Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20587-z

Enhancing hurdles athletes' performance analysis: A comparative study of cnn-based pose estimation frameworks

Pouya Jafarzadeh[1] · **Luca Zelioli**[1] · **Petra Virjonen**[1] · **Fahimeh Farahnakian**[1] · **Paavo Nevalainen**[1] · **Jukka Heikkonen**[1]

Received: 9 September 2024 / Revised: 11 December 2024 / Accepted: 26 December 2024

Abstract

Human pose estimation has gained significant attention in recent years for its potential to revolutionize athletic performance analysis, enhance understanding of player interactions, and optimize training regimes. Deep learning models, particularly Convolutional Neural Networks (CNNs), have outperformed traditional methods in pose estimation tasks. This study addresses a gap in sports analytics by applying two popular CNN-based frameworks, YOLO and DeepLabCut, to analyze pose estimation in hurdles athletes. Videos of a single female athlete during training sessions were used, and frames were manually annotated to capture three critical foot landmarks: ankle, heel, and big toe. The results highlight YOLOv8l's superior accuracy, achieving a Percentage of Correct Keypoints (PCK) of 79% for these landmarks, while demonstrating the feasibility of a low-cost setup for practical applications. Visual comparisons further validate the model's effectiveness in real-world scenarios. Additionally, YOLO predictions were utilized to analyze step progression in the time domain, providing actionable insights into athletic movement. This study underscores that even modest video equipment, combined with CNN-based methods, can equip coaches with powerful tools to analyze and optimize movements and techniques, paving the way for data-driven advancements in sports performance.

✉ Pouya Jafarzadeh
poujaf@utu.fi

Luca Zelioli
luca.l.zelioli@utu.fi

Petra Virjonen
pekavir@utu.fi

Fahimeh Farahnakian
fahfar@utu.fi

Paavo Nevalainen
ptneva@utu.fi

Jukka Heikkonen
jukhei@utu.fi

[1] Department of Computing, University of Turku, Turku 20014, Finland

Published online: 22 January 2025

Keywords Human pose estimation · Convolutional neural network · Deep learning · Hurdles training · Sports technology

1 Introduction

Sprint hurdling is a sport event requiring speed and an excellent sense of rhythm. The event consists of running 110 m/100 m with 10 hurdles (men/women, respectively). The athlete usually accelerates with 7 or 8 strides to the first hurdle and clears the following hurdles using four-step contacts between consecutive hurdles. The performance of an athlete can be analyzed by deriving relevant data from the runs to facilitate improvement. This requires analyzing the run phase by phase. Nowadays affordable cameras are available with rather high frame rates, and thus, detailed information can be extracted. Going through manually hundreds of frames is, however, a time-consuming task. Automatic pose estimation can help to find the joints of a human body, which makes it easy to calculate essential parameters, such as the step air and contact time. Visualizations of these parameters can assist coaches to evaluate athletes' performance during different stages of hurdling. Moreover, pose estimation can assist in designing specific training programs targeting the improvement of certain movements or muscle groups crucial for hurdling. It also can be used during competitions to analyze athletes' performances. For these reasons, pose estimation has been used in several sports such as triple and long jump [1], ski jumping [2], diving [3], rugby [4], bowling [5], swimming [6] and basketball [7]. Based on our review of existing literature, no prior research has been conducted specifically focusing on pose estimation for sprint hurdling. Our previous work [8] stands as a pioneering effort, being the first instance in which we applied OpenPose [9] technology in this sport event. This groundbreaking application of OpenPose in sprint hurdling represents a novel contribution to the field, addressing a previously unexplored area in athletic performance analysis.

The existing pose estimation approaches can be divided into two main groups: top-down and bottom-up. Top-down methods [10–13] first employ a person detector and then perform single-person pose estimation for each detection. Therefore, the complexity of these methods linearly increases with the number of persons in an image, and they are not applicable in real-time applications which are requiring constant runtime. In contrast, Bottom-up methods [9, 14, 15] offer faster runtime solutions since they rely on heatmaps for all key-points detection and then group them into individual persons. OpenPose is a widely used method that employs a combination of bottom-up and top-down strategies. It first detects keypoints using a bottom-up approach and then refines and associates keypoints to form human poses. It uses part affinity fields (PAFs) to link body parts, achieving accurate pose estimation. The top-down paradigm is more accurate but more costly due to an extra-person detection process, and the bottom-up paradigm is more efficient.

Deep Learning (DL) plays a critical role in pose estimation techniques as it able to automatically learn hierarchical representations from raw input data such as images or video frames. While traditional approaches need hand-crafted features like Harris Corner Detection [16], Scale-Invariant Feature Transform (SIFT) [17], or Speeded-Up Robust Features (SURF) [18]. In addition, they have low performance in handling occlusions, multi-person scenarios, and variations in poses and environments. One of the most common DL models for pose estimation is the Convolutional Neural Network (CNN). For instance, in [19], the authors utilized a CNN to iteratively refine the heatmap predictions of body joint locations. It consists of multiple stages, each refining the estimation of keypoints, and gradually improving

accuracy through multiple iterations. Another example is PoseNet [20] which is developed by Google for estimating the human pose in real-time using a single RGB camera. It uses a CNN to regress the 2D pose of a person from an input image.

This paper is a comparative study of two well-known CNN-based frameworks, YOLO [21] and DeepLabCut [22], subjected to hurdles athletes pose estimation. From these frameworks, we investigate the performance of 13 different CNN-based models for the pose estimation of hurdles athletes. For this purpose, a real dataset is collected from an athlete during her training. Compared with our previous work [8] for the same application, we only used 17 COCO dataset COCO dataset [23], in this paper, we manually annotated three main key points on the athlete's legs relevant to hurdle sport: right /left ankle, heel and big toe. These newly labeled points can provide insights into the athlete's technique, such as take-off and landing positions, stride length, or foot placement. Moreover, this information can be valuable for, biomechanists, or sports scientists aiming to analyze and improve athletic performance.

Each model is thought by manual digitization to follow the heel and the big toe of the athlete, and to register both the impact moments and the moment of separation. The timed events are then represented as a plot, where both the step positions and the contact duration are shown. Especially the preparation to hurl jump and recovery after the jump are visualized well. The resulting time series data allows further clusterization and comparison of several runners. An important aspect of the problem is that the contact event deform the feet and shoe while a holonomic rolling movement at the big toe makes the actual period of contact hard to define. We provide here a pipeline for registering the step length progression both in time and spatial domain and it is more sophisticated for analysing athlete's steps and performance. The experimental results indicate superior performance from YOLO compared to DeepLabCut models.

