Optimising Deep Learning Models for Pose Estimation

ABHISHEK AHUJA

ACM Reference Format:
Abhishek Ahuja. 2024. Optimising Deep Learning Models for Pose Estimation. 1, 1 (August 2024), 15 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Human pose estimation, a fundamental problem in computer vision, has seen remarkable advancements in recent years, primarily driven by deep learning techniques. As these models have grown in complexity and accuracy, they have also become increasingly computationally expensive, presenting challenges for real-world applications, especially on resource-constrained devices. This paper addresses the crucial need for optimizing deep learning models in pose estimation, focusing on the balance between model performance and computational efficiency.

The field of pose estimation has evolved from early approaches using pictorial structure models to sophisticated deep learning architectures. Convolutional Neural Networks (CNNs) have been at the forefront of this revolution, with landmark models such as DeepPose [12], Stacked Hourglass Networks [8], and High-Resolution Networks (HRNet) [11] pushing the boundaries of accuracy. However, these advancements often come at the cost of increased model size and computational complexity. More recently, Transformer-based architectures have emerged as a promising direction in pose estimation. Models like TransPose [16] have demonstrated competitive performance while using fewer parameters compared to traditional CNN-based approaches. TransPose combines the strengths of CNNs for feature extraction with the global context modeling capabilities of Transformers, offering a new paradigm in efficient pose estimation. Despite these innovations, there remains a significant challenge in deploying state-of-the-art pose estimation models in real-world scenarios, particularly on edge devices or in applications requiring real-time performance. The self-attention mechanism in Transformers, while powerful, introduces its own computational bottlenecks, especially for larger input sizes. This research focuses on optimizing the TransPose-R model, a variant of the TransPose architecture, with the primary goal of reducing model size while maintaining acceptable performance. The Network Slimming technique is used for pruning convolutional layers of the model.

By applying this optimization technique, we aim to create a more efficient pose estimation model that can be deployed in resource-constrained environments without significantly compromising accuracy. Our work contributes to the ongoing effort in the field to develop models that are not only accurate but also computationally efficient and suitable for real-time applications. The rest of this paper is organized as follows: Section 2 provides an overview of related work in pose estimation and model optimization techniques. Section 3 details our methodology, including the modifications made to the TransPose-R architecture and the application of pruning techniques. Section 4 presents our experimental results, followed by an analysis in Section 5. Finally, we conclude our findings and discuss future directions in Section 6.

Author's Contact Information: Abhishek Ahuja, a.ashwaniahuja@student.utwente.nl.

ACM XXXX-XXXX/2024/8-ART
https://doi.org/XXXXXXX.XXXXXXX

The code associated with this work can be found at https://github.com/abhishek1ahuja/TransPose.

2 Related Work

2.1 Human Pose Estimation

Human pose estimation, a fundamental problem in computer vision, involves detecting and localizing human body parts or joints in images or videos. This field has seen significant advancements in recent years, primarily due to the advent of deep learning techniques.

Early approaches to pose estimation relied on pictorial structure models and deformable part models. However, the introduction of deep learning methods, particularly Convolutional Neural Networks (CNNs), has dramatically improved the accuracy and robustness of pose estimation systems.

One of the pioneering works in deep learning-based pose estimation was DeepPose[12]. This method directly regressed joint coordinates using a cascade of CNNs. However, subsequent research showed that heatmap-based approaches, which predict a probability distribution for each joint, generally outperform direct regression methods.

A significant milestone in the field was the introduction of the Stacked Hourglass Network[8]. This architecture, characterized by repeated bottom-up and top-down processing with skip connections, allowed for better integration of features across scales. The Stacked Hourglass Network and its variants have been widely adopted and serve as the backbone for many state-of-the-art pose estimation models.

Another influential approach is the Convolutional Pose Machines (CPM) [14]. CPM uses a sequential prediction framework to iteratively refine pose estimates, demonstrating the importance of large receptive fields in capturing long-range dependencies between body parts.

More recent advancements include the High-Resolution Network (HRNet) [11], which maintains high-resolution representations throughout the network, leading to more precise keypoint localization. HRNet has achieved state-of-the-art performance on various pose estimation benchmarks.

Lightweight architectures like MobileNetV2 [10] and ShuffleNet [17] have been adapted for pose estimation tasks. For instance, [15] proposed Simple Baselines for Human Pose Estimation and Tracking, which achieved competitive results with a relatively simple and efficient architecture based on ResNet [2].

Recent work has also explored the use of Transformers in pose estimation. Tokenpose [6] and PRTR [5] have shown promising results by leveraging the global context modeling capabilities of Transformer architectures [13].

