Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/36/20230434

Evolution of CNNs in human tracking applications

Yancheng Chen

The High School Affiliated to Remin University of China, Beijing, 100097, China

yancheng@iplab.cn

Abstract. Tracking the movement of objects in videos or footage from CCTV systems plays an integral role in crime investigations, surveillance, security predictions, and many other domains. Historically, this task was primarily entrusted to dedicated observers or analysts who would be summoned to review pre-recorded footage post-event. With the advent of machine learning and AI, convolutional neural networks (CNNs) have paved the way for computers to augment human capabilities in analyzing streaming videos or archived recordings. Among the various tracking methodologies that machine learning offers, YOLO (You Only Look Once) and R-CNN (Region-based Convolutional Neural Networks), along with its iterations, stand out as some of the most reliable and precise. However, the scope of analysis often extends beyond these techniques. To enhance accuracy and provide an adept classification mechanism, the Deep SORT (Simple Online and Realtime Tracking) algorithm emerges as pivotal. Its synergy with human detection remains a significant area of discussion and will be deliberated upon in this study. This review aims to elucidate the intricacies of these state-of-the-art methods and their interplay in modern tracking systems.

Keywords: deep learning, human tracking, YOLO, RCNN, Deep-SORT.

1. Introduction

YOLO is a groundbreaking detection algorithm grounded in convolutional network systems. The original YOLO model comprised 24 convolutional layers, succeeded by two fully connected layers [1]. What sets YOLO apart is its remarkable processing speed and its prowess in detecting smaller objects. Furthermore, YOLO has the ability to learn and transfer generalized features, making it a prime candidate for transfer learning, which can further enhance detection accuracy. This article will primarily delve into the intricacies of YOLO v3, with later versions touched upon in the conclusion. On the flip side, R-CNN has evolved into two significant variations: Fast R-CNN and Faster R-CNN. Both variants root their mechanisms in the concept of "Regional Proposal," which focuses on pooling features from individual frames within a video. Specifically, Fast R-CNN employs a method known as selective search. To boost efficiency, Faster R-CNN substitutes the selective search process with the RPN (Regional Proposal Network), a comprehensive convolutional neural network [2]. This article will also shed light on a proposed modification of R-CNN, emphasizing its potential in vehicle tracking and speculating its applicability in human tracking. A common challenge in object detection within streamed or recorded videos is occlusion. During instances of occlusion, many detection algorithms risk losing track of objects. Enter the Deep Sort algorithm, which offers a solution by predicting an object's trajectory during

DOI: 10.54254/2755-2721/36/20230434

occlusions. We will also touch upon the pivotal role of Deep Sort in tracking individuals in areas with heavy pedestrian traffic.

2. Detection algorithms

2.1. FAST R-CNN & FASTER R-CNN

R-CNN, introduced by Ross Girshick et al. in 2013, was developed to showcase the superiority of ConvNet-based object detection over blockwise-oriented histograms like SIFT and HOG [3]. While it maintained the traditional sliding window approach, it expanded the original two pooling convolution layers to five. The receptive fields and strides were also upscaled to 195×195 and 32×32 respectively. In testing, the algorithm impressively detected and classified humans, achieving a mean average precision (mAP) of 53.7 on PASCAL VOC 2010 and 31.4 on ILSVRC2023. The subsequent Fast R-CNN and Faster R-CNN, unveiled in 2015, were advancements of the original R-CNN [4]. These models, especially Faster R-CNN, were tweaked for enhanced speed and precision in object detection. Zhiqiang Li et al.'s 2016 proposal stood out, concentrating on human detection by emphasizing head and shoulder features to account for possible occlusions [5]. Their approach involved modifying parameters in Girshick's model, employing stochastic gradient descent for training the RPN, and incorporating online hard example mining (OHEM) to heighten accuracy. The results, especially when combined with kernel correlation filters, were promising.

However, in 2020, Vignesh Kanna, J.S. et al. pivoted towards a unique objective: tracking individual suspects in crowded scenarios [6]. Their approach veered from the norm by focusing on specific features like gender, shirt pattern, and spectacle status rather than the typical facial markers. They harnessed the power of linear SVMs and introduced the Soft max regression function to optimize bounding boxes for both speed and accuracy, achieving an impressive 87% detection rate.The momentum in the development of R-CNN derivatives didn't stop there. Leveraging modern GPU capabilities, two-stage detectors emerged, boasting enhanced object localization and classification [7]. One striking innovation was the Granulated R-CNN or G-RCNN, formulated by Anima Parmanik et al. It was designed to address Faster R-CNN's shortcomings in processing temporal data. The G-RCNN brought to the table a mechanism that estimated foreground regions with higher object occurrence probability, refining resource allocation for image analysis. Notably, Parmanik's team employed irregular-shaped granules in their pooling feature map to better localize objects within video frames.

