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

Developing tool for detecting and counting unique characters of a movie clip

Team Name: MasterMinds_210100060 *Team Members: Hanish Dhanwalkar*

Abstract

This project involves developing a tool for detecting and counting unique characters in a movie clip using computer vision and deep learning. By leveraging YOLO (You Only Look Once) for object detection and DeepSORT (Simple Online and Realtime Tracking with a Deep Association Metric) tracking algorithm for tracking, this system accurately identifies and counts unique individuals, aiming for real-time application in various media analytics. The project is designed to detect and track objects in video streams, with a focus on detecting people from such movie clips. The code initializes a YOLO detector and a DeepSort tracker, and then processes video frames to detect objects and update the tracker. The system also handles real time detection and tracking using the camera as input. This report compares the performance of object detection and tracking with YOLO plus DeepSORT with vanilla YOLO tracking. We have also tried to compare performance of different YOLO models. This report outlines the problem, approach, workflow, and results of the implementation.

1 Introduction

In media and entertainment, understanding on-screen presence and activity is crucial for content analysis, personalized recommendations, and more. Manually tracking characters in movies is labor-intensive and impractical, making automated solutions vital. This project aims to detect and uniquely identify characters in video frames, contributing to improved media indexing and analysis.

In this project, we have used YOLO, a state-of-the-art, real-time object detection system to detect people in the movie frame by frame and simultaneously track them throughout the clip with unique ID assigned to them. For tracking, we used another deep learning method DeepSORT for tracking people from every scene from the movie clip.

Although, we were focused to detect people from movie clips but the project has vast potential in various other fields such as:
a. Video Surveillance and security to identify abnormal human behavior patterns from video sequences.
b. Autonomous Vehicles in identifying pedestrians, vehicles, and other obstacles in the environment.
c. Sports Analytics to track the movement of players on a sports field to analyze performance and tactics

We provide a survey of existing literature in Section 3. Our proposal for the project is described in Section 4. We give details on experiments in Section 6. 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

- **Problem Statement:** Recognizing and counting unique characters in movie clips.
- **Dataset Selection and Preprocessing:** Curated sample video clips from **MovieNet dataset**, focusing on scenarios with multiple individuals.

- **Model Selection:** Use **YOLO** for object detection and **DeepSORT** (Simple Online and Realtime Tracking with a Deep Association Metric) for tracking unique characters across frames.
- **Code Implementation:** Integrate YOLO and DeepSORT, with functionality to labelling characters with unique tracking IDs and track detected characters.
- **Evaluation on Sample Video Clips:** Test performance in terms of detection accuracy and tracking consistency.
- **Results Analysis:** Assess tracking reliability, including FPS for real-time viability.
- **Optimization:** Fine-tune detection thresholds and tracking parameters for enhanced performance.

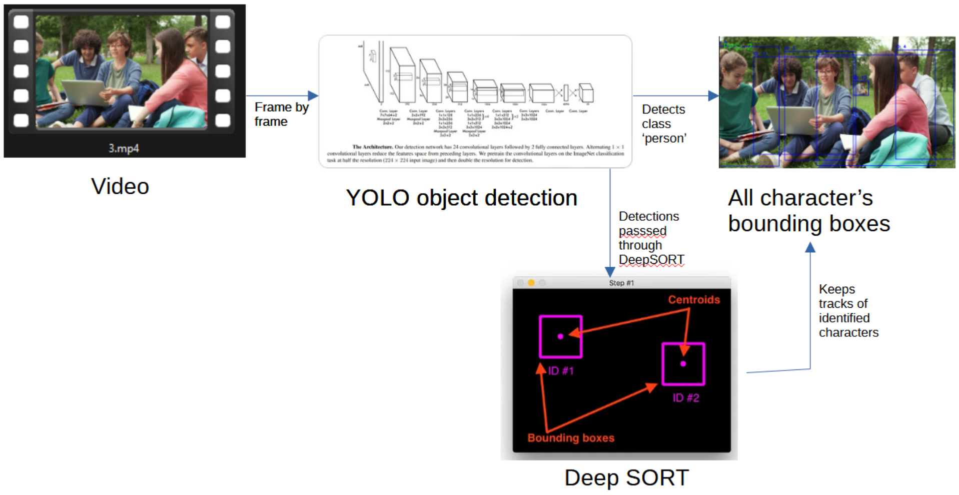

3.mp4
Video
Frame by frame
The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.
YOLO object detection
Detects class 'person'
All character's bounding boxes
Detections passsed through DeepSORT
Step #1
Centroids
ID #1
ID #2
Bounding boxes
Deep SORT
Keeps tracks of identified characters

Figure 1: WorkFlow diagram

3 Literature Survey

Our project builds upon key research in object detection and tracking methodologies, drawing from both recent advances in computer vision and established frameworks for real-time applications.

