UDC 004.8 **doi: 10.26906/SUNZ.2023.2.075**

Daniil Vyshnivskyi, Oleksii Liashenko, Nataliia Yeromina

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

HUMAN POSE ESTIMATION SYSTEM USING DEEP LEARNING ALGORITHMS

Abstract. The purpose of this work is the software implementation of neural network that can solve problem of Human Pose Estimation. With rapid improvements of neural network models and computing resources over last 10 years it's become possible to automate a lot of processes, carry out research and improve quality of life. One of the directions is Computer Vision: it allows to recognize objects, track motions, image segmentation, facial recognition etc. Human pose estimation is the part of Computer Vision area of research. It allows to capture human pose from a video or an image and have many uses in medicine, sport, augmented reality, video games etc. Therefore, the goal of this work is to find and optimize algorithm, that is relatively accurate, for identifying and classifying the joints in the human body. To achieve the goal, the following tasks were solved: current methods and technologies that is commonly used to solve problem of human pose estimation were reviewed and analyzed, artificial neural networks were used as a mathematical apparatus for the model, software implementation for human pose estimation was developed and tested, outputs from model were analyzed and evaluated, results and conclusion were formulated.

Keywords: human pose estimation; classification of objects; object detection; convolutional neural networks.

Introduction

Human Pose Estimation (HPE) is a fundamental and rapidly evolving area of research within the field of computer vision. It aims to estimate the positions and orientations of human body joints in images or videos, enabling machines to understand and analyze human body poses. The ability to perceive and interpret human poses has far-reaching implications for numerous applications, such as:

Human-computer interaction: Pose estimation can be used to create natural user interfaces for controlling devices, applications, or games using body gestures.

Activity recognition: By analyzing the body poses of people in a scene, you can determine the activities they are engaged in, such as walking, running, or sitting.

Animation and gaming: Pose estimation can be used to drive the animation of virtual characters, enabling realistic motion and interaction in video games or computer-generated films.

Sports analysis: Coaches and athletes can use pose estimation to analyze and improve their techniques, postures, and movements during training or competitions.

Healthcare and rehabilitation: Pose estimation can be employed in monitoring patients during physiotherapy, tracking their progress, and providing feedback on their exercises.

Fitness and wellness: Applications can leverage pose estimation to offer real-time guidance and feedback on workout routines, yoga postures, or dance moves.

Surveillance and safety: Pose estimation can aid in detecting unusual or dangerous activities in public places, such as detecting falls, fights, or accidents.

Robotics: Robots can use HPE to understand human activities and interact with people more effectively, enabling better collaboration between humans and robots.

Fashion and retail: Pose estimation can be used to create virtual fitting rooms, allowing customers to see how clothes fit on a digital representation of themselves.

Augmented reality: By estimating human poses, augmented reality applications can overlay virtual content on real-world scenes in a more context-aware and interactive manner.

Over the past decade, HPE has witnessed significant advancements, primarily driven by the proliferation of deep learning techniques and the availability of large-scale annotated datasets. Convolutional Neural Networks (CNNs) [1] have emerged as a powerful approach to model the complex spatial relationships between body joints and achieve state-of-the-art performance on various HPE benchmarks.

There are several popular deep learning architectures [2] for HPE, such as Stacked Hourglass Networks [3], and Simple Baselines [4]. These architectures utilize CNNs to estimate human poses. Recent advancements also involve using Transformer-based models, such as Vision Transformers [5] (ViT), for pose estimation tasks.

Analysis of recent research and publications. Vision Transformers have emerged as a powerful architecture for computer vision tasks, originally proposed by Dosovitskiy et al. for image classification. They have shown remarkable performance, often surpassing traditional CNNs on various tasks. The core idea behind Vision Transformers is to leverage the self-attention mechanism from the Transformer architecture, initially designed for natural language processing, and apply it to images by treating them as sequences of tokens.

Recently, researchers have started to explore the potential of ViT for pose estimation tasks. To adapt the ViT architecture for human pose estimation, several modifications and approaches can be considered:

Tokenization: divide the input image into non-overlapping patches and linearly embed them as tokens. Include an additional learnable token, known as the class token, to aggregate global information across the image.

Position embeddings: add learnable position embeddings to the patch tokens to capture the spatial information, which is essential for pose estimation tasks.

Multi-scale feature extraction: integrate multi-scale feature extraction into the ViT architecture to capture both local and global context. This can be achieved by incorporating multi-scale patch sizes, multi-head self-attention, or pyramid-like architectures.

