Yoga-82: A New Dataset for Fine-grained Classification of Human Poses

Manisha Verma[1]*, Sudhakar Kumawat[2]*, Yuta Nakashima[1], Shanmuganathan Raman[2]
[1]Osaka University, Japan [2]Indian Institute of Technology Gandhinagar, India,
[1]{mverma,n-yuta}@ids.osaka-u.ac.jp [2]{sudhakar.kumawat,shanmuga}@iitgn.ac.in

Abstract

Human pose estimation is a well-known problem in computer vision to locate joint positions. Existing datasets for learning of poses are observed to be not challenging enough in terms of pose diversity, object occlusion and view points. This makes the pose annotation process relatively simple and restricts the application of the models that have been trained on them. To handle more variety in human poses, we propose the concept of fine-grained hierarchical pose classification, in which we formulate the pose estimation as a classification task, and propose a dataset, Yoga-82[§], for large-scale yoga pose recognition with 82 classes. Yoga-82 consists of complex poses where fine annotations may not be possible. To resolve this, we provide hierarchical labels for yoga poses based on the body configuration of the pose. The dataset contains a three-level hierarchy including body positions, variations in body positions, and the actual pose names. We present the classification accuracy of the state-of-the-art convolutional neural network architectures on Yoga-82. We also present several hierarchical variants of DenseNet in order to utilize the hierarchical labels.

1. Introduction

Human pose estimation has been an important problem in computer vision with its applications in visual surveillance [6], behaviour analysis [12], assisted living [8], and intelligent driver assistance systems [20]. With the emergence of deep neural networks, pose estimation has achieved drastic performance boost. To some extent, this success can be attributed to the availability of large-scale human pose datasets such as MPII [4], FLIC [23], SHPD [6], and LSP [17]. The quality of keypoint and skeleton annotations in these datasets play an important role in the success of the state-of-the-art pose estimation models. However, the manual annotation process is prone to human errors and can be severely affected by various factors such as resolution, occlusion, illumination, view point, and diversity of poses. For example, Fig. 1 showcases some human pose images for the yoga activity which inherently consist of some of the most diverse poses that a human body can perform. It can be noticed that some of these poses are too complex to be captured from a single point of view. This becomes more difficult with the changes in image resolution and occlusions. Due to these factors, producing fine pose annotations such as keypoints and skeleton for the target objects in these images may not be possible as it will lead to false and complex annotations.

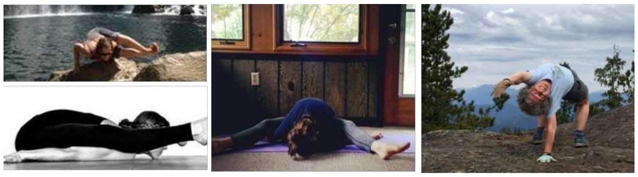


Figure 1. Example human poses from the yoga activity.

In order to solve this problem, we propose the concept of fine-grained hierarchical pose classification. Instead of producing fine keypoints and skeleton annotations for human subjects which may not be possible due to various factors, we propose hierarchical labeling of human poses where the classes are separated by the variations in body postures, which involve much in their appearances. One important benefit of hierarchical labeling is that the categorical error can be restricted to particular subcategories, such that it is more informative than the classic flat N-way classification. For example, consider the two yoga poses shown in Fig. 2, the *upward bow pose* and the *upward facing two-foot staff pose*. The two poses differ in the manner that the *upward*

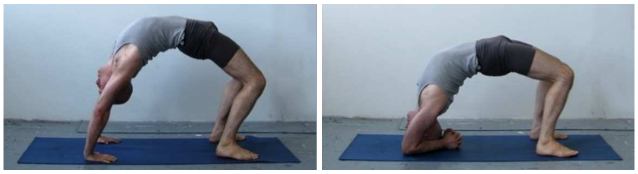


Figure 2. The *upward bow pose* (left) and *upward facing two-foot staff pose* (right). Both the poses have same superclass called *upfacing wheel pose*.

[*] Authors contributed equally to this work.
[§] https://sites.google.com/view/yoga-82/home

Table 1. Comparison of human pose datasets.

Datasets	#Train	#Test	Source	Target poses
MPII [4]	25,000	-	YouTube	Diverse
LSP [17]	1,000	1,000	Flickr	Sports
LSP-Ext. [18]	1,0000	-	Flickr	Sports
FLIC [23]	6,543	1,016	Movies	Diverse
SHPD [6]	18,334	5,000	Surveillance	Pedestrian
Yoga-82	21,009	7,469	Bing	Yoga

facing two-foot staff pose puts headstand together with the *upward bow pose*. Apart from this, both the poses have many similarities such as the way in which the back is bent (*wheel pose*), the orientation of faces, and the placement of legs. Therefore, both these poses can be put in a single superclass pose called *up-facing wheel pose*. An advantage of this type of label structure is that, once the network learns that it is a type of up-facing pose, it will not confuse it with any down-facing poses such as the *cat-cow pose* which have a similar *wheel* type structure. Further separation of the classes at the lowest level will help the network to focus on specific parts of the body. For example, the headstand part of the *upward facing two-foot staff pose*.

