6 본 연구에서 제안한 CRNN 개선 방향

본 논문에서 제안하고 있는 접근법 ECRNN (Encoder-based CRNN)은 기존에 CNN의 성능을 개선하기 위해 제안되어 사용되고 있는 CRNN 알고리즘과 비교하면 다음과 같은 점에서 차별점을 갖는다.

먼저, CRNN에서는 앞서 언급한 바와 같이, 합성곱과 풀링을 적용시켜 나온 결과물을 flatten 한 후, 이를 RNN의 입력값으로 사용한다.

이 경우 flatten 된 값들은 학습 이미지가 각 필터들과 학습되어 추출된 특징이 위치한 픽셀값들이 특징의 위치값이 아닌 서로 다른 순서로 나타난다. 때문에, RNN을 통해 이미지 내 각 픽셀의 순서가 제대로 학습되지 않게 된다. 또한, flatten 한 값을 그대로 입력값으로 사용하고 있으므로 RNN에 입력되는 픽셀값의 순서에 대한 학습은



〈그림 5〉 CRNN 아키텍처

이루어지지 않는다(Shi *et al.*, 2010).

반면에, 본 연구에서 제안한 ECRNN은 합성곱과 풀링을 적용시켜 나온 결과물을 flatten 한 후, 이 flatten 된 값을 RNN의 입력값으로 바로 넣어주지 않는다. 대신 이 값들을 인코더에 통과시켜 인코딩된 잠재변수로 차원을 축소시킨 후, 이 잠재변수를 다시 디코더에 통과시켜 원래의 입력값과 동일한 차원의 출력값을 생성한다. 이후 이 디코더의 출력값을 RNN의 입력값으로 넣어주게 된다. 이러한 변환 과정은 분류 모델을 학습할 때 RNN에 입력되는 픽셀값의 순서까지 최적화하여 학습할 수 있게 함으로써, 최적의 입력 순서를 찾게 해준다. 이를 통해, 이미지 내 픽셀값들이 고정된 순서로, 그리고 동일한 위상에 있는 픽셀값들이 서로 다른 순서로 RNN에 입력됨으로써 나타날 수 있는 CRNN 모델의 한계점을 개선하여, 더욱 향상된 성능의 모델을 제안하고 이를 이미지 분류 학습에 활용할 수 있도록 하였다.

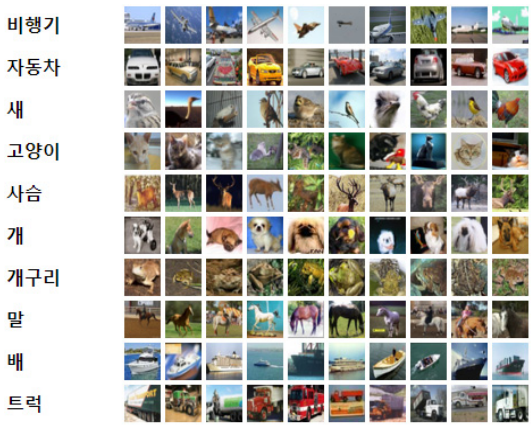
Ⅲ. 연구방법

3.1 데이터 및 전처리

본 논문에서 실험을 위해 사용한 데이터셋은 이미지 데이터로 구성된 CIFAR-10 데이터셋으로, 이 데이터셋은 CNN 학습 알고리즘으로 사용되고 있는 AlexNet을 만든 Alex Krizhevsky가 제공하는 오픈 데이터셋이다. 본 데이터셋은 <그림 6>과 같이, 10개의 클래스로 구성된 컬러 이미지 데이터를 포함하고 있으며, 각 이미지 데이터는 32×32의 해상도 크기를 갖는다(박경철, 2019).

본 연구는 본 연구에서 제안한 방법론의 성능에 대한 신뢰도를 높이기 위해 다음과 같은 작업을 수행하였다.

