

세탁물 관리를 위한 문자인식 딥러닝 모델 경량화

Lightweight Deep Learning Model of Optical Character Recognition for Laundry Management

임승진¹, 이상협², 박장식^{3*}

Seung-Jin Im¹, Sang-Hyeop Lee², Jang-Sik Park^{3*}

〈Abstract〉

In this paper, we propose a low-cost, low-power embedded environment-based deep learning lightweight model for input images to recognize laundry management codes. Laundry franchise companies mainly use barcode recognition-based systems to record laundry consignee information and laundry information for laundry collection management. Conventional laundry collection management systems using barcodes require barcode printing costs, and due to barcode damage and contamination, it is necessary to improve the cost of reprinting the barcode book in its entirety of 1 billion won annually. It is also difficult to do. Recognition performance is improved by applying the VGG model with 7 layers, which is a reduced-transformation of the VGGNet model for number recognition. As a result of the numerical recognition experiment of service parts drawings, the proposed method obtained a significantly improved result over the conventional method with an F1-Score of 0.95.

Keywords : Lightweight Deep Learning, Optical Character Recognition, VGG Model, Mathematical Morphology

1 부산교통공사

2 경성대학교 전자공학과

3* 교신저자, 경성대학교 전자공학과, 교수,

E-mail: jsipark@ks.ac.kr

1 Busan Transportation Corporation

2 Dept. of Electronic Eng., Kyung Sung University, Student

3* Corresponding Author, Dept. of Electronic Eng., Kyung Sung

University, Professor (E-mail: jsipark@ks.ac.kr)

1. 서론

세탁물 프랜차이즈 업체는 세탁물 수거관리를 위해서 세탁물 수탁자 정보 및 세탁물 정보 기록을 위해서 바코드(bar-code) 인식 기반 시스템이 주로 활용되고 있다. 종래 바코드를 이용한 세탁물 수거 관리시스템은 바코드 인쇄비용 소요되고, 바코드의 훼손 및 오염으로 인하여 바코드를 전면 재인쇄 하는데 연간 약 10억원의 비용이 발생하고 있어 바코드를 대체할 방안이 필요하다. 또한 관리하는 측면에서 바코드의 훼손을 관리자가 확인하는데도 어려움이 있다. 따라서 원활한 세탁물 수거 및 배송 관리를 위한 숫자를 활용한 태그를 활용하는 것이 필요하고, 보다 효율적인 관리 운영을 위하여 자동으로 태그를 인식하는 문자인식 기술이 필요하다.

문자인식 기술은 일반적으로 광학문자인식(OCR, Optical Character Recognition)이라 한다[1]. 최근에는 OCR시스템 구현을 위하여 CNN (Convolutional Neural Network) 등과 같은 딥러닝 기술을 적용한 문자를 인식하는 방법에 대하여 연구가 활발히 진행되고 있다[2].

광학 문자 인식은 사람의 필기체 또는 촬영된 사진, 인쇄된 문서 등에서 문자 또는 숫자를 인식하여 디지털 텍스트로 변환하는 방법을 의미한다. 광학 문자 인식은 차량 번호판, 영수증, 은행, 학교, 여권, 수표 인식 등 아날로그로 기록된 자료를 디지털화가 필요한 다양 분야에서 사용되고 있다.

Google의 Vision API는 LSTM(Long Short-Term Memory) 딥러닝으로 50개 이상의 언어에 대한 문자 인식 가능하며, 인식률은 95% 정도이다[4]. ABBY의 Fine Reader는 199개에 언어를 인식하며, 문자 인식 이외에도 바코드 인식과 고전 양식의 글꼴 인식이 가능하다. 웹 서비스 API(SOAP와 REST API)를 제공하여 범용성을 확장하였다[4]. 두 서비스 모두 영어와 숫자를 대상으로 95%

이상의 인식률을 제공하지만, 아시아권 언어에 대해서는 인식률이 낮은 실정이다. 반면 IntSig는 동양권 언어 인식에 CNN, ANN 알고리즘을 적용하여 기존의 모델보다 인식률을 개선하였다[5].

문자인식 기술은 2가지 방식이 있다. 첫 번째 방식은 고정된 위치와 크기의 문자를 인식하는 것으로, 문자의 위치와 크기를 지정하여 객체를 검출하는 과정을 생략하고 문자를 인식한다.

두 번째 방식은 객체검출(Object Detection)을 사용하여 문자의 위치와 크기를 찾고 문자를 인식한다. 앞에서 언급한 고정된 위치와 크기를 사용하는 방식은 객체검출 방식에 비해 계산량은 적으나 입력영상에 대한 잡음과 변이 등의 상태 변화에 따라 객체검출을 적용한 방식에 비해 영향을 많이 받는다.

