

http://dx.doi.org/10.17703/JCCT.2024.10.2.517

JCCT 2024-3-61

웹페이지 분석을 위한 딥러닝 모델 학습과 구현에 관한 연구

Research on Training and Implementation of Deep Learning Models for Web Page Analysis

김정환*, 조재원**, 김진산***, 이한진****

Jung Hwan Kim*, Jae Won Cho**, Jin San Kim***, Han Jin Lee****

요약 본 연구는 ChatGPT 서비스의 출시 이후 인공지능 혁명이라 일컬어지는 시대적 배경 속에서, 웹사이트의 제작과 인공지능의 융합을 위해 딥러닝 모델을 학습 및 구현하고자 한다. 딥러닝 모델은 수집한 3,000개의 웹페이지 이미지를 구성요소와 레이아웃 분류체계 기반의 데이터 가공을 통해 학습하였으며, 다음과 같은 세 가지 단계로 구분하여 진행하였다. 첫째, 인공지능 모델에 관한 선행연구를 조사하여 구현하고자 하는 모델에 가장 적합한 알고리즘을 선택하였다. 둘째, 적합한 웹페이지 및 단락 이미지를 수집하고 분류 및 가공하였다. 셋째, 딥러닝 모델을 학습시키고 서버 인터페이스를 연동해 모델의 실제 결과를 확인하였다. 이렇게 구현된 모델은 실제 웹페이지를 구성하는 복수의 단락을 탐지하고, 단락별 규모, 요소, 특징을 분석하여 분류체계를 기반으로 의미 있는 데이터를 도출할 것이다. 이 과정은 점차 발전하여 웹페이지를 보다 정밀하게 분석할 수 있게 될 것이다. 그리고 정밀 분석기법을 역으로 설계하여, 인공지능이 완벽한 웹페이지를 자동으로 생성할 수 있는 연구의 초석이 될 것으로 기대한다.

주요어 : 웹페이지 분석, 웹사이트 자동생성, 딥러닝, 웹사이트 분류체계

Abstract This study aims to train and implement a deep learning model for the fusion of website creation and artificial intelligence, in the era known as the AI revolution following the launch of the ChatGPT service. The deep learning model was trained using 3,000 collected web page images, processed based on a system of component and layout classification. This process was divided into three stages. First, prior research on AI models was reviewed to select the most appropriate algorithm for the model we intended to implement. Second, suitable web page and paragraph images were collected, categorized, and processed. Third, the deep learning model was trained, and a serving interface was integrated to verify the actual outcomes of the model. This implemented model will be used to detect multiple paragraphs on a web page, analyzing the number of lines, elements, and features in each paragraph, and deriving meaningful data based on the classification system. This process is expected to evolve, enabling more precise analysis of web pages. Furthermore, it is anticipated that the development of precise analysis techniques will lay the groundwork for research into AI's capability to automatically generate perfect web pages.

Key words : Web Page Analysis, Automatic Website Generation, Deep Learning, Website Classification

*정회원, 고려대학교 디지털경영학과 박사 (제1저자)
(컴퓨터공학 석사, 주식회사 위븐 대표이사)

**정회원, 명지전문대학교 산업디자인학과 전문학사 (참여저자)

***정회원, 서울사이버대 빅데이터 정보보호학과 학사 (참여저자)

****정회원, 한동대학교 창의융합교육원 교수 (교신저자)

접수일: 2024년 1월 2일, 수정완료일: 2024년 1월 23일

게재확정일: 2024년 1월 31일

Received: January 2, 2024 / Revised: January 23, 2024

Accepted: January 31, 2024

*Corresponding Author: cus@handong.edu

School of Creative Convergence, Handong Global Univ.,

Pohang, Korea

I. 서 론

1980년대 후반의 미약했던 월드 와이드 웹(World Wide Web, WWW)의 시작은, 현재 전 세계에 걸쳐 다양한 서비스와 콘텐츠를 제공하며 일상 생활에 없어서는 안 되는 부분이 되었다. 초창기 웹사이트는 주로 텍스트로 구성되어 있었고, 그래픽으로 사용되는 이미지가 드물었다. 그러나 지금의 웹사이트는 상호작용과 사용자경험이 필수적인 정교한 수준으로 발전해 왔다[1]. 지속적으로 증가하는 제작의 수요와 높은 요구수준에 대응하기 위해 다양한 제작 기법이 도입되었으며, 웹사이트 제작 방법은 시기별로 다음의 몇 단계를 거쳐 발전하였다.

