- **Project Name:** Optimus

- **Problem Statement:**

  Retraining classification and detection models takes a lot of time and effort. First, we need to prepare new datasets and merge them with older ones, which can be complex and time-consuming. Then, we have to evaluate the models by generating performance reports like graphs and confusion matrices. After that, it's important to store and organize these trained models properly so they can be used again when needed. On top of that, deploying these models for use adds another layer of difficulty. Managing all these steps together is a big challenge, and it slows down our overall progress.

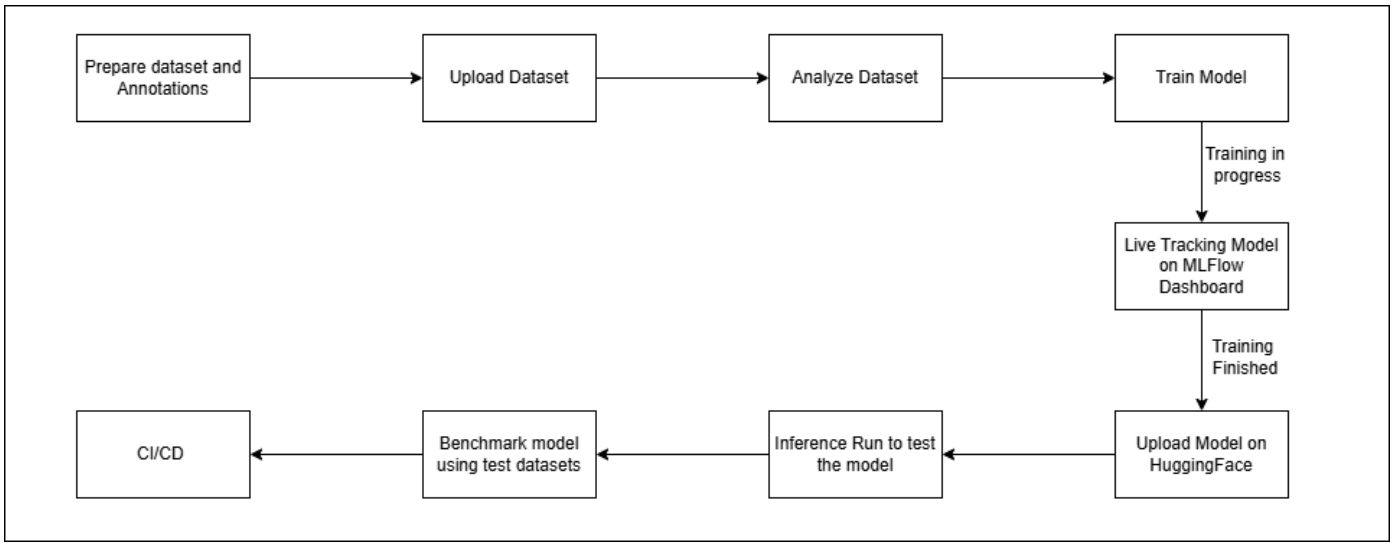
- **Project Objective:**

  The goal of this project is to create an easy-to-use Streamlit application that simplifies the process of managing machine learning workflows using UI. This app will allow to:

  1. Manage Datasets: Add, update, and organize datasets efficiently in one place.
  2. Visualize Datasets: Explore and analyze datasets through interactive visualizations to better understand their content.
  3. Train Models: Select multiple datasets for training machine learning models and run the training process directly from the app.
  4. Track Performance: Automatically save and display important model performance metrics, such as graphs and confusion matrices, using an integrated MLflow dashboard.
  5. Deploy Models: Seamlessly deploy trained models to Hugging Face, making them accessible for further use or sharing.

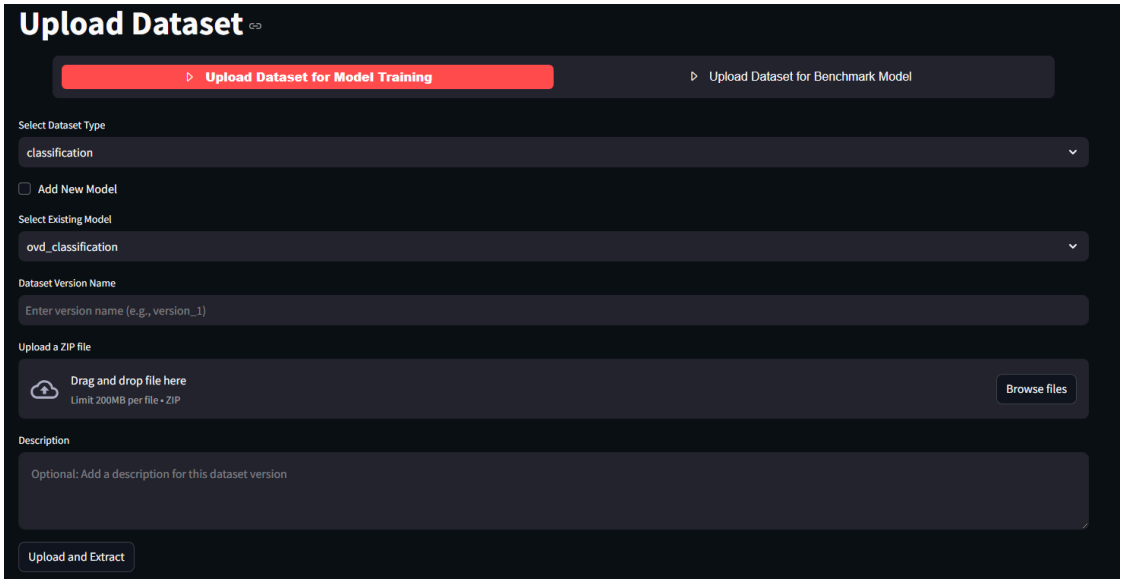
  This app will streamline the end-to-end workflow, making it easier to manage datasets, train models, and monitor their performance.

