



# YOLO를 이용한 객체 탐색

# YOLO

- YOLO

- 실시간 객체 탐지(Object Detection)를 수행하는 딥러닝 기반 알고리즘
- 한 번의 신경망 연산으로 이미지에서 객체를 감지할 수 있도록 설계
- 기존의 RCNN 기반 모델보다 빠르고 효율적인 성능 제공

- YOLO의 특징

- 한 번만 처리 (One-Pass Detection)
  - 이미지를 한 번의 신경망(forward pass) 연산만으로 객체를 탐지
  - 기존 R-CNN 계열의 모델들은 후보 영역(Region Proposal)을 먼저 생성한 후 분류하는 방식이지만, YOLO는 이를 한 번에 해결하여 속도가 빠름
- 실시간 처리 가능
  - 다른 객체 탐지 모델보다 연산 속도가 훨씬 빨라 실시간 객체 탐지가 가능
  - 예를 들어, YOLOv8의 경우 CPU에서도 빠르게 동작하며, GPU에서는 초당 100 프레임 이상의 속도를 낼 수도 있음
- 그리드 기반 객체 탐지
  - 이미지를  $s \times s$  그리드로 나누고, 각 그리드에서 객체가 포함될 가능성이 있는 위치를 예측
  - 각 셀은 하나 이상의 객체를 탐지할 수 있으며, 해당 객체의 클래스 확률(Class Probability), 바운딩 박스(Bounding Box) 좌표, 신뢰도(Confidence Score)를 출력

- YOLO 버전

- YOLOv1(2016): 최초의 YOLO 모델, 빠르지만 정확도가 상대적으로 낮음.
- YOLOv2(2017), YOLOv3(2018): 성능 및 정확도 향상.
- YOLOv4(2020): 연구자들이 더욱 최적화하여 실시간 성능 개선.
- YOLOv5(2020): PyTorch 기반으로 설계되어 사용하기 쉬움.
- YOLOv6(2022), YOLOv7(2022): 더욱 최적화된 구조.
- **YOLOv8(2023)**: 최신 버전으로, 객체 탐지뿐만 아니라 세그멘테이션, 추적 기능도 포함됨

- YOLOv8의 모델

모델	크기	속도(FPS)	정확도(mAP@0.5:0.95)
YOLOv8n	Nano	빠름	낮음
YOLOv8s	Small	보통	보통
YOLOv8m	Medium	느림	높음
YOLOv8l	Large	더 느림	더 높음
YOLOv8x	XLarge	가장 느림	최고 성능

- 객체 탐지 모델 비교

모델	속도 (FPS)	정확도 (mAP)	방식
Faster R-CNN	느림	높음	RPN(Region Proposal Network)
SSD (Single Shot Detector)	중간	중간	멀티 스케일 예측
YOLOv3/v4/v5	빠름	중간~높음	그리드 기반 예측
YOLOv8	매우 빠 름	매우 높음	PyTorch 기반 최적화

- YOLO 활용분야

- 자율주행(Autonomous Driving) → 차량, 보행자 탐지
- 스마트 CCTV 및 보안(Surveillance) → 침입자 감지
- 의료 영상 분석(Medical Imaging) → 암세포 탐지
- 스포츠 분석(Sports Analytics) → 선수 움직임 추적
- 산업 자동화(Industry 4.0) → 로봇 비전 시스템



- YOLO 성능평가

- Precision (정밀도) : 모델이 탐지한 객체 중 실제로 맞는 비율
- Recall (재현율) : 실제 객체 중에서 모델이 정확하게 탐지한 비율

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

- TP(True Positive): 올바르게 탐지한 경우
- FP(False Positive): 잘못 탐지한 경우
- FN(False Negative): 놓친 객체

## • YOLO 성능평가

- IoU(Intersection over Union) 계산예측한 바운딩 박스와 실제 객체 박스의 겹친 정도를 나타냄

$$IoU = \frac{\text{예측박스} \cap \text{실제박스}}{\text{예측박스} \cup \text{실제박스}}$$

- 일반적으로  $IoU \geq 0.5$ 를 True Positive(TP)로 간주
- AP(Average Precision) 계산
  - Precision-Recall 곡선의 면적(Area Under Curve, AUC)을 계산
  - AP는 특정 클래스에서 Precision-Recall 곡선의 평균 정밀도를 나타냄

- mAP(Mean Average Precision)

- 모든 클래스의 AP를 평균하여 mAP 계산

$$mAP = \frac{1}{N} \sum AP_i$$

- 객체 탐지 모델의 성능을 평가하는 지표로, 모델이 얼마나 정확하게 객체를 탐지하고 분류하는지를 측정

- YOLO 성능평가

- mAP(Mean Average Precision) 예

클래스	AP (IoU 0.5)
사람	0.85
자동차	0.78
개	0.82

**mAP 계산:**

$$mAP = \frac{0.85 + 0.78 + 0.82}{3} = 0.816$$

- mAP@0.5 : IoU  $\geq 0.5$  일 때 AP 평균
    - 빠르고 직관적인 평가
  - mAP@0.5:0.95 : IoU 0.5~0.95 (0.05 간격)에서 평균
    - 더 엄격한 평가, 정확도 중요할 때 사용

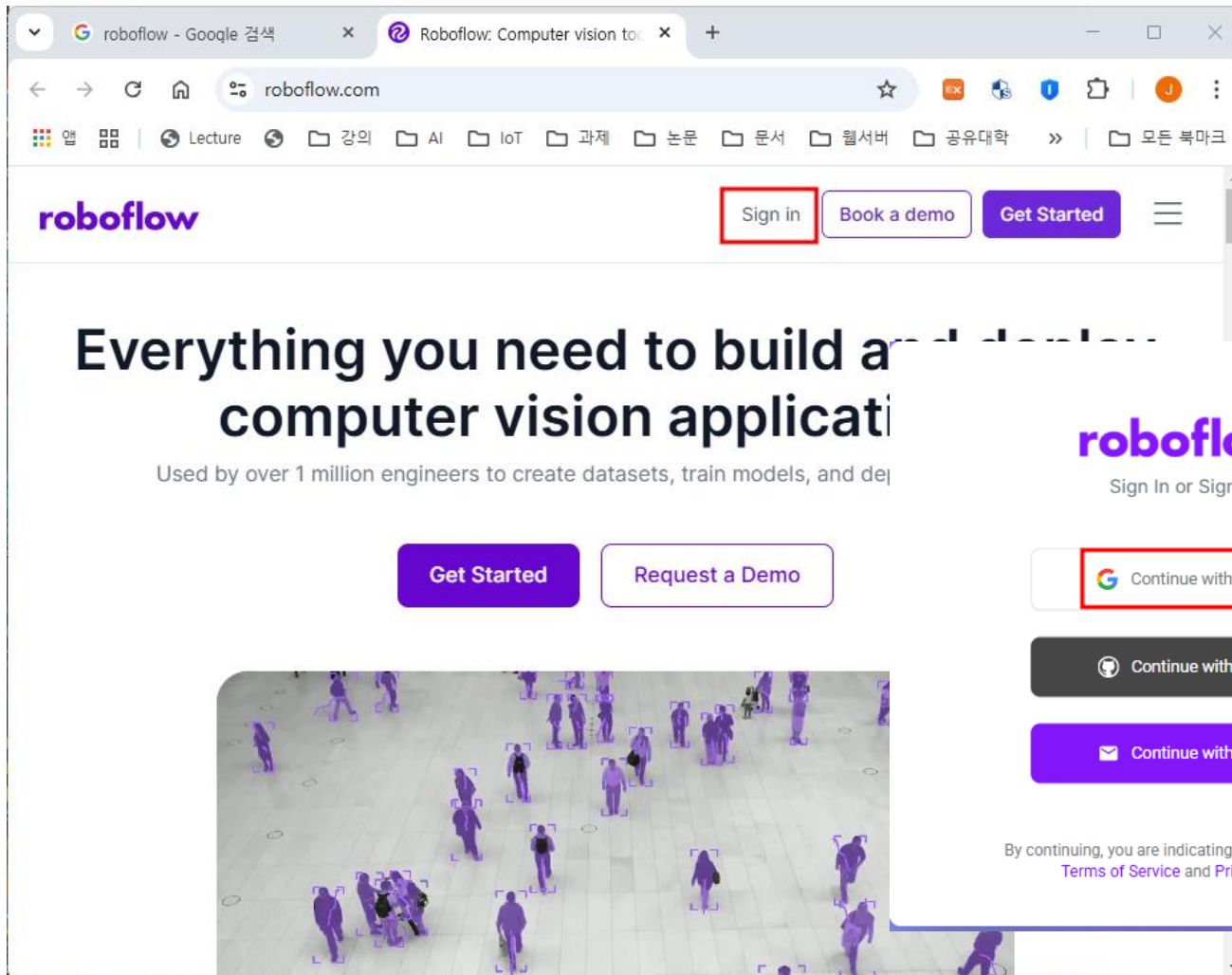
- YOLOv8n.pt
  - COCO 데이터셋으로 학습되어 있어, 총 80개의 클래스(객체)를 인식

번호	클래스명	클래스명	클래스명	클래스명	클래스명
1	person	dog	sports ball	sandwich	mouse
2	bicycle	horse	kite	orange	remote
3	car	sheep	baseball bat	broccoli	keyboard
4	motorcycle	cow	baseball glove	carrot	cell phone
5	airplane	elephant	skateboard	hot dog	microwave
6	bus	bear	surfboard	pizza	oven
7	train	zebra	tennis racket	donut	toaster
8	truck	giraffe	bottle	cake	sink
9	boat	backpack	wine glass	chair	refrigerator
10	traffic light	umbrella	cup	couch	book
11	fire hydrant	handbag	fork	potted plant	clock
12	stop sign	tie	knife	bed	vase
13	parking meter	suitcase	spoon	dining table	scissors
14	bench	frisbee	bowl	toilet	teddy bear
15	bird	skis	banana	TV	hair drier
16	cat	snowboard	apple	laptop	toothbrush