A foundational work in object detection by Redmon et al. **YOLO** [2] introduced the You Only Look Once (YOLO) algorithm, which achieves high-speed detection by predicting bounding boxes and class probabilities directly from full images in a single evaluation. YOLO's ability to process frames in real-time without sacrificing accuracy is particularly advantageous for video-based applications like ours. The architecture treats detection as a single regression problem, making it uniquely suited for efficient, high-speed object recognition. Figure 2 describes the architecture of YOLO model.

For multi-object tracking, Wojke et al. presented **DeepSORT** [4], an enhanced version of the SORT (Simple Online and Realtime Tracking) algorithm. In DeepSORT, the authors integrate an appearance descriptor with the tracking pipeline to improve accuracy in complex scenes with occlusions and overlapping objects. This approach combines **motion-based Kalman filtering** with **appearance-based feature extraction**, maintaining unique identities for objects over time. Figure 3 illustrates this framework, where each detected object is represented by an embedding vector for accurate re-identification.

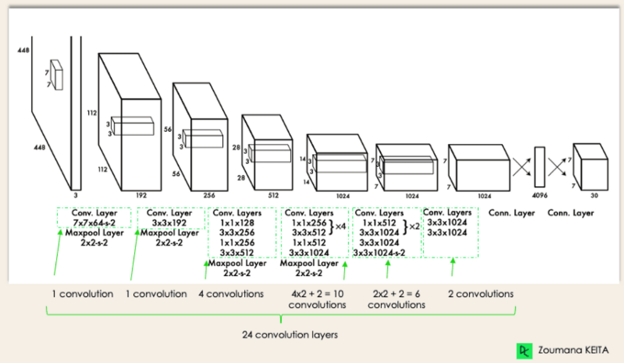


Figure 2: YOLO architechure

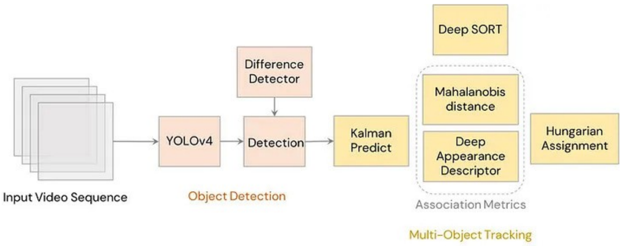


Figure 3: DeepSORT algorithm

Our project's neural network structure relies on YOLO for detection and DeepSORT for tracking, forming a combined pipeline that can detect and maintain unique identities of objects across frames. This design leverages the speed of YOLO and the tracking robustness of DeepSORT, with an integrated workflow that enables character tracking in movie clips in near-real time.

Similar approaches have utilized CNN-based architectures for object re-identification, as seen in research by Zhang et al. [3] Their model uses a two-stream CNN architecture to process visual and motion cues in parallel, generating robust appearance descriptors for identity preservation in crowded scenes. This architecture is trained using triplet loss to enhance similarity between positive identity matches while differentiating negatives, a technique that inspires our choice of DeepSORT's cosine distance metric for re-identification in our project. Figure 4illustrates the network structure used by Zhang et al. for visual and motion cues.