A significant recent development in the field is the introduction of Transformer-based architectures for pose estimation. The TransPose model [16], represents a paradigm shift in this direction. TransPose combines the strengths of CNNs for feature extraction with the global context modeling capabilities of Transformers. The TransPose architecture consists of three main components:

1. A CNN backbone for low-level feature extraction
2. A Transformer encoder to capture long-range spatial interactions
3. A prediction head to generate keypoint heatmaps

TransPose has demonstrated competitive performance compared to state-of-the-art CNN-based models while using fewer parameters and achieving faster inference speeds. The model's ability to explicitly capture global dependencies through self-attention mechanisms has proven particularly effective for pose estimation tasks.

However, despite its efficiency gains compared to some larger CNN models, TransPose still presents opportunities for further optimization. The self-attention mechanism in Transformers has a quadratic computational complexity with respect to the input size, which can be a bottleneck for real-time applications or deployment on edge devices.

The field continues to evolve, with current research focusing on multi-person pose estimation, 3D pose estimation, and pose estimation in challenging scenarios such as occlusions and unusual poses. Furthermore, there is an increasing emphasis on developing models that are not only accurate but also computationally efficient and suitable for real-time applications.

Despite these advancements, there remains a crucial trade-off between model performance and computational efficiency. As pose estimation models become more accurate, they often grow in size and complexity, making them challenging to deploy in real-world applications with limited computational resources. This highlights the need for optimization techniques that can maintain high accuracy while reducing model size and computational requirements.

Recent research has focused on various optimization techniques to address this challenge. These include network pruning, knowledge distillation, and efficient attention mechanisms. For instance, DynamicViT [9] introduced a method to dynamically prune redundant tokens in Vision Transformers, which could potentially be adapted for pose estimation models like TransPose.

In this context, optimizing Transformer-based models like TransPose represents a promising direction for future research, potentially leading to more efficient and deployable pose estimation systems while maintaining high accuracy.

Variants TransPose-R and TransPose-H have performed at par with SimpleBaseline and HRNet, while having 85% and 72% reduction in model size compared to these models. This paper aims to further the efficiency of TransPose-R.

2.2 Network Pruning

As deep learning models grow in size and complexity, there's an increasing need for methods to reduce their computational requirements without significantly compromising performance. Pruning techniques have emerged as effective approaches to achieve this goal. We focus on the Network Slimming technique for convolutional layers.

2.2.1 Network Slimming. Pruning techniques for convolutional neural networks (CNNs) have been extensively studied due to the widespread use of CNNs in various computer vision tasks. Early approaches to CNN pruning focused on removing individual weights or neurons based on certain criteria.

[1] introduced a method of iterative pruning and fine-tuning, where weights below a certain threshold are pruned, followed by retraining to recover accuracy. This approach, while effective, often results in unstructured sparsity, which is challenging to accelerate on common hardware.

To address this, structured pruning methods were developed. [4] proposed pruning entire filters in CNNs, demonstrating that some filters are redundant and can be removed with minimal impact on performance. This approach leads to direct reductions in computational costs and model size.

Channel pruning is another structured pruning technique that has gained popularity. [3] introduced a method to prune channels by minimizing the reconstruction error of feature maps. This approach maintains the original network structure while reducing its width.

Network Slimming [7] represents a significant advancement in structured pruning techniques. This method introduces scaling factors in batch normalization layers and imposes L1 regularization on these factors during training. The scaling factors then serve as indicators of the importance of corresponding channels. Channels with small scaling factors are pruned, resulting in a slimmer network.

Network Slimming offers several advantages:

1. It automatically identifies and removes redundant channels during training.
2. It results in a compact model with reduced computational cost.

3. The pruned model can be easily fine-tuned to recover accuracy.

This technique has proven effective for various CNN architectures and has been widely adopted in the field of model compression.

3 Methodology

This research focuses on optimizing the TransPose-R model for human pose estimation, with the primary goal of reducing model size while maintaining acceptable performance. The optimization process involves applying the Network Slimming technique for convolutional layers in the backbone of the model.

3.1 Research Questions

With the motivations elucidated above we establish the following research questions:

What are the effects of applying Network Slimming to transformer-based pose estimation models, specifically TransPose-R, on model compression and accuracy?

3.2 Model Architecture

The base model for this study is TransPose-R, a variant of the TransPose architecture that combines convolutional neural networks (CNNs) and transformer layers for efficient pose estimation. The model consists of a ResNet CNN backbone for feature extraction, followed by transformer layers for capturing global dependencies, and a prediction head for generating keypoint heatmaps.

The backbone of the TransPose-R model uses selected initial layers from ResNet-50 [2].