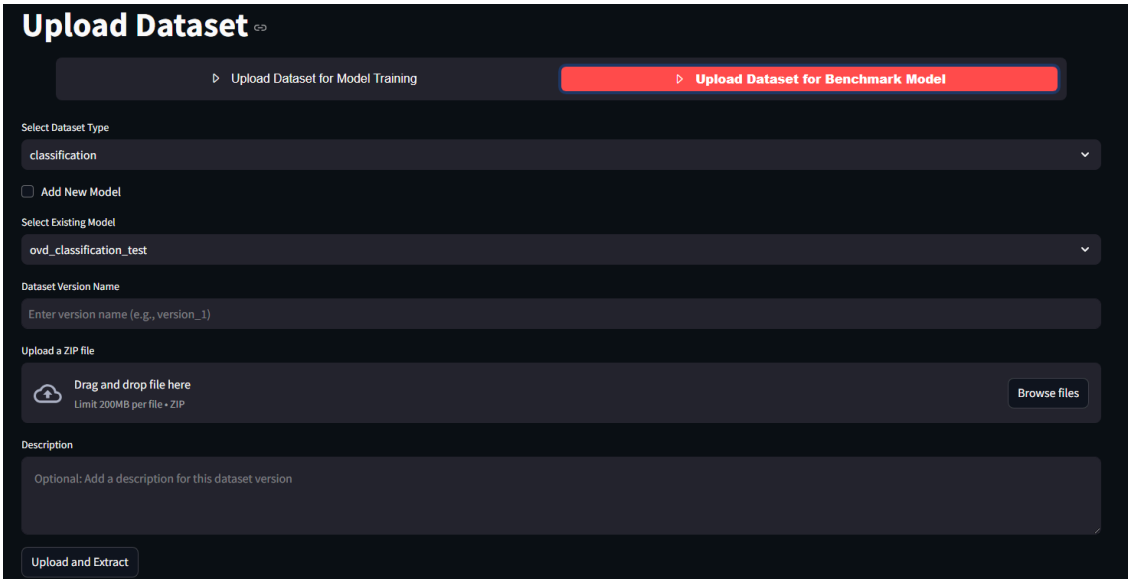
- **Flow Diagram:**



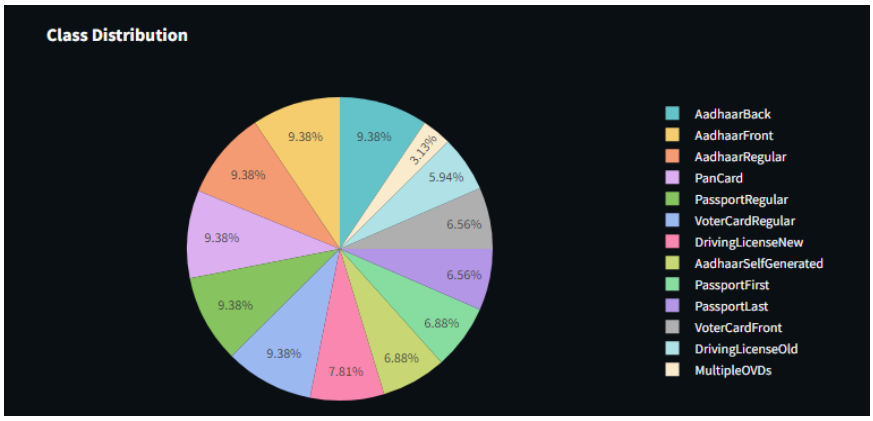
- **Tech Stack:** Python, Streamlit, MLFlow, Docker, HuggingFace, Ultralytics, Pytorch, Label Studio