2000년대 초반에는 각종 라이브러리의 구축과 재사용이 가능하도록, 소프트웨어 개발 방법론을 기존의 절차지향 기법에서 객체지향 기법으로 접근하여 생산성을 높이고 시장에 대응해왔다[2]. 당시에는 웹서버를 임대하여 인터넷 데이터 센터(IDC, Internet Data Center)에 두고 접속환경을 구현하거나, 이를 전문으로 해주는 서비스인 웹호스팅(Web Hosting)서비스를 기반으로 웹사이트를 유지할 수 있도록 하였다. 여기서 발전하여 2000년대 중반부터는 호스팅 중인 서버를 복제하는 등의 방식으로 생산성을 높였고[3], 2010년대에 들어서는 클라우드가 보급되며, 웹빌더 시스템을 온라인에서 서비스하며 제작과 운영을 한 번에 할 수 있는 환경으로 변화했다[4]. 웹빌더는 많은 템플릿과 웹 편집기를 기반으로 디자인과 개발이 가능하고, 웹사이트와 쇼핑몰 운영을 모두 할 수 있는 서비스이다[5]. 이 서비스가 COVID-19 팬데믹으로 수요가 폭증하던 시장의 수요를 충족시키며 견인한 대표 서비스로써, 이후 발전된 웹빌더 서비스까지 등장하는 발판이 되었다.

최근 ChatGPT가 세계를 강타하며, 인공지능에 관한 사람들의 인식이 크게 달라졌다. 나아가 이를 활용한 서비스 개발이 많아지며 다양한 서비스들이 홍수처럼 쏟아지고 있다. 많은 IT 기업의 성공 전략에 인공지능이 빠짐없이 등장하고 있고, 세상의 변화를 주도하고 있다. 이제는 온라인의 근간이 되는 웹사이트를 제작하는 방법에도 인공지능을 융합하여 보다 쉽고 빠르게 정교한 제작 방법을 모색해야만 한다.

웹사이트 제작에 인공지능을 도입하기 위한 방법으로 컴포넌트 등 사용자 인터페이스를 분석하여 시제품

(목업)-디자인을 미리 만들어주거나[6], 예제를 기반으로 추천 및 제작을 도와주기도 한다[7-8]. 아울러 스케치를 기반으로 인공지능이 분석 및 생성을 도와주는[9] 등 웹사이트의 부분 제작에 인공지능을 융합한 연구가 있었다[10].

하지만, 목업 기반의 제작 방법은 컴포넌트나 사용자 인터페이스에 대한 이해가 필수적이었다. 이에 이러한 부분을 그림으로 표현해 지시하는 부분은 연구 및 상용화의 한계로 여겨졌다. 한편 예제 기반의 제작 방법은 수많은 웹사이트 데이터 수집이 필수로 필요해 연구가 활발히 이뤄지지 못했다. 아울러 스케치를 기반으로 분석 및 생성을 도와주는 방법은 사용자가 손으로 스케치하는 표현의 한계가 있었다. 최근에는 생성형AI에 대한 연구나 이미지 생성기반 서비스[11]가 매우 활발해지며, 혁신성과 결과물의 완성도에 대한 기대가 매우 높아졌다. 따라서 상단에서 언급한 목업, 예제, 스케치 기법을 적용한 인공지능 기반 웹사이트 생성 방법의 한계들을 극복하고 발전된 기법으로 접근할 필요성이 있다.

이에 본 연구에서는 체계화된 쇼핑몰 분야의 웹사이트 분류체계를 바탕으로 웹페이지 이미지를 수집하고, 이를 분석할 수 있는 인공지능 모델을 학습 및 구현하고자 한다. 현재까지는 유사성에 기반해 생성하는 부분에만 초점이 맞춰졌으나, 웹페이지를 분석하는 접근은 시도되지 않았다. 이것은 선행연구가 갖는 데이터 입력단의 한계를 근본적으로 해결할 방법이라 여겨진다. 향후 본 연구를 발전시켜 정밀화된 웹페이지 분석이 가능하게 되고, 누적된 분석 결과를 바탕으로 재학습이 이뤄진다면 완전한 웹페이지의 자동생성까지도 가능하리라 여겨진다. 본 고에서는 이 시스템의 핵심인 인공지능 모델의 구현과 실제 웹페이지 분석결과를 확인할 것이다. 이를 위하여 AI 모델에 관한 배경지식을 습득하고 분석에 적합한 알고리즘을 선택[12]할 것이며, 분류체계를 바탕으로 이미지를 수집하여 가공하고자 한다. 나아가 선택된 알고리즘을 바탕으로 모델의 학습과 추론기를 구현하여 실제 결과까지 구현해 보고자 한다.

II. 선행연구 검토

1. 심층학습과 신경망(Deep Learning and Neural Networks)

심층학습은 기계학습의 한 분야로, 복잡한 데이터 구

조에서 패턴과 관계를 학습하는데 사용된다[13]. 기본적으로, 심층학습은 인간 뇌의 신경망을 모방한 인공 신경망(ANN: Artificial Neural Network)을 사용한다. 이러한 신경망은 일반적으로 세 개 이상의 계층으로 구성되어 있으며, 각 계층은 데이터로부터 점차적으로 더 복잡하고 추상적인 특징을 추출한다. 심층학습의 주된 특징은 대량의 데이터를 사용하여 학습한다는 점이다[14].