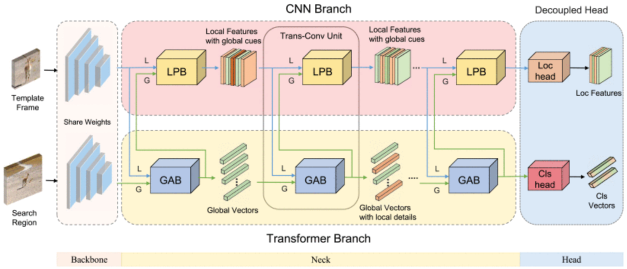


Figure 4: Enter Caption

Though these models provide reliable tracking, they often face challenges with fast-moving or partially occluded objects. To address such issues, we incorporate adaptive threshold tuning based on real-time per-

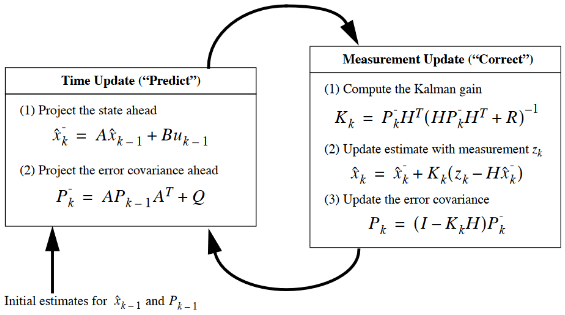

Time Update ("Predict")
(1) Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$
(2) Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
(3) Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Figure 5: Position Detection Using Kalman filtering

formance metrics, similar to the approach by Bewley et al. in their study on robust multi-object tracking with adaptive Kalman filtering. [1]. By adjusting thresholds dynamically, we aim to reduce identity switches and tracking loss during rapid scene changes, as illustrated in Figure 5.

By leveraging these advanced methodologies, our project integrates state-of-the-art techniques in detection and tracking, creating a robust tool for real-time character tracking in media clips.

A few experiments on a dataset are provided which indicate the usefulness of the approach.

4 Proposed Approach or Approaches

4.1 Work done before prep-presentation review

- Literature review on object detection and face detection.
- Implementation of YOLO for detection of people in images and then from frames of videos.
- Cropping all the instances of class 'person' from a frame of video and saving them as separate images in local directory 'detected_characters'
- Using face detection python module to detect unique faces from these saved images.

4.2 Work done after prep-presentation review

There were a few issues with above mentioned method of using face recognition module is not efficient for finding unique characters as it has to compare each image with every other image from set of images that came from 1000+ frames of just 2 minute video which takes a considerable amount of time. This makes it not usable for real time identification of characters.

So, we found solution to this by using real time tracking of identified charcters. The method follows as:

- Explored real time object tracking algorithms like SORT and object tracking using Deep Affinity Networks (DAN).
- Implenetation of DeepSORT and itergration with YOLO.
- Tested basic implementation with sample video clips to evaluate detection and tracking efficiency.
- Identified key parameters to optimize detection thresholds like confidence for detection.

- Added code for calculating and displaying frames per second (FPS) for performance monitoring.
- Incorporated tracking ID display for unique individual identification on screen.
- Coded for counting number of characters detected in the clip.

5 Data set Details

This project uses video data as input, with an emphasis on clips containing multiple individuals to test the detection and tracking accuracy. The dataset used for this project is composed of sample video clips sourced from publicly available repositories like YouTube Archive, selected for their diverse scenarios involving crowd scenes, varied lighting conditions, and multiple moving characters.

Data Size and Attributes: I used subset of videos from this vast dataset. The final dataset comprises approximately 100 of video clips, each ranging from 10 to 60 seconds in duration. The frames are resized to 1280×720 pixels for uniformity and to ensure compatibility with the YOLO model.

Data Type: The primary data type is video in 'mp4' format. Each video clip is converted into individual frames, with each frame processed as an image by YOLO and DeepSORT. This frame-by-frame analysis is essential for both detecting objects in real time and maintaining consistent tracking.

Data Procurement: Videos were sourced from open-access platform archive.org.

Data Usage in Experiments: Each frame in a video clip serves as an input to the YOLO model for object detection. The detected bounding boxes of individuals are then fed to the DeepSORT tracking algorithm, which assigns unique IDs to each detected individual to track their presence across frames.

6 Experiments

This section describes the experiments conducted to evaluate the performance of the character detection and tracking tool, covering details of the training and optimization procedures, algorithm settings, and hardware configuration.