But the G-RCNN's standout feature was its ability to handle objects of varied sizes. While the Faster R-CNN+MCD-SORT produced remarkable multi-object tracking accuracy (MOTA) and precision (MOTP) metrics, G-RCNN surpassed these benchmarks, especially when tracking smaller objects [8, 9]. This addressed a significant shortcoming observed in other algorithms like YOLO. In conclusion, the advancements made from the original R-CNN to the more refined versions like G-RCNN demonstrate the significant strides in the field of object detection and tracking. As technology continues to evolve, it's exciting to ponder where the next breakthrough will emerge.

2.2. The YOLO algorithm

Proposed by Joseph Redmon et al. in 2015, shortly after the introduction of R-CNN, the YOLO algorithm rapidly emerged as a state-of-the-art object detection system. From 2015 to 2023, eight distinct YOLO versions were developed, each catering to various applications. Among them, three versions – YOLO v3, YOLO v5, and YOLO v8 – were primarily tailored for object detection.

To offer an alternative to R-CNN's intricate training process and complex feature processing mechanism, Redmon aimed to reconceptualize object detection as a singular regression problem. As he framed it, bounding boxes should derive directly from image pixels. To achieve this, he introduced a unified convolutional network that seamlessly combines the processes of detection and classification. The maiden YOLO model was already a feat. Running on a Titan X GPU, it could clock in at 45 fps, and even soared to 150 fps when using a more streamlined version [10]. The distinction between YOLO and R-CNN, and other traditional CNNs, is pronounced: YOLO processes the entire image in one go

DOI: 10.54254/2755-2721/36/20230434

rather than relying on sliding window detectors. This means it isn't swayed by background patches or shapes that R-CNN might mistakenly factor in. Thanks to this unique approach, YOLO can process images much faster than even Faster R-CNN. However, it comes with the caveat that YOLO can sometimes falter in detecting and pinpointing smaller objects. The foundational YOLO framework comprised 24 convolutional layers, capped off with two reduction sibling layers – one measuring 1×1 and the other 3×3. Beyond this main model, a streamlined version was also rolled out, encompassing just nine layers with a leaner filter set. The overarching goal was to filter the image through an array of convolution layers of varying dimensions, isolating features in the downscaled output. Initially, the YOLO model was calibrated and evaluated on the VOC 2007 and VOC 2012 datasets. Its performance was commendable, especially given the GPU constraints of 2016. Under real-time conditions (45 fps) across both datasets, YOLO achieved a mAP of 63.4 and a stellar 52.7 mAP at an impressive 155fps. For comparison, the DPM algorithm, with operation speeds of 100Hz and 30Hz, lagged considerably. In a sub-real-time context, YOLO VGG-16 trailed behind Faster R-CNN with a 66.4 mAP, compared to Faster R-CNN VGG-16's 73.2 mAP. Yet, when comparing processing speeds, YOLO VGG-16's metrics at 21 fps clearly outperformed Faster R-CNN VGG-16's 7fps. When aligning their processing speeds, Faster R-CNN ZF, operating at 18fps, recorded a mAP of 62.1 – still short of YOLO VGG-16's 66.4 mAP. Redmon even postulated a fusion of Fast R-CNN with YOLO to amalgamate R-CNN's precision with YOLO's rapidity and selective processing. In this hybrid model, Fast R-CNN would spearhead detections, while YOLO would refine and possibly discard any spurious or redundant features. This integrated model yielded a mAP of 70.7 on the VOC 2012 test, while standalone YOLO managed a mAP of 57.9. Regrettably, there's no mention of the fps under which this hybrid system was tested, so any potential speed-to-mAP trade-offs remain unknown.

The standout among the trio of aforementioned YOLO versions is YOLO v3. This iteration brought on board an integrated feature extractor with 53 convolutional layers and included shortcuts to bolster processing speeds. When refining this version, Redmon and his team toyed with four experimental integrations, all of which were eventually shelved: Predicting the x, y offset of anchor boxes; Linear x, y predictions as a substitute for logistics; Focal loss; Dual IOU thresholds and truth assignment.