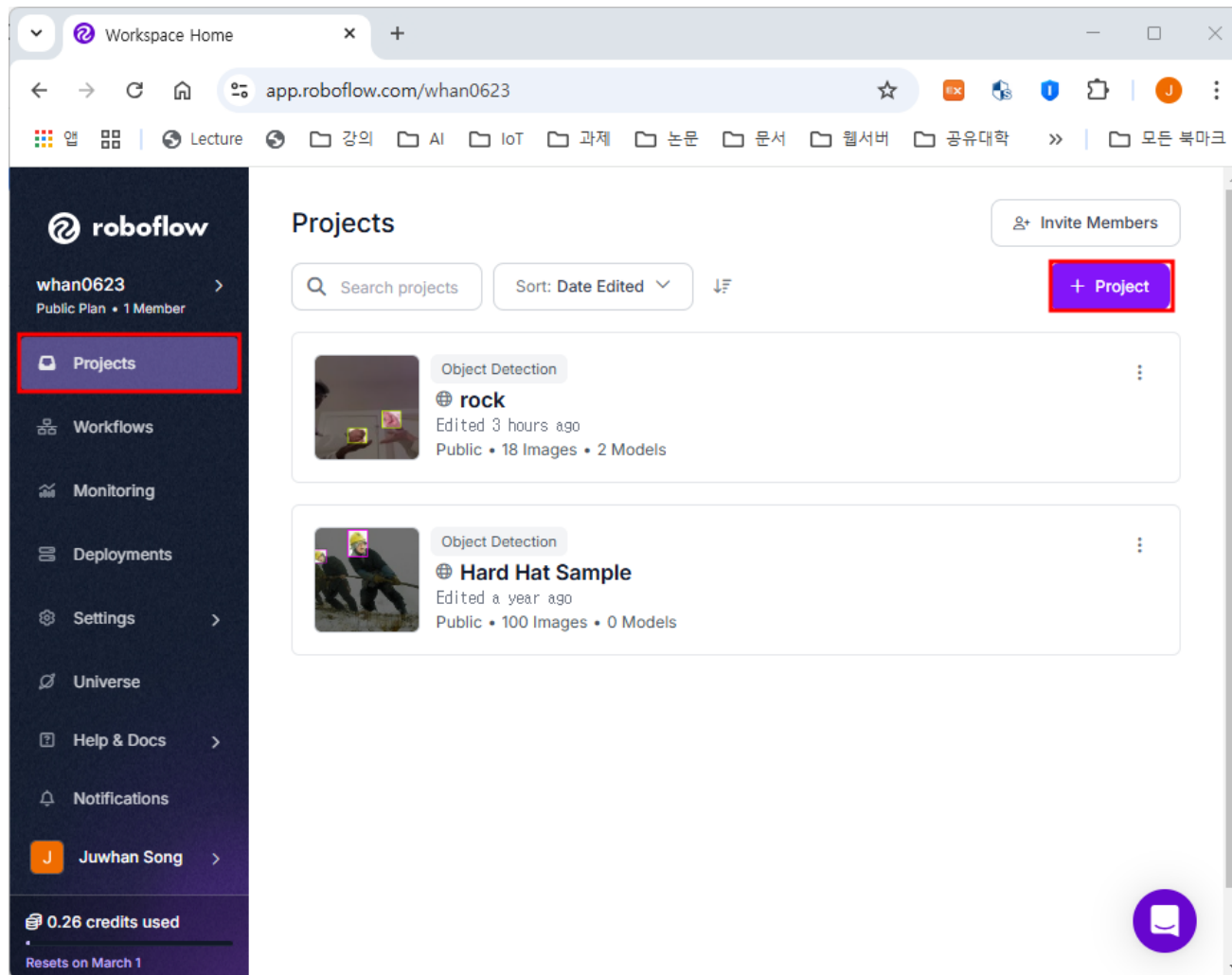
- YOLOv8에서 제공하는 데이터셋
  - COCO: 80개의 객체 카테고리가 포함된 대규모 데이터셋
  - COCO8: COCO의 8개 이미지 하위 집합으로, 빠른 테스트용 -> coco8.yaml
  - LVIS: 1203개의 객체 카테고리가 포함된 광범위한 데이터셋
  - Argoverse: 도시 환경의 3D 추적 및 모션 예측 데이터
  - VisDrone: 드론으로 촬영한 이미지의 객체 감지 데이터
  - SKU-110K: 소매 환경의 고밀도 물체 감지 데이터셋
  - VOC: 20개 객체 클래스, 11,000개 이상의 이미지
  - xView: 60개 객체 카테고리, 100만 개 이상의 주석
  - Roboflow 100: 7개 이미지 도메인의 100개 데이터셋
  - COCO128: COCO 데이터셋의 처음 128개 이미지, 소규모 서브셋, 테스트에 적합
  - Global Wheat 2020: 2020년 글로벌 밀 챌린지를 위해 밀 이삭의 이미지
  - Objects365: 365개의 객체 카테고리와 60만 개 이상의 주석이 달린 이미지
  - OpenImagesV7: Google에서 제공, 170만 개 학습 이미지와 4만 2천 개의 검증 이미지
  - African-wildlife: 버팔로, 코끼리, 코뿔소, 얼룩말 등 아프리카 야생 동물의 이미지
  - Signature: 다양한 문서에 대한 서명이 주석으로 달린 이미지로 구성된 데이터셋, 문서 검증 및 사기 탐지 연구에 활용
  - Medical-pills: 의약품의 품질 관리, 분류 및 산업 표준 준수를 돕기 위해 라벨이 지정된 의약품 이미지 데이터셋

# Rovoflow를 이용한 DataSet 생성

- Roboflow 웹사이트에 로그인



- 프로젝트 생성 : Projects 메뉴에서 +Project 버튼 클릭





- 프로젝트 설정 : 프로젝트 이름, 라벨, 라이선스

Create Project

app.roboflow.com/whan0623/create

roboflow

## Let's create your project.

whan0623 > New Public Project

Project Name: rock-paper-scissors

Annotation Group: handgame

License: CC BY 4.0

Project Type

**Object Detection**

☒ Bounding Boxes ☐ # Counts ☐ Tracking

Identify objects and their positions with bounding boxes.

**Classification**

☒ Image Labels ☐ Filtering ☐ Content Moderation

Assign labels to the entire image.

