Transactions on Machine Intelligence 6(4) (2023) 182-197

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

An Improved Object Tracking Technique for Remote Weapon Station Using Yolov5_Deepsort_Dlib Architecture

O. Ezekiel Olorunshola[1,*], M. Ekata Irhebhude[2], A. Eseoghene Evwiekpaefe[3]

[1] Computer Science Department, Air Force Institute of Technology, Kaduna State, Nigeria
[2] Computer Science Department, Nigerian Defence Academy, Nigeria
[3] Computer Science Department, Nigerian Defence Academy, Kaduna, Nigeria

ARTICLE INFO

Article History:
Received 2 September 2023
Received in revised form 8 October 2023
Accepted 5 November 2023
Available online 11 November 2023

Keywords:
Object Detection, Object Tracking, YOLO, Deepsort, SAHI, Dlib

ABSTRACT

This paper introduces an advanced tracking object architecture named DeepSORT_YOLOv5_Dlib. Building upon the DeepSORT_YOLOv3 framework, the study [1] integrates the Digital Library's correlation tracker into the traditional DeepSORT_YOLOv3 to minimize identity switches. Notably, the architecture is designed to operate in parallel, enhancing its operational speed. Experimental results indicate that the proposed approach outperforms the conventional DeepSort_YOLOv3, showcasing reduced identity switches and increased operational speed across various video testing scenarios. The custom model employed in this study adopts a confidence threshold of 0.2 and an image size of 416 x 416, consistent with the training size. To boost detection within YOLOv5, the model incorporates the Slicing Aided Hyper Inference (SAHI) technique. The overall inference speed in this study reaches 314.8fps, a notable improvement compared to Dang's 218.6fps. Evaluation using the COCO dataset demonstrates the model's precision at 0.98 and a recall of 0.81. Additionally, the proposed custom model exhibits a MOTA of 0.86, surpassing the benchmark's 0.83. Notably, our model achieves a significantly lower identity switch count of 1881 compared to the benchmark's count of 2288. Furthermore, it outperforms the benchmark in object detection capabilities. By incorporating SAHI inference with YOLOv5, the study enhances detection accuracy, resulting in an overall tracking accuracy improvement from 56% to 79%. These findings highlight the efficacy of the proposed custom model in achieving superior performance in object tracking and detection.

1. INTRODUCTION

Object tracking plays a pivotal role in Computer Vision (CV), involving the continuous tracking of objects across successive frames in a video. These objects can range from people and animals to vehicles and other items of interest [2]. The applications of object tracking are diverse, encompassing areas such as surveillance, medical imaging, traffic

* Corresponding Author: seyisola25@yahoo.com
Computer Science Department, Air Force Institute of Technology, Kaduna State, Nigeria

http://dx.doi.org/10.47176/TMI.2023.182

flow analysis, self-driving cars, unmanned aerial vehicles, people counting, and audience flow analysis [3]. However, before object tracking can take place, object detection becomes a prerequisite. This initial step involves identifying and delineating objects in an image, often assigning bounding boxes around them [4].

In the context of object tracking, as discussed in [5], each detected object is assigned a unique identity (ID). The tracking algorithm aims to sustain this ID across subsequent frames while accurately determining the object's new position. Object detection generates an array of rectangles encapsulating the identified objects [5]. Over the years, object detection has witnessed significant advancements, with a shift from traditional statistical and machine learning methods to more robust deep learning approaches. This transition has substantially improved the accuracy and efficacy of object detection [6]. While some argue that object tracking and object detection share similarities, it is generally acknowledged that object tracking is a more sophisticated process compared to object detection [7].

The integration of Computer Vision (CV) within artificial intelligence (AI) has played a crucial role in minimizing human errors associated with visual and optical tasks, particularly in the context of effective ground troop fire support [8]. Fire support involves the strategic use of acquired target data, employing various lethal or non-lethal means against ground targets to provide additional firepower to forward troops. The Battle of Crécy serves as a historical illustration, showcasing the tactical advantage of firepower, as a smaller force prevailed over a larger and seemingly superior enemy, marking the onset of the "age of firepower" [9].

A significant approach to bolstering fire support for troops involves the implementation of Remote Weapon Stations (RWS). According to [10], a remotely operated weaponized system (RWS) is equipped with a fire-control system designed for light and medium-caliber weaponry, suitable for mounting on various ground-based vehicles. These vehicles include tracked or wheeled combat vehicles, tactical vehicles, light vehicles, and trucks. The RWS allows gunners to operate from within the vehicle, ensuring their safety. The incorporation of artificial intelligence into the operating system of an RWS enhances its accuracy [11]. This technology provides a secure environment for operators, maintaining tactical advantages, particularly in scenarios where accurately targeting fast-moving aircraft or small drones proves challenging [12].

The sighting and tracking models deployed in a Remote Weapon Station (RWS) encounter challenges related to maintaining high operating speeds, accuracy, and addressing identity switching issues [6]. Identity switching, defined by [13], occurs when the identities of two or more tracked targets are mistakenly interchanged due to similarities or overlap. Such switches are common when two targets meet and share a significant overlap, analogous to player identity exchanges in sports when players are in close proximity. Identity switching poses a substantial challenge in object tracking, potentially leading to false positive results and diminishing the overall accuracy of the tracking model. The sighting component of an RWS comprises a camera operating on a detection-by-tracking model, which detects and tracks the objects of interest.

To address the challenge of identity switching in the context of Remote Weapon Stations (RWS), this research introduces an innovative approach by integrating the correlation tracker of the digital library (Dlib) algorithm into the DeepSORT and YOLOv3 architecture [1]. The proposed technique underwent testing on two videos, demonstrating a substantial reduction in identity switches from 3.96% to 1.21% in the first video and from 4.93% to 3.15% in the second video. Although the reduction percentages were 30.6% and 63.9%, respectively, there remains a necessity to develop a technique that significantly minimizes identity switches to enhance the overall effectiveness of RWS. Furthermore, there is a need to design a simplified tracking technique to improve operational speed.

This research aims to enhance the performance of object detection and tracking techniques using YOLOv5 and DeepSORT, ultimately contributing to superior RWS performance. Extensive simulations and tests were conducted to validate the effectiveness of the improved technique, comparing it against existing methodologies. The study focuses on four classes of objects: person, handgun, rifle, and knife, chosen strategically to enable the system to detect and track individuals and weapons entering the designated region of interest (ROI) where the system is deployed. Given that the model is primarily intended for RWS use, it prioritizes objects posing potential security threats, including people, rifles, handguns, and knives. This diverse range of weapons empowers RWS users to tailor countermeasures according to the specific type of threat. Implementation of this enhanced RWS technique and architecture not only boosts the accuracy of the RWS but also enhances the gun's efficiency, increases the probability of hitting the target, reduces identity switches, minimizes ammunition waste, yields cost savings, and enhances

morale for the fighter. The incorporation of Slicing Aided Hyper Inference (SAHI) into the proposed architecture enhances object detection in the YOLOv5 model. The proposed technique for improving the object detector-tracker detects small object instances of interest, which may be located at substantial distances from the system deployment site. This research presents a simplified tracking architecture that enhances real-time tracking speed, minimizes identity switches (IDs), and improves Multiple Object Tracking Accuracy (MOTA).

