Journal of KIIT. Vol. 23, No. 1, pp. 177-190, Jan. 31, 2025. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2025.23.1.177

키포인트 기반 객체 탐지 모델을 이용한 데이터 시각화 왜곡 탐지 및 교정 시스템 개발

김예지*, 남궁기태**, 최영***, 이호형****, 양선우*****,이연진******,
예성철*******, 유길상********

Development of a Data Visualization Distortion Detection and Correction System using a Keypoint-based Object Detection Model

Yeji Kim*, Gitae Namgung**, Young Choi***, HoHyeong Lee****, Sunoo Yang*****,
Yeonjin Lee******, SungChul Yea*******, and Gilsang Yoo********

This research was supported by Basic Science Research Program through the
National Research Foundation of Korea(NRF) funded by the Ministry of Education(RS-2023-00246191)

요 약

오늘날 데이터 시각화는 빅데이터 분석에서 중요한 의사결정 도구로 활용되며, 광고, 마케팅, 뉴스 보도 등 다양한 분야에서 사용되고 있다. 그러나 그래프는 제작자의 의도에 따라 쉽게 변형될 수 있어 왜곡의 위험이 존재하며, 이는 소비자나 사용자에게 잘못된 정보를 제공해 오해를 초래할 수 있다. 본 연구에서는 이러한 문제를 해결하기 위해 왜곡된 차트를 탐지하고 교정된 차트를 제공하는 시스템을 제안하였다. 제안된 시스템은 Hourglass 네트워크 기반의 keypoint 객체 탐지 모델을 활용하여 막대, 선, 원 그래프 유형에서 시각적 요소의 위치를 추출하고, 이를 바탕으로 왜곡을 탐지하고 교정하는 알고리즘을 적용하였다. 실험 결과, 그래프 탐지 검증 정확도는 98%의 우수한 성능을 보였다. 구현 결과 다양한 웹 시각 자료에서 높은 정확도로 왜곡을 탐지하고 교정된 결과를 제공하여 데이터 시각화의 신뢰성을 크게 높일 수 있을 것으로 기대한다.

Abstract

Recently, data visualization serves as a critical decision-making tool in big data analysis and is widely utilized across fields such as advertising, marketing, and news reporting. However, graphs can be easily manipulated based on the creator's intent, posing risks of distortion that may mislead consumers or users. This study proposes a system designed to detect and correct distorted charts to address this issue. The proposed system employs a keypoint object detection model based on the Hourglass network to extract the positions of visual elements in bar, line, and pie charts, applying algorithms to detect and correct visual distortions. Experimental results demonstrate a graph detection accuracy of 98%. The implementation is expected to significantly enhance the reliability of data visualization by accurately detecting and correcting distortions in various web-based visual materials.

Keyword

data visualization, chart analysis, graphical deception correction, key point detection

* 워싱턴대학교 수학과
- ORCID: https://orcid.org/0009-0007-0481-6145
** 가천대학교 일반대학원 나노과학기술융합학과
- ORCID: https://orcid.org/0000-0001-8864-0667
*** 서울여자대학교 소프트웨어융합학부
- ORCID: https://orcid.org/0009-0003-5288-2666
**** 고려대학교 철학과
- ORCID: https://orcid.org/0009-0008-1505-9618
***** 서강대학교 심리학과
- ORCID: https://orcid.org/0009-0005-4674-6419
****** 고려대학교 지능정보SW아카데미
- ORCID: https://orcid.org/0009-0008-8521-0651
******* 고려대학교 공과대학 화공생명공학과
- ORCID: https://orcid.org/0009-0003-3595-5085
******** 고려대학교 정보대학 정보창의교육연구소(교신저자)
- ORCID: https://orcid.org/0009-0002-1085-5355

· Received: Oct. 07, 2024, Revised: Nov. 29, 2024, Accepted: Dec. 02, 2024
· Corresponding Author: Gilsang Yoo
Creative Informatics & Computing Institute, 145 Anam-ro, Seongbuk-gu, Seoul, S.Korea
Tel.: +82-2-3290-1674, Email: ksyoo@korea.ac.kr

Ⅰ. 서 론

현대 사회에서 데이터 시각화는 복잡한 정보로부터 유의미한 가치를 쉽게 전달할 수 있는 강력한 도구로 자리 잡고 있다. 그래프와 같은 시각적 자료는 데이터의 핵심 내용을 직관적으로 파악하게 하며, 다양한 분야에서 의사결정 과정에 중요한 역할을 수행한다[1][2]. 기업 보고서, 학술 논문, 언론 기사, 광고 등 다양한 매체에서 그래프는 필수적인 요소로 사용되며, 이러한 시각적 표현은 데이터를 더욱 효과적으로 전달하고 대중의 이해를 돕는 데 기여하고 있다[3][4]. 그러나 이러한 그래프가 잘못된 방식으로 사용되거나 고의로 왜곡될 경우, 정보 전달의 본래 목적을 벗어나 오해를 불러일으키고 잘못된 결정으로 유도할 수 있다[5].

그래프 왜곡은 시각적으로 명확해 보이는 정보라도 실제로는 잘못된 해석을 초래한다. 그래프를 통해 전달되는 정보는 제작자의 의도나 해석이 강하게 반영될 수 있으며, 데이터의 특정 측면을 부각하거나 축소함으로써 수용자가 인지하는 정보의 의미가 크게 달라질 수 있다[6]. 이러한 왜곡은 의도적인 조작뿐만 아니라 무지나 부주의로 인해 발생할 수도 있다. 대표적인 그래프 왜곡 방식으로는 축의 단위 조작, 부정확한 비율 사용, 데이터의 일부분만을 선택적으로 보여주는 방식 등이 있으며, 이와 같은 방식으로 표현된 그래프는 소비자가 편향된 해석을 유발할 수 있다. 왜곡된 그래프는 단순한 시각적 오류를 넘어 사회적, 경제적, 정치적 문제로 확산될 가능성이 있다. 특히 광고, 정치 캠페인, 투자 보고서 등에서 그래프가 의도적으로 왜곡될 경우, 대중의 신뢰를 잃을 뿐만 아니라, 잘못된 정보를 바탕으로 중요한 결정을 내리게 하는 결과를 초래할 수 있다. 이러한 문제의 심각성을 인지한 일부 국가들은 광고에서의 그래프 왜곡을 규제하는 법적 장치를 마련하였으며, 우리나라에서도 대선 캠페인에서의 그래프 왜곡을 방지하기 위한 법안이 제안되는 등, 그래프 왜곡에 대한 인식과 대응이 점차 확대되고 있다.

그러나 그래프 왜곡을 법적으로 규제하는 데는 여러 가지 어려움이 따른다. 그래프는 제작자의 가치 판단에 의해 시각적으로 표현되며, 언제든 "오해의 소지"를 불러일으킬 수 있는 요소가 존재한다. 또한 왜곡된 그래프와 그로 인한 피해 사이의 인과관계를 명확히 규명하기 어렵고, 법적으로 피해 사실을 입증하기도 쉽지 않다. 결국 그래프가 데이터의 본질을 전달하기 위한 도구로 사용되지 않고, 제작자의 의도를 반영한 왜곡된 정보로 변질될 때, 그 피해는 고스란히 소비자에게 돌아간다.