객체검출은 2 가지 방식으로 나뉜다. 첫 번째 방식은 2단계(Two-Stage) 검출로 영역 제안(Region Proposal)을 후보 객체를 추정되는 영역을 선택적 탐색(Selective Search)을 수행하거나 영역 제안 네트워크(Region Proposal Network)를 통해 객체의 위치를 검출하고 이를 인식한다. 대표적으로 R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN 등의 모델이 있다[6-7].

두 번째 방식은 단일 단계(One-Stage) 검출로 원본 영상을 정해진 영역(Grid)으로 나누어, 해당 영역에 대해 객체의 크기와 개수를 예측한다. 대표적으로 YOLO(You Only Look Once), SSD(Single Shot Detector), 등의 알고리즘이 있다[8-9]. 단일 단계 검출은 2 단계 검출보다 상대적으로 정확도가 떨어지지만 빠른 처리가 가능하다.

본 논문에서는 세탁물 프랜차이즈 업체에서 요구하는 세탁물 수거 및 배송 관리를 위한 문자인식 시스템을 위한 딥러닝 경량화 모델을 제안한다. 문자인식을 위하여 객체의 위치를 검출하고 세탁 등에 의한 잡음을 제거하기 위하여 수학적 형태학 필터(Mathematical Morphology Filter)를 적용한

3. 제안하는 광학 문자 인식 시스템

3.1 세탁물 관리 코드 인식 시스템

본 논문에서는 세탁물 관리 숫자 코드 인식을 위한 오염된 숫자에 대하여 전처리를 적용하여 오검출을 줄이고, 임베디드 시스템에서 적용할 수 있는 경량화 딥모델 모델을 제안한다.

제안하는 방법은 전처리는 수학적형태학 필터링을 통하여 문자의 위치를 인식한다. 전처리된 숫자에 대하여 경량화 딥러닝 모델로 숫자를 인식한다.

3.2 수학적 형태학 전처리

수학적 형태학 연산은 형태 구조 요소(element)를 이용하여 반복적으로 영역을 확장 및 축소하여 잡음 제거 및 선 잇기가 가능한 연산이다. 형태학 연산에는 침식(erosion), 팽창(dilation)으로 구성되어 있다.

침식 연산은 Fig. 2의 (a)와 같이 객체의 형태

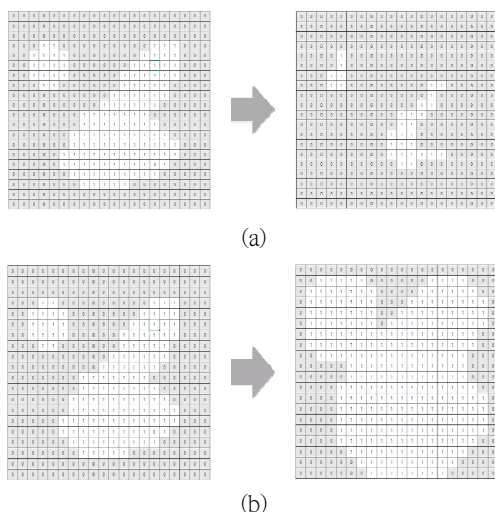


Fig. 2 Mathematical morphological operations
(a) Erosion (b) Dilation

를 주변 배경과 일정한 크기로 축소시키는 연산이다. 연산을 수행하게 되면 객체의 크기는 줄어들고, 배경은 확대되어 영상의 객체와 배경 사이에 잡음이 있을 때에 이 잡음을 제거하거나 아주 작은 객체를 제거할 수 있으며, 영상에서의 외부 돌출부는 감소시키고, 내부 돌출부는 증가시켜 객체 분리도 가능하며, 최소값 필터의 역할을 수행하는 연산이다.

팽창 연산은 Fig. 2의 (b)와 같이 객체의 형태를 내부의 크기는 감소하고 외부 크기는 증가시켜 객체의 전체적인 크기를 확장하고 배경은 축소하는 연산이다. 객체 내부에 발생한 구멍과 같은 공간을 채우거나 짧게 끊어진 영역을 다시 연결하는데 사용되며 최대값 필터의 역할을 수행하는 연산이다.

침식과 팽창을 수행하는 방법에 따라 나뉘는데 열림 연산(opening)은 축소 연산 후 팽창 연산을 수행하는 것이고 닫힘 연산(closing)은 팽창 연산 후 축소 연산을 수행하는 것이다.

세탁물 관리 숫자 코드는 세탁으로 인해 발생하는 오염과 훼손을 포함하는 잡음으로 인하여 단순한 패턴 매칭 숫자인식은 적용하기 어렵다. 잡음의 종류는 연필, 볼펜 등으로 쓰인 낙서과 번짐, 흐려짐 등으로 인한 이점이 있으며, 잡음으로 인하여 기존의 광학 문자 인식은 숫자 이외의 많은 정보를 그대로 인식하기 때문에 인식률도 떨어지며 처리 속도 또한 떨어진다. 본 논문에서는 이를 개선하기 위하여 형태학 연산을 이용하여 잡음을 제거 또는 축소하여 영상의 인식률을 높이고 처리량을 줄여서 처리 속도를 개선하였다.