Section 2 provides a brief review of previous works in object detection and tracking, while Section 3 outlines the methodology. Section 4 examines the outcomes and confirms the efficacy of the developed methodology, and ultimately, Section 5 outlines the findings of this investigation.

2. LITERATURE REVIEW

A system was developed that uses a Raspberry Pi 3 Model B and a Pi camera to detect intruders attempting to breach Homeland Security. An autonomous targeting system directs a laser towards any humans detected within its visual range. Traditional methods rely on expensive equipment, including optoelectronics and radar, to identify potential intruders, which drives up maintenance costs. By incorporating an enhanced Continuously Adaptive Mean Shift (CAMShift) algorithm, the authors significantly reduced the system's time complexity and resource consumption. The raspberry pi-based system prioritized diminishing complexity and cost, while concurrently enhancing accuracy. The Raspberry Pi 3 was powered by the Broadcom BCM2837 system-on-chip, which included four high-performance ARM Cortex-A53 processing cores running at 1.2GHz, 32kB Level 1 and 512kB Level 2 cache memory, and a Video Core IV graphics processor. Additionally, it was linked to a 1GB LPDDR2 memory module located at the rear of the board. Like its predecessor, the Raspberry Pi 3 utilized the SMSC LAN9514 chip, which provided 10/100 Ethernet connectivity and four USB channels. The SMSC chip connects to the System on Chip using a single USB channel, functioning as both a USB-to-Ethernet adapter and a USB hub. The ATS system can be installed on a mobile vehicle or a caterpillar wheeled remote-controlled or autonomous drone [11].

The study involved an investigation into the performance of diverse object detection algorithms on surveillance video. The study evaluated the stand-alone performance of the object detection algorithm and correlated that with the overall performance of a tracking-by-detection system. In the tracking stage, visual descriptors were implemented with the aid of Microsoft Visual Object Tagging Tool. The study indicates a substantial correlation between the individual performance of the object detection algorithm utilized in the tracking-by-detection system and the overall tracking performance of the system. However, the authors faced a significant limitation in training the model due to restricted annotated data, which resulted in the absence of an accurate evaluation of its performance [14]. In their research, they proposed a unified neural network with capabilities for object detection, multiple object tracking, and vehicle re-identification (RE-ID).

The authors utilized a region of interest (ROI) feature vector as input and implemented triplet loss to train the branch. They then integrated the detector with the RE-ID model into an end-to-end network by adding an additional track branch for tracking in the Faster Region-based Convolutional Neural Network (RCNN) architecture. To extract feature maps from detected object images, the RE-ID model integrated into DeepSORT required the use of deep CNNs. By implementing a unified network, the researchers trained the entire model end-to-end using multi loss, which demonstrated significant benefits in other recent studies and reduced computation. The system achieved a mean Average Precision of 57.79% (mAP) and exhibited remarkable performance in human eye vehicle tracking. Limitations included incorrect matches and track misses, which were attributed in part to imperfect dataset annotations. Another factor contributing to the method's low accuracy was the use of center loss or other metrics to enhance the model's robustness. Additionally, the threshold set for determining whether two vehicles are identical during inference in a single camera scenario may not be appropriate for a multi-camera scenario [15].

A new multiple object tracker has been proposed following the popular tracking-by-detection scheme. An optical flow network was implemented to address the camera motion problem and an auxiliary tracker was utilized to handle the missing detection problem. The approach significantly enhanced the performance of multiple object tracking while maintaining high efficiency, as evidenced by experimental results from the VisDrone-MOT dataset. The primary evaluation metric for the test set is the mean average precision (mAP) across object classes at varying

thresholds. The Flow-Tracker utilized in the study achieved an average precision of 30.87%, outperforming all baseline methods, and maintaining a running speed of 5 FPS [16].

To achieve real-time detection and tracking of multi-person targets in surveillance videos end-to-end, a multi-target tracking algorithm employing DeepSORT was utilized based on deep neural network [17]. The experiment revealed that the differential YOLOv2 detector's implementation resulted in an average MOTA value of 63.75% for the algorithm, with a maximum value of 86.8. The experiment revealed that the differential YOLOv2 detector's implementation resulted in an average MOTA value of 63.75% for the algorithm, with a maximum value of 86.8. The algorithm also operated at an average speed of 81.6 Hz, with the fastest speed recorded at 140 Hz. Additionally, the differential YOLOv3 detector's use showed an average MOTA value of 78.4% for the algorithm, with a maximum of 92.3%. The algorithm ran at an average speed of 67.1 Hz and recorded its fastest speed at 117 Hz. The experiment revealed that the differential YOLOv2 detector's implementation resulted in an average MOTA value of 63.75% for the algorithm, with a maximum value of 86.8. As a result, the combination algorithm was successful in providing accurate real-time tracking. This algorithm's detection accuracy is generally consistent.

Efficient data has significantly improved speed and accuracy benchmarks. The use of computer vision (CV) and artificial intelligence (AI) has visualized this impact. Two technologies, CV and AI, have empowered major tasks such as object detection and tracking for traffic vigilance systems. The increasing number of image features has led to a higher demand for algorithms that can efficiently excavate hidden features. The CNN model was created to detect single objects in the urban vehicle dataset, while YOLOv3 was utilized for detecting multiple objects in the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) as well as Common Objects in Context (COCO) datasets. The research supported the distinctiveness of highly advanced networks such as DarkNet. The CNN model, trained on a dataset of road vehicles for single object detection, achieved a validation accuracy of 95.7% for automobiles, 95.5% for cars, and 96% for heavy vehicles in day images [18].

A comparative analysis of various object detection and tracking algorithms was conducted by [19], who deployed several state-of-the-art algorithms to detect and track different classes of vehicles in their region of interest. The detection algorithms utilized were CenterNet, Detectron2, YOLOv4, and EfficientDet, whereas the tracking algorithms employed were intersection over union tracker (IOU), Kalman intersection over union (KIOU), simple online and real-time tracking (SORT), and DeepSORT. The primary objective of this study was to achieve precise detection and tracking of vehicles for obtaining a reliable vehicle count. Out of the sixteen models resulting from combining detection and tracking algorithms, experimental results indicate that the most effective combinations are YOLOv4 and DeepSORT, Detectron2 and DeepSORT, and CenterNet and DeepSORT detector-tracker models. The system detected certain vehicles multiple times due to occlusion and poor visibility, resulting in identity switches.

The system proposed utilizes YOLOv3 for object detection and the DeepSORT algorithm for multiple object tracking. YOLOv3 and DeepSORT models were employed in the detection and counting of vehicles on a global scale. The ultimate objective was to achieve automation of volumetric surveys in Detailed Notice Inviting Tender (DNIT) in a manner that is both non-invasive and cost-effective. The global vehicle count achieved a precision level exceeding 90%. Furthermore, the system outperformed other proposed tools with a 99.15% precision rate on public datasets. A lack of data to train the system to accurately classify vehicles according to DNIT PNCT vehicle classes based on their axles was a limitation of the study [20].

