

코바늘 객체 인식 기반 기호도 자동 생성 연구

오장훈, 이송희, 김문기, 전휘빈, 김유현, 유진희, 권민규*

성균관대학교, 건국대학교, 동국대학교, 상지대학교, 런던대학교, 안양대학교, *서울대학교

orichardev@g.skku.edu, sh713@konkuk.ac.kr, rlaansrlssp@dgu.ac.kr, 201574031@sangji.ac.kr, yuhyeon.kim.20@alumni.ucl.ac.uk, 2020E7219@gs.anyang.ac.kr, *kmk0610@snu.ac.kr

Automatic Generation of Crochet Symbol Charts Based on Object Recognition

Oh Jang Hun, Lee Song Hee, Kim Moon Gi, Jeon Hwi Bin, Kim Yu Hyeon, Ryu Jin Hee, Kwon Min Kyu*

Sungkyunkwan Univ., Konkuk Univ., Dongguk Univ., Sangji Univ., University College London., Anyang Univ., *Seoul National Univ.

요약

본 연구는 코바늘 직물의 스티치를 객체로 인식해 기호도(symbol chart)로 자동 변환하는 가능성을 검증하였다. 295 장의 이미지를 기반으로 7 개 클래스, 13,861 개 스티치 데이터셋을 구축하고, YOLO 와 Detectron2 모델을 비교하였다. 실험 결과 YOLOv11x 가 F1-score 0.869 로 최고 성능을 보였으며, 탐지 결과를 정렬·보정해 기호도 생성을 시범 적용하였다. 본 연구는 코바늘 기호도의 디지털 자동화를 통해 전통적 수작업 의존성을 줄이고, 패턴 공유 및 창작의 효율성을 높일 수 있는 확장 가능성을 제시한다.

I. 서론

코바늘 도안은 완성된 작품을 기호(symbol)로 해석하여 작성되며, 이는 작품 제작의 필수 과정이지만 높은 숙련도와 많은 시간이 요구된다. 현재 기호도의 대부분은 개인 간 거래나 소규모 공유에 의존하고 있어 접근성 및 확장성이 부족한 상황이다. 또한 코바늘 도안 제작 도구의 부재로 제작 과정은 과도한 창작자의 수작업에 의존해 비효율적이고 오류 가능성이 있다.[1]

기존 연구는 주로 대바늘 뜨개질(knitting)의 기하학적 모델링이었으며[2], 일부 코바늘(crochet) 패턴 자동화 연구[3]는 새로운 무늬 창작에 국한되어 주어진 작품을 표현하는 기호도 생성의 연구는 없다. 본 연구는 이러한 상황에서 코바늘 기법별 객체 인식을 기반으로 코바늘 기호도 자동 생성을 위한 연구를 제안한다.

II. 본론

가. 데이터셋 구축

기존 공개 데이터셋(Crochet Stitch Detect CV Model, 72 장, 4 클래스)[4]은 해상도와 라벨 품질의 한계가 있어, 직접 Pinterest, 블로그, 상업 이미지 사이트 등에서 고해상도 이미지를 기반으로 총 295 장의 원본 데이터셋을 수집하였다. 각 이미지에는 평균 47 개 개별 코(stitch)가 포함되어 전체 13,861 개의 개별 코 데이터셋을 구축하였다. 개별 코의 종류는 기본적인

사슬뜨기, 한길긴뜨기, 짧은뜨기와 긴뜨기 및 두길긴뜨기는 앞/뒤 형태 차이를 반영하여 Figure1 과 같은 총 7 종류로 구분하였다.

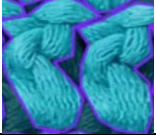
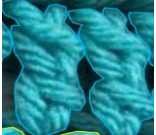
클래스	개수	클래스	개수
ch(사슬뜨기)		dc(한길긴뜨기)	
	2,725		2,333
sc(짧은뜨기)		tr_1(두길긴뜨기_앞)	
	3,612		1,342
hdc_1(긴뜨기_앞)		tr_2(두길긴뜨기_뒤)	
	1,458		1,215
hdc_2(긴뜨기_뒤)			
	1,176		

Figure 1. 코바늘 데이터셋 클래스별 객체 분포

나. 모델 선정 및 연구 프로세스 개요

본 연구는 코바늘 직물의 기호도 생성을 위해 4단계 절차를 구성하였다. 첫째, 객체 탐지 단계는 직물 이미지에서 스티치 단위를 인식하고 위치를 추출하는 과정이다. 둘째, 정렬 및 보정 단계는 탐지된 객체를 도안의 격자 구조에 맞게 배열하고, 위치 오류와 중복을 보정하는 과정이다. 셋째, 기호 매핑 단계는 정렬된 객체를 스티치 유형별 정의된 기호에 대응시키는 과정이다. 넷째, 도안 생성 단계는 변환된 기호를 체계적으로 배열하여 최종 기호도를 출력하는 과정이다.

이 중 객체 탐지 단계에서는 대표적 one-stage 계열인 YOLO와 two-stage 계열인 Detectron2를 비교 대상으로 선정하였으며, 두 모델이 코바늘 스티치 객체 탐지 및 기호도 변환 과정에 적용되는 적합성을 평가하였다.