이러한 문제를 해결하기 위해서는 그래프의 왜곡을 자동으로 탐지하고 수정할 수 있는 기술적 접근이 필요하다. 그래프를 단순히 '사용자' 관점에서 제작하고 해석하는 것이 아니라, '데이터' 관점에서 바라볼 수 있도록 데이터의 본질을 유지하면서도 정확하고 객관적인 시각적 표현이 요구된다[7]. 이에 따라 본 논문에서는 왜곡된 그래프를 자동으로 탐지하고 이를 수정하기 위한 새로운 알고리즘과 모델을 제안하였다. 제안한 기법은 다양한 시각적 왜곡 패턴을 인식하고 이를 기반으로 그래프 왜곡의 다양한 형태를 체계적으로 분류한 후, 각 형태에 따른 탐지 기법을 적용하였다. 또한, 탐지된 왜곡을 수정하여 데이터의 본질을 전달하는 객관적이고 중립적인 그래프를 생성하는 방법을 제시하였다.

본 논문의 구성은 다음과 같다. 2장에서는 시각화와 관련된 기존 연구를 기술하였으며, 3장에서는 제안한 시각화 왜곡 탐지 및 교정 시스템을 소개하였다. 4장에서는 실험 결과를 분석하였으며, 5장에서는 본 연구의 기대 효과 및 향후 연구 방향에 대해 논의하였다.

Ⅱ. 관련 연구

데이터 시각화는 복잡한 데이터를 시각적으로 표현하여 사용자들이 데이터를 이해하고 의사결정을 내리는 데 중요한 역할을 한다. 특히, 빅데이터 시대의 도래와 함께 데이터 시각화 기술은 광고, 마케팅, 언론 보도 등 다양한 분야에서 폭넓게 활용되고 있으며, 정보의 직관적 전달이라는 장점을 가지고 있다. 그러나 데이터 시각화의 이러한 유용성에도 불구하고, 시각적 왜곡은 데이터의 본질을 왜곡시켜 잘못된 정보 해석을 유도할 수 있는 위험 요소로 작용한다. 따라서 데이터 시각화의 정확성과 신뢰성을 높이기 위한 연구가 활발히 이루어져 왔다[8][9].

2.1 데이터 시각화의 중요성 및 왜곡 문제

데이터 시각화의 중요성은 정보의 직관적 전달에 있다. W. S. Cleveland and R. McGill은 그래프와 같은 시각적 표현이 수치 데이터를 효율적으로 전달하는 데 유리하다고 주장하며, 인간의 시각적 인지가 수치적 인지보다 더 빠르고 정확하다는 점을 강조하였다[10]. 또한 S. Few는 데이터 시각화가 정보 탐색, 데이터 패턴 식별, 데이터 비교 및 트렌드 분석에 필수적인 도구임을 제시하였다[11]. 이러한 장점에도 불구하고, 데이터 시각화는 제작자의 의도나 기술적 한계로 인해 시각적 왜곡이 발생할 수 있다.

이와 관련하여 E. R. Tufte는 '그래픽 무결성(Graphical integrity)'의 개념을 도입하며, 그래프 제작 시 데이터의 정확한 전달을 위해 데이터와 시각적 표현의 비율이 왜곡되지 않아야 한다고 강조하였다[12]. 이후 많은 연구자들이 Tufte의 개념을 확장하여 시각적 왜곡을 체계적으로 분석하였다. J. Heer and M. Bostock은 시각화 도구의 발전이 오히려 비전문가들이 왜곡된 그래프를 생성하는 위험성을 높일 수 있음을 지적하였다[13]. 이를 해결하기 위해 최근에는 왜곡 탐지 및 교정 알고리즘 개발에 대한 연구가 시도되고 있다.

2.2 그래프 왜곡 탐지 및 교정 기술

데이터 시각화에서 발생하는 왜곡 문제를 해결하기 위해 여러 연구가 다양한 접근 방법으로 시도되었다. 시각적 왜곡 탐지 분야에서 초기 연구들은 주로 비율 왜곡, 축 왜곡, 색상 왜곡 등의 특정 유형의 왜곡을 탐지하는 데 중점을 두었다. J. Talbot et al.은 비율 왜곡을 탐지하기 위해 차트의 시각적 요소와 데이터 값의 비율 차이를 분석하는 기법을 제안하였으며, L. Bartram et al.은 색상 왜곡을 줄이기 위해 시각적 대비를 최적화하는 방법을 개발하였다[14][15].

4차산업혁명 이후, 머신러닝 및 딥러닝 기술의 발전과 함께 왜곡 탐지 알고리즘에도 인공지능 기반 접근이 도입이 요구되고 있다.

2.3 Hourglass 네트워크 기반의 그래프 왜곡 교정 연구

Hourglass 네트워크는 객체의 위치 및 형태 정보를 정확하게 예측할 수 있는 딥러닝 모델로, 주로 인간의 포즈 추정, 얼굴 인식 등의 분야에서 사용되고 있다. 데이터 시각화 분야에서는 그래프 내 시각적 요소의 위치를 정밀하게 추출하여 왜곡을 탐지하고 교정하는 데 적용될 수 있다[16]. 특히, Hourglass 네트워크는 계층적 구조를 이용하여 시각적 객체의 위치를 예측하고, 이를 바탕으로 다양한 시각적 왜곡 유형을 탐지하는 데 유리한 특성을 가진다.

ChartReader는 다양한 차트를 복원하고 이해하는 데 있어 규칙 기반 접근 없이 동작하는 통합 프레임워크이다[17]. 이 프레임워크의 핵심은 Key point 기반 객체 탐지 모델로, 합성곱 신경망을 활용하여 차트의 중요한 키포인트를 탐지한다.

ChartOCR은 차트 이미지로부터 데이터를 추출하는 또 다른 프레임워크로, CornerNet의 변형된 버전을 기반으로 범용 키포인트 탐지 모델을 사용한다[18]. CornerNet은 객체 검출 분야에서 혁신적인 접근 방식을 제시한 딥러닝 모델로, 코너(Keypoint)를 예측하여 객체의 위치를 파악한다. 전통적인 객체 검출 모델들이 바운딩 박스(Bounding box) 전체를 예측하는 것과 달리, CornerNet은 객체의 상단 왼쪽(Top-left) 코너와 하단 오른쪽(Bottom-right) 코너를 예측하고, 이 두 코너를 쌍으로 묶어 객체를 검출한다. 특히 코너 풀링 기법을 적용하여 CNN (Convolutional Neural Networks) 모델의 과적합 문제를 해결하고, 차트 유형 분류를 위한 맥스 풀링을 활용한다.

ChartReader와 ChartOCR 모두 차트의 키포인트 탐지 및 그룹화에 Hourglass 네트워크를 사용한다. Hourglass 네트워크는 특징 맵을 축소 및 확장하면서 다양한 크기의 특징을 탐지하는데, 이 과정을 통해 차트의 중요한 구성 요소를 효율적으로 식별한다[16]. 키포인트 탐지 과정에서 Hourglass 네트워크는 반복적인 함수를 통해 키포인트를 탐지하고, Attention 기법을 통해 관련 키포인트들을 그룹화하여 차트의 구성 요소를 형성한다.

또한, 시그모이드 정규화와 같은 기법을 사용하여 히트맵에서 발생하는 노이즈를 줄이고, 키포인트의 정확한 위치를 더 잘 표현할 수 있도록 한다.

III. 시각화 왜곡 탐지 및 교정 시스템

본 연구에서 제안한 전체 시스템의 흐름은 그림 1과 같다. 웹에서 수집된 이미지가 입력되면 MobileNetV2[19]를 사용해 그래프와 비그래프 이미지로 분류한다. 그래프로 확인된 이미지는 Chart Reader로 전달되어, 그래프의 구성 요소를 탐지하고 각 요소의 위치를 파악하게 된다.