In response, [21] developed a deep learning-based framework that employs object detection and tracking models to facilitate the implementation of social distancing measures to address the rising number of COVID-19 cases. To achieve a balance of speed and accuracy, we utilized YOLOv3 and DeepSORT as object detection and tracking techniques. Each detected object was enclosed in bounding boxes, which were then used to calculate the pairwise distance of the coordinate from the origin of the vector space (L2 norm). This computationally efficient vectorized representation was employed to identify clusters of individuals who were not adhering to social distancing guidelines. YOLOv3 and DeepSORT were utilized to perform statistical analysis by simulating the total number of social groups displayed by the same color encoding. A violation index term was computed as the ratio of the number of people to the number of groups. However, one of the major limitations of this approach is that accuracy and precision are both necessary to operate in real-time environments. Inaccuracy may result in false positive alarms, which can cause panic among individuals.

The Inception v2 model had the highest ratio of accuracy to the number of parameters, demonstrating adequate classification accuracy with minimal trainable parameters compared to other models. Therefore, it is employed as a backbone architecture for efficiency in computations in faster RCNN and SSD object detection models. Meanwhile, YOLOv3 incorporates a different architecture, Darknet-53[21].

Presented is a deep learning object detection and tracking system for lane-specific vehicle counting and velocity estimation using YOLOV4 and DeepSORT. The object tracking system was implemented with the introduction of VirtualLines and Hot Zones, improving vehicle tracking accuracy and reducing multiple counts of specific vehicles during subsequent calculations. There were limitations in predicting accuracy, particularly in lane-specific counting. Additionally, the security of wireless sensor networks (WSNs) used for transmission was a challenge, as the data is transmitted wirelessly to the control center.

To address these issues, a real-time and robust system was proposed for counting movement-specific vehicles at crowded intersections using YOLO detection. To achieve this, a vehicle counting scheme based on cosine similarity was applied. Firstly, we identified the direction proposals of each track using cosine similarity, and then we predicted the correct movement by measuring the similarity between the trajectory of the track and the pre-defined movement trajectories. This process determines the direction of movement of the vehicle, enabling us to track objects smoothly without being affected by sudden disappearances due to weather or visibility factors. The limited dataset used to train the model caused the technique to miss and also affected its speed performance.

Yolov4-CSP, Yolov4-CSP-0.25, and Yolov5x. Yolov4-CSP with a gamma value of 0.25 exhibited faster FPS than both Yolov5x and Yolov4-CSP, while maintaining a similar mAP to Yolov4-CSP and a higher mAP than Yolov5x. The Yolov4-CSP-0.25 model, denoted as Yolov4-CSP-0.25-sync[23], incorporates pre-training with synchronized batch normalization to further improve mAP with a batch-size of 8. The authors also proposed utilizing the DeepSORT multi-object tracking algorithm to enable the detection and tracking of pedestrians and vehicles in traffic scenes.

The study compared the performance of three object detection models: The Yolov4-CSP-0.25 model, denoted as Yolov4-CSP-0.25-sync [23], incorporates pre-training with synchronized batch normalization to further improve mAP with a batch-size of 8. The authors also proposed utilizing the DeepSORT multi-object tracking algorithm to enable the detection and tracking of pedestrians and vehicles in traffic scenes. In this study, we initially trained a model for detecting vehicles and pedestrians in traffic scenes using YOLOv4. To minimize ID switching and address occlusion loss during real-time scenarios, we combined YOLOv4 with DeepSORT. Each identified object was assigned a unique ID, with minimal switching occurring during the movement of most vehicles and pedestrians. The algorithm attained a real-time speed of 35 FPS as the tracking ID moved with fixed targets. Since YOLOv4 was utilized to minimize ID switching, YOLOv5 is expected to perform even better. The efficiency of the current multi-target tracking depends heavily on its detection performance, which improved by 18.9% through a switch in detectors. The detection performance can be increased by 18.9%, allowing it to match the 2016 Self-Organizing Tree Algorithm (SOTA) algorithm. Additionally, the speed achieved 260 Hz, which is 20 times faster than other detection algorithms currently available [24].

They attempted to determine social distancing using YOLO object detection on video footage and images. The YOLOv5 object detection method was employed to identify individuals in a video, achieving a faster processing time that was capable of providing real-time results without sacrificing precision, even in complicated settings. However, the system was implemented with virtual cameras and not tested in real-life scenarios [25].

In [26], the aim was to develop a real-time multiple object tracking (MOT) framework using the DeepSORT algorithm. To achieve this, a modified DeepSORT was combined with YOLO detection methods resulting in improved detection and execution performance. Custom training of YOLO on the UA-DETRAC dataset contributed to this success. Although the DeepSORT algorithm has the capability of tracking and labeling more than one class simultaneously and adjusting the tracking process to accommodate camera movement, these features were not attained. During the training process, an average loss of 1.583 and a mAP of 98.68% were measured, indicating favorable results that could enhance the performance of the DeepSORT framework. The YOLOv4 detector trained on the UA-DETRAC dataset successfully accomplished faultless numbering and tracking throughout the entire test scene, commencing with the "Racetrack 480p" sequence. Furthermore, the execution performance saw an increase

of approximately 10%. developed granulated RCNN (G-RCNN) and multi-class DeepSORT (MCD-SORT), for object detection and tracking. Image Processing (IP) techniques or machine vision system (MVS) were used for object detection and tracking. The characteristics features were demonstrated over 37 videos containing single–class, and multi-class objects. Among all the combinations of detectors and trackers considered in the experiment, the one like G-RCNN + MCD-SORT was found to be the best. The speed of these tracking algorithms, such as Statement of Purpose (SOP), AMIR15, and AM, were 14 fps, 6 fps, and 11 fps, respectively [27].

Proposed is a motion compensation model, KFHT, utilizing the Kalman Filter and Homography Transformation for multiple object tracking (MOT) to alleviate tracking position drift caused by camera fast movement. The efficacy of the improved algorithm was assessed through experimental evaluation on the VisDrone2019 dataset, utilizing DeepSORT, YOLOv5 detection results, and prior ground truth. The study's findings indicate that the algorithm described in their paper resulted in fewer identity switches by 17% with YOLOv5 and 66% with prior ground truth. Additionally, the tracking accuracy improved by approximately 1.5% and 3.6% in MOTA, respectively [28].

The researchers employed a novel mixed model approach that combined YOLOv5 and DeepSORT for fire detection. The authors enhanced the accuracy of fire detection by utilizing flame characteristics extracted from training data to improve YOLOv5 and DeepSORT's detection and tracking capabilities, which addressed the issue of high false alarm rates. The labeling tool utilized was Intel's Computer Vision Annotation Tool (CVAT) to minimize labeling uncertainties. The findings indicated that YOLOv5 had an accuracy rate of 75% at 253 frames and 77% at 527 frames [29].

Evaluated the effectiveness of YOLOv5 algorithm with DeepSORT object tracking on river user footage for flood safety surveillance. Investigated the suitability of YOLO algorithm for human detection using COCO database. Although YOLOv5 algorithm occasionally failed to recognize partially occluded individuals on a frame-by-frame basis, it successfully identified river users, indicating that YOLOv5 coupled with DeepSORT tracking presents a practical model for human detection. Testing demonstrated that the algorithm could identify individuals without obstruction at approximately 200 feet, using a high-definition camera [30].