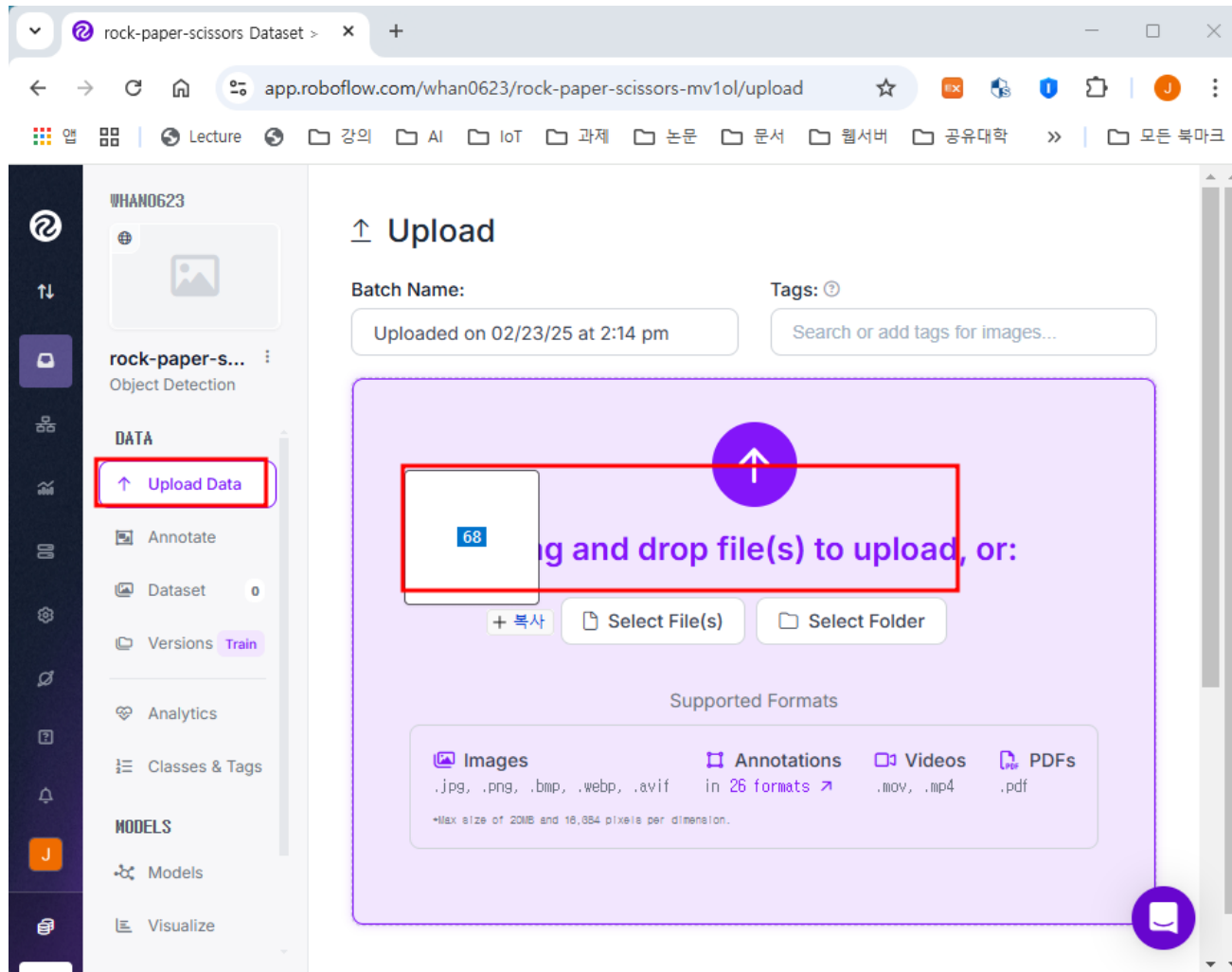
☒ Single-Label ☐ Multi-Label

ball

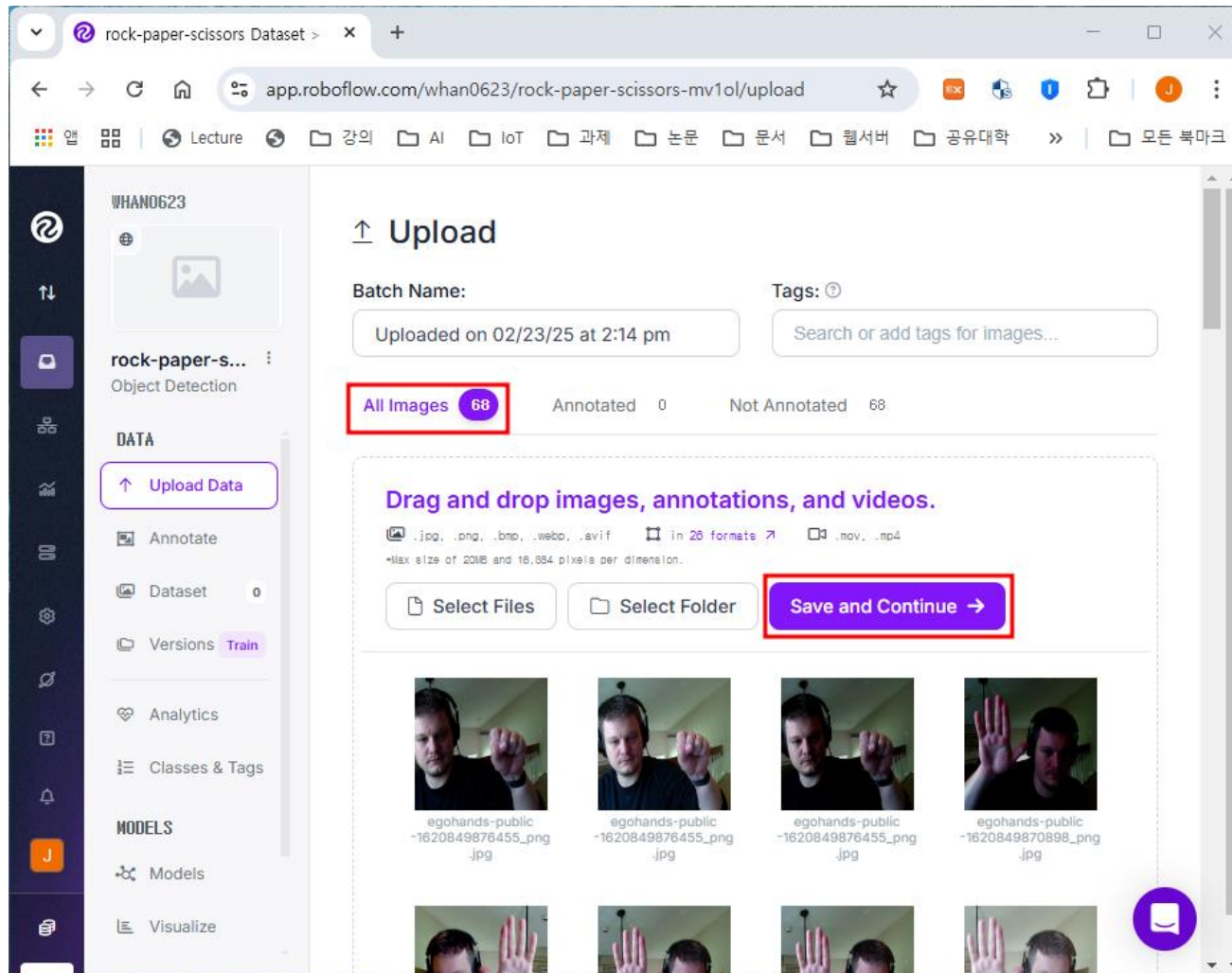
player

Cancel Create Public Project

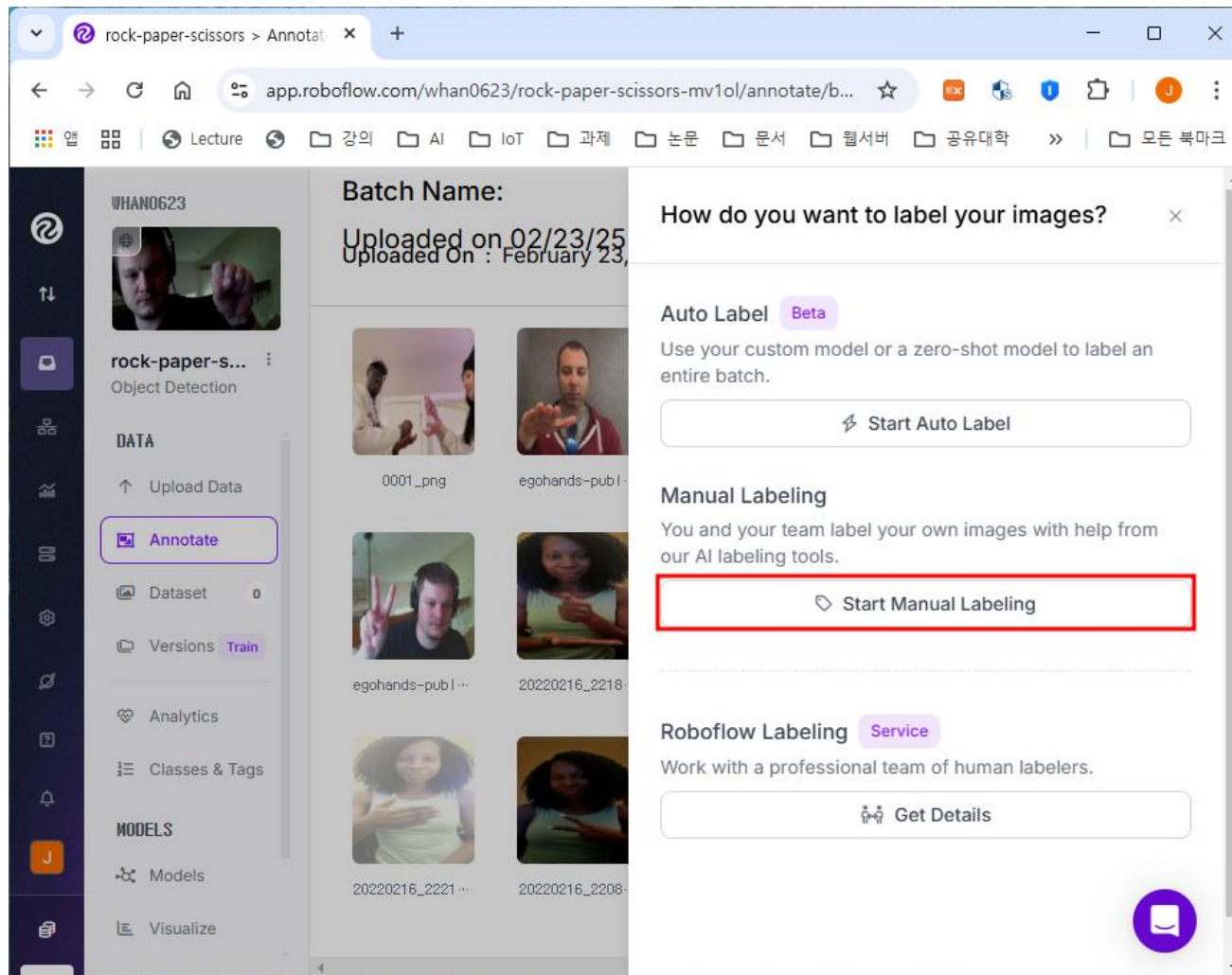
- 데이터 업로드 : Upload Data 선택 후
  - 라벨링할 이미지 파일들을 드래그 앤 드롭으로 업로드



