AIML- PGCP : Capstone Project

The team-based Capstone Project completion is the objective of the PG Certification in AIML program. It will provide an opportunity for the participants to implement an end-to-end AIML project. The expectation from this Capstone Project is that the participants should be able to apply the learnings from the Advanced Certification program and demonstrate a full-fledged deployed AIML solution for the selected Problem Statement/Projects as given below.

PROJECT 1

Title: Multimodal Media Retrieval and Captioning System

Objective: Build a model capable of retrieving images and text based on cross-modal queries and generating descriptive captions for images lacking textual information.

Dataset Link: COCO Dataset, Conceptual Captions Dataset

Dataset Description:
A collection of diverse images with corresponding textual captions. The COCO dataset consists of 330k images, each annotated with five captions, capturing various objects and scenes. The Conceptual Captions dataset has millions of images with high-level, real-world context descriptions, ideal for training robust caption generation and retrieval tasks.

Project Overview:
Retrieving relevant visual and textual information based on cross-modal queries is a crucial challenge in AI, enhancing applications like content recommendation, digital archiving, and accessibility tools. This project involves creating a unified multimodal neural network that enables image-to-text and text-to-image retrieval, alongside generating accurate captions for images. The project will leverage both Computer Vision and Natural Language Processing techniques, embedding visual and textual data in a shared space for seamless multimodal interaction. The system's practical use cases include enhancing digital media archives, improving accessibility for visually impaired users, and providing content recommendations. The model will be evaluated on retrieval and captioning tasks, and the deployment will support real-time content curation applications.

Tools: Natural Language Toolkit, TensorFlow, PyTorch, Keras

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Final Submissions:

GitHub Repository of the project
Project Technical Report
Project Presentation with desired outcomes
Summary of 3 research papers

PROJECT 2

Project Title: Real world RAG System

Objective:

Develop an efficient Retrieval-Augmented Generation (RAG) pipeline capable of delivering accurate answers to the user queries.

Dataset : https://huggingface.co/datasets/rungalileo/ragbench

Dataset Description : RAG Bench is a large-scale benchmark dataset specifically designed for training and evaluating RAG systems. It contains 100k examples spanning five industry-specific domains:

- Biomedical Research
- General Knowledge
- Legal
- Customer Support
- Finance

The dataset supports various RAG task types and includes evaluation metrics such as context relevance, context utilization, answer faithfulness and answer completeness.

The dataset samples are sourced from real-world industry corpora, such as user manuals, making the dataset highly relevant for practical applications. This comprehensive dataset enables a detailed assessment of RAG system performance, ensuring robust and reliable evaluation.

Project Overview: Generative large language models often face challenges such as producing outdated information or fabricating facts. Retrieval-Augmented Generation (RAG) techniques address these limitations by integrating pretraining and retrieval-based approaches. This combination creates a robust framework for improving model accuracy and reliability.

RAG offers the added benefit of enabling rapid deployment of domain-specific applications without requiring updates to the model parameters. As long as relevant documents are available for a query, the system can adapt to organizational or domain-specific needs efficiently.

A typical RAG workflow involves Query Classification, Retrieval, Reranking, Repacking and Summarization. Implementing RAG requires careful consideration of various factors, such as properly splitting documents into manageable chunks, selecting suitable embeddings to semantically represent these chunks, choosing vector databases for efficient storage and retrieval of feature representations.

By addressing these elements, RAG systems can deliver accurate, domain-specific, and context-aware responses, making them a powerful tool for real-world applications.

References:

1) Searching for Best Practices in Retrieval-Augmented Generation,

2) RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems

Tools: Transformers, Hugging Face, PyTorch, Langchain, LlamaIndex

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Final Submissions:

- Project technical report & presentation with desired outcomes
- An overview of the modeling techniques used for the problem
- GitHub Repository of the project
- Summary of 3 research papers

PROJECT 3

Title: Image tagging and road object detection

Objective: Detect object tagging in the video and examine how parallel object detection on multiple patches can allow the detection of smaller objects in the overall image without decreasing the resolution.

Dataset Link: BDD 100K Dataset.

Dataset description: The Berkeley Deep Drive (BDD) dataset is one of the largest and most diverse video datasets for autonomous vehicles.

- The dataset contains 100,000 video clips collected from more than 50,000 rides covering New York, San Francisco Bay Area, and other regions.
- The dataset contains diverse scene types such as city streets, residential areas, and highways.
- Furthermore, the videos were recorded in diverse weather conditions at different times of the day.

Project Overview: Object detection and segmentation methods are one of the most challenging problems in computer vision which aim to identify all target objects and determine the categories and position information. Numerous approaches have been proposed to solve this problem, mainly inspired by methods of computer vision and deep learning. In this project, we aim to build a model which detects multiple objects and segmentation in a moving video. For eg. Image tagging, lane detection, drivable area segmentation, road object detection, semantic segmentation, instance segmentation, multi-object detection tracking, multi-object segmentation tracking, domain adaptation, and imitation learning.

Tools: TensorFlow, PyTorch, Keras

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Final Submissions:

- GitHub Repository of the project
- Project Technical Report
- Project Presentation with desired outcomes
- Summary of 3 research papers

PROJECT 4

Title : Automatic Speech Recognition(ASR)

Objective: Build an ASR model for converting speech to text.

