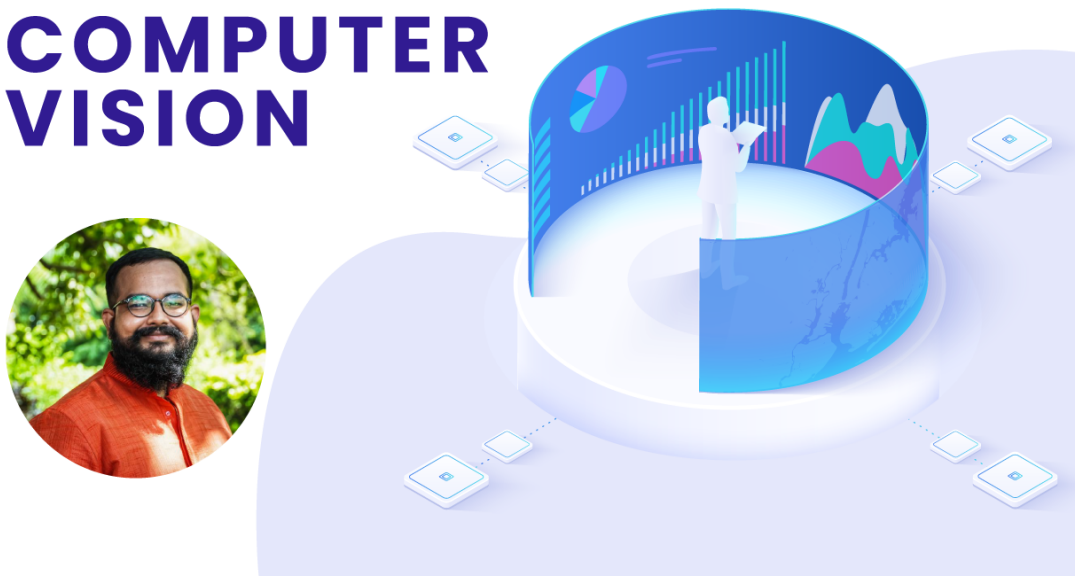
Welcome to ineuron.ai



Computer Vision

Description:

Computer Vision is a field of Artificial Intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information. If AI enables computers to think, Computer vision allows them to see, observe and understand. Computer vision is used in industries ranging from energy and utilities to manufacturing and automotive, and the market is continuing to grow.

Start Date:

Doubt Clear Time:

Course Time:

Features:

Roadmap

Challenges

Downloadable resources

Quizzes

Completion Certificate

Hands-on Practicals

What we learn:

Basic fundamentals of computer vision.

CNN architectures, classification, object detection & segmentation

Image formation, motion estimation, tracking & other fundamentals with computer visio

Various architecture usages with Computer vision for advanced-level works.

Requirements:

Python programming is needed

A system with a stable internet connection.

Your dedication.

Instructor:

Name:

Sourangshu Pal

Description:

Visual Computing Engineer and instructor at iNeuron.ai having 3 years of diverse experience in the discipline of visual computing with specialization in Deep Learning and Computer Graphics. Loves to analyze, process, and model visual data then interpret the insights to create actionable plans for solving challenging business problems.

>Introduction to Course:

>>Introduction to Course

>>Course Overview

>>Course Outcome

>>Installing Anaconda, Pycharm & Postman

>>Working with Conda Envs

>>Pycharm Introduction

>>Pycharm with Conda

>>Pycharm with venv

>>Pycharm with Pipenv

>CNN Foundations:

>>Why CNN? Building an Intution for CNN

>>CNN, Kernels, Channels, Feature Maps, Stride, Padding

>>Receptive Fields, Image Output Dimensationality Calculations, MNIST Dataset Explo

>>MNIST CNN Intutiton, Tensorspace.js, CNN Explained, CIFAR 10 Dataset Exploratio

>>Dropout & Custom Image CLassification Dog Cat Dataset

>>Deployment in Heroku,AWS,Azure

>>Deployment in GCP,AWS EBS

>CNN Architectures:

>>LeNet-5

>>LeNet-5 Practical

>>AlexNet

>>AlexNet Practical

>>VGGNet

>>VGG16 Practical

>>Inception

>>Inception Practical

>>ResNet

>>Resnet Practical

>Image Classification Hyper Parameter Tuning:

>>Keras Tuner

>>Building a simple model

>>Tuning with Keras Tuner

>Data Augmentation:

>>What is Data Augmenation?

>>Benefits of Data Augmentation

>>Exploring Papers like RICAP,Random Erasing, Cutout

>>Exploring Augmentor

>>Exploring Roboflow

>Object Detection Basics:

>>What is Object Detection?

>>Competitions for Object Detection

>>Bounding Boxes

>>Bounding Box Regression

>>Intersection over Union (IoU)

>>Precision & Recall

>>What is Average Precision?

