# Iteration 2

This project aims to develop a human action recognition system that can accurately predict the action performed in each video sequence. The system is built using a combination of computer vision and deep learning techniques, including object detection using Detectron2 and action recognition using an LSTM model. The system is integrated into a web application using the Flask web framework, providing a user-friendly interface for users to upload their videos and view the predicted action label.

Human action recognition is an important task in computer vision that has applications in various fields such as surveillance, robotics, and sports analysis. With the increasing availability of video data, there is a need for automated systems that can accurately recognize human actions in videos. This project aims to build such a system using state-of-the-art computer vision and deep learning techniques.

**Methodology:**

The proposed system consists of three main components: object detection using Detectron2, feature extraction using an LSTM model, and web application using Flask. The system architecture is shown in Figure 1.

**Object Detection using Detectron2:**

The first component of the system is object detection using Detectron2, a popular open-source object detection framework. The framework is used to detect humans in video frames and extract their bounding boxes. These bounding boxes are then used to extract features that are used by the LSTM model for action recognition.

**Feature Extraction using LSTM Model:**

The second component of the system is featuring extraction using an LSTM model. The model is trained on a large dataset of human action videos and can extract temporal features that capture the motion patterns of different actions. The LSTM model takes as input the features extracted from the human bounding boxes detected in the video frames and outputs the predicted action label.

**Web Application using Flask:**

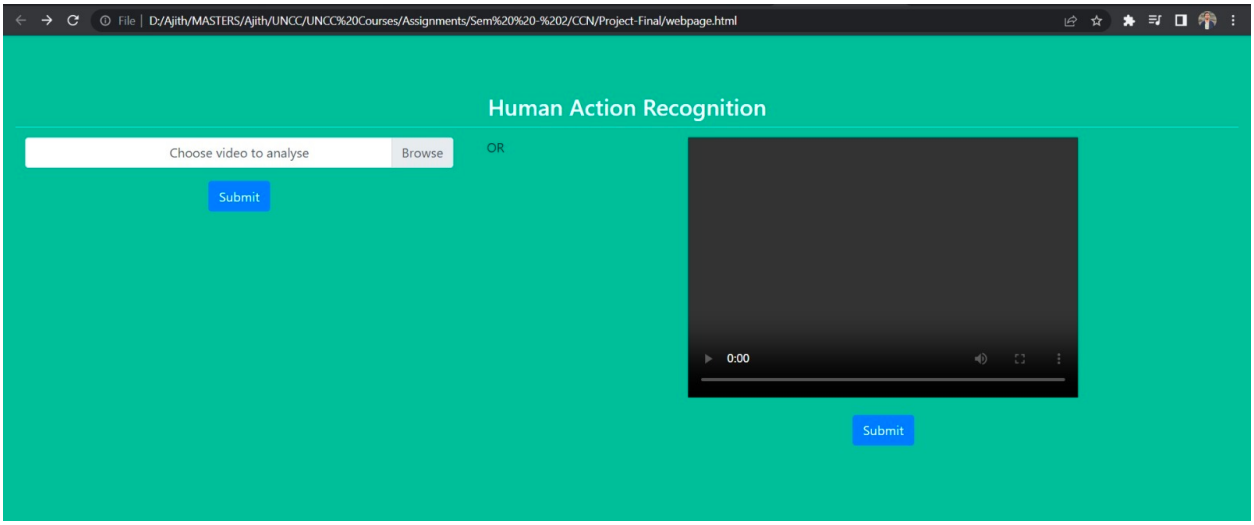
The third component of the system is the web application that is developed using the Flask web framework. The web application provides a user-friendly interface for users to upload their videos and view the predicted action label. The user interface allows users to select a video file from their local system and upload it to the server. Once the video is uploaded, the system processes

the video frames using the object detection and LSTM models to predict the action label for the video. The predicted action label is then displayed to the user on the web interface.

