Journal of Real-Time Image Processing
https://doi.org/10.1007/s11554-021-01132-9

SPECIAL ISSUE PAPER

EfficientHRNet

Efficient and scalable high-resolution networks for real-time multi-person 2D human pose estimation

Christopher Neff[1] · Aneri Sheth[1] · Steven Furgurson[1] · John Middleton[1] · Hamed Tabkhi[1]

Received: 6 January 2021 / Accepted: 13 May 2021

Abstract
There is an increasing demand for lightweight multi-person pose estimation for many emerging smart IoT applications. However, the existing algorithms tend to have large model sizes and intense computational requirements, making them ill-suited for real-time applications and deployment on resource-constrained hardware. Lightweight and real-time approaches are exceedingly rare and come at the cost of inferior accuracy. In this paper, we present EfficientHRNet, a family of lightweight multi-person human pose estimators that are able to perform in real-time on resource-constrained devices. By unifying recent advances in model scaling with high-resolution feature representations, EfficientHRNet creates highly accurate models while reducing computation enough to achieve real-time performance. The largest model is able to come within 4.4% accuracy of the current state-of-the-art, while having 1/3 the model size and 1/6 the computation, achieving 23 FPS on Nvidia Jetson Xavier. Compared to the top real-time approach, EfficientHRNet increases accuracy by 22% while achieving similar FPS with $\frac{1}{3}$ the power. At every level, EfficientHRNet proves to be more computationally efficient than other bottom-up 2D human pose estimation approaches, while achieving highly competitive accuracy.

Keywords Human pose estimation · High-resolution networks · Model scaling · Real-time · Lightweight

1 Introduction

Two-dimensional human pose estimation is a common task used in many popular smart applications and has made substantial progress in recent years. There are two primary approaches to 2D human pose estimation. The first is a top-down approach, where cropped images of humans are provided and the network uses those cropped images to produce human keypoints. Top-down approaches rely on object detectors to provide initial human crops, thus they often come with relatively higher computation cost, and are not truly end-to-end. The second is a bottom-up approach, where a network works off the original image and produces human keypoints for all people in the image. While these methods often do not quite reach the accuracy that is possible with state-of-the-art (SotA) top-down approaches, they come with relatively lower model size and computational overhead. Even so, SotA bottom-up approaches are still quite large and computationally expensive. The current SotA [13] having 63.8 million parameters and requiring 154.3 billion floating-point operations.

Many emerging Internet-of-Things (IoT) applications require lightweight real-time multi-person pose estimation at the edge, next to the cameras. This is more pronounced in a broad range of smart and connected applications with demands for continuous human activity analysis and

This research is supported by the National Science Foundation (NSF) under Award no. 1831795.

✉ Christopher Neff
cneff1@uncc.edu

Aneri Sheth
asheth2@uncc.edu

Steven Furgurson
sfurgurs@uncc.edu

John Middleton
jmiddl11@uncc.edu

Hamed Tabkhi
htabkhiv@uncc.edu

[1] University of North Carolina at Charlotte, Charlotte, NC, USA

Published online: 01 June 2021

behavioral monitoring. Few examples are video surveillance, patient monitoring, and public safety [6, 18, 38]. These applications demand agile but highly accurate human pose estimation that can run next to the cameras on the IoT edge devices. Despite this, there has been a dearth of attention towards developing lightweight bottom-up methods capable of real-time execution under constrained computational resources. To address the gap, there is a need for a family of lightweight real-time human pose estimation models that achieves accuracy comparable to SotA.

In this paper, we present EfficientHRNet,[1] a family of lightweight scalable networks for high-resolution and efficient real-time bottom-up multi-person pose estimation. EfficientHRNet unifies the principles of SotA EfficientNet [56] and HRNet [54], and presents a new formulation that enables near SotA human pose estimation while being more computationally efficient than all other bottom-up methods. Similar to HRNet, EfficientHRNet uses multiple resolutions of features to generate keypoints, but in a much more efficient manner. At the same time, it uses EfficientNet as a backbone and adapts its scaling methodology to be better suited for human pose estimation. To enable lightweight real-time execution, EfficientHRNet further expands the EfficientNet formulation to not only scale below the baseline, but also jointly scale down the input resolution, High-Resolution Network, and Heatmap Prediction Network. Through this, we create a family of networks that can address the entire domain of real-time 2D human pose estimation while being flexible towards accuracy and computation requirements of an application.

We evaluate accuracy on the COCO dataset [37] and real-time performance on the Nvidia NX Xavier. Figure 1 demonstrates how our models provide equivalent or higher accuracy at lower computational costs than their direct peers. When comparing to SotA bottom-up models, baseline EfficientNet competes in accuracy while requiring much less computation, resulting in faster inference. Compared to HRNet[2] [54], EfficientHRNet achieves 0.4% higher accuracy while reducing computation requirements by 34%. When comparing to HigherHRNet [13] and PersonLab [44], EfficientHRNet sees between a 1.7 and 5.1% decrease in accuracy, while reducing computation requirements by an impressive 83–93%. This results in a 3.4 × FPS increase over HigherHRNet. Even when comparing to models designed specifically for lightweight execution, such as Lightweight OpenPose [43], a scaled-down EfficientHRNet is able to achieve 10.1% higher accuracy while further reducing computation by 15%, maintaining similar FPS while requiring

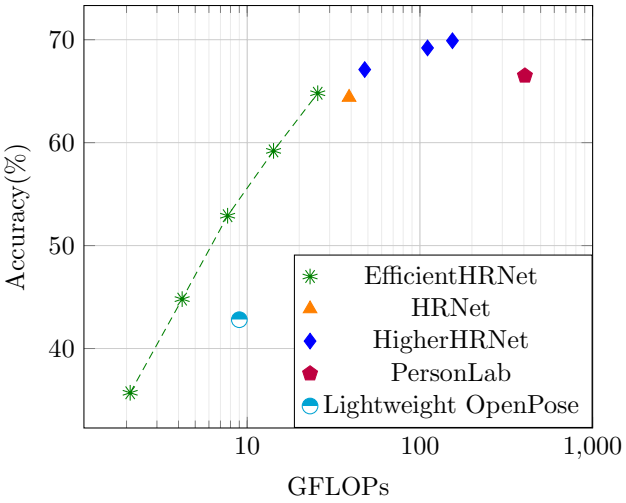

Fig. 1 Comparison of computational complexity and accuracy between bottom-up human pose estimation methods measured on COCO val dataset. *X*-axis is logarithmic in scale

$\frac{1}{3}$ the power. In addition, the scaled-down backbone models have been evaluated in isolation on ImageNet. The results demonstrate competitive accuracies while achieving greater efficiency than their peers.