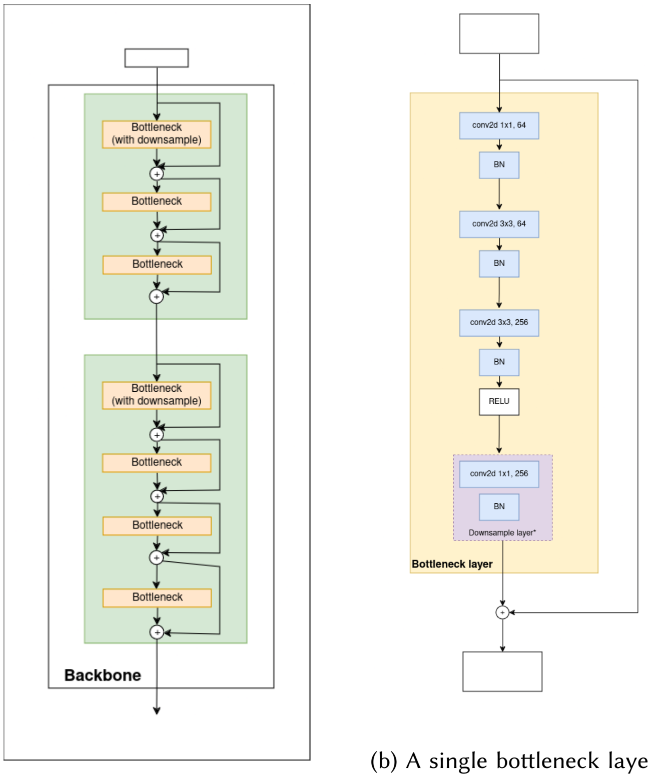
3.3 Network Slimming for Convolutional Layers

3.3.1 Channel Selection Layers. To facilitate the pruning process, a key modification was made to the original TransPose-R architecture - adding channel selection layers. A channel selection layer was added after the first convolution-batch normalization (Conv-BN) pair in each bottleneck block of the CNN backbone. As in [7], channel selection layers are used to "soft-prune" the first convolutional layer of a bottleneck layer, in order to preserve the structural integrity of the network, since residual connections will not be valid if the first layer of a subsequent block is not the same shape as the inputs to the block.

After the initial experiments, it was realized that Channel selection layers were not applied correctly and thus in further experiments Channel selection layers were not used anymore. This has enabled overcoming the "soft-pruning" and led to greater reduction in model parameters from the pruning action.

3.3.2 Pruning conditions. The last layer of each bottleneck block is exempt from pruning to maintain structural consistency, as the number of output channels in this layer must match the number of input channels in the subsequent layer. Only the BN values from the bottleneck layers are considered in this scoring process, excluding those from downsample layers and other parts of the model.

Pruning Threshold: A rule was implemented to dynamically adjust the pruning criteria. When the number of channels in a layer becomes too low after pruning, that layer is excluded from further pruning rounds to prevent losing all channels in any layer. This can be seen from Table 1, where some layers reach a low channel number in early rounds, and then then the channel numbers remain fixed. In all experiments, a threshold of 10 channels was set. When number of channels are below this, then a layer is no longer considered for pruning.



(a) Backbone Architecture

(b) A single bottleneck layer, *inner downsample layer is optional

Fig. 1. Architecture of Transpose backbone

Table 1. Number of channels in convolution layers over multiple rounds of pruning. This corresponds to Experiment 1

Round	layer 1	layer 2	layer 3	4	5	6	7	8	9	10	11	12	13	14	ignore layers
Original	64	64	64	64	64	64	128	128	128	128	128	128	128	128	-
1	33	31	39	52	50	62	117	119	38	95	93	113	90	123	-
2	32	22	31	35	36	59	105	98	17	55	61	89	57	94	-
3	13	16	12	29	19	59	81	88	1	51	22	87	26	89	-
4	12	13	5	21	9	56	49	76	1	39	8	66	8	82	9
5	1	12	5	17	9	50	30	54	1	30	8	44	8	72	3,5,9,11,13
6	1	9	5	11	9	44	17	38	1	23	8	31	8	58	1,3,5,9,11,13
7	1	9	5	7	9	35	11	32	1	20	8	17	8	44	1,2,3,4,5,9,11,13

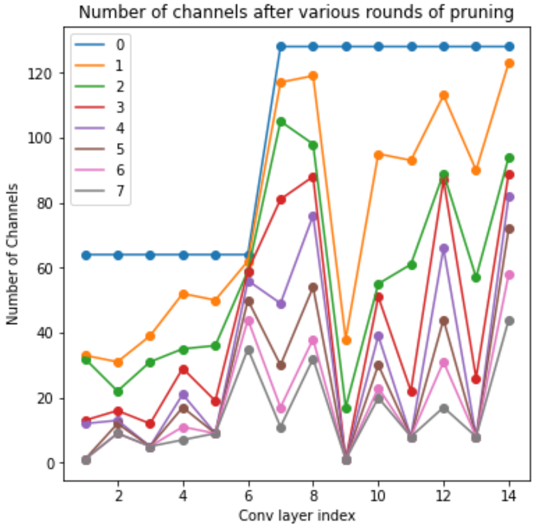


Fig. 2. Number of channels over multiple rounds of pruning. Each line in this chart shows the number of channels in the convolutional layers eligible for pruning at a certain round of the iterative pruning process. The blue line shows the number of channels in the original model backbone. Each line below it represents the number of channels at a further round of pruning. This figure corresponds to Experiment 1.