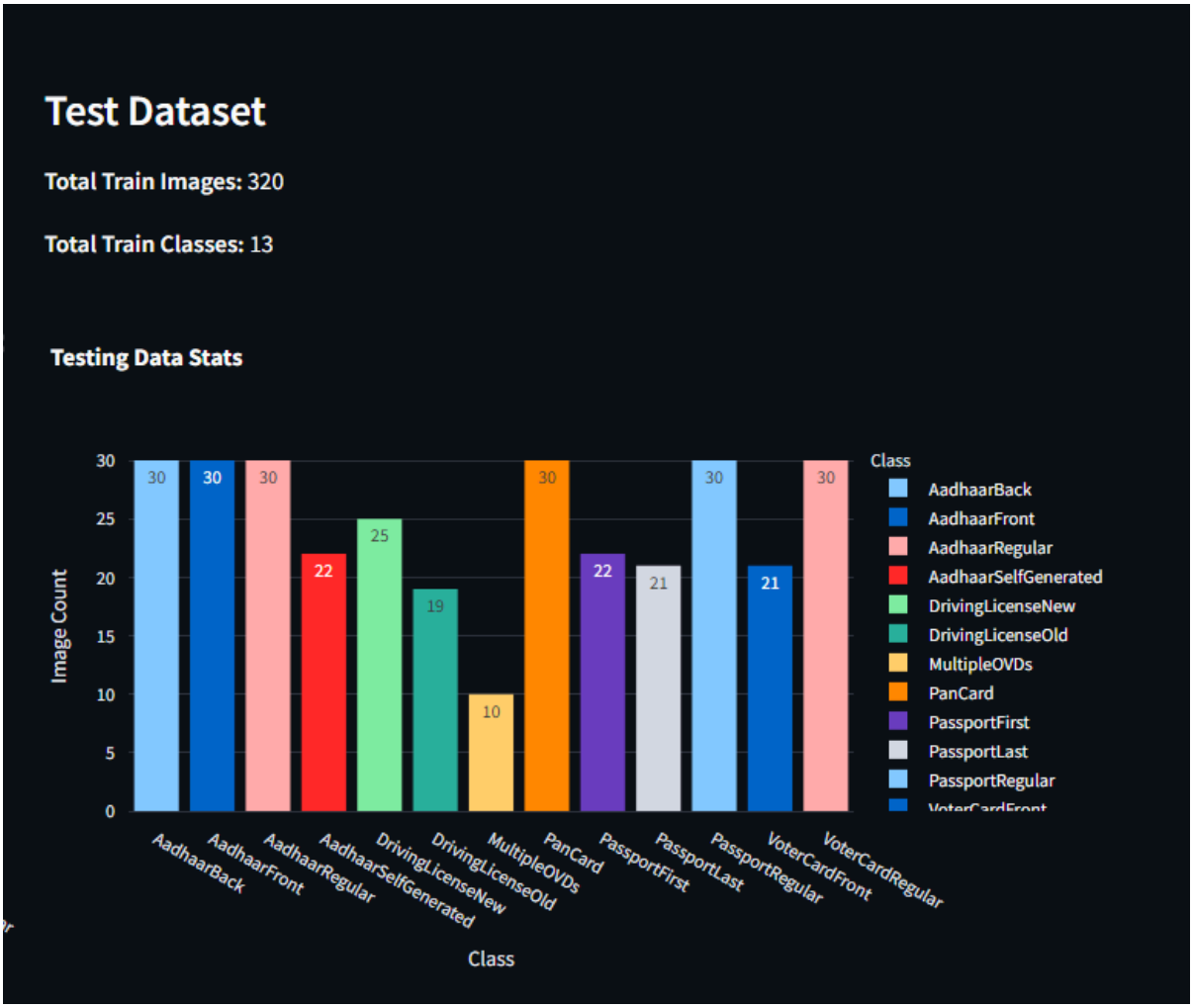
- **Workflow Overview:**
    1. **Upload Dataset -**
        - Upload a new dataset directly through the app by providing zip of dataset or select an existing dataset from a predefined list.
        - The app ensures the datasets are stored in an organized structure for easy access.
        - Basic validation is performed to check the dataset format, ensuring compatibility with the model training process.



