Task

CVAT Annotation Tool

CVAT (Computer Vision Annotation Tool) is an open-source tool designed for annotating digital images and videos. It's widely used for creating annotations for machine learning and deep learning models in computer vision.

Key Features:

- **Multiple Annotation Types**: Supports bounding boxes, polygons, points, lines, and masks.
- **Collaborative Annotation**: Multiple users can work on the same project.
- **Automation**: Integrates with models to provide automatic annotations.
- **Customizable**: Users can create custom attributes for annotations.
- **Integration**: Can be integrated with other tools and frameworks via its API.

Requirements for CVAT:

- Python 3.7 or higher
- Docker
- Web browser (Chrome is recommended)

Requirements for Training a Model on a Custom Dataset

- **Data Collection**: Collect a diverse and representative dataset.
- **Data Annotation**: Use tools like CVAT to annotate the dataset accurately.
- **Data Preprocessing:** Normalize, resize, and augment the data.
- **Model Selection:** Choose an appropriate model architecture (e.g., YOLO, Faster R-CNN).
- **Training Environment:** High-performance GPU, sufficient RAM, and storage.
- **Frameworks**: Deep learning frameworks such as TensorFlow, PyTorch, or Keras.
- **Hyperparameter Tuning:** Learning rate, batch size, number of epochs, etc.

Amount of Dataset Needed for Training

The amount of data required depends on the **complexity of the task and the variability in the data.** For object detection tasks using models like YOLO:

- **Small Datasets**: Around 1,000-5,000 images might be sufficient for simpler tasks with fewer classes.
- **Moderate Datasets**: 10,000-50,000 images for moderate complexity.
- **Large Datasets**: Over 100,000 images for high complexity tasks with many classes and variations.

Recall

Recall (also known as sensitivity or true positive rate) is the ratio of correctly predicted positive observations to all actual positives. It measures the ability of a model to identify all relevant instances in a dataset.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Precision

Precision (also known as positive predictive value) is the ratio of correctly predicted positive observations to the total predicted positives. It measures the accuracy of the positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Ideal Value of Precision/Recall

The ideal values of precision and recall depend on the application:

- **High Precision, Low Recall:** Preferred in applications where false positives are costly (e.g., spam detection).
- **High Recall, Low Precision**: Preferred in applications where false negatives are costly (e.g., disease diagnosis).

In many cases, a balance between precision and recall is sought using the **F1 score**, which is the harmonic mean of precision and recall:

Best YOLO Model for Training on Custom Dataset

The YOLO (You Only Look Once) family of models includes several versions. The choice of model depends on the specific requirements such as accuracy, speed, and hardware constraints:

- **YOLOv3**: Good balance between speed and accuracy.
- **YOLOv4**: Improved accuracy and speed over YOLOv3.
- **YOLOv5**: High accuracy and efficiency, with easy-to-use PyTorch implementation.
- **YOLOv7**: State-of-the-art performance with better accuracy and speed.
- **YOLOv8**: Latest improvements with the best performance across various metrics.

For custom datasets, **YOLOv5** and **YOLOv7** are popular choices due to their balance of performance and ease of use.