In this work, building on the concept of fine-grained hierarchical pose classification (as discussed above), we propose a large-scale yoga dataset. We choose yoga activity since it contains a wide variety of finely varying complex body postures with rich hierarchical structures. This dataset contains over 28.4K yoga pose images distributed among 82 classes. These 82 classes are then merged/collapsed based on the similarities in body postures to form 20 superclasses, which are then further merged/collapsed to form 6 superclasses at the top level of the hierarchy (Fig. 3). To the best of our knowledge, Yoga-82 is the first dataset that comes with class hierarchy information. In summary, the main contributions of this work are as follows.

- We propose the concept of fine-grained hierarchical pose classification and propose a large-scale pose dataset called Yoga-82, comprising of multi-level class hierarchy based on the visual appearance of the pose.
- We present performance evaluation of pose recognition on our dataset using well-known CNN architectures.
- We present modifications of DenseNet in order to utilize the hierarchy of our dataset for achieving better pose recognition.

Related work. Human pose estimation has been an important problem in computer vision and many benchmark datasets have been proposed in the past. Summary of some of the most commonly used human pose datasets is presented in Table 1. Many of these datasets are collected from the sources such as online videos, movies, images, sports videos, etc. Some of them provide rich label information but lack in human pose diversity. Most of the poses in these datasets ([4], [6], and [23]) are of standing, walking, bending, sitting, etc. and not even close to comparison with complex yoga poses (Fig. 1). Chen *et al.* [6], recently observed that the images considered for annotations are of very high quality with large target objects. For example, in the MPII dataset [4], around 70% of the images consists of human objects with height over 250 pixels. Thus, without much diversity in human poses and target object size in these datasets, they can not meet the high-quality requirements of applications such as behaviour analysis. Our proposed Yoga-82 dataset is very different from these datasets in the two aspects discussed above. We choose yoga activity, which we believe consists of some of the most diverse and complex examples of human poses. Furthermore, the images considered from the wild are with different viewpoints, illumination conditions, resolution, and occlusions. Few works have been done on yoga pose classification for applications such as self training [5, 26, 21, 15]. However these works involve yoga dataset with a less number of images or videos and does not consider vast variety of poses. Hence, they lack in generalization and are far from complex yoga pose classification.

2. The Yoga-82 Dataset

Data Acquisition. The dataset contains yoga pose images downloaded from web using the Bing search engine. The taxonomy about yoga poses (name and appearance) is collected from various websites and books [16, 19, 2, 1]. Both Sanskrit and English names of yoga poses were used to search for images and the downloaded images were cleaned and annotated manually. Every image contains one or more people doing the same yoga pose. Furthermore, images have poses captured from different camera view angles. There are a total of 82 yoga pose classes in the dataset. The dataset has a varying number of images in each class from 64 (min.) to 1133 (max.) with an average of 347 images per class. Some of the images are downloaded from a specific yoga website. Hence, they contain only yoga pose with clean background. However, there are many images with random backgrounds (e.g., forest, beach, indoor, etc.). Some images only contain silhouette, sketch, and drawing version of yoga poses and they were kept in the dataset as yoga pose is more about the overall structure of body and not the texture of clothes and skin. For the sake of easiness in understanding and readability, here we use only English names for the yoga poses. However, Sanskrit names are available as well in the dataset for reference.

Label Hierarchy and Annotation. Existing pose datasets (Table 1) available publicly for evaluation do not impose hierarchical annotations. Hierarchical annotations can be beneficial for part-based learning in which few parts of