Researched the use of modern YOLO algorithms, including YOLOv3, v4, and v5, for multiclass 3D object detection and recognition, utilizing the large-scale Pascal VOC dataset. The results indicate that each YOLO algorithm exhibited unique outcomes, with YOLOv3 achieving the highest recognition accuracy and YOLOv5 demonstrating the lowest processing time [31].

Proposed detection of domesticated chickens in videos with varying complex backgrounds using YOLOv5 and DeepSORT. The YOLOv5 model proposed in this study includes a Cross Stage Partial (CSP) backbone network. The object detection function refines flock density features by deploying convolutional networks to facilitate the generation of trajectory movements, counting, and tracking. Using the Kalman filter, this study tracks multiple chickens simultaneously and aims to associate individual chickens across video frames for real-time and online applications. The Chick Track model's recall and precision values confirm its superior performance in detecting chickens within congested scenes, despite various occlusions and distribution densities. The study revealed that the suggested model for the identification, enumeration, and tracking of chickens is resilient and has the capacity to be integrated into farming operations. The uneven distribution of individual measurements and flock concentration presents difficulties in detecting, tallying, and monitoring the chickens' movements [32].

The authors propose a new region-based deep learning methodology to automate product counting, utilizing a customized YOLOv5 object detection pipeline and DeepSORT algorithm. The authors developed a framework tailored for automatic retail checkout. They encountered challenges with diverse images, object shapes, and noise. The proposed method involves initially constructing a strong object detection model with YOLOv5. The proposed solution has been validated on challenging real-world video data and has the potential to have immediate positive impacts on society by enabling fully-autonomous product checkout. The proposed method was awarded 4th place in Track 4 of the 2022 AI City Challenge with an F1 score of 0.4400 on experimental validation data[33].

This study aims to enhance the YOLOv5 detection algorithm and construct a tracking model using the improved YOLOv5_DeepSORT_Dlib architecture for object tracking. We implemented this tracking architecture on RWS.

1.1. Background of the Proposed Algorithms

In this study, YOLOv5 detection model, Dlib tracker and the DeepSORT tracker are used to achieve the proposed model. The background of these methods is discussed as follows:

The YOLOv5 is the latest release of the YOLO models. The YOLOv5 is nearly 90% smaller in size than YOLOv4 [34]. The YOLOv5 utilizes Cross Stage Partial Network (CSPNet) in DarkNet to create CSPDarknet53 as the backbone of YOLOv5 shown in Figure 1.

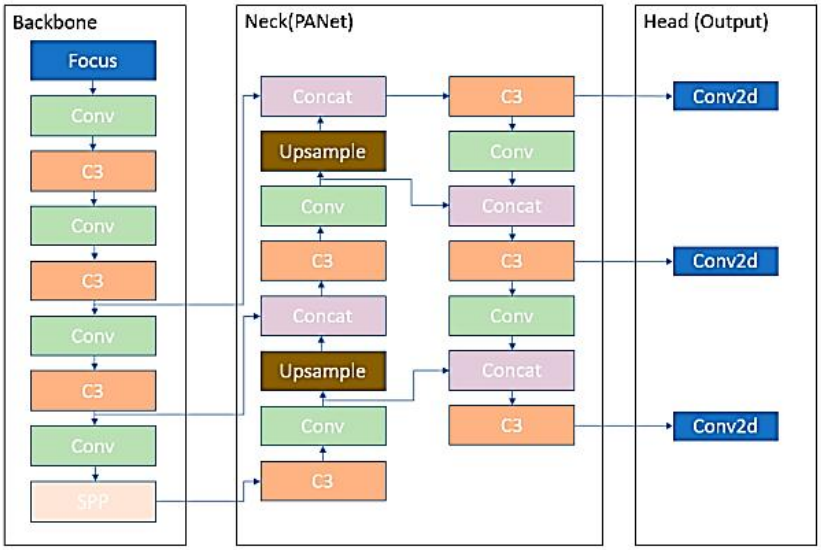

Fig. 1. YOLOv5 Architecture [34].

This model backbone solves the issue of redundant gradient information in large backbones while incorporating gradient change into the feature map. This model backbone solves the issue of redundant gradient information in large backbones while incorporating gradient change into the feature map. As a result, inference speed improves while model accuracy increases and model size decreases by reducing the parameters. A neck incorporating path aggregation network (PANet) enhances the information flow. PANet utilizes a novel feature pyramid network (FPN), including multiple bottom-up and top-down layers, which improves low-level feature propagation in the model. This model backbone solves the issue of redundant gradient information in large backbones while incorporating gradient change into the feature map. PANet enhances the object's localization accuracy by improving localization in the lower layers. Yolov5 utilizes a Focus structure with CSPdarkent53 as its backbone, in contrast to Yolov4, which uses CSPdarknet53 only, and Yolov3, which uses Darknet53. A Focus layer decreases the requisite CUDA memory, reduces layers of the algorithm, and increases forward and back propagation [35].

The Digital Library (Dlib) tracker is an open-source library with numerous commonly used image processing algorithms and trained models. It is extensively implemented for face detection in the field [36]. In particular, the Dlib employs a correlation tracker to monitor the object's location across successive frames of a video sequence. It makes use of the boundary box of the object in current frame to identify the object to track in subsequent frame. The mathematical formula is as follows:

$$S^{(t+1)} = S^{(t)} + r_t(I, S^{(t)}) \quad (1)$$

rt is the prediction residual based on the feature, S(t) represents the shape of the feature point predicted in the previous iteration, $r_t(I,S(t))$ represents the residual calculated in the current layer, $S(t+1)$ represents the result of the t_{th} iteration.

DeepSORT is an improvement of the traditional Simple Online and Real-time Tracking (SORT) algorithm which uses kalman filter (Liu & Juang, 2021). The DeepSORT computes bounding boxes using a detection model, uses SORT's Kalman filter and identification model (ReID) to link bounding boxes and tracks and then if no link can be made, a new ID is assigned and it is newly added to tracks [22]. DeepSORT is a tracking-by-detection method,

which defines the tracking scenario on an eight-dimensional state vector (x, y, γ, h, x', y', γ', h') that contains the bounding box center position (x,y) and height h from detection algorithm, aspect ratio γ and their respective velocities in image coordinates. The updated trajectory is predicted using a standard Kalman filter with constant velocity motion and linear observation model. The direct observations of the object state are bounding coordinates (x, y, γ, h). For each track, there is a threshold a_k for recording the time from the last successful match to the current time. When the value is greater than the threshold Amax set in advance, the track is considered to be terminated.

3. METHODOLOGY

The initial objective of this research is to collect and process data. Deep learning greatly relies on data science, and the effectiveness of a deep learning approach is largely dependent on the quantity and quality of the data used to train the model [37]. Deep learning greatly relies on data science, and the effectiveness of a deep learning approach is largely dependent on the quantity and quality of the data used to train the model [37]. To build an RWS, the improved detector-tracker technique is utilized ensuring the weapon can securely track its target, while the operator controls the trigger function. Since deep learning techniques are employed, the targeting accuracy is expected to improve over time due to the continuous feeding of new data into the system with every operator correction [38].