전처리 과정을 통해 문자의 위치와 크기를 파악하여 객체검출을 사용하지 않고 문자를 인식한다.

3.3 VGG 모델 경량화

VGGNet은 컨볼루션, 풀링(Pooling), FC(Fully

connected) 계층으로 구성되어 있으며, 계층의 수에 따라 VGG-16, VGG-19 등으로 불린다. 3×3 크기의 커널을 반복 수행함으로써 커널 크기가 큰 컨볼루션과 대응할 수 있도록 한다[3]. 본 논문에서 제안 적용하는 모델은 Fig. 3과 Table 1과 같이 VGG 모델을 경량화하여 데이터 입력 크기 기존의 224×224 를 대신 28×28 로 7 계층으로 구성된다. VGG-16과 VGG-19는 컨볼루션 계층의 개수를 16, 19 개를 사용한다. 이는 입력 영상의 크기가 224×224 에 대해 7×7 크기의 컨볼루션 계층의 피쳐맵(featuremap)을 생성하기 위함으로 기존의 입력 영상보다 크기가 작은 세탁물 관리 코드의 경우 28×28 크기의 영상을 224×224 크기로 변형하여 사용하기 때문에 오히려 크기 변환에서 계산량이 늘어나게 된다. 이를 방지하기 위해 7 개의 계층만을 사용하여 경량화를 수행하였다.

커널 크기(Kernel size)는 3×3 으로 VGG과 같도록 설계하였다.

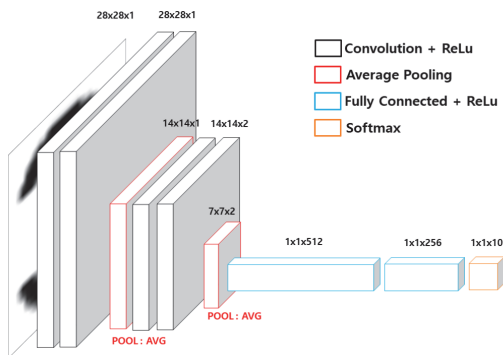


Fig. 3 Structure of VGG

세탁물 관리 숫자 코드의 문자 크기는 넓이 15부터 30까지, 높이 20에서 40까지 작은 크기의 이미지이며, 학습과정에서 이미지 손실을 줄이기 위해 입력 데이터의 크기가 작은 VGG 경량화 모델이 적절하다. OCR 시스템에 학습된 모델을 적용하여 숫자를 인식하는 과정에서 계층의 크기가

작기 때문에 전처리 과정을 포함한 경량화 모델이 인식 처리 속도가 빠르다. 제안하는 경량화 모델은 계산량이 상대적으로 적기 때문에 임베디드 시스템 적용이 용이하다.

Table 1. VGG architecture

Layer(type)		Size
Conv1_1	Convolution	$28 \times 28 \times 1$
Conv1_2	Convolution	$28 \times 28 \times 1$
Pool_1	Avg Pooling	$14 \times 14 \times 1$
Conv2_1	Convolution	$14 \times 14 \times 2$
Conv2_2	Convolution	$14 \times 14 \times 2$
Pool_2	Avg Pooling	$7 \times 7 \times 2$
Fc_1	FC	512
Fc_2	FC	256
Fc_3	FC	10
Softmax	FC	10

4. 실험 및 고찰

4.1 실험 환경

세탁물 관리 숫자코드 인식을 위한 개발환경은 운영체제로 Windows 10 64bit, 하드웨어는 32GB RAM, i7 5960 3.5GHz, Nvidia Titan X 12GB를 사용하였으며, Python 3.6 기반의 pyTorch, Darknet, OpenCV 3.4.1를 이용하여 Visual Studio로 개발한다.

데이터셋은 0부터 9까지 28×28 크기의 숫자를 정상 데이터 12,257건, 오염 데이터 12,257건 총 24,514 건을 추출하여 데이터셋을 구성하였고, 학습에 70% 시험에 30%를 사용하였다.

성능 비교를 위하여 VGG-16, VGG-19와 YOLOv4, CRNN과 비교한다. 전처리 연산에 의한 성능 분석을 한다.

4.2 실험 결과 및 고찰

Fig. 4와 같은 정상데이터와 오염된 데이터를 사용하여 제안하는 OCR에 대한 성능 비교를 수행하였다.