3.3.3 Importance Scoring. For the layers that are eligible for pruning, the scaling factors (γ) of the batch normalization layers are used as indicators of channel importance. Based on the importance scores, channels are selected for pruning.

Before pruning, the network is trained with a modified loss function. An L1-regularization term is added to the weights in the network corresponding to the scaling factors, which primes the scaling factors for pruning. We shall refer to this as the 'scaling' step of the pruning process.

A pruning ratio determines the proportion of channels to be pruned in each round. Pruning ratios of 0.1 and 0.25 were used in different experiments. A pruning ratio of 0.1 would mean that the 10% lowest magnitude scaling factors would be marked for removal from the network along with their corresponding convolution layer channels.

3.3.4 Pruning. The model is resized based on the selected channels from the previous step, and the filtered weights are retained in the pruned model. A new model is created with the updated channel parameters for the pruned layers, and the corresponding weights are copied to the new model. For all layers of the model which are not affected by pruning, their weights are retained as is.

3.3.5 Fine-tuning. After each pruning round, the model is fine-tuned to recover accuracy lost due to pruning. Fine-tuning involves training the pruned model with the prepared dataset. The number of epochs is taken as a fraction of steps used for training the original TransPose model. In some experiments, the pruned model has been trained with the 10% subset dataset for finetuning. In later experiments, the full dataset was used to determine any missed potential from using a smaller subset of data for finetuning.

This process is performed iteratively, with multiple rounds of pruning and fine-tuning to gradually compress the model while maintaining performance.

3.4 Pruning strategies

Multiple batches of experiments were carried out with two different pruning ratios (0.1 and 0.25) for each round of pruning. In experiments 3,4,5 and 6, before pruning, the model was trained with an added regularisation term to the loss function, for adjusting the scaling factors for pruning. In experiment 1 and 2, this step was not done.

These strategies allow for a comparison between more gradual and more aggressive pruning approaches, in combination with other varying configurations, providing insights into the trade-offs between compression rate and model performance.

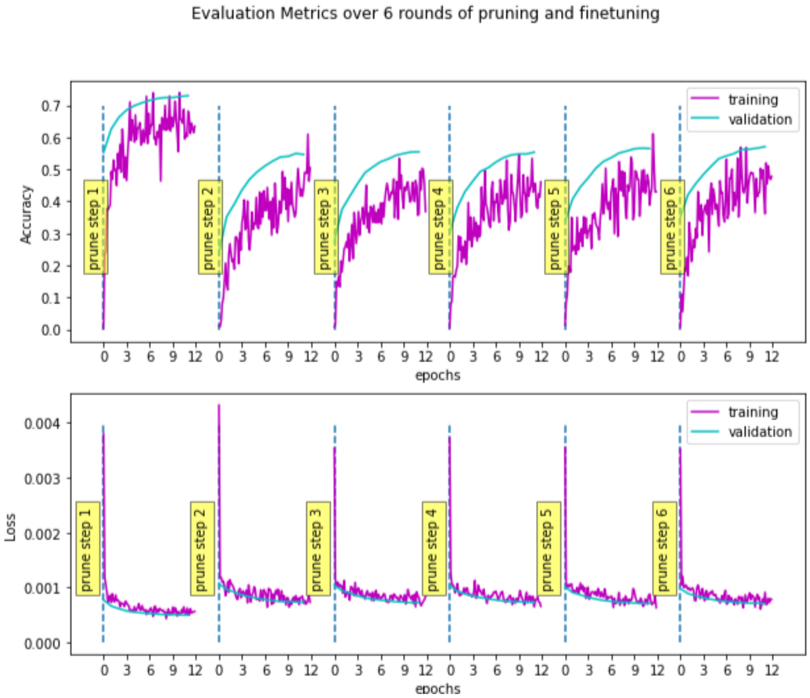


Fig. 3. Evaluation metrics over 6 rounds of pruning and finetuning. This chart shows the learning metrics during pruning and finetuning. The top chart shows training and validation accuracy, the bottom one shows training and validation loss. For representation purposes we show only the first 6 of 16 pruning rounds of experiment 2. Each finetuning step is for 12 epochs, and is punctuated by a pruning step.

3.5 Dataset and Training

The model optimization process uses a subset of the COCO dataset for training and evaluation:

Training Data: The model optimisation process uses either of two datasets for training. A subset comprising 10% of the COCO training images, along with the corresponding pose estimation ground truth annotations provided by the TransPose project. In some experiments, the full dataset is used for training.

Validation Data: The COCO validation dataset is used to evaluate the model's performance after each round of pruning and fine-tuning.

