To IITD-AIA FSM

Subject: Weekly progress report (week-1)

https://github.com/Kush-2103/FSM-INT-2023/tree/main/Week%201

Dear Sir,

In this week, I have Completed the learning task given to me and also implemented the learnings in python for practice. I was also allotted with project in this week. I researched and understood about the libraries needed and the approach for this project.

Topics covered-

- Deep Generative modelling
- Reinforcement learning
- Face clustering lab and also learned about Streamlit
- Gantt chart
- Open3D
- Preprocessing of the 3D point cloud

June 5 Deep Generative Modelling

Generative modelling takes input training samples and learn a model that represents that distribution. It is an example of unsupervised learning.

2 methods- density estimation, sample generation.

It helps in

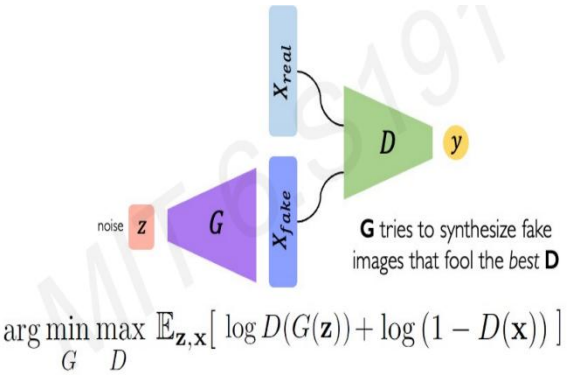
Debiasing- capable of uncovering underlying features in dataset.

Outlier Detection- Detect outliers to avoid unpredictable behaviour when training.

Latent variables are the true explanatory features or hidden factors in the observed data. Then I learnt about Latent Variable models

1) **Autoencoders**- encoders and decoders are used. It is a form of compression. Dimensionality of latent space is directly proportional to reconstruction quality. Reconstruction loss is used for capturing more data information.
2) **Variational auto encoders** are used in practical purpose. Some randomness to latent space will be given such that image reconstructed will be similar but not strictly same. It will have some diff feature. We define a mean and variance over that latent variable which capture the probabilistic distribution. Loss function consists of reconstruction loss and regularization term.
   Connectivity + Completeness is ensured by the regularization term. It helps in generating similar data for closer related latent space and meaningful data respectively.

3) **Generative Adversarial Networks (GANs)-** 2 neural Networks compete with each other. Generator turns the noise into the meaningful data and discriminator compare the Real data and fake data generated from Generator. These are distribution transformers.



$$\arg\min_G \max_D \mathbb{E}_{\mathbf{z},\mathbf{x}}[\log D(G(\mathbf{z})) + \log(1 - D(\mathbf{x}))]$$

We can control the nature of output by putting the condition on label.

Cycle GAN- used for domain translation.

Conditional GAN- paired translation.

June-6 Reinforcement Learning

Then I learn about the Reinforcement learning (RL), it is a machine learning subfield that focuses on sequential decision-making. In RL, an agent interacts with an environment and they receive rewards.

The goal of RL is to maximize cumulative rewards over time.

In RL, an agent interacts with an environment composed of states, takes actions, receives rewards, and learns to maximize its cumulative reward over time

Agent: The agent is the learner or decision-maker that interacts with the environment

State: A state represents the current situation or configuration of the environment. It contains all relevant information necessary for decision-making.

Action: An action represents the decision made by the agent in a particular state. It can be discrete or continuous.

Reward: A reward is a numerical feedback signal provided to the agent after it takes an action. It acts as the metrics for determining the quality of algorithm.

Q-function: It is also known as action value function.

It estimates the expected cumulative reward when the agent takes a particular action in a specific state and follows a specific policy thereafter. The Q-function is represented as Q(s, a), where 's' is the state and 'a' is the action. It guides the agent's decision-making by evaluating the potential outcomes of different actions.

