IE643: Deep Learning - Theory and Practice **July-Nov 2024**

Project Report: Detecting and counting unique people in a movie clip

Team Name: Pixel Pioneers *Team Members: 22b0913, 22b0972*

Abstract

This project report details our work on **detecting, counting, and tracking unique people or characters in a movie clip**. We describe the deep learning tools utilized, including the dataset selection, pre-trained models, various approaches to accomplish the task, and the quantitative performance metrics used to evaluate model efficacy. Additionally, we discuss the challenges encountered during the project.

For person detection, we used pre-trained **YOLO** and **Detectron2** models with **fine-tuning**. To assign and track unique IDs to individuals within the video and count the number of unique people, we implemented multiple tracking algorithms, including **DeepSort** and **BotSort**. Among these, DeepSort provided the most accurate tracking results, outperforming BotSort. We found that YOLO models outperformed Detectron2 models in terms of detection accuracy.

1 Introduction

Our project is to detect, count and track unique people/characters withing a given movie clip. This is an important task in real life world examples like

- Security and Surveillance - Detecting and tracking people in video footage to ensure safety and monitor spaces.
- Entertainment field - Tracking people in films or sports videos can enhance editing, analysis, and post-production work
- Retail Field - Understand customer behavior in stores by analyzing foot traffic and engagement.

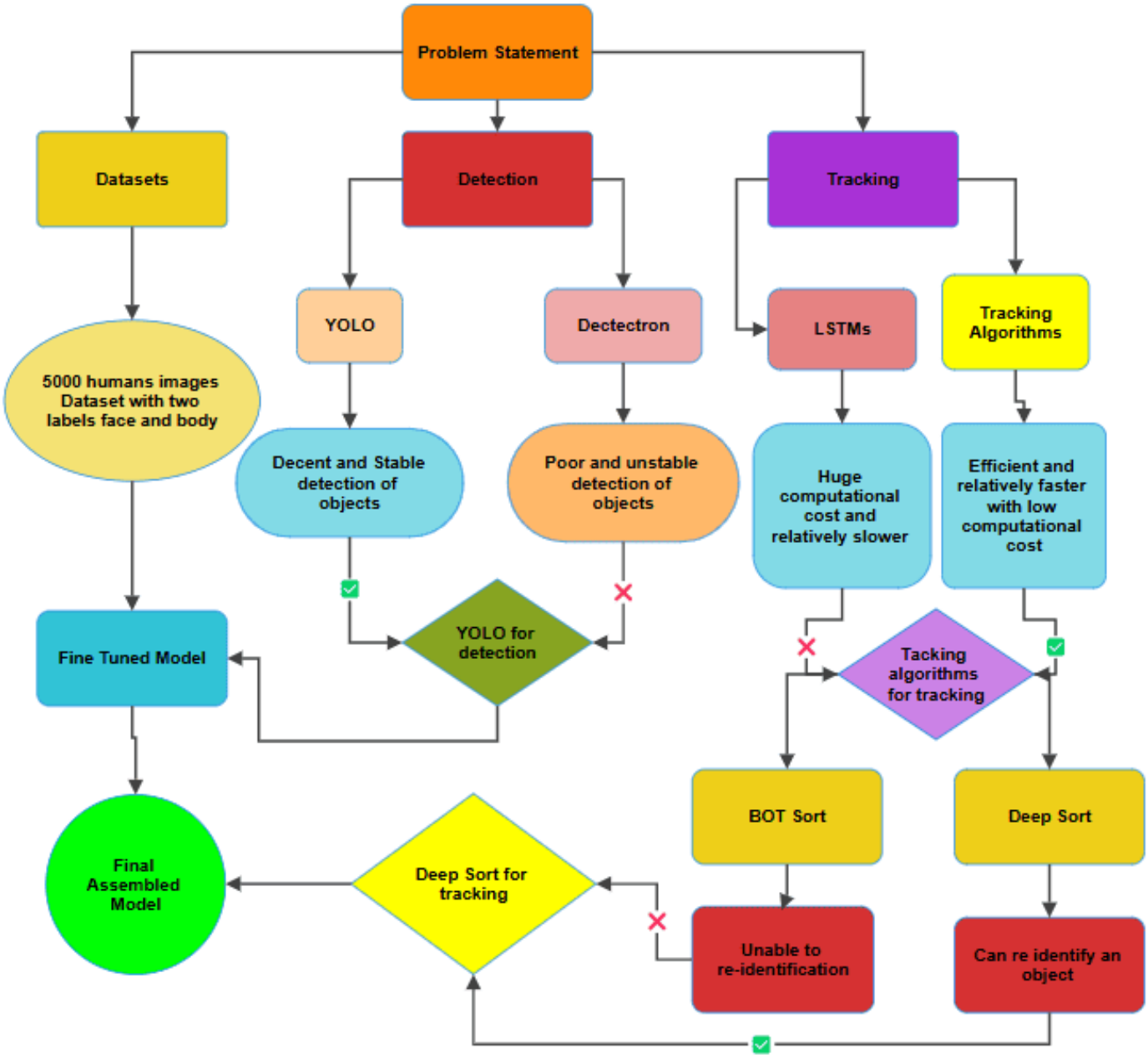
Since it has many applications in real life world problems it is very important to do this accurately.

We can use Deep learning Techniques to perform this accurately. Since the models can learn complex patterns, flexible and adaptable to various conditions. Initially we need to find or create the correct dataset based on our task then we need to detect people using some detection based models and perform fine-tuning so that the dection models can detect the objects as per requirement accurately. Then track the detected objects using tracking algorithms like DeepSort and BotSort or we can also use Long Short-Term Memory RNNs to perform the tracking task and count the unique persons or characters in the video.

We provide our Project Workflow in Section 2. We provide a survey of existing literature in Section 3. Our proposal for the project is described in Section 4. We give details of the datasets used in our project in Section 5. Details on experiments in Section 6. Detailed results achieved in each experiment and quantitative metrics in Section 7. A description of future work is given in Section 8. We conclude with a short summary and pointers to forthcoming work in Section 9.