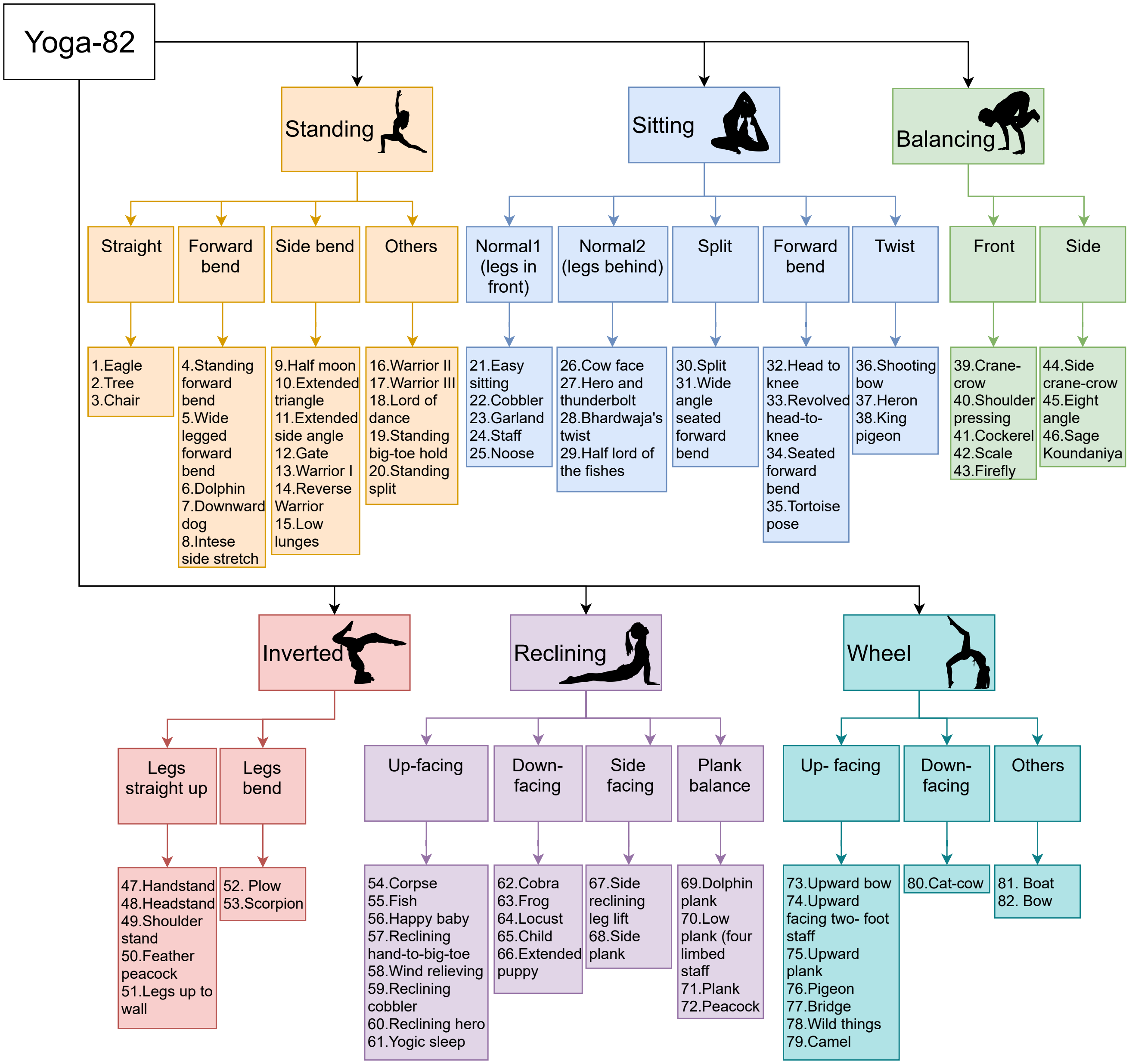


Figure 3. Yoga-82 dataset label structure. Hierarchical class names at level 1, 2, and 3.

the network will learn features based on the hierarchical classes. Many hierarchical networks have been observed to perform better as compared to their baseline CNN models [28, 9]. Hierarchical annotations are beneficial for learning the network as they provide rich information to users not only about the pose names but also about the body postures (*standing*, *sitting*, etc.), the effect on the spine (e.g., *forward bend*, *back bend* in *wheel pose*, etc.), and others (e.g. *down-facing* or *up-facing*).

Our labels are with a three-level hierarchical structure where the third level is a leaf node (yoga pose class). There are 6, 20, and 82 classes in the first-, second-, and third-levels, respectively as illustrated in Fig. 3. References for these classes have been collected from websites and books [24, 16, 19, 2, 1, 3]. There is no established hierarchy in yoga poses. However, *standing*, *sitting*, *inverted*, etc. are well defined as per their configuration. In this work, we have taken guidelines from [24, 16, 19, 2, 1] in order to define the first level classes and defined a new class (*wheel*) based on the posture of the subject's body in a certain pose. The second level further divides the first level classes into different classes as per subject's body parts configuration. However, it is hard to define 82 leaf classes in 6 super classes perfectly. Yet, we have made an attempt to briefly