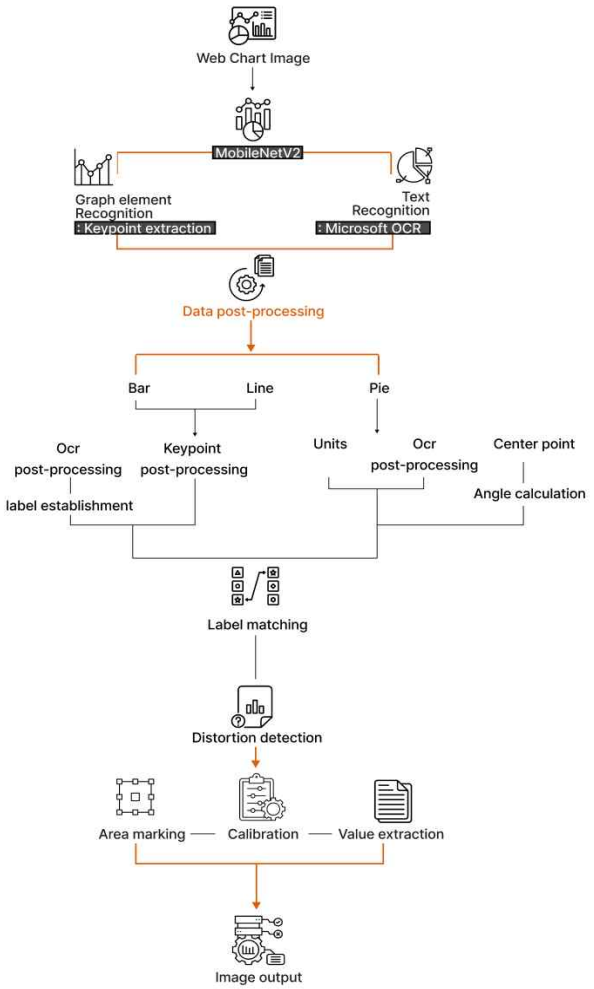


그림 1 . 제안한 모델의 전체 시스템 흐름도
Fig. 1. Overall flowchart of the proposed model

이 과정에서 그래프의 주요 포인트들이 추출되며, 텍스트도 개별적으로 처리하여 인식한다. 이후 그래프의 유형별로 왜곡을 탐지하는 알고리즘을 통해 왜곡 여부를 확인한 뒤, 왜곡된 요소를 파악하고 교정하여 최종 결과물에 반영된다. 왜곡이 발견된 그래프는 교정된 그래프와 함께 왜곡된 부분을 시각적으로 표시하여 사용자에게 제공된다.

3.1 그래프 탐지

MobileNetV2는 ImageNet으로 학습된 경량화 모델로, 높은 정확도와 빠른 계산 속도를 제공한다[20]-[22]. 본 연구에서는 이 모델을 전이 학습하여 그림 2와 같이 웹상의 다양한 이미지를 자동으로 그래프와 비그래프 두 가지 클래스로 이진 분류한다.

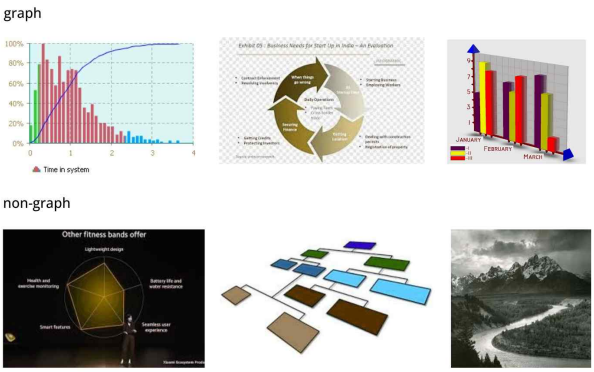


그림 2. 다양한 시각화 그래프 예
Fig. 2. Examples of various visualization graphs

각 그래프 유형별로 약 1,200개의 막대 그래프, 선 그래프, 원 그래프를 그래프 데이터로 수집하였고, 일반 사진 이미지 약 2,200개, 흐름도와 표 각각 800개를 비그래프 데이터로 수집하였다. 이 데이터는 8:2 비율로 학습 데이터와 검증 데이터로 나누어 사용하였다. 이때 rescale, rotation, shift, flip과 같은 다양한 데이터 증강 기법을 사용하여 모델의 일반화 능력을 향상시키고, 다양한 변형된 데이터에 대한 견고성을 높여 성능을 개선하였다.

Inverted Residual Block을 통해 특징을 효율적으로 추출하였다. 깊이별 분리 합성곱을 활용하여 계산의 효율성을 높이면서, 그래프의 중요한 시각적 특징을 효과적으로 포착한다. 예를 들어, 선과 막대, 원 그래프의 호와 같은 미세한 차이를 정확히 인식한다. 이 블록의 잔차 연결은 정보 손실을 최소화하여 그래프와 비그래프 (흐름도 및 표) 간의 미세한 차이를 더 정확하게 구별한다.

또한, Linear Bottleneck Layer는 선형 변환만을 사용하여 레이어 간 정보 손실을 줄이며, 그래프의 구조적 패턴과 비그래프의 다채로운 요소를 효과적으로 구별한다.

ImageNet[23]으로 사전 학습된 모델을 사용한 학습에서 복잡한 이미지에 대해서는 마지막 레이어만을 학습시키는 것이 높은 정확도를 보였지만, 비그래프 (흐름도 및 테이블)와 같은 구조적 이미지에서는 정확도가 낮게 나타났다. 이러한 이유로, 모델의 100번째 레이어부터 파인튜닝을 진행하였다. 학습률은 0.001로 시작하여 검증 손실이 5 에포크 동안 개선되지 않을 경우 학습률을 20%씩 감소시켰다. 이때, 학습률의 최소값은 0.00001로 제한하였다. 이렇게 식별된 그래프 이미지는 이후 Chart Reader 모델로 전달되어 차트 구성요소를 추출한다.

3.2 차트 구성요소 추출

그래프의 구성 요소를 효과적으로 감지하기 위하여 ChartReader 플랫폼을 사용하였다. 이 모델은 그래프의 key point를 기반으로 구성 요소를 추출하고 인식하는 모델로 Hourglass 네트워크를 통해 각 그래프 요소를 재구성하는 우수한 성능을 보인다.

3.2.1 차트 데이터 수집

Bar, Pie, Line 그래프를 포함한 총 11만 개의 데이터를 가진 EC400K 데이터셋을 채택하였다. EC400K 데이터셋은 실제 그래프와 유사한 구조를 가지며, 다양한 그래프 유형에 대한 학습에 적합하다[24].

3.2.2 데이터 증강

데이터의 다양성을 증가시키기 위해 두 가지 데이터 증강 기법을 사용하였다. 첫째, 그림 3와 같이 color jittering 기법을 적용하여 색상, 밝기, 채도, 명암의 변화를 모델이 견딜 수 있도록 하였다. 둘째, 그림 4와 같이 grid mask 기법을 활용하여 그래프의 일부 요소가 가려지더라도 모델이 정상적으로 요소를 인식할 수 있도록 하였다. 이러한 전처리 기법들은 다양한 그래프 상황에서 모델의 성능을 높인다.