Summarily, the contributions of this article are:

- We propose EfficientHRNet as the first approach to provide lightweight, scalable models for bottom-up real-time multi-person 2D human pose estimation that achieves comparable accuracy to the SotA.
- We propose a novel formulation integrating the scalability of EfficientNet throughout our high-resolution networks to reduce the computational complexity and allow for real-time execution.
- We are the first to provide a downwards scaling formulation to create compact EfficientNet models that scale below the baseline for embedded and edge IoT devices with restrained computation power.
- We provide comprehensive analysis on the challenging COCO dataset [37] to show how our models compare against SotA and real-time bottom-up approaches in accuracy, model size, computational complexity, efficiency, and real-time execution.
- We perform extensive performance analysis on SotA embedded IoT GPU (Nvidia Jetson NX [28]) to demonstrate the execution

The rest of this paper continues as follows: Sect. 2 provides an overview of related work in the field of human pose estimation, high-resolution networks, and scalable neural networks. Section 3 details the EfficientHRNet architecture and

[1] The source code of EfficientHRNet has been provided here: https://github.com/TeCSAR-UNCC/EfficientHRNet.

[2] Bottom-up implementation reported in [13].

the novel joint scaling formulation. Section 4 provides extensive results and comprehensive analysis on the challenging COCO dataset, comparisons to other SotA and real-time approaches, and a qualitative assessment of EfficientHRNet's scaling. Finally, Sect. 5 presents concluding remarks.

2 Related work

2.1 Top-down methods

Top-down methods rely on first identifying all the persons in an image using a detector, and then detecting keypoints for a single person within a defined bounding box. These single person [2, 14, 27, 40, 58, 60, 64] and multi-person [10, 17, 24, 26, 45] pose estimation methods often generate person bounding boxes using object detector [11, 12, 36, 48]. Regional Multi-Person Pose Estimation [17] adds symmetric spatial transformer network on top of single person pose estimator stacked hourglass network [40] to get high-quality regions from inaccurate bounding boxes, then detects poses using parametric non-maximum suppression.

2.2 Bottom-up methods

Bottom-up methods [4, 5, 25, 32, 33, 39, 44, 46] detect identity-free keypoints in an image and group them into persons using various keypoints grouping techniques. Methods like [25, 46] perform grouping by integer linear program and non-maximum suppression. This allows for faster inference times compared to top-down methods with almost similar accuracies. Other methods further improve upon prediction time by using greedy grouping techniques, along with other optimizations, as seen in [4, 5, 33, 39, 44]. For instance, OpenPose [4, 5] is a multi-stage network where one branch detects keypoints in the form of heatmaps, while the other branch generates Part Affinity Fields that are used to associate keypoints with each other. Grouping is done by calculating the line integral between all keypoints and grouping the pair that has the highest integral. Lightweight OpenPose [43] replaces larger backbone with MobileNet [22] to achieve real-time performance with fewer parameters and FLOPs while compromising on accuracy. PifPaf [33] uses Part intensity fields to detect body parts and Part associative fields for associating parts with each other to form human poses. In [39], a stacked hourglass network [40] is used both for predicting heatmaps and grouping keypoints. Grouping is done by assigning each keypoint with an embedding, called a tag, and then associating those keypoints based on the L_2 distance between the tag vectors. In this paper, we mainly focus on a highly accurate, end-to-end multi-person pose estimation method as in [39].

2.3 Top-down vs bottom-up

While both top-down and bottom-up approaches can be applied to the domain of multi-person pose estimation, the way they function is inherently different. While bottom-up methods are designed specifically for end-to-end multi-person pose estimation, most top-down approaches require multiple instances and the use of external detectors, and are generally not end-to-end in nature. This makes direct quantitative comparisons between these two approaches impractical. As such, this paper focuses primarily on the domain of bottom-up multi-person pose estimation.

2.4 Multi-scale high-resolution networks

Feature pyramid networks augmented with multi-scale representations are widely adopted for complex and necessary computer vision applications like segmentation and pose estimation [7, 8, 10, 36, 63]. Recovering high-resolution feature maps using techniques like upsampling, dilated convolution, and deconvolution are also widely popular for object detection [36], semantic segmentation [1, 9, 41, 49, 61, 65] and pose estimation [3, 10, 25, 30, 40, 46, 63]. Moreover, there are several works that focus on generating high-resolution feature maps directly [13, 23, 52, 54, 55, 59, 67]. HRNet [54, 55] proposes to maintain high-resolution feature maps throughout the entire network. HRNet consists of multiple branches with different resolutions across multiple stages. With multi-scale fusion, HRNet is able to generate high resolution feature maps and has found its application in object detection, semantic segmentation, and pose estimation [54, 55, 59], achieving remarkable accuracy. Recently, DSPNet [66] is proposed for lightweight single-person pose estimation. EfficientNet [56] based, it has a pyramid architecture using lightweight up-sampling unit and achieves high accuracy, becoming the SotA top-down approach. Following HRNet, HigherHRNet for multi-person pose estimation [13] is proposed which uses HRNet as base network to generate high resolution feature maps, and further adds a deconvolution module to predict accurate, high-quality heatmaps. HigherHRNet achieves SotA accuracy on the COCO dataset [37], surpassing all existing bottom-up methods. In this paper, we adopt the principles of HigherHRNet [13] for generating high-resolution feature maps with multi-scale fusion for predicting high quality heatmaps.

2.5 Model scaling

Previous works on bottom-up pose estimation [4, 5, 13, 39, 40, 54] often rely on large backbone networks, like ResNet [21] or VGGNet [53], or large input resolutions and multi-scale training for achieving SotA accuracy. Recent works [13, 54] show that increasing the channel dimension of

otherwise identical models can further improve accuracy. EfficientNet [56] and RegNet [47] show that by jointly scaling network width, depth, and input resolution, better efficiency for image classification can be achieved compared to previous SotA networks using much larger models. More recently, EfficientNet's lite models remove elements, such as squeeze and excite and swish layers, to make the network more hardware friendly. Inspired by EfficientNet, EfficientDet [57] proposes a compound scaling method for object detection along with efficient multi-scale feature fusion. We observe that there is a lack of an efficient scaling method for multi-person pose estimation, especially for embedded devices. Lightweight pose estimation models which are scalable and comparatively accurate are needed for computer vision applications which focus on real-time performance. Our proposed compound scaling, also inspired by EfficientNet, is a method that jointly scales the width, depth, and input resolution of EfficientHRNet, as well as the repetition within the high-resolution modules, explained in Sect. 3. In addition, this compound scaling allows our EfficientNet backbone to scale below the baseline B0, creating even lighter weight models.

2.6 Real-time pose estimation

While most work in the field focuses on accuracy in isolation, some recent works have been developed that shift the focus more to real-time inference. In [62], focus is placed on real-time execution using a densely connected residual module and high-resolution feature maps, similar to [54], for accurate and lightweight single person pose estimation able to achieve real-time performance with an impressive 39 FPS on an Nvidia 1080TI. In [43], OpenPose [4] is modified to use a MobileNet [22] backbone and fewer refinement stages, creating a multi-person bottom-up model that achieves 28 FPS using the Intel OpenVINO Toolkit [42] on an Intel NUC 6i7KYB. Nvidia has also been focusing on real-time inference, releasing trt_pose [29], a single person pose estimation model optimized with TensorRT and DeepStream [51], achieving up to 251 FPS on the Nvidia Jetson Xavier [16].

3 EfficientHRNet

We have developed a family of lightweight, scalable networks for real-time multi-person human pose estimation called EfficientHRNet. This section gives an overview of EfficientHRNet and introduces the formulation for the compound scaling of EfficientHRNet's sub-networks.

3.1 Network architecture and formulation

EfficientHRNet, shown in Fig. 2, comprises of three sub-networks: (1) backbone network, (2) high-resolution network, and (3) heatmap prediction network.

3.1.1 Backbone network

The first stage of EfficientHRNet is the backbone, consisting of a modified of EfficientNet [56] altered to scale below the baseline, as discussed in Sect. 3.2. The backbone outputs four different resolution feature maps of decreasing resolutions $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$ the size of the input image. These feature maps are passed into the main body of the network, called the High-Resolution Network.

3.1.2 High-resolution network

The high-resolution network is inspired by HRNet [54, 55] and HigherHRNet [13]. Borrowing the principles of these higher resolution networks brings two major advantages:

1. By maintaining multiple high-resolution feature representations throughout the network, heatmaps with a higher degree of spatial precision are generated.
2. Repeated multi-scale fusions allow for high-resolution feature representations to inform lower-resolution representations, and vice versa, resulting in robust multi-resolution feature representations that are ideal for multi-person pose estimation.

Figure 2 presents a detailed architecture illustration of EfficientHRNet. It shows the three sub-networks: the backbone network, the high-resolution network, and the Heatmap Prediction Network. It also provides equations showing how the network scales the input resolution R_{input} and the width of feature maps W_{b_n}, which will be further explained in Sect. 3.2.

The High-Resolution Network has three stages s_1, s_2, and s_3 containing four parallel branches b_1, b_2, b_3, and b_4 of different resolutions. The first stage s_1 starts with two branches b_1 and b_2, with each consecutive stage adding an additional branch, until all four branches are present in s_3. These four branches each consist of high resolution modules with a width of W_{b_n}. Each branch b_n contains feature representations of decreasing resolutions that mirror the resolutions output by the Backbone Network, as shown in Fig. 2 and the following expression:

$$W_{b_n} \times \frac{R_{\text{input}}}{2^n + 1} \tag{1}$$

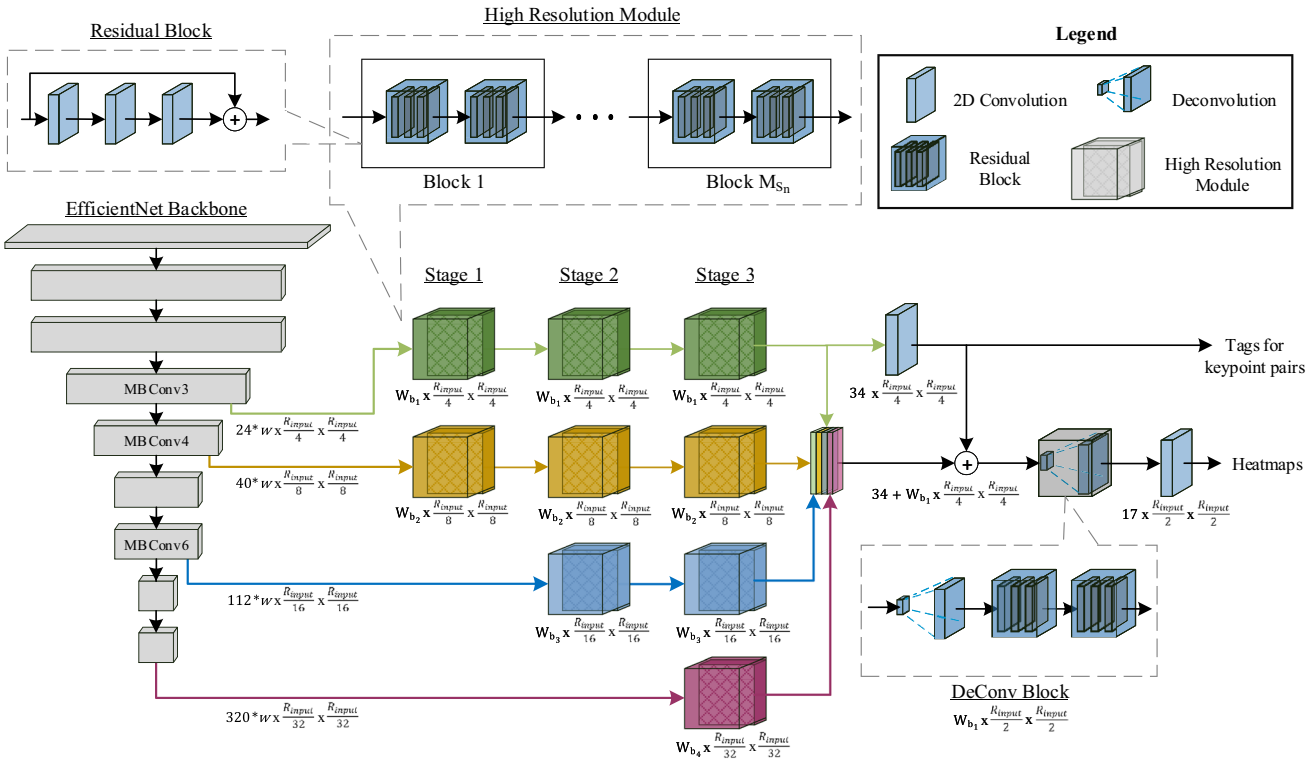

Fig. 2 A detailed illustration of the EfficientHRNet architecture. Consisting of a backbone EfficientNet, a high-resolution network with three stages and four branches (denoted by different colors), and a Heatmap Prediction Network. EfficinetHRNet is completely scalable, allowing network complexity to be customized for target applications

For instance, stage 2 (s_2) has three branches of resolutions $\frac{1}{4}$, $\frac{1}{8}$, and $\frac{1}{16}$ of the original input image resolution and a width W_{b_n}. Moreover, each high resolution module is made up of a number of blocks, M_{s_n}, each containing two residual blocks, of which each perform three convolution operations with a residual connection.

3.1.3 Heatmap prediction network

The Heatmap prediction network is used to generate human keypoint predictions. To predict more accurate heatmaps, a DeConv block is added on top of the High-Resolution Network, as proposed in [13]. Transposed convolution is used to generate high quality feature maps which are $\frac{1}{2}$ the original input resolution. The input to the DeConv block is the concatenation of the feature maps and predicted heatmaps from the High-Resolution Network, as shown below:

$$34 + W_{b_1} \times \frac{R_{\text{input}}}{4} \times \frac{R_{\text{input}}}{4} \tag{2}$$

Two residual blocks are added after the deconvolution to refine the up-sampled feature maps. After the DeConv block, 1×1 convolution is used to predict heatmaps and tagmaps in a similar fashion to [39], the feature map size of each shown below:

$$\begin{aligned} T_{\text{size}} &= 34 \times \frac{R_{\text{input}}}{4} \times \frac{R_{\text{input}}}{4} \\ H_{\text{size}} &= 17 \times \frac{R_{\text{input}}}{2} \times \frac{R_{\text{input}}}{2} \end{aligned} \tag{3}$$

The grouping process clusters keypoints into multiple persons by grouping keypoints whose tags have minimum L_2 distance. Like [13], the high-resolution network is scale-aware and uses multi-resolution supervision for heatmaps during training to allow the network to learn with more precision, even for small-scale persons. From the ground truth, heatmaps for different resolutions are generated to match the predicted keypoints of different scales. Thus, the final heatmaps loss is the sum of mean squared errors for all resolutions. However, as high resolutions tagmaps do not converge well, tagmaps are trained on a resolution $\frac{1}{4}$ of the original input resolution, as in [39].

3.2 Compound scaling method

This section details the compound scaling methodology, which jointly scales all parts of EfficientHRNet, as seen in

Table 1 Efficient scaling configs for EfficientHRNet

Model	Input size (R_{input})	Backbone network	Width per branch ($W_{b_1}, W_{b_2}, W_{b_3}, W_{b_4}$)	Blocks per stage ($M_{s_2}, M_{s_3}, M_{s_4}$)	Tags (T_{size})	Heat-maps (H_{size})
H_0 ($\phi = 0$)	512	$B0$	32, 64, 128, 256	1, 4, 3	128	256
H_{-1} ($\phi = -1$)	480	B_{-1}	26, 52, 103, 206	1, 3, 3	120	240
H_{-2} ($\phi = -2$)	448	B_{-2}	21, 42, 83, 166	1, 2, 3	112	224
H_{-3} ($\phi = -3$)	416	B_{-3}	17, 34, 67, 133	1, 1, 3	104	208
H_{-4} ($\phi = -4$)	384	B_{-4}	14, 27, 54, 107	1, 1, 2	96	192

Table 2 Compact EfficientNet performance on ImageNet and CIFAR-100 datasets

Model	Input size	FLOPs	ImageNet		CIFAR-100	
			Params	Top-1	Params	Top-1
B0	224	0.4B	5.3M	75	4.1M	81.9
B_{-1}	195	0.3B	4.5M	73.8	3.5M	81.4
B_{-2}	170	0.2B	3.4M	71.3	2.5M	79.8
B_{-3}	145	0.1B	2.8M	68.5	1.9M	78.2
B_{-4}	128	0.05B	1.3M	65.6	1.3M	74.3

Fig. 2 and Table 1. The aim of EfficientHRNet is to provide a family of models optimized for both accuracy and efficiency, which can be scaled to meet a diverse set of memory and compute constraints.

Previous works on bottom-up human pose estimation and semantic segmentation mostly scale the base network by using bigger backbone networks like ResNet [21] and VGG-Net [53], using large input image sizes, or using multi-scale training to achieve high accuracies. However, these methods rely on scaling only a single dimension, which has limited effectiveness. Recent works [47, 56] show notable performance on image classification by jointly scaling the width, depth, and input image resolution. Inspired by EfficientNet, EfficientDet [57] proposes a similar compound scaling method for object detection, which jointly scales the backbone network, multi-scale feature network, and the object detector network. We propose a heuristic-based compound scaling methodology for computer vision applications, specifically bottom-up human pose estimation and semantic segmentation, using EfficientHRNet. Based on [56], EfficientHRNet's methodology uses a scaling coefficient ϕ to jointly scale the Backbone Network, the High-Resolution Network, and Task-Specific Head. More precisely, the EfficientNet backbone is scaled below the baseline and the rest of EfficientHRNet is scaled down to maintain near SotA accuracy while creating lightweight and flexible networks.

3.2.1 Backbone network

The same width and depth scaling coefficients are maintained as in EfficientNet [56]. To meet the demands of running models on constrained devices, a new formulation for scaling EfficientNet below the baseline and into a more compact model is provided.

Starting with the baseline EfficientNet-B0 scaling coefficients:

$$\begin{aligned} \text{depth} &: d = 1.2^{\phi} \\ \text{width} &: w = 1.1^{\phi} \\ \text{resolution} &: r = 1.15^{\phi} \end{aligned} \tag{4}$$

ϕ, i.e. $\phi = -1, -2, -3, -4$, is inverted to calculate the scaling multipliers for the compact EfficientNet models, which is symbolized as B_{-1}, B_{-2}, B_{-3} and B_{-4} respectively. As an example, to take the baseline resolution, 224, and scale it down for our B_{-1} model, we would take r, from (4), with $\phi = -1$. This would result in a resolution scaling coefficient of 1.15^{-1}, i.e. 0.87, leaving a scaled resolution size of ceil$(224 * 0.87) = 195$. This pattern repeats for B_{-2} through B_{-4}, and can be seen in Table 2. We train these compact EfficientNet models (B_{-1} to B_{-4}) on ImageNet and use the resulting models for the Backbone Network in EfficientHRNet.

3.2.2 High-resolution network

The high-resolution network has three stages and four branches with four different feature map sizes. Each branch n also has a different width W_{b_n} and our baseline H_0 model has a width of 32, 64, 128, and 256 for each branch respectively. We selectively pick a width scaling factor of 1.25 and scale down the width using the following equation:

$$W_{b_n} = (n \cdot 32) \cdot (1.25)^{\phi}, \tag{5}$$

where n is a particular branch number and ϕ is the compound scaling coefficient.

Furthermore, within each stage, each high resolution module has multiple blocks M_{s_n} which repeat a number of times, as seen in Table 1. In our baseline EfficientHRNet H_0 model, blocks within each stage repeat 1, 4, and 3 times respectively. We found that the number of repetitions in stage 3 had the largest impact on accuracy. Therefore, the number of repetitions within a high resolution module M_{s_2} decreases linearly as the models are scaled down, starting with stage 2 until reaching a single repetition and then moving on to stage 3, as shown in Table 1.

3.2.3 Heatmap prediction network

The DeConv block is scaled in the same manner as the width of the high resolution network (5). The Heatmap prediction network outputs tags and heatmaps whose width remains fixed across all models.

3.2.4 Input image resolution

The EfficientNet layers downsample the original input image resolution by 32 times. Thus, the input resolution of EfficientHRNet must be dividable by 32, and is linearly scaled down as shown in the following equation:

$$R_{\text{input}} = 512 + 32 \cdot \phi. \tag{6}$$

The final result of this compound scaling methodology on EfficientHRNet H_0 to H_{-4} can be seen in Table 1.

