**COMPUTER VISION ASSIGNMENT**

**Question 1 -**

Train a deep learning model which would classify the vegetables based on the images provided. The dataset can be accessed from the given link.

Link

https://www.kaggle.com/datasets/misrakahmed/vegetable-image-dataset

Note - 1. Use PyTorch as the framework for training model

2. Use Distributed Parallel Training technique to optimize training time.

3. Achieve an accuracy of at least 85% on the validation dataset.

4. Use albumentations library for image transformation

5. Use TensorBoard logging for visualizing training performance

6. Use custom modular Python scripts to train model

7. Only Jupyter notebooks will not be allowed

8. Write code comments wherever needed for understanding.

**Answer 1 CV:-**

**Question 2 -**

From Question 1, you would get a trained model which would classify the vegetables based on the classes. You need to convert the trained model to ONNX format and achieve faster inference

Note - 1. There is no set inference time, but try to achieve as low an inference time as possible

2. Create a web app to interact with the model, where the user can upload the image and get predictions

3. Try to reduce the model size considerably so that inference time can be faster

4. Use modular Python scripts to train and infer the model

5. Only Jupyter notebooks will not be allowed

6. Write code comments whenever needed for understanding

**Answer 2 CV:-**

**Question 3 -**

Scrap the images from popular e-commerce websites for various product images sold on those websites. Your goal is to fetch the images from the website, create categories of different product classes and train a deep learning model to classify the same based on the user input.

Note - 1. You can use any framework of your choice like TensorFlow or PyTorch

2. You have to not use any pre-trained model, but instead create your own custom architecture and then train the model.

3. Write code comments wherever needed for understanding

4. Try to use little big dataset so that model can be generalized

5. Write modular Python scripts to train and infer the model

6. Only Jupyter Notebook will be not allowed

7. Write code comments wherever needed for code understanding

**Answer 3 CV:-**

**Question 4 -**

You have to train a custom YOLO V7 model on the dataset which is linked below. Your goal is to detect different products based on the given classes based on the user input Link - https://drive.google.com/file/d/1MEgDYJwO\_PVVfAbyfjaRHXt7qoiBBHYt/view? usp=share\_link

Note - 1. You have to use PyTorch implementation of YOLO V7

2. The dataset consists of 102 classes with train, validation, and test images already in the respective folders. 3. Labeling is already done, given with the dataset, so need for annotation

4. Since the dataset is small, try to achieve at least an mAP of 85

5. Write modular Python scripts to train the model

6. Write code comments wherever needed for understanding Computer Vision Assessment iNeuron 3

7. Only Jupyter Notebook will not be allowed

**Answer 4 CV:-**

**Question 5 -**

From Question 4, you would have a custom-trained YOLO model. Your goal is to need to convert the model to ONNX format and reduce the inference time. Note -

1. Reduce the inference time to as much as possible

2. Try to reduce the model size by using techniques like Quantization, etc

3. Create a web app for users to interact with your model where users can upload images and get predictions. 4. Write modular Python scripts to infer the model.

5. Only Jupyter notebooks are not allowed.

6. Write code comments wherever needed for code understanding

**Answer 5 CV:-**

**Question 6 -**

You have to train a custom segmentation model based on Detectron 2 framework. Your goal is to segment the given images based on the user input into the different classes Link - https://www.kaggle.com/competitions/open-images-2019-instance-segmenta tion/data

Note - 1. For this, only the Jupyter Notebook is fine

2. Labels are in COCO format.

3. Write code comments wherever needed for understanding

**Answer 6 CV:-**

**Question 7 -**

From Question 6, you would have custom trained segmentation model. Your goal is to reduce the model inference time

Note - 1. Reduce inference time to as much as possible

2. Create a web app for users to interact with your model where they can upload images and get predictions 3. Write code comments wherever needed for code understanding.

**Answer 7 CV:-**

To reduce the inference time of a custom trained segmentation model, you can employ various techniques. Here's an approach you can take:

1. Model Optimization:
   * Quantization: Apply quantization techniques such as post-training quantization or quantization-aware training to reduce the precision of model weights and activations, thereby reducing memory usage and improving inference speed.
   * Pruning: Use pruning techniques to remove unnecessary connections or channels from the model, reducing the model size and inference time.
   * Model architecture optimization: Consider using lighter model architectures that trade off some accuracy for faster inference, such as MobileNet, EfficientNet, or lightweight variants of existing architectures.
2. Inference Optimization:
   * Batch processing: Perform inference on multiple images simultaneously by batching them together. This utilizes parallel processing capabilities of modern hardware and speeds up inference.
   * GPU/CPU optimization: Ensure that your model and data are properly placed on GPUs or CPUs based on their availability and capabilities.
   * Input preprocessing: Optimize image resizing, normalization, and other preprocessing steps to minimize the time spent on these operations during inference.
3. Web App Integration:
   * Create a web app using a web framework like Flask or Django.
   * Implement an image upload functionality in the web app to allow users to upload images.
   * Integrate the custom trained segmentation model into the web app and perform inference on the uploaded images.
   * Display the segmentation predictions to the users and provide the option to download or visualize the results.

**Question 8 -**

You have to train a custom object detection model based on DETR (Detection Transformer)

Link - https://www.kaggle.com/datasets/andrewmvd/helmet-detection

Note - 1. You need to use HuggingFace PyTorch as the framework

2. The dataset is about detecting football players from the images provided

3. Data Annotations are already in COCO format.

4. Write custom Python scripts for training.

5. Write code comments wherever needed for code understanding

6. Only Jupyter Notebooks are not allowed

**Answer 8 CV:-**

**Question 9 -**

From Question 8, you would have a custom object detection model

Note - 1. Try to reduce the model size using quantization

2. Create a web app where the users can interact with your model

3. Write modular Python script for model inference

4. Only Jupyter Notebooks are not allowed

5. Write code comments wherever needed for code understanding

**Answer 9 CV:-**