**AI ALGORITHMS**

**AIDI-1002-01**

**JEFFREY ARUNASALAM**

**100783993**

**Data Acquisition and Understanding + SOW (V2)**

**Executive Summary:**

The general proposal is creating an AI web client that Create a 3-dimensional equivalent object from analyzing the 2-dimensional image inputted. [3D PORTRAIT OF 2D IMAGE]

Out of scope (limitations) of the project is relaying/recreating textures (texture mapping) from the image onto the 3D object, attempting to create weight paint and bones (animation recreation purposes).

**Rationale Statement:**

2D to 3D conversions are usually paywalled, on sketchy sites, or not that reliable (non-payed, advertising sites). The goal of the project is to make this type of stuff more accessible to everyone.

**Problem:**

Website client needs to be created for users to start accessing the website.

Currently using NumPy-STL to output mesh as STL and renderer in Blender (not using any custom renderer, just outputting meshes).

**Preliminary Data manipulations Requirements:**

-Resizing all images and turning them to grey (using pillow module)

-convert to NumPy array data (to save data using pickle, and manipulate the image data via arrays)

-\*Datasets can be any be pretty much any type objects of images (varying, not all the same)

**Feature Engineering**

-use pytorch3D module to create a renderer (instead of relying on blender)

**Data & Model/Architecture Approach:**

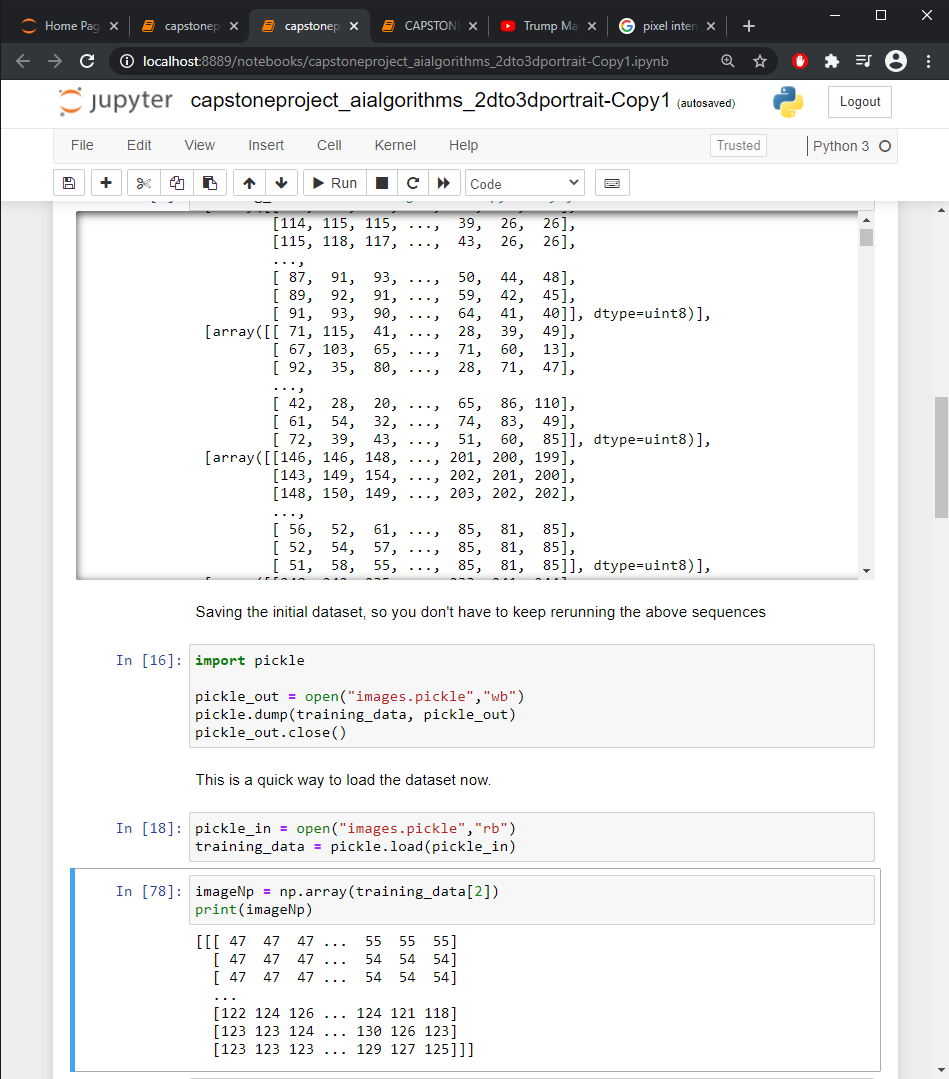
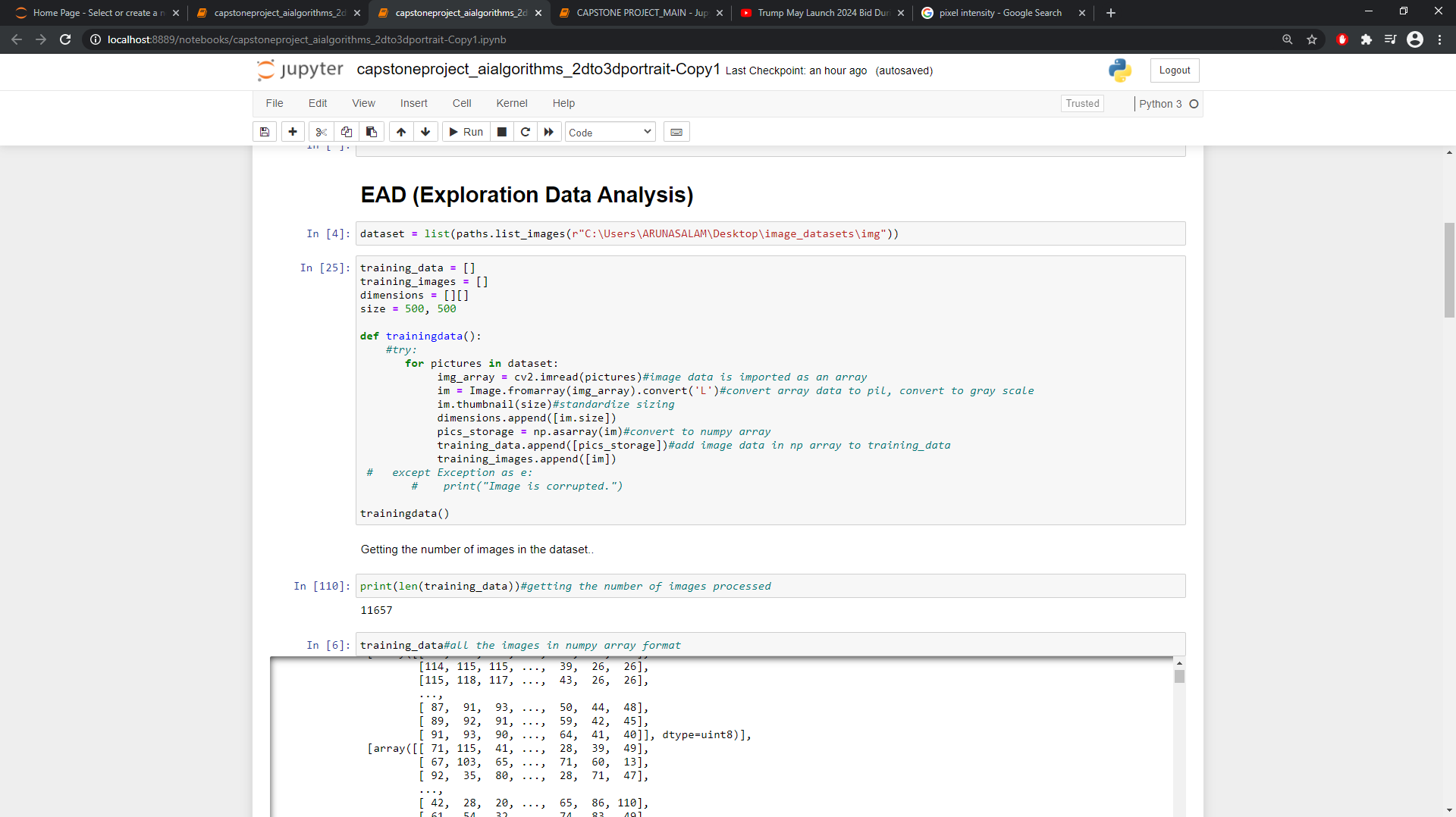
\*Approach is now different instead of using multiple images of different views, now a single image is used and using pixel intensity map a “3D Portrait” of the image (previous data & model architectures was too advanced, not enough time (time constraints) for the scope of the assignment/project.

