Computer Market Hub - AI Intern Assignment

Problem Statement:

Your task is to develop an application that automates the counting of sheet stacks in a manufacturing plant. Currently, this task is performed manually, which is time-consuming and prone to errors. The goal is to create a solution that can accurately count the number of sheets from images of stacks, like the ones provided

Overall Approach

The objective of this project is to develop a Streamlit application for detecting, tracking, and counting sheets in images, videos, and webcam feeds. The application uses a pre-trained YOLO model to detect sheets and a custom tracker to track and count the sheets across frames. The core functionality is implemented in Python, leveraging various libraries for computer vision, data processing, and web application development.

Frameworks/Libraries/Tools

- **OpenCV**: For image and video processing.

- **NumPy**: For numerical operations and array manipulations.

- **Pandas**: For handling and processing detection results.

- **cvzone**: For additional computer vision utilities.

- **Streamlit**: For building the web application interface.

- **ultralytics**: For loading and running the YOLO model.

- **Tracker**: Custom tracking module for tracking detected sheets.

- **tempfile**: For handling temporary files in video processing.

Challenges and Solutions

1. **Data Annotation**: Use tool like labelImg and can use roboflow for data annotation.

2. **Model Integration**: Integrating the YOLO model with the Streamlit app required careful handling of the model's input and output data formats. The solution involved converting model predictions to a suitable format for further processing and visualization.

3. **Real-time Processing**: Processing video streams and webcam feeds in real-time posed performance challenges. Efficient frame processing and using optimized functions from OpenCV and NumPy helped mitigate these issues.

4. **Tracking Accuracy**: Ensuring accurate tracking of sheets across frames was challenging due to potential overlaps and occlusions. Implementing a custom tracker and fine-tuning the tracking parameters improved tracking accuracy.

Future Scope

- **Model Optimization**: Optimize the YOLO model and tracking algorithm for better performance and accuracy.

- **Multi-class Detection**: Extend the application to detect and count multiple classes of objects simultaneously.

- **User Interface Enhancements**: Improve the Streamlit interface with more interactive features and real-time analytics.

- **Deployment**: Deploy the application as a cloud service to make it accessible to a wider audience.

- **Integration with Other Tools**: Integrate with other data processing and visualization tools for enhanced reporting and analysis.

for computer vision, data processing, and web application development.

##Challenges and Solutions

1. **Model Integration: Integrating the YOLO model with the Streamlit app required careful handling of the model's input and output data formats. The solution involved converting model predictions to a suitable format for further processing and visualization.**

2. **Real-time Processing: Processing video streams and webcam feeds in real-time posed performance challenges. Efficient frame processing and using optimized functions from OpenCV and NumPy helped mitigate these issues.**

3. **Tracking Accuracy: Ensuring accurate tracking of sheets across frames was challenging due to potential overlaps and occlusions. Implementing a custom tracker and fine-tuning the tracking parameters improved tracking accuracy.**