Table 2. Configurations for different experiments

Exp #	Scaling dataset	Scaling steps	Pruning Ratio	Rounds	Finetuning Dataset	Finetuning steps
1	-	-	0.25	7	10% COCO subset	20 epochs
2	-	-	0.1	16	10% COCO subset	12 epochs
3	10% COCO subset	3 epochs	0.25	6	10% COCO subset	12 epochs
4	10% COCO subset	3 epochs	0.1	6	10% COCO subset	12 epochs
5	10% COCO subset	10 epochs	0.25	9	COCO full dataset	8 epochs
6	10% COCO subset	10 epochs	0.1	11	COCO full dataset	8 epochs

The earlier experiments were carried out with a subset of the training set, because of time constraints. The later experiments were finetuned using the full dataset to compare the difference in performance gain and to add greater validity to the study.

3.6 Evaluation Metrics

To assess the effectiveness of the pruning techniques and the overall model optimization, the following metrics are used:
Validation Accuracy: This metric measures the model's performance on the pose estimation task using the full COCO validation dataset.
Compression Ratio: The compression ratio for pruned models is calculated by the following formula.

$$\text{Compression Ratio} = 1 - \frac{\text{Parameter Count of Pruned Model}}{\text{Parameter Count of Original Model}} \tag{1}$$

This metric quantifies the extent of model size reduction.

These metrics allow for a multi-faceted evaluation of the trade-offs between model compression and performance.

This methodology provides a comprehensive approach to optimizing the TransPose-R model, focusing on reducing model size while maintaining acceptable performance for the pose estimation task.

4 Experiments

This section presents a series of experiments designed to evaluate the effectiveness of Network Slimming on the TransPose-R model for human pose estimation. We conducted six distinct experiments, each exploring different aspects of the pruning process, including pruning ratios, the impact of the scaling step, and the effect of dataset size on fine-tuning. Our goal was to assess the trade-off between model compression and performance accuracy, providing insights into the optimal strategies for reducing model size while maintaining acceptable accuracy. The following subsections detail our experimental setup, the metrics used for evaluation, and a summary of our results.

4.1 Experimental Setup

We used the TransPose-R-A4 model as our baseline, with an input size of 256×192 pixels. The model was trained and evaluated on the COCO dataset, using either the full dataset or a subset of 10% of the training images for faster experimentation cycles, and the full validation dataset for performance evaluation.

Six pruning strategies were explored as described in 3.4 and Table2. Experiments 1 and 2 were carried out without the scaling step. Experiment 1 involved 7 rounds of iterative pruning and finetuning, with a pruning ratio of 0.25. Whereas Experiment 2 involved 16 rounds of iterative pruning and finetuning, with a pruning ratio of 0.1. The next sets of experiments also have pruning ratios of 0.1 and 0.25, but with added configuration changes.

In Experiments 3 and 4, the scaling step is included before each round of pruning. Thus one round of pruning involves first training for the scaling factors, then pruning a specified portion of channels, and finally finetuning the model. Additionally, channel selection layers were not used in the model, in these experiments. This also increased the per-round reduction in model size for a given pruning ratio. The training steps in the experiments thus far all involve using the 10% subset of the COCO dataset.

In Experiments 5 and 6, the configurations of Experiment 3 and 4 are repeated, but in this case, the full COCO dataset is used for finetuning the model, while for scaling steps, the earlier 10% subset is used. This choice was made as it was observed that the model shows a greater recovery of accuracy when finetuned with the full dataset than with a subset.

4.2 Performance Metrics

Table 3 summarizes the results of our experiments:

With the baseline TransPose-R-A4 model having a validation acuracy of 0.86 with 5.99M parameters, the modified models show a reduction in size but accompanied with a drop in performance in all cases. In some experiments the drop in accuracy is as little as 0.03, while in others the loss in performance is much greater.

It can be realised that the scaling step is crucial to the pruning process, as we see the loss in performance in experiments 1 and 2 makes the model much less useful, with validation accuracy going as low as 0.49 and 0.59, and AP values as low as 0.31 and 0.20. In early experiments, this crucial step was erroneously skipped, and thus the experiments may be considered invalid. However the results are reproduced here to emphasise the effect of the lack of the scaling step before pruning.

It can also be seen across all experiments that pruning with a smaller pruning ratio leads to a more stable model, with lower losses in performance, while it takes more steps to reach a smaller model size. And with a larger pruning ratio, each step made in reducing model size is larger, but with a corresponding larger dip in performance.

Using the full dataset for retraining has also shown significant merit. The loss in validation accuracy for experiments 3 and 4, which used the 10% subset of COCO for finetuning, have shown twice as much loss in performance as compared to Experiments 5 and 6.

The usage of channel selection layers in Experiments 1 and 2, can be seen in the level of model compression they have achieved. Channel selection layers were wrongly applied in Experiments 1 and 2. These are applicable when a batch normalisation layer corresponding to a convolution layer is present not in the same block of the model, but in a subsequent block. In such a case, channel selection layers are useful for logically pruning the last convolutional layer of blocks. Thus Experiment 3 onwards, Channel selection layers were not used, and this results in greater compression ratio in fewer rounds. Application of channel selection layers requires modification of the block structure of the model, and this can be considered in future research.