4 Experimental results

This section evaluates our method for scaling EfficientNet below the baseline through classification on the popular ImageNet [15] and CIFAR-100 [34] datasets. Then, an exhaustive evaluation of five different EfficientHRNet models is conducted on the challenging COCO [37] dataset and compared to SotA methods. Additional, metrics on real-time inference are reported using the Nvidia Jetson NX Xavier and compared to SotA lightwieght approaches. Finally, a qualitative evaluation of EfficientHRNet is presented, illustrating both where the models excel and where they fall short.

4.1 Classification for compact EfficientNet

4.1.1 Dataset

ImageNet [15] has been a long time standard benchmark for object classification and detection thanks to its annual contest, the ImageNet Large Scale Visual Recognition Challenge, that debuted in 2010. The challenge uses a subset of the full dataset with over a million images spread out over 1000 object classes. For training, validating, and testing purposes, the trimmed ImageNet is divided into three sets: 800 K images will be used for training the network, 150 K will be used for validation after each epoch, and 50 K will be used for testing the fully trained model. CIFAR-100 [34] consists of 100 object classes each with 500 images for training, and 100 for testing. This relatively small dataset helps illuminate our lightweight models, which start to struggle with the larger ImageNet as ϕ decreases, designed for resource constrained devices that might not need to classify as many object classes.

4.1.2 Training

We use random rotation, random scale, and random aspect ratio to crop the input images to the desired resolutions based on the current EfficientNet model. Color jitter was also used to randomly change the brightness, contrast, saturation, and hue of the RGB channels using principle component analysis [35]. The images are then normalized using per channel mean and standard deviation. Each model was trained using Stochastic Gradient Descent with Momentum [50] and a weight decay of 1e−4. The weights were initialized using the Xavier algorithm [20] and underwent five warm-up epochs with a learning rate of 1e−4 that increased linearly until it reached 0.05. The networks were then trained for an additional 195 epochs and followed the step decay learning rate scheduler [19] that reduces the learning rate by a factor of 10 every 30 epochs.

4.1.3 Testing

Compact EfficientNet models were tested for accuracy based on their respective test sets. For a fair comparison, the number of ImageNet test samples were reduced to 10,000 to match the test set of CIFAR-100, where the batch size is 1. These results can be seen in Table 2.

4.1.4 Results on ImageNet and CIFAR-100

Looking at B_{-1} there is a 15% reduction in parameters and 25% reduction in operations, yet an accuracy drop of only 1.2% and 0.5% on ImageNet and CIFAR-100 respectively. More impressively, B_{-2} sees a 35–40% reduction in parameters and a 50% reduction in operations, yet only a 3.7% and 2.1% drop in accuracy. This minor accuracy loss is negligible compared to the massive reduction in model size and computation, allowing for much faster inference and deployment on low-power and resource constrained devices. In the most extreme, B_{-4} shows a parameter reduction of 68–75%

and a 87.5% decrease in operations while having an accuracy drop of 9.4% and 7.6% on ImageNet and CIFAR-100. While the accuracy drop is a bit more significant here, the massive reduction in computation allows for much more flexibility when it comes to deployment in systems where a lightweight approach is needed. This gives us a solid foundation on which to build EfficientHRNet.

4.2 2D human pose estimation for EfficientHRNet

4.2.1 Dataset

COCO [37] consists of over 200 K images with 250 K person instances, each annotated with 17 keypoints. COCO is divided into three sets, train, val, and test, which have 57 K, 5 K, and 40 K images respectively. Additionally, test-dev is a subset of test with 20 K images and is used for fair comparison with other works, where possible. COCO evaluation metrics use mean average precision (AP) and are detailed on the COCO website.[3]

4.2.2 Training

We use random rotation, random scale, and random translation for data augmentation. Following [13], we generate two ground truth heatmaps of different sizes, $\frac{1}{2}$ and $\frac{1}{4}$ of the original input size respectively. Each EfficientHRNet model is trained using Adam optimizer [31] and weight decay of 1e−4. While we saw little difference in accuracy between using Adam and SGD with momentum in our initial testing, Adam was selected for its higher speed of convergence and overall effect on training time. All models from H_0 to H_{-4} are trained for a total of 300 epochs with a base learning rate of 1e−3, decreasing to 1e−4 and 1e−5 at 200th and 260th epochs respectively. To maintain balance between heatmap loss and grouping loss, we weight the losses at 1 and 1e−3 respectively.

4.2.3 Testing

Models are tested using both single scale and multi-scale heatmaps, as is common. Following [39], the output detection heatmaps across different scales are averaged and the tags concatenated into higher dimension tags, making the models considerably more scale-invariant.

4.2.4 Results on COCO2017 test-dev

Table 3 compares EfficientHRNet with other works on COCO test-dev set. As explained in Sect. 2.3, top-down methods, such as [17, 54, 68], are inherently not end-to-end. As such, we limit our comparisons to the domain of bottom-up multi-person human pose estimation. The baseline H_0 model with single-scale testing serves as an efficient and accurate model for bottom-up methods as it is almost comparable to HRNet [13] in accuracy, losing by only 0.1%, while having 18% less parameter and 34% fewer FLOPs. H_0 outperforms Hourglass [40] in both single-scale and multi-scale testing by 7.4% and 1.6% respectively, with H_0 remarkably having about $\frac{1}{10}$ the model size and number of FLOPs as Hourglass. The highest performing of all models on COCO test-dev, HigherHRNet [13], beats H_0 in accuracy by 4.4%, but at the cost of nearly triple the model size and more than 6x the computation. In all cases where H_0 loses in accuracy, it more than makes up for it in a reduction in parameters and operations. Additionally, our H_{-1} model, with only 16M parameters and 14.2B FLOPs, outperforms both OpenPose [4, 5] and Hourglass [40], demonstrating EfficientHRNet's

Table 3 Comparisons with SotA bottom-up methods on COCO2017 test-dev dataset

Method	Backbone	Input size	# Params	FLOPs	AP
w/o multi-scale test					
OpenPose	–	–	25.94M	160B	61.8
Hourglass	Hourglass	512	277.8M	206.9B	56.6
PersonLab	ResNet-152	1401	68.7M	405.5B	66.5
PifPaf	ResNet-152	–	–	–	66.7
HRNet	HRNet-W32	512	28.5M	38.9B	64.1
HigherHRNet	HRNet-W32	512	28.6M	47.9B	66.4
HigherHRNet	HRNet-W48	640	63.8M	154.3B	68.4
H_0	$B0$	512	23.3M	25.6B	64.0
H_{-1}	B_{-1}	480	16M	14.2B	59.1
H_{-2}	B_{-2}	448	10.3M	7.7B	52.8
H_{-3}	B_{-3}	416	6.9M	4.2B	44.5
H_{-4}	B_{-4}	384	3.7M	2.1B	35.5
w/ multi-scale test					
Hourglass	Hourglass	512	277.8M	206.9B	63.0
Hourglass	Hourglass	512	277.8M	206.9B	65.5
PersonLab	ResNet-152	1401	68.7M	405.5B	68.7
HigherHRNet	HRNet-W48	640	63.8M	154.3B	70.5
H_0	$B0$	512	23.3M	25.6B	67.1
H_{-1}	B_{-1}	480	16M	14.2B	62.3
H_{-2}	B_{-2}	448	10.3M	7.7B	55.0
H_{-3}	B_{-3}	416	6.9M	4.2B	45.5
H_{-4}	B_{-4}	384	3.7M	2.1B	39.7

Numbers for HRNet come from a bottom-up approach outlined in [13]

[3] http://cocodataset.org/#keypoints-eval.

Table 4 Comparisons with bottom-up methods on COCO2017 val dataset

Model	Input size	AP	# Params	FLOPs	FPS
PersonLab	1401	66.5	68.7M	405.5B	–
HRNet	512	64.4	28.5M	38.9B	–
HigherHRNet	512	67.1	28.6M	47.9B	6.68
Lightweight OpenPose	368	42.8	4.1M	9.0B	26
H_0 ($\phi = 0$)	512	64.8	23.3M	25.6B	22.95
H_{-1} ($\phi = -1$)	480	59.2	16M	14.2B	20.43
H_{-2} ($\phi = -2$)	448	52.9	10.3M	7.7B	24.53
H_{-3} ($\phi = -3$)	416	44.8	6.9M	4.2B	33.78
H_{-4} ($\phi = -4$)	384	35.7	3.7M	2.1B	50.96

Metrics and accuracy for HRNet come from a bottom-up approach outlined in [13] (FPS not reported). Lightweight OpenPose numbers were reported on the Intel NUC 6i7KYB. All other FPS results were preformed on the Nvidia Jetson NX Xavier [28]

efficiency and suitability for low-power and resource constrained devices.

As EfficientHRNet is scaled down using the compound scaling method mentioned in Sect. 3.2, we see somewhat minor drops in accuracy with significant drops parameters and FLOPs as compared to the baseline H_0 model. H_{-1} has 31.3% less parameters and 44.5% less FLOPs as compared to H_0 while only being 4.9% less accurate. Similarly, our lightest model H_{-4} is 84% smaller and has 91.7% less FLOPs, with a less than 45% drop in accuracy. Interestingly, EfficientHRNet is the only bottom-up pose estimator that is able to provide such lightweight models while still having accuracies that are comparable to SotA bottom-up methods, as illustrated by both Table 3 and Fig. 1. These results nicely show the validity of our approach to scalability and efficiency in EfficientHRNet.

4.2.5 Results on COCO2017 val

We report EfficientHRNet accuracy on COCO val, noting the number of parameters and FLOPs, and compare it with other bottom-up methods. In addition, to accurately assess suitability for real-time performance on embedded devices, we inference our models as well as one of our closest competitors, HigherHRnet [13], on the Jetson NX Xavier, first converting the models to ONNX and then inferencing in TensorRT. FPS results for Lightweight OpenPose are on an Intel NUC 6i7KYB as reported in [43]. Looking at Table 4, we can see that PersonLab is a very large network. With nearly 3 × as many parameters and 16 × as many operations as our most complex model H_0 it is too large to even run on the NX Xavier, despite an improvement of less than 2% accuracy. Still, the baseline H_0 model outperforms HRNet [54] with 0.4% more accuracy, 18% fewer parameters and 34% fewer FLOPs. H_{-2} and H_{-3} models outperform Lightweight OpenPose [43] in accuracy while having fewer FLOPs. H_{-4} has the worst accuracy of any model in Table 4. However, it boasts both the smallest model size and fewest number of operations, seeing an over 75% reduction from its lightest weight competitor. When looking at FPS, HigherHRNet becomes much less desirable, being the only model unable to achieve at least 20 FPS. H_0 is 3.4 × faster while only being 2.3% less accurate. Comparing to Lightweight OpenPose, H_0 is 22% more accurate while only being 2 FPS slower. Scaling down to H_{-3} reduces EfficientHRNet's accuracy lead by to only 2%, but increases throughput to be 1.3 × greater than Lightweight OpenPose. Our smallest model H_{-4} achieves an impressive 50 FPS, but at a substantial cost in accuracy. Note that we see an unusually high FPS drop in H_{-1}. This is due to H_{-1}'s input resolution and intermediate feature map sizes resulting in memory tiles that map poorly to the NX Xavier's Tensor Core architecture. In the following subsection, we provide further analysis and comparison regarding real-time execution on NX Xavier.

Table 5 Æ comparisons with lightweight bottom-up approaches

Model	AP	FPS	Efficiency	Æ
HigherHRNet	67.1	6.68	0.445	29.850
Lightweight OpenPose	42.8	26	0.578	24.738
H_0 ($\phi = 0$)	64.8	22.95	1.530	99.144
H_{-1} ($\phi = -1$)	59.2	20.43	1.362	80.630
H_{-2} ($\phi = -2$)	52.9	24.53	1.635	86.492
H_{-3} ($\phi = -3$)	44.8	33.78	2.252	100.89
H_{-4} ($\phi = -4$)	35.7	50.96	3.397	121.273

Lightweight OpenPose reported on Intel NUC 6i7KYB (45W). All others Nvidia Jetson NX Xavier (15W)

4.3 Real-time execution analysis on edge

Since real-time inference is highly dependent on the hardware utilized, we must account for more than just accuracy and throughput in our comparisons. To best compare accuracy and efficiency across differing platforms, we adopt the Accuracy · Efficiency (Æ) metric from [38]. Æ is simply the product of accuracy (measured in AP) and efficiency (measured in FPS per Watt). Table 5 shows how EfficientHRNet

compares when taking this into account. Note that Lightweight OpenPose reports results on Intel NUC 6i7KYB, which has a TDP of 45 Watts [43], while all other methods were measured on the Jetson NX Xavier with a maximum power draw of 15 Watts. We use these power numbers across all approaches in an attempt to create as fair a comparison as possible. EfficientHRNet greatly outperforms the competition in terms of Æ, with all models achieving an Æ score of over 80 while Lightweight OpenPose and HigherHRNet achieve scores of 25 and 30 respectively.

In terms of Æ, EfficientHRNet outperforms the competition between 3 × and 5 ×. This is largely due to the poor throughput of HigherHRNet and the relatively higher power NUC that Lightweight OpenPose reports on. HigherHRNet excels in accuracy and Lightweight OpenPose excels in FPS and model size, while EfficientHRNet is more equally balanced between accuracy, model size, throughput, and power consumption. This gives EfficientHRNet a leg up in terms of low-power, real-time inference, making its scalable models the new SotA for lightweight bottom-up human pose estimation for real-time edge applications.

4.4 Qualitative analysis

To further demonstrate how EfficientHRNet models perform in relation to one another, we present qualitative results on COCO. Figure 3 shows simple, medium, and complex examples for EfficientHRNet models H_0 to H_{-4}. We see that H_0 can accurately detect all but the most distant and occluded individuals. This accuracy is functionally identical to SotA models but is able to inference in real-time, making it immensely valuable for applications that require high accuracy but need to run in real-time or on low-power devices. Looking at H_{-1} we see that keypoints are accurately detected, but in medium and complex scenarios keypoint groupings become confused. Here, the confusion is minor enough that it can be filtered out with additional post processing, meaning that applications that require predicting complex scenarios on devices that can not fit H_0 can use H_{-1} with slightly decreased accuracy. For the medium scenario, there is also missed detection from the distant person occluded by the left-most surfboard, though such missed detections are relatively uncommon throughout the dataset. However, for simple scenarios there is little to no difference when compared with H_0. These qualities make H_{-1} a compelling model when using a device without enough memory resources for H_0 and when a minor amount of error is acceptable, or for applications that will only deal with simple scenarios. H_{-2} looks a lot like H_{-1}, but the confusion is worse, with multiple keypoint grouping being detected for a single person, and this even extends to simple scenarios. This would again require additional post processing, depending on the application. H_{-3} and H_{-4} follow the same pattern, with confusion continuing to get worse. Again, we see that actual keypoint detections themselves are fairly accurate, and the greatly reduced model size and computational complexity open up a wide range of additional devices capable of real-time performance. This makes the smaller models extremely compelling for real-time applications that can afford a certain amount of error but require the use of highly resource constrained devices, particularly in the case of simple scenarios. This analysis helps visualize the relationship between detected keypoints and model size, and shows the overall affect on accurate human pose prediction as we move towards smaller models. This further validates EfficientHRNet as a family of high accuracy and efficient models capable of real-time 2D human pose estimation for a variety of embedded and resource constrained devices.

5 Conclusion

In this paper, we have presented EfficientHRNet, a family of scalable networks for high-resolution and efficient bottom-up multi-person pose estimation made for real-time execution on low-power edge devices. EfficientHRNet unifies the principles of SotA EfficientNet [56] and HRNet [54] to create a network architecture for lightweight real-time human pose estimation, and proposes a new compound scaling method that jointly scales down the input resolution, backbone network, and high-resolution feature network. EfficientHRNet is not only more efficient than all other bottom-up human pose estimation methods, but it can maintain accuracy competitive with SotA models on the challenging COCO dataset. Remarkably, EfficientHRNet can achieve this near SotA accuracy with fewer parameters and less computational complexity than other bottom-up multi-person pose estimation networks, all while being able to achieve 23 FPS on an Nvidia Jetson NX Xavier.

Fig. 3 Qualitative results for EfficientHRNet models on COCO2017 test. Left to right: simple, medium and complex examples

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder–decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
2. Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. CoRR arXiv:1609.01743 (2016)
3. Bulat, A., Tzimiropoulos, G.: Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. CoRR arXiv:1703.00862 (2017)
4. Cao, Z., Hidalgo, G., Simon, T., Wei, S., Sheikh, Y.: Openpose: realtime multi-person 2d pose estimation using part affinity fields. CoRR arXiv:1812.08008 (2018)

5. Cao, Z., Simon, T., Wei, S., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. CoRR arXiv:1611.08050 (2016)
6. Chen, K., Gabriel, P., Alasfour, A., Gong, C., Doyle, W.K., Devinsky, O., Friedman, D., Dugan, P., Melloni, L., Thesen, T., Gonda, D., Sattar, S., Wang, S., Gilja, V.: Patient-specific pose estimation in clinical environments. IEEE J. Transl. Eng. Health Med. **6**, 1–11 (2018). https://doi.org/10.1109/JTEHM.2018.2875464
7. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. CoRR arXiv:1606.00915 (2016)
8. Chen, L., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. CoRR arXiv:1511.03339 (2015)
9. Chen, L., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. CoRR arXiv:1802.02611 (2018)
10. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. CoRR arXiv:1711.07319 (2017)
11. Cheng, B., Wei, Y., Shi, H., Feris, R.S., Xiong, J., Huang, T.S.: Decoupled classification refinement: Hard false positive suppression for object detection. CoRR arXiv:1810.04002 (2018)
12. Cheng, B., Wei, Y., Shi, H., Feris, R.S., Xiong, J., Huang, T.S.: Revisiting RCNN: on awakening the classification power of faster RCNN. CoRR arXiv:1803.06799 (2018)
13. Cheng, B., Xiao, B., Wang, J., Shi, H., Huang T.S., Zhang, L.: HigherHRNet: scale-aware representation learning for bottom-up human pose estimation. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5385–5394. https://doi.org/10.1109/CVPR42600.2020.00543
14. Dantone, M., Gall, J., Leistner, C., Van Gool, L.: Human pose estimation using body parts dependent joint regressors. In: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3041–3048. https://doi.org/10.1109/CVPR.2013.391
15. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848
16. Ditty, M., Karandikar, A., Reed, D.: Nvidia xavier soc (2018)
17. Fang, H., Xie, S., Tai, Y., Lu, C.: RMPE: regional multi-person pose estimation. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2353–2362. https://doi.org/10.1109/ICCV.2017.256
18. Fang, Z., López, A.M.: Intention recognition of pedestrians and cyclists by 2d pose estimation. IEEE Trans. Intell. Transport. Syst. **21**(11), 4773–4783 (2020). https://doi.org/10.1109/TITS.2019.2946642
19. Ge, R., Kakade, S.M., Kidambi, R., Netrapalli, P.: The step decay schedule: A near optimal, geometrically decaying learning rate procedure. CoRR arXiv:1904.12838 (2019)
20. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Y.W. Teh, M. Titterington (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, *Proceedings of Machine Learning Research*, vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (2010). http://proceedings.mlr.press/v9/glorot10a.html
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR arXiv:1512.03385 (2015)
22. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: efficient convolutional neural networks for mobile vision applications (2017)
23. Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., Weinberger, K.Q.: Multi-scale dense convolutional networks for efficient prediction. CoRR arXiv:1703.09844 (2017)
24. Huang, S., Gong, M., Tao, D.: A coarse-fine network for keypoint localization. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3047–3056. https://doi.org/10.1109/ICCV.2017.329
25. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: Deepercut: A deeper, stronger, and faster multi-person pose estimation model. CoRR arXiv:1605.03170 (2016)
26. Iqbal, U., Gall, J.: Multi-person pose estimation with local joint-to-person associations. CoRR arXiv:1608.08526 (2016)
27. Jain, A., Tompson, J., Andriluka, M., Taylor, G. W., Bregler, C.: Learning human pose estimation features with convolutional networks. In: Proceedings of the 2nd international conference on learning representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings
28. Jetson xavier nx developer kit (2020). https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit. Accessed 8 Nov 2020
29. John: trt_pose. https://github.com/NVIDIA-AI-IOT/trt_pose. Accessed 9 Nov 2020
30. Ke, L., Chang, M., Qi, H., Lyu, S.: Multi-scale structure-aware network for human pose estimation. CoRR. arXiv:1803.09894 (2018)
31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR arXiv:1412.6980 (2015)
32. Kocabas, M., Karagoz, S., Akbas, E.: Multiposenet: Fast multi-person pose estimation using pose residual network. CoRR arXiv:1807.04067 (2018)
33. Kreiss, S., Bertoni, L., Alahi, A.: Pifpaf: Composite fields for human pose estimation. CoRR arXiv:1903.06593 (2019)
34. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009). http://www.cs.toronto.edu/~kriz/cifar.html
35. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems: Volume 1, NIPS'12, pp. 1097–1105. Curran Associates Inc., Red Hook, NY, USA (2012)
36. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. CoRR arXiv:1612.03144 (2016)
37. Lin TY. et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
38. Neff, C., Mendieta, M., Mohan, S., Baharani, M., Rogers, S., Tabkhi, H.: Revamp2t: real-time edge video analytics for multi-camera privacy-aware pedestrian tracking. IEEE Internet Things J. **7**(4), 2591–2602 (2020). https://doi.org/10.1109/JIOT.2019.2954804
39. Newell, A., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. CoRR arXiv:1611.05424 (2016)
40. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. CoRR arXiv:1603.06937 (2016)
41. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. CoRR arXiv:1505.04366 (2015)
42. Openvino toolkit. https://software.intel.com/en-us/openvino-toolkit. Accessed 8 Nov 2020.
43. Osokin, D.: Real-time 2d multi-person pose estimation on CPU: lightweight openpose. CoRR arXiv:1811.12004 (2018)

44. Papandreou, G., Zhu, T., Chen, L., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. CoRR arXiv:1803.08225 (2018)
45. Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., Murphy, K.P.: Towards accurate multi-person pose estimation in the wild. CoRR arXiv:1701.01779 (2017)
46. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P.V., Schiele, B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. CoRR arXiv:1511.06645 (2015)
47. Radosavovic, I., Kosaraju, R.P., Girshick, R., He K., Dollár, P.: Designing network design spaces. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10425–10433. https://doi.org/10.1109/CVPR42600.2020.01044
48. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR arXiv:1506.01497 (2015)
49. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR arXiv:1505.04597 (2015)
50. Ruder, S.: An overview of gradient descent optimization algorithms. CoRR arXiv:1609.04747 (2016)
51. Saharan, A.: Creating a human pose estimation application with nvidia deepstream (2020). https://developer.nvidia.com/blog/creating-a-human-pose-estimation-application-with-deepstream-sdk/. Accessed 8 Nov 2020
52. Saxena, S., Verbeek, J.: Convolutional neural fabrics. CoRR arXiv:1606.02492 (2016)
53. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
54. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5686–5696. https://doi.org/10.1109/CVPR.2019.00584
55. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. CoRR arXiv:1904.04514 (2019)
56. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. CoRR arXiv:1905.11946 (2019)
57. Tan, M., Pang R., Le, Q.V.: EfficientDet: scalable and efficient object detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079
58. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. CoRR arXiv:1312.4659 (2013)
59. Wang, J., et al.: Deep high-resolution representation learning for visual recognition. In: Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence. https://doi.org/10.1109/TPAMI.2020.2983686
60. Wei, S., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. CoRR arXiv:1602.00134 (2016)
61. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. CoRR arXiv:1807.10221 (2018)
62. Yang, L., Qin, Y., Zhang, X.: Lightweight densely connected residual network for human pose estimation. J Real-Time Image Proc **18**, 825–837 (2021)
63. Yang, W., Li, S., Ouyang, W., Li, H., Wang, X.: Learning feature pyramids for human pose estimation. CoRR arXiv:1708.01101 (2017)
64. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. CVPR **2011**, 1385–1392 (2011)
65. Zhang, Z., Zhang, X., Peng, C., Cheng, D., Sun, J.: Exfuse: enhancing feature fusion for semantic segmentation. CoRR arXiv:1804.03821 (2018)
66. Zhong, F., Li, M., Zhang, K., Hu, J., Liu, L.: Dspnet: a low computational-cost network for human pose estimation. Neurocomputing **423**, 327–335 (2021)
67. Zhou, Y., Hu, X., Zhang, B.: Interlinked convolutional neural networks for face parsing. CoRR arXiv:1806.02479 (2018)
68. Zhu, H., Qiao, Y., Xu, G., Deng, L., Yu, Y.F.: Dspnet: a lightweight dilated convolution neural networks for spectral deconvolution with self-paced learning. IEEE Trans. Ind. Inform. **16**(12), 7392–7401 (2020). https://doi.org/10.1109/TII.2019.2960837

Christopher Neff received a B.S. in Computer Engineering from the University of North Carolina at Charlotte, Spring of 2019. He is currently pursuing a Ph.D. in Electrical Engineering under the mentorship of Dr. Hamed Tabkhi. His area of research is focused on enabling real-time artificial intelligence on low-power edge devices. This includes architecture and algorithmic co-design, system-level optimization, and end-to-end development with regards to power, scalability, and latency requirements.

Aneri Sheth received a B.Tech. in Information and Communication Technology from Ahmedabad University, India, Spring of 2018. She recently received M.S. in Electrical Engineering from the University of North Carolina at Charlotte, Fall of 2020. Her area of interest includes real-time computer vision and algorithm design and development for AI research and production.

Steven Furgurson received a B.S. in Computer Engineering from the University of Carolina at Charlotte, Spring of 2019 and his M.S in Electrical Engineering from the University of North Carolina at Charlotte, Summer of 2020. His area of research is focused on solving real problems related to the intersection of computer architecture engineering and machine learning. In particular: Hardware modeling of deep machine learning algorithms such as convolutional and recurrent neural networks, and the efficiency improvement of hardware accelerator of compute-intensive algorithms though using reconfigurable hardware architectures.

John Middleton received a B.S. in Computer Engineering from the University of North Carolina at Charlotte, Spring of 2020. He is currently starting a Master's in Electrical Engineering at the University of North Carolina at Charlotte under the mentorship of Dr. Hamed Tabkhi. His area of research is yet to be decided, but currently focuses on the real-time aspects of artificial intelligence, particularly pertaining to the systems and execution within the low-power environment.

Hamed Tabkhi is an Assistant Professor in the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, USA. He was a post-doctoral research associate at Northeastern University. Hamed Tabkhi received his Ph.D. degree in 2014 from Northeastern University under the direction of Prof. Gunar Schirner. Overall, his research focuses on transformative computer systems and architecture for cyber-physical, real-time streaming and emerging machine learning applications.