제안하는 OCR에 대한 성능 검증을 위하여 각 모델에 대하여 True Positive, True Negative, False Positive, False Negative을 산출하고 재현률(Recall), 정밀도(Precision) 그리고 F1-Score 계산하여 비교 분석한다. 재현률, 정밀도 그리고 F1-Score를 계산하기 위한 수식은 다음과 같다.

$$Recall = \frac{TP}{TP + FP}$$

$$Precision = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 \frac{(Precision * Recall)}{(Precision + Recall)}$$

VGG 계열과 VGG 경량화의 성능 분석 결과는 Table 2와 같다. 전처리를 적용한 경우 F1-Score가 약 5% 증가하는 것을 확인하였다. 제안하는 OCR이 약 95%로 VGG-16, VGG-19 비해서 성능이 가장 우수한 것을 확인할 수 있다.

Table 3은 검출 및 분류 모델과 제안하는 경량화 모델의 성능과 계산량을 비교한 것이다. 계산량 비교는 FLOPs를 산출하였으며, FLOPs는 다음과 같다.

$$FLOPs = cores * clock * \frac{FLOPs}{cycle}$$

제안하는 VGG 경량화 모델이 YOLOv4 및 CRNN에 비해서 3% 정도 성능이 저하되었으나 계산량은 23G FLOPs로 상대적으로 적은 것을 확인할 수 있다. 제안하는 VGG 경량화 모델이 임베디드 시스템에 적용하는 것이 효율적임을

확인할 수 있다.

Table 2. Comparison of performance by model according to preprocessing

Model	Precision	Recall	F1-Score
Proposed Model	0.93	0.97	0.95
VGG-16 Pre-Pros.	0.71	0.78	0.74
VGG-19 Pre-Pros.	0.79	0.81	0.80
VGG-16	0.68	0.72	0.70
VGG-19	0.72	0.79	0.75

Table 3. Comparison of accuracy and computations of models

Model	Precision	Recall	F1-Score	FLOPs
Proposed Model	0.93	0.97	0.95	23G
YOLOv4	0.98	0.99	0.98	142.8G
CRNN	0.99	0.99	0.99	260.9G



Fig. 4 Normal and anomaly input datasets

5. 결론

본 논문은 세탁물 관리를 위한 문자 인식의 오검출을 줄이고 처리 속도를 개선하기 위해 수학적 형태학 연산을 전처리로 적용하고 경량화 딥러닝 모델을 개발 적용하였다. VGG-19가 75% 인식률을 보였지만 제안하는 OCR은 95%로 성능이 개

선되었고, 전처리를 수행하기 전에는 75%, 적용한 모델이 80%로 성능이 개선하였다. YOLOv4와 CRNN에 비하여 문자인식 성능이 저하되지만, 계산량은 10배 이상 계산량이 개선되는 것을 확인하였다. 향후 추가적인 연구를 통하여 미검출되는 문자나 잡음을 오인식하는 문제점을 개선하고자 한다.

감사의 글

본 논문(저서)는 2022년도 지식학협력 기업 R&D 지원사업의 세탁물 관리 코드 인식을 위한 AI 문자(숫자) 인식 판독기 개발과 부산광역시 및 (재)부산인재평생교육진흥원의 BB21플러스사업의 지원 연구 결과임.

참고문헌

- [1] Jeong P.S., Cho Y.H., "User Authentication System using OCR," *Journal of the Korea Convergence Society*, vol. 9, no. 9, pp. 15-22, (2018).
- [2] Min K.H., Lee A.R., Kim K.S., Kim J.E., Gang H.S., Lee G.H., "Recent Trends in Deep Learning-Based Optical Character Recognition," *Electronics and Telecommunications Trends*, vol. 37, no. 5, pp. 22-32, (2022).
- [3] Hwang S.M., Yeom H.G., "An Implementation of a System for Video Translation on Window Platform Using OCR," *Journal of The Korea Internet of Things Society*, vol. 5, no. 2, pp. 15-20, (2019).
- [4] Tafti A.P., Baghaie A., Assefi M., Arabnia H.R., Yu Z., Peissig, P.L., "OCR as a Service: An Experimental Evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym," *ISVC*, pp. 735-746, (2016).
- [5] Min G.H., Lee A.R., Kang H.S., "A Study on preprocessing for AI based Chinese character segmentation in historical documents," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, pp. 597-598, (2021).
- [6] Xie X., Cheng G., Wang J., Yao X., Han J., "Oriented R-CNN for Object Detection," *Computer Vision and Pattern Recognition*, (2021).
- [7] Sun P., Zhang R., Jiang Y., Kong Tao., Xu C., Zhan W., Tomizuka M., Li L., Yuan Z., Wang C., Luo Ping., "Sparse R-CNN: End-to-End Object Detection with Learnable Proposals," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (2020).
- [8] Jiang P., Ergu D., Liu F., Cai Y., Ma, B., "A Review of Yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066-1073, (2022).
- [9] WANG C.Y., BOCHKOVSKIY A. LIAO H.Y., "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, (2022).