These experiments demonstrate that our Network Slimming approach can effectively reduce the size of the TransPose-R model while maintaining performance up to a certain level of compression. The conservative approach proved more successful in balancing compression and accuracy, while the aggressive approach achieved similar size reduction at the cost of more significant performance degradation.

The model derived using experiment 6 is the most promising model based on performance, as it has lowest drop in validation accuracy, it achieves 83% accuracy with 17% fewer parameters than the original model. The

Table 3. Results of 6 experiments with various configurations.

Exp #	scaling step	Full dataset	Pruning Ratio	# of Rounds	AP	AR	Valid. acc.	Params
Original	-	✓	-	-	0.75	0.78	0.86	5.99M
1	-	-	0.25	7	0.31	0.37	0.49	5.05M (↓15.5%)
2	-	-	0.1	16	0.20	0.26	0.59	5.09M (↓14.9%)
3	✓	-	0.25	6	0.51	0.55	0.72	4.82M (↓19.4%)
4	✓	-	0.1	6	0.61	0.65	0.78	5.22M (↓12.8%)
5	✓	✓	0.25	9	0.63	0.67	0.79	**4.75M (↓20.6%)**
6	✓	✓	0.1	11	**0.68**	**0.72**	**0.83**	4.97M (↓17.0%)

model from experiment 5 has a compression ratio of 20.6%, and it's accuracy has dropped to 0.79, 0.07 lower than the original model.

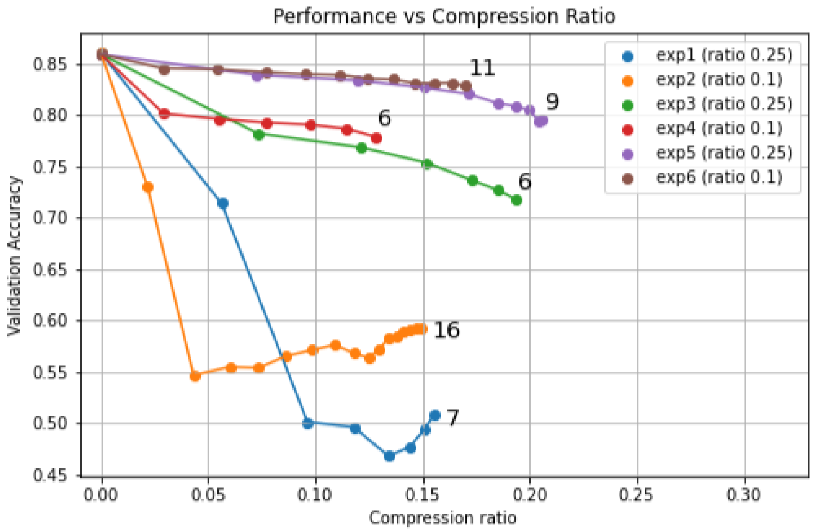


Fig. 4. Performance relative to Compression ratio for all 6 experiments. Each line corresponds to a single experiment, and each data point represents one round of pruning and finetuning. The numbers at the right end of each line show the number of rounds of iterative pruning in the corresponding experiment.

In summary, our experiments demonstrate that Network Slimming can effectively reduce the size of the TransPose-R model, achieving up to 20.6% parameter reduction. However, this compression comes with varying degrees of performance degradation, highlighting the delicate balance between model size and accuracy. The inclusion of the scaling step and the use of the full dataset for fine-tuning proved crucial in maintaining model performance. Conservative pruning strategies (with a ratio of 0.1) showed more stable performance across pruning rounds, while aggressive strategies (with a ratio of 0.25) achieved higher compression rates at the cost of more significant performance drops. The most promising result came from Experiment 6, which achieved 83% accuracy with 17% fewer parameters than the original model. These findings provide valuable insights for future optimization efforts and set the stage for further analysis in the subsequent sections.

5 Analysis

Our experiments with Network Slimming on the TransPose-R model have yielded several important insights into the optimization of deep learning models for pose estimation. This section provides a detailed analysis of our findings, focusing on the effectiveness of our approach, the trade-offs involved, and the implications for future research.

5.1 Effectiveness of Network Slimming

The application of Network Slimming to the TransPose-R model demonstrated its potential for significant model compression while maintaining performance up to a certain threshold:

1. Parameter Reduction: Across our experiments, we achieved parameter reductions ranging from 12.8% to 20.6%. The most successful compression was observed in Experiment 5, which reduced the model size from 5.99M to 4.75M parameters (20.6% reduction) while maintaining a validation accuracy of 0.79, compared to the original 0.86.

2. Performance-Compression Trade-off: As expected, increased compression generally led to decreased performance. However, the relationship was not linear. For instance, Experiment 6 achieved a 17% parameter reduction with only a 0.03 drop in validation accuracy, suggesting that a significant portion of the pruned parameters were indeed redundant.

3. Dataset Impact: Experiments 5 and 6, which used the full COCO dataset for fine-tuning, showed superior performance retention compared to experiments using only a subset. This underscores the importance of comprehensive fine-tuning data in recovering performance post-pruning.