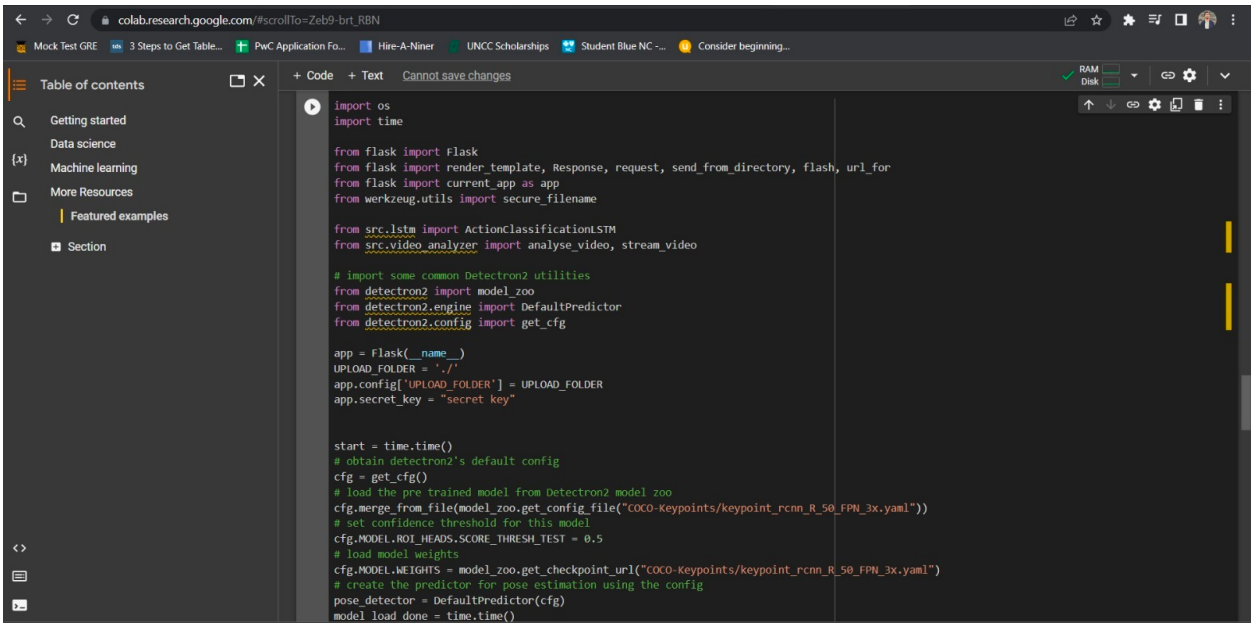
**Versions:**

- **Flask 1.1.2:** Flask is a Python web framework that provides tools for building web applications. It is lightweight and flexible, making it a popular choice for web development.
- **Flask-Bootstrap 3.3.7.1:** Flask-Bootstrap is an extension for Flask that provides integration with Bootstrap, a popular front-end framework for web development. It makes it easy to add pre-built user interface elements to Flask applications.
- **Flask-Uploads 0.2.1:** Flask-Uploads is an extension for Flask that provides easy file uploads for web applications. It includes features such as file type validation and file size limits.
- **Numpy 1.19.5:** NumPy is a Python library for numerical computing that provides tools for working with arrays, matrices, and mathematical functions. It is commonly used in scientific computing and data analysis.
- **opencv-python 4.1.2.30:** OpenCV is an open-source computer vision library that provides tools for image and video processing. The Python version is called opencv-python and is widely used in computer vision applications.
- pytorch_lightning 1.3.3: PyTorch Lightning is a lightweight PyTorch wrapper that provides a high-level interface for training deep learning models. It includes features such as automatic checkpointing and distributed training.
- **Torch 1.8.1:** PyTorch is an open-source machine learning library that provides tools for building and training deep learning models. It is widely used in research and industry for a variety of machine learning tasks.
- **Torchvision 0.9.1:** Torchvision is a PyTorch library that provides tools for working with computer vision datasets and models. It includes pre-trained models and datasets for common computer vision tasks.
- **Pyyaml 5.1:** PyYAML is a Python library that provides tools for working with YAML files. YAML is a human-readable data serialization format that is commonly used for configuration files.
- **Werkzeug 0.15.6:** Werkzeug is a Python library that provides tools for building web applications. It is used as the underlying WSGI library for Flask applications.
- **torchtext:** Torchtext is a PyTorch library that provides tools for working with natural language processing (NLP) datasets and models. It includes pre-trained models and datasets for common NLP tasks.

Front-end snapshot of the web application:




We have used Google Colab platform to code and deploy the web application.




The code defines a Flask web application that allows users to upload videos and analyze them for human actions using a combination of Detectron2 pose estimation and an LSTM action classification model.

When the user uploads a video file, the server checks if the file is of the correct file type and saves it to a designated upload folder. The uploaded video can then be displayed on the web page and analyzed by clicking on the "analyze" button.

The analysis is performed by first using the Detectron2 model to estimate human poses from each frame of the video. Then, the estimated poses are fed into the LSTM model to classify the action being performed in the video. The analysis results are saved to a file and can be downloaded by the user.

Additionally, the application provides a "sample" button to display a pre-uploaded sample video for analysis.