2 Project Workflow

Flowchart of the Workflow:



Flow Chart

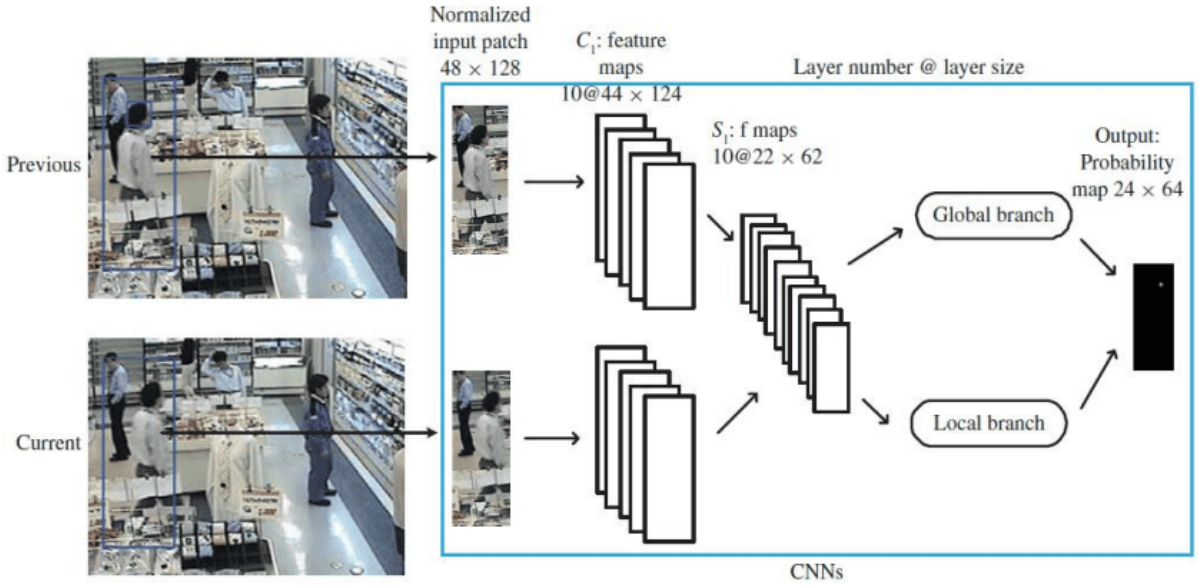
Description of the Workflow:

- First, we identified and obtained the dataset required for fine-tuning our model, which consists of 5,822 images across both training and test sets.
- This is a labeled dataset with two classes for face and body detection.
- We then evaluated object detection models, specifically experimenting with `YOLO` and `Detectron`.
- Through our testing, we found that `Detectron`'s performance was unstable and less accurate compared to `YOLO`.
- Consequently, we decided to proceed with `YOLO` for the detection component.
- For the tracking phase, we explored two approaches: employing LSTMs or using established tracking algorithms.
- Our research into LSTMs indicated that they are computationally inefficient and require a long training time.
- As a result, we opted for tracking algorithms to handle object tracking.
- We examined two tracking methods: BOT-SORT and Deep SORT.
- While BOT-SORT provided satisfactory tracking accuracy, it lacks support for re-identification (Re-ID), meaning it cannot reliably track an individual who temporarily exits and re-enters the video frame.
- Deep SORT, however, includes Re-ID functionality, enabling it to manage cases where an individual reappears after disappearing from view.
- Therefore, we chose Deep SORT for the object tracking component.
- We fine-tuned a `YOLOv11` object detection model on our dataset, resulting in significant improvements in detection accuracy.
- Finally, we integrated the fine-tuned `YOLOv11` model with Deep SORT, forming our complete model.

3 Literature Survey

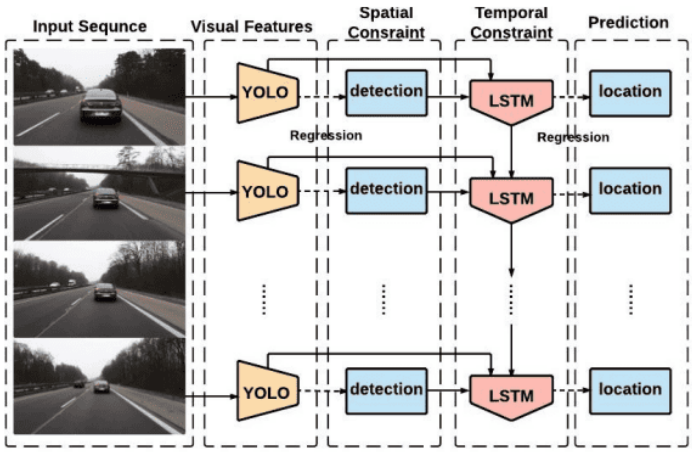
Our main idea on how to track objects in a video came from the work [1] done by Chen, Minghai and Ding, Guiguang and Zhao, Sicheng and Chen, Hui and Han, Jungong and Liu, Qiang in Reference Based LSTM for Image Captioning. Where authors uses model that combines YOLO [8] and LSTM. YOLO model's detailed information about detected objects, including class probabilities are used as input for LSTM. Authors say that using LSTM allows us generate more accurate captions. From this paper the key take away is that we can integrate YOLO [8] with LSTM for tracking proposes.

From the idea of the previous paper we researched papers where YOLO [8] and LSTM are integrated. Work done by the author Poormehdi Ghaemmoghami, Masoumeh in the book [7] shows that we can integreate LSTM and YOLO [8]. The main idea behind this is the outputs of the YOLO [8] detection models are passed as inputs to the LSTM where we can keep track of the objects by observing the bounding boxes. This method uses a shift-variant CNN architecture to track objects by learning spatial and tem-



Shift Variant CNN

poral features from pairs of adjacent frames, helping prevent tracking drift in cluttered environments. By using an object's prior location and appearance, it effectively transforms an object detector into a tracker that estimates an object's current position and scale. And for tracking we pass the frames from YOLO dectection to the LSTM as shown below,

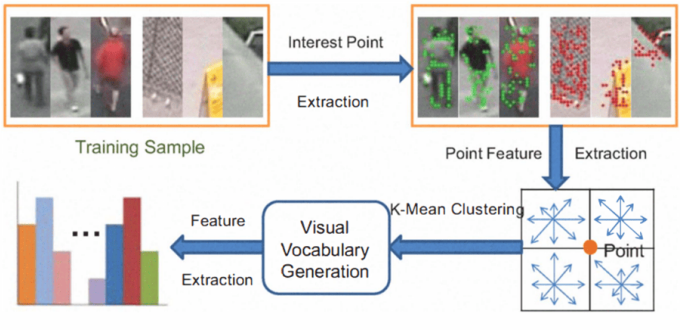


Track using YOLO and LSTM

We also took inspiration from work done by M. Everingham and A. Zisserman [2] in Identifying individuals in video by combining 'generative' and discriminative head models, Where the authors work on a TV series clip and find the individuals in using using Deep Learning Techniques. They generated 3D face models by using multiple images of a person and trained a tree based model. This we can't use directly in our project since we don't know the persons/characters beforehand. But one key take away from this [2] is that we can detect persons if we somehow detect faces of the persons/characters.