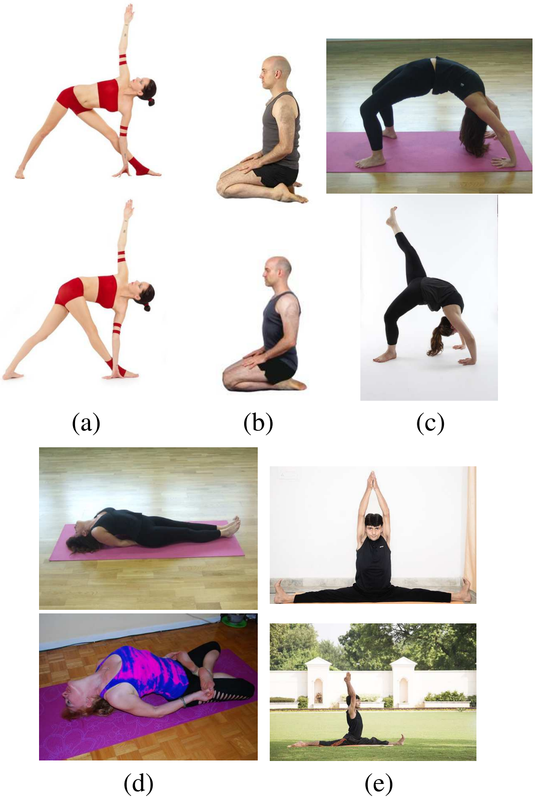
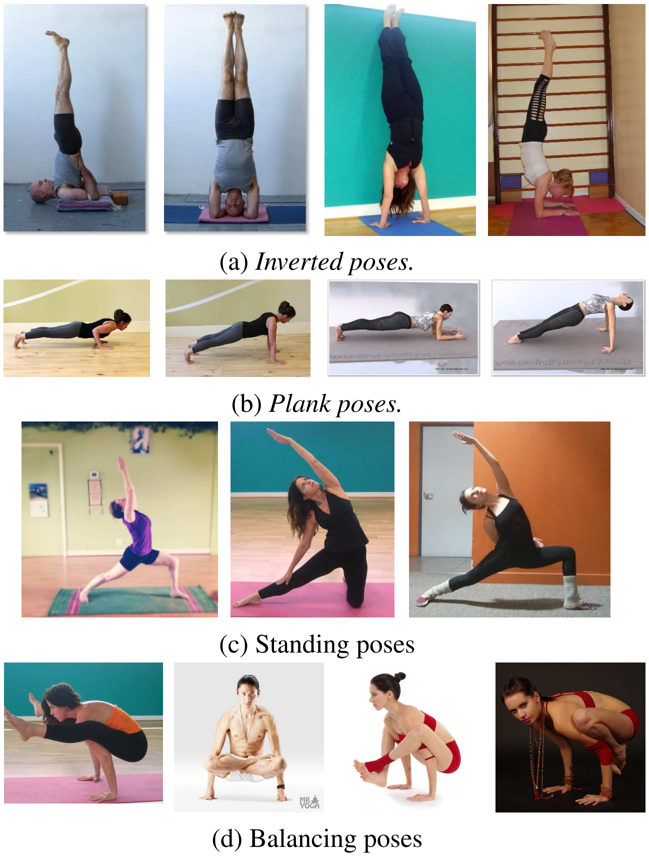


Figure 4. Different variations of the same pose in one class (a) *Extended triangle pose* and *revolved triangle pose*, (b) *Hero pose* and *thunderbolt pose*, (c) *Upward bow pose* and its variation, (d) *Fish pose* and its variation, and (e) *Side spilt* and *front split pose*.



(a) *Inverted poses.*

(b) *Plank poses.*

(c) Standing poses

(d) Balancing poses

Figure 5. Some example of different classes in very similar appearances.

describe the 6 first level classes as follows.

Standing: Subject is standing while keeping their body straight or bending. Both or one leg will be on the ground. When only one leg is on the ground, the other leg is in air either held by one hand or free.

Sitting: Subject is sitting on the ground. Subject's hip will be on ground or very close to the ground (e.g., *garland pose*).

Balancing: Subject is balancing their body on palms. Both the palms are on ground and the rest of the body is in air. Subject's body is not in the inverted position.

Inverted: Subject's body is upside down. Lower body is either in air or close to the ground (e.g., *plow pose*).

Reclining: Subject's body is lying on the ground. Either spine (*upward facing*) or stomach (*downward facing*) or side body (*side facing*) touching or very near to the ground, or subject's body is in 180° angle (approximately) alongside ground (e.g, *plank poses*).

Wheel: Subjects body is in half circle or close to half circle on ground. In *upward facing* or *downward facing poses*, both the palm and the feet will touch the ground. In *others* category, either only hip or stomach will touch the ground; the other body parts will be in air.

