Allison DeCicco
Billy Miller
Lian Showl

The Average Joe Classifier
Design Homework One

Product:

Our product is a classifier for someone who does not know/understand machine learning. The user will be able to upload their dataset, label the object(s) they want to classify, and then create a classifier without a large time commitment or an understanding of what is happening.

The design represents good software engineering practices because we are not building a monolithic architecture. Since this project is a group effort, the way that our project's design naturally structured into different microservices (see components section below). Because the product is broken down into separate services, the product can be developed in parallel which takes advantage of the team.

Systems:

Our system includes a webpage communicating with a database and python component. The interactions are visually mapped in the diagram labeled "Architecture Diagram". The Javascript UI communicates with the Python which trains the You Only Look Once (YOLO) classifier and then stores its predictions in the database. The UI will display the results of the training from the database to the user so that they can properly correct and retrain the classifier. The Python script reads the images from the database, trains the classifier, and then saves its predictions back to the database for the UI to display.

Containers:

- User Interface: This includes the Javascript user interface, the python scripts, and the YOLO command line interface.
- Database: We are using a nonrelational database (MongoDB) so that it grows dynamically as our users upload images. The database is being hosted on a SEAS server through a docker instance.

Components:

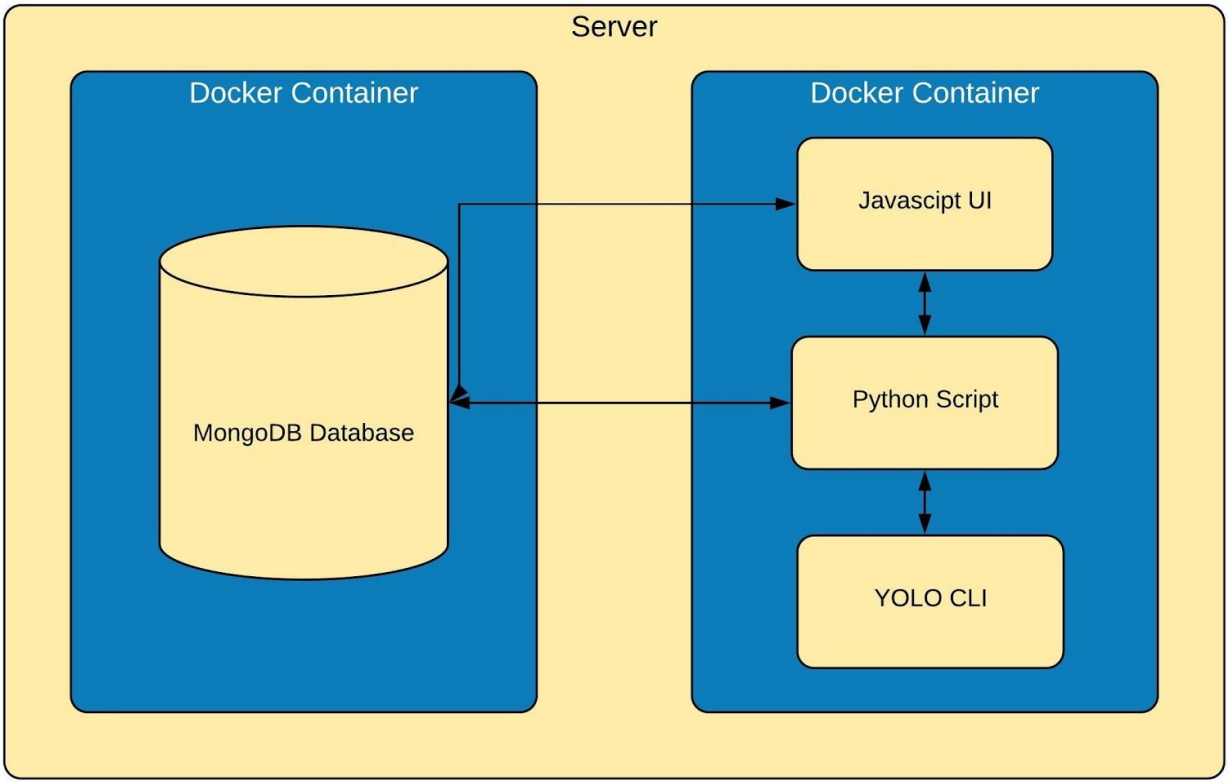
The components in the User Interface container are the Python, Javascript, and the YOLO command line interface. For the Javascript to interact with Python, it is being done as a command line interface. They Python script is written so the Javascript will run it with certain arguments to get the results it needs. The YOLO CLI is written in the same way as the Python, so the Python script then calls the YOLO CLI for the actual training and testing. The Python script is also responsible for data manipulation. It will communicate with the MongoDB database using the pymongo Python package. The script will need to load the images into memory for training and testing and then store its results back to the database.

The UI is interacting with our database by first creating a connection with the database. It then makes requests based on what the user inputs. One example is it creates a request to store the image. When it does this, the program creates a file path. The image is being split up so that it can be properly stored in the database.The UI code is an ejs file. The ejs allows us to have JavasScript embedded into our HTML. Another request example is in the ejs file. There is a form action where it submits a POST request. The request is submitted when a picture's

delete button is clicked. The picture's id is submitted and then deleted from the database based off it's ID. The annotator is also using JavaScript.

The annotation tool uses jQuery to use its event handling functions. Since the user draws bounding boxes on the image, the event handling is necessary to receive the coordinates of the desired selection in real-time. The functions are showCoords(), checkCoords(), clearCoords() and showPreview(). The functions are called within the UI to work together to show the user's selections.The coordinates that the user maps are handed off to the Python script for storage into the database. There will be many bounding boxes for the user to select more than one object in a single picture using OpenCV. The multiple bounding boxes will allow the user to label many instances of the object to then be handed to the Python script.

Architecture Diagram:



Responsibilities:
Billy: YOLO, Python, Database Setup
Allison: Basic UI design and interaction with the database
Lian: Annotator component with UI - label multiple bounding boxes in a single image