While YOLO v3 might not have clinched the top spot in terms of accuracy, it preserved YOLO's signature trait – speed. YOLOv3-608 clocked an inference speed of 51ms, and a mAP-50 of 57.9. Under equivalent conditions, FPN FRCN was the sole contender outpacing YOLO v3 with a mAP-50 of 59.1. However, its inference time of 172ms was over three times that of YOLOv3-608. Leveraging its capabilities, YOLO v3 became the bedrock for numerous human tracking systems. In 2020, Imran Ahmed and team introduced a top-down multi-human tracking technique harnessing YOLO v3 combined with transfer learning and Deep SORT. Even though subsequent YOLO versions had emerged by then, YOLO v3 was chosen for its prowess in generic object detection. Given the overhead perspective of their videos, the algorithm primarily honed in on individual heads. Once YOLO v3 identified a person, a Kalman filter with linear observation and constant velocity was deployed. Pathways of moving individuals were projected across successive frames. Features were extracted via a dedicated CNN before transitioning to the Deep SORT phase. A threshold was set to vet these predicted paths, discarding any that didn't pass muster and approving those that did for immediate updates. This methodology's precision was almost unerring. When amalgamating YOLO v3 and Deep SORT, sans transfer learning, it achieved a MOTA of 92 and a MOTP of 90. With transfer learning in the mix, these metrics surged to a MOTA of 96 and a MOTP of 95.

Fast forward to recent developments, and we see eight more versions of the YOLO algorithm, with YOLO 8 NAS being both the swiftest and most precise. Capitalizing on cutting-edge hardware, YOLO 8 NAS emerges as the go-to for real-time object tracking, especially when high fps is non-negotiable. Deploying it in CCTV systems would allow for in-depth people flow analytics, as well as insights into demographics like gender, ethnicity, and age. Beyond these mainstream applications, YOLO's versatility extends to surveillance in challenging terrains like mountainous forests, open seas, flood zones, and more. Apart from real-time analysis, YOLO can retrospectively decipher the movement paths of individuals or objects in the lead-up to or aftermath of incidents.

DOI: 10.54254/2755-2721/36/20230434

2.3. Path prediction and SORT

Before the introduction of deep-SORT, traditional path prediction methods like Multiple Hypothesis Tracking (MHT) and Joint Probabilistic Data Association (JPDAF) were widely used. In MHT, every potential hypothesis was tracked, while JPDAF processed data association frame-by-frame. Both methods had a significant limitation – their computational and implementation complexity was high. Addressing these challenges, N. Wojke and colleagues introduced the Simple Online and Real-Time Tracking method in 2017. This approach leaned on the single hypothesis tracking strategy, incorporating recursive Kalman filtering, a technique previously discussed in the detection algorithm section. Additionally, the SORT method integrated the Hungarian method, specifically designed to measure bounding box overlaps. When put to the test in the MOT challenge, SORT demonstrated more identity switches but still retained solid overall performance. To enhance handling of occlusions, Wojke's team substituted the original association metric with a more refined metric that combined motion and appearance data. They also incorporated another convolutional neural network, trained meticulously to distinguish individual pedestrians using a large-scale dataset. Surprisingly, during tests, SORT actually outperformed deep SORT. With a MOTA of 59.8 and a MOTP of 79.6 operating at 60Hz, SORT stood slightly ahead of deep SORT, which achieved a MOTA of 61.4 and a MOTP of 79.1 at 40Hz. In situations of slower runtimes, where all algorithm runtimes did not exceed 3Hz, accuracy figures increased. The lowest was NOMTwSDP16, clocking in a MOTA of 62.2 at a 3Hz runtime. In contrast, KDNT achieved the highest, registering a MOTA of 68.2 and a MOTP of 79.4, though at a slower 0.7 Hz runtime. Nevertheless, it's worth noting that these slower runtime situations do not mimic real-time scenarios. As such, in real-world video or footage analysis, both SORT and deep SORT would likely outperform them.