The remainder of this paper is organized as follows. Section 2 discusses the most important related research. We describe the data in Sect. 3. The proposed framework and methods are explained in Sect. 4. Section 5 and Sect. 6 provide the experimental designs and results. The discussion and conclusions are drawn in Sect. 7.

2 Related work

Hurdling is a demanding sport to analyze since the action is distributed over a relatively long distance, and it has several parts, which all have to be in delicate energy and momentum conservation balance. A good overview of the movement sequence of clearing a hurdle is in [24].

Gait analysis [25] in many different degrees of qualitative and quantitative forms have been a part of diagnostics in general therapeutic and medical practice. One of the early spring-mass skeletal models is in [26], where it is argued that although the walking activity is governed by the continuous muscular control and cannot be described well by spring-mass systems, running is dominated by linear elasticity.

Nowadays pose estimation in sport has been as area of growing interest to offer valuable insights for athlete performance analysis, training optimization, and sport science research. A rather generic human behavior registration capability of [27] uses YOLOV8 and the skeleton model includes the heel and the big toe. The application supports partially hidden bodyframe, e.g. upper torso visible over a table. Also, a good adaptation to the presence of obstructions and robustness under a wide variety of lighting conditions has been reached. Our videometrics approach requires somehow controlled and limited environment with some moderate

restrictions requiring a rather homogeneous background and suitable sportswear. A thorough summary of wearable devices including inertial sensors (inertial mass unit (IMU) and such, is [28]. A promising position of a coin sized IMU is on top of big toes, with wireless control unit at the belt.

Peak forward foot speed during the swing phase is approximately twice as fast as the running speed [29]. Video analysis of running has a basic problem of interpolating images: a 20 m/s forward movement of an ankle is captured by 120 fps video resulting a 17 cm movement between frames. A modern counter-move against this difficulty is flow-agnostic video representations, [30] which increase both accuracy and tolerance to changing lighting, background and sports gear variations.

There are some cases of DL-based architectures used in computer vision tasks, in which they can not show great performances. For instance, when these methods face digital images including objects of variable size and scale or when they have to perform visual task at pixel level. In these circumstances, DL-based techniques such as Faster-RCNN and YOLO are unsuccessful since visual features from small areas vanish during convolutional and pooling processes. One of solutions that is recently been used in to extract features in any resolutions and scales is to employ a CNN-based multi-scale network architectures where the information from high resolution feature maps (small scale) integrates with information from low resolution feature maps (large scale) by neural networks [19, 31]. In general, multi-scale network architectures are classified into three groups: multi-column network [32], skip-net [33] and multi-scale input [34]. In the multi-column network, input data are fed into various columns. The output data of each parallel column are then interconnected as the final output. The skip-net connects low-scale features with a large-scale output. Thus, features of distinct scales are composed and fed into an output layer. In the multi-scale input method, the input images are divided into several scales and then they feed into the network.

DeepPose [35] was an early CNN-based method for human pose estimation. It directly regresses joint positions from images using a deep convolutional network. Although relatively simpler compared to more recent architectures, it demonstrated the potential of CNNs for pose estimation. Stacked Hourglass Networks (SHNs) [11] utilize an encoder-decoder architecture stacked in multiple stages, where each stage predicts keypoint heatmaps at different resolutions. The network is designed to capture and refine spatial dependencies among body joints, enabling accurate pose estimation. In addition, Residual Networks (ResNets) [36] particularly deeper variants like ResNet-101 or ResNet-152, have been employed in pose estimation tasks as feature extractors or backbones due to their ability to capture hierarchical features. Several pose estimation frameworks or architectures [22] use MobileNet [37] as a backbone network due to its advantages in terms of efficiency and speed. By integrating MobileNet as a feature extractor, pose estimation models can benefit from its ability to process images efficiently while maintaining reasonable accuracy.

DL-based pose estimation finds various applications in sports. For example in [7], VGG19-based pose estimation can analyze a player's shooting form, tracking body alignment, arm angles, and body posture to provide feedback for improving shooting accuracy. In [6], they proposed CNN to automatically infer the body parts of swimmer which helps in evaluating swimming strokes, body positions, and movements to refine techniques and enhance swimming efficiency. AI-coach [2] utilizes CNN-based pose estimation to track the body movements, angles, and positions of ski jumpers during their jumps. DiveNet [3] has used a Temporal Convolution Network (TCN) over a backbone feature extractor to localize diving actions, with low latency. In [4], authors proposed Long Short-Term Memory (LSTM) to create a goal prediction model from goal-kick in rugby's videos to provide the player with

feedback. Bowlingdl [5] is a CNN-based method for bowling players' pose estimation and classification.

Our research aligns with these state-of-the-art methods, focusing on real-time applications in sports analytics, such as optimizing hurdling techniques and reducing injury risks. Similarly, advancements in other domains, such as smart farming, have demonstrated the utility of generative models like variational autoencoders (VAEs) and generative adversarial networks (GANs) in generating synthetic data for decision-making [38]. These parallels underline the transformative potential of data-driven approaches in real-time, low-cost, and scalable applications. By leveraging pose estimation models, coaches can gain actionable insights into athletes' performance, enhancing training efficacy and competitive readiness.

3 Dataset

The video samples were gathered in June 2018 in a sport field (Jyväskylä, Finland) during a training situation. One female athlete (100 m hurdles PB 12.81 s) attended the training. Three cameras (120 fps, 720 × 1920 or 1080 × 1920 pixels) were placed 11 ms from the edge of the chosen running lane. The first camera view included the third and the fourth hurdle, the second included the fourth hurdle and the third camera view included the fourth and the fifth hurdle. All the camera views captured six-step contacts. No other persons than the hurdler were present in the videos. The weather was half cloudy, so the lighting conditions changed only a little during the video shoot. We extracted images from 8 videos that totally generated 344 images with the size 1920 × 1080. Then, we manually annotated three main points of the athlete's feet including left/right ankle, heel, and big toe as shown in Fig. 1.