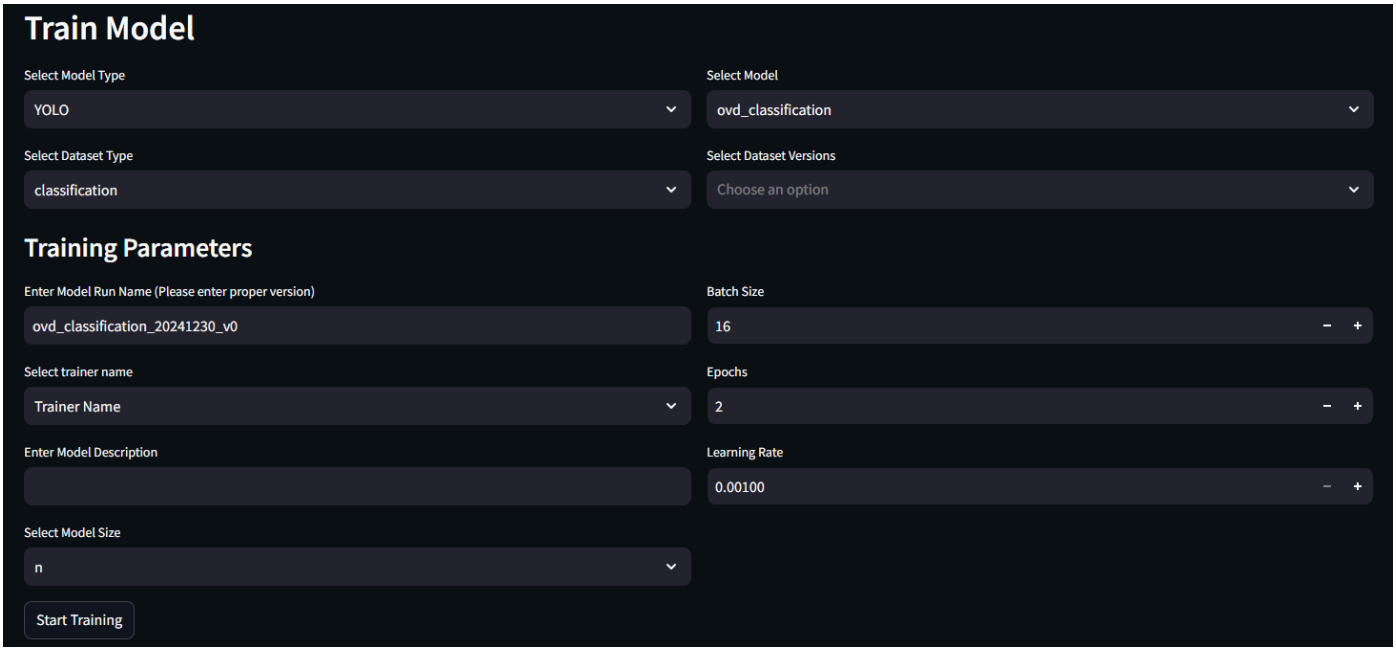


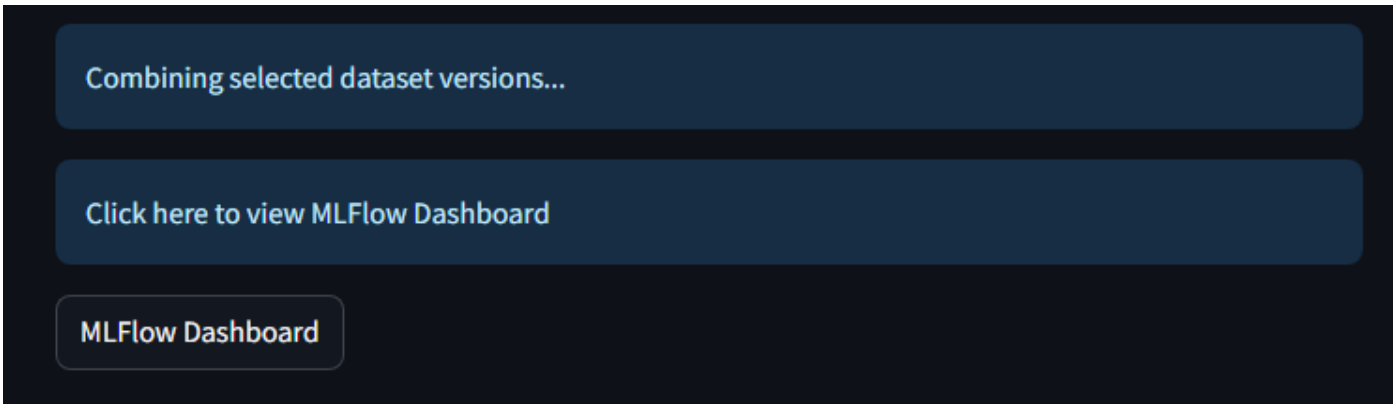
2. **Data Visualization-**
    - The app provides interactive visualization tools to explore the dataset.
    - Users can view data distributions, identify missing or inconsistent values, and understand key features.
    - Common visualization techniques like bar graphs,pie charts, and scatter plots are available to analyze the dataset effectively