All the class levels are shown in Fig. 3. The class names and their images are based on the body configuration. For example, *forward bend* is a second-level class in both standing and sitting. As it is clear from its name, this class includes poses where the subject needs to bend forward while standing or sitting. Similar names are given to the other second-level classes. The poses that do not fit in any second-level classes are kept in *others* (*standing*), *twist* (*sitting*), *normal1* (*sitting*), *normal2* (*sitting*), etc.

Analysis over our Dataset. Some of the poses have variations of their own. For example, *extended triangle pose* and *revolved triangle pose*, *head-to-knee pose* and *revolved head-to-knee pose*, *hero pose*, *reclining hero pose*, etc. These poses are kept in the same class or different classes in the third level based on the differences in their visual appearances. For example, *extended triangle pose* and *revolved triangle pose* (Fig. 4(a)) are in the same class, while *head-to-knee pose* and *revolved head-to-knee pose* are in different classes.

Some completely different poses (e.g. *hero pose* and *thunderbolt pose*, *side spilt pose* and *front split pose*, etc.) are kept in same third-level classes as they appear to be very

Table 2. Performance of the state-of-the-art architectures on Yoga-82 using third-level class (82 classes).

Architecture	Depth	# Params	Model size	Top-1	Top-5
ResNet-50	50	23.70 M	190.4 MB	63.44	82.55
ResNet-101	101	42.72 M	343.4 MB	65.84	84.21
ResNet-50-V2	50	23.68 M	190.3 MB	62.56	82.28
ResNet-101-V2	101	42.69 M	343.1 MB	61.81	82.39
DenseNet-121	121	7.03 M	57.9 MB	73.48	90.71
DenseNet-169	169	12.6 M	103.4 MB	74.73	**91.44**
DenseNet-201	201	18.25 M	149.1 MB	**74.91**	91.30
MobileNet	88	3.29 M	26.7 MB	67.55	86.81
MobileNet-V2	88	2.33 M	19.3 MB	71.11	88.50
ResNext-50	50	23.15 M	186.1 MB	68.45	86.42
ResNext-101	101	42.29 M	340.2 MB	65.24	84.76

Table 3. Classification performances of our three variants. L1, L2, and L3 stand for the first-, second-, and third-level classification, respectively.

Network	# Params	Top-1			Top-5		
		L1	L2	L3	L1	L2	L3
Variant 1	18.27 M	83.84	85.10	79.35	99.40	97.08	93.47
Variant 2	18.27 M	89.81	84.59	79.08	99.83	97.03	92.84
Variant 3	22.59 M	87.20	84.42	78.88	99.69	97.28	92.66

similar to each other. For example, *hero pose* and *thunderbolt pose* (Fig. 4(b)) have a minute difference that legs to be placed near the thighs and under the thighs, respectively. Other class separations were carefully made using suggestions of three of the authors based on the appearance of the poses.

Our dataset is very challenging in terms of similarity between different classes. There are many classes at the third level that are very similar to each other that are treated as different poses. For example, *inverted poses* (level 2) has poses that differ from each other if the subject is in inverted position and balancing their body up straight on hands (*handstand pose*), head (*headstand pose*), or forearms (*feathered peacock pose*) as shown in Fig. 5(a). Similarly, *plank poses* differ from each other based on plank's height from the ground and whether its on palms or forearms (Fig. 5(b)). Few similar poses are shown in Fig. 5. These poses make the dataset very challenging as this is not covered in any previous pose datasets [5].

3. Experiments

We divide our experiments into two parts. In the first part, we conduct benchmark experiments on the Yoga-82 dataset. In the second part, we present three CNN architectures that exploit the class hierarchy in our Yoga-82 dataset to analyze the performance using hierarchical labels.

3.1. Benchmarking Yoga-82 Dataset

We evaluate the performance of several popular CNN architectures on the Yoga-82 dataset that have recently achieved state-of-the-art accuracies on image recognition tasks on the ImageNet [7] dataset.

Benchmark models. Table 2 gives a comprehensive list of network architectures that we used for benchmarking our Yoga-82 dataset. They are selected such that they differ in structures, depth, convolutional techniques, as well as computation and memory efficiencies. For example, ResNet [10, 11] and DenseNet [14] differ in the manner the skip connections are applied. MobileNet [13, 22] uses separable convolutions for better computational and memory efficiency. ResNext [25] uses group convolutions for better performance and reduces space-time complexity.