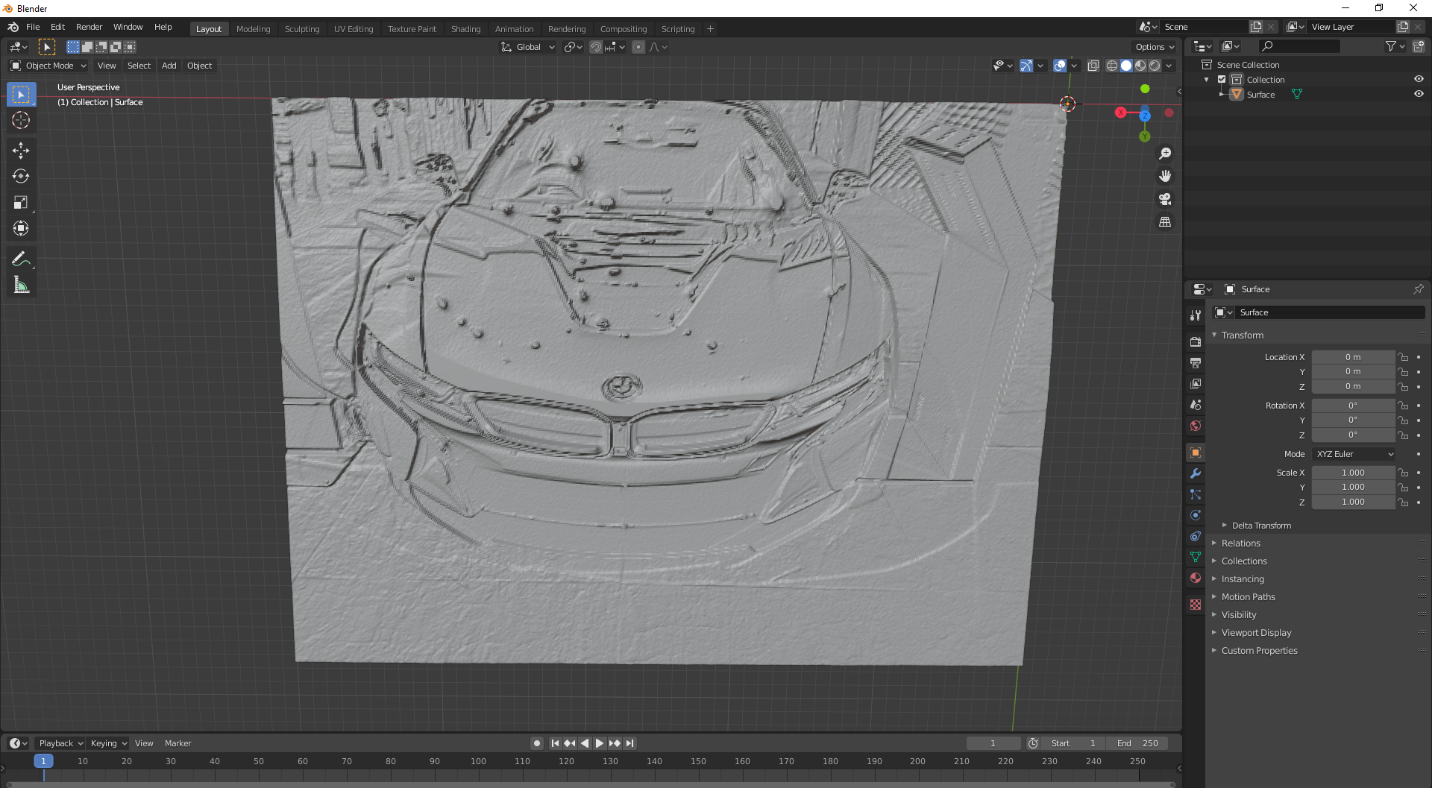
<https://pypi.org/project/numpy-stl/>

<https://pillow.readthedocs.io/en/stable/>

Prototype inputs and outputs (single run input, no ai) using NumPy array data from the image to calculate pixel intensity:

\*Use tensorflow object detection to attempt to only get major objects in a mesh, instead of the whole image.



current simulation (no ml):