3. Challenges

Faster R-CNN and Yolo 8 are undoubtedly at the forefront of CNN-based detection algorithms. Their prowess in image detection and classification has transformed various applications, from surveillance to autonomous vehicles. However, when paired with some of the most advanced deep learning GPUs in the market—like the pricey H100—the reality becomes evident: no solution is perfect, and cost-effective implementation is still a challenge. Both algorithms, though stemming from diverse architectural foundations, encounter unique setbacks. Faster R-CNN, while having pioneered a significant shift in object detection, still grapples with several issues. Its processing speed, especially in real-time scenarios, leaves much to be desired. Despite being a formidable tool, its accuracy now seems somewhat constrained when pitted against the likes of Yolo v8. Moreover, training Faster R-CNN is no small feat. The time and space complexity involved, not to mention the intricate setup required, can be prohibitive, especially for smaller institutions or individuals without extensive computational resources. Yolo, while excelling in speed and broader object detection, stumbles when it comes to the nitty-gritty details. One of its notable pitfalls is its struggle to accurately detect and localize smaller objects within an image. This limitation can be crucial in scenarios where every object, regardless of its size, is of paramount importance. Future enhancements in these models should holistically address these challenges. For Faster R-CNN, a primary focus should be on enhancing its processing speed without compromising its depth of analysis. Simplifying its training process, or perhaps devising methods to utilize pre-trained models more effectively, could make it more accessible and widely used.

4. Conclusion

The object detection algorithms, YOLO and R-CNN, have carved out unique niches in the domain of image and video analysis. Each is tailored to specific use cases, offering robust solutions based on the demands of different situations. In real-time detection and analysis scenarios, YOLO stands out due to its impressive processing speed, seamlessly identifying objects even when milliseconds count. Its ability to assess images holistically, rather than piece-by-piece, makes it particularly adept at these tasks. On the other hand, R-CNN excels in more granular, frame-by-frame analyses. For instance, in sports analytics, where the precise trajectory of a ball can be critical, R-CNN's detailed examination can

DOI: 10.54254/2755-2721/36/20230434

provide more accurate predictions. This nuanced breakdown is pivotal when every frame might hold the key to nuanced insights or predictions. Enter SORT, an algorithm designed for path prediction. It shines not just because of its high accuracy, but also its efficiency. Training the algorithm is relatively swift, and its computational footprint is modest. Its design facilitates straightforward implementation, which can be a boon for developers and institutions with varying levels of technical resources. However, it's essential to acknowledge that while these algorithms are impressive, they're not without challenges. Future work should focus on making these algorithms more versatile, ensuring consistent performance across a range of hardware setups. Especially for setups with limited GPU capabilities, it's crucial that these algorithms maintain their efficiency without compromising on accuracy. In the ever-evolving world of tech, the pursuit of optimal performance paired with broad accessibility will continue to drive innovations in this space.

References

[1] Faito J A, Steffens J, Steffens C, et al. development of a model with critical factors of success, predominant in implementation of a membrane system in the wastewater treatment-review of the case study of a dairy industry [J]. 2019.

[2] Ganesh S S, Abhilash S, Joseph S, et al. A review of the development and implementation of a tropical cyclone prediction system for North Indian Ocean in a multi-model ensemble framework [J]. Mausam, 2021, 72(1): 57-76.

[3] Xiaogeng Z, Yingxin L. A Review of the Development of Legal History as a Discipline in the Past 70 Years since 1949: Taking Renmin University as an Example [J]. Law and Modernization, 2019.

[4] Mingan J. A Review of the Development of Administrative Procedure Law in the 21st Century [J]. Journal of Comparative Law, 2019.

[5] Endacott J, Alam S. Mainstreaming displacement in development policies: An analysis of Solomon Islands and Vanuatu approaches [J]. Review of European, Comparative and International Environmental Law, 2022, 32(1): 136-148.

[6] Peuler M, Mccallister K C. Virtual and Valued: A Review of the Successes (and a Few Failures) of the Creation, Implementation, and Evaluation of an Inaugural Virtual Conference and Monthly Webinars [J]. Journal of library & information services for distance learning, 2019, 13(1-2): 104-114.

[7] Connor M, Conboy K, Dennehy D. Time is of the essence: a systematic literature review of temporality in?information systems development research [J].Information Technology & People, 2023, 36(3): 1200-1234.

[8] Sadri E, Harsej F, Hajiaghaei-Keshteli M, et al.Evaluation of the components of intelligence and greenness in Iranian ports based on network data envelopment analysis (DEA) approach [J].Journal of modelling in management, 2022.

[9] Kourtesis P, Collina S, Doumas L A A, et al.Technological Competence is a Precondition for Effective Implementation of Virtual Reality Head Mounted Displays in Human Neuroscience: A Technological Review and Meta-analysis [J]. 2021.

[10] Yadav D, Yadav S, Veer K. A comprehensive assessment of Brain Computer Interfaces: Recent trends and challenges[J]. Journal of Neuroscience Methods, 2020, 346: 108918.