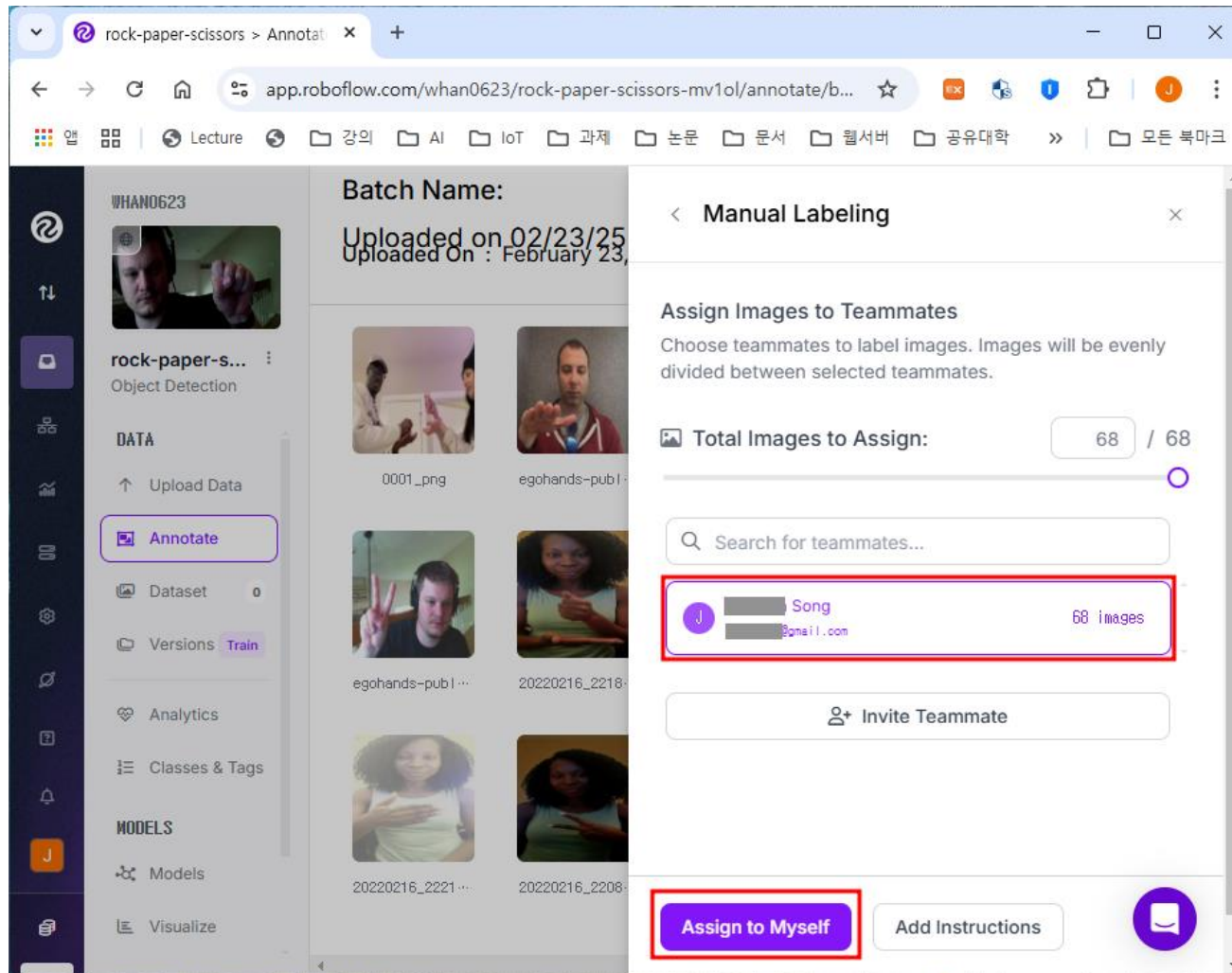
- 이미지 저장
  - Save and Continue 클릭



- 라벨링 작업 방법 선택
  - Start Manual Labelling 클릭하여 라벨링 작업 시작

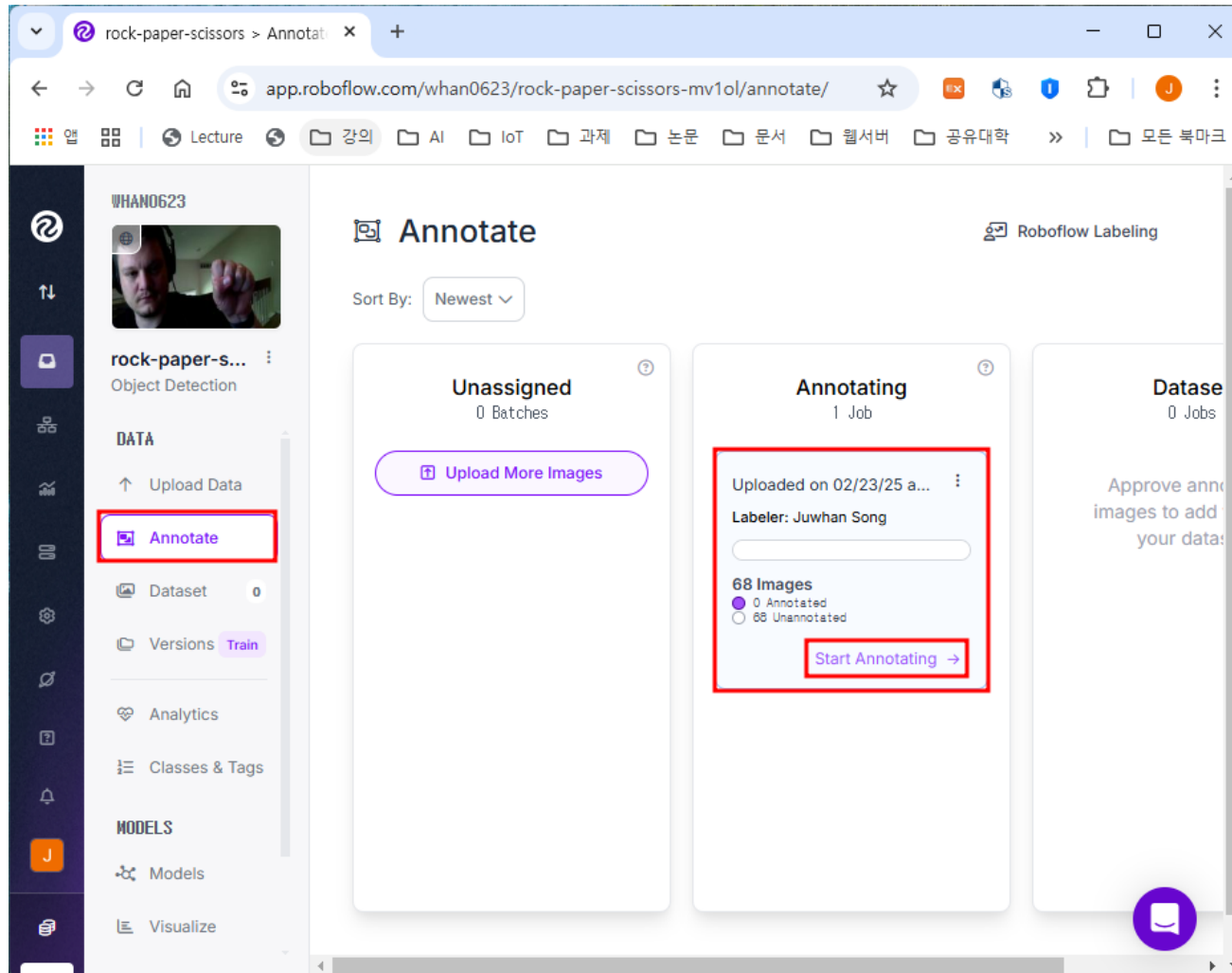


- 라벨링 작업자 선택

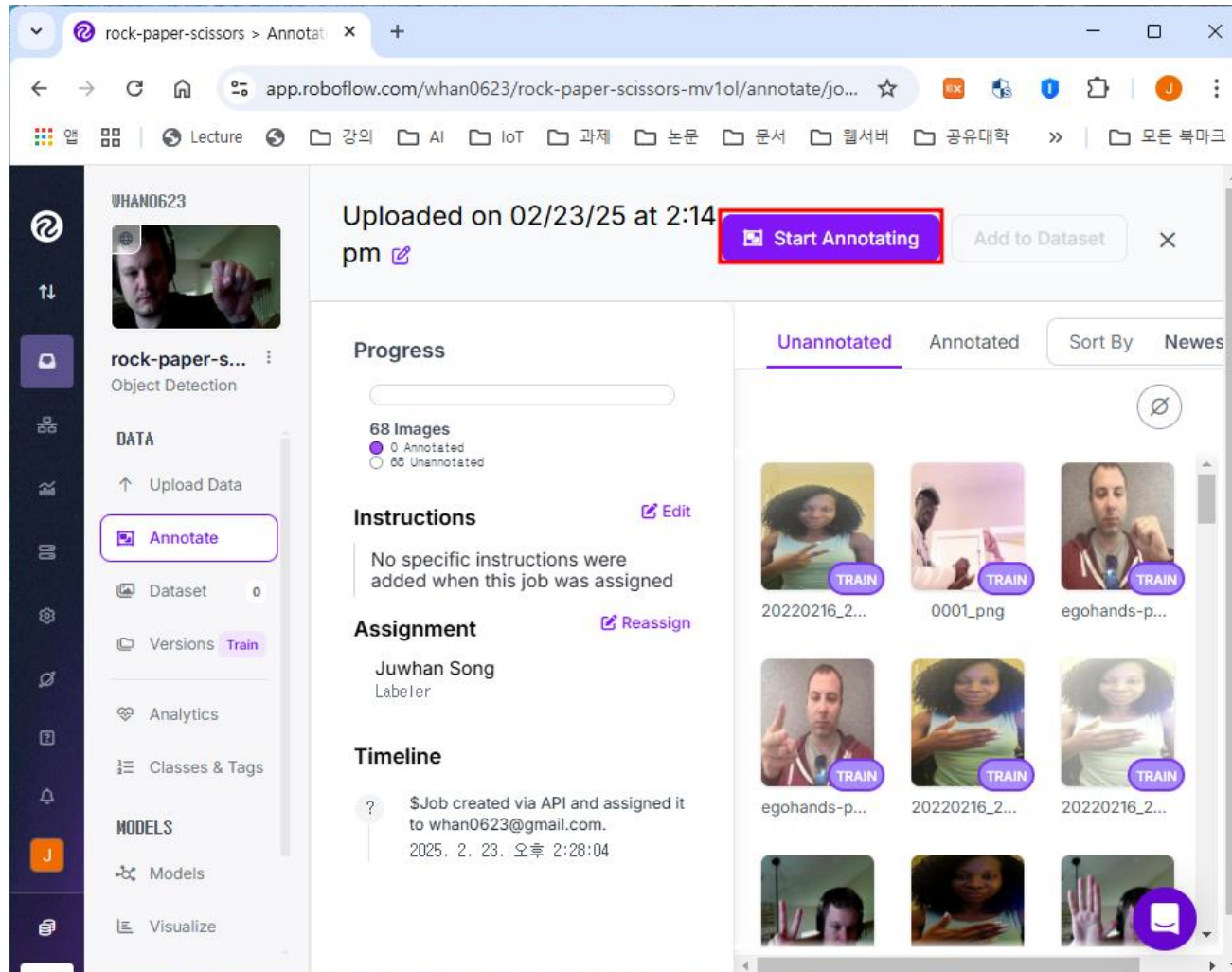




- 라벨링 작업 창으로 이동
  - Start Annotation을 클릭하여 라벨링 작업 창으로 이동

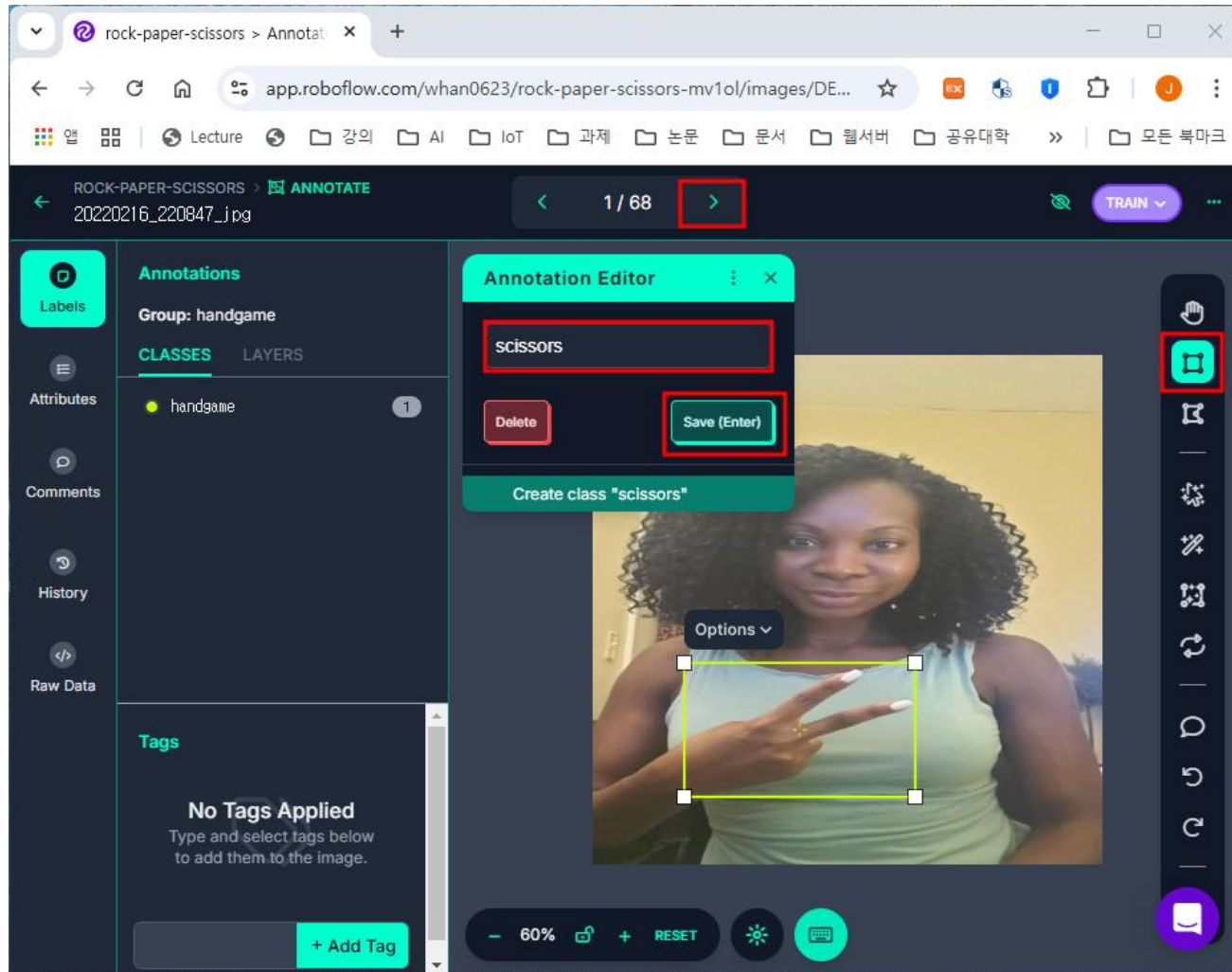