3.1. Dataset Description

Google Open Images Dataset [40] , Roboflow Public Dataset [39] and locally sourced images were used to train a version of the model in order to be used effectively with the RWS. Primarily, the images gotten from Google Open Images which is a large-scale dataset with different trainable classes were a total of 5808 and it comprises of Person, Handgun, Rifle and Knife. A total of 2971 images of Pistols were also gotten from Roboflow Public Dataset and modified to class "Handgun" and added to the dataset to make a total of 8779 images. Locally sourced dataset of about 1000 images were captured using a high-definition D5100 DSLR camera. Images captured have resolution of 720 pixels (1280 x 720). These local images comprises of different images of Person, Handgun, Rifle and Knife. These images were gathered, cleaned and annotated using Roboflow Annotation tool [39].

Data preprocessing done on the dataset was Auto-Orient and the images were resized to 416 x 416 (weight x height) size which is the size used by YOLOv5. The entire dataset used for the training were 9779 images containing 21,561 annotations of the four classes (13545 Persons, 4159 Handguns, 3007 Rifles and 850 Knives). The dataset was split into training, testing and validation on ratio 60:20:20 of the number of images annotated. 5867 images which makes up 60% of the dataset was used for training while 1955 images which makes up 20% of the total images was used for testing and 1955 images which amounts to 20% of the total images remaining was used for validation. After these images were preprocessed and ready (training data), they were then fed to the deep learning algorithm. This was done via Google Collaboratory (Colab) platform which provides an environment for training and running machine learning codes. The Microsoft Common Object in Contexts (MS COCO) 2017 dataset which is a public benchmark dataset for detection and segmentation was used to train another version of the model.

This was done primarily to verify that the model developed truly performs better than already existing models and that it is not dataset bias. The MS COCO 2017 dataset consists of 163,957 images. The dataset was split into training, testing and validation with 118,287 images for training, 40,670 images for testing, and 5,000 images for validation. Lastly, the trained model was also evaluated with the MOT17 Challenge dataset which is a benchmark dataset for single camera multiple objects tracking models. The MOT17 dataset contains sequence of video frames of different video samples with each sequence provided with three detectors; Deformable Parts Model (DPM), Faster Region-based Convolutional Neural Network (Faster-RCNN) and scale-dependent pooling (SDP). The training video samples contains a total of 15948 frames (645s total video length), 1638 tracks and with FPS of range 14 to 30 on different resolutions. The test video samples contain a total of 17757 frames (744s video length), 2355 tracks and with FPS of range 14 to 30 on different resolutions. Figure 2 shows the sample images of the dataset gotten from Google Open Images Dataset (left), Roboflow Public Dataset (centre) and locally sourced images (right). The diagram of the mechanism of the RWS is shown in Figure 3.

Fig. 2. Sample images of the dataset used in the research.

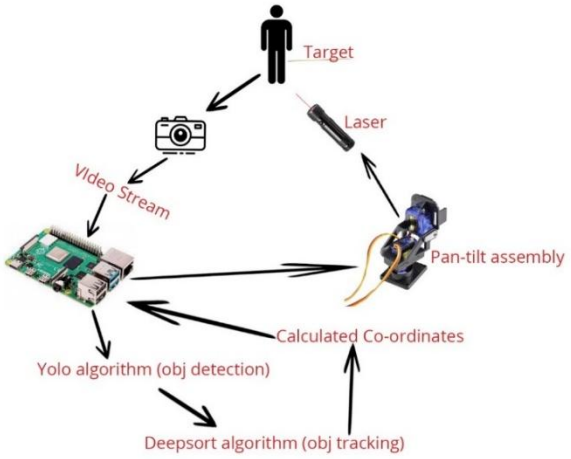

Fig. 3. Diagram of the Mechanism of the RWS

When the camera detects a target using the custom trained model connected to the Raspberry Pi 4, the pan and tilt mechanism, representing elevation and azimuth movement of the RWS, tracks the target in both directions. The weapon's movements in two axes are controlled by servo motors with feedback. The system's motion operates in a closed loop. The deep learning algorithm operates on a designated single-board computer (SBC) that connects to a camera. The camera captures live video and transmits it to the processor for processing (running the deep learning model). The centroid of the objective transmits to the SBC's input and output pins using the serial or i2c protocol. The motion control unit utilizes the transmitted data to actuate the system. The motion control unit is equipped with a 32-bit ARM processor at its core that is highly modular for future upgrades to both the CV unit and motion control unit. The system can remain stationary while the pan-tilt assembly moves to detect and point at the object. The pan tilt assembly is mounted with a laser that rotates in both the x and y axis to aim at the target.

The servo responsible for moving the assembly on the x-axis holds the weight of the servo responsible for handling movement on the y-axis. The servo responsible for moving the assembly on the x-axis holds the weight of the servo responsible for handling movement on the y-axis. The y-axis servo is equipped with a laser pointer that targets the desired location. The assembly requires a voltage of five volts, which can be supplied through either a computer-connected Raspberry Pi or an AC to DC step-down transformer charger connected directly to the main power supply. In addition, a breadboard is necessary to establish the necessary connections from the servo motors to the Raspberry Pi. The RWS can be mounted on a mobile vehicle or situated in the region of interest for the users.

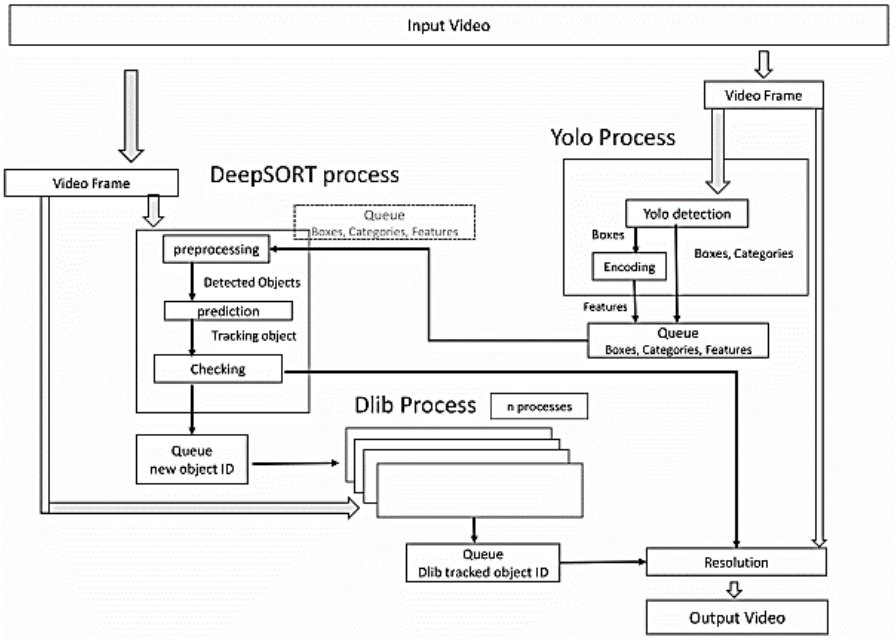

Fig. 4. Proposed Architecture by [1]

