How to evaluate training performance

Gain insight to real world application performance and areas for improvement

1. Weights Files (best.pt and last.pt):

- **best.pt**: Contains weights of the model at the epoch where it achieved best performance on validation set according to metric(usually validation accuracy/ lowest validation loss). Use this file when deploying models/perform further evaluations as it's the most effective version.
- **last.pt:** Contains weights at last training epoch. Use this if you plan to continue the training model from where it left off. It's the most recent state of the model(neglecting performance)

2. Configuration File (args.yaml):

- **args.yaml:** A record of training environment & parameters. Includes thing slike architecture used, batch size, learning rate, epoch #, and other hyperparameters/settings. Use this to replicate the training process, understand conditions, and identify parameter related issues for optimization.

3. Confusion Matrices:

- **confusion_matrix.png**: Displays model's predictions, with rows(actual classes) and columns(predicted classes). High values on the diagonal indicates correct classification. High off-diagonal values suggest confusion between classes.
- **confusion_matrix_normalized.png**: Shows same info, but in terms of proportion/percentages. High values = close to 1 (100%)

4. Performance Curves:

- **F1_curve.png:** Represents harmonic mean of precision and recall. Provides a single metric that balances both. A stable and high F1 score curve indicates that the model maintains a good balance between precision and recall across its predictions.
- **P_curve.png (Precision):** Measures accuracy of positive predictions. Shows how precision changes over different epochs and thresholds. Higher value means greater proportion of true positives among positive predictions.
- **R_curve.png (Recall):** Measures how well the model identifies actual positives across different epochs / thresholds. Higher value means good catching positive cases. High value should be maintained.
- **PR_curve.png:**
  The Precision-Recall curve is critical for evaluating models, especially on imbalanced datasets. A model that maintains high precision at high recall levels is considered robust. The area under this curve (AUC) is a key metric: the closer it is to 1, the better.

5. Label Analysis:

- **labels.jpg:** Label distribution/ analysis of specific label occurrences. Is the dataset balanced or any anomalies in label distribution.

- **labels_correlogram.jpg:** If your task is multi-label, this correlogram can show how often different labels occur together, highlighting potential dependencies or redundancies between labels that could affect model training and interpretation.

6. **Results Files:**
   - **results.csv**: Provides epoch-wise numerical data, such as loss, accuracy, or other metrics, allowing for a detailed analysis of the training progression and stability. You can use this data to identify patterns or issues such as early overfitting or underfitting.
   - **results.png:** Visual representation of training progress, typically showing loss and accuracy trends over epochs for both training and validation sets. You're looking for converging trends indicating learning stability and avoiding overfitting (where validation metrics diverge negatively from training metrics).

7. **Batch Images for Training and Validation:**
   - **train_batchX.jpg:** Visual samples from training batches can help you verify the actual data fed into the model and ensure that data augmentation or preprocessing steps are applied correctly.
   - **val_batch0_labels.jpg and val_batch0_pred.jpg**: Comparing these allows you to visually assess the model's performance on the validation set. It provides a direct insight into what the model is getting right or wrong at the instance level, which can be particularly illuminating for understanding misclassifications or biases in the model's predictions.