Heatmap prediction: modify the output layer of the ViT architecture to predict heatmaps for each keypoint.

Heatmaps are 2D probability maps that represent the likelihood of a keypoint being present at each pixel location. The final keypoint locations can be obtained by finding the maximum value in each heatmap.

Multi-stage refinement: incorporate a multi-stage architecture to refine the pose estimation iteratively. Each stage can consist of a Vision Transformer module followed by an output prediction layer. Intermediate supervision can be applied to encourage the model to learn better intermediate representations.

Multi-person pose estimation: for multi-person pose estimation tasks, incorporate an additional output layer to predict the presence of individuals in the image, along with their corresponding keypoints.

Several recent works have successfully applied ViT to HPE tasks, achieving competitive performance compared to traditional CNN-based methods. Some notable examples include Swin Transformer and VisTR which demonstrate the potential of Transformer-based architectures for pose estimation.

While Vision Transformers have shown promise in pose estimation tasks, there remain challenges to address, such as computational complexity and the need for large amounts of training data. Ongoing research in this area aims to overcome these challenges and further improve the performance of Vision Transformers for HPE and other computer vision task

The Stacked Hourglass Network, proposed by Newell et al. in their paper "Stacked Hourglass Networks for Human Pose Estimation," is a deep learning architecture specifically designed for HPE tasks. This model has been particularly influential in the field, as it introduced a novel approach to capturing and combining multi-scale contextual information in a deep learning framework.

The Stacked Hourglass Network consists of multiple stacked hourglass modules, each of which is responsible for predicting the heatmaps for each keypoint. The overall architecture is symmetric, with a series of downsampling layers followed by upsampling layers, resembling an hourglass shape. Here's an overview of the key components Stacked Hourglass Network:

Hourglass module: The hourglass module is designed to capture and process multi-scale contextual information within an image. It consists of a series of convolutional and pooling layers that progressively downsample the input, followed by a series of upsampling and convolutional layers that gradually restore the original resolution. The downsampling and upsampling stages are connected through skip connections, which allow the model to learn both local and global context.

Intermediate supervision: The Stacked Hourglass Network employs intermediate supervision between the hourglass modules. After each hourglass module, the model predicts a heatmap for each keypoint. The predicted heatmaps are compared to the ground truth heatmaps, and the loss is backpropagated to refine the model's predictions. This intermediate supervision encourages the model to learn better intermediate representations.

Stacking hourglass modules: multiple hourglass modules are stacked together to form the complete Stacked Hourglass Network. The output of each module is passed as input to the next module, allowing the model to refine its predictions iteratively. The final pose estimation is obtained by combining the predictions from all the stacked modules.

Heatmap prediction: The Stacked Hourglass Network predicts a heatmap for each keypoint, which represents the likelihood of the keypoint being present at each pixel location in the image. The final keypoint locations are obtained by finding the maximum value in each heatmap.

The Stacked Hourglass Network has been highly successful in HPE tasks, achieving state-of-the-art performance on several benchmarks when it was introduced. Its design and principles have influenced many subsequent pose estimation models. While newer architectures, such as those based on Vision Transformers, have emerged, the Stacked Hourglass Network remains a valuable reference point for understanding and developing HPE models.

The Simple Baselines model for HPE, proposed by Xiao et al. in their paper "Simple Baselines for Human Pose Estimation and Tracking," is a deep learning-based approach that focuses on simplicity and efficiency while achieving competitive performance compared to more complex models. The Simple Baselines model uses a ResNet backbone for feature extraction, followed by a few deconvolutional layers to predict heatmaps for each keypoint.

The model employs a ResNet backbone (e.g., ResNet-50, ResNet-101, or ResNet-152) to extract high-level features from the input image. ResNet architectures have proven to be effective for various computer vision tasks due to their ability to learn deep representations while mitigating the vanishing gradient problem through residual connections.

“The Simple Baselines for Human Pose Estimation and Tracking” model stands out for its simplicity and computational efficiency while maintaining competitive performance in HPE tasks.

Its straightforward design makes it easier to implement, understand, and extend compared to more complex models, making it an attractive choice for various pose estimation applications.

Materials and methods

HPE involves predicting (Fig. 1) the spatial locations of human body joints or keypoints from input data such as images or videos [6]. It typically focuses on 2D or 3D pose estimation:

2D Pose Estimation [7]: this estimates the 2D coordinates (x, y) of the body keypoints in an image or video frame. It is widely used due to its computational efficiency and applicability to various applications.

3D Pose Estimation: this estimates the 3D coordinates (x, y, z) of the body keypoints. It provides more information about the pose but requires additional information or assumptions, such as camera parameters, depth information, or temporal information across video frames.

Fig. 1. Examples of HPE model output

Here are the main approaches to HPE:

– Top-Down Approach: this involves two main steps: human detection and keypoint localization. First, a human detector (e.g., Faster R-CNN, YOLO, or SSD) is used to locate people in the input image. Then, a pose estimation model is applied to each detected bounding box to estimate the keypoints.

– Bottom-Up Approach: this approach first predicts the individual body keypoints for all the people in the input image. Then, a grouping algorithm (e.g., greedy inference, bipartite matching) is used to associate the detected keypoints with each person. This approach is computationally efficient for scenes with multiple people.

HPE is one of the problems that are solved using image classification. Image classification is a fundamental computer vision task that aims to assign a predefined class label to an input image based on the objects or features present within it. A typical algorithm for performing image classification consists of several steps, such as preprocessing, feature extraction, and classification. In recent years, deep learning techniques, particularly (CNNs), have become the go-to approach for image classification tasks due to their superior performance [8, 9].

Here is a high-level overview of the algorithm for performing image classification using a CNN:

– Data preprocessing: prepare the input data by performing operations such as resizing, normalization, and data augmentation. These steps help ensure that the input images are in a consistent format and improve the robustness of the model to variations in the data.

– Feature extraction: a CNN consists of several convolutional, pooling, and activation layers that learn to extract meaningful features from the input image. These layers capture both low-level features, such as edges and textures, and high-level features, such as object parts and semantic information. The output of the feature extraction stage is a high-dimensional feature map that represents the input image in a more compact and informative manner.

Fully connected layers: after feature extraction, one or more fully connected layers, also known as dense layers, are used to process the high-dimensional feature maps. These layers help the model learn non-linear combinations of the extracted features, which can improve the discriminative power of the model.

Output layer: the final layer of the CNN is a fully connected layer with as many neurons as there are class labels. This layer is typically followed by a softmax activation function, which converts the output of the neurons into class probabilities.

Loss function and optimization: the CNN is trained using a suitable loss function, such as cross-entropy loss, which measures the difference between the predicted class probabilities and the ground truth class labels. The model parameters are optimized using gradient-based optimization techniques, such as stochastic gradient descent (SGD) or adaptive optimizers like Adam, to minimize the loss function.

Evaluation: once the model is trained, it can be evaluated on a test set to measure its performance using metrics such as accuracy, precision, recall, or F1-score.

Inference: for a new input image, the trained CNN processes the image through the same series of layers, and the softmax activation function outputs a probability distribution over the class labels. The class label with the highest probability is assigned to the input image as the final classification result.

For this paper Simple Baselines model was chosen because of its simplicity and resource efficiency while still achieving state-of-the-art performance.

This model takes a ResNet as basis and several changes that will be described below.

ResNet is a deep learning architecture proposed by He et al. in their paper "Deep Residual Learning for Image Recognition." ResNet introduces residual connections, or skip connections, to mitigate the vanishing gradient problem that arises in deep neural networks. This allows the model to learn deep representations more effectively and achieve state-of-the-art performance on various computer vision tasks.

Basic building block: The basic building block of a ResNet model consists of two or three convolutional layers followed by batch normalization [10], [11] and ReLU (Rectified Linear Unit) activation functions. Let's denote the input to a basic block as x, and the output of the convolutional layers, batch normalization, and ReLU as F(x). In a simple case, where the input and output have the same dimensions, the residual connection is applied as follows:

$$y = F(x) + x$$

Here, y represents the output of the basic block, and the addition operation is performed element-wise.

Bottleneck block: For deeper ResNet models, a bottleneck block is often used to reduce the number of parameters and computational complexity.

Projection shortcut: When the input and output dimensions of a basic or bottleneck block are different, a projection shortcut is used to match the dimensions.

Stacking layers: The ResNet architecture is constructed by stacking basic or bottleneck blocks, forming a deep neural network. It begins with an initial convolutional layer and pooling layer, followed by a series of blocks grouped into stages, and finally, an average pooling layer and a fully connected layer for classification.

Loss function: ResNet is trained using the cross-entropy loss function, which is commonly used for multi-class classification problems.

In summary, the ResNet model leverages residual connections to enable the training of deep neural networks, allowing it to learn complex hierarchical features effectively. The simplicity and effectiveness of the ResNet architecture have made it widely adopted in various computer vision tasks, including image classification, object detection, and semantic segmentation. Simple Baselines (Fig. 2) propose a few changes to the original model:

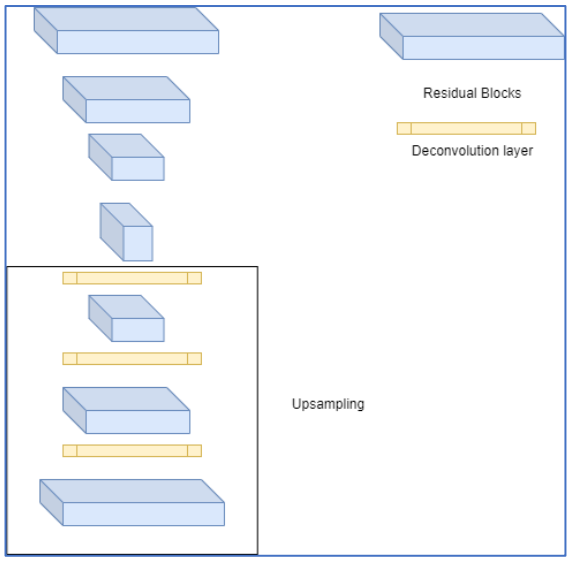

Fig. 2 Simple Baselines architecture

Deconvolutional layers: After feature extraction, the model utilizes a series of deconvolutional layers to upsample the feature maps to a higher resolution. Deconvolutional layers, also known as transposed convolution or fractionally-strided convolution, are used to increase the spatial dimensions of the feature maps while preserving the spatial information. The purpose of these layers is to generate heatmaps for each keypoint at a suitable resolution, enabling accurate keypoint localization.

Batch normalization and ReLU: Between the deconvolutional layers, batch normalization and ReLU activation functions are applied to normalize the feature distribution and introduce nonlinearity, respectively.

Heatmap prediction: The final layer of the model is a 1x1 convolutional layer that predicts a heatmap for each keypoint. Heatmaps represent the likelihood of a keypoint being present at each pixel location in the image. The final keypoint locations are obtained by finding the maximum value in each heatmap.

Loss function: the model is trained using the Mean Squared Error (MSE) loss between the predicted heatmaps and the ground truth heatmaps. This loss encourages the model to learn accurate keypoint localization.

So, to build solution using this model following steps were performed:

For the training a dataset was chosen and collected – COCO (Common Objects in Context) [12] 2017 training dataset. It's set of data with already classified objects in it. Next step is building a neural network mode. As the backbone the ResNet50, pretrained on ImageNet is used. After that deconvolution layers are added. Next stage is training on previously prepared COCO 2017 dataset, using following params: Mean Square error as loss function, Adam optimization, accuracy metric. After this training steps quality of model is evaluated using default metrics for COCO Keypoint Detection.

In the context of HPE, Average Precision (AP) and Average Recall (AR) are two evaluation metrics commonly used to quantify the performance of a model. They are based on precision and recall metrics that are widely used in various computer vision tasks (Table 1).

Table 1 – **Metric results using ResNet 50 and 256x192 image size**

Metric Name	Value	Metric Name	Value
AP	70.4	AR	76.2
AP .5	88.6	AR .5	93
AP .75	77.8	AR .75	83
AP (M)	67	AR (M)	71.9
AP (L)	76.9	AR (L)	82.4

Precision: Precision measures the proportion of true positive predictions (correct keypoint detections) out of all positive predictions made by the model. In HPE, it assesses how accurate the model is in detecting keypoints.

Recall: Recall measures the proportion of true positive predictions (correct keypoint detections) out of all actual positive instances (ground truth keypoints) in the dataset. In HPE, it assesses how well the model detects all keypoints present in the image.

AP and AR for HPE are calculated as follows:

Average Precision (AP): to compute the AP for HPE, we first calculate the precision at different levels of

keypoint localization error (thresholds). For example, the model's predictions are considered correct if the distance between the predicted and ground truth keypoints is below a certain threshold (e.g., a percentage of the head segment length). The AP is then calculated as the average of precision values across different error thresholds. The AP provides a single value that summarizes the model's performance in terms of both accuracy and robustness to keypoint localization errors.

