Label Studio Export Formats – Quick Overview

General Purpose Formats

JSON – Full export of everything: raw data + all annotations in Label Studio's native structure. Use this when you want a complete backup or need to re-import into Label Studio later.

JSON-MIN – A trimmed-down version of JSON that keeps only the annotation values (`from_name`, `to_name`), dropping metadata. Use when you only care about the labels, not the full project structure.

CSV – Annotations exported as a spreadsheet with column names derived from your label fields. Easy to open in Excel or Pandas for quick inspection or custom processing.

TSV – Same as CSV but tab-separated instead of comma-separated. Useful when your label values contain commas.

Object Detection / Segmentation Formats

COCO – The industry-standard JSON format used by the COCO benchmark dataset. Supports bounding boxes, polygons, and keypoints. Use this for most deep learning frameworks (Detectron2, MMDetection, etc.) that expect COCO-style annotations.

COCO with Images – Same as COCO but also downloads and packages the actual image files alongside the annotation JSON. Use when you need a self-contained dataset folder ready to train with.

Pascal VOC XML – An older XML-based format widely used before COCO became dominant. Still required by some frameworks (older TensorFlow pipelines, some custom tools). One `.xml` file per image.

YOLO – Plain TXT format where each image gets one `.txt` file. Each line in the file is one object: `class_id center_x center_y width height` (all normalized 0-1). This is exactly what the script in our project uses for training YOLOv8.

YOLO with Images – Same YOLO TXT format but bundles the image files with it. Equivalent to what the script in our project produces after the train/val split – ready to point `data.yaml` at.

YOLOv8 OBB – A specialized YOLO variant for **Oriented Bounding Boxes**. Instead of a standard axis-aligned rectangle, each box is defined by its 4 corner points, allowing the box to be rotated. Use this when objects appear at angles (aerial imagery, rotated text, tilted products on a conveyor belt).

YOLOv8 OBB with Images – Same OBB format but with images included.

Specialized / Other Formats

CoNLL2003 – Text-based format for Named Entity Recognition (NER) tasks. Each token (word) gets a label tag on its own line. Standard format for sequence labeling tasks like tagging names, locations, and organizations in text.

Brush labels to NumPy – Exports pixel-level brush stroke annotations as `.npy` 2D arrays (one file per label). Use when you need raw mask arrays for custom image segmentation processing in Python.

Brush labels to PNG – Same brush stroke masks but saved as standard PNG image files instead of NumPy arrays. Easier to preview visually or use with tools that accept image masks.

Brush labels to COCO – Converts brush stroke annotations (stored internally as RLE-encoded masks) into COCO polygon format. Bridges the gap between manual brush painting and COCO-compatible training pipelines.

ASR Manifest – Exports audio transcription annotations as a JSON manifest file formatted for NVIDIA NeMo speech recognition models. Each line is a JSON object linking an audio file to its transcript.

Quick Decision Guide

Your goal	Use this format
Train YOLOv8 (our project)	**YOLO with Images**
Train with rotated boxes	**YOLOv8 OBB with Images**
Train with Detectron2 / MMDetection	**COCO with Images**
Use with older TensorFlow pipelines	**Pascal VOC XML**
Backup / re-import to Label Studio	**JSON**
Quick data inspection in Excel	**CSV**
Pixel-level segmentation masks	**Brush labels to PNG**
NER / text tagging tasks	**CoNLL2003**
NVIDIA NeMo speech models	**ASR Manifest**