5.2 Impact of Pruning Strategies

The comparison between aggressive (0.25 pruning ratio) and conservative (0.1 pruning ratio) approaches revealed:

1. Stability vs. Compression: Conservative pruning (Experiments 2, 4, and 6) generally led to more stable performance across pruning rounds, while aggressive pruning (Experiments 1, 3, and 5) achieved higher compression rates but with more significant performance drops.

2. Recovery Potential: Interestingly, models subjected to aggressive pruning showed a capacity for partial recovery in later rounds, particularly when fine-tuned on the full dataset (Experiment 5). This suggests that the network can adapt to significant structural changes given sufficient training data and time.

3. Optimal Pruning Schedule: The results indicate that a dynamic pruning schedule, possibly starting with more aggressive pruning and becoming more conservative in later rounds, might yield optimal results.

5.3 Implications for Model Design or Pruning Strategy Design

Our findings have some implications for the design of efficient pose estimation models:

1. Pruning Strategy Optimization: The significant difference in outcomes between conservative and aggressive pruning strategies highlights the need for careful tuning of pruning hyperparameters. Since we realise that small pruning ratios are more stable and larger ones lead to faster pruning - we must design a pruning strategy that shall make use of small as well as larger pruning ratios to achieve better results with fewer rounds of pruning and thus lesser re-training time.

2. It may also be considered to use full dataset for finetuning, not for each finetuning round but perhaps every 4 rounds, for instance. This can be experimented to check if this is sufficient for performance recovery, and if

so the finetuning process can be made more efficient, as the finetuning training can be shorter with a smaller dataset.

3. Architectural Considerations: The varying pruning rates across different layers suggest that some parts of the network are more critical than others. This insight could inform future architectural designs, potentially leading to more efficient base models.

4. Balance of Techniques: While Network Slimming proved effective for convolutional layers, the gains can be limited to the backbone of the model. Thus combining Network Slimming with other pruning techniques that apply to the transformer component of the model can possibly yield better results.

5.4 Limitations and Future Directions

Our analysis also reveals some limitations and areas for future research:

1. Generalization: These experiments were conducted on a subset of the COCO dataset. Further testing on full datasets and other pose estimation benchmarks would be necessary to confirm the generalizability of our findings.

2. Long-term Stability: While we observed some recovery in performance over multiple pruning rounds, the long-term stability of heavily pruned models remains an open question. Extended training and evaluation on diverse datasets could provide insights into this aspect.

3. Hardware Considerations: While we focused on reducing parameter count, future work should also consider the impact on inference speed and memory usage across different hardware platforms.

4. Combination with Transformer Pruning techniques: The next step in our research is to implement and analyze the effect of combining Network Slimming with one or more pruning techniques that are specialised for the transformer layers of the model.

In conclusion, our analysis demonstrates that Network Slimming is a promising approach for optimizing the TransPose-R model, offering a viable path to creating more efficient pose estimation models. However, it also highlights the complex interplay between model compression and performance, emphasizing the need for careful balancing and potentially hybrid approaches in future optimizations.

6 Conclusion

This study set out to optimize the TransPose-R model for human pose estimation, with the primary goal of reducing model size while maintaining acceptable performance. Our research focused on the application of Network Slimming to the convolutional layers of the model, aiming to address the crucial need for efficient, deployable pose estimation systems in resource-constrained environments.

Our key findings can be summarized as follows:

1. Effectiveness of Network Slimming: We successfully reduced the TransPose-R model size by up to 20.6% (from 5.99M to 4.75M parameters) while maintaining a validation accuracy of 0.79, compared to the original 0.86. This demonstrates the potential of Network Slimming for creating more efficient pose estimation models.

2. Trade-off Between Compression and Performance: We observed a clear but non-linear relationship between model compression and performance. Conservative pruning strategies (0.1 pruning ratio) showed more stable performance across pruning rounds, while aggressive strategies (0.25 pruning ratio) achieved higher compression rates at the cost of more significant performance drops.

3. Adaptive Pruning Dynamics: Our experiments revealed varying levels of redundancy across different layers of the network, with some layers more amenable to pruning than others. This suggests the potential for more nuanced, layer-specific optimization approaches in future work.

4. Dataset Impact: Using the full COCO dataset for fine-tuning showed superior performance retention compared to using only a subset, underscoring the importance of comprehensive training data in recovering performance post-pruning.

These findings contribute to the broader field of efficient deep learning for computer vision tasks, particularly in the domain of human pose estimation. They demonstrate that careful application of pruning techniques can lead to more efficient models without significant performance loss, potentially enabling the deployment of advanced pose estimation systems on resource-constrained devices.

However, our work also highlights several areas for future research:

1. Generalization and Robustness: Further testing on other datasets and diverse pose estimation tasks is necessary to confirm the generalizability of our findings and the robustness of the pruned models.