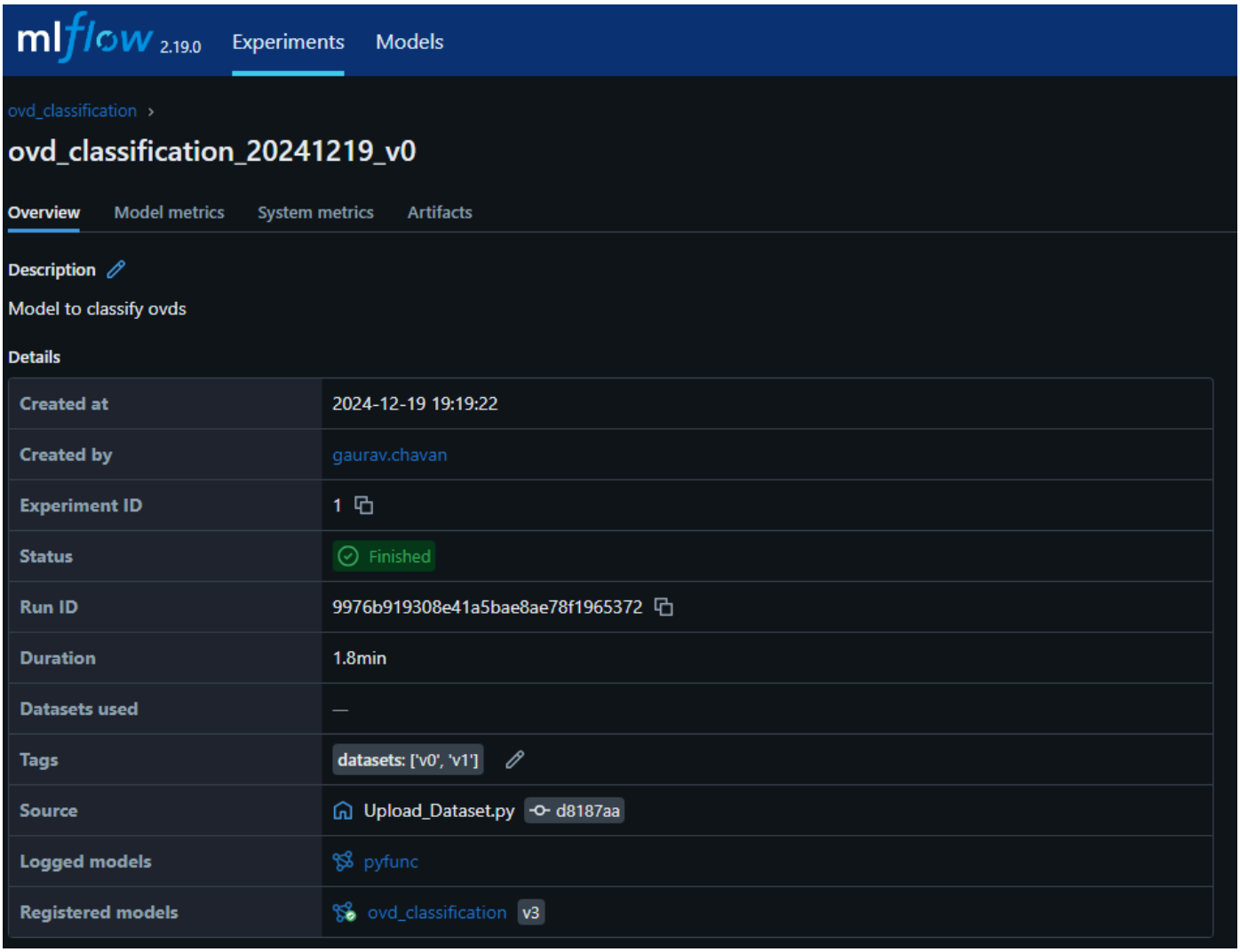


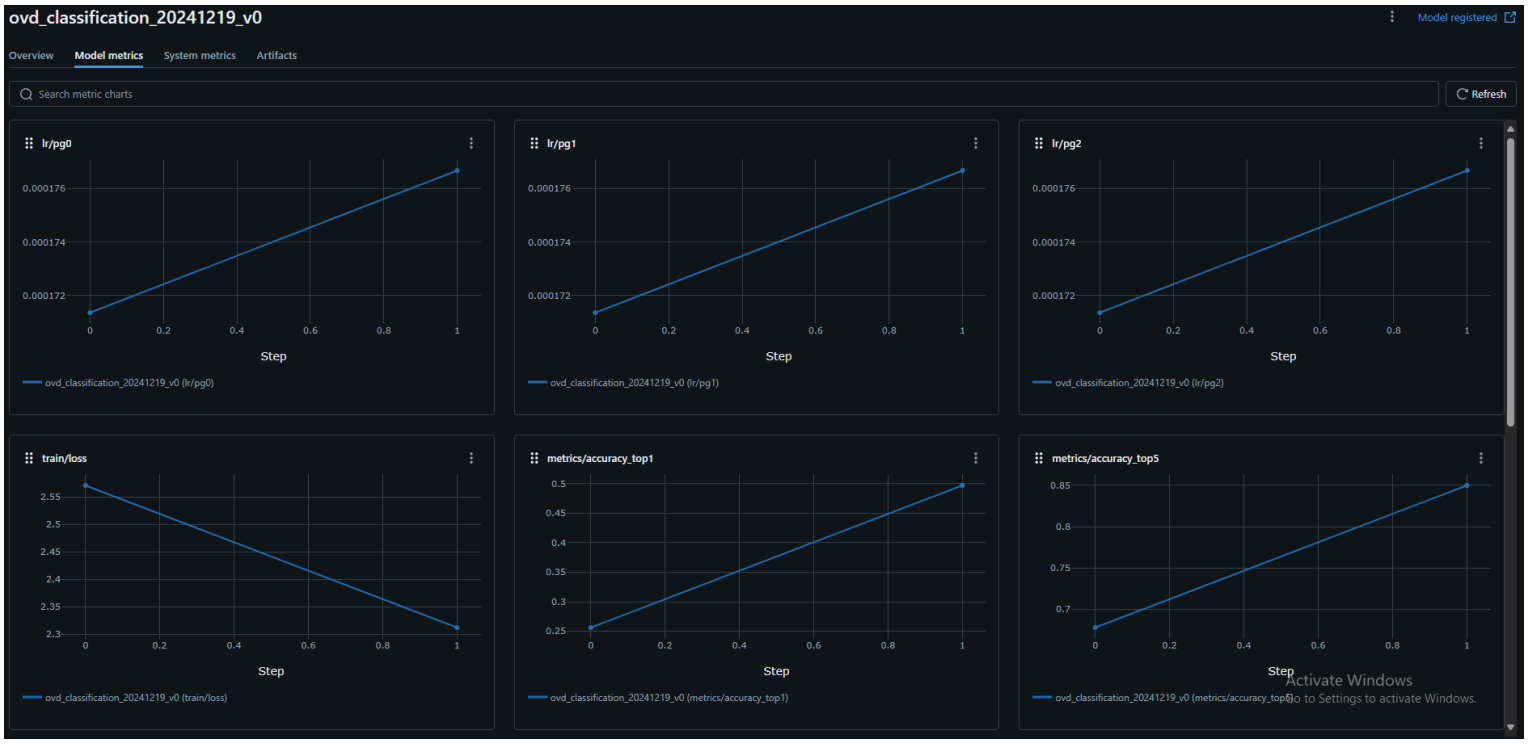
3. **Train the model:**
    - Select a dataset and initiate the training process by specifying the model configuration.
    - Select multiple datasets and train the model on combined dataset.
    - The app supports multiple training parameters (e.g., epochs, learning rate) and allows users to customize them.
    - After training, the app saves logs and provides details of trained model in the MLflow dashboard.