- 라벨링 작업 시작
  - Start Annotation을 클릭하여 라벨링 작업 시작



- 라벨링

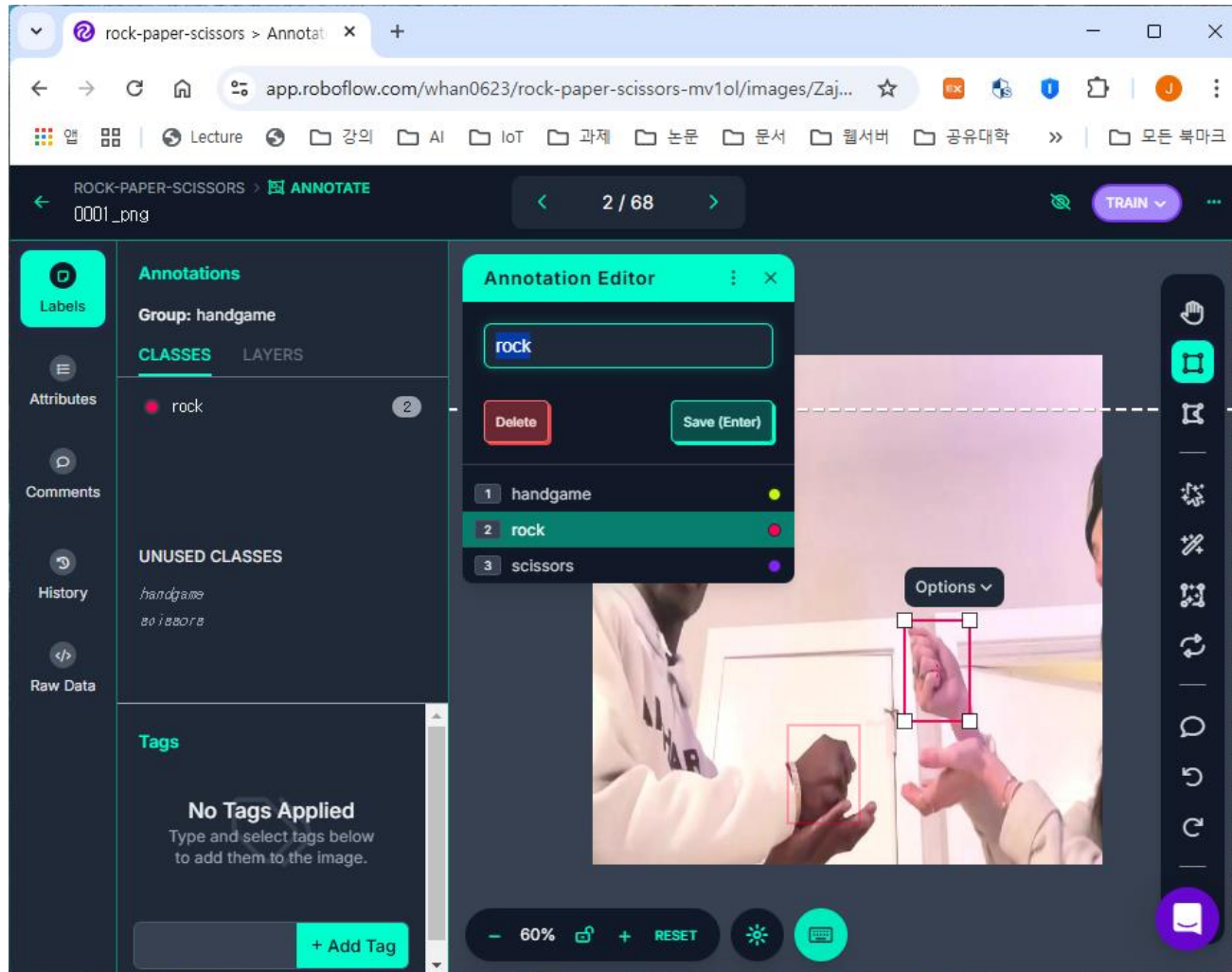
- 이미지에 bounding box를 그리고 class 선택 or 생성





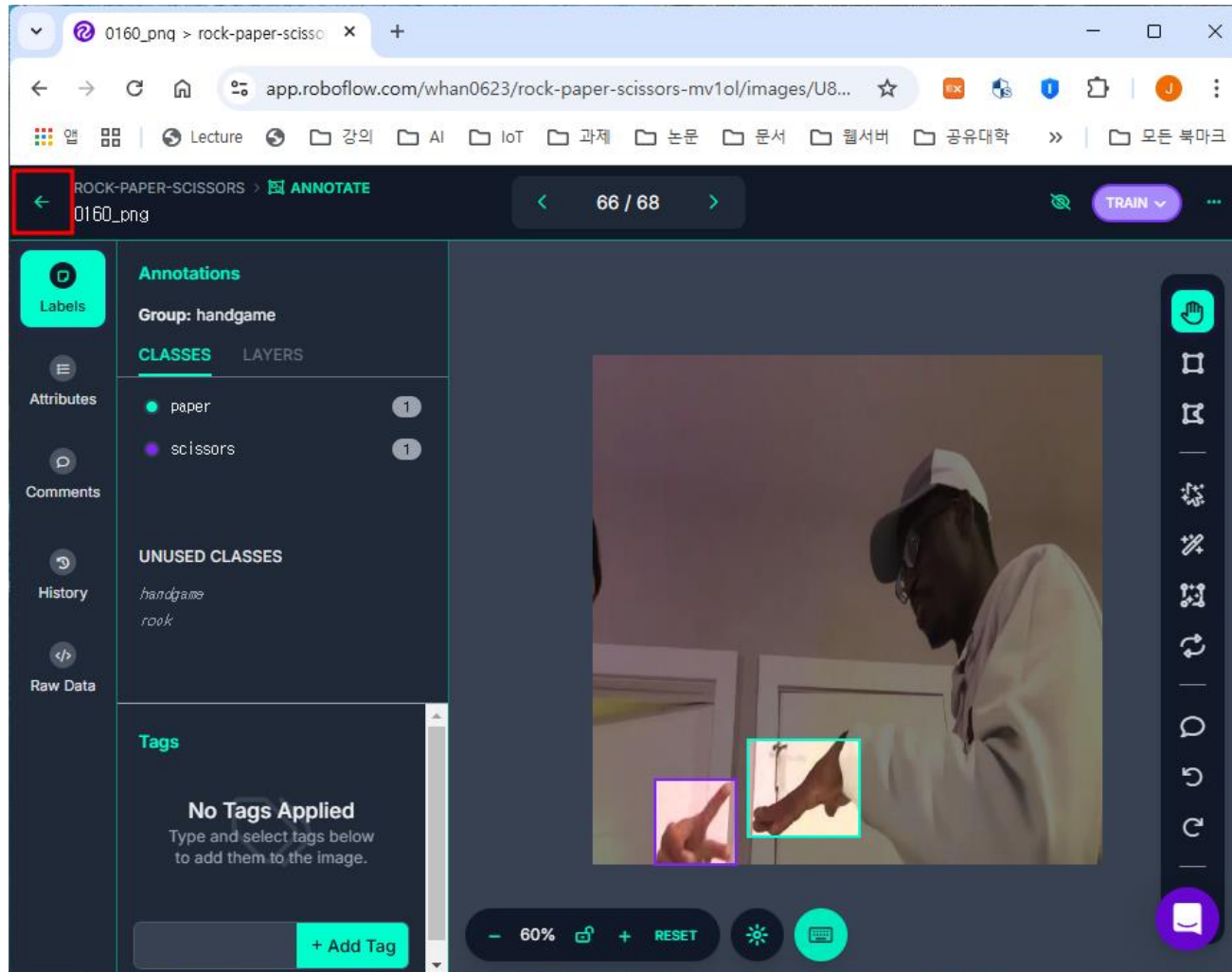
- 라벨링

- 이미지에서 여러 개의 bounding box를 그리고 class 선택 or 생성 가능

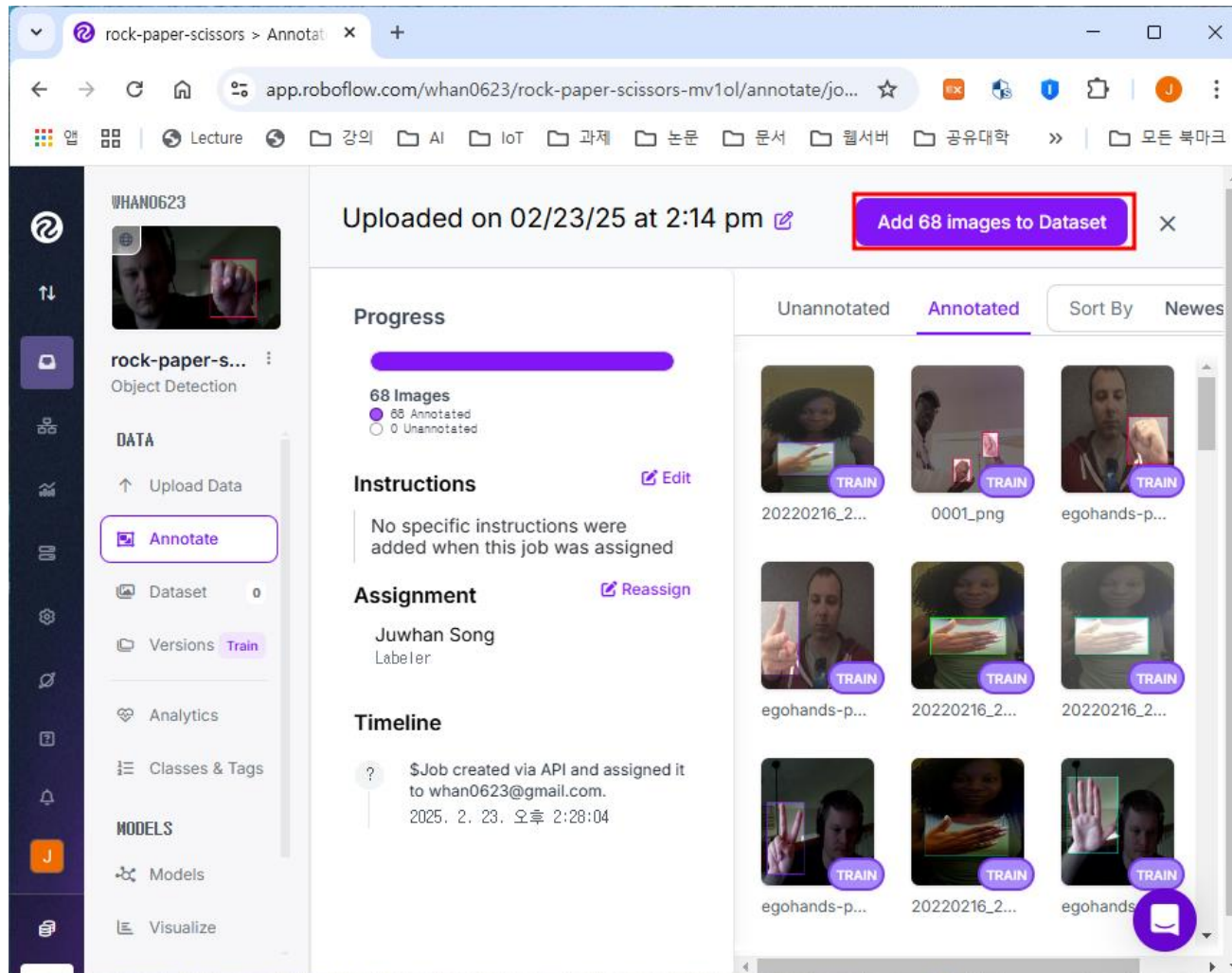


- 라벨링 종료

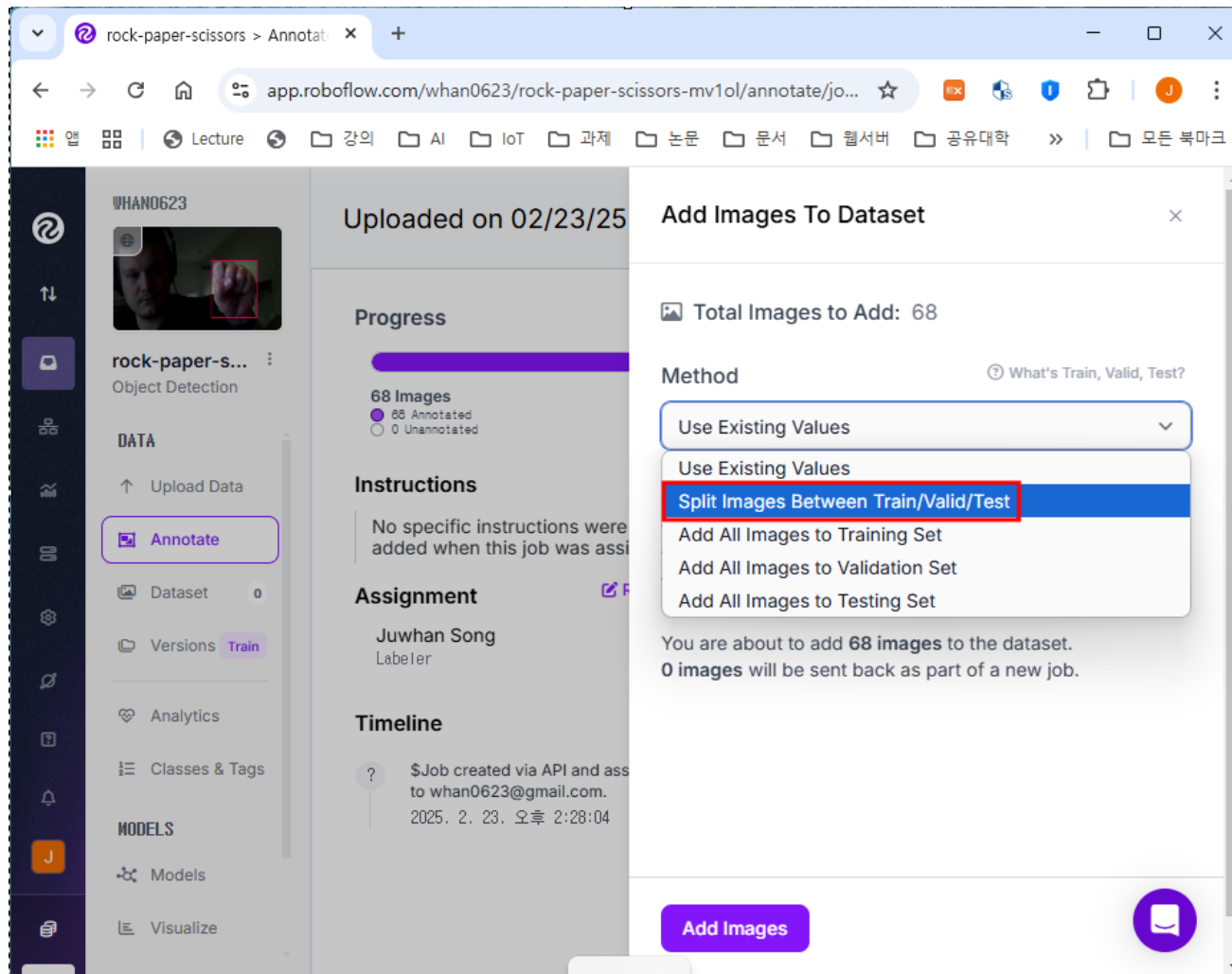
