**INTERNSHIP PROJECT**

NAME: B SHYAM SUNDHAR

TITLE: 3D RECONSTRUCTION

INTERNSHIP DURATION: 26.6.24 – 10.7.24

**Abstract:**

I developed a 3D reconstruction model capable of generating a 3D model from a single RGB image provided by the user. This model has been integrated into a web application using Flask, allowing users to view the generated 3D model directly within the web-app.

**Technologies used:**

**MiDaS:**

A pre-trained deep learning model available from the PyTorch Hub, designed for monocular depth estimation. It predicts the depth value for each pixel in a single RGB image, creating detailed depth maps from 2D images. The model leverages advanced neural network architectures for accurate depth inference.

**OpenCV:**

Used in image processing, feature detection, and matching, offering tools for filtering, edge detection, and morphological operations. It provides robust algorithms for extracting and matching key points and descriptors, useful for object recognition, tracking, and image stitching. Additionally, OpenCV integrates with neural networks and CNNs, supporting frameworks like TensorFlow, PyTorch, and Caffe for tasks such as image classification, object detection, and semantic segmentation.

**Open3D:**

A versatile library that excels in handling and processing 3D data, including point clouds, meshes, and voxel grids. Specifically, Open3D provides advanced algorithms and utilities for point cloud processing. These include capabilities for filtering, down sampling, registration, and segmentation of point clouds.

**Three.js:**

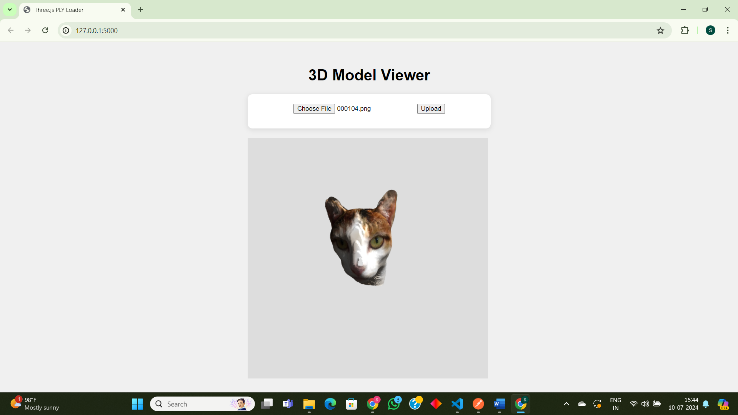
A JavaScript library that enables developers to create and display dynamic 3D graphics and animations directly within web browsers. It harnesses the power of WebGL, allowing for the rendering of complex scenes, interactive visualizations, and immersive experiences seamlessly integrated into web applications.

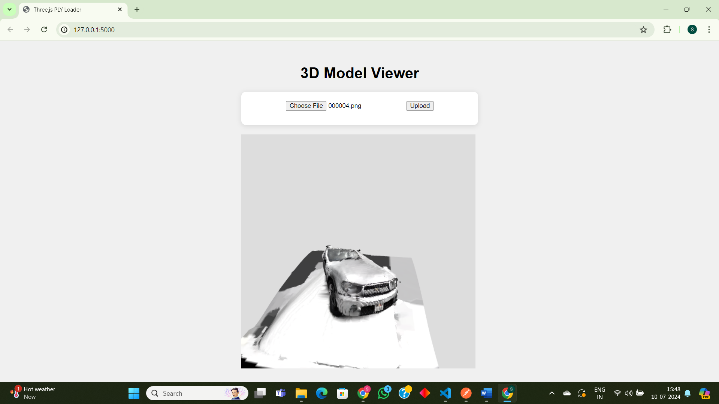
**Working:**

1. User uploads an image, initiating the workflow.
2. The image is transmitted to the backend via Flask for processing.
3. Upon receiving the image, backend processing begins, focusing initially on depth estimation using the MiDaS model.
4. Using Open3D, an RGB-D image is created from the original RGB image and the depth map (o3d.geometry. RGBDImage.create\_from\_color\_and\_depth (rgb\_image, depth\_image)).
5. The RGB-D image is then used to generate a point cloud, utilizing camera intrinsic parameters (o3d.geometry. PointCloud.create\_from\_rgbd\_image (rgbd\_image, camera\_intrinsics)).
6. Subsequently, a mesh representation is derived from the point cloud using the Poisson surface reconstruction method in Open3D(o3d.geometry.TriangleMesh.create\_from\_point\_cloud\_poisson(pcd, depth)).
7. The resulting mesh is saved as a .ply file format using Open3D's file I/O capabilities( o3d.io.write\_triangle\_mesh(output\_file\_path, mesh)).
8. The generated .ply file is sent back to the frontend via Flask for visualization.
9. On the frontend, the received .ply file is visualized using the Three.js library, employing THREE.PLYLoader() directly within the webpage.

**Sample output Screenshots:**

Sample Input image: Reconstructed 3d output:

References:

<https://medium.com/@AaronLeeIV/3d-reconstruction-from-2d-images-cb21096631ad>

<https://www.youtube.com/watch?v=DoZJaqBzSso>

<https://www.open3d.org/docs/latest/tutorial/Advanced/surface_reconstruction.html>

<https://pytorch.org/hub/intelisl_midas_v2/>

<https://github.com/niconielsen32/ComputerVision/blob/master/pointClouds/pointCloud.py>

<https://github.com/AbdulRehman555/3D-Mesh-Generation/blob/main/reconstruction.ipynb>

**Conclusion**

In completing the 3D reconstruction model for the web app within a limited internship duration of two weeks, significant progress has been achieved, yielding above-average accuracy. However, it is acknowledged that further enhancements could be made by training the model with a larger dataset comprising multiple images and their corresponding 3D models. This iterative approach would likely lead to improved accuracy and robustness. Despite the time constraints, the project successfully demonstrated the feasibility of integrating 3D reconstruction into a web application, laying a solid foundation for future development and refinement. Moving forward, dedicating more time to dataset augmentation and model training would be essential to realizing a more sophisticated and accurate solution.