첫째, 기존 모델과 본 연구에서 제안한 방법론을 이용한 모델을 개발할 때 방법론마다 3회씩 실험을 반복 수행하였다.



〈그림 6〉 CIFAR-10 데이터셋의 클래스와 10개의 이미지

둘째, 매 실험마다 모델의 성능을 검증하기 위한 검증 데이터셋이 달라질 수 있도록 하였다. 이를 위해, CIFAR-10에서 제공하는 고정된 검증 데이터셋을 이용하지 않고, CIFAR-10에서 제공하는 학습 데이터 50,000개를 실험마다 7:3의 비율로 랜덤하게 분할하여 학습 데이터셋과 검증 데이터셋으로 구성하였다(박경철, 2019).

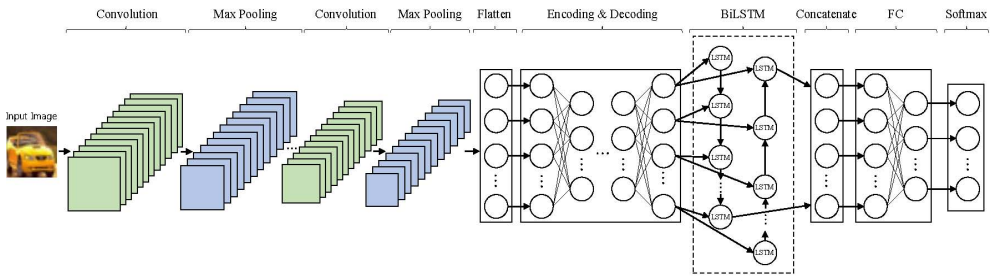
또한, 본 논문에 사용된 학습 데이터들은 데이터 전체의 평균을 빼주고 표준편차로 나누어 주는 표준화를 진행한 후 분석에 사용하였다.

본 논문에서 수행한 데이터에 대한 모든 전처리 작업과 모델 개발은 python을 이용하여 진행하였다.

3.2 ECRNN

본 연구에서 제안한 방법론은 <그림 7>와 같다. CRNN 알고리즘 개선을 위해 인코더와 디코더에 기반한 새로운 하이브리드 방식의 CRNN인 ECRNN을 제안하였다. ECRNN은 크게 CNN 영역, 인코더 영역, 그리고 RNN 영역의 3개 영역으로 구성된다. 대상 이미지가 학습되는 과정은 다음과 같다.

먼저, CNN 영역의 첫 번째 단계에서는 이미지에 대해 합성곱과 풀링 작업을 반복적으로 수행함으로써 차원 축소와 정규화 등을 거친 이미지의



〈그림 7〉 본 연구에서 제안한 ECRNN 아키텍처

특징을 추출한다. 두 번째 단계에서는 앞에서 학습된 3차원 데이터를 flatten 하여 이미지 데이터를 1차원으로 펼치게 된다.

다음으로, 인코더 영역에서는 앞의 CNN 영역의 출력값인 1차원의 flatten 된 이미지 데이터를 입력값으로 받아, 인코딩을 수행하여 차원이 축소된 잠재변수를 생성하고, 이 잠재변수를 다시 디코딩하여 인코더의 입력값과 동일한 차원의 출력

〈표 1〉 ECRNN의 네트워크 구조(BiLSTM: 1-layer, 16 nodes)