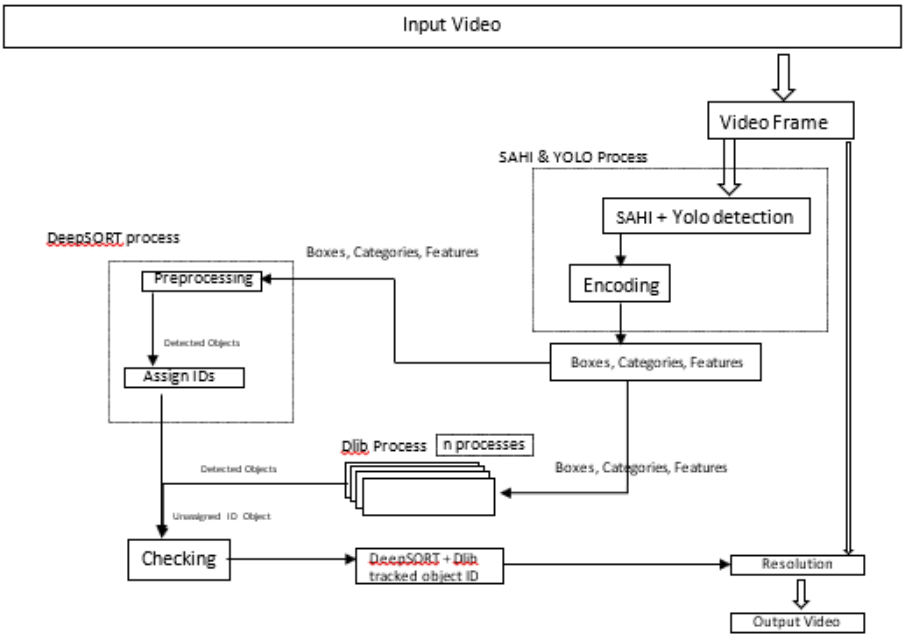

Fig. 5. Improved Object Tracking Model

3.2. Improved Object Tracking Model

The architecture proposed in Figure 5 is an improvement over the architecture presented in Figure 4 of [1]. First, the input video is read by the video reader of the OpenCV library and then transformed into frames. Next, the selected video frames are fed into both the detection model (YOLOv5) and Slicing Aided Hyper Inference (SAHI). SAHI performs sliced inference on the video frames [41]. SAHI was integrated with YOLOv5, the existing detection

algorithm, to improve inference. The SAHI inference tool enhances the detection of custom classes by identifying smaller objects, improving the overall detection process [40]. The SAHI-detected classes, along with their bounding box details (encoding), are forwarded to DeepSORT and the Dlib tracker for tracking. ID is assigned to each detected object by the DeepSORT tracker prior to tracking initiation. The Dlib algorithms review the identified and tracked entities. In instances where DeepSORT does not assign an ID to a detected entity, the Dlib tracker steps in and commences tracking. By doing so, the Dlib tracker boosts accuracy in tracking while also reducing IDs in the tracking process. Overall, this research proposes a streamlined tracking process that eliminates some processes known to significantly reduce the efficiency of the architecture.

4. RESULTS AND ANALYSIS

The performance metrics used in this research are Speed, Multiple Object Tracking Accuracy (MOTA), number of occurrences of Identity Switches (IDs), Precision, Recall, mAP. The speed of both the object detection and tracking is measured in Frames per Second (FPS). MOTA is used to record the accuracy of the model while IDs is used to measure the rate at which a detected object is lost or reassigned during tracking.

4.1. Experimental Results

All experiments were conducted on an HP Probook 6570b, utilizing the Google Chrome browser to access Google Colab for running both the training and testing phases of the proposed custom model. The results obtained from the training and testing phases were saved on Google Drive and can be further utilized. The proposed architecture was tested using two different video files and compared with a similar architecture devised previously in [1]. The video file used contained pedestrians on the road, which was trimmed to 10 seconds for the experiment. The custom model employs a 0.2 confidence threshold and an image size of 416 x 416, which matches the size used during training. In order to enhance the detection abilities in YOLOv5, the SAHI inference model was contrasted with the original YOLOv5 detection. The experiment performed on the COCO dataset demonstrated advancements in metrics, including precision, recall, accuracy, and speed. The COCO validation dataset encompasses 80 classes that have been annotated with 5000 images. The validation dataset was predicted using both the model in [1] and the proposed model. The custom dataset yielded the following performance results: precision score of 84.2%, recall value of 76.1%, and mAP@0.5 of 82.7% for all trained classes. The same dataset was then validated with the following performance results: precision score of 62.2%, recall value of 54.9%, and mAP@0.5 of 56.1%. The results of the training, testing, and validation are displayed in Tables 1, 2, and 3 respectively.

Table 1. Performance Result of Training of the Custom Model

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
All	5292	12858	0.842	0.761	0.827	0.587
Handgun	5292	2479	0.915	0.953	0.974	0.748
Knife	5292	504	0.858	0.923	0.952	0.711
Person	5292	8088	0.78	0.571	0.676	0.342
Rifle	5292	1787	0.818	0.595	0.703	0.348

Table 2. Performance Result of Testing of the Custom Model

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
All	1767	4259	0.626	0.534	0.553	0.342
Handgun	1767	837	0.819	0.785	0.829	0.599
Knife	1767	181	0.716	0.695	0.74	0.488
Person	1767	2610	0.511	0.41	0.398	0.181
Rifle	1767	631	0.458	0.247	0.242	0.101

Table 3. Performance Result of Validation of the Custom Model

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
All	1764	4444	0.622	0.549	0.561	0.333
Handgun	1764	843	0.832	0.796	0.861	0.616
Knife	1764	165	0.632	0.685	0.697	0.43
Person	1764	2847	0.543	0.377	0.377	0.164
Rifle	1764	589	0.482	0.339	0.309	0.12

4.1.1. Operating Speed

The tracking scenarios' operating speed improved after implementing changes in the model, as demonstrated in Table 4. Upon running a video file featuring a pedestrian on the road, DeepSORT and YOLOv5 yielded the following results: 0.4fps pre-processing speed, 27.5fps inference speed, 1.7fps Non-Maximum Suspension (NMS) speed, and 73.7fps DeepSORT update on 60 out of 120 frames. When we used YOLOv3 as the detector with DeepSORT tracker on the same file, we achieved a reduction in inference speed of 20.1fps and 60.1fps DeepSORT update. We added Dlib tracker to the YOLOv3 and DeepSORT (to give [1] model) and attained 0.5fps pre-process speed, 27.4fps inference speed, 2.0fps NMS speed and 70.2 DeepSORT update speed. On testing the proposed model, we achieved 0.7fps pre-process speed, 29.0 inference speed, 2.4fps NMS speed, and 77.5fps speed for DeepSORT update, as illustrated in Table 4. In experimentation with the COCO dataset, the proposed model achieved an overall inference speed of 314.8fps compared to Dang's 218.6fps. This indicates a notable advancement in operational speed as each operation saw an increase. Table 4 displays a comparison of operation speed results obtained through changes in the model, which resulted in scenarios of tracking to exhibit improvement in the proposed model.