심층신경망(DNN: Deep Neural Network)은 심층학습 기술에서 핵심적인 역할을 하는 구조로, 다층으로 구성된 복잡한 신경망을 기반으로 한다[15]. DNN은 입력층, 여러 은닉층, 그리고 출력층으로 구성되어 있다. 이러한 다층 구조는 복잡한 문제를 해결하기 위한 효과적인 표현 학습에 적합하다. 각 층은 데이터의 점진적이고 추상적인 특징을 추출하는 역할을 한다. DNN은 깊이가 깊은 네트워크를 특징으로 한다. 네트워크의 깊이는 모델이 학습할 수 있는 데이터의 복잡성과 다양성을 결정하는 중요한 요소이다. 데이터에서 자동으로 특징을 추출하는 능력을 갖추고 있다. 이는 전통적인 기계학습 방법론에서 필요한 수동 특징 추출작업을 대체한다. 대량의 데이터에서 복잡한 패턴과 관계를 학습할 수 있다. 이를 통해 높은 수준의 성능을 달성할 수 있다[16].

2. 이미지 분류 대표 알고리즘

1) ResNet

ResNet은 마이크로소프트에서 개발한 알고리즘으로, 심층학습 분야에서 많이 사용되는 유명한 모델이다[16]. 이 모델은 CNN이 계층(Layer) 깊이가 깊어질수록 매개변수 수의 증가로 인한 학습오류 기울기 소실(Gradient vanishing)과 폭주(exploding) 같은 문제를 해결하기 위해 제안되었다[17]. 이것은 Layer를 많이 쌓을수록 데이터 표현력이 증가하기 때문에 학습이 잘 될 것 같아 보인다. 하지만 실제로는 Layer가 많아질수록 학습이 잘 되지 않는 현상으로, 역전파(Back-propagation) 과정에서 출력층에서 멀어질수록 Gradient 값이 매우 작아진다[18].

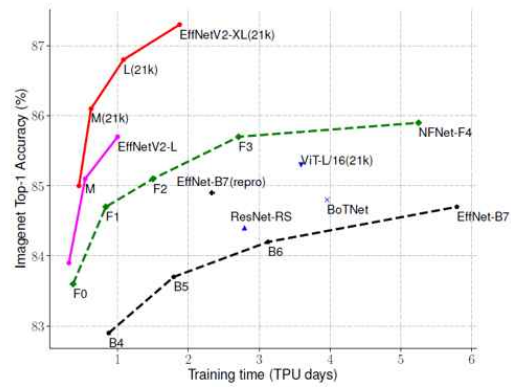
2) EfficientNet

이 모델은 Layer의 수와 Filter의 수 그리고 이미지 사이즈에 대한 최적의 값을 찾아 좋은 성능을 얻은 모델이다. 2019년도에 발표되어 Image 분야에서 가장 중요한 모델이 되었고, 성능의 차이가 압도적인 수준으로 알려졌다[20]. 2019년 이후 인공지능 학계를 강타한

EfficientNet은 2021년에 두 번째 버전인 EfficientNetV2까지 발표했다[20]. <그림 1>에서 보는 바와 같이 비슷한 정확도에서 오히려 절반 밖에 안되는 매개변수를 보여주며 효율성의 극대화를 선보였다.

3. 객체 탐지 대표 알고리즘

1) Faster RCNN



(a) Training efficiency.

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

그림 1. EfficientNetV2의 파라미터와 정확도

Figure 1. Parameters and Accuracy of EfficientNetV2

R-CNN 모델은 객체 검출 및 분류를 위한 대표적인 딥러닝 기반 모델로, 2013년에 처음 제안되었다[21]. 객체 검출 작업은 일반적으로 1-stage와 2-stage 방법으로 나눌 수 있는데, R-CNN은 우수한 성능 구현을 위해 2-stage 방법을 채택하고 있다. R-CNN은 입력 이미지를 전처리 한다. 이미지 크기를 조절하고 채널 값을 정규화 한다. 이러한 전처리 단계는 신경망 모델에 입력하기 전에 수행된다. 입력 이미지에서 객체가 있을 것으로 예상되는 후보 영역(Region Proposal)을 추천한다.

이를 위해 Selective Search와 같은 알고리즘을 사용하여 이미지 내의 객체 후보 영역을 생성한다. 이때 최대 2,000개의 후보 영역을 추천한다. 추천된 후보 영역을 CNN 모델의 입력 크기에 맞게 자르거나 감싼다. 이후, 각 영역에서 특징을 추출한다. 추출된 특징은 분류를 위해 CNN 모델의 입력으로 사용된다. 각 후보 영역에 대해 모델을 적용하여 객체의 클래스를 예측한다[21]. <그림 2>는 검출 분류 방식을 이미지화 한 흐름

도이다.