번호	영역	Layer	Output	Param	Kernel	Stride	Padding
1	CNN 영역	Input Layer	32, 32, 3	0	-	-	-
2		Conv2D	32, 32, 32	896	3 x 3	1	same
3		Batch Normalization	32, 32, 32	128	-	-	-
4		Max Pooling	16, 16, 32	0	2 x 2	2	same
5		Conv2D_1	16, 16, 64	18,496	3 x 3	1	same
6		Batch Normalization_1	16, 16, 64	256	-	-	-
7		Max Pooling_1	8, 8, 64	0	2 x 2	2	same
8		Conv2D_2	8, 8, 128	73,856	3 x 3	1	same
9		Batch Normalization_2	8, 8, 128	512	-	-	-
10		Max Pooling_2	4, 4, 128	0	2 x 2	2	same
11		Conv2D_2	4, 4, 256	295,168	3 x 3	1	same
12		Batch Normalization_3	4, 4, 256	1,024	-	-	-
13		Max Pooling_3	2, 2, 256	0	2 x 2	2	same
14		Flatten	1,024	0	-	-	-
15	인코더 영역	Dense	512	524,800	-	-	-
16		Batch Normalization_4	512	2,048	-	-	-
17		Dense_1	256	131,328	-	-	-
18		Batch Normalization_5	256	1,024	-	-	-
19		Dense_2	512	131,584	-	-	-
20		Batch Normalization_6	512	2,048	-	-	-
21		Dense_3	1,024	525,312	-	-	-
22		Batch Normalization_7	1,024	4,096	-	-	-
23		Reshape	1,024, 1	0	-	-	-
24	RNN 영역	BiLSTM	32	2,304	-	-	-
25		Dense_4	10	330	-	-	-
26		Softmax	10	0	-	-	-
27	Total(전체 사용된 파라미터 수): 1,715,210						

값을 생성하게 된다.

마지막으로, RNN 영역에서는 인코더 영역의 출력값을 입력값으로 받아들이고, LSTM 또는 BiLSTM의 알고리즘 특성을 사용하여 시계열 학습을 수행한 후, 최종적으로 이미지를 분류하게 된다.

본 논문에서 제시한 ECRNN 알고리즘을 학습이 진행되는 layer 별로 정리하면 <표 1>과 같다.

Ⅳ. 실험결과

본 논문은 기존 이미지 학습에 많이 사용되고 있는 CNN 그리고 CNN과 RNN을 연결한 이미지 분류에 사용되는 CRNN, 또 이번 연구에서 새롭게 제안한 ECRNN 등 3가지 알고리즘을 각각 사용하여 구성된 모델을 학습시키고 그 결과를 다음과 같이 비교 분석하였다.

첫째, CNN과 CRNN 알고리즘을 사용한 모델의 정확도(accuracy)를 비교하는 실험을 통해, CNN 대비 CRNN의 성능 개선 효과를 분석하였다. 둘째, CNN과 새롭게 제안한 ECRNN 알고리즘을 사용한 모델의 정확도를 비교하는 실험을 통해, CNN 대비 ECRNN의 성능 개선 효과를 분석하였다. 셋째, CRNN과 ECRNN 알고리즘을 사용한 모델의 정확도를 비교하는 실험을 통해, CRNN 대비 ECRNN의 성능 개선 효과를 분석하였다. 마지막으로, CRNN과 ECRNN에서 사용하는 RNN 계열의 알고리즘인 LSTM과 BiLSTM의 정확도를 비교하는 실험을 진행하였다.

4.1 CNN과 CRNN 알고리즘 정확도 비교

첫 번째는 CNN과 CRNN 알고리즘을 사용하여 학습을 진행한 실험결과를 비교하였다. 실험결과는 <표 2>에서 보는 바와 같이, 비교 대상이 되는 CNN은 73.25%의 정확도를 보인 반면, CRNN은 최고 74.52%의 정확도를 보여, 최대 1.27%p 만큼 더 높은 정확도를 보였고, paired t-test 수행 결과 통계적으로 유의한 것으로 나타났다.

CRNN 내부의 은닉층의 노드 수에 따른 정확도 차이를 살펴보면, 1개의 은닉층으로 구성된 LSTM과 2개의 은닉층으로 구성된 LSTM 모두 은닉층의 노드 수가 16개일 때가 32개 일 때보다 약간 더 높은 정확도를 보였다. 반대로 BiLSTM의 경우, 1개의 은닉층으로 구성된 BiLSTM과 2개의 은닉층으로 구성된 BiLSTM 모두 은닉층의 노드 수가 32개일 때가 16개 일 때 보다 약간 더 높은 정확도를 보였다.