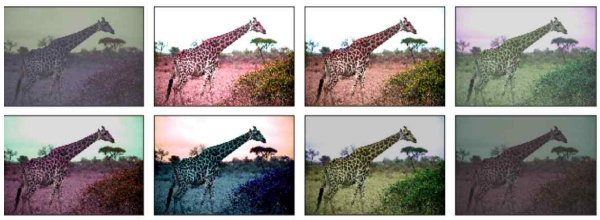


그림 3. 데이터 증강 기법 (color jittering)
Fig. 3. Data augmentation technique using color jittering

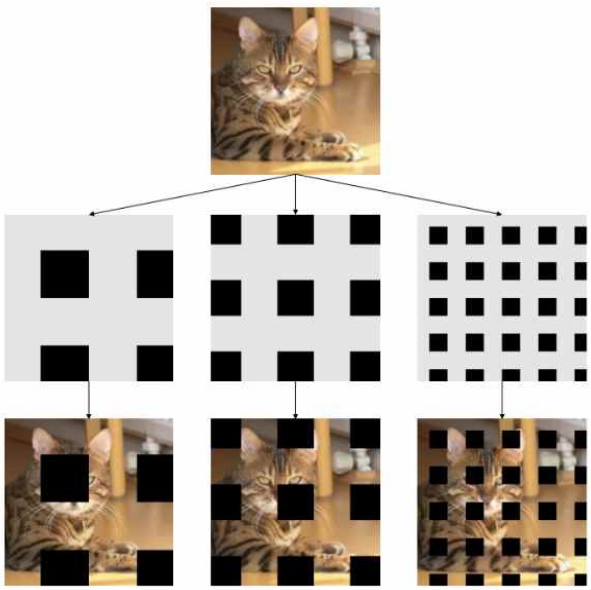


그림 4. 데이터 증강 기법 (grid mask)
Fig. 4. Data augmentation technique using grid mask

Chart Reader는 그림 5와 같이 다양한 크기의 feature를 반영하기 위해 stacked 구조를 가지며, skip layer를 사용하여 정보 손실을 최소화한다.

Key Point Detection 단계에서는 Hourglass 네트워크를 통해 key point를 추출한다. 이 과정에서 중요한 의미를 가지는 key point를 정확히 찾아내어 그래프 요소를 인식하는 데 필요한 정보를 제공한다.

Key Point Grouping 단계에서는 추출된 key point들은 중심점과의 연관성을 검사하여 그룹화한다. Attention 기법을 활용하여 요소 간의 연관성을 분석하고, 연관성이 높은 key point들을 그룹화하여 하나의 객체로 인식한다.

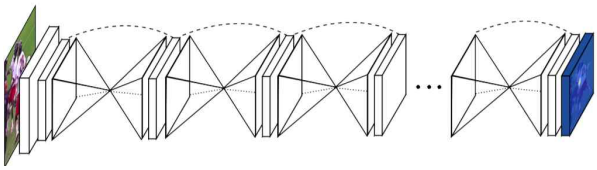


그림 5. Hourglass 네트워크 시각화 과정
Fig. 5. Hourglass network visualization process

마지막으로 객체 재구성 단계에서는, 그룹화된 key point를 바탕으로 그래프의 구성 요소가 재구성된다. Bounding Box를 이용한 전통적인 객체 인식 모델로는 구성 요소를 정확히 재구성하기 어려운 단점을 key point 추출 방식으로 해결하였다.

3.2.3 모델 학습

모델 학습은 다음과 같은 단계로 진행한다. 표 1의 내용과 같이 Key Point Detection 모델을 학습하였다. 사전 학습된 Chart OCR 모델을 활용하여 처음 5000 iteration까지는 전이 학습을 진행하였고, 이후 5만 iteration까지 전체 모델 학습을 진행하였다. Learning rate는 0.00025에서 0.000025로 감소시켰다.

표 1. Key point detection 모델의 하이퍼파라미터 및 성능지표
Table 1. Hyperparameters and performance metrics of the key point detection model

Dataset	EC400K
Iteration	50,000
Learning rate	0.00025 ⇒ 0.000025
Train loss	0.378
Validation loss	0.441

3.3 막대 및 선그래프 왜곡 탐지 알고리즘

막대와 선 그래프는 데이터를 높이로 표현하는 공통점을 가지고 있으며, 이를 활용하여 데이터 왜곡을 탐지하고 교정하는 알고리즘을 제안하였다.

막대와 선 그래프 알고리즘의 경우 두 가지 경우로 나뉜다. y축이 있는 경우 y축의 시작점이 0인지 확인하고, 축 눈금의 비율이 일정한지 검토하며, y축 중간에 삭제된 부분이 없는지 확인한다. y축이 없는 경우 OCR을 통해 추출한 값과 막대의 높이 비율이 일치하는지를 검토하여 데이터의 신뢰성을 보장한다. 비율 기반의 왜곡 탐지 알고리즘은 다음과 같이 수행된다.

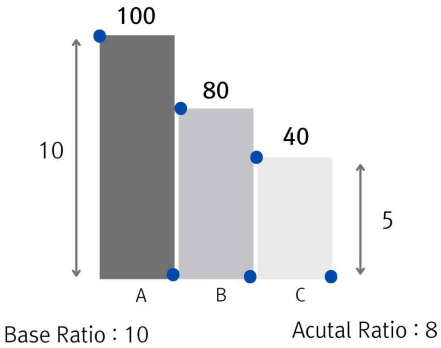


그림 6. 기준 비율 및 실제 비율 계산
Fig. 6. Calculation of base ratio and actual ratio

그림 6을 예시로 왜곡 탐지 알고리즘은 높이가 가장 높은 막대를 기준으로 높이와 실제값의 비율을 지정한다. 이후 모든 막대의 높이와 실제값을 저장하고 사용자가 임의로 지정한 비율 기반 오차비율 식 (1)에 기반 하여 그래프의 왜곡을 판별한다. 그림 6의 가장 높은 막대의 높이는 10, 실제값은 100으로 나타나고 가장 짧은 막대는 5의 높이와 40의 실제값을 지니는데 높이의 비율은 2:1 실제값의 비율은 5:2로 임의 지정한 오차율 (기본값 5%)을 초과하기에 해당 그래프는 왜곡되었다는 결과값을 출력하게 된다. 막대, 선 그래프에 대한 탐지 알고리즘은 표 2와 같다.

$$error\ rate = \frac{|base\ ratio - actual\ ratio|}{base\ ratio} \quad (1)$$

OCR 결과에서 숫자와 텍스트의 혼합으로 인한 인식 오류가 발생하는 문제를 해결하기 위해, OCR 값의 후처리가 필요하다. 그림 7에서처럼 날짜 레이블에서 특정 연도가 누락 되거나 숫자 사이의 간격이 일정하지 않은 문제를 해결하기 위한 알고리즘은 표 2와 같다.

1) 특수문자 제거 및 숫자 추출: OCR 결괏값에서 특수문자를 제거하고 공백으로 분리한다.

2) 공통 접미사 추출: 공통된 접미사를 찾아서 두 번 이상 나타나는 경우 공통 접미사 리스트에 추가한다.

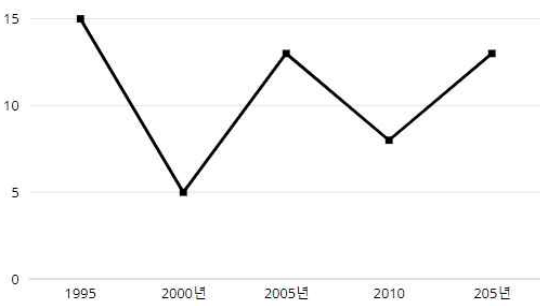
3) 데이터 일관성 검토: OCR의 bbox와 저장된 숫자들을 비교하여 누락된 숫자들을 추가한다.

4) 레이블 재구성: 공통 접미사를 저장된 숫자들의 끝에 결합하여 레이블을 재구성한다.

후처리 결과, 특수문자가 제거되고 누락된 데이터가 채워지고 이를 통해 데이터의 정확성이 크게 향상된다.