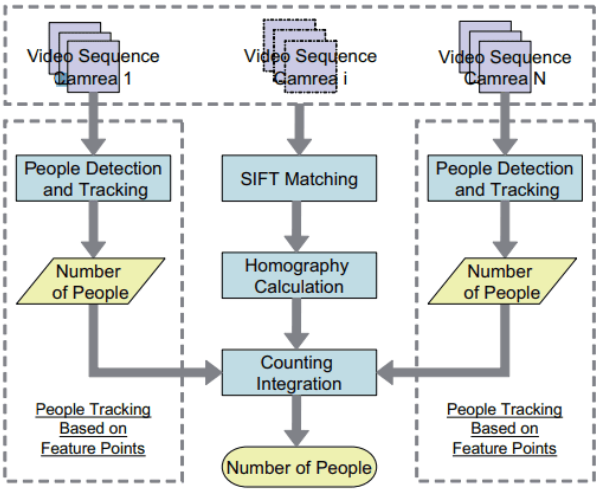
Detect faces using 3D face models

The papers [4] and [5] uses different method from YOLO and LSTM to count people in videos. For example in the work [4] by Li Jingwen and Huang Lei and Liu Changping in Tracking of Humans in Video Stream Using LSTM Recurrent Neural Network they used a unique method to achieve the goal of the counting people in a video. From the above picture we can clearly see that from each frame we extract



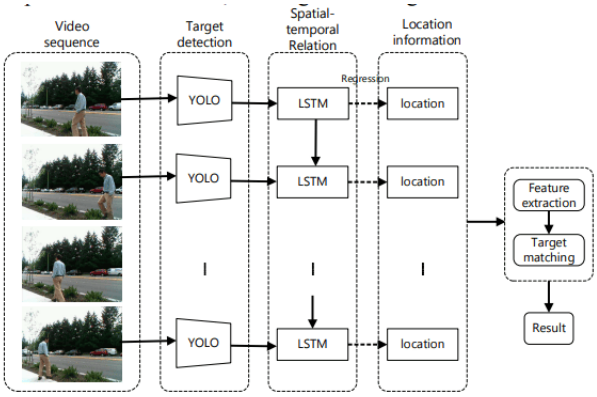
LSTM and RNN

interest points and then use K-means clustering which later does feature extraction, and that data is sent to LSTM and count number of people using LSTM RNN.
The work [6] by Jingwen Li, Lei Huang, Changping Liu in the People Counting across Multiple Cameras for Intelligent Video Surveillance depicts a way how to re-identify a object in video. Where the authors presents a people counting system that utilizes multiple cameras for video surveillance which focuses on developing algorithms to accurately track and count people across different camera views and addresses challenges like occlusions, varying camera angles, and people moving in and out of camera view. Which helps in dealing potential applications include crowd management and security monitoring



Flowchart of the proposed multi-camera people counting system.

Our project draws main inspiration from the work[9] done by Li Tan, Xu Dong, Yuxi Ma, Chongchong Yu in A Multiple Object Tracking Algorithm Based on YOLO Detection which utilizes the YOLO object detection model for real-time multi-object detection. This also uses a Kalman filter-based tracking algorithm to estimate the trajectory of each object. Which helps us in real-time multi-object tracking by combining YOLO detection and tracking components demonstrates promising performance in terms of accuracy and computational efficiency. From this paper we got to know that using a tracking algorithm is much more



Multi-target tracking algorithm framework

efficient than that of developing an LSTM from scratch and training it on large dataset.

4 Proposed Approach or Approaches

4.1 Human Detection

The first step in our project is to detect humans in the video effectively. For this we used pre-trained models like YOLO, detectron and compared the performances of these two in Section: 7 and found that Detectron is not good as YOLO, hence we used YOLO as the final human Detection pre-trained model. Recently on 30th September `YOLOv11` model was released which is better than previous versions of YOLO, we can clearly see the mAp score on COCO Dataset is highest for YOLOv11 model from the Fig:2 and we used this 11th version of YOLO in our project. Since YOLO can detect upto 80 classes the naive model which was published might not good enough to detect humans in a image. Hence we need to use a dataset that contain human images and labels Fig:1. More about this dataset in Section 5. More about the internal operations in the fine tuning in Section 6.

Without Fine tuning

With Fine Tuning

Figure 1: Comparison between fine tuned and normal detection

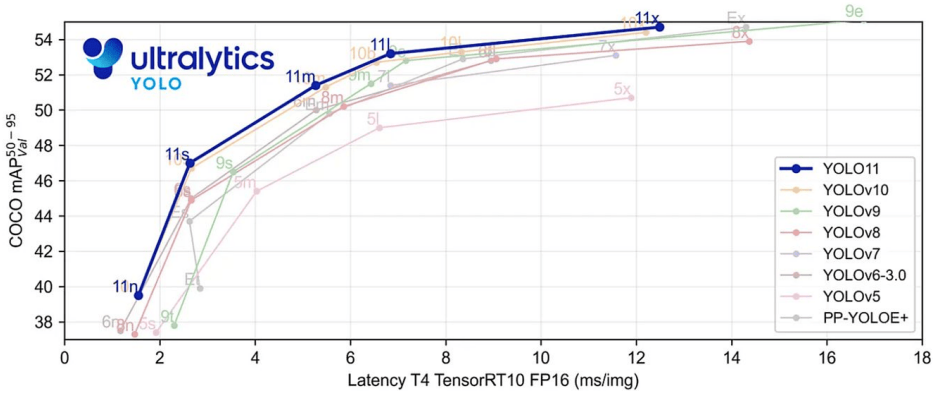


Figure 2: YOLO models comparision

4.2 Human Tracking

Next step in our project is to track the detected humans in a video. For this we initially though of using LSTM integrated with YOLO. Where the basic idea is to pass the YOLO model's output of the detected objects as the input to LSTM. But we realized that doing this requires more computational cost for both training and testing and it is a slower process. Hence we used tracking algorithms to track objects in the

video we tested on two algorithms BOT SORT and Deep SORT. Using BOT SORT we couldn't assign re-ID that is a person entering the video and left in between and reappearing again. But we achieved this using Deep SORT. Few results of the tracking are shown below,9 Few points from the inference:

- From all the frames we can clearly see that main person who is walking with headphones has ID=1 throughout all the frames from $t = 1$ to $t = 9$.
- We can also see that boxes are stable through out the whole video.
- There is another person whose ID was given 3 was detected lately because of he is almost un-notified because of his size.
- Once given an ID the ID for that person is maintained through out the video.
- Therefore using the Deep Sort integrated with YOLO tracks the persons effectively.
- Comparison between the Deep Sort and BOT Sort can be seen in Section:7

t=1 t=2 t=3

t=4 t=5 t=6

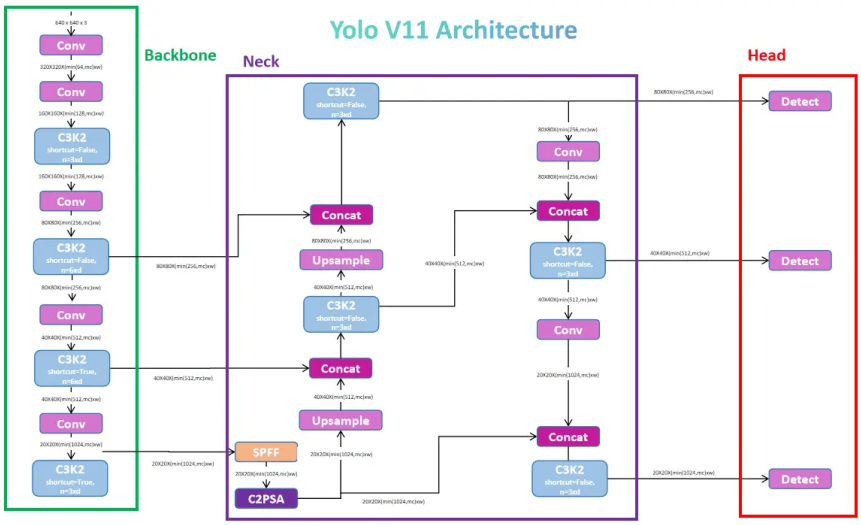
t=7 t=8 t=9

Figure 3: Tracking of a person in a video

4.3 YOLOv11 Architecture

As mentioned in the work [3] YOLOV11: AN OVERVIEW OF THE KEY ARCHITECTURAL ENHANCEMENTS done by Rahima Khanam and Muhammad Hussain yolov11 Architecture is as follows:

- It mainly contains three fundamental parts the **backbone**, the **neck** and the **head**.
- The **backbone** acts as the primary feature extractor, utilizing convolutional neural networks to transform raw image data into multi-scale feature maps.
  - This YOLO11 uses C3k2 block instead of C2f Block which is computationally more efficient implementing Cross Stage Partial Bottleneck which uses two smaller convolutions instead of one larger
- The **neck** component acts as an intermediate processing stage, employing specialized layers to aggregate and enhance feature representations across different scales.
- The **head** is designed to adaptively adjust to image complexity, utilizing a multi-scale detection strategy. It can handle objects at varying scales by analyzing features from different layers, allowing the model to detect both large and small objects accurately. This dynamic aspect also reduces processing time, making YOLOv11 highly efficient for real-time applications.
- The **Cross Stage Partial** with Spatial Attention (**C2PSA**) block is a new feature in YOLOv11 that applies attention to focus on significant regions within an image. This block improves detection of partially occluded or small objects by emphasizing spatial relevance in feature maps, thus allowing for more accurate and context-sensitive object detection
- Key characteristics of the C3k2 block are **Faster processing** since it uses two smaller convolutions reduces the computational overhead and **Parameter efficiency**: C3k2 is a more compact version of the CSP bottleneck, making the architecture more efficient in terms of the number of trainable parameters.
- **CBS Blocks**: Head includes several CBS, these further refine the feature maps by Extracting relevant features for accurate object detection, Stabilizing and normalizing the data flow through batch normalization and Utilizing the Sigmoid Linear Unit (SiLU) activation function for non-linearity, which improves model performance.



YOLOv11 Architecture

Therefore, Dynamic Head and Transformer Backbone of the yolov11 model which uses C3k2 blocks makes it faster without compromising accuracy. The Cross Stage Partial with Spatial Attention (C2PSA) block

improves YOLOv11's precision by focusing on important image regions leading us to good accuracies. Hence these two makes yolov11 to be highly efficient for image detection with low computational cost. Hence we used this in our project.

4.4 Work done before prep-presentation review

Before prep-presentation we completed the following tasks.

- Compared the detection performance of Detectron and YOLO. See Section 6 for details.
- Selected YOLOv11 as the final model for detection tasks.
- Unable to track unique IDs for individual persons in video sequences.
- Considered integrating YOLO with LSTM for object tracking purposes.
- Defined quantitative metrics for evaluation.
- Utilized a basic dataset without occlusions or complex poses.
- Planned to create an interface using Flask, Python, HTML, and JavaScript.

4.5 Work done after prep-presentation review

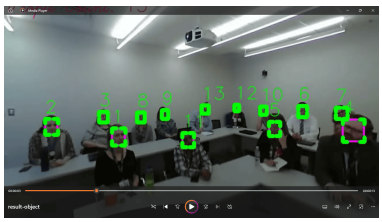
After the prep-presentation review we completed the following tasks:

- Achieved stable bounding box detection for persons in videos.
- Successfully assigned and tracked unique IDs for individuals in video sequences.
- Updated measurement metrics to prevent negative values.
- Decided to use `gradio` instead of Flask, Python, HTML, and JavaScript for the interface.
- Fine-tuned the initial YOLOv11 model using a more complex dataset with 5,822 images for both training and testing.
- Implemented tracking algorithms such as DEEP SORT and BOT SORT, replacing LSTM for improved efficiency and accuracy.
- Developed the interface and integrated the fine-tuned YOLOv11 model with DEEP SORT to complete the final pipeline.

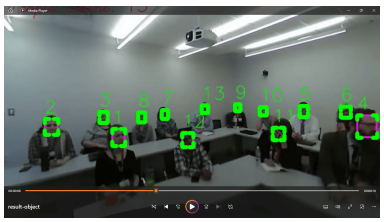
4.6 Significance of the work done after the prep-presentation review:

4.6.1 Stable detection of IDs:

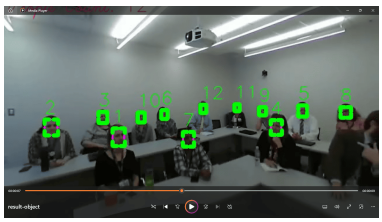
We couldn't detect the stable ID Detection and maintain same ID for a person through out the video, before the pre-presentation:



Frame 1

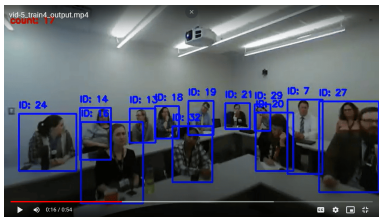


Frame 2

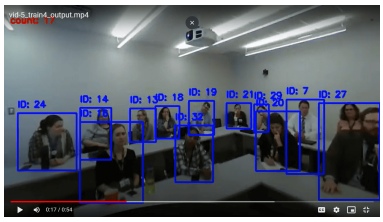


Frame 3

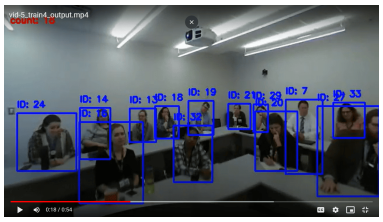
But after integrating our detection model with tracking algorithms we can are able to do that