4.2 CNN과 ECRNN 알고리즘 정확도 비교

두 번째는 CNN과 ECRNN 알고리즘을 사용하

〈표 2〉 CNN과 CRNN 정확도 비교

(단위: %, %p)

Methods		Number of Nodes in RNN Layers			
		1-Layer		2-Layers	
		16	32	16-16	32-32
CNN		73.25			
CRNN	LSTM	74.08	73.96	73.69	73.08
	BiLSTM	73.40	74.52	73.76	74.49
CNN vs CRNN	LSTM	0.83**	0.71*	0.44	-0.17
	BiLSTM	0.15	1.27***	0.51	1.25***

*p<.05, **p<.005, ***p<.0001.

〈표 3〉 CNN과 ECRNN 정확도 비교

(단위: %, %p)

Methods		Number of Nodes in RNN Layers			
		1-Layer		2-Layers	
		16	32	16-16	32-32
CNN		73.25			
ECRNN	LSTM	75.97	75.60	75.85	75.55
	BiLSTM	77.08	75.95	75.74	75.20
CNN vs ECRNN	LSTM	2.72***	2.36***	2.60***	2.30***
	BiLSTM	3.83***	2.70***	2.49***	1.96***

*p<.05, **p<.005, ***p<.0001.

여 학습을 진행한 실험결과를 비교하였다. <표 3>에서 보는 바와 같이, ECRNN은 은닉층의 개수가 1개이고 은닉층의 노드 수가 16일 때 77.08%의 가장 높은 정확도를 보여, CNN의 정확도 73.25% 보다 3.83%p 더 높은 정확도를 보였다.

ECRNN 내부의 은닉층의 개수에 따른 정확도 차이를 살펴보면, LSTM과 BiLSTM 모두 은닉층별 노드의 수에 상관없이 은닉층의 개수가 1개일 때가 2개일 때보다 약간 더 높은 정확도를 보여주었다. 추가적으로, ECRNN 내부의 은닉층의 노드 수에 따른 정확도 차이를 살펴보면, LSTM과 BiLSTM 모두 은닉층의 개수에 상관없이 은닉층별 노드의 수가 16개일 때가 32개일 때보다 약간 더 높은 정확도를 보여주었다.

4.3 CRNN과 ECRNN 알고리즘 정확도 비교

세 번째는 CRNN과 ECRNN의 알고리즘을 사용하여 학습을 진행한 실험결과를 비교하였다. <표 4>에서 보는 바와 같이, ECRNN이 CRNN보다 최대 2.56%p 더 높은 정확도를 보이는 것으로 나타났다. 전체적인 결과를 보면 ECRNN은 은닉층의 개수와 은닉층별 노드의 수를 달리 하였을 때에도 모든 경우에 있어서, CRNN 보다 더 높은 정확도를

〈표 4〉 CRNN과 ECRNN 정확도 비교

(단위: %, %p)

Methods		Number of Nodes in RNN Layers			
		1-Layer		2-Layers	
		16	32	16-16	32-32
CRNN	LSTM	74.08	73.96	73.69	73.08
	BiLSTM	73.40	74.52	73.76	74.49
ECRNN	LSTM	75.97	75.60	75.85	75.55
	BiLSTM	77.08	75.95	75.74	75.20
CRNN vs ECRNN	LSTM	1.89***	1.64***	2.16***	2.47***
	LSTM(Best)	1.89***			
	BiLSTM	3.68***	1.43***	1.98***	0.71*
	BiLSTM(Best)	2.56***			

*p<.05, **p<.005, ***p<.0001.

보였으며, 모두 통계적으로 유의하게 나타났다.