Table 4. Result of the Speed of Operation

Models	Video (Pedestrian)			COCO Dataset (fps)
	Pre-process (fps)	Inference (fps)	DeepSORT update (fps)	
YOLOv3 + DeepSORT	0.4	20.1	60.1	-
YOLOv5 + DeepSORT	0.4	27.5	73.7	-
Dang et al. (2020)	0.5	27.4	70.2	218.6
Improved Object Tracking Model	0.7	29.0	77.3	314.8

4.1.2. Multiple Object Tracking Accuracy

Using pymot evaluation tool, the MOTA was deduced using the ground truth data and predicted data. The MOTA for the proposed custom model is 0.86 compared to 0.83 of [1] proposed architecture for video of pedestrian. The accuracy was improved on as a result of the use of SAHI inference with YOLOv5 as shown in Table 5. The MOTA closely represents human visual assessment and it is one of the most widely used metric for MOT. The MOTA is done at the detection level where association is measured using IDs. The MOTA measures three types of tracking errors which are False Positive, False Negative and IDs.

$$MOTA = 1 - \frac{|FN|+|FP|+|IDs|}{|gtDet|} \quad (2)$$

Where *FN = False Negative*

FP = False Positive

IDs = Identity Switch and

gtDet = Groundtruth Object Count.

Average Precision (AP) serves as a measure to evaluate the performance of object detectors. It is a single number metric that encapsulates both precision and recall curve by averaging precision across recall values from 0 to 1. The mAP in essence averages AP over the number of classes in the dataset. It also uses an IOU threshold usually at 0.5

to achieve its value. The evaluation using COCO dataset deduced the results showing the overall improvement in terms of detection metrics such as Precision (P), Recall (R), mAP and MOTA.

Table 5. Result of MOTA using pymot and pycocotool

Author(s)	Video	COCO Dataset				
	MOTA	P	R	mAP@.5	mAP@.95	MOTA
Dang et al. (2020)	83%	0.67	0.51	0.56	0.37	56%
Improved Object Tracking Model	86%	0.98	0.81	0.56	0.53	79%

4.1.3. Number of Identity Switches

An IDs is a True Positive (TP) which has a predicted ID that is different from the predicted ID of the previous TP (that has the same groundtruth ID). IDs only measure association errors compared to the single previous TP, and does not count errors where the same predicted ID swaps to a different groundtruth ID (ID Transfer) [43]. As incorporated as MOTA as CLEARMOT evaluation metrics, MOTA measures three types of tracking errors which are IDs association errors, False Negatives and False Positives detections errors.

$$IDs = \sum_t ids_{i,t} \tag{3}$$

Table 6 shows the result of the number of the occurrences of IDs. The proposed simplified architecture got similar result when compared to the proposed model of Dang et al. [1] architecture in terms of number of IDs. Both DeepSORT and Dlib were utilized by both models, there were similar results in the aspect of tracking the detected objects. The proposed improved model is able to detect more objects, most especially smaller objects due to the addition of SAHI to the proposed model. Using TrackEval [43], a tool for MOT evaluation, the following results were obtained as shown in Table 6. Using tracking sample videos from the MOT Challenge dataset, a significant reduction in the number of Identity switches occurrence using the proposed (YOLOv5-SAHI-Dlib-DeepSORT) model was achieved.

Table 6. Result of the Occurrence of IDs.

Author(s)	MOT17-02-FRCNN(IDs)	MOT17-04-FRCNN(IDs)	Overall IDs
Dang et al. (2020)	1021	1267	2288
Improved Object Tracking Model	894	987	1881

The Overall IDs from tracking with the two samples of MOT17 dataset significantly reduced with the use of the Improved Object Tracking Model proposed in this study compared with [1] tracking model.

5. CONCLUSION

This paper introduces an enhanced object tracking model designed for Remote Weapon Stations (RWS). The proposed architecture exhibits superior performance in terms of detection and tracking accuracy, reduction in identity (ID) occurrences, and increased operating speed compared to the architecture proposed by Dang et al. Through experimentation, it was established that the custom model surpasses the selected state-of-the-art model in detection accuracy, operational speed, and the occurrence of IDs. Notably, the custom model demonstrated a faster operating speed in contrast to the benchmark technique. The custom model achieved a Multiple Object Tracking Accuracy (MOTA) of 0.86, outperforming the benchmark's 0.83. Additionally, the custom model identified 1881 objects, while the benchmark identified only 2288. The implementation of the proposed architecture is credited for these improvements, making the custom model a potent tool for enhancing security systems. The suggested model has been specifically tested in a RWS context, making it particularly suitable for security applications. Future research avenues may involve deploying the detector-tracker architecture with alternative datasets to construct

models applicable in diverse fields, including traffic control, agriculture, medical operations, human counting, and education.

REFERENCES

[1] Dang, T. L., Nguyen, G. T., & Cao, T. (2020). Object tracking using improved deep SORT YOLOv3 architecture. ICIC Express Letters, 14(10), 961-969.

[2] Kothiya, S. V., & Mistree, K. B. (2015). A review on real time object tracking in video sequences. 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO). Visakhapatnam. https://doi.org/10.1109/EESCO.2015.7253705

[3] Vishwakarma, A., & Khare, A. (2018). Vehicle detection and tracking for traffic surveillance applications: A review paper. https://doi.org/10.26438/ijcse/v6i7.721724

[4] Sachan, A. (2019). Zero to hero: A quick guide to object tracking: Mdnet, goturn, rolo. CV-Tricks. https://cv-tricks.com/object-tracking/quick-guide-mdnet-goturn-rolo/

[5] Taguri, Y., Erlichmen, S., & Lussato, R. (2015). Object Tracking in Deep Learning - MissingLink.ai. https://missinglink.ai/guides/computer-vision/object-tracking-deep-learning/

[6] Sagar, R. (2019). How the deep learning approach for object detection evolved over the years. Retrieved 1 January 2024, from Analytics India Magazine website: https://analyticsindiamag.com/how-the-deep-learning-approach-for-object-detection-evolved-over-the-years/

[7] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. International Journal of Computer Vision, 128(2), 261-318. https://doi.org/10.1007/s11263-019-01247-4

[8] Boulanin V., Verbruggen M. (2017). SIPRI Mapping the development of autonomy in weapon systems. Boulanin, M. Verbruggen, SIPRI, Solna: SIPRI.

[9] Nolan, C. J. (2017). The allure of battle: A history of how wars have been won and lost. Oxford University Press.

[10] Mohamed, M., Jens, L., Sören, A., Claus, S., & Sebastian, H. (2012). About: Remote controlled weapon station. DBpedia. https://dbpedia.org/page/Remote_controlled_weapon_station

[11] Melanie, S. (2016). The Inevitable Militarization of Artificial Intelligence. In Cyber Defense Review.

[12] Rebello, L. (2018). Autonomous Targeting System using Open CV. International Journal for Research in Applied Science and Engineering Technology, 6(3), 2545-2549. https://doi.org/10.22214/ijraset.2018.3412

[13] Liang, Q., Wu, W., Yang, Y., Zhang, R., Peng, Y., & Xu, M. (2020). Multi-player tracking for multi-view sports videos with improved k-shortest path algorithm. Applied Sciences, 10(3), 864. https://doi.org/10.3390/app10030864

[14] Nyström, A. (2019). Evaluation of Multiple Object Tracking in Surveillance Video.