Learned how to implement policy gradients. **Training policy gradients** is a popular approach in reinforcement learning (RL) for optimizing policies directly through gradient-based methods.

Conclusion is RL provides a framework for an agent to learn through trial and error, while policy gradients offer a specific approach to directly optimize policies by estimating and updating the policy parameters.

Applications: Game playing, Robotics, Autonomous vehicles, health care, financial

Learned about how to use RL in every application which gave me the idea of this algorithm.

June 7 Face clustering and Streamlit

I implemented the face clustering model using DBSCAN algorithm. I also used the CNN model for getting the face encodings.

The project has been deployed as a Streamlit Web App which provides users the facility to upload their own images (or images they want to test with) and be delivered with the images grouped according to the individual unique faces contained in them.

I learned about various clustering Algorithms such as DBSCAN, Kmeans, spectral clustering.

Code for this project is given in my github-

https://github.com/Kush-2103/FSM-INT-2023/blob/main/Week%201/face_cluster%20(1).ipynb

https://github.com/Kush-2103/FSM-INT-2023/blob/main/Week%201/app.py

JUNE-8 INTP23-ML-08 (Computer vision to detect defects in 3D Printed parts).

Started with the objective and scope of the project by researching through open source and Google.

Scope-This project will help in the early detection of defect and will save cost and wastage, which will be beneficial to the company. It will increase quality control and efficiency.

Objective- Do the feature extraction to detect defects and then classify the defects.

Researched about the 3D printing process and learned about various types of defects we need to classify, as given in the statement.

- Warping, layer misalignment, Void, Surface imperfections, Structural irregularities.

Prepared the rough draft for implementation of the project. It will be used in making a **Gantt chart of 50 days.**

It can be done in 7 phases (Data preprocessing, Algo development, detection of defects, defects classification, visualization, Testing, Deployment, and fine-tuning)

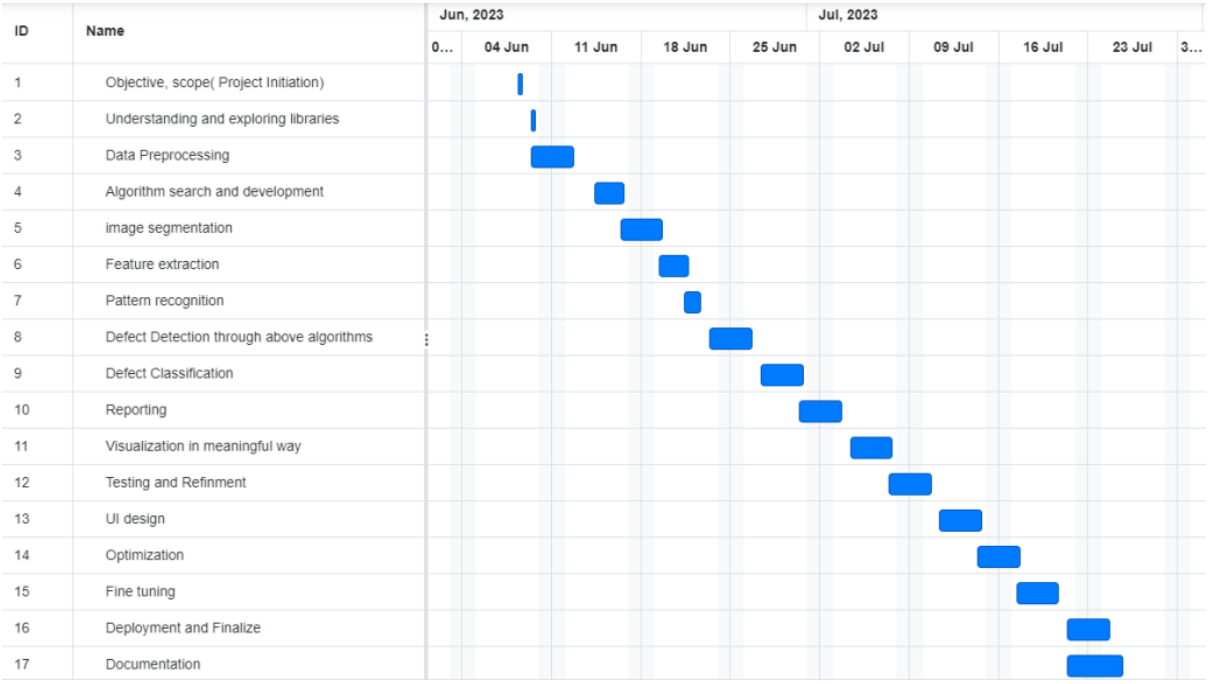
Also got some ideas of libraries that can be used- Point Cloud Library, OpenCV, skit-learn,numpy, pandas, and TensorFlow. Will study it in depth, how can it be used, and the functions of every library which can be used in the model.