Frame 1



Frame 2



Frame 3

4.6.2 Updated measuring metrics:

Before Prep-presentaion our measuring metrics:

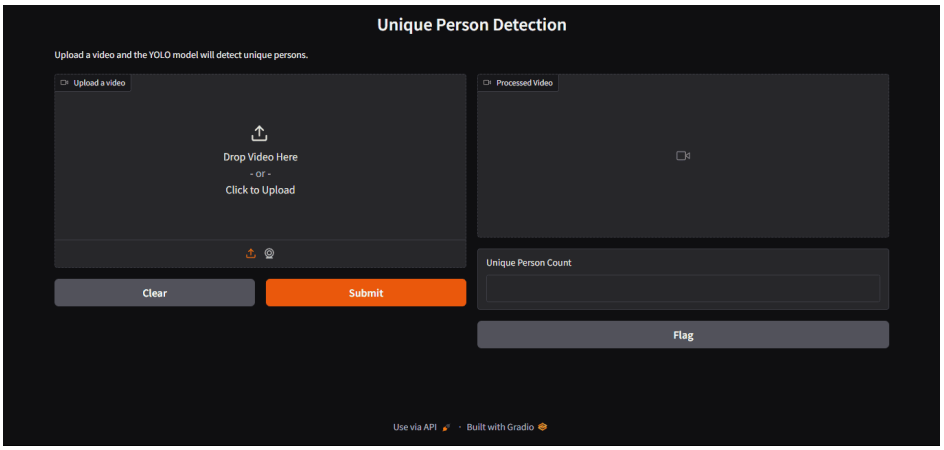
$$Accuracy = 100 - 100 * \frac{\text{abs(No. of detected persons in a video - real people count in that video)}}{\text{real people count in that video}}$$

This can lead to negative accuracies when abs(No. of detected persons in a video - real people count in that video) > real people count in that video hence we updated measuring metrics to avoid negative accuracies to:

$$Accuracy = max((100 - 100 * \frac{\text{abs(No. of detected persons in a video - real people count in that video)}}{\text{real people count in that video}}), 0)$$

4.6.3 Final Interface:

Our final interface looks like:



Final Interface

4.6.4 RE-ID Implementation

We integrated DEEP SORT with YOLO for re-identification of a person in a video that is a person appears in the video in the beginning of the video and later disappears and then reappears. We handled this case the results can be seen in the figure 11
We can clearly see that the person's ID remained same before and after the video i.e, ID = 3.

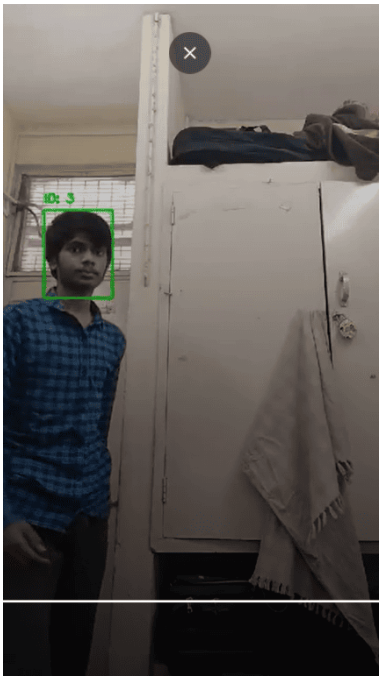
Frame 1

Frame 2

Frame 3

Frame 4

Frame 5



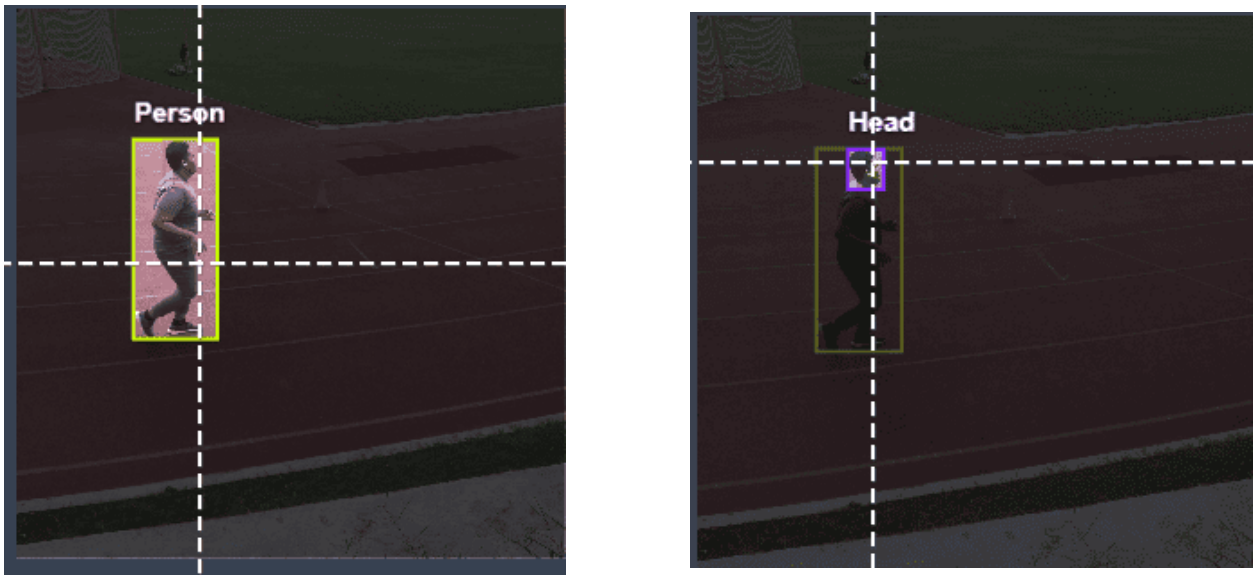
Frame 6

Figure 4: Re Identification of a person

5 Data set Details

The Dataset we used in this project are:

- Since we used YOLO pre-trained model which was trained on the COCO Dataset.
  - It is also called Common Object in Context Dataset,
  - It contains 123,387 images with 80 object categories and 886,284 object instances
  - Has bounding boxes annotations for object detection, segmented masks for image segmentation tasks
  - It is an image Dataset
  - It was obtained in the website Click here to visit
- Dataset used for fine tuning:
  - It is an image dataset.
  - Contains 5,822 images across both training and test sets.
  - It was obtained in this website Click here to visit
  - It is a labeled dataset with two classes one for Head and other for person.
  - Has bounding boxes annotations for object detection
  - Sample Image:



Person Label

Head Label

Figure 5: Labeling of a Person and Head for an Image in this Dataset

6 Experiments

6.1 YOLO

Data/Training :

- Yolo uses coco dataset(it is a labeled data set) for training.
- Used a custom data set which contains humans for for updating the weights and fine-tuning the model above the coco dataset.