Experimental protocol and setting. All our experiments were conducted on a system with Intel Xeon Gold CPU (3.60 GHz × 12), 96 GB RAM, and an NVIDIA Quadro RTX 8000 GPU with 48 GB memory. We used Keras with Tensorflow backend as the deep learning framework. For training the networks, we used stochastic gradient descent (SGD) with momentum 0.9. We started with a learning rate of 0.003 and decreased it by the factor of 10 when the validation loss plateaus. All weights were initialized with the orthogonal initializer. We did not apply any data augmentation techniques on the input images. All images were resized to 224 × 224, before feeding into the networks. We split our dataset into training and testing sets, which contain 21009 and 7469 images, respectively. As mentioned earlier, we provide train-test splits of the dataset for consistent evaluation and fair comparison over the dataset in future.

Results. The results of the benchmark experiments are shown in Table 2. Both the top-1 and top-5 classification accuracies are reported. We observe that deeper net-

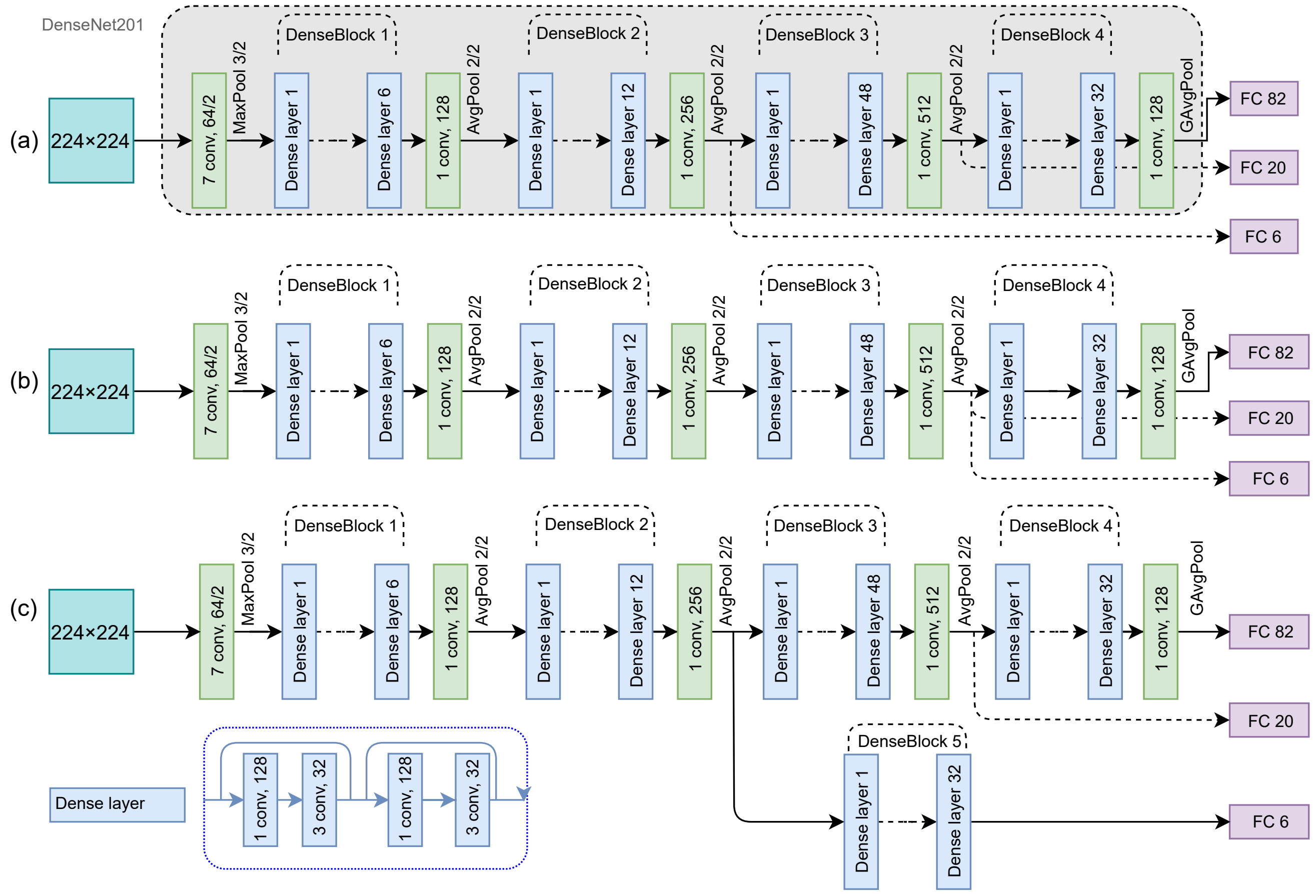


Figure 6. DenseNet-201 modified hierarchical architectures.

works have a clear edge over their shallower versions. For example, 101-layer ResNet architecture (ResNet-101) outperformed its 50-layer variant (ResNet-50). Furthermore, deeper networks with dense skip connections, such as the DenseNet architectures, performed better than the networks with sparse skip connections. DenseNet-201 gives the best performance, achieving top-1 classification accuracies of 74.91% .