Average Recall (AR): similarly, the AR for HPE is calculated by computing the recall at different levels of keypoint localization error (thresholds). The AR is then obtained as the average of recall values across different error thresholds. The AR offers a single value that summarizes the model's performance in terms of both the ability to detect keypoints and its robustness to localization errors.

Both AP and AR can be computed for individual keypoints or averaged across all keypoints to obtain a single value that represents the overall performance of the HPE model. High AP and AR values indicate that the model is both accurate and robust in detecting and localizing keypoints in the input images.

Conclusions

Problem of Human pose estimation was reviewed in this paper. Methods and models for human joints classification were explored and of them was implemented and evaluated. It has showed results that meet current expectations of accuracy for neural networks for HPE. It's important to note that this is only one of the models that can be used for HPE and can't be considered as best and only solution. As main object of HPE is human it will only increase its actuality in future. Results can be further used in activity recognition, animation, gaming sports analysis, healthcare and rehabilitation, fitness and wellness surveillance and safety, robotics, fashion and retail, augmented reality modern approaches what results in further growth of demand of such solutions in future.

REFERENCES

1. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017), "ImageNet classification with deep convolutional neural networks", Communications of the ACM, Vol. 60, No. 6
2. Alexander Toshev, Christian Szegedy (2014), "DeepPose: Human Pose Estimation via Deep Neural Networks", DOI: https://doi.org/10.1109/CVPR.2014.214
3. Alejandro Newell, Kaiyu Yang, Jia Deng (2016), "Stacked Hourglass Networks for Human Pose Estimation", DOI: https://doi.org/10.48550/arXiv.1603.06937
4. Bin Xiao, Haiping Wu, Yichen Wei (2018), "Simple Baselines for Human Pose Estimation and Tracking", DOI: https://doi.org/10.48550/arXiv.1804.06208
5. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby (2020), "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", DOI: https://doi.org/10.48550/arXiv.2010.11929
6. Girdhar, R., Gkioxari, G., Torresani, L., Paluri, M., Tran, D., "Detect-and-track: Efficient pose estimation in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition"
7. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B. 20(14) "2d human pose estimation: New benchmark and state of the art analysis. In: IEEE Conference on Computer Vision and Pattern Recognition "
8. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015), "Deep Residual Learning for Image Recognition", DOI: https://doi.org/10.48550/arXiv.1512.03385
9. Huang, Gao; Liu, Zhuang; Van Der Maaten, Laurens; Weinberger, Kilian Q. (2017). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), doi:10.1109/CVPR.2017.243
10. Ioffe, S., Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning."
11. Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik, (2018) "End-to-end Recovery of Human Shape and Pose", DOI: https://doi.org/10.48550/arXiv.1712.06584
12. COCO2017 dataset, available at https://cocodataset.org/?ref=blog.roboflow.com#download

Received (Надійшла) 12.03.2023
Accepted for publication (Прийнята до друку) 16.05.2023

Система оцінка пози людини з використанням алгоритмів глибокого навчання

Д. Вишнівський, О. Ляшенко, Н. Єрьоміна

Анотація. Метою даної роботи є програмна реалізація нейронної мережі, яка може вирішити задачу оцінки пози людини. Завдяки швидкому вдосконаленню моделей нейронної мережі та обчислювальних ресурсів за останні 10 років стало можливим автоматизувати багато процесів, проводити дослідження та покращувати якість життя. Одним із напрямків є комп'ютерний зір: він дозволяє розпізнавати об'єкти, відстежувати рухи, сегментувати зображення, розпізнавати обличчя тощо. Оцінка пози людини є частиною напряму досліджень комп'ютерного зору. Це дозволяє захопити людську позу з відео або зображення та має багато застосувань у медицині, спорті, доповненій реальності, відеоіграх тощо. Таким чином, ціль цієї роботи полягає в тому, щоб знайти та оптимізувати алгоритм, який є відносно точним, для ідентифікації та класифікація суглобів в тілі людини. Для досягнення поставленої мети були вирішені наступні завдання: розглянуто та проаналізовано сучасні методи та технології, які зазвичай використовуються для вирішення задачі оцінки пози людини, використано штучні нейронні мережі як математичний апарат для моделі, програмна реалізація для оцінки пози людини. було розроблено та протестовано, результати моделі проаналізовано та оцінено, результати та висновки сформульовано.

Ключові слова: оцінка пози людини; класифікація об'єктів; виявлення об'єктів; згорткові нейронні мережі.