- 모든 이미지의 라벨링이 끝나면 <- 부분을 클릭해서 라벨링 종료



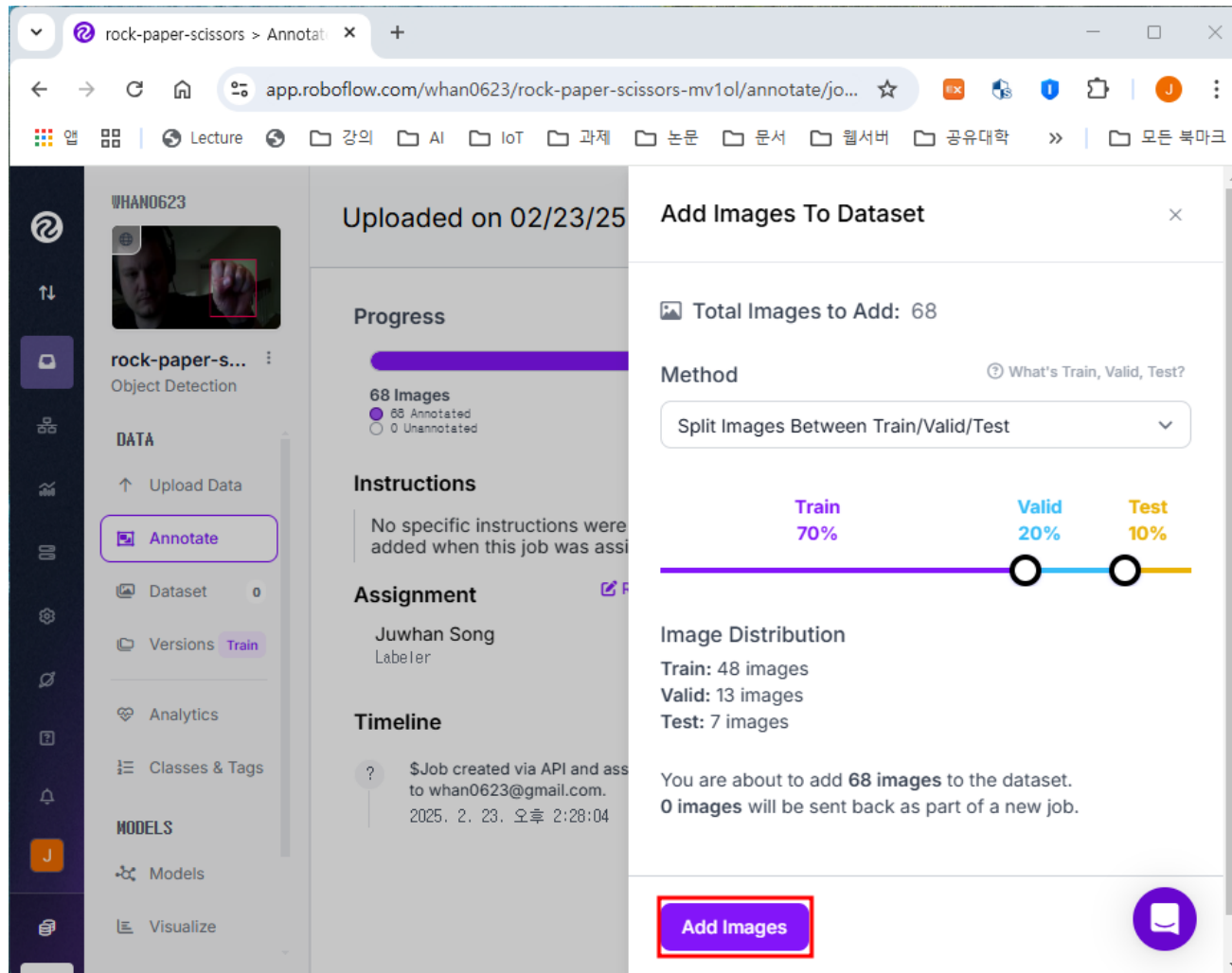
- DataSet 생성
  - Add X Images to Dataset 을 클릭하여 Dataset 생성



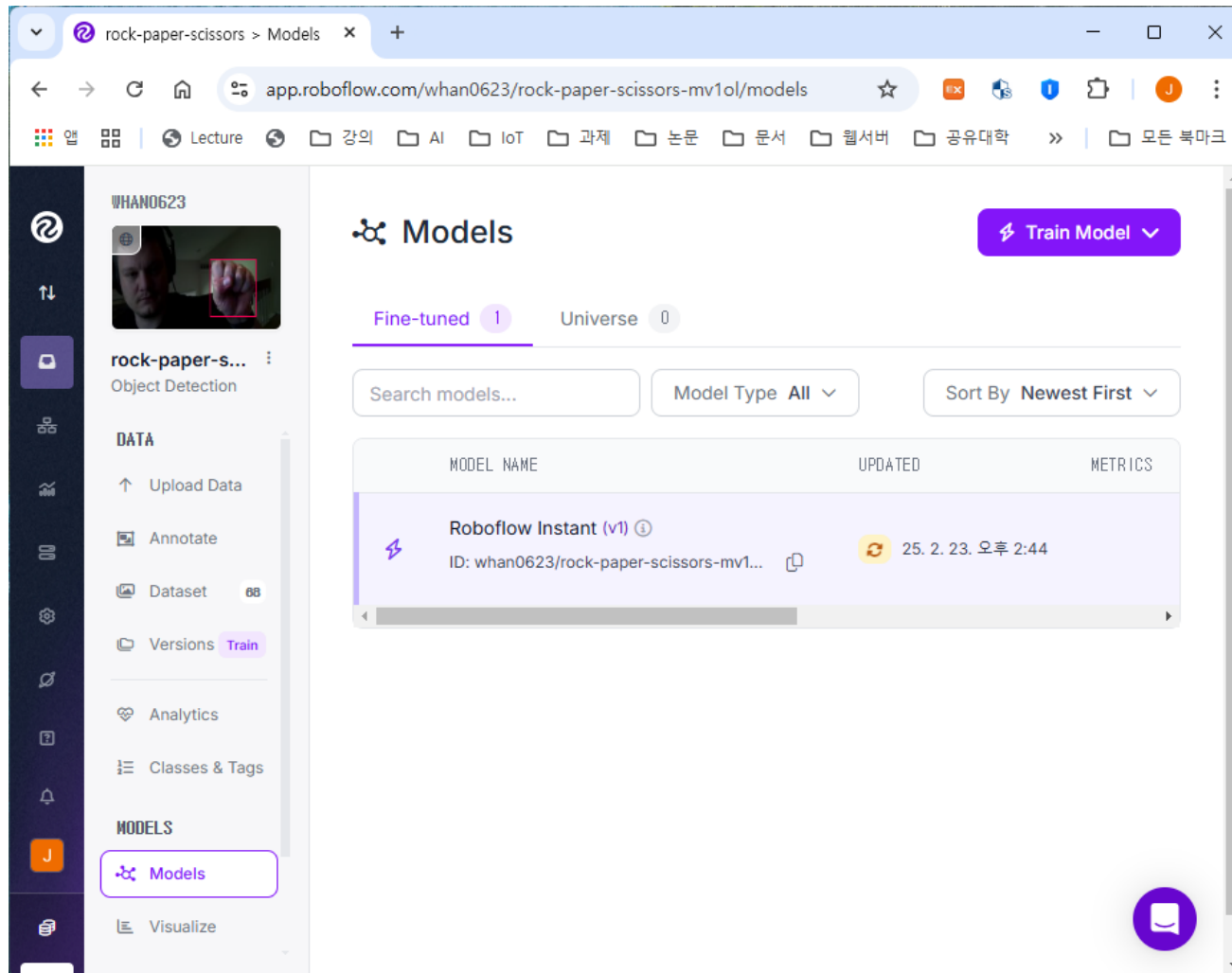
- Dataset을 Train, Valid, Test 용으로 분할 or 한곳에 몰아주기
  - Train, Valid, Test로 분할하여 저장



- Dataset을 Train, Valid, Test 용으로 분할 or 한곳에 몰아주기
  - Train, Valid, Test를 7:2:1로 분할하여 저장