Algorithm :

- YOLO utilizes one CNN to predict bounding boxes and class probabilities directly from images, enhancing speed.
- The image is divided into an S×S grid, with each cell responsible for detecting objects whose centers fall within it.
- Each grid cell predicts fixed bounding boxes and confidence scores, indicating the likelihood of containing an object.
- Grid cells predict class probabilities for each bounding box, combining these with confidence scores for overall object detection.
- NMS removes duplicate boxes by keeping only the highest-scoring boxes for each detected object.

We further fine-tuned this model to especially detect persons.

Fine-tuning/Optimization :

- We have kept weights constant in all the layers expect the last layer.
- We updated the weights of only the last layer while training using the custom dataset such that loss is minimized.

Hardware configuration :

- **GPU :**T4 GPU used in google collab.
- **RAM :** 16 GB

6.2 Detectron2

Data/Training :

- This model also uses COCO dataset(it is a labelled data set).

Algorithm :

- Detectron2 uses CNNs like ResNet or MobileNet to extract features from images. These networks help the model understand different visual patterns and details.
- The FPN combines features from various layers of the CNN. This helps Detectron2 detect objects of different sizes by using both fine and coarse features.
- The RPN uses CNN layers to find potential object locations in the image. It generates "proposals" that suggest where objects might be.

- This step processes the features from the CNN to create fixed-size maps for each proposed region. It improves accuracy by ensuring that the details are correctly aligned.
- Detectron2 has separate parts (heads) for classifying objects and refining their locations. These heads use the features from the CNN to make predictions.

Hardware configuration :

- **GPU :**T4 GPU used in google collab.
- **RAM :** 16 GB

6.3 BotSort

We finalized to use YOLOv11 for the detection of the humans.Now we are looking into tracking the detections one sorting technique we experimented with is BotSort.

Data or training :

- The inputs passed to the deep-sort are the outputs from the yolo model.The outputs we get from the yolo model are the bounding boxes around the detected persons in each frame.

Algorithm:

- Agents are detected and initialized with bounding boxes from an object detection model, establishing their initial positions and identities.
- Appearance features of each agent are extracted using a convolutional neural network (CNN) to create unique descriptors for identification.
- The algorithm predicts each agent's future position using a motion model like kalman filter.
- But this model doeenot have any re-identification model integrated with it.

Hardware configuration :

- **GPU :**T4 GPU used in google collab.
- **RAM :** 16 GB

6.4 Deepsort

We finalized to use YOLOv11 for the detection of the humans.Now we are looking into tracking the detections one sorting technique we experimented with is Deep Sort.

Data or training :

- The inputs passed to the deep-sort are the outputs from the yolo model.The outputs we get from the yolo model are the bounding boxes around the detected persons in each frame.

Algorithm:

- In each frame people are detected using yolo model and given bounding boxes.
- we will try to find which track each bounding box belongs to or if it is a new track.

- To find which track each bounding box belongs to we check for the appearance features stored for the track and also the intersection of the new bounding box with the track using hungarian algoritm.
- The further positions of a person are predicted using **kalman filter**,useful in the case of occlusions to maintain a unique id.
- It also have a re-identification model to indentify people when they go out of a video and then come in.

Deepsort is better than botsort as it better handles occlusions which is clearly showed in the RESULTS section by comparing the outputs.

Optimization parameters:

- **max_age :**maximum number of frames a track can be inactive (not updated with new detections) before it is considered lost and removed from the tracking list.
- **n_init :**the number of consecutive frames a detection must be present to initialize a new track.
- **nn_budget :**number of appearance features (embeddings) stored for each track in the memory.
- **nms_max_overlap :** determines the maximum allowable overlap (usually defined by Intersection over Union, IoU) between two bounding boxes for one to be suppressed (removed).
- **max_iou_distance :**measure of how well the predicted bounding box overlaps with the ground truth or existing track bounding box.
- **max_cosine_distance :**cosine distance (a measure of similarity) between the appearance embeddings of detected objects and those of existing tracks.
- **embedder_model_name :**Here we can specify which re-identification model we want to use.This is useful when a person goes outside a video and then comes back.Two good re identification models we can use are
  - **OSNET :**It uses a feedforwrd neural network trained from the features extracted from each person,But in this feed-forward neural network there might be connection between any two layers not only consecutive layers.
  - **RSNET :**This also uses a feedforwrd neural network trained from the features extracted from each person,But in this feed-forward neural network there might be connection between any two layers not only consecutive layers.

  Comparing both OSNET and RSNET,OSNET gives better results due to it's loss function being good and osnet is trained on fewer parameters so OSNET is also little bit faster when compared to RSNET.

Optimized these parameters by using a measuring metric.We tried to change these parameters over a range and checked which parameters give the minimum loss.

Hardware configuration :

- **GPU :**T4 GPU used in google collab.
- **RAM :** 16 GB

LINk TO THE GITHUB REPO :

https://github.com/TUC117/counting-unique-people/tree/main

7 Results

7.1 YOLOv11(fine-tuned) vs Detectron2

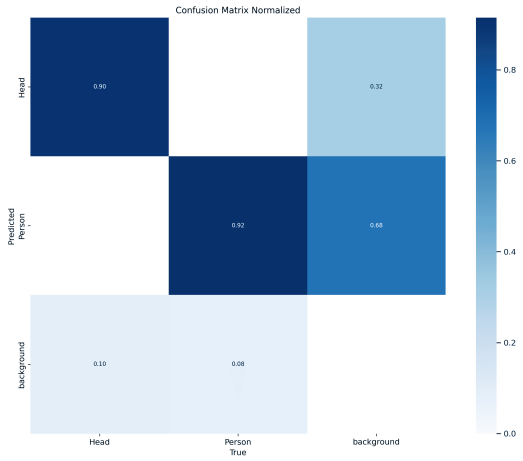


Figure 6: Caption

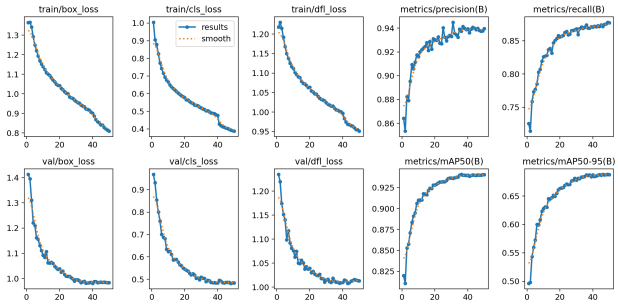


Figure 7: Caption

Above we can clearly see in the the graphs for the fine-tuned model:

- The loss for both training and validation set are decresing continiously.
- MAP value is also increasing for both training and validation set.
- From the confusion matrix you can see that person detection has 92 percent and head detection has 90 percent accuracy which are pretty good accuracies.