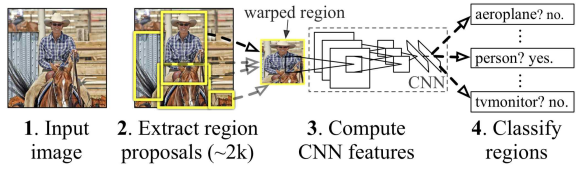


그림 2. CNN 특징과 탐지 영역
Figure 2. CNN Features and Detection Areas

2) YOLO

YOLO (You Only Look Once)는 워싱턴 대학교의 Joseph Redmon과 Ali Farhadi가 개발한 인기 높은 객체 탐지(Object Detection) 및 이미지 분할 모델이다 [22]. 2015년에 출시된 YOLO는 빠른 속도와 정확성으로 인해 큰 인기를 얻었다.

YOLO는 1-stage방식의 심층학습 알고리즘으로, 빠른 속도와 다양한 객체를 정확하게 탐지하고 분석하는데 유리하다. 즉, 입력한 이미지를 한 번의 스캐닝으로 처리하여 빠른 속도를 자랑하며 실시간 응용프로그램에 적합하다.

YOLOv8은 YOLO의 최신 버전이며, 8가지의 다양한 버전이 있다. 탐지, 분할, 자세추정, 추적, 분류를 포함하여 다양한 컴퓨터비전 분야의 학습을 지원하도록 했다. <그림 3>과 같이 YOLO의 버전별 매개변수와 정확도에 대한 성능 향상을 알 수 있었다.

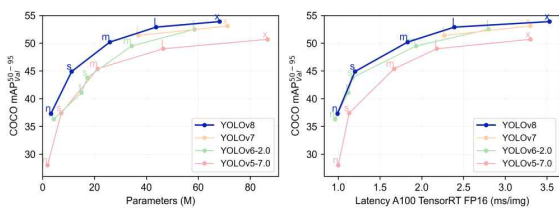


그림 3. 다양한 YOLO 모델 버전의 성능 비교 [23]
Figure 3. Performance Comparison of Different Versions of the YOLO Model [23]

객체 검출 분야에서 YOLO는 정확도 측면에 아쉬운 부분이 있으나, 가장 빠른 객체 검출을 자랑하는 모델로 알려져 있다. 또한 2023년도에 등장한 YOLOv8은 ultralytics에서 발표한 것으로 통합 프레임워크와 함께 제공되며 쉬운 사용성을 보여주어 인기가 높아졌다.

III. 모델의 선택

1. 객체 탐지 모델 YOLOv8

연구를 이어가기 위해 단락들을 탐지하거나, 요소를 탐지하는 등 크고 작은 다양한 특징을 탐지해내야 한다. 이를 위해 선행연구에 대한 검토를 바탕으로 가장 유명하고 효율이 좋다고 알려진 YOLOv8모델을 선택하였다. 본 연구의 목적에 부합하기 위해서는 다수의 호출에 빠르게 응답할 수 있어야 하며 안정적인 응답 능력이 요구된다. YOLO 모델은 다양한 분야에서 실시간 검출로 활용되어 그 성능이 검증되었다. 특히 최근 발표된 모델은 통합 프레임워크의 제공으로 인해 개발의 난이도가 매우 낮아졌으며, 객체 탐지분야의 심층학습 개발자들에게 인기가 매우 높은 만큼 지식을 얻기가 용이할 것으로 판단되었다.

2. 이미지 분류 모델 EfficientNetV2

이미지의 분류는 탐지된 단락을 유형에 맞게 분류할 때 사용할 것이다. 대부분의 특징과 요소의 유무를 판별하고 분석 결과를 추출하기 위해서는 분류 모델 보다는 객체 탐지 모델을 기반으로 수행될 수 있다. 하지만, 예를 들어 로고의 유무를 바탕으로 해당 단락이 Header, Footer, Body 중 어떤 단락인지 분류할 필요가 있다. 또 좌측 메뉴형인지 가운데형인지 등 불필요한 객체 탐지를 한번의 분류로 빠른 분석이 가능하다.

이런 현재 상황을 고려할 때 EfficientNetV2는 매우 적합한 모델이다. 이 모델은 현재 매우 빠른 학습속도를 자랑하며, 정확도마저 우수하다는 산업과 학계의 평이 있기 때문이다. YOLOv8을 통한 학습 비용, 객체 탐지 모델 추론 비용, 개발자의 연산 알고리즘 개발 비용, 확장성까지 총 4가지 변수와 EfficientNetV2를 통한 학습 비용, 이미지 분류 추론 비용, 그리고 확장성의 3가지 변수를 비교하여 고려해왔다.