[15] Xu, Y., & Wang, J. (2019). A unified neural network for object detection, multiple object tracking and vehicle re-identification. arXiv preprint arXiv:1907.03465.

[16] Li, W., Mu, J., & Liu, G. (2019). Multiple object tracking with motion and appearance cues. In Proceedings

of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0-0). https://doi.org/10.1109/ICCVW.2019.00025

[17] Zhang, X., Hao, X., Liu, S., Wang, J., Xu, J., & Hu, J. (2019). Multi-target tracking of surveillance video with differential YOLO and DeepSort. In X. Jiang & J.-N. Hwang (Eds.), Eleventh International Conference on Digital Image Processing (ICDIP 2019). https://doi.org/10.1117/12.2540269

[18] Mohana, H. V., & Ravish, A. (2019). Object Detection and Classification Algorithms using Deep Learning for Video Surveillance Applications. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(8), 386-395.

[19] Mandal, V., & Adu-Gyamfi, Y. (2020). Object detection and tracking algorithms for vehicle counting: a comparative analysis. Journal of Big Data Analytics in Transportation, 2(3), 251-261. https://doi.org/10.1007/s42421-020-00025-w

[20] Santos, A. M., Bastos-Filho, C. J. A., Maciel, A. M. A., & Lima, E. (2020, November). Counting vehicle with high-precision in Brazilian roads using YOLOv3 and deep SORT. 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). Porto de Galinhas, Brazil. doi:

[21] Santos, A. M., Bastos-Filho, C. J., Maciel, A. M., & Lima, E. (2020). Counting vehicle with high-precision in brazilian roads using yolov3 and deep sort. In 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 69-76). IEEE. https://doi.org/10.1109/sibgrapi51738.2020.00018

[22] Punn, N. S., Sonbhadra, S. K., Agarwal, S., & Rai, G. (2020). Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques. Retrieved from http://arxiv.org/abs/2005.01385

[23] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. International Journal of Computer Vision, 128(2), 261-318. https://doi.org/10.1007/s11263-019-01247-4

[24] Tran, V. H., Dang, L. H. H., Nguyen, C. N., Le, N. H. L., Bui, K. P., Dam, L. T., & Huynh, D. H. (2021). Real-time and robust system for counting movement-specific vehicle at crowded intersections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4228-4235). https://doi.org/10.1109/CVPRW53098.2021.00478

[25] Duan, C., & Li, X. (2021). Multi-target tracking based on deep sort in traffic scene. Journal of Physics. Conference Series, 1952(2), 022074. https://doi.org/10.1088/1742-6596/1952/2/022074

[26] Shukla, R. I. T. I. K., Mahapatra, A. K., & Selvin Paul Peter, J. (2021). Social distancing tracker using yolo v5. Turkish Journal of Physiotherapy and Rehabilitation, 1785-1793.

[27] Meimetis, D., Daramouskas, I., Perikos, I., & Hatzilygeroudis, I. (2023). Real-time multiple object tracking using deep learning methods. Neural Computing & Applications, 35(1), 89-118. https://doi.org/10.1007/s00521-021-06391-y

[28] Pramanik, A., Pal, S. K., Maiti, J., & Mitra, P. (2022). Granulated RCNN and multi-class deep SORT for multi-object detection and tracking. IEEE Transactions on Emerging Topics in Computational Intelligence, 6(1), 171-181. https://doi.org/10.1109/TETCI.2020.3041019

[29] Wu, P., Xu, H., Ding, Y., Wang, Z., & Zhang, J. (2021). An improved online multiple object tracking algorithm based on KFHT motion compensation model in the aerial videos. In Seventh Symposium on Novel Photoelectronic Detection Technology and Applications (Vol. 11763, pp. 2431-2436). SPIE. https://doi.org/10.1117/12.2587667

[30] Gao, G., & Lee, S. (2021). Design and Implementation of Fire Detection System Using New Model Mixing. International Journal Advanced Culture Technology, 9(4), 260-267.

[31] Lewert, J. (2021). Human Detection for Flood Rescue: Application of YOLOv5 Algorithm and DeepSORT Object Tracking (Doctoral dissertation).

[32] Francies, M. L., Ata, M. M., & Mohamed, M. A. (2022). A robust multiclass 3D object recognition based on modern YOLO deep learning algorithms. Concurrency and Computation: Practice & Experience, 34(1). https://doi.org/10.1002/cpe.6517.

[33] Neethirajan, S. (2022). ChickTrack-A quantitative tracking tool for measuring chicken activity. https://doi.org/10.36227/techrxiv.15031440

[34] Shoman, M., Aboah, A., Morehead, A., Duan, Y., Daud, A., & Adu-Gyamfi, Y. (2022, June). A region-based deep learning approach to automated retail checkout. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). New Orleans, LA, USA. https://doi.org/10.1109/cvprw56347.2022.00362.

[35] Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. Sensors (Switzerland), 22(2), 464. https://doi.org/10.3390/s22020464

[36] Patel, K., Bhatt, C., & Mazzeo, P. L. (2022). Deep learning-based automatic detection of ships: An experimental study using satellite images. Journal of Imaging, 8(7), 182. https://doi.org/10.3390/jimaging8070182

[37] Ye, K., Dong, J., & Zhang, L. (2022). Digital analysis of movements on characters based on OpenPose and Dlib from video. Journal of Physics. Conference Series, 2218(1), 012021. https://doi.org/10.1088/1742-6596/2218/1/012021

[38] Schmidt, J., Marques, M. R. G., Botti, S., & Marques, M. A. L. (2019). Recent advances and applications of machine learning in solid-state materials science. Npj Computational Materials, 5(1). https://doi.org/10.1038/s41524-019-0221-0

[39] Abbasi, M., Shahraki, A., & Taherkordi, A. (2021). Deep learning for network traffic monitoring and analysis (NTMA): A survey. Computer Communications, 170, 19-41. https://doi.org/10.1016/j.comcom.2021.01.021

[40] Roboflow (n.d). Roboflow Public Dataset (n.d). Public Dataset of Pistols. Retrieved from https://public.roboflow.com/object-detection/pistols

[41] Google Open Images. (n.d.). Google Open Images Dataset of Person, Handgun, Rifle and Knife. Retrieved from https://storage.googleapis.com/openimages/ web/visualizer/index.html.

[42] Akyon, F. C., Altinuc, S. O., & Temizel, A. (2022). Slicing aided hyper inference and fine-tuning for small object detection. In 2022 IEEE International Conference on Image Processing (ICIP) (pp. 966-970). IEEE. https://doi.org/10.1109/ICIP46576.2022.9897990

[43] Galanty, A., Danel, T., Węgrzyn, M., Podolak, I., & Podolak, I. (2021). Deep convolutional neural network for preliminary in-field classification of lichen species. biosystems engineering, 204, 15-25. https://doi.org/10.1016/j.biosystemseng.2021.01.004

[44] Luiten, J., Os Ep, A. A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., & Leibe, B. (2021). HOTA: A higher order metric for evaluating multi-object tracking. International Journal of Computer Vision, 129(2), 548–578. https://doi.org/10.1007/s11263-020-01375-2.