For Detectron2 we obsereved to have many unstable boxes when we ran it on videos,So we stick to YOLOV11 model which is giving pretty good detections, then we fine tuned yolo model for detecting persons reaching 92 % accuracy.

7.2 Botsort vs Deepsort

Here is the video output from Botsort :

t=1 t=2 t=3

t=4 t=5 t=6

t=7 t=8 t=9

Figure 8: Tracking of a person in a video

Here is the link to the complete video :
https://drive.google.com/file/d/1oKrDwUsQjLUzaQpWuCrWgcdSeb0sP1Qb/view?usp=sharing

As you can see in the video,

- From t=4 to t=6 when a new person entered he is given an id 3 which is an already assigned id but not a new id.

- The detection of the persons on backside are also not stable.

Here is the video output from Deepsort :

t=1 t=2 t=3

t=4 t=5 t=6

t=7 t=8 t=9

Figure 9: Tracking of a person in a video

Here is the link to the complete video : https://drive.google.com/file/d/1vySvINEp0CmcFyfj_v4mhOo1FIS-ugLa/view?usp=sharing

Here everything is perfect:

- Clearly from t=4 to t=6 the person is given a new id 11 which is not an already used id.
- The persons on backside are detected constanly.
- Even in the case of occlusion from t=7 to t=9 it kept the same id 11.

So this is the reason **Deep sort is better than Bot sort.**

7.3 RE-id

Even if a person goes out of a video for a small amount of time and comes into it gives the same id,But here there is a trade-off if we see the first parameter in deep sort max_age it is the max frames after which a track is deleted, so if we want a person who disappears from the video and comes after 20 sec this max_ age should be kept to a large value which

causes problems in the detection,If it is a small value we are able to get good results but we will not able to detect the same person if he disappears and appears after 20 sec.

t=1

t=2

t=3

Here is the link to the full video : https://drive.google.com/file/d/1hBMdb5SbogpQ2v8SKAwZpIrxiXupI view?usp=drive_link
Clearly Detections are very nice without any extra boxes as maxage is set to 50 but this is not the case if it is set to 200 we will get many extra boxes in the middle of the video as shown below.

t=1

t=2

t=3

Here is the link to the full video : https://drive.google.com/file/d/1a33HHlO__wQ0eSoZZPReJ-szLJy6r view?usp=drive_link
So we can see some extra boxes even though there are no persons in that boxes ans some other problems too clearly in the video when max_age is set to 200.

Frame 1

Frame 2

Frame 3

Frame 4

Frame 5

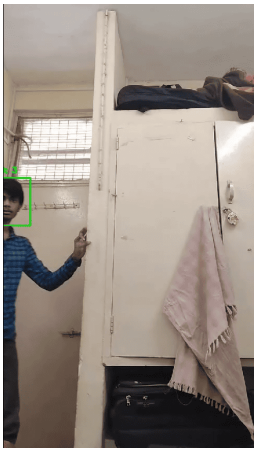
Frame 6

Figure 10: Re Identification of a person

Here is the link to the full video:
https://drive.google.com/file/d/1-0YnGS7mZA5MAFHuL8jWvuAyUHCjStI4/view?usp=sharing
As you can clearly see no id change for the person even if he leaves the video and comes back after 4 sec this is because max_age is set to a high value 200. So in this video keeping max_ age as 200 is very useful But in the same video lets see the result if max_age is set to 50.

Frame 1

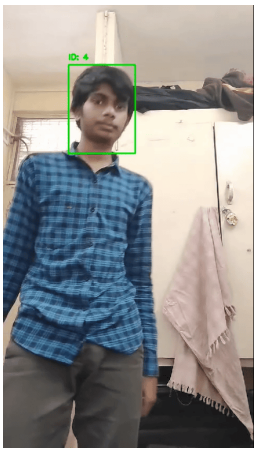
Frame 2



Frame 3

Frame 4

Frame 5



Frame 6

Figure 11: Re Identification of a person

Here is the link to the complete video:
https://drive.google.com/file/d/1-3DTcBeIjRB2b7XMC0eYw4-K77KFhyJJ/view?usp=drive_link
Clearly you can see in the video there is change in the id of the same person as he gone out of frame for 3 sec and then came back in this is because max_age is set to 50 which is small

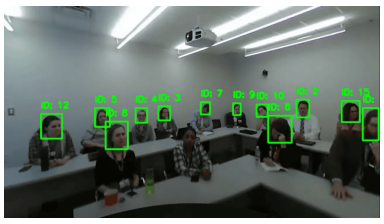
7.4 Challenges that remained

- **Face detection vs Body detection**:

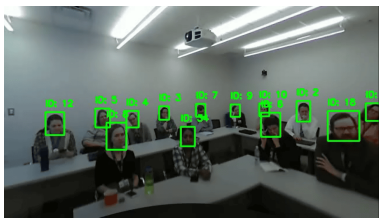
  In some cases Body detection is better where as in some cases face detection is better,But if we integrate bot of them then tracking becomes difficult as in some frames face might be detected with out body where as in some frames body is detected. Here are some results showing the difference between Face detection and body detection:

  - **Video1 :**
    - **Face detection :**

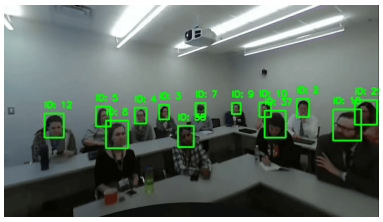
t=1



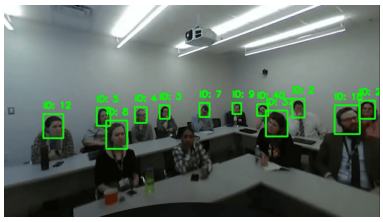
t=2



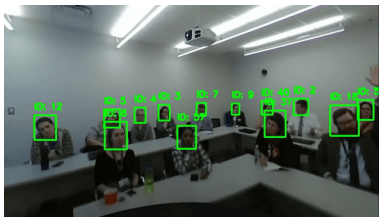
t=3



t=4



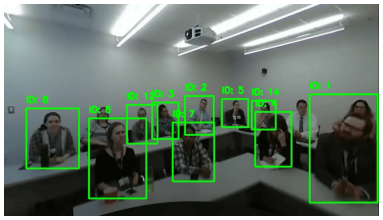
t=5



t=6

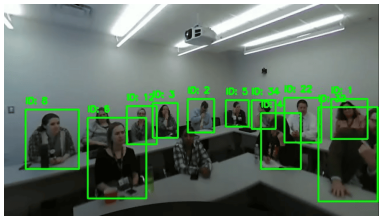
Here is the link to the complete video :
https://drive.google.com/file/d/14-284xzt3YE1OEZ_f_5LJSP4Q9-ngPeT/view?usp=sharing
Here in face detection Even the faces on the backside are also properly detected which is not possible in the case of Body detection.