표 2. 막대, 선 그래프 왜곡 탐지 알고리즘
Table 2. Bar and line graph distortion detection algorithm

```
# find distortion based on len
def is_distorted(data):
    # Set the longest bar as the reference
    get base, base_len, base_ocr, base_ratio

    # Output results and determine distortion
    threshold = 0.05  # 5% error margin

    for each result in matching_result:
        get point_name, ocr, point_coord, point_len

        # Calculate ratio
        if point_len is not 0:
            ratio = ocr/ point_len
        else:
            ratio = 0

        # Calculate error margin
        if base_ratio is not 0:
            error = abs(ratio-base_ratio)/base_ratio
        else:
            error = 0

        # Determine distortion
        if error > threshold:
            return True
        return False
```



label = [1995, 2000년, 2005년, 2010, 205년]
suffixes_dict = [1995, 2000, 2005, 2010, 205]
common_suffixes = ['년']
reconstruction_label = [1995년, 2000년, 2005년, 2010년, 2015년]

그림 7. OCR 후처리 예
Fig. 7. Example of OCR Post-processing

막대 그래프와 동일하게 선 그래프도 가장 높게 위치한 지점에서 식 (1)과 표 3과 같이 비율 기반 알고리즘으로 왜곡을 탐지한다. 이 방법은 데이터의 상대적 차이를 직관적으로 이해할 수 있으며, 스케일의 영향을 받지 않으므로 데이터 세트 간의 비교가 일관성을 유지할 수 있다.

표 3. 막대, 선 그래프 데이터 교정 알고리즘
Table 3. Bar and line graph data correction algorithm

```
function find_common_suffixes(labels):
    suffixes = empty list

    # Extract suffixes
    for each item in labels:
        add item to suffixes if item is not a digit

    # Count suffix frequencies
    element_counts= count frequency of each in suffixes
    common_suffixes= empty list
    for each element and its count in element_counts:
        add element to common_suffixes if count is 2 or
more

    # Initialize suffix dictionary
    suffixes_dict= empty dictionary
    for each common_suffix in common_suffixes:
        suffixes_dict[common_suffix] = empty list

    # Fill suffix dictionary
    for each common_suffix in common_suffixes:
        for each item in labels:
            if common_suffix is in item:
                add item to suffixes_dict[common_suffix]
            else:
                add None to suffixes_dict[common_suffix]

    return common_suffixes, suffixes_dict

function update_suffixes_dict(suffixes_dict):
    for each suffix in suffixes_dict:
        for each item value and index in
suffixes_dict[suffix]:
            if value is None:
                if suffix is year:
                    suffixes_dict[suffix][index]
                    = suffixes_dict[suffix][index - 1]
                if suffix is month:
                    previous_value=
suffixes_dict[suffix][index - 1]
                ...
    return suffixes_dict
```

3.4 원 그래프 왜곡 탐지 알고리즘

원 그래프의 면적 왜곡 문제를 효율적으로 탐색하기 위해 각도를 활용한 알고리즘은 다음과 같다. 먼저 각 조각의 각도를 계산하기 위해 중심점, 중앙점, 외각점의 정보를 이용하는데, KeyPoint 결과에서는 첫 번째 좌표가 중앙점을 나타내며, 이후 좌표들은 무작위로 배치된다. 특히, 원 조각이 두 개일 경우, 중심점과 외각점이 두 번씩 나타나기 때문에 단순히 중복 횟수만으로는 중심점을 정확히 식별하기 어려운 문제가 발생한다. 이 문제를 해결하기 위해 그림 8과 같이 코사인 유사도를 활용한 중심점 탐색 알고리즘을 제안하였다. 두 중앙점이 외각점을 향할 때, 이들 벡터의 방향이 유사하여 높은 코사인 유사도가 나타난다. 반면, 두 중앙점이 서로를 향하는 경우, 방향이 상반되어 코사인 유사도가 낮아진다. 따라서, 낮은 코사인 유사도를 가지는 방향의 점을 중심점으로 식별할 수 있다.

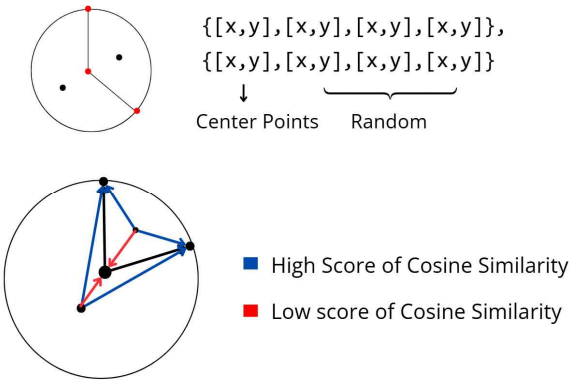


그림 8. 코사인 유사도를 활용한 중앙점 구분
Fig. 8. Distinguishing the central point using cosine similarity

이후, 주어진 중심점, 중앙점, 외각점을 활용하여 각 원 조각의 각도를 계산한다. 각도는 실제 값 또는 백분율과 비교하여 왜곡을 효과적으로 검출할 수 있는 지표로 작용한다. 중심점과 외각점을 벡터화하여 코사인 법칙을 적용하여 각도를 계산하지만, 180도 범위 내에서는 외각이 아닌 내각을 계산하는 문제가 발생한다. 이를 해결하기 위해 그림 9과 같이 외적을 이용하여 외각과 내각을 구분하는 알고리즘을 제안하였다. 벡터 ab, ac, ad를 생성한 후, ab와 ac, ab와 ad의 외적을 계산하고, 외적의 부호를 통해 방향성을 분석하여 외각과 내각을 정확히 구분한다. 외각의 경우 360도에서 세타를 빼고, 내각의 경우 세타를 사용하여 각 원 조각의 각도를 구한다.

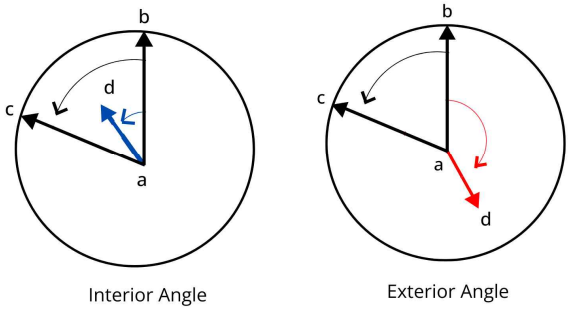


그림 9. 외적 부호 비교를 활용한 각도 구분
Fig. 9. Distinguishing angles using external sign comparison

계산된 각 조각의 각도, 실제 값 또는 백분율, 및 범례 간의 대응이 필요하다. 원 그래프는 막대 그래프나 선 그래프와 달리 범례의 위치가 비정형화 되어 있어 추가적인 범례 대응 알고리즘을 제안하였다. 탐지 과정은 다음과 같다.

1) 각 원 조각의 중앙점과 가장 가까운 숫자를 해당 값으로 대응
2) 위에서 구한 숫자와 가장 가까운 범례를 대응
3) 대응된 모든 요소를 통합.

1)의 경우, 하나의 원 조각에 여러 숫자가 대응되는 현상을 막기 위해 유클리디안 거리를 활용하여 가장 가까운 숫자 값만을 저장하고, 나머지 값의 경우 그다음으로 가까운 조각에 대응하도록 설계하였다. 2)의 경우, OCR(광학 문자 인식)이 가로 방향으로 텍스트를 인식하여 분리하기 때문에 범례가 한줄 이상으로 표기 된 경우, 숫자와 텍스트의 유클리디안 거리가 임계치 이내인 모든 값을 범례로 저장한다. 이때, 웹상의 그래프는 이미지 크기가 다양하여서 동적으로 임계치를 조정하여 대응의 정확도를 높였다.