4.4 LSTM과 BiLSTM 알고리즘 정확도 비교

마지막으로 네 번째는 두 개의 알고리즘 CRNN과 ECRNN에서 사용하는 LSTM과 BiLSTM의 정확도 비교를 위한 실험을 진행하였다. 실험결과는 <표 5>에서 보는 바와 같이, CRNN에서 LSTM을 사용하였을 때와 BiLSTM을 사용하였을 때의 결과를 비교하면, 은닉층의 개수가 1개인 경우 은닉층 노드의 수가 16개일 때 LSTM이 통계적으로 유의한 0.68%p 더 높은 정확도를 보였고, 은닉층의 개수가 2개인 경우 은닉층별 노드의 수가 32개일 때 BiLSTM이 통계적으로 유의한 1.42%p 더 높은 정확도를 보였다. 하지만, LSTM과 BiLSTM 각각 정확도가 가장 높게 나왔을 때를 비교하였을 경우, BiLSTM이 LSTM 보다 다소 높게 나타났으나, 통계적으로 유의하지 않게 나타났다.

ECRNN에서 LSTM을 사용하였을 때와 BiLSTM을 사용하였을 때의 결과를 비교하면, 은닉층의 개수가 1개인 경우 은닉층 노드의 수가 16개일 때 BiLSTM이 통계적으로 유의한 1.11%p 더 높은 정확도를 보였고, 은닉층의 개수가 2개인 경우 두 알고리즘 간 정확도 차이는 통계적으로 유의하지 않게 나타났다. LSTM과 BiLSTM 각각 정확도가 가장 높게 나왔을 때를 비교하였을 경우, BiLSTM이 LSTM 보다 통계적으로 유의한 1.11%p 더 높은 정확도를 보였다.

Ⅴ. 결 론

본 연구는 기존에 제안된 CRNN 알고리즘의 한계점을 개선하기 위해 인코더와 디코더에 기반한 CNN과 RNN의 새로운 하이브리드 방식을 제안하였다. 기존의 CNN 및 CRNN과 본 연구에서 새롭게 제안된 방식의 알고리즘을 사용한 ECRNN의 정확도 비교를 위해 다양한 실험을 진행한 결과, CRNN과 ECRNN 모두 CNN보다 더 높은 정확도를 보이는 것으로 나타났다. 특히 본 논문에서 제안하고 있는 ECRNN은 CNN 대비 CRNN보다 더 높은 정확도 향상을 보이는 것으로 나타났는데, 본 논문에서 제안한 ECRNN의 최고 정확도는 77.08%로 CRNN의 최고 정확도 74.52% 보다 2.56%p 더 높은 정확도를 보이는 것으로 나타났다. 또한,

〈표 5〉 LSTM과 BiLSTM 정확도 비교

(단위: %, %p)

Methods		Number of Nodes in RNN Layers			
		1-Layer		2-Layers	
		16	32	16-16	32-32
CRNN	LSTM	74.08	73.96	73.69	73.08
	BiLSTM	73.40	74.52	73.76	74.49
ECRNN	LSTM	75.97	75.60	75.85	75.55
	BiLSTM	77.08	75.95	75.74	75.20
CRNN	LSTM vs BiLSTM	-0.68*	0.56	0.07	1.42***
	LSTM(Best) vs BiLSTM(Best)	0.44			
ECRNN	LSTM vs BiLSTM	*1.11***	0.35	-0.11	-0.34
	LSTM(Best) vs BiLSTM(Best)	*1.11***			

*p<.05, **p<.005, ***p<.0001.

ECRNN의 경우, 은닉층의 개수가 2개 보다는 1개, 은닉층별 노드의 수가 32개 보다는 16개일 때, 더 높은 정확도를 보이는 것으로 나타났다.