>Object Detection Architectures:

>>Object Detection Family

>>RCNN

>>RCNN Network Architecture

>>Cons of RCNN

>>FAST RCNN

>>FAST RCNN Network Architecture

>>Cons of FAST RCNN

>>FASTER RCNN

>>FASTER RCNN Network Architecture

>>YOLO

>>YOLO Architecture

>>YOLO lImitations

>>SSD

>>SSD Network

>Practicals Object Detection using Tensorflow 1.x:

>>Introduction to TFOD1.x

>>Using Google Colab with Google Drive

>>Installation of Libraries in Colab

>>TFOD1.x Setup in Colab

>>Visiting the Model Zoo

>>Inferencing in Colab

>>Inferencing in Local

>>Important Configuration Files

>>Webcam Testing

>Practicals Training a Custom Cards Detector using Tensorflow1.x:

>>Custom Model Training in TFOD1.x

>>Our Custom Dataset

>>Doing Annotations or labeling data

>>Selection of Pretrained Model from Model Zoo

>>Files Setup for Training

>>Let's start Training in Colab

>>Export Frozen Inference Graph

>>Inferencing with our trained model in Colab

>>Training in Local

>>Inferencing with our trained model in Local

>Practicals Creating an Cards

Detector Web App with TFOD1:

>>Creating a Pycharm project & Environment Setup

>>WebApp Workflow

>>Code Understanding

>>Prediction with Postman

>>Debugging our Application

>Practicals Object Detection using Tensorflow 2.x:

>>Introduction to TFOD2.x

>>Using the Default Colab Notebook

>>Google Colab & Drive Setup

>>Visting TFOD2.x Model Garden

>>Inference using Pretrained Model

>>Inferencing in Local with a pretrained model

>Practicals Training a Custom Chess Piece Detector using Tensorflow2:

>>Custom Model training in TFOD2.x

>>Our Custom Datset

>>File Setup for Training

>>Let's start Training

>>Stop Training or resume Training

>>Evaluating the trained model

>>Convert CKPT to Saved Model

>>Inferencing using the Custom Trained Model in Colab

>>Inferencing using the Custom Trained Model in Local PC

>Practicals Creating an Chess Piece Detector Web App with TFOD2:

>>Creating a Pycharm project & Environment Setup

>>Application Workflow

>>Code understanding

>>Testing our App with PoSTmaN

>>Debugging our Application

>Practicals Object Detection using Detectron2:

>>Introduction to Detectron2

>>Detectron2 Colab Setup

>>Visiting Detectron2 Model Zoo

>>Detectron2 Pretrained Model Inferencing

>Practicals Training a Custom Detector using Detectron2:

>>Detectron2 Custom Training

>>Exploring the Dataset

>>Registering Dataset for Training

>>Let's start Training

>>Inferencing using the Custom Trained Model in Colab

>>Evaluating the Model

>Practicals Creating an Custom Detector Web App with Detectron2:

>>Creating a Pycharm project & Environment Setup

>>Application Workflow

>>Code understanding

>>Testing our App with Postman

>>Debugging our Application

>Practicals Object Detection using YoloV5:

>>Introduction to YoloV5

>>YoloV5 Colab Setup

>>Inferencing using Pre Trained Model

>Practicals Training a Custom Warehouse Apparel Detector using YoloV5:

>>Custom Training with YoloV5

>>Exploring the Dataset

>>Doing Annotations or labeling data

>>Setting up Google Colab & Drive

>>Let's start Training

>>Inferencing using the Custom Trained Model in Colab

>Practicals Creating an Warehouse Apparel Detector Web App with YOLOV5:

>>Creating a Pycharm project & Environment Setup

>>Application Workflow

>>Code understanding

>>Testing our App with Postman

>>Debugging our Application

>Image Segmentation:

>>Segmentation Introduction

>>From Bounding Box to Polygon Masks

>>What is Image Segmentation?

>>Types of Segmenation

>>MASKRCNN

>>MASK RCNN Architecture

>MASK RCNN Practicaals with TFOD:

>>Segmentation with TFOD1.x

>>Local Setup

>>Exploring the Dataset

>>Data Annotation

>>Model Selection

>>Files Setup for Training

>>Model Training

>>Export Frozen Inference Graph

>>Model Prediction

>MASKRCNN parctical with Detectron2:

>>Introduction to Detectron2

>>Detectron2 Colab Notebook

>>Exploring the Model Zoo

>>Detecron2 Colab Setup

>>Custom Training with Detectron2

>>Exploring our Dataset

>>Data Annotation

>>Data Preparation

>>Setup for Training

>>Let's start Training

>>Inferencing using the Custom Trained Model in Colab

>>Evaluating the Model

>Face Recognition Project:

>>Introduction to Project

>>Requirement Gathering

>>Techstack Selection

>>Project Installation

>>Project Demo

>>Project Workflow

>>Core Components of the Application

>>Data Collection Module

>>Generate Face Embeddings

>>Training Face Recogniton Module

>>Prediction Pipeline

>>Entrypoint of the Application

>>Application Workflow

>>Debugging Application

>>Data Collection Module

>>Generate Face Embeddings

>>Training Face Recognition Module

>>Prediction Pipeline

>>Entrypoint of the Application

>>Application Workflow

>Object Tracking Project:

>GANS: