Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Study Project on Lecture
"Deep Learning for Robot Vision"

Human Pose Estimation Project Report

Krishna Teja Nallanukala (9039518,knalla2s)

Supervised by

Prof. Dr.-Ing. Sebastian Houben

May 2022

1 Introduction

Human Pose Estimation (HPE) is a task in computer vision that focuses on identifying the position of a human body in a specific scene obtained from images or videos and form a skeleton like representation of the human body. It basically identifies and classifies the joints in the human body as a set of coordinates called as keypoints. The connection between the key points is called as a pair and inorder to form a pair the connections between the keypoints has to be significant. The connection between these key points can be used to describe the pose of a person.

Figure 1: Sample Skeleton output of pose estimation. Image credit: [2]

The classical machine learning models used for human pose estimation lacked correlation, accuracy and generalization capabilities. Hence need for deep learning models for the problem of estimating human pose for task specific applications has increased.

In general human pose estimation problem is classified into two types 2D pose estimation and 3D pose estimation. 2D pose estimation estimates the (x,y) coordinates of the joints of human bodies present in the scene. 3D pose estimation on the other hand estimates (x,y,z) coordinates by providing an additional depth parameter z.

1.1 Problem Statement

- The current state-of-the-art human pose estimation methods depend on learning algorithms to estimate model parameters from training data and use complicated models.
- The performance of these methods crucially depends on the availability of the annotated training images that are representative for the appearance of people clothing, strong articulation, partial (self-)occlusions and truncation at image borders [3].
- In this project we are interested to perform a performance evaluation of the state-of-the-art Human Pose Estimation using deep learning methods.
- Some of the possible challenges that can be considered in this project are, accurate estimation of pose under various circumstances like, detection in crowds, occlusion due to light, partial availability of human data in the images.

2 Related work

- Classical approaches of human pose estimation used shallow machine learning algorithms like random forest with in a pictorial structure framework along with feature building methods like histograms, contours and (HOG) histogram oriented gaussian techniques.
- The deep learning methods of 2D human pose estimation is classified into two approaches, single person pipeline and multiperson pipeline.
- Single person pipeline uses direct regression or heat map based methods to identify the key points.

- Multiperson pipeline uses top-down and bottom-up approaches to estimate the pose of the person.
- The main objective of top-down approach is to first localize the humans in the scene and then estimate the joints followed by calculating the pose.

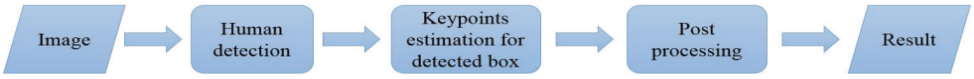

Figure 2: Framework of top-down pipeline [4]

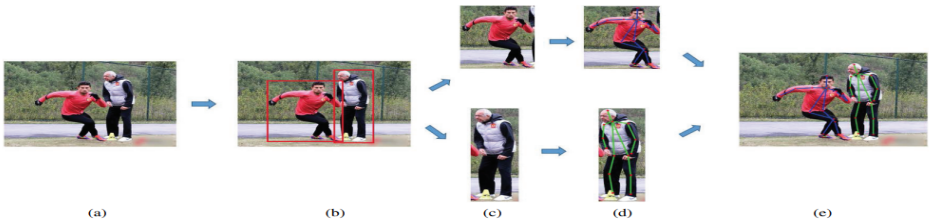

Figure 3: An illustration of top down pipeline. (a) Input image, (b) two persons detected by human detector, (c) cropped single person image, (d) single person pose detection result, and (e) multi-person pose detection result.[4]

- Bottom up approach aims to estimate the joints of the human body directly followed by calculating the pose.

Figure 4: Framework of bottom-up pipeline.[4]

Figure 5: An illustration of bottom-up pipeline. (a) Input image, (b) keypoints of all the person, and (c) all detected keypoints are connected to form human instance.[4]

- Both pipelines uses 3 types of representation to visualize the pose of a person. (Skeleton based, Contour based and Volume based models).
- State of the art 2D human pose estimation techniques include,
  - Top-down approaches
    - Simple Baseline model
    - KAPAO
    - Deep Pose
    - Alpha Pose
  - Bottom-up approaches
    - Open Pose
    - DeepCut

3 Model Architectures

3.1 Simple Baseline Model

3.1.1 Overview

A common state of the art model used for feature extraction is ResNet. The same can be used for feature extraction for the purpose of human pose estimation. The complete model for human pose estimation of simple baseline method uses a deconvolutional head added over to the last stages of ResNet. Figure 6 shown below illustrates the architecture of the simple baseline model. In this project ResNet50

(C_5)is used as the backbone for feature extraction purpose. The deconvolutional head is denoted by D_3.

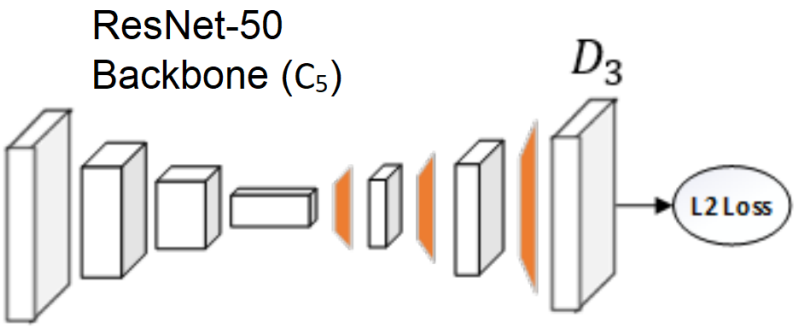

Figure 6: Architecture of Simple Baseline [11]

3.1.2 Details on Deconvolutional Head

The deconvolutional head consists of three layers each with ReLU activation function and batch normalization. Each layer of the deconolutional head consists of 256 number of 4x4 filters with a stride length of 2 pixels. The last layer consists of a 1x1 convolutional layer which generates heatmaps $H_1, H_2, ...H_k$ for all k human joints which are otherwise known as keypoints.

3.1.3 Miscellaneous Details About the Model

Mean squared error (MSE) is used as loss function between the predicted heatmap and the target truth heatmap. The target heatmap is generated by placing a 2D Gaussian distribution over all K joints of the human body. The ResNet backbone used is pretrained on ImageNet dataset. Learning rate used:

- Base: 1e-3
- At 90 epochs: 1e-4
- At 120 epochs: 1e-5

The model is trained for a total of 140 epochs using minibatch gradient descent with a batch size of 128 images using Adam optimizer.

3.1.4 Complete Pipeline (Human Detection + Keypoint Estimation)

The complete pipepline for human pose estimation involves, firstly detection of humans. The simple baseline method uses faster R-CNN for human detection. In this project we use a pretrained faster R-CNN. The faster R-CNN is not trained on the COCO dataset for human detection.

The human class as well as its bounding box is provided as output from faster R-CNN, using which the input image is cropped and only the human region in the input image is fed to the simple baseline model sequentially for keypoint detection. The overall pipeline is illsutrated in Figure 7 as shown below.

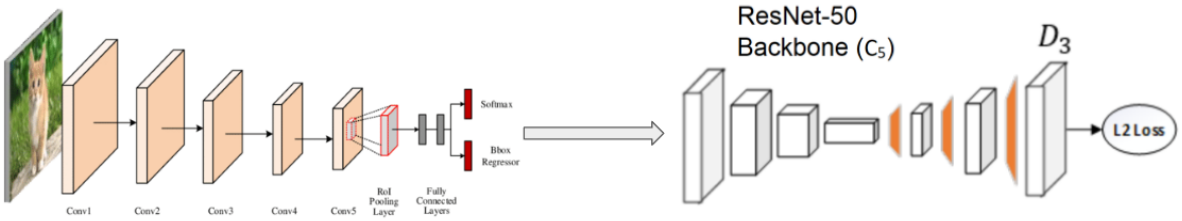

Figure 7: Complete Pipeline [11]

3.2 KAPAO

KAPAO [7] is a fast single-stage multi-person human pose estimation approach that uses a dense anchor-based detection framework to model key points and poses as objects simultaneously. KAPAO is a new heatmap-free keypoint detector that recognizes pose objects and keypoint objects at the same time and combines the results to estimate human poses. KAPAO uses YOLO [9][10] object detector for the feature extraction and detection of the humans in the target image. In the key point object O_k, the coordinates are represented at the center of a small bounding box with equal width and height, which is just an adaption of the usual object representation. In the below Figure 8, keypoint objects are represented at the knee joint of a human with bounding box coordinates.

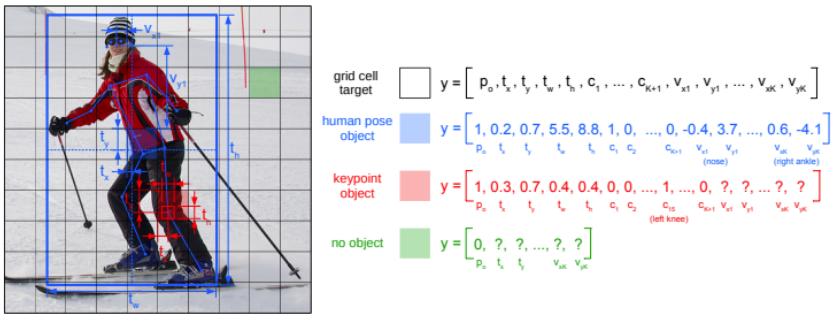

Figure 8: sample KAPAO detetction targets. Image Credit: [7]

Pose object o^p is also an adaption of general object representation but additionally, it also includes key points associated with objects. In Figure 8, the human is detected with a blue bounding box which also contains keypoints belonging to person. Pose object comprises both person class as well as keypoints associated with a person. Both keypoint and pose objects have unique advantages that can be leveraged for multiperson human pose estimation without the need for a bottom-up approach. Keypoint objects are focused on the detection of individual keypoints that are prominent local features. However, keypoint objects don't convey any knowledge about the idea of a person or a position. Pose objects, on the other hand, let the network understand the spatial relationships within a group of keypoints, making them better suitable for localizing keypoints with poor local characteristics. Knowing that a pose object's subspace contains keypoint objects, the KAPAO network was developed to identify both item kinds simultaneously using employing a single shared network head with a low computational overhead.

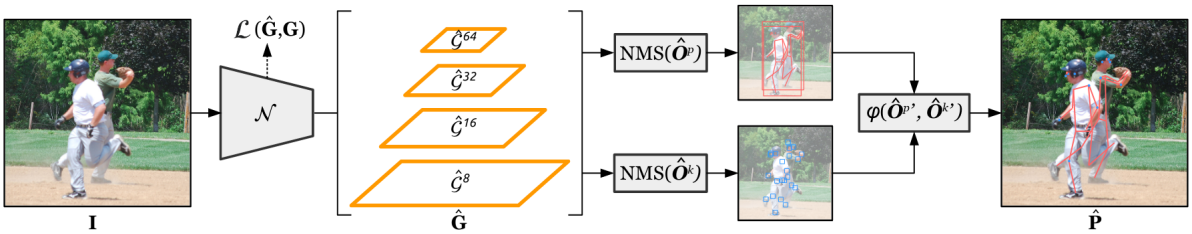

Figure 9: KAPAO architecture. Image Credit: [7]

A diagram of the KAPAO pipeline is provided in Figure 9. KAPAO maps an RGB image I to a set of output grids G at different scales using a YOLO style feature extractor to enable the model to detect objects even in smaller sizes in an image. These grids comprise the projected pose objects O_p and keypoint objects O_k using a dense detection network N trained with the multi-task loss L. The multitask loss consists of objectness, bounding boxes loss, class loss, and pose object keypoints loss. The candidate detections $O_p^{'}$ and $O_k^{'}$ are obtained using non-maximum suppression (NMS), which are then merged together using a matching algorithm to get the final human posture predictions $\hat{p}$.

3.3 Evaluation Metrics

The evaluation metrics used for measuring the performance of these two models for Human pose detection are Average Precision(AP) and Average Recall(AR)[7]. These metrics are similar to metrics used for object detection tasks but the similarity measure between ground truth and predictions is done using Object Keypoint Similarity(OKS)[1] in the case of KAPAO, whereas Simple baseline uses regression technique to measure the similarity as it is based on heatmap technique. In KAPAO, Object detection tasks use Intersection Over Union(IOU) for similarity measure. The principle behind OKS is it measures the Euclidian distance between the ground truth and predicted key points.

4 Experimental Setup

4.1 Dataset - COCO 2017 [6]

- The COCO Object Detection Task is intended to advance the object detection approach.
- Using bounding box output or object segmentation output presents two challenges in object detection.
- More than 200,000 pictures from 80 different object categories make up the COCO train, validation, and test sets.

- A thorough segmentation mask is attached to every instance of an object. Keypoints are labeled for human instances.
- For Human Pose Estimation task, we are only interested in the person class.
- Dataset Description:
  - 56000 person class images in total.
  - Trainset size - 50000
  - Validation size - 6000
  - Image resolution - 640 * 480

4.2 Hardware and software tools/frameworks

- Frameworks and packages: PyTorch, NumPy, SciPy, Pandas
- GPU used: 1 x 12 GB Nvidia Tesla K8

4.3 Methods

In this project, we trained three models. The first one is the simple baseline model, and the other two are kapao models. Since the training time for kapo is longer, to reduce the training time and to get some inference we trained two kapao models with different image sizes, 512x512 and 1280x1280. The hyperparameters used for training are more or less the same as the authors used to train their models. The only exception is for the batch size because the authors were using multiple GPUs and doing parallel training across these GPUs. But in our case, we are using only one 12 GB Tesla k8 GPU card. The hyperparameters used for all three models are shown in Table 1.

Hyper-parameter	**BaseLine**	**Kapao**	**Kapao_resized**
ImageSize	192x256	1280x1280	512x512
BatchSize	96	16	32
LearningRate	0.001	LR Scheduler	LR Scheduler
Epochs	70	118	468
LossFunction	MeanSquared	BCE	BCE
Optimizer	Adam	SGD(Nesterov Momentum)	SGD(Nesterov Momentum)
OKS Threshold	0.5	-	-
IOU Threshold	-	0.5	0.5

Table 1: Hyper-parameters for all the three experiments

5 Results

Training and validation plots obtained for the three models are discussed in this section.

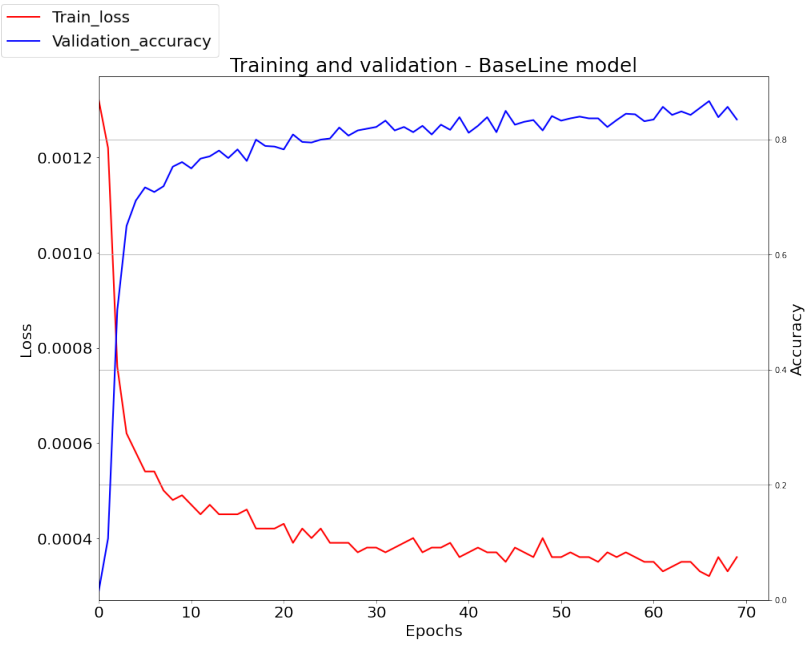

Figure 10: Training loss Vs Validation accuracy of BaseLine model

Figure 13, shows that the average precision of the simple baseline model is more

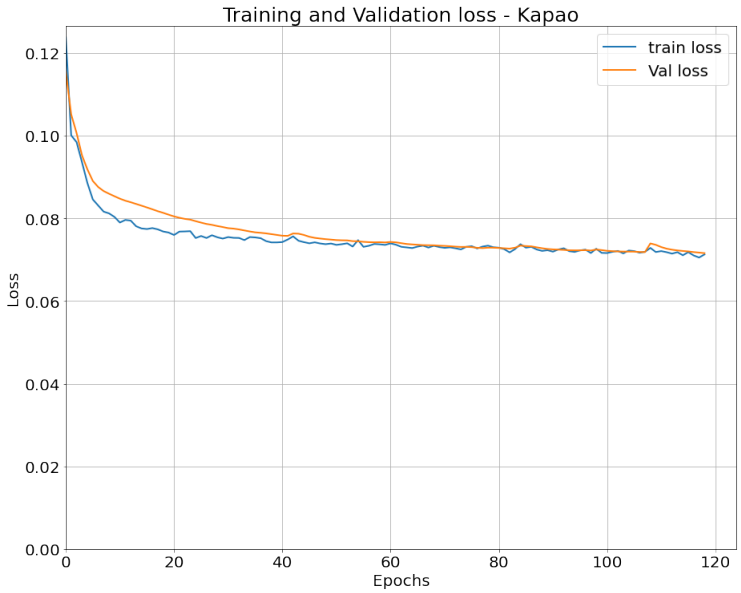

Figure 11: Training Vs Validation loss of Kapao model with image size 1280x1280.

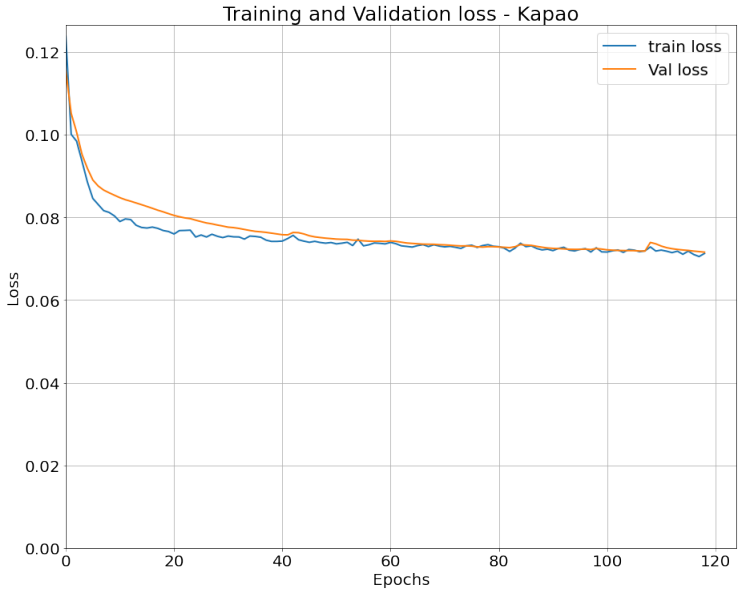

Figure 12: Training Vs Validation loss of Kapao model with image size 512x512.

than the kapao model.

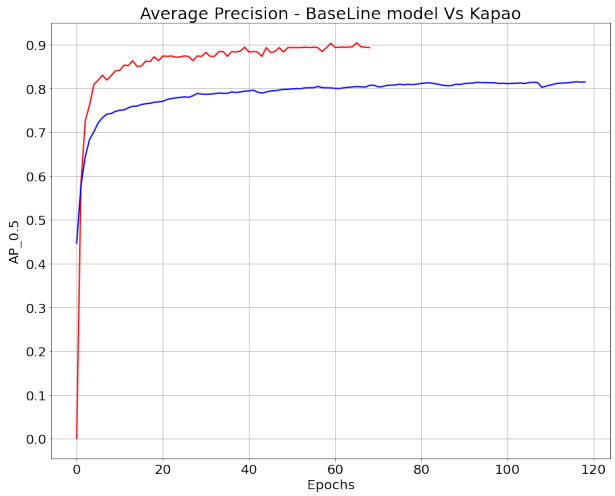

Figure 13: Average Precision of BaseLine Vs Kapao

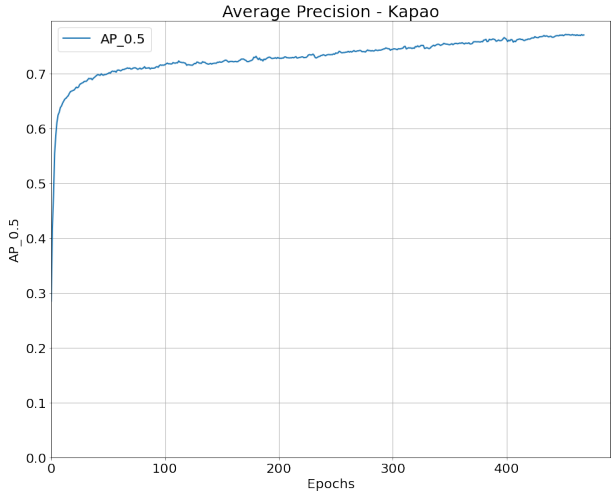

Figure 14: Average Precision of Kapao model with image size 512x512.

5.1 Inference

In this section, a comparison of pose predictions by both kapo and simple baseline models are shown. Out of the two kapo models, below predictions are of kapo 512 x 512 model. This kapao model is chosen for comparison because training time spent

on 1208x1208 model is very less and training loss is not greatly reduced during the training. Kapao 1208x1208 model needs more training time, so not chosen for evaluation purpose in our case.

Figure 15: Simple baseline example-1

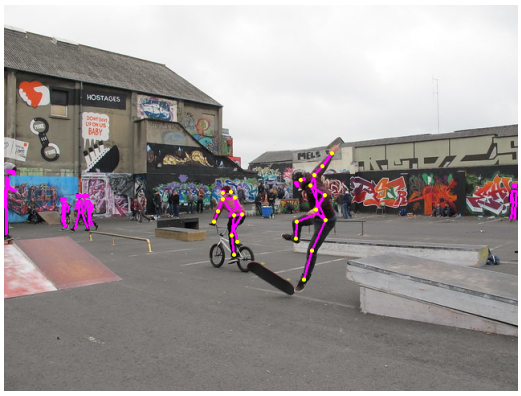

Figure 16: KAPAO example-1

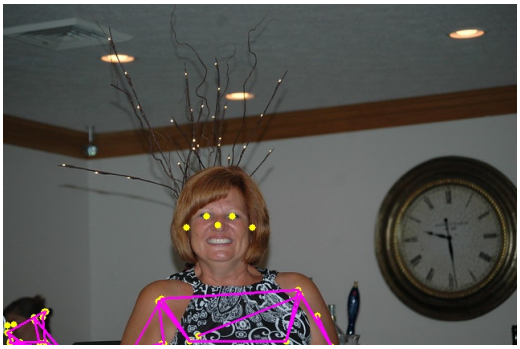

Figure 17: Simple baseline example-2

Figure 18: KAPAO example-2

Figure 19: Simple baseline example-3

Figure 20: KAPAO example-3

Figure 21: Simple baseline example-4

Figure 22: KAPAO example-4

Figure 23: Simple baseline example-5

Figure 24: KAPAO example-5

6 Conclusion

6.1 Simple Baseline Model

- Achieves an mAP of 73.7 % on COCO keypoint dataset.
- Winner of COCO 2017 keypoint detection challenge.

6.2 Kapo

- Achieves a mAP of 81.1 % on COCO 2017 dataset.
- This model is an extension of the YoloV5 object detector and is tested on the COCO 2017 dataset. Authors haven't participated in the coco challenge.

From the results, we can see that the multi-stage simple baseline model predicted the key points with better mean average precession when compared with the single-stage Kapo model. Kapo is a single-stage detector that uses a YoloV5 object detector as a backbone. Thus kapao prediction time is very less, making it very much convenient for the human pose tracking problem. Results show that there are no significant differences in the mean average precision of both models.

References

[1] COCO Dataset Overview. `https://cocodataset.org/#keypoints-2020`. Accessed: 2022-06-05.

[2] Skeleton human pose estimation . `https://nanonets.com/blog/human-pose-estimation-2d-guide/`.

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[4] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24 (6):663–676, 2019. doi: 10.26599/TST.2018.9010100.

[5] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] William McNally, Kanav Vats, Alexander Wong, and John McPhee. Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation. *arXiv preprint arXiv:2111.08557*, 2021.

[8] Author Name. Book title. *Lecture Notes in Autonomous System*, 1001:900–921, 2003. ISSN 0302-2345.

[9] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[10] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 13029–13038, 2021.

[11] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.