2. Investigation of pruning techniques specifically tailored for transformer layers to complement Network Slimming.

3. Structural Constraints: Addressing the limitation of not being able to prune layers immediately before a channel selection layer could lead to more comprehensive pruning strategies and potentially better compression-performance trade-offs.

4. Hardware-Specific Optimization: Future research should consider hardware-specific optimizations to fully leverage the reduced model size for improved inference speed across different platforms.

In conclusion, this study represents a significant step towards more efficient pose estimation models. By demonstrating the effectiveness of Network Slimming on the TransPose-R architecture, we have opened up new possibilities for deploying advanced pose estimation systems in resource-constrained environments. As the field continues to evolve, the insights gained from this research will contribute to the development of increasingly efficient and accurate pose estimation models, bringing us closer to ubiquitous, real-time human pose estimation across a wide range of devices and applications. However, the challenge of maintaining high performance while significantly reducing model size remains an open problem, inviting further innovation in model compression techniques for pose estimation tasks.

[1] I acknowledge the use of the AI language model Claude, developed by Anthropic, in the preparation of this thesis. Claude was used as a writing assistant to help with proofreading, formatting suggestions, and feedback about writing style. It was also used for interpreting and understanding code related to pruning and also helping with debugging and simplifying the code for the project. While Claude provided valuable assistance, all final decisions on content, analysis, and conclusions were made by me.

[1] Note on AI Assistance

Acknowledgments

I thank my primary supervisor, Dr Le Viet Duc, for his inspiration and guidance and through the course of my thesis. Dr Duc offered his continued support despite multiple long delays in execution of the thesis. I thank Dr. Faizan Ahmed for joining my committee as an external examiner and offering valuable feedback that improved the clarity and comprehensibility of my work.

I would like to express my deepest gratitude to my grandparents and my parents. Their unending support and encouragement enabled my studies in a foreign country. Their unwavering backing sustained me through the course of my thesis.

References

[1] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both Weights and Connections for Efficient Neural Network. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[3] Yihui He, Xiangyu Zhang, and Jian Sun. 2017. Channel Pruning for Accelerating Very Deep Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[4] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning Filters for Efficient ConvNets. *CoRR* abs/1608.08710 (2016). arXiv:1608.08710 http://arxiv.org/abs/1608.08710

[5] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. 2021. Pose Recognition With Cascade Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1944–1953.

[6] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. 2021. TokenPose: Learning Keypoint Tokens for Human Pose Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 11313–11322.

[7] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning Efficient Convolutional Networks Through Network Slimming. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[8] Alejandro Newell, Kaiyu Yang, and Jia Deng. 2016. Stacked Hourglass Networks for Human Pose Estimation. *CoRR* abs/1603.06937 (2016). arXiv:1603.06937 http://arxiv.org/abs/1603.06937

[9] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 13937–13949. https://proceedings.neurips.cc/paper_files/paper/2021/file/747d3443e319a22747fbb873e8b2f9f2-Paper.pdf

[10] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep High-Resolution Representation Learning for Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[12] Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[14] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple Baselines for Human Pose Estimation and Tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[16] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. 2021. TransPose: Keypoint Localization via Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 11802–11812.

[17] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A Sparsity Ratio

Figure 5 represents the sparsity from Figure 4 but here it considers the sparsity of the backbone alone. This indicates how much of the backbone weights are removed after the rounds of pruning in the 6 experiments. The size of the backbone is 1.77M parameters, which is about 30% of the whole model parameters. So as you can see, in experiment 5, from Figure 4 we understand that the model has 20% fewer parameters, from Figure 5 we see that the backbone number of parameters have reduced by 70%.

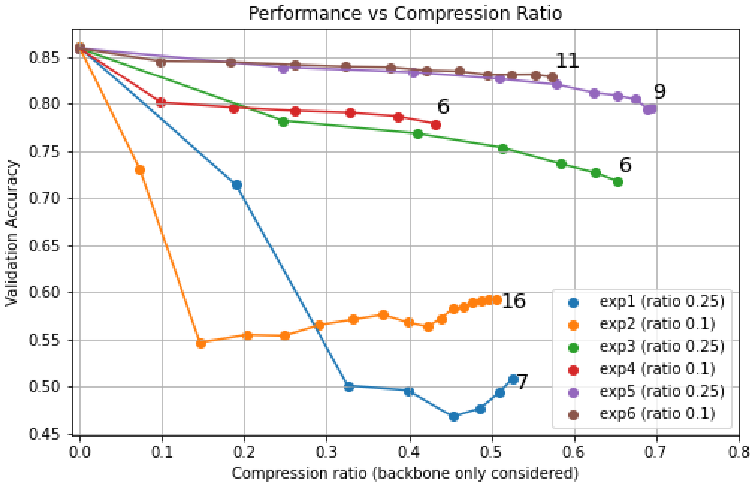


Fig. 5. Performance relative to sparsity for 2 strategies of pruning (sparsity ratio relative to backbone weights only)