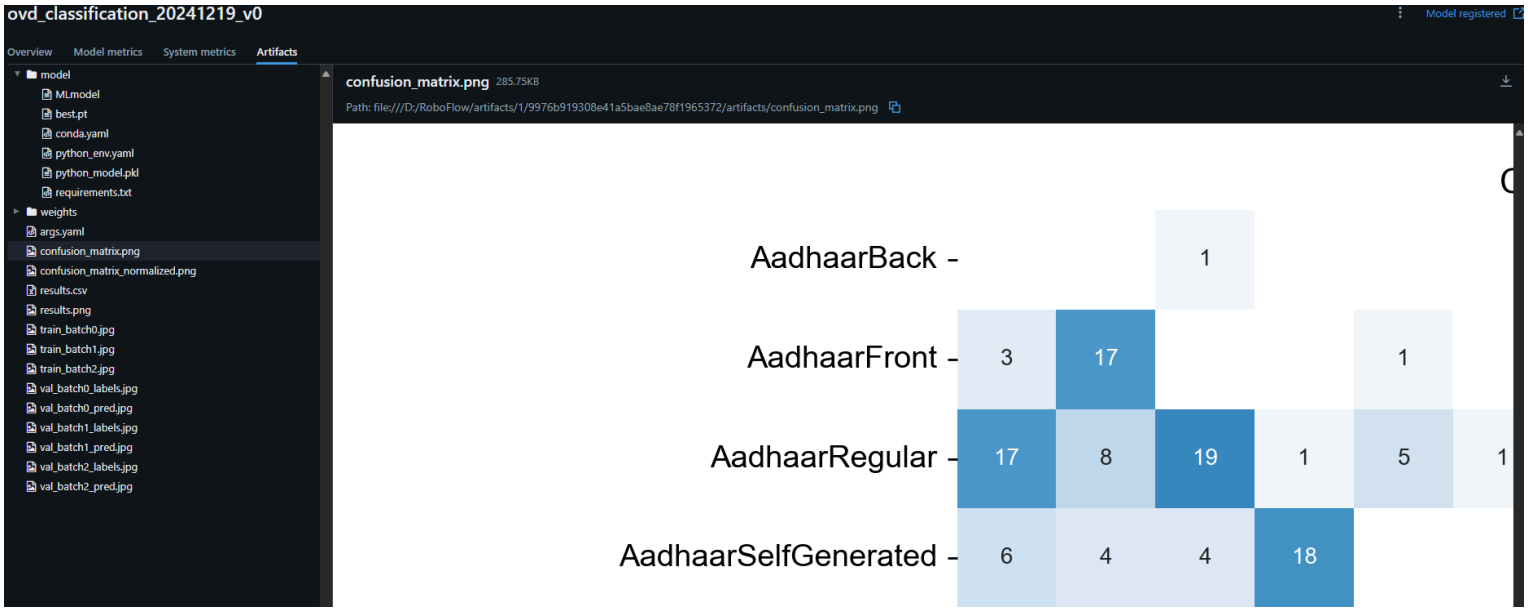




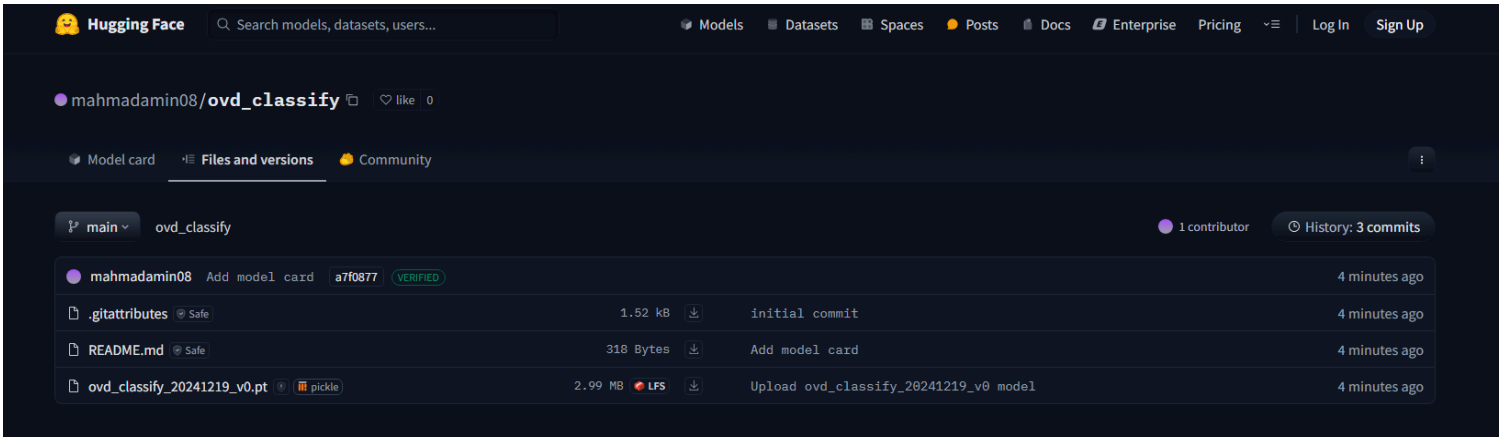
4. **Save Metrics in MLflow:**
    - Once the model is trained, the app automatically logs key performance metrics, such as accuracy, precision, recall, and confusion matrices, into MLflow.
    - MLflow also stores the trained model's version, making it easy to retrieve or compare later.
    - Users can access detailed performance reports through MLflow's dashboard.