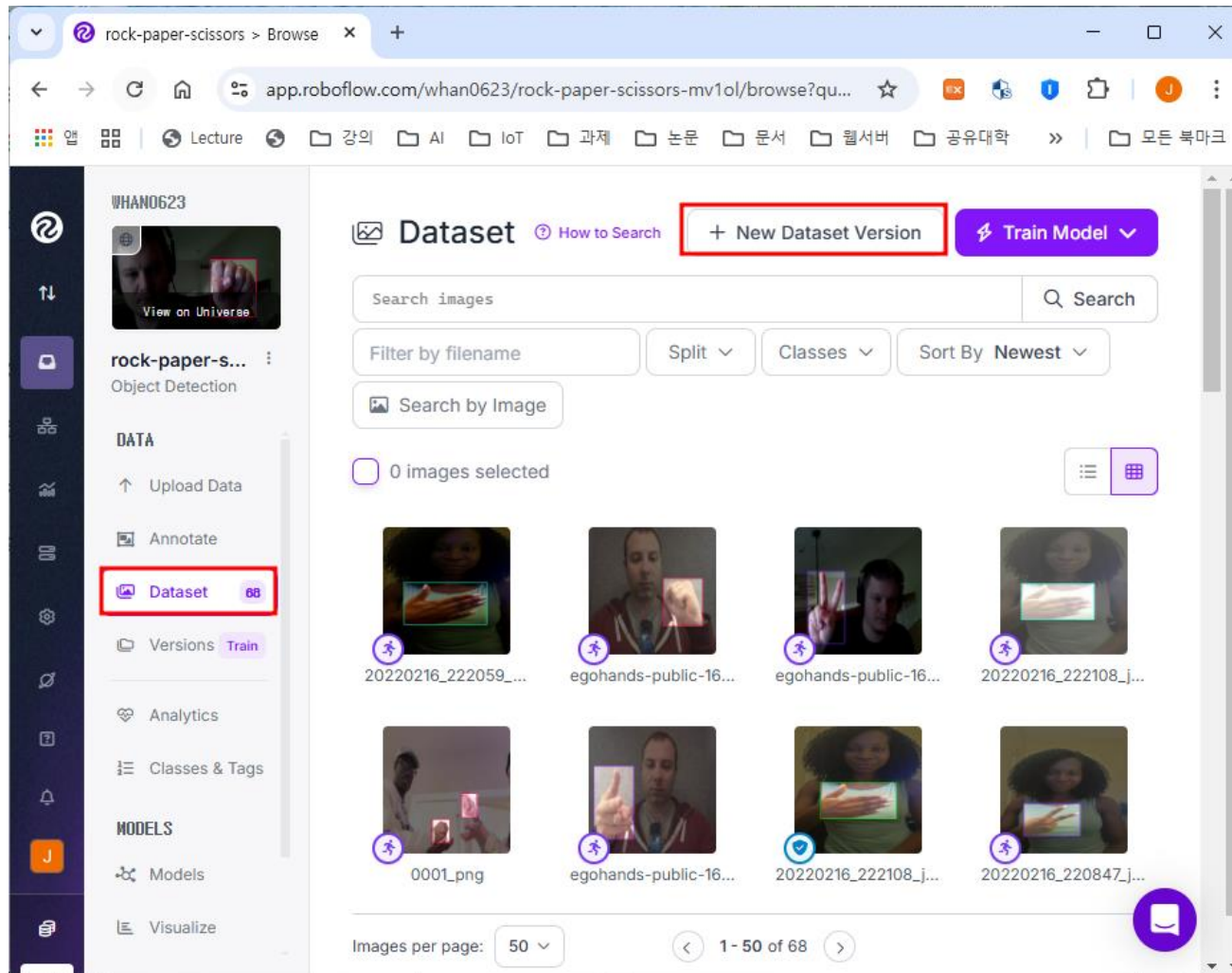


- Models 생성됨
  - 처리시간이 많이 소요됨

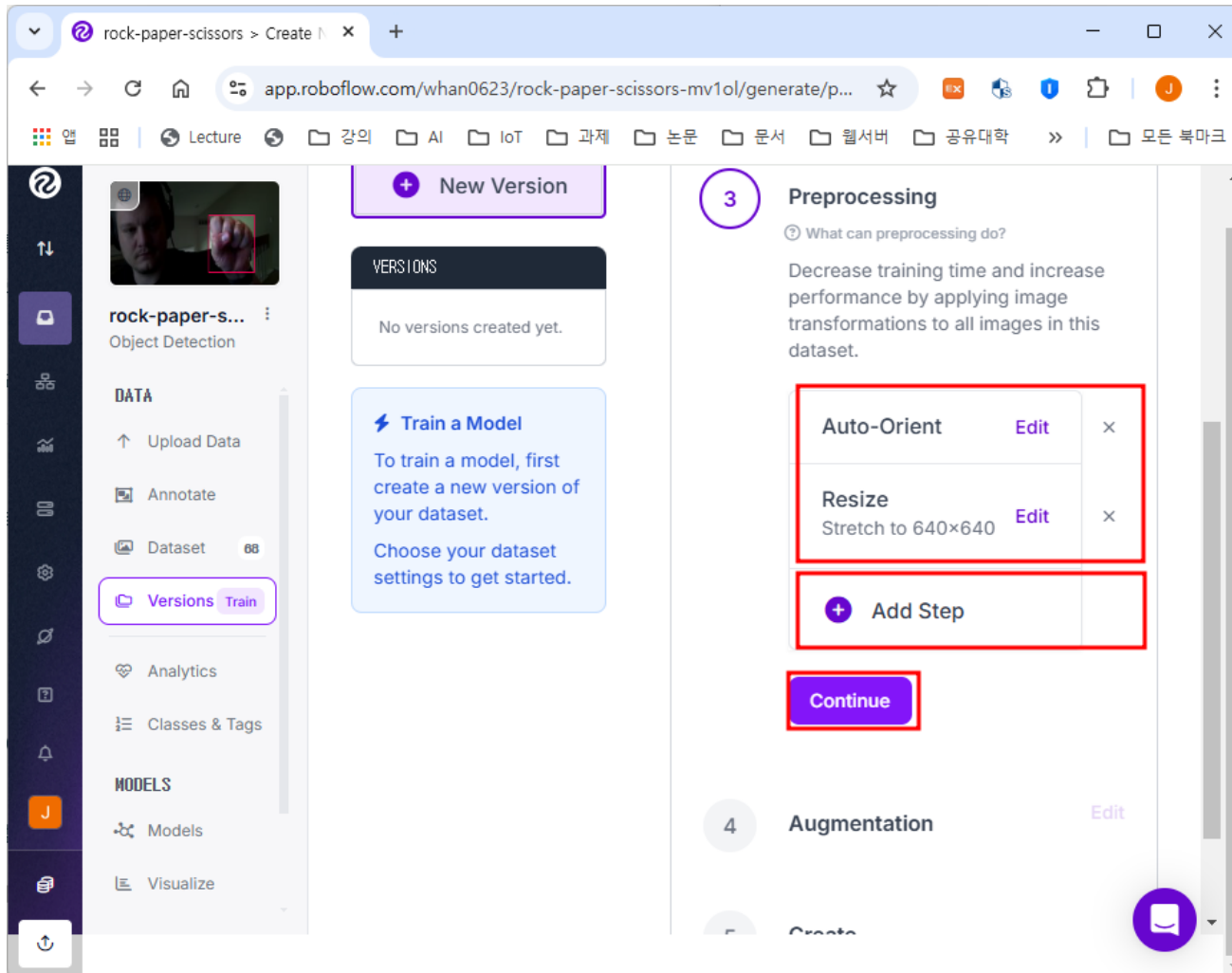




- DataSet 버전 생성
  - Dataset 메뉴 클릭 -> +New Dataset Version 클릭

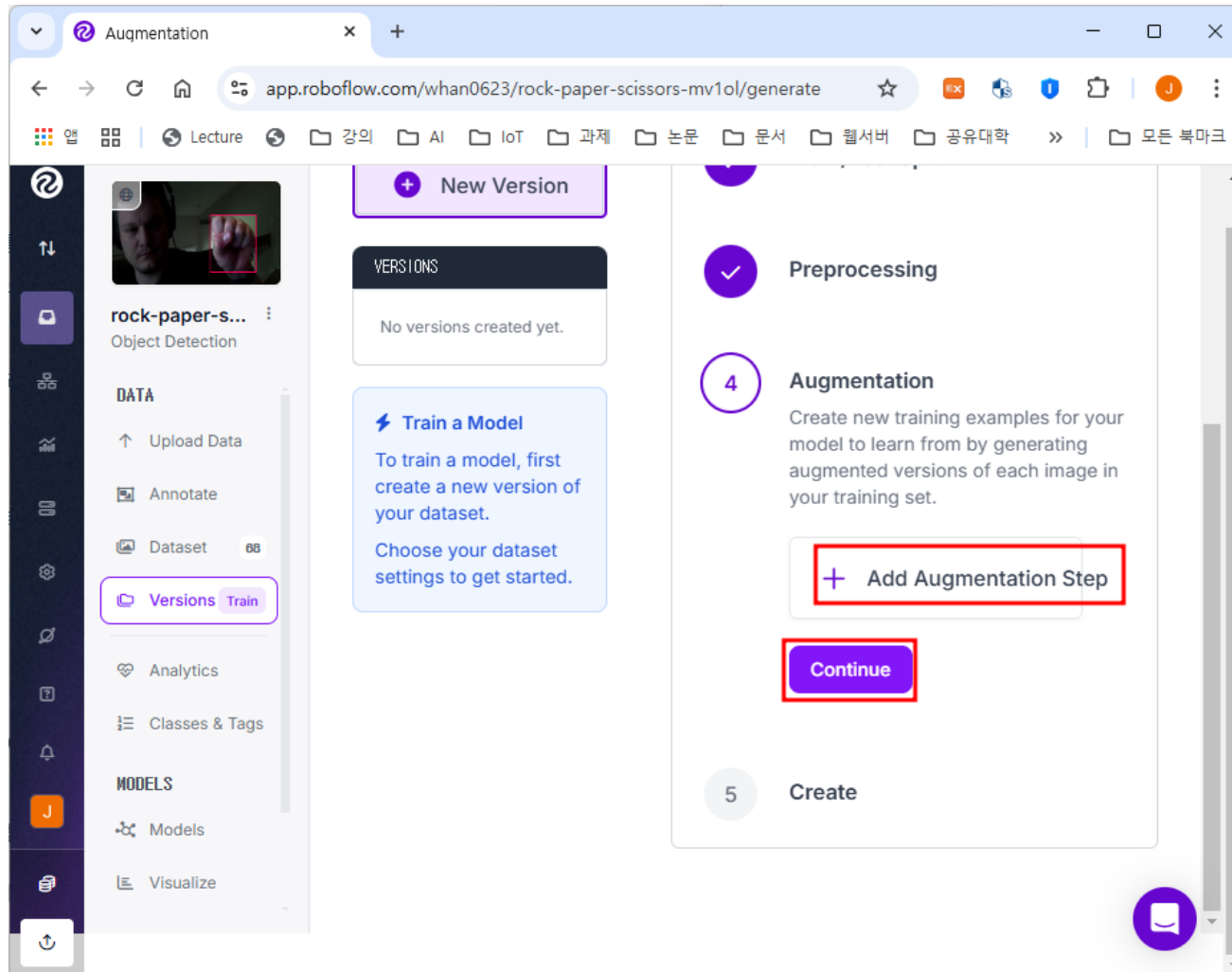


- DataSet 버전 생성
  - 필요한 전처리 선택 or 추가 후 Continue 클릭



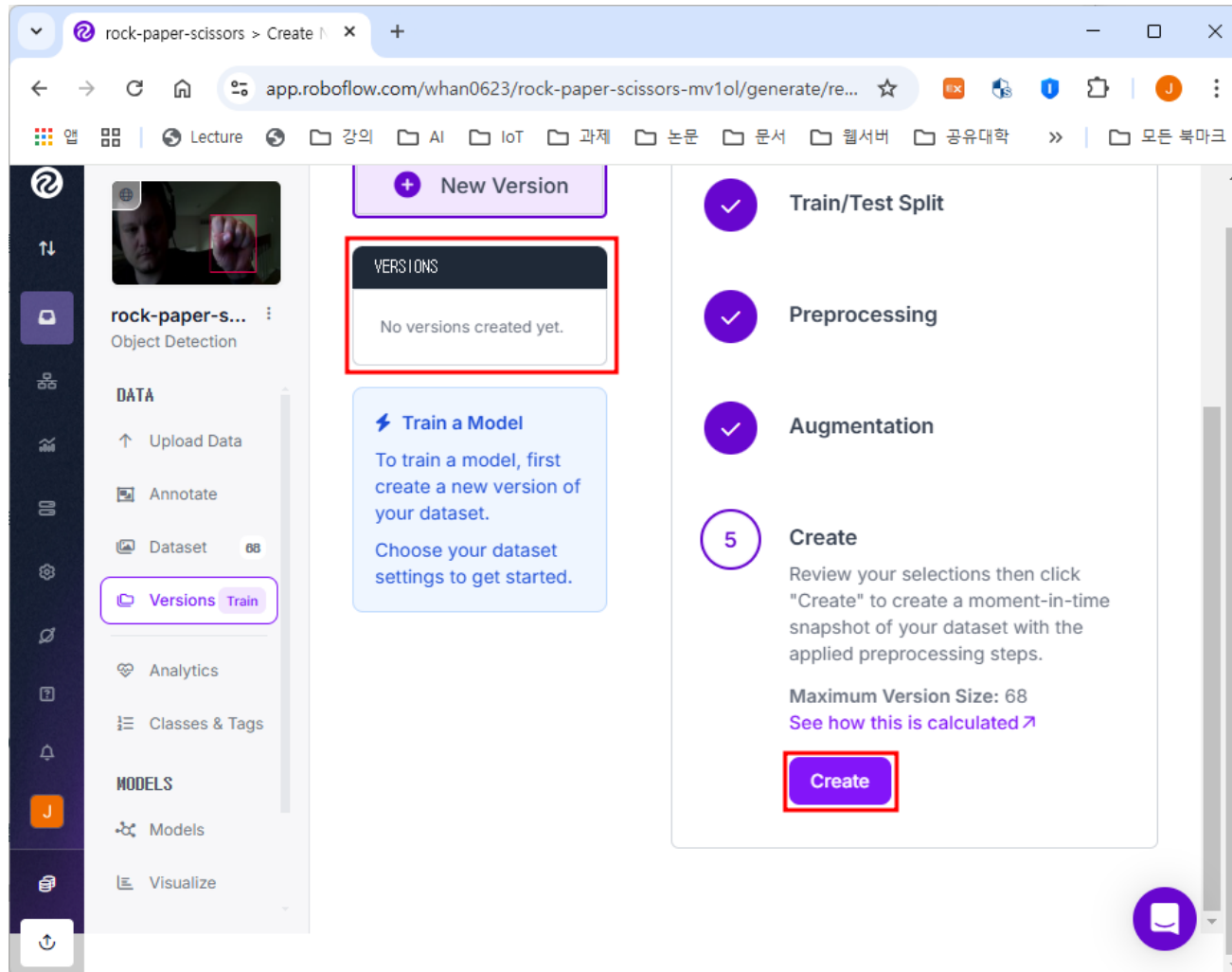


- DataSet 버전 생성
  - 필요한 증강 옵션 추가 후 Continue 클릭



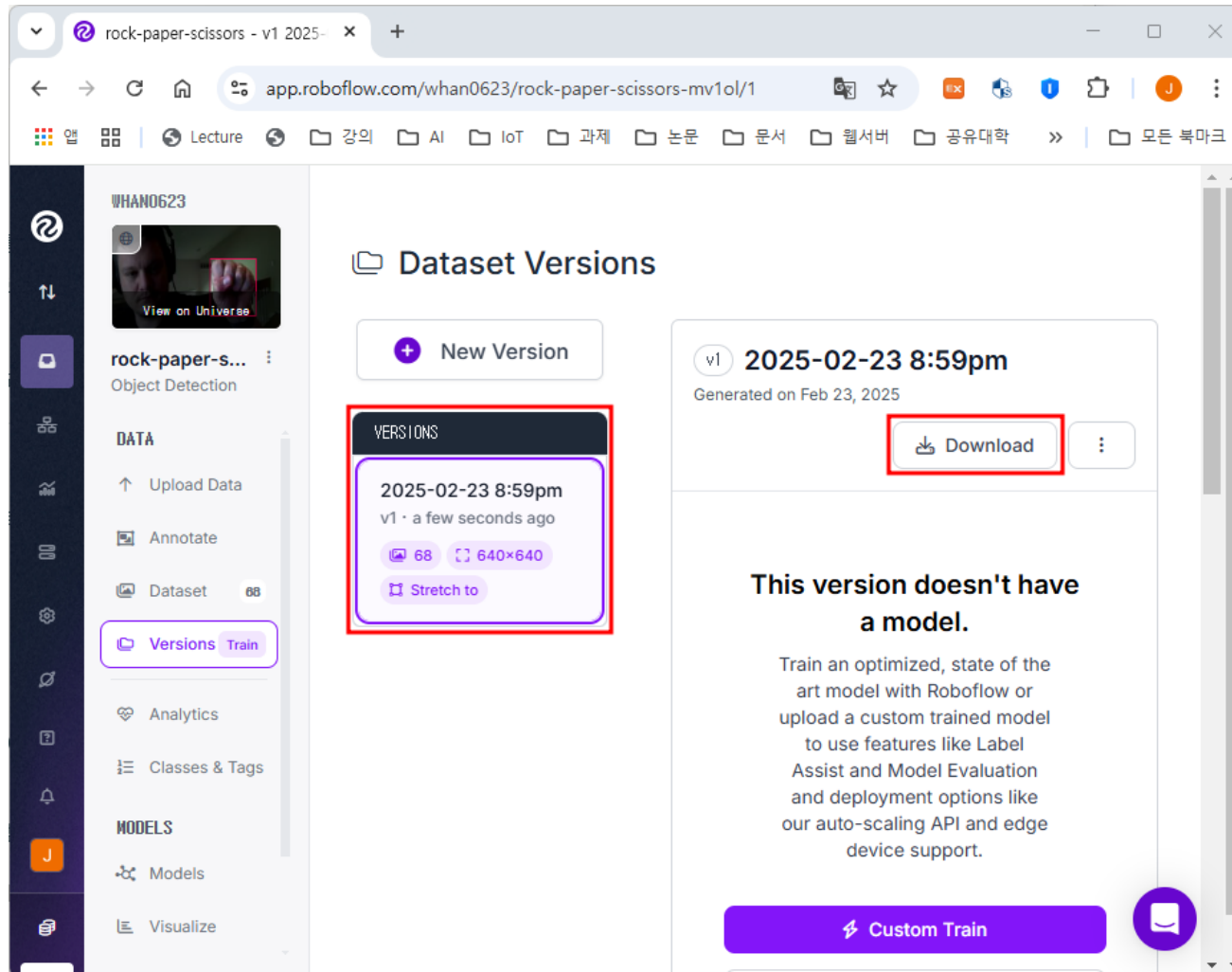
- DataSet 버전 생성

- Create를 클릭하면 새로운 버전이 생성됨, 현재는 No versions created yet.



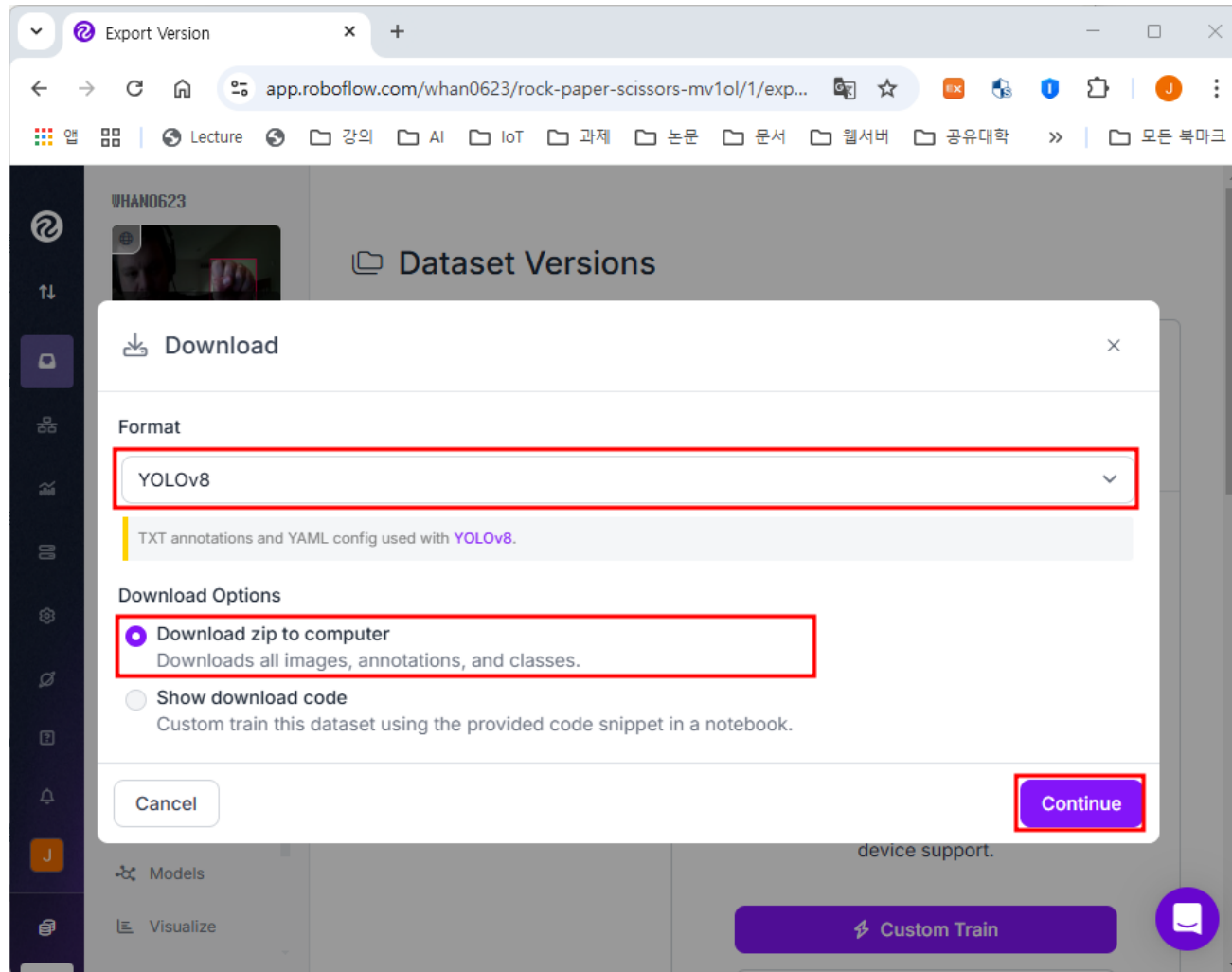
- DataSet 버전 생성

- 새로운 버전이 생성되었고, Download 버튼을 클릭하여 저장

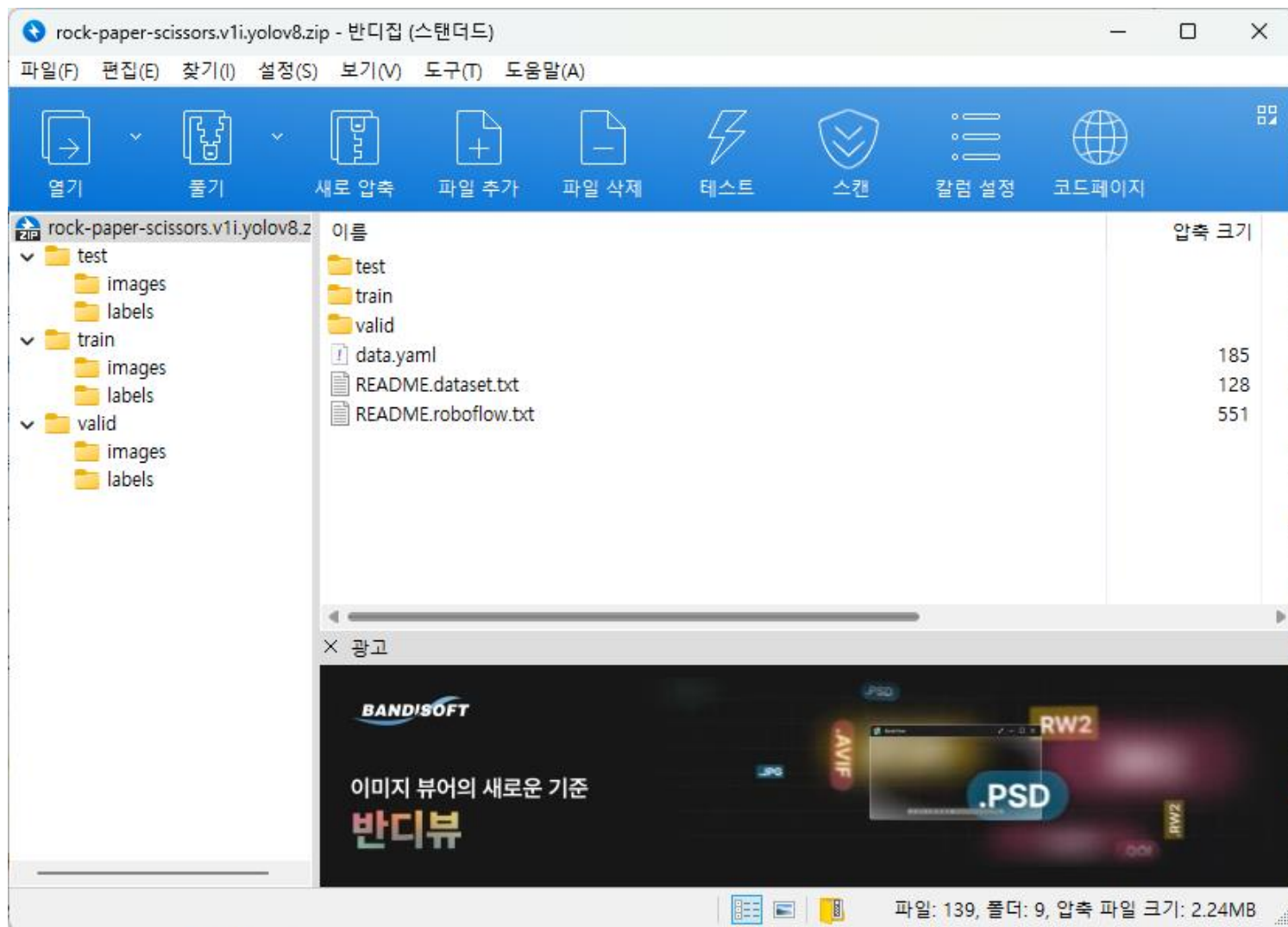


- DataSet 버전 생성

- Format: YOLOv8, Options: Download zip to computer 선택 후 Continue 클릭



- Zip 압축파일이 다운로드 됨



- DataSet 버전 생성

- 스크롤해서 아래를 보면 train 48개, valid 13개, test 7개로 7:2:1 비율임

The screenshot shows the RoboFlow web interface for a dataset named 'rock-paper-scissors'. The browser address bar shows 'app.roboflow.com/whan0623/rock-paper-scissors-mv1ol/1'. The interface includes a sidebar with navigation options like 'Upload Data', 'Annotate', 'Dataset', 'Versions', 'Train', 'Analytics', 'Classes & Tags', 'Models', and 'Visualize'. The main content area displays the dataset details, including a 'New Version' button, a 'VERSIONS' section showing the current version (v1) created on 2025-02-23 at 8:59pm, and a 'Dataset Split' section showing the distribution of images into Train (48 images, 71%), Valid (13 images, 19%), and Test (7 images, 10%) sets. The 'Dataset Split' section is highlighted with a red box. The 'VERSIONS' section is also highlighted with a red box. The 'Total Images' section shows 68 images with a 'View All Images' link. The 'Preprocessing' section is partially visible at the bottom.

Dataset Split	Count	Percentage
TRAIN SET	48 Images	71%
VALID SET	13 Images	19%
TEST SET	7 Images	10%



# thank you

본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된  
디지털신기술인재양성 혁신공유대학사업의 연구결과입니다.