본 연구는 다음과 같은 학문적 시사점을 가진다. 첫째, 기존에 많은 선행연구에서 주로 사용해 오던 CNN과 RNN의 하이브리드 방법이 가지고 있었던 한계점을 제시하고, 이를 인코더와 디코더의 개념을 응용하여 개선한 새로운 CNN-RNN 하이브리드 방법을 제안하였다. 둘째, 다양한 알고리즘 비교 실험을 통해, 새로운 하이브리드 방법의 효과성을 검증함으로써 인코더와 디코더 개념의 적용 가능성을 넓혔다. 셋째, 새로운 하이브리드 방법은 기존 하이브리드 방법에 비해, 복잡도가 많이 증가하지 않아 모델 학습 시간과 인프라 구축 비용 측면에서 이점을 가진다.

또한, 본 연구는 필기체 인식, 포털 등의 동식물명 인식, 의료분야에서의 영상 이미지 판독, 관세분야에서의 수입물품 이미지를 바탕으로 한 HS코드 추천 등 정확한 이미지 분류가 필요한 다양한 분야에서 제공되는 서비스의 품질을 높일 수 있는 가능성을 제시하였다는 점에서 실무적 시사점을 가진다.

본 논문의 실험은 선행 연구된 알고리즘을 대상으로 여러 번의 학습을 진행하였고, 새로운 하이브리드 알고리즘과 정확도를 비교하였다. 다만, 본 연구의 실험에서는 한 가지 종류의 학습데이터만을 사용하였기 때문에, 이미지 크기와 타겟 클래스 수의 변화에 따른 ECRNN의 성능 변화를 분석하지 못한 한계점이 있다. 향후 연구에서는 다양한 이미지 크기와 다양한 타겟 클래스를 가진 학습데이터를 이용하여 본 연구에서 제안한 방법론의 신뢰도를 높이는 연구를 진행하고자 한다.

참 고 문 헌

[1] 김윤진, *딥러닝(Deep Learning)을 활용한 이미지 빅데이터(Big Data) 분석 연구*, 박사학위논문. 중앙대학교 대학원, 2017.

[2] 박경철, *임베디드 플랫폼에서 빠르고 정확한 객체 분류를 위한 체계적 학습이 가능한 조건부 합성곱 신경망*, 석사학위논문, 서울시립대학교 대학원, 2019.

[3] 박호연, 김경재, "CNN-LSTM 조합모델을 이용한 영화리뷰 감성분석", *지능정보연구*, 제25권, 제4호, 2019, pp. 141-154.

[4] 성상하, *딥 러닝을 활용한 이미지 내 한글 텍스트 인식 알고리즘 개선에 관한 연구*, 석사학위논문, 동아대학교 대학원, 2019.

[5] 이동엽, 유원희, 임희석, "자질 보강과 양방향 LSTM-CNN-CRF 기반의 한국어 개체명 인식 모델", *한국융합학회논문지*, 제8권, 제12호, 2017, pp. 55-62.

[6] 이영욱, 박재한, 신수용, "CNN-LSTM 기반 낙상 감지 시스템 구현", *한국통신학회논문지*, 제47권, 제2호, 2022, pp. 340-347.

[7] 이정민, 이현, "단백질 기능 예측 문제에서 시퀀스 패턴 추출을 위한 작은 CNN-RNN 접목 모델 연구", *한국컴퓨터정보학회*, 제27권, 제8호, 2022, pp. 49-59.

[8] 임근영, 조영복, "임의 차원 데이터 대응 Dynamic RNN-CNN 멀웨어 분류기", *한국정보통신학회논문지*, 제23권, 제5호, 2019, pp. 533-539.

[9] 허인성, *합성곱 신경망의 기계학습 기법을 이용한 이미지 분류*, 석사학위논문, 서강대학교 대학원, 2015.

[10] 홍창우, 허견, "CNN-RNN 기반의 DNN을 활용한 DP 선박의 전력부하 예측", *Journal of the Korea Society for Naval Science and Technology*, 제4권, 제2호, pp. 121-126.」

[11] Chatfield, K., K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets", arXiv preprint arXiv:1405.3531, 2014.