Dataset Link : LibriSpeech

Dataset description: LibriSpeech is a corpus of reading English speech, suitable for training and evaluating speech recognition systems, published in 2015 by Johns Hopkins University. It is derived from audiobooks that are part of the LibriVox project and contains 1000 hours of speech sampled at 16 kHz of 2000 speakers. The LibriVox project1, a volunteer effort, is responsible for the creation of approximately 8000 public domain audiobooks, the majority of which are in English. Most of the recordings are based on texts from Project Gutenberg2, also in the public domain. The data is already divided into train/dev/test sets. The total size of the data is 60 GB and subsets are available of different sizes.

Initially, we recommend working only with 'dev-clean' and 'test-clean' datasets for building the model. We can use any one or a combination of both data sets as a training set. A subset of either 'dev-clean' or 'test-clean' can be used for testing purposes. Once modeling is done with these smaller data sets, start modeling using 'train-clean'/'train-other' data sets of larger sizes as a training set. Now, 'dev-clean', 'test-clean', and 'test-other' datasets are used for validation/testing purposes only.

Project Overview: Automatic speech recognition is the application of Machine learning or AI where human speech is processed and converted into readable text. We can find numerous applications such as Instagram for real-time captions, Spotify for podcast transcriptions, Youtube video transcription, Zoom meeting transcriptions, etc. The field has grown exponentially over the last few years. An explosion of applications taking advantage of ASR technology in their products to make audio and video data more accessible.

There are different approaches to Automatic Speech Recognition, viz. traditional HMM (Hidden Markov Models), GMM (Gaussian Mixture Models), end-to-end deep learning, and Self-supervised models. In this project, we aim to build and deploy a model that can generate the written text from the speech with a decent accuracy. The main aim is to build and deploy a self-supervised model such as wav2vec, a state-of-the-art ASR model that is built on Transformer architecture and uses the concept of contrastive learning.

Tools: PyTorch, Audio Processing tool/library, fairseq.

Operating system: Linux - Ubuntu

Deployments: FastAPI, Cloud Application Platform | Heroku, Streamlit, Cloud Computing, Hosting Services, and APIs | Google Cloud

Reference: Papers using libriSpeech

Final Submissions:

- GitHub Repository of the project
- Project Technical Report

- Project Presentation with desired outcomes
- Summary of 3 research papers

PROJECT 5

Title: Automatic Number Plate Recognition (ANPR)

Objective: Build a CV model for recognizing the Number Plate and Displaying the Number.

Dataset Link: Image Dataset

Dataset description: The dataset consists of 433 images with bounding box annotations of the car license plates within the image. Annotations are provided in the PASCAL VOC format i.e.; images are accompanied with an XML file containing the object annotations.

Project Overview:

AI and deep learning are being used everywhere, from voice assistants to self-driving cars. One such application is the Automatic Number Plate Recognition (ANPR) of Vehicles. ANPR is a technology that uses the power of AI and deep learning to automatically detect and recognize the characters of a vehicle's license plate.

With the increase in the number of vehicles, vehicle tracking has become an important research area for efficient traffic control, surveillance, and finding stolen cars. The specific use cases may be traffic violation control, parking management, toll booth payments, etc. For this purpose, efficient real-time license plate detection and recognition are of great importance.

Challenges: Due to the variation in the background and font color, font style, size of the license plate, and non-standard character along with the issue of robustness in varying weather conditions, license plate recognition is a great challenge, especially in developing countries. The given dataset is for building a baseline model. You are expected to include an additional 100 image data sets from Indian conditions and try to overcome these challenges by applying different techniques. You may have to create a bounding box for the number plate region in the additional images. Overall you have to complete two tasks. **Task 1**: Data collection and creation of bounding box, **Task 2**: ANPR

Automatic Number Plate Recognition (ANPR) implementation involves following three steps:

Step 1: Detect and localize a license plate in an input image/frame
Step 2: Extract the characters from the license plate
Step 3: Apply some form of Optical Character Recognition (OCR) to recognize the extracted character

Tools: TensorFlow, PyTorch, Keras, OpenCV, OCR-Tool, Yolo
Deployments: FastAPI, Streamlit, Gradio | Cloud computing & Hosting Services Platform : Heroku, AWS, Google Cloud etc.

Final Submissions:
- GitHub Repository of the project

- Project Technical Report
- Project Presentation with desired outcomes
- Summary of 3 research papers

Few points to note while working on the capstone project:

- Though this is a group project, faculties/mentors can access individual performance/contributions and may award different marks to each individual of the same group.

- Attendance is the minimum criterion for getting marks during the mentoring sessions.

- **Project Proposal Format:** It consists of a document not more than 7/8-pages, covering the title, problem statement, literature review, methodologies, possible outcomes in stages, and applicability in the real world. First page should contain the title and group number & team members name.

- **Final Report/PPT Format**: It consists of a Word Doc. & PPT, covering the title, problem statement, literature review, methodologies, outcomes in stages, final outcomes, challenges, and applicability in the real world. The first page should contain the title, group number & team members names.

- Demonstration of the deployed model is a must along with the final presentation and report.

- **There will be four checkpoints with the following expectations**:

    - **Check Point-1**: Preliminary Model & Training (on subset of data)
    - **Check Point-2**: Model building, training for complete dataset
    - **Check Point-3**: Deployment testing (on subset of data)
    - **Check Point-4**: Fine tuning & Final deployment