EfficientNetV2는 Neural Architecture Search(NAS)를 사용하여 모델의 구조를 최적화하고 경량화 하므로 객체 탐지 모델을 활용하는 것에 비해 추론 비용이 매우 효율적일 수 있으며, 추가로 학습을 위한 데이터의 가공이 단순하기에 확장성이 매우 용이할 것이다. 게다가 데이터를 학습하는 속도는 이미 학계의 정평이 난 만큼, 사용자의 제작의도와 관련한 이미지 분석에 보다 많이 활용해야 할 것으로 판단되었다.

IV. 수행결과

1. 데이터 수집 및 가공

모델을 학습하기 위하여 먼쇼핑몰 웹사이트의 메인 페이지를 캡처하는 방식으로 데이터 수집을 진행하였다. 이미지의 수집은 네이버 웨일 브라우저의 확장 프로그램을 통해 스크롤되는 웹페이지를 포함하여 전체 캡처가 약 15일간 진행되었다. 현재 운영되는 글로벌 쇼핑몰을 찾는 방법으로 코랭킹닷컴과 디비컷 등 웹사이트 데이터베이스 기반 서비스 페이지를 다수 참고했다. 전체 수집된 이미지 데이터는 3,000장이었으며, 전체 7.363GB의 크기로 수집되었다.

수집된 이미지에서 단락의 유형에 맞게 분류하거나 요소를 가공할 수 있도록 단락별로 자르는 1차 가공을 진행했다. 또한 직접 단락 캡처 등의 방식을 혼용하여 <그림 4>와 같이 상세 데이터의 수집을 진행하였다. 중복이나 불필요한 단락 일부가 제외한 전체 이미지 데이터는 28,721장이었으며, 전체 용량은 4.128GB이었다.

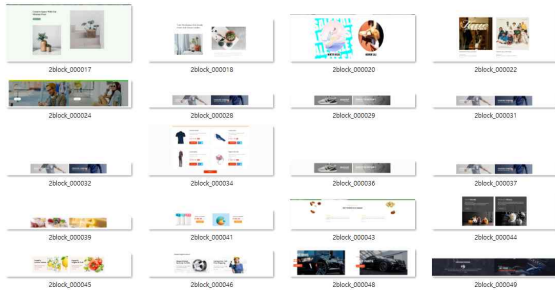


그림 4. 카테고리기반 쇼핑몰 웹사이트 캡처
 Figure 4. Data for Section in Web Pages

수집된 단락 데이터를 바탕으로 단락을 탐지하고 분류할 것이며, 추가로 단락 내 요소를 가공하여 요소를 탐지하고 특징도 탐지하고자 했다. 직전 연구에서 제안한 단락 및 웹페이지 구성요소의 분류체계는 <표 1>과 같다.

표 1. 단락과 요소의 분류 체계
 Table 1. Section and Element Classification System

Class	Classification
Section	Header, Footer, Banner, 2 Block, 3 Block, 4 Block, 5 Block, 6 Block, n Block, Gallery
Element	Header, Footer, Menu, Grid, Page, Title, Text, Image, Video, Banner, Table, Bulletin Board, Form, Map, Button, Icon, Line, File

2. 딥러닝 모델의 학습 및 구현



그림 5. 분석의 종류와 순서 예시
 Figure 5. Examples of Type and Sequence of Analysis

본 연구에서는 수집된 데이터를 가공하여 <그림 5>와 같이 단락을 탐지하고, 단락을 분류하며 단락내에 포함된 요소와 특징을 탐지하는 순서로 분석이 진행되도록 딥러닝 모델 3개를 학습 및 구현하고자 한다.

1) 단락 탐지 모델

이 연구의 초기에는 Header, Body, Footer의 세 가지 분류로 객체 검출을 시도했으나, Body의 다양한 단락의 특징들을 심층학습 모델이 적은 양의 데이터로 학습하기에 어려움이 있었다. 시행 착오에 따라 Body의 유형을 더욱 분류하여 'Header, 1단 Banner, 2단, 3단, 4단, 5단, 6단, 6단이상, 갤러리, Footer'의 10개 클래스로 나누었다. 이어 새로운 학습모델을 구성하였고, 이를 통해 성능적으로 향상된 결과를 얻었다. 웹사이트 단락 탐지 모델 개발은 초기 단계에서 1,000장의 이미지 샘플을 수집하고 분류 체계를 확립하는 것으로 시작되었다.

이 초기 단계에서 수집된 데이터는 모델 개발의 기초를 마련하는데 중요한 역할을 했다. 정해진 분류체계에 따라 대상 이미지들을 수집한 후, 레이블링 가공을 진행하여 초기 학습 데이터셋을 구축했다. 초기 테스트 모델 개발을 위해 1,000장의 이미지를 추가로 수집했으며, 이러한 이미지들에 대해서도 레이블링 작업을 수행하였다. 이 과정에서 이미지들의 레이아웃을 기반으로 단락을 가로로 나누는 중요한 과정을 도출할 수 있었다. 이 과정을 추가하여 이미지 수집 및 가공 단계를 총 16회에 걸쳐 반복하였고, 불순 데이터의 정제의 과정 등을 추가해 심층학습 모델의 학습을 지속하였다. 그 결과 <표 2>와 같이 Body의 세분화된 탐지가 가능

한 모델을 구현했다.

표 2. 단락 분류 레이블 체계
Table 2. Section Classification Labels System

Classification	Lables
Header	Logo_Left, Logo_Right, Header_Search
Body Block	Typo, Shop_List, Banner, SNS, Customer, Brand_Logo, Board, Video, Map, Review, Category, Search, Subscribe, Gallery, Inquiry
Footer	1Block, 2Block, 3Block, 4Block, MenuSNS Image, SNS Icon

2) 단락 분류 모델

탐지된 단락은 단 컷의 이미지로 잘라내어 서버에 저장된다. 그리고, 이 이미지를 다시 유형분류를 위한 모델에 투입하여 미리 준비한 태그 값을 추출하고 단락의 유형을 분류해낸다. 본 연구에서는 단락을 기반으로 추출되는 분석 데이터의 단위를 설명하려 하므로 어떤 분류에 속하는 단락인지는 웹페이지 전체의 속성을 규정짓기 위해 매우 중요하다. 단락을 분류할 때 Header, Block, Footer에 따라 <표 2>의 레이블을 복수로 갖도록 설계하고 학습 및 구현하였다. 섹션 탐지와 마찬가지로 선행 학습을 시도한 후 결과를 보며 재학습의 방법으로 시도하였다.

3) 요소 탐지 모델

단락 이미지를 바탕으로 분류까지 완료되었다면, 단락 이미지에 포함되어 있는 구성요소를 탐지해야 한다. 탐지를 위한 구성 요소의 대상은 메뉴, 그리드, 페이지, 타이틀, 텍스트, 이미지, 영상, 배너, 표, 게시판, 폼, 지도, 버튼, 아이콘, 라인, 파일 등이 있다. 또한 <그림 6>의 좌, 우로 흐르는 슬라이더 배너, 아이콘, 상품 리스트 등의 상호작용 요소를 정확히 분석함으로써, 이용자와 웹사이트는 더욱 적극적으로 소통할 수 있게 될 것이다[24].

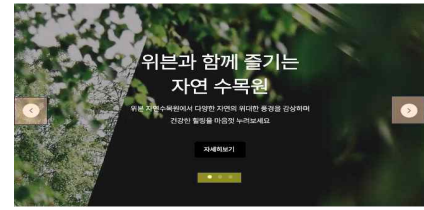


그림 6 슬라이드 배너의 왼쪽 오른쪽 버튼과 페이지 표시 버튼의 탐지 예시
Figure 6. Example of Detecting the left and right buttons of the slider banner and the pagination buttons

한편 요소를 탐지하는 모델은 단락을 탐지하는 모델과 동일한 YOLOv8을 적용해 학습하였다. 특히 작은 이미지 사이즈에 대한 탐지 성능을 강화하기 위해 학습 알고리즘의 하이퍼파라미터를 조정하여 작은 단락의 이미지나 섬세한 요소들에 대한 탐지 능력을 향상시켰다.

V. 결 론

본 연구에서는 인공지능 기술의 발전 속에 원천기술의 개발 못지않게 응용이 중요함[25]에 따라 웹사이트 제작과 인공지능 기술을 융합하고자 했다. 이에 선행연구의 한계를 극복할 수 있는 웹페이지 분석용 딥러닝 모델을 직접 설계하고 구현하였다. 모델의 구현을 위해 초기 1,000장의 데이터를 수집 및 가공하고, 학습 및 추론에 따른 결과 확인 후 오류를 수정하는 등 반복적인 수행을 실시했다. 이렇게 점진적 강화의 방법을 통해 최초로 시도되는 탐색적 연구를 효과적으로 수행할 수 있었다[26]. 나아가 정밀한 인공지능 모델을 실제 구현하고 테스트할 수도 있었다.

이에 해당 연구는 현재까지의 연구된 자동 생성기법들과 다른 접근으로, 생성이 아닌 의도분석에 주안점을 뒀다. 즉, 필수요구 수치의 불편한 입력 문제를 근본적으로 해결하고자 하였다. 나아가 연구자가 직접 수립한 분류체계를 바탕으로 데이터의 수집과 가공 및 실제 인공지능 서비스에서 사용하는 방법을 제시하였다.