* **Body detection :**



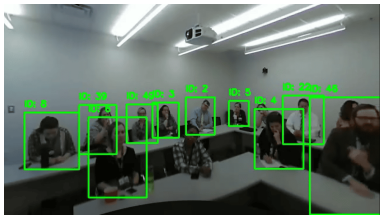
t=1

t=2

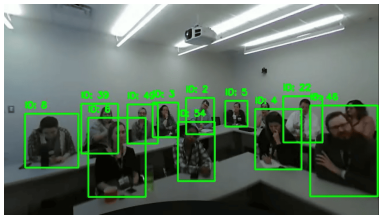


t=3

t=4



t=5



t=6

Here is the link to the complete video :

https://drive.google.com/file/d/1XpqQSGT9THcCIm-kcIiBA3gZy-23JO6j/view?usp=sharing

In body detection some people at the backside are missed. so here Face detection is better

- **Video2 :**

  * **Face detection :**

t=1

t=2

t=3

Here you can see that many faces are not detected clearly in the images from the video, as they are small So in cases like this detecting body is more efficient. Here is the link to the complete video : https://drive.google.com/file/d/1-9imdVCxlTHRAXLKtNv6Lowwu6tt-JHL/view?usp=sharing

* **Body detection :**

t=1

t=2

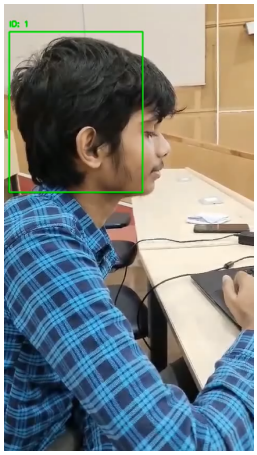
t=3

Here is the link to the full video : https://drive.google.com/file/d/1hBMdb5SbogpQ2v8SKAwZpIrxiXupIKQ-/view?usp=drive_link

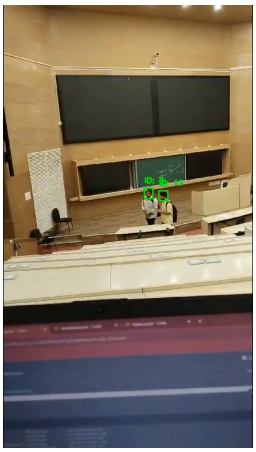
Clearly Every person is detected as we used body if we used face we will not detect all.

Here Body detection is better.

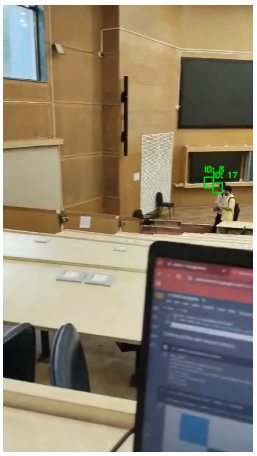
- **Wrong predictions by Deep sort due to rapid movement in camera :** When there is rapid movement the kalman filter is giving the predictions of a track wrongly as it is considering the speed of the camera,You can see this in the following video :



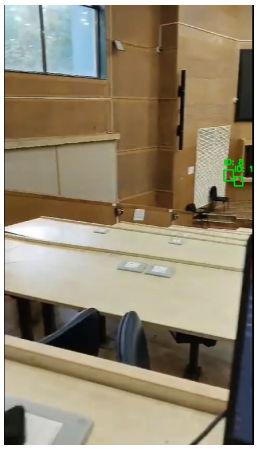
t=1



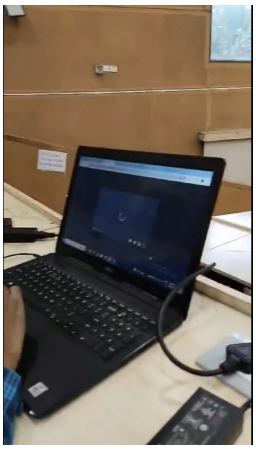
t=2



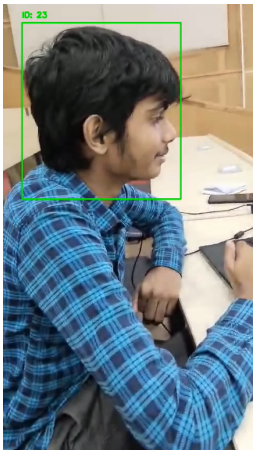
t=3



t=4



t=5



t=6

So if we see the frames from t=2 to t=5 clearly the boxes deteted in frame 2 move along the camera to frame 5 eventhough the persons are not moving this is beacuse of kalman filter wrong predictions. Here is the link to the full video :

https://drive.google.com/file/d/1-0aGZFa2Yua0M0txXHieuQxwP8gEpaCr/view?usp=sharing

8 Plan for Novelty Assessment

For the Novelty Assignment we are planning to include:

- Track of the person that is to display the path the character/person traveled.
- To detect the action which the character/person performing. i.e, Eating, Sleeping, Running, etc...

We didn't explore much into this Novelty Assessment, present we are thinking of this but based on the complexity and difficulty we may change our Novelty Assessment part.

9 Conclusion

This report consists of the detailed explanation of the project detecting and counting unique persons in a given movie clip. We divided the whole problem statement into two parts. Detecting and Tracking for detecting persons in the video we used YOLO v11 fine tuned model. We also tried with other models like detectron but they are not that accurate. For tracking we tried LSTM but it is computationally expensive, hence we used tracking algorithms like BOT SORT, DEEP SORT to track person in the video. DEEP SORT can also used for re-identification of a person in a video. Since we are working with the relation between the frames in a video we are assuming that we can also track the path a person/character took in a video and we can also detect the type of action. Hence we are assuming this is possible for Novelty Assessment.

References

[1] Minghai Chen, Guiguang Ding, Sicheng Zhao, Hui Chen, Jungong Han, and Qiang Liu. Reference based lstm for image captioning. 2017.

[2] M. Everingham and A. Zisserman. Identifying individuals in video by combining 'generative' and discriminative head models. 2005.

[3] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. 2024.

[4] Jingwen Li, Lei Huang, and Changping Liu. Tracking of humans in video stream using lstm recurrent neural network. 2011.

[5] Jingwen Li, Lei Huang, and Changping Liu. Online adaptive learning for multi-camera people counting. 2012.

[6] Jingwen Li, Lei Huang, and Changping Liu. People counting across multiple cameras for intelligent video surveillance. 2012.

[7] Masoumeh Poormehdi Ghaemmoghami. Tracking of humans in video stream using lstm recurrent neural network. 2017.

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. 2015.

[9] Li Tan, Xu Dong, Yuxi Ma, and Chongchong Yu. A multiple object tracking algorithm based on yolo detection. 2018.