MACHINE LEARNING PLATFORM FOR PRE-PROCESS AND POST PROCESS INSPECTION

STUDENTS: DAKOTA MATTHEWS, YOUSSEF LAHRICHI, JONATHAN WONG

Quality Control at Boeing

- Manufacturing processes require inspection for quality control
- One example is the appearance of air pockets in an additive manufacturing process
- With the ability to collect image data, Boeing is interested in automating quality control with modern computer vision techniques.

Platform Design

- The purpose of the ML Platform is to make machine-learning approaches accessible to domain scientists who are interested in apply machine learning, but do not have the resources to learn about the technologies in detail.
- With accessibility in mind, we designed the platform in a low-code, autoML manner.
- The user simply provides training data and a model type. The platform will automatically train and save all weights, metrics, and other model information in a reusable output directory.

Platform Implementation

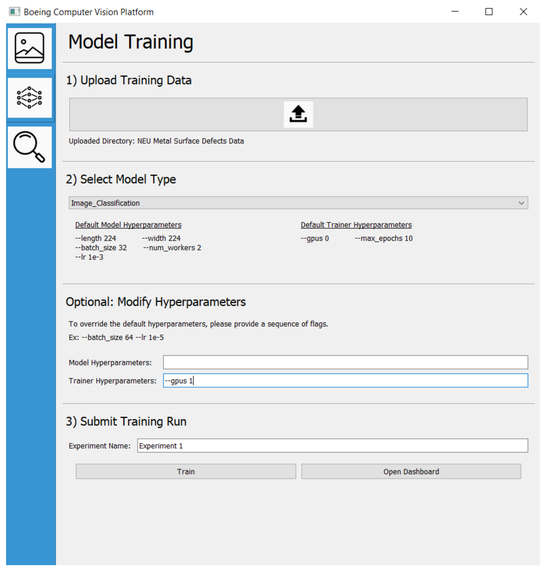
- The platform is designed with a standard Model-View-Controller (MVC) UI architecture.
- Application is written in PyTorch Lightning and PyQt.
- The model logic contains scripts for training and performing inference with a suite of machine learning models.
- The view logic contains logic for visually rendering the UI to the screen.
- The controller logic coordinates all interactions between the model logic and the view logic, . The controller has responsibilities such as responding to user events and writing to the database.
- The primary advantage of the MVC architecture is that each core component of the application is decoupled from the others. This modularity is helpful organization for developers who may seek to extend the application in the future.

Platform Requirements

To use application, we have provided Boeing with a private GitHub.
Included in the GitHub are instructions to setup the application environment and details about directory formatting or raw data.

Platform Features

Model Training Panel:
- Selection among suite of model types.
- Hyperparameter Configuration
- GPU Support
- Tensorboard Logging



Model Prediction Panel:
- Selection among trained models.
- Rename and Delete Functionality
- Image Explorer for Visualization of Model Results

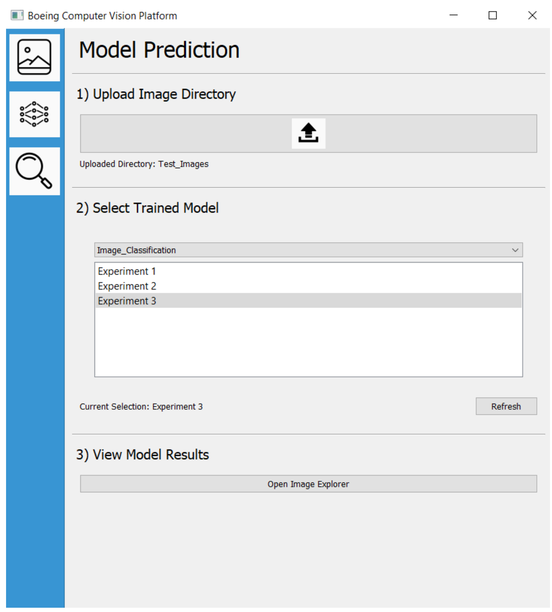
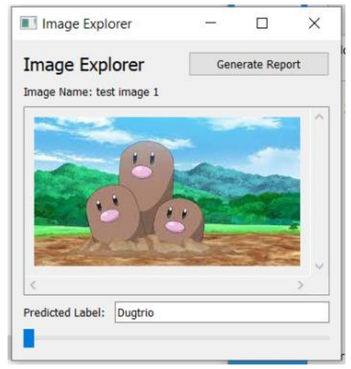


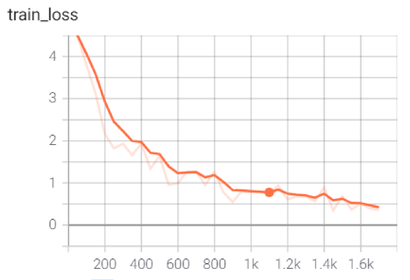
Image Explorer Panel:
- Resizable window for framing any size input
- Image Navigation with arrow keyboard shortcuts
- Report Generation containing information of trained model metrics and model predictions.

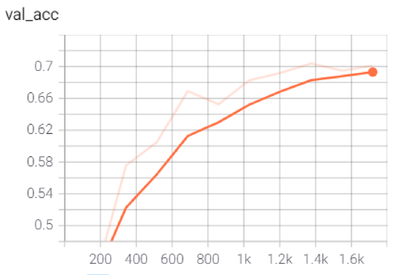


Platform Results

[2]

Trained on Original 150 Pokemon dataset:





Future Work and Extensions

- Client-Server Application Architecture
  - Web UI streamlines deployment and accessibility to end user.
  - Enables application to scale compute with distributed computing achieved through packages such as Ray.
- Database Infrastructure.

At scale, the application should have a dedicated data warehouse, as well as an automated means to access and transform the raw data into training data. Large ecosystem of tools exist here as well including Airflow and Prefect.

- Application Scale.

Each component of the application may be containerized and managed through Kubernetes for greater modularity, maintainability, and parallelization.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISOR: DAVID KROUT
SPONSOR: BOEING