향후 분석용 딥러닝 모델들을 시스템으로 연결하고, 자동으로 웹서핑이 가능한 스크래퍼(Scraper) 로봇을 구현한다면, 대량의 웹페이지를 자동으로 캡처하고 분석하여 저장된 다양한 웹페이지 분석 데이터가 매우 유의미하게 활용될 것이다[27]. 이는 지금까지 체계가 부족했던 웹사이트의 디자인과 기획 분야에 다양한 표준과 시사점을 제시할 수 있을 것이라 판단된다.

본 연구에서 구현한 딥러닝 모델은 데이터 수집의 한계와 고성능 컴퓨팅 자원의 한계 등으로 더 많은 요소의 분석이나 특징 분석이 제한되었다. 하지만 점차 모델의 수를 늘려 웹페이지 1개당 10가지 종류의 분석이 가능하게 될 경우, 전문가 수준의 정교한 분석이 이루어질 수 있을 것이다. 이것은 단순히 분석에 그치지 않고, 정교한 분석 데이터와 디자인 조각을 원본(raw) 및 주요 데이터(primary data)로써 학습시켜 완벽한 웹페이지 자동생성이 가능하리라 판단된다.

그리고 정교한 분석기는 최근 이슈화되고 있는 이미지 생성형 AI 서비스에서 부자연스러운 결과가 도출되는 치명적인 약점도 보완해 줄 수 있다[28]. 이것은 분석기를 토대로 벡터DB를 구성하고, 사용자가 입력한 명령어 값이나 결과를 그와 비교하여 교정함으로써, 보다 완성도 높은 조합으로 결과를 제공할 수 있다.

본 연구에서는 활발히 연구되는 인공지능 연구 분야 중 컴퓨터비전(CV: Computer Vision)의 새로운 접근법을 제시하고자 했으며, 산업에 바로 적용이 가능한 방법을 제안했다. 추가로 대부분의 이미지 생성기법이 대화형 서비스를 입력값으로 받는 점을 고려해, 벡터DB를 구축함으로써 서비스의 정교화를 제안하는 등 다양한 학술적 시사점을 제공했다.

그럼에도 불구하고, 본 연구는 다음의 몇 가지 한계점을 갖고 있어 밝혀둔다. 가장 먼저, 제안하는 시스템은 쇼핑몰 웹사이트 중 메인페이지와 같이 상당히 규격화된 경우만을 다루고 있어 유연성의 다양한 디자인 결과를 예측할 수 없다. 전 세계 추산 약 19억 개가 넘는 웹사이트의 유형과 디자인이 모두 다르기 때문에 웹사이트 분류가 세밀하게 확장되어야 하며 엄청난 이미지의 수집과 가공이 필요할 것으로 생각한다.

게다가 웹페이지 분석을 위해 사용자는 벤치마킹 웹사이트를 정확히 캡처해야만 한다. 만약 움직임이 있거나 동적인 웹페이지를 캡처하고자 한다면 초기 요구사항 입력 단계에서 매우 번거로울 수 있으므로, 선행연구의 한계를 완전히 넘어서지 못한다. 향후 연구에서는 자동으로 캡처가 가능한 스크래퍼를 함께 구현하고 제공할 필요가 있다.

종합적으로 볼 때, 본 연구는 웹사이트 자동생성 분야에 새로운 접근으로써 또 하나의 이정표가 될 것으로 기대한다. 동시에 이미 제작되어 있는 웹사이트들을 대상으로 분석하고 표준을 정립할 수 있는 현실적인 대안

을 토대로, 실제로 구현하고 데이터를 분석하는 등 구체화된 후속연구가 요구된다.

References

- [1] Kaluarachchi, T., and Wickramasinghe, M. (2023). A systematic literature review on automatic website generation. *Journal of Computer Languages*, 75. <https://doi.org/10.1016/j.cola.2023.101202>
- [2] Lee, J.-S. (2003). Aspect-Oriented Programming and Subject-Oriented Programming. *Korea Information Processing Society Review*, Vol. 21, No. 9, pp. 94-101.
- [3] Sivasubramanian, S., Szymaniak, M., Pierre, G., & Steen, M. v. (2004). Replication for web hosting systems. *ACM Computing Surveys (CSUR)*, Vol. 36, No. 3, pp. 291-334. DOI : <https://doi.org/10.1145/1035570.1035573>
- [4] Muhammad Garib, N. S. (2006). Online Website Builder for Non-Programmers.
- [5] Bangboonrit, C. (2004). Site builder: build, market and manage business website with CMS.
- [6] Xu, Y., Bo, L., Sun, X., Li, B., Jiang, J., & Zhou, W. (2021). image2emmet: Automatic code generation from web user interface image. *Journal of Software: Evolution and Process*, Vol. 33, No. 8, e2369, DOI : <https://doi.org/10.1002/smr.2369>
- [7] Hashimoto, Y. and T. Igarashi (2005). Retrieving Web Page Layouts using Sketches to Support Example-based Web Design. *SBM*, DOI : <http://dx.doi.org/10.2312/SBM/SBM05/155-164>
- [8] Chang, K. S.-P. and B. A. Myers (2012). WebCrystal: understanding and reusing examples in Web authoring. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. DOI : <https://doi.org/10.1145/2207676.2208740>
- [9] Baule, D., von Wangenheim, C. G., von Wangenheim, A., Hauck, J. C., & Junior, E. C. V. (2021). Automatic code generation from sketches of mobile applications in end-user development using Deep Learning. *arXiv preprint arXiv:2103.05704*. DOI : <https://doi.org/10.48550/arXiv.2103.05704>
- [10] Kaluarachchi, T. and M. Wickramasinghe (2023). "A systematic literature review on automatic Website generation." *Journal of Computer*