4 Methodology

4.1 Framework

The whole process of performing pose estimation for hurdles athlete is shown in Fig. 2. We first gathered the videos of an athlete while they were undergoing training sessions. The mobile phones were positioned on movable supports directly perpendicular to the run-

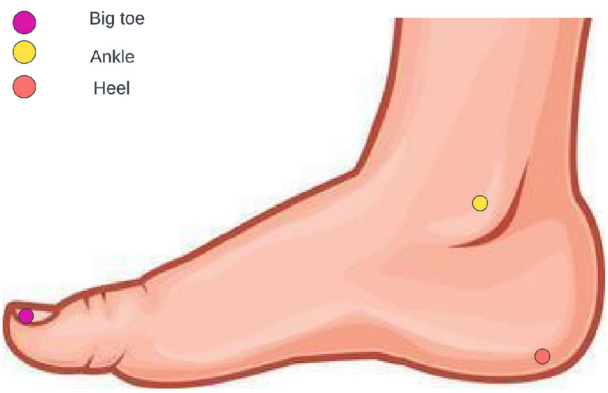

Fig. 1 The main annotation key-points of hurdles athlete's right and left feet in our dataset

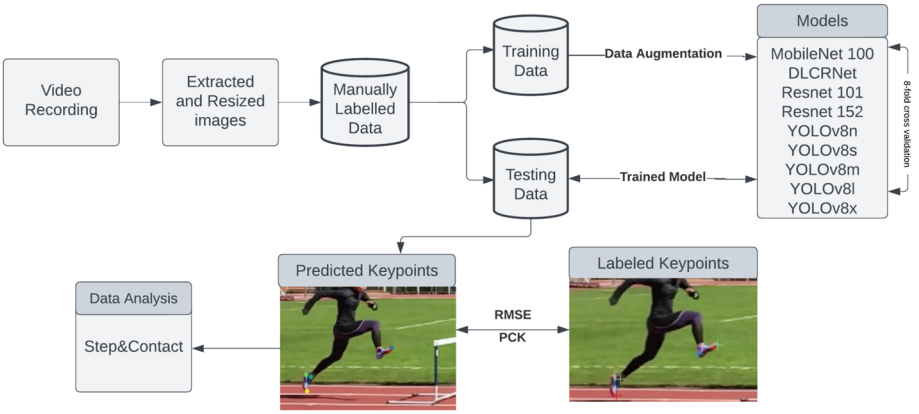

Fig. 2 Overview of the proposed framework for hurdles athlete pose estimation

ning track. The cameras were not calibrated but the central plane of the running track was measured and the mapping from pixel frame to horizontal and vertical was acquired via interpolation [39]. This mapping produced approximately 3 mm accurate locations and was subjected only to the output of the tracked points to keep the computations minimal.

The collected video material was used to extract frames and annotate specific key points including left/right ankle, heel, and big toe. These annotated key points are critical for understanding the biomechanics, movements, and feet positioning of the athlete during hurdling activities. The generated dataset is diverse enough to contain annotated images of athlete in various poses.

After that we divided the data into two training and validation datasets based on 8-fold cross validation. The training dataset was used to train the proposed DL models for pose estimation. The validation dataset contained a separate portion of annotated frames which were used for evaluating the trained model's performance.

To generate more data for training the model, avoid overfitting and improve its generalization ability, we performed data data augmentation. It involves applying various transformations or modifications to the existing annotated data to create new, synthetic data points while preserving the original labels or annotations. We will explain the proposed data augmentation techniques in Sect. 5.2. The final step involves evaluating these trained models using the test dataset to assess their performance and measure various evaluation metrics. Test dataset is an unseen video which is not used for training. In addition, the predicted key-points results are used for analysing the athletes movements and stepping trace.

4.2 DL-based pose estimation methods

In this sub-section, we reviewed two main DL-based methods which have been studied in this paper: YOLO and DeepLabCut. DLCRNet, ResNet and MobileNet are the proposed backbones which is used DeepLabCut toolbox [22] for pose estimation.

4.2.1 YOLO

You Only Look Once (YOLO) [21] is a popular CNN-based algorithm that is originally defined for object detection task in real-time. It divides the input image into a grid and directly

predicting bounding boxes and class probabilities for objects within each grid cell. Each cell is responsible for predicting objects if the center of an object falls within that cell. However, it can be adapted as part of a multi-task learning framework to perform pose estimation in conjunction with object detection. Over time, various versions and adaptations of YOLO (such as YOLOv2, YOLOv3, YOLOv4, etc.) have been developed, each with improvements in accuracy, speed, and architectural enhancements. In this work, we used YOLOv8 as it adjusts the output layer to predict keypoint coordinates instead of object bounding boxes and class probabilities for pose estimation.

4.2.2 ResNet

or Residual Network represents a supervised, feed-forward deep neural network architecture [36]. Unlike CNNs, ResNet introduces the concept of residual connections, where layers serve as learning activation functions that reference the input layers. This unique architecture enables the training of neural networks comprising a large number of layers, even extending to thousands, while maintaining a low percentage of training error. By leveraging residual connections, ResNet effectively addresses the vanishing gradient problem [40] commonly encountered during training with gradient-based learning methods. This innovation allows for more efficient and stable training, leading to improved performance in various tasks, including image classification, object detection, and human pose estimation.

4.2.3 DeepLabCut ResNet (DLCRNet)

is adapted ResNet model for improving pose estimation task by considering a multi-scale input architecture [22]. In this model, the input comprises a fusion of both low- and high-resolution feature maps. This unique feature enhances the ability of the DLCRNet method to mitigate missing keypoints, thereby facilitating the recognition of keypoints across different scale levels in the context of pose estimation systems. By incorporating information from multiple scales, DLCRNet improves the robustness and accuracy of the pose estimation process, particularly in scenarios involving variations in scale or perspective [19].

4.2.4 MobileNet

is a particular family of CNNs designed for deployment on mobile, internet-connected devices, and embedded systems to perform computer vision tasks such as object detection, face detection, and logo or text recognition [37]. In recent years, MobileNet has gained traction in the field of pose estimation, exemplified by the creation of models like the one proposed in [41], which is based on the MobileNet architecture.

The key advantage of MobileNet lies in its suitability for mobile applications, attributed to its compact model size (low number of parameters) and reduced complexity compared to other models. This is achieved by employing fewer multiplications and additions, resulting in improved accuracy, reduced memory footprint, and decreased computational time [42].

Structurally, MobileNet differs from traditional CNNs in its utilization of depthwise separable convolutions instead of standard convolution layers [42]. This architectural choice significantly reduces the number of parameters. In depthwise separable convolutions, the computation is divided into two stages: depthwise convolution and pointwise convolution. Firstly, a filter is applied to each input channel independently through depthwise convolution. Subsequently, pointwise convolution combines the outputs of the depthwise convolution by

applying a 1x1 convolution across all channels. Essentially, depthwise separable convolutions split the filtering and combining processes into two separate layers, distinguishing MobileNet from conventional CNN architectures.

5 Experimental setup

5.1 Model training

DeepLabCut toolbox (version 2.2.2) [22] has been used for body part pose estimation. DeepLabCut consists of a markerless technique that are invented to extract detailed human and animal poses without using any marker in locations with dynamic background. We evaluated DLCRNet with three backbones including ResNet152, ResNet101 and MobileNet100 which are included in DeepLabCut toolbox. In addition, YOLOv8 is evaluated with different scales (e.g., tiny, small, medium, large and xlarge) which are listed in Table 1. All models are pre-trained on COCO dataset.

We trained all networks with 10,000 training iterations with batch size 2. Adam [43] is used as an optimizer in all networks. To ensure the robustness and generalization of our models, we conducted thorough evaluations using an 8-fold cross-validation approach. This technique involved partitioning our dataset into 8 subsets, with each subset serving as a validation set once while the remaining data were used for training. By iteratively rotating through each subset as the validation set, we obtained 8 distinct evaluations for each model, providing a comprehensive assessment of its performance across different data partitions.

5.2 Effect of augmentation methods

We used image augmentation to generate more images in order to create a robust and accurate model. Augmentation is a promising solution for improving DL models' performance if the augmentation methods are chosen in a proper way and they do not affect the semantic information in the image [44]. We augmented the training dataset with two packages which is included in DeepLabCut including "Imagug" and "Tensorpack". ImgAug is a popular image augmentation library in Python, providing a wide range of augmentation techniques such as rotation, scaling, flipping, cropping, and color transformations. Tensorpack is a deep learning library built on top of TensorFlow, offering efficient data loading, preprocessing, and augmentation capabilities. To investigate the impact of different augmentation methods on the model performance, we assumed different argumentation techniques. In Table 2, we listed the augmentation methods and their parameters for that we set for both packages.

Table 1 Properties of Yolo version 8 models

Model	*Size (pixels)*	*Params (M)*
YOLOv8n	37.30	3.20
YOLOv8s	44.90	11.20
YOLOv8m	50.20	25.90
YOLOv8l	52.90	43.70
YOLOv8x	53.90	68.20

Table 2 The used augmentation methods and their values for both packages Tensorpack and Imgaug

Augmentation	Tensorpack	Imgaug
Crop size	(400,400)	(200,200)
Crop ratio	0.4	0.4
Rotation	5 degree	-
Rotation ratio	0.4	-
Contrast	[0.5, 2.0]	[0.4, 1.6]

5.3 Evaluation metrics

5.3.1 Root Mean Squared Error (RMSE)

We calculated the RMSE between the location of the predicted point and the reference point in pixels for each body part across all frames in the test dataset. The following formula is used to measure RMSE for each body part:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[(\hat{X}_i - X_i)^2 + (\hat{Y}_i - Y_i)^2\right]}$$

where N is the total number of frames. X_i and Y_i are the ground truth X and Y coordinates for the body part in frame i. $\hat{X}_i$ and $\hat{Y}_i$ are the predicted X and Y coordinates for a body part in frame i.

5.3.2 Percentage of correct keypoints (PCK)

measures the proportion of correctly localized keypoints within a certain threshold distance from the ground truth keypoints. The formula for calculating the total PCK for each body part is as follows:

$$PCK = \frac{\sum_{i=1}^{N} PCK_i}{M} \times 100$$

where M is the total number of keypoints evaluated. PCK_i is the PCK value for a body part in frame i. It represents the number of keypoints for which the absolute difference between the predicted joint and the ground truth joint is less than threshold T. For example, PCK@0.2 refers to the Percentage of Correct Keypoints (PCK) calculated with the threshold T set at 20% of a maximum allowable distance d. It means, if the distance between a predicted keypoint and its corresponding ground truth keypoint is less than 20% of the maximum allowable distance, it is considered correct. Otherwise, it is considered incorrect.

$$PCK_i@0.2 = |\hat{X}_i - X_i| + |\hat{Y}_i - Y_i| < 0.2 \times d$$

The maximum allowable distance should be proportional to the image size and the expected size of the athlete's body parts in the images. We also got help from domain experts to determine the maximum allowable distance for each foot body part. In Fig. 3, The shoe sizes of two proposed foot models (heel and big toe) are displayed in cm. We define the maximum allowable distance based on half of each body size as the real annotation is almost in the middle of each body part. For example, we use d = 6.8 for the heel which corresponds to

Fig. 3 The estimation measurements of hurdles athlete's foot

approx. 8 cm in this dataset. For the big toe and the ankle, we also set d to 4.0 and 5.4 which represent 4.5 and 7 cm in the dataset, respectively.

6 Results and discussion

6.1 Model performance

Table 3 shows RMSE of athlete's foot parts for different YOLO and DeepLabCut models. From the results, we can conclude the following observations:

1. Among the YOLO models evaluated, YOLOv8n demonstrated the lowest average RMSE of 1.75. The RMSE for the right ankle was observed to be the lowest among all foot parts, attributable to its relatively lesser occlusion in the images.
2. Between DeepLabCut models, ResNet152 achieved the minimum average RMSE of 6.39 when employing the "Imgaug" data augmentation technique.
3. The comparative analysis between the two augmentation methods reveals that "Imgaug" consistently outperforms "Tensorpack" across all models, evidenced by its lower RMSE values.
4. A key aspect of our analysis was evaluating the impact of data augmentation on model performance. Table 3 compares the RMSE values for pose estimation across various models under two scenarios: with and without augmentation. The results clearly indicate that data augmentation significantly improves the model's accuracy, particularly for YOLOv8 and ResNet-based frameworks. For example, YOLOv8l achieved an average RMSE of 1.85 with augmentation compared to 6.47 without augmentation. Similarly, the ResNet152 model demonstrated an average RMSE of 6.39 with augmentation (Imgaug) versus 6.47 without augmentation, showcasing the incremental improvement offered by augmentation techniques. These findings demonstrate that augmentation techniques play a critical role in enhancing model robustness by introducing variations in the training data, enabling better generalization to unseen scenarios. This analysis underscores the importance of integrating well-designed augmentation strategies in deep learning pipelines for pose estimation tasks.

In addition, we calculated PCK for the test data set for two frameworks. The PCK value for each body part represents the percentage of correctly predicted keypoints (e.g., heel,

Table 3 RMSE of the pose estimation for the proposed models in test dataset with and without data augmentation

Model	Left Ankle	Left Bigtoe	Left Heel	Right Ankle	Right Bigtoe	Right Heel	Average
YOLOv8n	1.77	1.70	1.72	1.62	1.95	1.75	**1.75**
YOLOv8s	1.82	1.74	1.74	1.65	1.97	1.78	1.78
YOLOv8m	1.90	1.80	1.78	1.76	1.94	1.88	1.84
YOLOv8l	1.90	1.81	1.79	1.76	1.94	1.88	1.85
YOLOv8x	6.55	6.55	6.52	6.66	6.77	6.71	6.63
MobileNet (Imgaug)	12.20	12.77	12.81	13.13	13.89	12.88	12.95
MobileNet (Tensopack)	14.77	15.84	12.62	14.62	17.62	13.45	14.82
DLCRNet (Imgaug)	6.56	7.24	6.68	7.09	7.73	7.00	7.05
DLCRNet (Tensopack)	6.76	7.84	7.24	6.22	6.96	6.04	6.84
ResNet101 (Imgaug)	6.33	7.45	6.52	6.05	6.53	6.23	6.52
ResNet101 (Tensopack)	7.08	8.10	7.38	7.03	7.85	7.44	7.48
ResNet152 (Imgaug)	5.71	7.37	6.42	6.05	6.84	5.97	6.39
ResNet152 (Tensopack)	6.76	7.66	7.20	6.24	6.86	6.21	6.82
MobileNet (Without)	16.12	19.34	17.28	16.70	19.79	19.48	18.11
DLCRNet (Without)	6.49	7.04	6.77	6.33	6.91	5.98	6.58
ResNet101 (Without)	6.22	7.53	6.47	6.07	6.82	6.33	6.57
ResNet152 (Without)	6.25	7.22	6.54	6.09	6.74	5.98	6.47

ankle, big toe) within a certain threshold distance from the ground truth keypoints. Figure 4 illustrates the PCK of feet parts by considering two distinct augmentation techniques and different thresholds' values in DeepLabCut. The results show that all models with 'Imgaug' consistently outperforming 'Tensorpack' across all thresholds. In addition, we can see the choice of threshold appears to significantly influence the accuracy of keypoint estimation, with higher threshold yielding a higher PCK value. A higher PCK value indicates better accuracy in predicting the body part's position. Between DeepLabCut models, ResNet152 (Fig. 4(g)) achieved the highest PCK value for right ankle estimation, with 43.1%. Between DL models, DLCRNet demonstrated the best accuracy in predicting the right heel position, achieving a PCK of 48.5%. In terms of big toe localization, ResNet152 yielded the highest PCK value of 19,1% for the right big toe.

Figure 5 demonstrates the PCK for YOLO models. The results show that right heel has maximum PCK compared to other feet's part for all models. Between Yolo models, YOLOv8l achieved the highest PCK value for the right ankle, right heel and left big toe estimation with PCK of 79.0%, 82.9% and 59.2%, respectively.

Figure 6 illustrates the PCK for various foot parts across different models used in DeepLabCut without data augmentation. The models utilizing data augmentation exhibit consistently higher PCK% values across all foot keypoints (Left Ankle, Left Bigtoe, Left Heel, Right Ankle, Right Bigtoe, and Right Heel). For example, the right heel shows significantly better performance with data augmentation, as the models adapt better to variations introduced

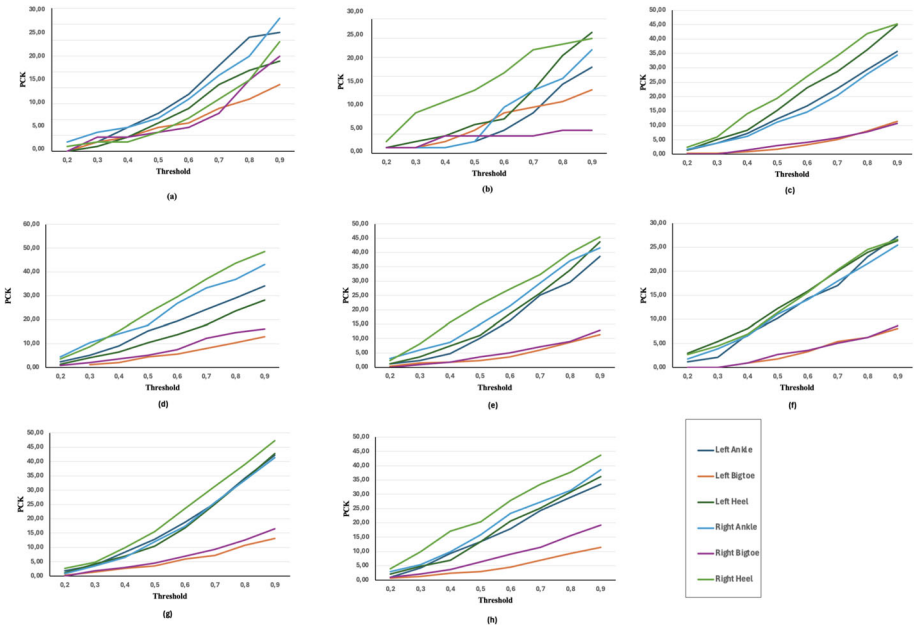

Fig. 4 The PCK (%) of feet parts in test dataset for models used in DeepLabCut based on two different augmentation techniques and thresholds for (a) MobileNet (Imgaug), (b) MobileNet (Tensopack), (c) DLCRNet (Imgaug), (d) DLCRNet (Tensopack), (e) ResNet101 (Imgaug), (f) ResNet101 (Tensopack), (g) ResNet152 (Imgaug) and (h) ResNet152 (Tensopack)

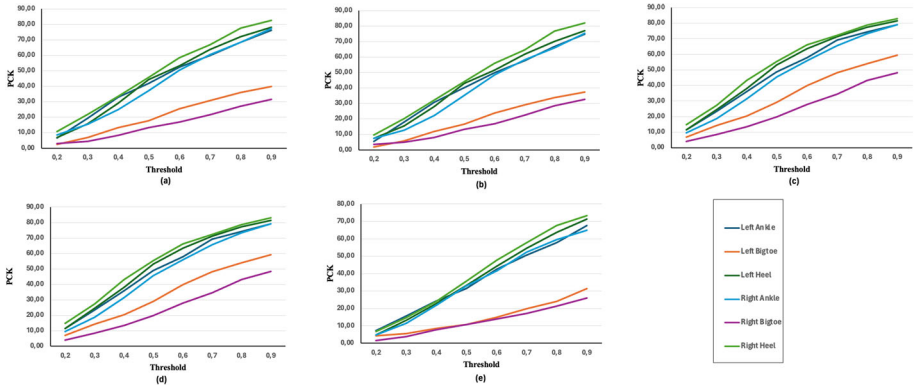

Fig. 5 The percentage of correct keypoints (%) of feet parts in test dataset for models used in YOLO based on different thresholds for (a) YOLOv8n, (b) YOLOv8s, (c) YOLOv8m, (d) YOLOv8l and (e) YOLOv8x

during training. Data augmentation enhances the generalizability of pose estimation models, making them more suitable for real-world applications where lighting, angles, and athlete variability can impact accuracy.

6.2 Example of applying pose estimates

Pose estimates could be deployed e.g. in estimating the contact and step air times. For the videos deployed in this study, the frame rate was 120 fps, which is not adequate for this

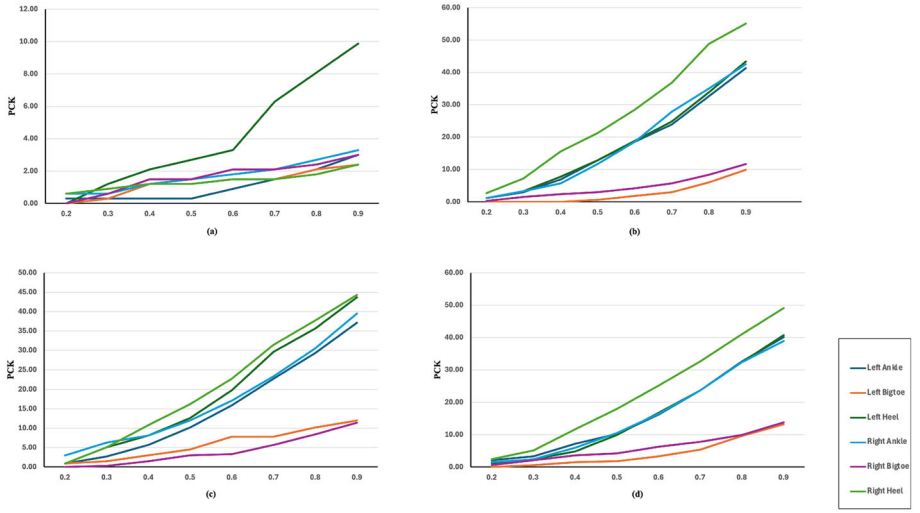

Fig. 6 The PCK (%) of feet parts in test dataset for models used in DeepLabCut (a) MobileNet, (b) DLCRNet, (c) ResNet101, (d) ResNet152 without data augmentation

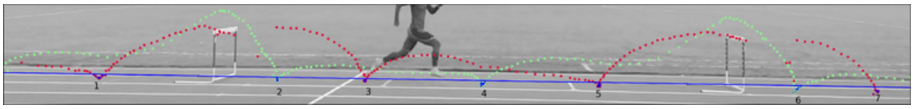

Fig. 7 Extracting the possible contact points. The predicted toe points are marked with dots (left: red dots, right: green dots. The left lane line is plotted with blue line, and the predicted toe points below the lane line with blue dots

kind of analysis, however, the benefits of using the pose estimation for this purpose can be demonstrated. High-speed cameras have become rather affordable nowadays, which enables a more detailed analysis. By finding the first and last touch of the contact, the error of the estimated contact time is approximately two frames at maximum. E.g. with 500 fps this leads to 4 ms maximum error. For such a high frame rate, manual searching through the frames for the contact frames can become tedious. Pose estimation offers a way to make this process easier.

Figure 7 shows an example of detecting the step contacts using the predicted toe points. Using the lane line, a subset of possible contact points can be found: For each predicted toe point, it was checked whether the point is below the left lane line. Using this procedure, the data points being searched through can be easily narrowed to a smaller subset. Figure 8 and 9 show the frames for the contacts 1–7 shown in Fig. 7. The first frame on the left is one frame before the predicted toe point descends below the lane line. The fourth frame on the right shows the last frame before the predicted toe point rices above the lane line. It can be seen that using the lane line as the threshold for finding the contact points is not adequate as such due to the inaccuracy of the toe point predictions. However, by enlarging the range of frames of being searched a few frames before and after the predicted point surpasses the lane line is enough to cover the range for first and last touch. This semi-automatic process offers a more efficient and fast search of the step contacts.

The step lengths could be determined as well with proper image rectification. Along with the rectification, using also other predicted joints such as knee and pelvis, a more detailed performance analysis of clearing the hurdle could be achieved by analyzing e.g. the takeoff angles.

One notable limitation of this study is the use of a 120 fps frame rate, which is suboptimal for detailed analysis of rapid movements, such as those in hurdling. This constraint may result in inaccuracies in detecting precise contact times and positions, as the temporal resolution is insufficient to capture subtle transitions within individual strides. Although high-speed

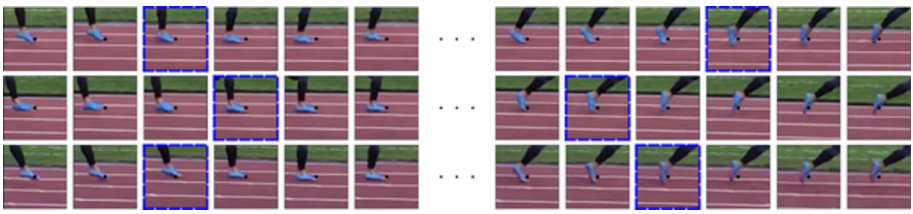

Fig. 8 Predicted toe points for the right foot (black circle). The first/second/third row shows the frames for the contact 2/4/6, respectively. The detected frames of the first and last touch are marked with blue dashed lines

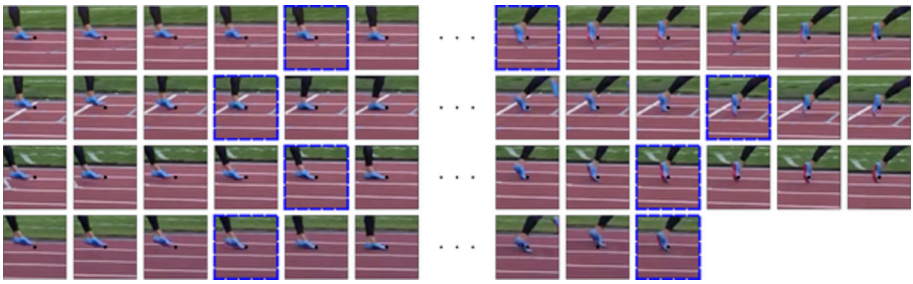

Fig. 9 Predicted toe points for the left foot (black circle). The first/second/third/fourth row shows the frames for the contact 1/3/5/7, respectively. The detected frames of the first and last touch are marked with blue dashed lines

cameras could significantly enhance the accuracy by reducing temporal gaps, their deployment might not always be feasible due to cost and accessibility. To address this limitation, future work could explore interpolation methods or machine learning-based motion prediction to approximate keypoint trajectories more accurately using lower frame rate data. These workarounds could extend the applicability of our approach to scenarios with limited resources while maintaining analytical rigor.

6.3 Qualitative results of pose estimation

In addition to the quantitative evaluation metrics, we provide qualitative analyses of the pose estimation results to offer deeper insights into the models' performance. Figure 10 illustrates a comparison between the predicted keypoints and the ground truth annotations across various frames with YOLOv8l. The visual alignment of the heel, big toe, and ankle positions highlights the model's ability to accurately estimate body poses, demonstrating its practical reliability for sports analytics. These visual examples complement the RMSE and PCK metrics, offering a more intuitive understanding of the model's efficacy in real-world applications.

Fig. 10 Visualization of pose estimation Performance using YOLOv8l: Predicted keypoints vs. ground truth annotations across frames

7 Conclusion

The application of human pose estimation can be used to derive valuable insights while optimizing technique and performance of the athletes. In this present study we wanted to verify the usability of the low-end video recorders such as mobile phones. We investigated two popular CNN-based frameworks, YOLO and DeepLabCut, for pose estimation in the context of hurdle athletes. For this purpose, we deployed video material from a female athlete during her training sessions and manually labeled three main points of interest in the images including heel, ankle, and big toe. These points can be utilized in modeling the step pattern of the athlete. We leveraged the YOLO results to predict the athlete's body skeletal keypoints, which reveal valuable information about the step progression in temporal domains. In summary, this study demonstrates, how simple camera arrangements and relatively fast in-the-field computation can help the coach to compare different athletes, or development trend of an athlete. YOLO particularly seems to be fit to this on-site sports analytics.

In addition to demonstrating the accuracy and effectiveness of YOLOv8l and related pose estimation models, this study highlights their practical applications in enhancing athletic training and performance. The detailed biomechanical insights provided by these models allow coaches to analyze athletes' movement patterns with precision. For instance, keypoint predictions can identify inefficiencies in stride length or hurdle clearance techniques, enabling targeted interventions to improve performance. Moreover, the temporal data generated by these models can help in optimizing step cadence and air time, which are crucial for maximizing speed and efficiency in hurdling. By integrating these pose estimation models into regular training regimens, coaches can track progress over time, identify injury risks through abnormal movement patterns, and design customized drills to enhance specific skills. These applications emphasize the tangible benefits of adopting AI-driven tools in sports analytics, offering a cost-effective and scalable solution to advancing athletic performance.

While the study provides valuable insights into pose estimation using CNN-based frameworks, the dataset's limitation to a single female athlete restricts the generalizability of the results. To address this, future research will aim to include a more diverse sample of athletes, representing varying body types, genders, and performance levels. This expansion will allow for testing the models' adaptability and effectiveness in different biomechanical and environmental contexts, thereby ensuring broader applicability and enhancing the credibility of the findings.

We also will focus on integrating explainable artificial intelligence techniques [45] into our pose estimation framework. For example, incorporating methods such as feature attribution maps or visual attention mechanisms could provide greater transparency into the model's decision-making process. This advancement will empower coaches and researchers to interpret the results more effectively, enabling targeted improvements in training methodologies and performance optimization.

Author Contributions Pouya Jafarzadeh: Conceptualization, Software, Methodology, Writing - Original Draft. Luca Zelioli: Data Curation, Software, Formal Analysis. Petra Virjonen: Investigation, Formal Analysis, Writing - Review & Editing. Fahimeh Farahnakian: Validation, Supervision, Writing - Review & Editing. Paavo Nevalainen: Validation, Supervision, Writing - Review & Editing. Jukka Heikkonen: Resources, Supervision.

Funding Open Access funding provided by University of Turku (including Turku University Central Hospital).

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethical Approval Ethics approval from the University of Turku ethics committee was not required for this study, as it involved only one subject who provided informed consent prior to participation. The inclusion of a high-quality athlete as the subject was deemed essential due to the natural test data generated by the athlete's video framerate and high movement speed. Recognizable facial features have been blurred since head movements during the performance were not within the scope of this study. The subject's consent includes the following conditions: (1) The distribution and use of videos are restricted to producing the results reported in this paper. (2) Video material will not be shared with other research groups without prior consultation with the test subject. (3) Facial features are blurred to protect the subject's identity.

Informed Consents Informed consent was obtained from all subjects involved in the study.

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Ludwig K, Scherer S, Einfalt M, Lienhart R (2021) Self-supervised learning for human pose estimation in sports, 1–6. https://doi.org/10.1109/ICMEW53276.2021.9456000
2. Wang J, Qiu K, Peng H, Fu J, Zhu J (2019) Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In: Proceedings of the 27th ACM International Conference on Multimedia. MM '19, pp. 374–382. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3343031.3350910
3. Murthy P, Taetz B, Lekhra A, Stricker D (2023) Divenet: Dive action localization and physical pose parameter extraction for high performance training. IEEE Access 11:37749–37767. https://doi.org/10.1109/ACCESS.2023.3265595
4. Pituxcoosuvarn M, Murakami Y (2022) Rugby goal kick prediction using openpose coordinates and lstm. In: 2022 26th International Computer Science and Engineering Conference (ICSEC), pp. 161–166. https://doi.org/10.1109/ICSEC56337.2022.10049358
5. Janbi NF, Almuaythir N (2023) Bowlingdl: A deep learning-based bowling players pose estimation and classification. In: 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), pp. 1–6. https://doi.org/10.1109/ICAISC56366.2023.10085434
6. Einfalt M, Zecha D, Lienhart R (2018) Activity-conditioned continuous human pose estimation for performance analysis of athletes using the example of swimming. arXiv:1802.00634
7. Sangüesa AA, Ballester C, Haro G (2019) Single-camera basketball tracker through pose and semantic feature fusion. arXiv:1906.02042
8. Jafarzadeh P, Virjonen P, Nevalainen P, Farahnakian F, Heikkonen J (2021) Pose estimation of hurdles athletes using openpose. In: 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), pp. 1–6. https://doi.org/10.1109/ICECCME52200.2021.9591066
9. Cao Z, Simon T, Wei S, Sheikh Y (2016) Realtime multi-person 2d pose estimation using part affinity fields. arXiv:1611.08050
10. Xiao B, Wu H, Wei Y (2018) Simple baselines for human pose estimation and tracking. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds) Computer Vision - ECCV 2018. Springer, Cham, pp 472–487

11. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. arXiv:1603.06937
12. Chen Y, Wang Z, Peng Y, Zhang Z, Yu G, Sun J (2017) Cascaded pyramid network for multi-person pose estimation. arXiv:1711.07319
13. Sun K, Xiao B, Liu D, Wang J (2019) Deep high-resolution representation learning for human pose estimation. arXiv:1902.09212
14. Geng Z, Sun K, Xiao B, Zhang Z, Wang J (2021) Bottom-up human pose estimation via disentangled keypoint regression. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14671–14681. https://doi.org/10.1109/CVPR46437.2021.01444
15. Newell A, Deng J (2016) Associative embedding: End-to-end learning for joint detection and grouping. arXiv:1611.05424
16. Ram P, Padmavathi S (2016) Analysis of harris corner detection for color images. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), pp. 405–410.https://doi.org/10.1109/SCOPES.2016.7955862
17. Burger W, Burge MJ (2016) Scale-Invariant Feature Transform (SIFT), pp. 609–664. Springer, London
18. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. In: Leonardis A, Bischof H, Pinz A (eds) Computer Vision - ECCV 2006. Springer, Berlin, Heidelberg, pp 404–417
19. Lauer J, Zhou M, Ye S, Menegas W, Nath T, Rahman MM, Di Santo V, Soberanes D, Feng G, Murthy VN, Lauder G, Dulac C, Mathis MW, Mathis A (2021) Multi-animal pose estimation and tracking with deeplabcut. https://doi.org/10.1101/2021.04.30.442096, https://www.biorxiv.org/content/early/2021/04/30/2021.04.30.442096.full.pdf
20. Kendall A, Grimes M, Cipolla R (2015) Convolutional networks for real-time 6-dof camera relocalization. arXiv:1505.07427
21. Redmon J, Divvala SK, Girshick RB, Farhadi A (2015) You only look once: Unified, real-time object detection. arXiv:1506.02640
22. Mathis A, Mamidanna P, Cury K, Abe T, Murthy V, Mathis M, Bethge M (2018) Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Nature Neuroscience **21**. https://doi.org/10.1038/s41593-018-0209-y
23. Lin T, Maire M, Belongie SJ, Bourdev LD, Girshick RB, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. arXiv:1405.0312
24. Čoh M, Bončina N, Štuhec S, Mackala K (2020) Comparative biomechanical analysis of the hurdle clearance technique of Colin Jackson and Dayron Robles: Key studies. Applied Sciences **10**(9). https://doi.org/10.3390/app10093302
25. Whittle MW (1993) 10 - gait analysis. In: McLatchie, G.R., Lennox, C.M.E. (eds.) The Soft Tissues, pp. 187–199. Butterworth-Heinemann. https://doi.org/10.1016/B978-0-7506-0170-2.50017-0, https://www.sciencedirect.com/science/article/pii/B9780750601702500170
26. Lipfert SW, Günther M, Renjewski D, Grimmer S, Seyfarth A (2012) A model-experiment comparison of system dynamics for human walking and running. Journal of Theoretical Biology 292:11–17.https://doi.org/10.1016/j.jtbi.2011.09.021
27. Liu J, Mu X, Liu Z, Li H (2023) Human skeleton behavior recognition model based on multi-object pose estimation with spatiotemporal semantics. Machine Vision and Applications 34:1432–1769. https://doi.org/10.1007/s00138-023-01396-0
28. Benson LC, Räisänen AM, Clermont CA, Ferber R (2022) Is this the real life, or is this just laboratory? a scoping review of imu-based running gait analysis. Sensors **22**(5). https://doi.org/10.3390/s22051722
29. Clark KP, Ryan CRMLJ, Stearne DJ (2023) Horizontal foot speed during submaximal and maximal running. J Hum Kinet **87**.https://doi.org/10.5114/jhk/159578
30. Kalluri T, Pathak D, Chandraker M, Tran D (2020) FLAVR: Flow-Agnostic Video Representations for Fast Frame Interpolation. arXiv e-prints, 2012–08512. https://doi.org/10.48550/arXiv.2012.08512
31. Chen LC, Yang Y, Wang J, Xu W, Yuille AL (2016) Attention to scale: Scale-aware semantic image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
32. Neverova N, Wolf C, Taylor GW, Nebout F (2015) Multi-scale deep learning for gesture detection and localization. In: Agapito L, Bronstein MM, Rother C (eds) Computer Vision - ECCV 2014 Workshops. Springer, Cham, pp 474–490
33. Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658
34. Farabet C, Couprie C, Najman L, LeCun Y (2013) Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell 35(8):1915–1929. https://doi.org/10.1109/TPAMI.2012.231

35. Toshev A, Szegedy C (2013) Deeppose: Human pose estimation via deep neural networks. arXiv:1312.4659
36. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:1512.03385
37. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://doi.org/10.48550/ARXIV.1704.04861
38. Akkem Y, Kumar BS, Varanasi A (2023) A comprehensive review of synthetic data generation in smart farming by using variational autoencoder and generative adversarial network. Indian Journal of Science and Technology 16(48):4688–4702. https://doi.org/10.17485/IJST/v16i48.2850
39. Saito H, Kimura M, Yaguchi S, Inamoto N (2002) View interpolation of multiple cameras based on projective geometry. https://api.semanticscholar.org/CorpusID:14110176
40. Basodi S, Ji C, Zhang H, Pan Y (2020) Gradient amplification: An efficient way to train deep neural networks. Big Data Mining and Analytics 3(3):196–207. https://doi.org/10.26599/BDMA.2020.9020004
41. Debnath B, O'Brien M, Yamaguchi M, Behera A (2018) Adapting mobilenets for mobile based upper body pose estimation. In: 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–. https://doi.org/10.1109/AVSS.2018.8639378
42. Chollet F (2016) Xception: Deep Learning with Depthwise Separable Convolutions.https://doi.org/10.48550/ARXIV.1610.02357
43. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. International Conference on Learning Representations
44. Mathis A, Schneider S, Lauer J, Mathis MW (2020) A primer on motion capture with deep learning: Principles, pitfalls and perspectives. arXiv:2009.00564
45. Akkem Y, Kumar BS, Varanasi A (2023) Streamlit application for advanced ensemble learning methods in crop recommendation systems – a review and implementation. Indian Journal of Science and Technology 16(48):4688–4702. https://doi.org/10.17485/IJST/v16i48.2850