[12] He, K., X. Zhang, S. Ren, and J. Sun, "Deep Residual learning for image recognition", *Pro-*

ceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[13] Hu, J., L. Shen, and G. Sun, "Squeeze-and-excitation networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132-7141.

[14] Krizhevsky, A., I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional Neural Networks", *Advances in neural Information Processing System*, Vol.25, 2012, pp. 1-9.

[15] LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, Vol.86, No.11, 1998, pp. 2278-2324.

[16] Shi, B., X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.39, No.11, 2016, pp. 2298-2304.

[17] Simonyan, K. and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv, 2014, pp. 1409-1556.

[18] Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and A. Rabinovich, "Going deeper with convolutions", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.

[19] Tan, M. and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks", *In International Conference on Machine Learning*, 2019, pp. 6105-6114.

Information Systems Review
Volume 25 Number 1
February 2023

New Hybrid Approach of CNN and RNN based on Encoder and Decoder

Jongwoo Woo* · Gunwoo Kim** · Keunho Choi***

Abstract

In the era of big data, the field of artificial intelligence is showing remarkable growth, and in particular, the image classification learning methods by deep learning are becoming an important area. Various studies have been actively conducted to further improve the performance of CNNs, which have been widely used in image classification, among which a representative method is the Convolutional Recurrent Neural Network (CRNN) algorithm. The CRNN algorithm consists of a combination of CNN for image classification and RNNs for recognizing time series elements. However, since the inputs used in the RNN area of CRNN are the flatten values extracted by applying the convolution and pooling technique to the image, pixel values in the same phase in the image appear in different order. And this makes it difficult to properly learn the sequence of arrangements in the image intended by the RNN. Therefore, this study aims to improve image classification performance by proposing a novel hybrid method of CNN and RNN applying the concepts of encoder and decoder. In this study, the effectiveness of the new hybrid method was verified through various experiments. This study has academic implications in that it broadens the applicability of encoder and decoder concepts, and the proposed method has advantages in terms of model learning time and infrastructure construction costs as it does not significantly increase complexity compared to conventional hybrid methods. In addition, this study has practical implications in that it presents the possibility of improving the quality of services provided in various fields that require accurate image classification.

Keywords: Deep Learning, CNN, RNN, LSTM, BiLSTM, CRNN

* Assistant Director, DataWorld Co., Ltd.
** Professor, Department of Business Administration, Hanbat National University
*** Corresponding Author, Associate Professor, Department of Business Administration, Hanbat National University

◐ 저 자 소 개 ◑

우 종 우 (anarchip@daum.net)

국립한밭대학교에서 경영학 석사학위를 수여하였으며, 현재 ㈜데이터월드에 재직 중이다. 관세청 등 국가공공기관 관련 시스템 구축 및 유지 사업에 참여하고 있으며, 주요 관심분야는 머신러닝, 딥러닝, 자연어처리, 연관 규칙 등이다.

김 건 우 (gkim@hanbat.ac.kr)

고려대학교에서 경영학 박사학위를 수여하였으며, 현재 국립한밭대학교 융합경영학과에서 교수로 재직 중이다. 주요 관심분야는 비즈니스 온톨로지 모델, 빅데이터 분석, 핀테크 기술 및 전략 등이다.

최 근 호 (keunho@hanbat.ac.kr)

고려대학교에서 경영학 박사학위를 수여하였으며, 현재 국립한밭대학교 융합경영학과에서 부교수로 재직 중이다. 주요 관심분야는 추천시스템, 의료 빅데이터 분석, 딥러닝, 머신러닝 등이다.

논문접수일 : 2022년 10월 27일 **게재확정일 : 2023년 01월 10일**
1차 수정일 : 2022년 12월 22일