- Languages 75. DOI : <https://doi.org/10.1016/j.cola.2023.101202>
- [11] Xi, C., and Chung, J. (2023). A Study on Character Design Using [Midjourney] Application. *International Journal of Advanced Culture Technology*, Vol. 11, No. 2, pp. 409–414. DOI : <https://doi.org/10.17703/IJACT.2023.11.2.409>
- [12] McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, Vol. 27, No. 4, pp. 12–12. DOI : <https://doi.org/10.1609/aimag.v27i4.1904>
- [13] Hinton, G.E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, Vol. 18, No. 7, 1527–1554. DOI : <https://doi.org/10.1162/neco.2006.18.7.1527>
- [14] Cai, S., Bileschi, S., & Nielsen, E. (2020). *Deep Learning with JavaScript: Neural networks in TensorFlow.js*. Manning. <https://books.google.co.kr/id=N2dswgEACAAJ>
- [15] Mahesh, B. (2020). Machine learning algorithms—a review. *International Journal of Science and Research*, Vol. 9, No. 1, pp. 381–386. DOI : <https://doi.org/10.21275/ART20203995>
- [16] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, Vol. 60, No. 6, pp. 84–90. DOI : <https://doi.org/10.1145/3065386>
- [17] Lozano-Diez, A., Zazo, R., Toledano, D. T., & Gonzalez-Rodriguez, J. (2017). An analysis of the influence of deep neural network (DNN) topology in bottleneck feature based language recognition. *PloS one*, Vol. 12, No. 8, e0182580. DOI : <https://doi.org/10.1371/journal.pone.0182580>
- [18] Amanatullah. (2023). Vanishing Gradient Problem in Deep Learning: Understanding, Intuition, and Solutions. Retrieved from <https://medium.com/@amanatulla1606>
- [19] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). DOI: <https://doi.org/10.1109/CVPR.2016.90>
- [20] Tan, M., and Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR. DOI : <https://doi.org/10.48550/arXiv.1905.11946>
- [21] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587). DOI : <https://doi.org/10.48550/arXiv.1311.2524>
- [22] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788). DOI : <https://doi.org/10.1109/CVPR.2016.91>
- [23] Rath, S. (2023). *YOLOv8 : Comprehensive Guide to State Of The Art Object Detection*. <https://learnopencv.com/ultralytics-yolov8/>
- [24] Hanjin Lee, Soyeon Kwon, & Daihwan Min (2021), The Empirical Research on the User Satisfaction of Mobile Grocery Shopping Customer Journey, *Journal of Information Technology Applications & Management*, Vol. 28, No. 4, pp. 59–78. DOI : <https://doi.org/10.21219/jitam.2021.28.4.059>
- [25] Hyun-ju Kim, and Jinyoung Lee (2024). A Study on A Study on the University Education Plan Using ChatGPTfor University Students, *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 10, No. 1, pp. 71–79. DOI : <http://dx.doi.org/10.17703/JCCT.2024.10.1.71>
- [26] Hanjin Lee, Young-geun Park, & Daihwan Min, (2020). Analysis of Factors Affecting the Continuance Intention to Use Mobile Grocery Shopping. *The Journal of Information Systems*, 29(2), 95 - 110. DOI : <https://doi.org/10.5859/KAI S.2020.29.2.95>
- [27] Suhyun Park, Yeeun Lee, & Hanjin Lee (2024). Research on Enhancing Customer Experience through AI-Supported Review Generation, *The Transactions of the Korean Institute of Electrical Engineers*, Vol. 73, No. 2, pp. 334–342. DOI : <https://doi.org/10.5370/KIEE.2024.73.2.334>
- [28] Hyunjin Kim, Yeongjo Kim, Donghyeon Yun, & Hanjin Lee (2024). Empirical Research on the Interaction between Visual Art Creation and Artificial Intelligence Collaboration, *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 10, No. 1, pp.517–524. DOI : <http://dx.doi.org/10.17703/JCCT.2024.1.571>