

Computer Vision & Machine Learning Engineering Assignment

Assignment Title: Developing a YOLOv7 Prototype for Boxing Punch Detection and Statistics

Objective: Build a prototype model in YOLOv7 that can detect and classify any one type of punch. The different types of boxing punches are - 1. jab, 2. cross, 3. hook, 4. uppercut, and generate basic punch statistics. This project should be developed from scratch, without using any pre-existing public datasets, to simulate real-world model building for sports analysis.

Due Date:

1 week from the date of assignment. Submission date of **15 November 3pm UK time**.

Requirements:

1. Dataset Creation:

- **Collect Raw Video Data:** Use videos of professional boxing fights available on YouTube and convert them into frames. This can be done easily on platforms like Roboflow by pasting the YouTube link to extract individual frames for labeling.
- **Labeling & Annotation:** Create a dataset by labeling different types of punches (jab, cross, hook, uppercut) along with their bounding boxes. Use annotation tools like Roboflow or LabelImg or Label Studio, to label the selected punch type manually.
- **Data Augmentation:** Use data augmentation techniques (e.g., rotations, flips, color adjustments, etc.) to expand the dataset and improve model robustness.
- **Number of Annotations:** 100 to 200 should be enough, if not you can annotate more.

2. Model Implementation (YOLOv7):

- **YOLOv7 Setup:** Set up the YOLOv7 framework for training. Use official YOLOv7 GitHub repositories or documentation as guidance for implementation.
- **Training the Model:** Train your YOLOv7 model on the annotated dataset, ensuring it can detect and classify punches accurately. Training can be done on Google Colab, Roboflow, or other available cloud-based platforms.
- **Hyperparameter Tuning:** Experiment with different hyperparameters to optimize model performance for punch detection.

3. Metrics for Evaluation:

- **Accuracy & Precision of Detection:** Ensure your model can detect punches accurately. Track metrics like precision, recall, and F1 score for each punch type.
- **F1 Score Explanation:** The F1 score is a metric that combines precision and recall into a single number, providing a balanced measure. It is the harmonic mean of precision and recall, making it useful for cases where the model's ability to identify all punch types (recall) is just as important as accurately classifying each one (precision).

- **Statistics Generation:** Integrate a simple statistics function that counts the number and types of punches detected, simulating real-time punch stat generation.
4. **Documentation:**
- **Project Report:** Document the process, covering data collection, annotation, model architecture, training, results, and insights.
 - **Code & Instructions:** Provide a well-organized code repository with a README that includes setup instructions, dependencies, and a guide to reproduce the results.
 - Please do NOT use ChatGPT copy/paste for Documentation.
5. **Submission:**
- Submit the **code in a GitHub repository link** and the **fully trained model (.pt file)** in a **zip folder**.
 - Package the project into a shareable file format (e.g., zip) or provide a link to a private GitHub repository.

Evaluation Criteria:

1. **Completeness:** Was each stage of the project (data collection, annotation, model training, and stats generation) completed?
2. **Quality of Model Training:** How effectively does the YOLOv7 model detect and classify different punches?
3. **Code Organization & Documentation:** Is the code well-structured, and is the documentation clear and thorough?
4. **Creativity & Resourcefulness:** How well did the applicant address the challenge of building a dataset from scratch?
5. **Insights & Analysis:** Are the evaluation metrics presented in a meaningful way? Did the applicant gain valuable insights from the model performance?

Tools & References:

- **Annotation Tools:** LabelImg, Roboflow, Label Studio or similar tools.
- **YOLOv7 Resources:** Official YOLOv7 documentation, tutorials, and repository guidelines.
- **Cloud Platforms for Training:** Google Colab, Roboflow or similar tools.