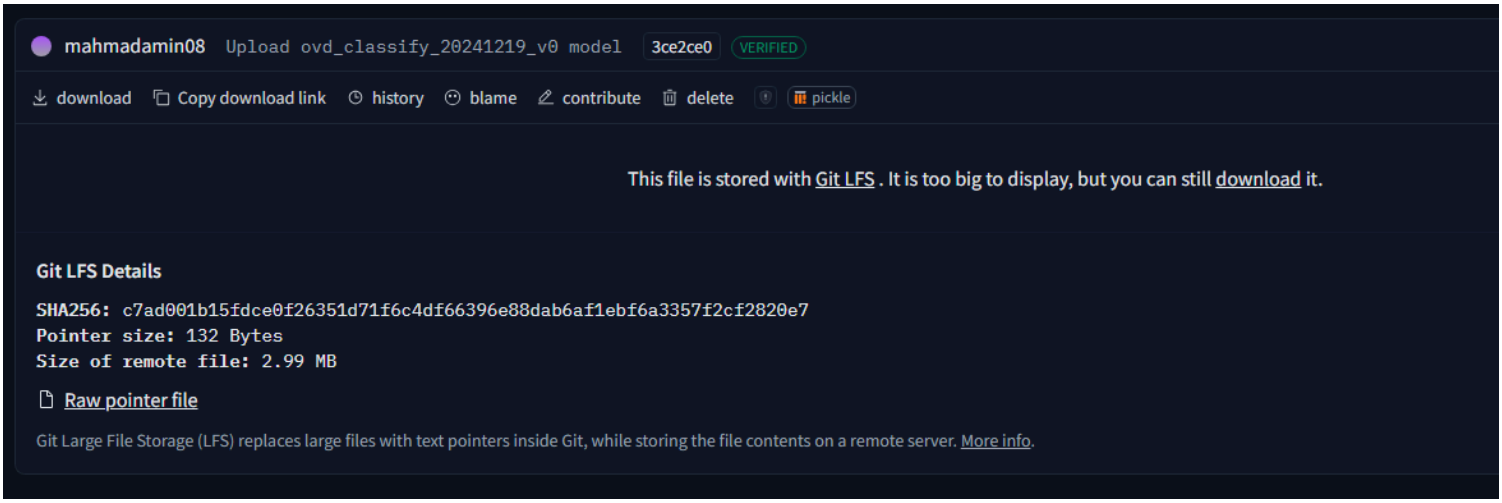



ovd_classification_20241219_v0
Model registered
Overview
Model metrics
System metrics
Artifacts
Search metric charts
Refresh
lr/pg0
lr/pg1
lr/pg2
train/loss
metrics/accuracy_top1
metrics/accuracy_top5
Step
ovd_classification_20241219_v0 (lr/pg0)
ovd_classification_20241219_v0 (lr/pg1)
ovd_classification_20241219_v0 (lr/pg2)
ovd_classification_20241219_v0 (train/loss)
ovd_classification_20241219_v0 (metrics/accuracy_top1)
ovd_classification_20241219_v0 (metrics/accuracy_top5)
Activate Windows
Go to Settings to activate Windows.

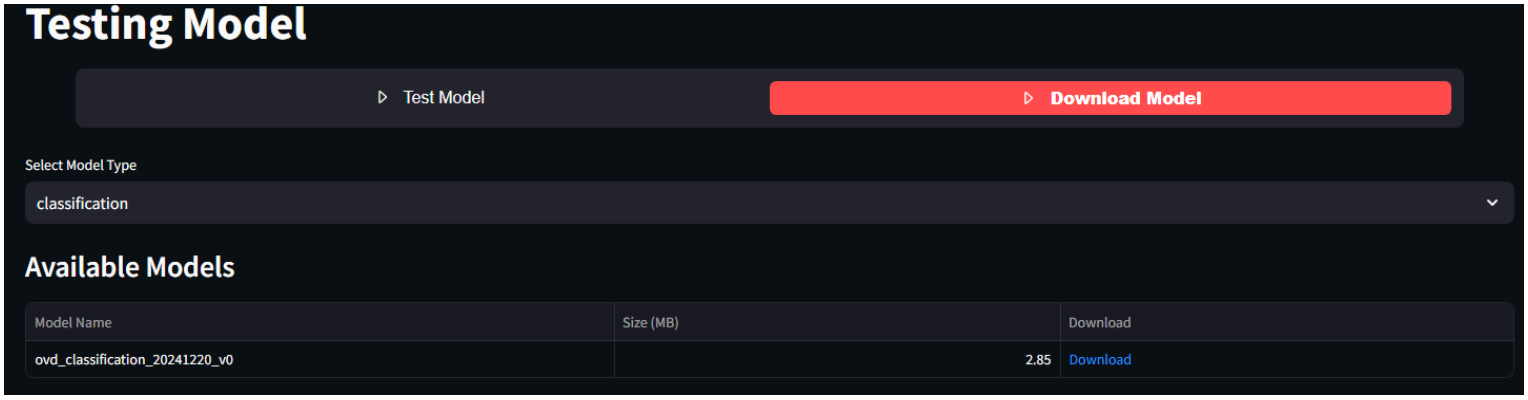

ovd_classification_20241219_v0
Model registered
Overview
Model metrics
System metrics
Artifacts
model
MLmodel
best.pt
conda.yaml
python_env.yaml
python_model.pkl
requirements.txt
weights
args.yaml
confusion_matrix.png
confusion_matrix_normalized.png
results.csv
results.png
train_batch0.jpg
train_batch1.jpg
train_batch2.jpg
val_batch0_labels.jpg
val_batch0_pred.jpg
val_batch1_labels.jpg
val_batch1_pred.jpg
val_batch2_labels.jpg
val_batch2_pred.jpg
confusion_matrix.png 285.75KB
Path: file:///D:/RoboFlow/artifacts/1/9976b919308e41a5bae8ae78f1965372/artifacts/confusion_matrix.png
AadhaarBack
AadhaarFront
AadhaarRegular
AadhaarSelfGenerated
1
3
17
1
17
8
19
1
5
1
6
4
4
18

5. **Deploy the Model to Hugging Face:**
    - After training, the app provides an option to deploy the model directly to Hugging Face.
    - The app ensures the model is converted into the required format and uploads it along with relevant metadata.
    - Once deployed, the model is accessible via Hugging Face's interface for further usage or sharing with others.





6. **Inference Run:**
    - Users can choose from a list of deployed models through an intuitive interface, ensuring flexibility in testing different versions or architectures.
    - The system allows users to upload images directly, providing a seamless method to test the model with custom inputs.
    - Uploaded images are processed in real time, with predictions and associated metrics displayed instantly, enabling immediate evaluation of model performance.