3.2. Hierarchical Architectures

Our dataset, Yoga-82, provides a rich hierarchical structure in the labels, which can be utilized in order to enhance the performance of pose recognition. Based on [28], we modify Densenet-201 architecture to make use of the structure. That is, due to the hierarchical structure, label prediction in any level can be deducted from the third-level prediction results. However, since the hierarchy is based much on visual similarity between different poses, training with upper-level labels may help lower-level boost the prediction in lower-level label, and vice versa.

Variant 1. In this variant, hierarchical connections are added in DenseNet-201 after DenseBlock 2 and DenseBlock 3 for class level 1 (6 classes) and class level 2 (20 classes), respectively, as shown in Fig. 6(a). Coarser classes are classified at the middle layers and finer classes are at the end layers of the network. The intuition behind this variant is to utilize hierarchy structure in the dataset. Initial-to-mid layers learn to classify the first level and the details in the input image is passed on to next layers for the second-level classification, and so on. Layers shared by all three levels (up to DenseBlock 2) learn basic structure of pose and further layers refine it for specific details. The branch for the first-level classification applies batch normalization and the ReLU activation, followed by global average pooling. The same applies to the branch for the second-level classification. The main branch is for the third-level classification with 82 classes. Softmax-cross entropy loss is computed for all three levels and weighted sum is evaluated as the final loss as follows:

$$L = \sum_{i=1}^{3} w_i \sum_{j=1}^{N_i} t_{ij} \log(y_{ij}), \qquad (1)$$

where N_i $(i = 1, 2, 3)$ is the number of labels in level l, i.e., 6, 20, and 82 for i = 1, 2, and 3, respectively. $t_{ij} \in \{0, 1\}$ is ground truth for label j of level l. y_{ij} is the output of the

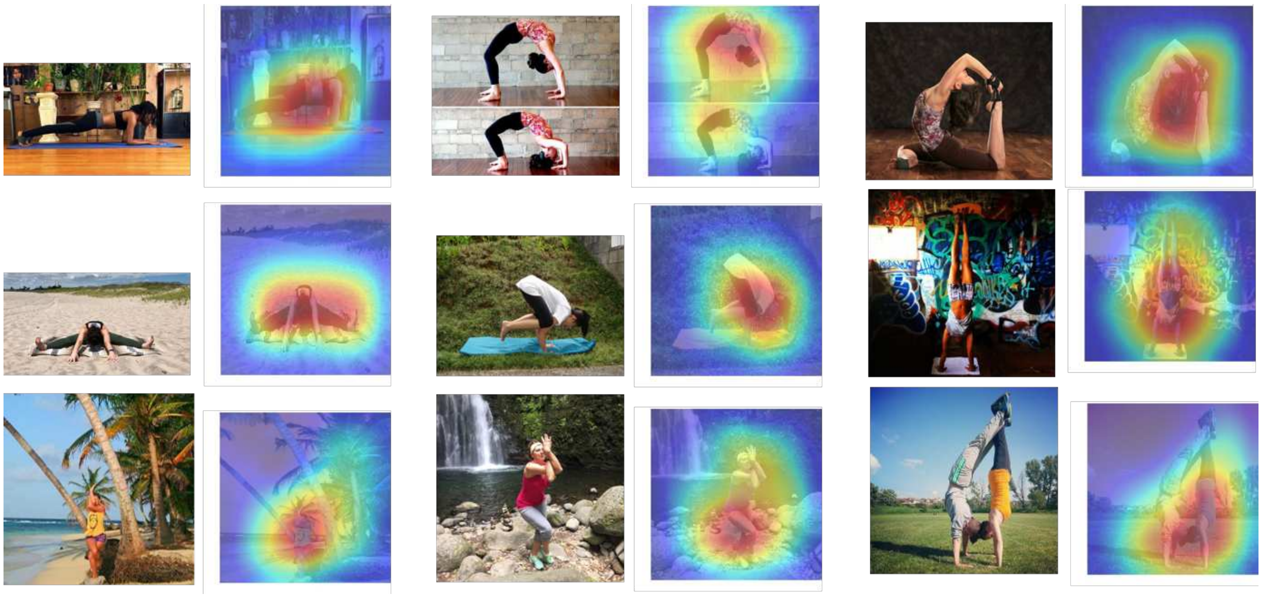


Figure 7. Activation maps learned using variant 2.

softmax layer. w_i is the weight for level i. All weights are set to one as we consider that all levels are equally important.