JUNE-9

Yesterday, I divided the project into 7 phases. So, To create Gantt chart of 50 days, I divided every phase more finely.

There are 17 tasks in my project for its end to end deployment.

I prepared a gantt chart using the free online software



Issue Faced- Started with installing PCL(Point Cloud Library), Faced many problems because this library was last updated 5 years ago. Tried installing it with binary files, whl files, github clone. Nothing worked.

Solution-So, I moved on to its alternative library, open3D it can also perform pre-processing on 3D images. library provides many more functionalities for point cloud and mesh processing, such as filtering, segmentation, surface reconstruction.

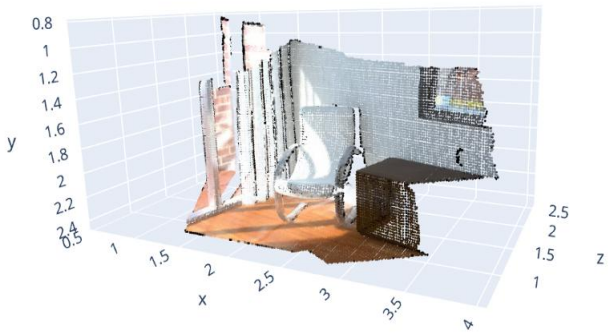
It reads point cloud file and do the operations on it.

Challenges faced- PCL library not getting installed, tried various method.

JUNE-10

Today, I started with the learning of the Open3D library. Learned about how 3D images are pre-processed and various operations that can be done which will help me in the project.

Open3D works on point clouds, it is a collection of 3D points in space.Each point can contain extra information about colours and surface normals. It is useful in CV, robotics, AR.



Implemented the **pre-processing** operations needed before proceeding for ML model.

Downsampling- create a uniformly downsampled point cloud from an input point cloud by bucketed the points into voxel.

for **filtering**, remove the outliers using statistical values.

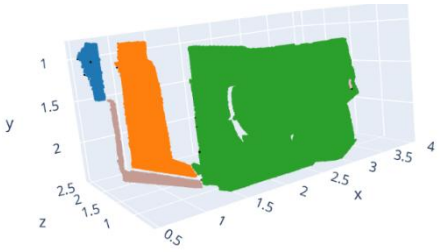
Normalization- normal of the downsampled point cloud.

crop the point cloud to focus on a specific region of interest. We can use the Crop_box function. It can be used, depending on the dataset.

For **Feature Extraction,** I did the 2 operations

1) **DBSCAN Clustering**- group local point cloud clusters together

2) **Segmentation**- segmentation of geometric primitives from point clouds using RANSAC



Apart from this, I also learned how to convert 2d image to 3d pimage and pcd file. It can be done using using open3d. only. having two files 2d image and corresponding depth map.

features obtained after the feature extraction, can the be used for training the model and classify the detection.

Code link- https://github.com/Kush-2103/FSM-INT-2023/blob/main/Week%201/open3D.ipynb