대응된 각 조각의 각도가 실제 값의 비율 또는 실제 표기된 백분율과 일치하는지, 그리고 전체 백분율의 합이 100인지 검증하는 과정을 거친다.

표 4와 같이 원 그래프 왜곡 탐지 알고리즘은 원 그래프의 면적 왜곡을 효과적으로 탐지하며, 각 조각의 각도를 정밀하게 계산하여 데이터 왜곡을 판별한다. 코사인 유사도와 외적을 활용해 중심점과 각도를 정확히 식별하고, 유클리디안 거리와 OCR 기술을 통합한 범례 대응 알고리즘으로 데이터의 일관성과 정확성을 보장한다.

표 4. 원 그래프 왜곡 탐지 알고리즘
Table 4. Pie chart distortion detection algorithm

```
Function is_distorted(angles):
  Result as a list [0, 0]
  Initialize sum as 0

  For each item in angles
    Add real to sum

    If abs(percentage - real) > 2.0:
      Set result[0] to 1

  If sum is less than 98:
    Return True

  If result[0] is 1:
    Return True

  Else:
    Return False
```

3.5 교정된 그래프 생성 알고리즘

본 연구는 교정된 그래프와 원본 그래프에서 왜곡된 영역을 표시해 주는 두 가지 그래프를 생성한다. 교정된 막대 그래프의 경우 직선적인 특징 활용하여 왜곡 영역을 표시한다. 원그래프의 경우, 그림 10과 같이 12시 방향에서 시작하여 시계방향으로 가장 가깝게 위치한 외각점를 찾아 해당 외각점이 시작점이 된다.

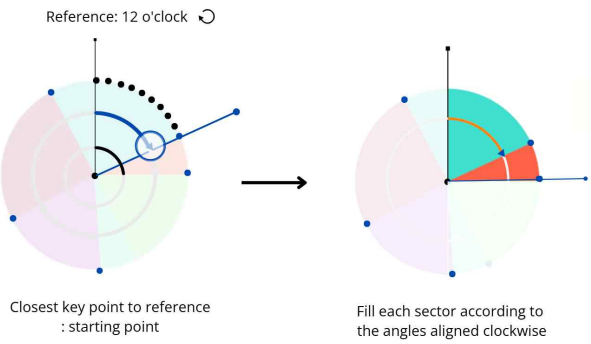


그림 10. 교정된 원 그래프 생성 과정
Fig. 10. Process of creating a corrected pie chart

이후 두번째 키포인트부터는 올바른 두번째 키포인트의 각도와 첫번째 키포인트의 각도를 계산해서 백분율을 구한다. 이후 부채꼴의 외각을 채워가며 영역을 표시한다.

Ⅳ. 실험 결과

4.1 그래프 식별 결과

MobileNetV2 모델의 학습 결과는 표 5와 같다. 학습 손실은 0.1644, 학습 정확도는 0.9329로 기록되었고, 검증 손실은 0.1867, 검증 정확도는 0.9800로 나타났다. 이 결과는 모델이 훈련 데이터에 대해 높은 성능을 보였으며, 검증 데이터에 대해서도 비교적 우수한 성능을 유지하고 있음을 나타낸다. 특히, 웹상에 존재하는 다양한 이미지에서 막대그래프, 선그래프, 원 그래프를 효과적으로 그래프로 분류할 수 있음을 보여준다.

표 5. MobileNetV2 모델의 학습결과
Table 5. Training results of the MobileNetV2 model

	Initial value	Final value
Training loss	0.7822	0.164
Training accuracy	0.4531	0.9329
Validation loss	0.1867	
Validation accuracy	0.9800	
Learning rate	0.001	0.00001 (Dynamic adjustment minimum value)

4.2 왜곡 탐지 결과

최종적으로 Key Point Detection 모델을 통해 차트의 key point와 중심점을 추출한다. 막대그래프에 대한 Key Point Detection 결과는 그림 11과 같다. 각각의 그래프 상단 왼쪽, 하단 오른쪽, 중앙점 등의 key point를 반환한다.

중심점과 key point 사이의 연관성 검사를 통해 그림 12와 같이 key point들이 그룹화되고, 각 객체별 id 값이 부여된다. 이후 그룹화된 요소들은 각각의 그래프의 왜곡 탐지 알고리즘에 의해 처리된다.

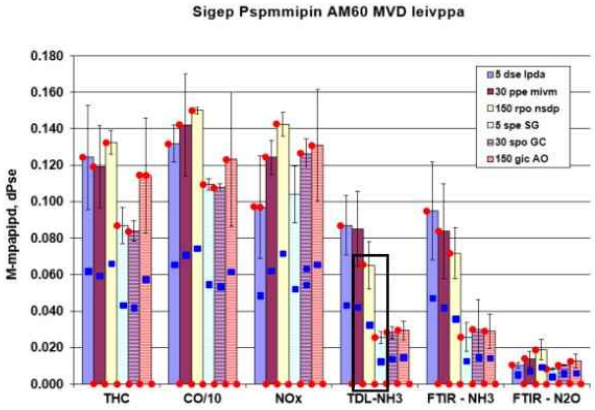


그림 11. Key point detection 결과 (막대 차트)
Fig. 11. Key point detection results (Bar chart)

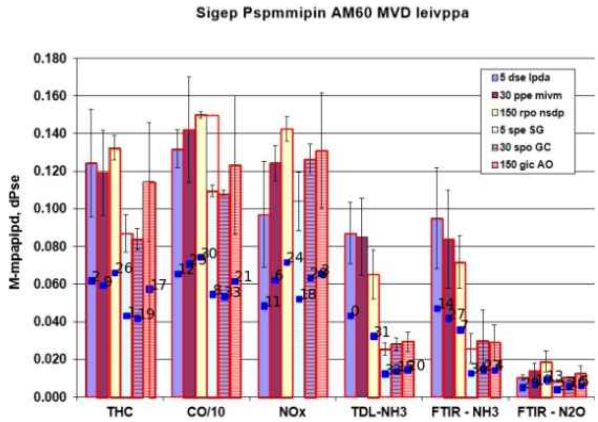


그림 12. Key point grouping 결과 (막대 차트)
Fig. 12. Key point grouping results (Bar chart)

원 그래프에 대한 Key Point Detection 결과와 그룹핑 결과는 각각 그림 12, 그림 13과 같다.

실험 결과, 제안한 모델은 다양한 차트 구성 요소를 효과적으로 감지하고 재구성하는 데 뛰어난 성능을 보여주고 있으며, 다양한 그래프 유형에 대한 유연한 대응이 가능한 것으로 확인되었다.