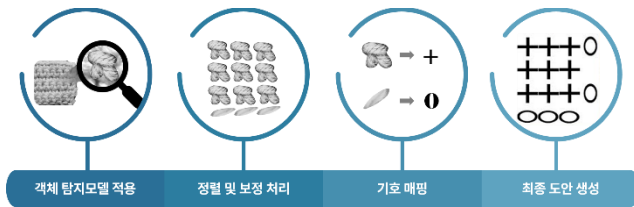


Figure 2. 연구 프로세스 개요 다이어그램

III. 실험 및 결과

가. 평가 지표 및 선정 기준

본 연구에서는 제일 핵심적인 각 코바늘 스티치 검출과 관련된 모델평가는 F1-Score 주된 지표로 하였으며, 세밀한 성능 차이를 설명하기에 부족한 MAP50 은 보조 지표로만 사용하였다.

나. 실험 절차 및 결과

1. 모델 성능 비교

여러 객체 탐지 모델들로 자체적으로 구축한 데이터를 기반으로 실험한 결과, 가장 성능이 좋게 나온 YOLO, Detectron 모델을 본 연구의 주 모델로 선정하였다. 그 결과 Figure 3 에서와 같이 YOLOv11x 의 모델 기반으로 수집된 데이터 증강을 활용한 모델이 F1-Score 0.869 성능을 보였다. 특히 hdc_1, hdc_2, dc, sc 등 구별이 난해한 클래스에 있어서도 Precision 과 Recall 이 모두 0.8 이상의 성능이 나와 이를 코바늘 이미지에 대한 패턴 인식 최종 모델로 선정하였다.

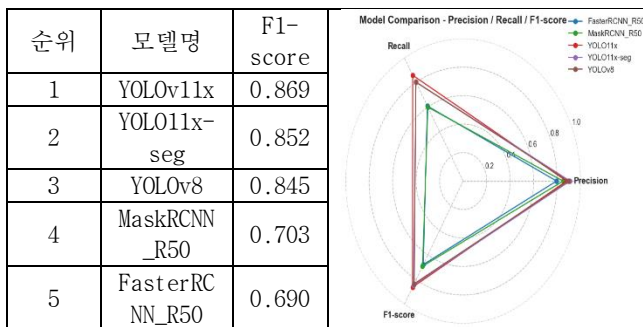


Figure 3. 성능 지표 그래프

2. 기호도 생성 시범 적용

탐지된 객체를 기호도로 변환하기 위해 먼저 스티치 좌표를 기반으로 층간 높이를 클러스터링하여 행 단위로 정렬하였다. 기호도는 규칙적 행·열 구조를 전제로 하므로, 실제 직물의 불규칙 배치를 보정할 필요가 있기 때문이다. 이후 누락된 스티치는 연속적인 패턴을 유지하기 위해 보간 처리로 채워 넣었으며 X 축 방향에서는 균등한 배열을 확보하기 위해 비거나 초과되는 개수를 보정하였다. 이렇게 보정된 스티치 행렬을 클래스별 기호(symbol)로 매핑하여 최종 기호도를 생성하였다(Figure 4).

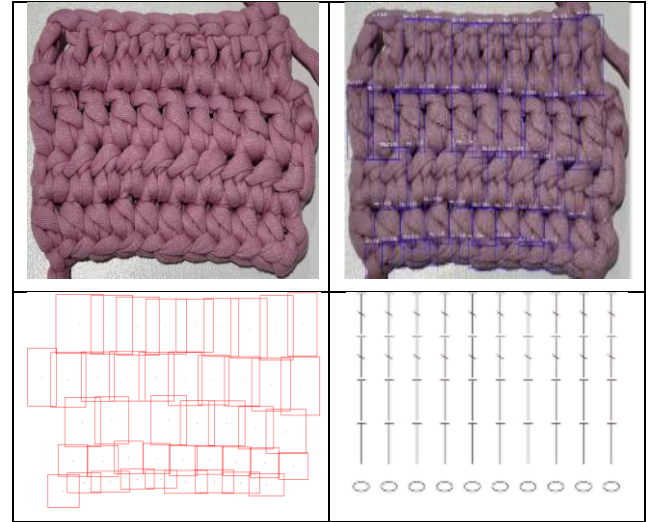


Figure 4. 기호도 변환 절차 (탐지-정렬-기호화)

IV. 결론

본 연구는 코바늘 직물의 스티치 단위를 객체로 인식하고 이를 기호도로 변환하는 자동화 가능성을 검증하였다. YOLO 와 Detectron2 계열 모델을 비교한 결과, YOLOv11x 가 가장 안정적인 성능을 보여 코바늘 직물의 세밀한 패턴 인식에 적합함을 확인하였다. 나아가 구축한 데이터셋과 탐지-정렬-기호화 과정은 단순한 성능 비교를 넘어, 코바늘 기호도의 디지털 보존과 자동 제작으로 확장될 수 있음을 시사한다. 향후에는 여러 스티치가 복합적인 직물을 기반으로 전체 도안을 자동 생성하는 시스템 구현으로 발전시킬 계획이다.

참 고 문 헌

- [1] Seitz, K., Rein, P., Lincke, J., Hirschfeld, R., "Digital Crochet: Toward a Visual Language for Pattern Description", In Proc. Onward! 2022 (ACM), pp. 48-62, 2022.
- [2] Gray, K., Bell, B., Sieper, D., Kobourov, S., Schreiber, F., Klein, K., Hong, S. "Wooly Graphs: A Mathematical Framework For Knitting," arXiv, (https://arxiv.org/abs/2407.00511), 2024.
- [3] Martinez, M., Lipnicki, A. T. "Automating Crochet Patterns for Surfaces of Revolution," arXiv, (https://arxiv.org/abs/2302.02205), 2023.
- [4] Roboflow, "Crochet Stitch Detect CV Model", https://universe.roboflow.com/akinorewek/crochet-stitch-detect (Accessed: 2025.08.16)