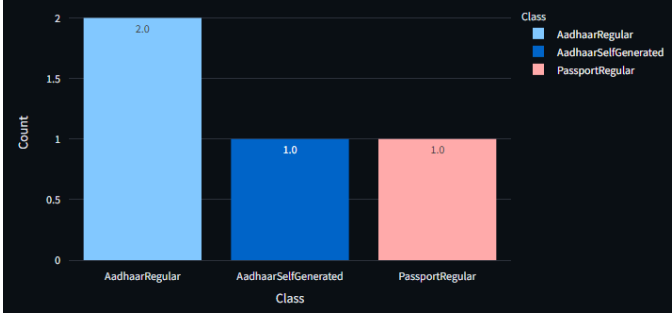


7. **Benchmark Model:**
    - Provide an intuitive interface to choose any deployed model for benchmarking against a predefined test dataset or user-uploaded data.
    - Show annotated examples of both correct and incorrect predictions, with corresponding confidence scores, to provide intuitive insights into model behavior.
    - Generate and display an interactive confusion matrix, illustrating true positives, false positives, false negatives, and true negatives for each class, helping users identify specific areas of model strength and weakness.

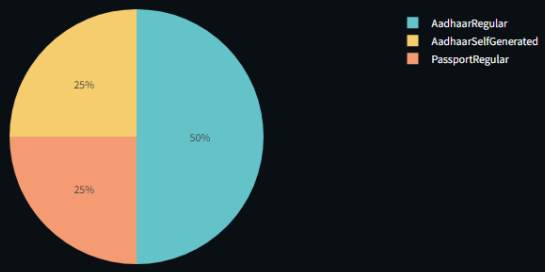
Classified Data Summary

	Class	Count
0	AadhaarRegular	2
1	AadhaarSelfGenerated	1
2	PassportRegular	1

Misclassified Data Stats



Class Distribution

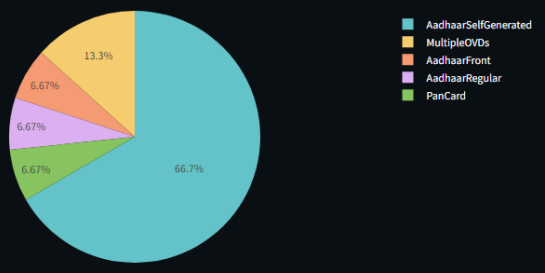


Misclassified Data Summary

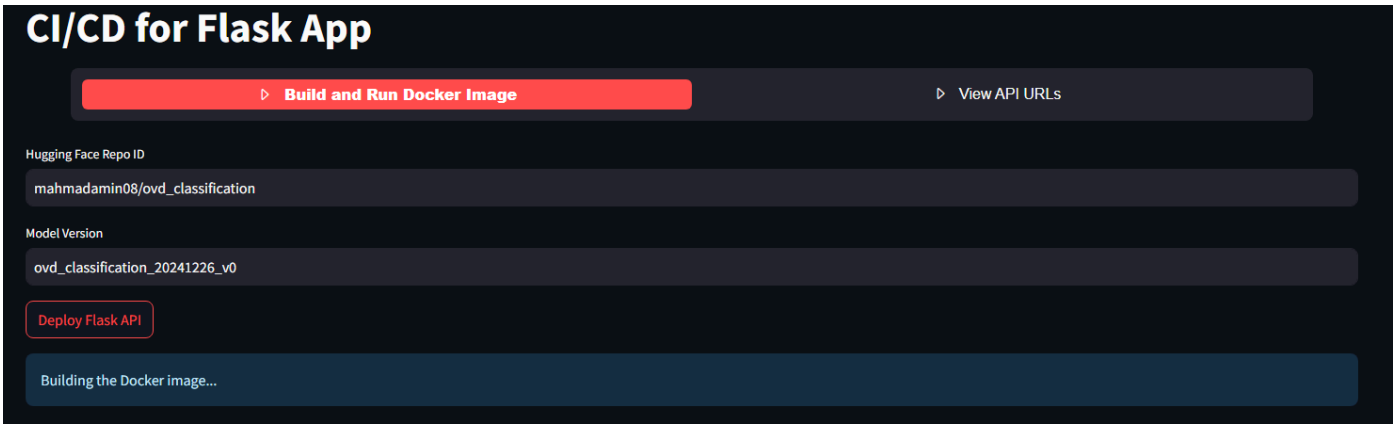
	Class	Count
0	AadhaarSelfGenerated	10
1	MultipleOVDs	2
2	AadhaarFront	1
3	AadhaarRegular	1
4	PanCard	1

Misclassified Data Stats

Class Distribution



8. **CI/CD using streamlit:**
    - Allow users to input the repository ID and model version, enabling automated setup for testing different versions of a model within the CI/CD pipeline.
    - Once the repository ID and model version are provided, the system automatically generates a Docker image by pulling the relevant code and dependencies from the specified repository, ensuring a consistent and isolated environment for running the model.



- **Conclusion:**

  This project provides an integrated platform for dataset management, model training, performance tracking, and deployment. Users can easily upload or select datasets, visualize and analyze the data, and train models with customizable parameters. The app tracks model performance in real-time, logs key metrics into MLflow, and enables easy model deployment to Hugging Face. By automating key processes and providing intuitive tools for exploration and tracking, this project streamlines the machine learning workflow, making it more efficient and accessible for users to train, evaluate, and deploy models.