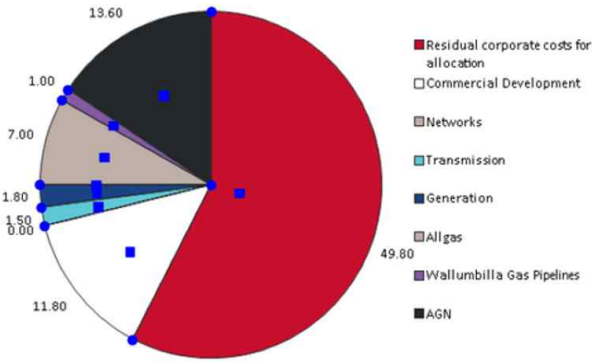


그림 13. Key point detection 결과 (원 그래프)
Fig. 13. Key point detection results (Pie chart)

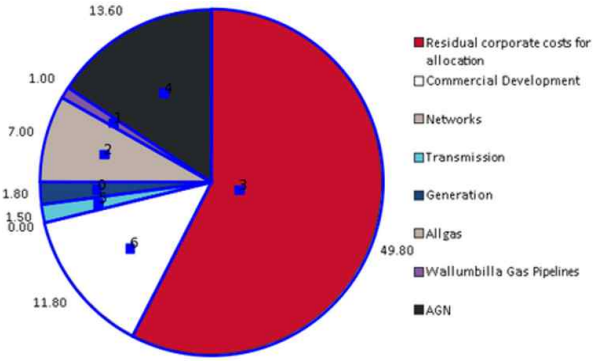
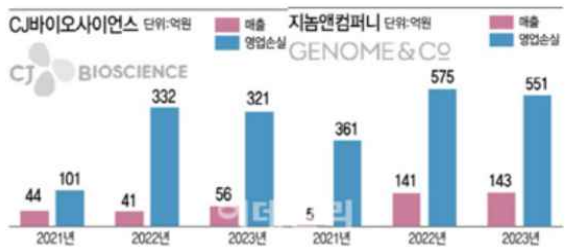


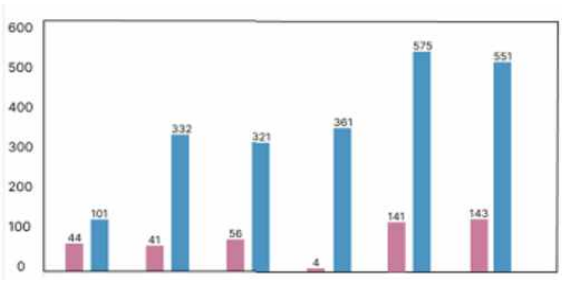
그림 14. Key point grouping 결과 (원 그래프)
Fig. 14. Key point grouping results (Pie chart)

4.3 교정 그래프 출력

왜곡 영역 표시본은 왜곡이 어디서 어떻게 발생하였는지 구체적이고 직관적으로 이해할 수 있다. 각각의 그림 15, 그림 16, 그림 17은 막대 그래프, 선 그래프, 원 그래프에 대한 교정 전, 후의 그래프를 보여주고 있다[25]-[27]. 이를 통해 사용자는 시각화 오류를 쉽게 확인할 수 있음을 확인하였다. 그림 17의 왼쪽 그림은 원본 영상에 교정된 부분의 차이를 붉은색으로 중첩한 결과로 실제 원본과 차이를 확인할 수 있다.



(a)



(b)

그림 15. 막대 그래프 (a)원본 (b) 교정 후 결과
Fig. 15. Graph (a) Original bar (b) Corrected result

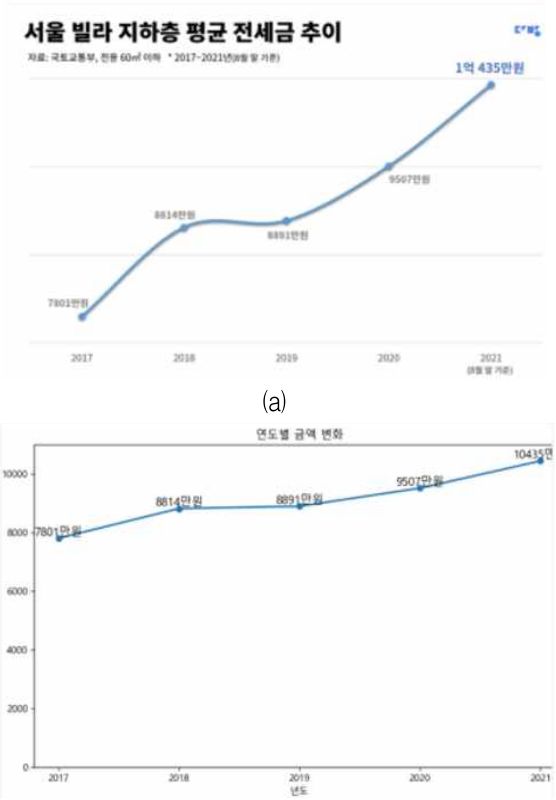


그림 16. 선 그래프 (a) 원본 (b) 교정 후 결과
Fig. 16. Line graph (a) Original (b) Corrected results

그림 17. 원 그래프 (a) 원본 (b) 교정 후 결과
Fig. 17. Pie chart (a) Original (b) Corrected results

4.4 시스템 구현

구현된 시스템은 크롬 브라우즈의 확장 기능(Chrome extension)을 제공하는 플러그인으로 개발되었으며, 웹사이트에서 그래프 이미지를 클릭하면 서버로 전송되어 모델과 알고리즘을 통해 왜곡 여부를 판단하고 그림 18과 같이 왜곡된 그래프와 별도로 실시간으로 교정된 그래프를 제공한다[25]. 구현 결과는 복잡한 설정 없이 사용자 친화적이며 다양한 웹사이트와 콘텐츠에 범용적으로 사용할 수 있다.

그림 18. Chrome extension으로 구현된 결과
Fig. 18. Chrome extension implementation results

V. 결론 및 향후 과제

데이터 시각화는 현대 사회에서 중요한 의사결정 도구로 활용되지만, 잘못된 사용이나 의도적인 왜곡은 잘못된 정보를 제공하여 다양한 편향을 유발할 수 있다. 이를 해결하기 위해 본 논문에서는 여러 그래프 유형에서 발생하는 왜곡을 자동으로 탐지하고 교정할 수 있는 시스템을 제안하였다. 제안된 시스템은 Keypoint 기반한 객체 탐지와 그래프 교정 알고리즘을 제안하였고, 이를 통해 막대, 선, 원 그래프에 적용하여 높은 정확도로 왜곡을 탐지하고 수정된 시각화 그래프를 제공하였다. 이로써 사용자들은 왜곡되지 않은 정보를 바탕으로 더 정확한 의사결정을 할 수 있도록 하였다. 본 연구는 데이터 시각화에서 발생할 수 있는 왜곡을 효과적으로 교정함으로써 데이터의 본질적 가치를 정확하게 전달할 수 있다. 앞으로의 연구는 더 다양한 그래프 유형을 시스템에 적용하고, 지연 없는 실시간 왜곡 탐지 및 교정 기능을 추가함으로써 데이터 시각화의 신뢰성과 객관성을 더욱 높일 수 있는 후속 연구가 진행될 예정이다.

References

[1] G. Richer, A. Pister, M. Abdelaal, J.-D. Fekete, M. Sedlmair, and D. Weiskopf, "Scalability in Visualization", IEEE Transactions on Visualization and Computer Graphics, Vol. 30, No. 7, pp.

3314-3330, Jul. 2024. https://doi.org/10.1109/TVCG.2022.3231230.

[2] H. Kim, "A Qualitative Study on Implementing Data Visualization Class in High School Art and Information Design Education", Doctoral dissertation, Seoul National University, 2018.

[3] J. Walny, C. Frisson, M. West, D. Kosminsky, S. Knudsen, S. Carpendale, and W. Willett, "Data Changes Everything: Challenges and Opportunities in Data Visualization Design Handoff", IEEE Transactions on Visualization and Computer Graphics, Vol. 26, pp. 12-22, Jan. 2020. https://doi.org/10.1109/TVCG.2019.2934538.

[4] G. Yoo, S. Choi, and H. Kim, "Visualization analysis of KOCW learning contents by major field before and after pandemic", Proc. of the Korean Society of Computer Education Conference, Yeosu, Korea, Vol. 17, No 1, pp. 363-366, Jan. 2023.

[5] W. Kang, J. Lim, H. Choi, and Newsjelly "Visualize data at a glance", Wikibooks, Paju, 2020.

[6] P. Parsons, "Understanding Data Visualization Design Practice", IEEE Transactions on Visualization and Computer Graphics, Vol. 28, No. 1, pp. 665-675, Jan. 2022. https://doi.org/10.1109/TVCG.2021.3114959.