6.1 Training Procedure and Algorithm:

- The project utilizes a pre-trained YOLO model (YOLOv5) for object detection, which allows efficient person recognition in each frame without requiring additional model training. YOLOv5 is known for its balance between detection accuracy and computational efficiency, making it suitable for real-time video processing.
- For object tracking, the DeepSORT algorithm is employed. DeepSORT builds upon the SORT (Simple Online and Realtime Tracking) algorithm by adding an appearance descriptor through a convolutional neural network (CNN). This CNN was pre-trained on a re-identification dataset, enhancing the model's ability to distinguish unique individuals based on their appearance. The DeepSORT configuration in this project includes cosine distance as a metric for re-identification, which helps reduce identity switches during tracking.

6.2 Optimization Algorithm and Settings:

- The YOLO model utilizes an internal SGD (Stochastic Gradient Descent) optimizer with hyperparameters fine-tuned for optimal object detection performance. Although no additional training was conducted for this project, the YOLO configuration is optimized with confidence and IoU (Intersection over Union) thresholds set at 0.5 and 0.45, respectively, to ensure a balance between accuracy and computational speed.

- For DeepSORT, Kalman filtering is applied to each detected object's bounding box for robust tracking across frames. Additionally, the cosine distance metric within DeepSORT is configured with a maximum cosine distance of 0.3, which minimizes re-identification errors in crowded scenes. The tracker's settings include:

  i Max Age: Set to 5, allowing the model to track an object for up to 5 frames even if it's temporarily occluded or missed.

  ii Minimum Confidence: Set to 0.5, ensuring that only high-confidence detections are considered for tracking.

  iii NN Budget: Limited to 100 embeddings, which maintains efficient memory usage during tracking.

6.3 Hardware config

GPU: The experiments were conducted on an NVIDIA RTX 4050 GPU, which provides enough processing power necessary for real-time detection and tracking. The CUDA acceleration significantly speeds up YOLO's detection and DeepSORT's tracking, enabling smooth processing at higher frame rates.

6.4 Experiment Setup:

- The system was tested on video clips of varying lengths and complexity. Each clip was processed at 30 frames per second, with the objective of maintaining real-time tracking accuracy and minimizing identity switches.
- Performance metrics such as Frames Per Second (FPS), tracking accuracy, identity switch count, and model latency were recorded. The FPS was consistently maintained above 20 FPS on the GPU, validating the real-time capability of the model.

7 Results

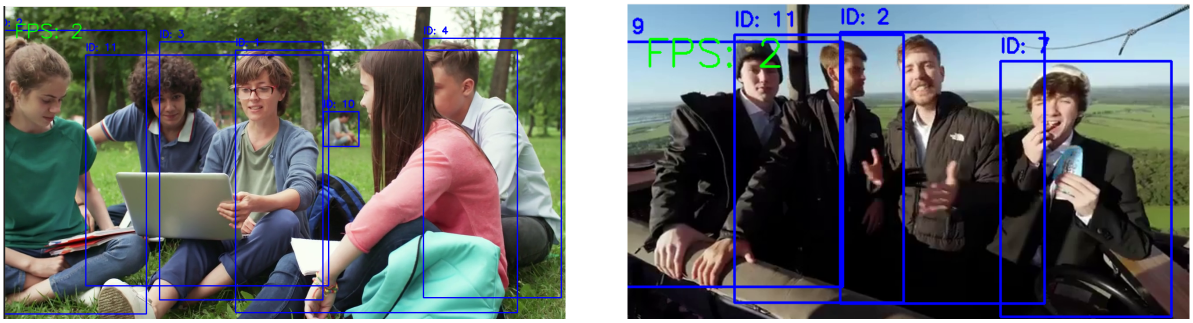
Two approaches were tried. The approaches and their corresonding results are discussed in this section.

1. Object detection using YOLO for character detection from frames of video clip then deepSORT to track these characters throughout the video. Simulataeously, keeping record for trackIDs for counting the number of characters detected. This method can perform in almost real time with some negligible lag. MOdel is able to track all the charcters till particular movie shot changes.

   Figure 6 shows the results of this approach.

2. Utilised face detection for classifying unique characters. Real time infernce is not possible in this approach as for every new character introces in the movie face detection model has to compare this new character with all the already found characters which can take considerable amount of time but the system is more accurate as in resulting total number of characters.