[7] H. Son, Y. Han, K. Nam, S. Han, and G. Yoo, "Development of a News Trend Visualization System based on KPF-BERT for Event Changes and Entity Sentiment Analysis", The Journal of Korean Institute of Information Technology, Vol. 22, No. 1, pp. 203-213, Jan. 2024. https://doi.org/10.14801/jkiit.2024.22.1.203.

[8] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Big Data and Data Visualization Challenges", 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, pp. 6227-6229, Dec. 2023. https://doi.org/10.1109/BigData59044.2023.10386491.

[9] X. Qin, Y. Luo, N. Tang, and G. Li, "DeepEye: An automatic big data visualization framework", Big Data Mining and Analytics, Vol. 1, Vo. 1, pp. 75-82, Mar. 2018. https://doi.org/10.26599/BDMA.2018.9020007.

[10] W. S. Cleveland and R. McGill, "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods", Journal of the American Statistical Association, Vol. 79 No. 387, pp. 531-554, 1984. https://doi.org/10.1080/01621459.1984.10478080.

[11] S. Few, "Information Dashboard Design: The Effective Visual Communication of Data", O'Reilly Media, Inc., 2006.

[12] E. R. Tufte, "The Visual Display of Quantitative Information", Graphics Press, 1983.

[13] J. Heer and M. Bostock, "Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design", Proc. of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta Georgia USA, pp. 203-212, Apr. 2010. https://doi.org/10.1145/1753326.1753357.

[14] J. Talbot, V. Setlur, and A. Anand, "Four Experiments on the Perception of Bar Charts", IEEE Transactions on Visualization and Computer Graphics, Vol. 20, No. 12, pp. 2152-2160, Dec. 2014. https://doi.org/10.1109/TVCG.2014.2346320.

[15] L. Bartram, A. Patra, and M. Stone, "Affective Color in Visualization", Proc. of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17) Association for Computing Machinery, Denver Colorado USA, pp. 1364-1374, May 2017. https://doi.org/10.1145/3025453.3026041.

[16] S. Obeidavi, M. Gandomkar, and G. Hirtz, "In-Pose Estimation of Covered and Uncovered Human Body from Thermal Camera Images Using Multi-Scale Stacked Hourglass (MSSHg) Network", 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Dijon, France, pp. 84-09, Oct. 2022.

https://doi.org/10.1109/SITIS57111.2022.00021.

[17] C. Rane, S. M. Subramanya, D. S. Endluri, J. Wu, and C. L. Giles, "ChartReader: Automatic Parsing of Bar-Plots", 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI), Las Vegas, NV, USA, pp. 318-325, Aug. 2021. https://doi.org/10.1109/IRI51335.2021.00050.

[18] J. Luo, Z. Li, J. Wang, and C.-Y. Lin, "ChartOCR: Data Extraction from Charts Images via a Deep Hybrid Framework", 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, pp. 1916-1924, Jan. 2021. https://doi.org/10.1109/WACV48630.2021.00196.

[19] S. Bouskour, M. H. Zaggaf, and L. Bahatti, "Deep Learning Recognition of Wheat Leaf Disease Using MobileNetV2 Model", 2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), FEZ, Morocco, pp. 1-6, May 2024. https://doi.org/10.1109/IRASET60544.2024.10548207.

[20] L. Si, et al., "A Novel Coal-Gangue Recognition Method for Top Coal Caving Face Based on IALO-VMD and Improved MobileNetV2 Network", IEEE Transactions on Instrumentation and Measurement, Vol. 72, pp. 1-16, Oct. 2023. https://doi.org/10.1109/TIM.2023.3316250.

[21] K. Mittal, K. S. Gill, K. S. Aggarwal, R. S. Rawat, and S. Aluvala, "Using MobileNetV2 Deep Convolutional Neural Networks and Transfer Learning for Shell and Pebble Classification", 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, pp. 1-5, Apr. 2024. https://doi.org/10.1109/I2CT61223.2024.10544024.

[22] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 Model for Image Classification", 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, pp. 476-480, Dec. 2020. https://doi.org/10.1109/ITCA52113.2020.00106.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, pp. 248-255, Jun. 2009. https://doi.org/10.1109/CVPR.2009.5206848.

[24] J. Luo, Z. Li, J. Wang, and C.-Y. Lin, "ChartOCR: Data Extraction from Charts Images via a Deep Hybrid Framework", 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, pp. 1916-1924, Jun. 2021. https://doi.org/10.1109/WACV48630.2021.00196.

[25] Pharm Edaily, https://pharm.edaily.co.kr/news/read?newsId=01315286638918768. [accessed: Nov. 29, 2024]

[26] Economy, https://www.mk.co.kr/economy/view.php?sc=50000001&year=2021&no=865256. [accessed: Nov. 29, 2024]

[27] YonhapNews, https://www.yna.co.kr/view/AKR20240517026400004. [accessed: Nov. 29, 2024]

저자소개

김 예 지 (Yeji Kim)

2024년 6월 : 고려대학교 지능정보 SW아카데미 4기 수료(640H)
2023년 6월 ~ 현재 : 워싱턴 대학교 수학과 학사과정
관심분야 : 자연어 처리, 딥러닝, 멀티모달, 생체데이터분석

남궁 기태 (Gitae Namgung)

2019년 2월 : 가천대학교 일반대학원 나노과학기술융합학과 (석사)
2024년 6월 : 고려대학교 지능정보 SW아카데미 4기 수료(640H)
관심분야 : 딥러닝, 머신러닝, 데이터사이언스

최 영 (Young Choi)

2024년 6월 : 고려대학교 지능정보 SW아카데미 4기 수료(640H)
2024년 3월 ~ 현재 : 서울여자대학교 소프트웨어융합학과 학사과정
관심분야 : 딥러닝, 머신러닝, 데이터 사이언스

이 호 형 (HoHyeong Lee)

2024년 6월 : 고려대학교 지능정보 SW아카데미 4기 수료(640H)
2019년 3월 ~ 현재 : 고려대학교 철학과, 뇌인지과학 학사과정
관심분야 : 자연어 처리, 딥러닝, 머신러닝, 데이터사이언스

양 선 우 (Sunwoo Yang)

2024년 2월 : 서강대학교 심리학과(학사)
2024년 6월 : 고려대학교 지능정보 SW아카데미 4기 수료(640H)
2024년 9월 ~ : 고려대학교 일반대학원 심리학부 심리데이터과학 석사과정
관심분야 : 자연어 처리, 딥러닝, 심리 측정

이 연 진 (Yeonjin Lee)

2024년 6월 : 고려대학교 지능정보 SW아카데미 4기 수료(640H)
관심분야 : 딥러닝, 머신러닝, 데이터 사이언스

예 성 철 (SungChul Yea)

2021년 2월 : 고려대학교 화공생명공학과(공학사)
2023년 12월 : 고려대학교 지능정보 SW아카데미 4기 수료(640H)
관심분야 : 자연어 처리, 컴퓨터비전, 딥러닝, 머신러닝

유 길 상 (Gilsang Yoo)

2010년 2월 : 중앙대학교 영상공학과(박사)
2010년 3월 ~ 현재 : (사)한국컴퓨터게임학회 부회장
2011년 3월 ~ 현재 : 고려대학교 정보대학 정보창의교육연구소/지능정보 SW아카데미 교수
2023년 3월 ~ 현재 : (사)한국미디어 아트산업협회 수석부회장
관심분야 : 데이터사이언스, 데이터 시각화, 빅데이터 분석, 3D영상 콘텐츠, 머신러닝, 딥러닝, 컴퓨터교육