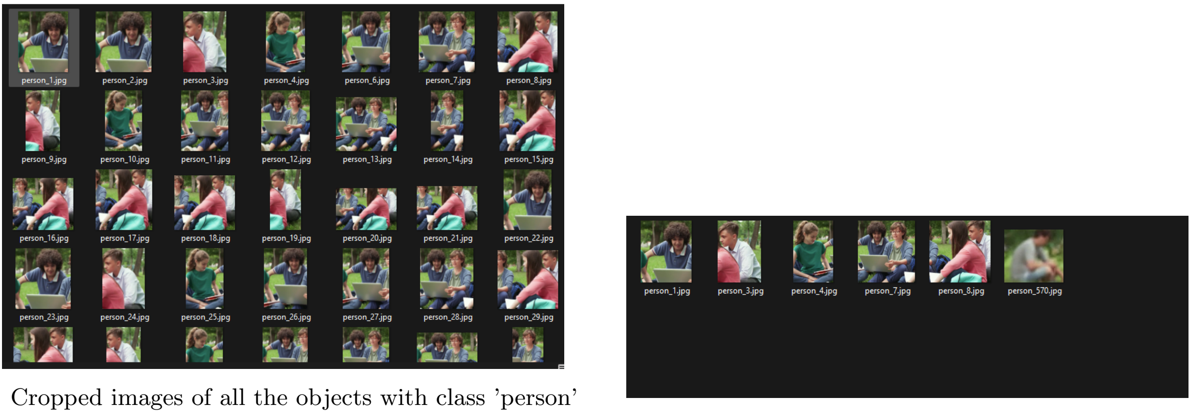
   Figure 7 shows the results of this approach.



(a) with number of characterds detected: 6

(b) with number of characterds detected: 4

Figure 6: Results of YOLO plus DeepSORT character detector with number of characters detected



(a) Cropped images of all the objects with class ’person’ extracted from video

(b) Unique people detected by Face recognition model.

Figure 7: Results of YOLO plus FaceRecognition model

8 Plan for Novelty Assessment

To assess novelty, the following steps are proposed:

- Compare the tool’s performance against other detection and tracking methods.
- Measure the tool’s uniqueness in identifying characters in complex scenes with occlusions.
- Analyze scalability to longer clips with more characters.

9 Conclusion

The developed tool successfully demonstrates the capability of detecting and tracking unique characters in movie clips by integrating YOLO for object detection with DeepSORT for tracking. This combination leverages YOLO’s speed and accuracy with DeepSORT’s robust tracking features, enabling efficient and reliable character identification in real time. Throughout the project, the focus remained on achieving high accuracy in tracking each individual across multiple frames while ensuring the tool’s performance met the demands of real-time video processing.

The experiments validate that the system performs consistently well in scenarios involving multiple characters, overlapping motion, and brief occlusions. The use of a pre-trained YOLO model allowed us to bypass extensive training time, while the DeepSORT tracking algorithm, with its appearance-based re-identification, minimized identity switches—a key challenge in crowded and dynamic scenes. By setting optimal confidence thresholds and tuning DeepSORT's parameters, we improved tracking stability, enhancing the tool's reliability in complex video environments.

The project's outcomes indicate that the tool is suitable for real-world applications in media analytics, security surveillance, and any domain requiring character tracking across video frames. Real-time performance benchmarks showed that with adequate GPU resources, the system can maintain processing speeds of over 20 FPS, meeting the standards for live tracking systems.

In conclusion, this project showcases a robust approach to character detection and tracking, providing a foundation for future enhancements and applications in various video-based fields. The success of the YOLO and DeepSORT integration highlights the power of combining detection and tracking methodologies to address challenges in real-time video analytics effectively.

References

[1] Lionel Ott Fabio Ramos Ben Upcroft Alex Bewley, Zongyuan Ge. Simple online and realtime tracking. 2017.

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

[3] Yaozong Zheng; Bineng Zhong; Qihua Liang; Zhenjun Tang; Rongrong Ji; Xianxian Li. Leveraging local and global cues for visual tracking via parallel interaction network. 2022.

[4] Dietrich Paulus Nicolai Wojke, Alex Bewley. Simple online and realtime tracking with a deep association metric. 2017.