Variant 2. In variant 1, the first-level classifier does have access to only DenseBlock 1 and DenseBlock 2 that comprises of 6 and 12 dense layers, respectively, whereas DenseBlock 3, which classifies level 2, has 48 dense layers. Hence, the accuracy of the first-level classifier may be degraded in variant 1 because of insufficient representation capability. Since our focus is to classify images into classes in all three levels correctly, we make branches for the first- and second-level classifiers from the same position (Fig. 6), so that the first-level classifier can have more representation capability. We classify both levels after DenseBlock 3 as illustrated in Fig. 6(b). Batch normalization, the ReLU activation, global average pooling, and loss function are the same as variant 1.

Variant 3. Another attempt to classify all three levels equally is made in variant 3. We employ a similar architecture as variant 1, except that we add DenseBlock 5 with 32 dense layers for the first-level classifier branch (Fig. 6(c)). This variant gives more trainable parameters to the first-level classifiers while keeping the hierarchical structure of network. This variant increases the number of parameters compared to the others.

Results and Discussion. The performances of all three variants are presented in Table 3 along with the numbers of parameters. All three variants stem from DenseNet-201 and thus the numbers of parameters differ only because of the addition of DenseBlock 5 in variant 3. Clearly, the hierarchical structures boosted the performances. As shown in Tables 2 and 3, the accuracy of the third-level classifier (82 classes) was boosted from 74.91% to 79.35% with hierarchical connections added in DenseNet-201.

We can see that the third-level classifiers (L3 in Table 3) give similar accuracies varying within 1% in all three variants. In contrast, we see huge variations in the accuracy of the first-level classifier. From these results, we may say that the performance depends more on the number of layers or the parameters responsible for a certain level classifier as well as on the number of classes. For example, variant 1 uses two DenseBlocks (6 + 12 dense layers) for the first-level classification and three DenseBlocks ($6+12+48$ dense layers) for the second-level classification. This huge gap between the numbers of parameters used for the first- and second-level classification may cause the difference in performances. This gap is reduced with variant 2 whose first- and second-level classifiers branch at the same point (i.e., after DenseBlock 3). As expected, the accuracy of the first level is less than that of the second level 2 and the accuracy of the second level is less than that of the third level. Similarly, variant 3 has extra layers added for the first-level classification. Hence, the accuracies decrease in the order of the first level to the third level classifiers. Variant 3 increases the performance at the cost of additional parameters in the network. In conclusion, variant 2 can balance well.

In Fig. 7, we present the class activation maps obtained from variant 2 using [27]. It can be observed that our model responded to the person doing a certain pose. Furthermore, we observe that, for a particular pose, the model focuses on one or specific parts of the body. For example, for *eagle pose* (Fig. 7, second column), the model focused on the configuration of the legs of the person.

4. Conclusion

In this work, we explored human pose recognition from a different direction by proposing a new dataset, Yoga-82, with 82 yoga pose classes. We define a hierarchy in labels by grasping the knowledge of body configurations in yoga poses. In particular, we present a three-level hierarchical label structure consisting of 6, 20, and 82 classes in the first to third levels. We conducted extensive experiments using popular state-of-the-art CNN architectures and reported benchmark results for the Yoga-82 dataset. We present modified DenseNet architecture to utilize the hierarchy labels and get a performance boost as compared to the flat n label classification. It is evident that hierarchy information provided with dataset improves the performance because of additional learning supervision. It is visible from results that there is sufficient room for accuracy improvement in yoga pose classification. In future, we will focus on adding explicit constraints among predicted labels for different class levels.

References

[1] Wikipedia: List of asanas. https://en.wikipedia.org/wiki/List_of_asanas. 2, 3

[2] Yoga journal. https://www.yogajournal.com/poses. 2, 3

[3] Yoga sequence builder. https://www.tummee.com/. 3

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014. 1, 2

[5] Hua-Tsung Chen, Yu-Zhen He, Chun-Chieh Hsu, Chien-Li Chou, Suh-Yin Lee, and Bao-Shuh P Lin. Yoga posture recognition for self-training. In *MMM*, pages 496–505, 2014. 2, 5

[6] Qiuhui Chen, Chongyang Zhang, Weiwei Liu, and Dan Wang. SHPD: Surveillance human pose dataset and performance evaluation for coarse-grained pose estimation. In *ICIP*, pages 4088–4092, 2018. 1, 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[8] Philipe Ambrozio Dias, Damiano Malafronte, Henry Medeiros, and Francesca Odone. Gaze estimation for assisted living environments. In *WACV*, pages 290–299, 2020. 1

[9] Ruigang Fu, Biao Li, Yinghui Gao, and Ping Wang. CNN with coarse-to-fine layer for hierarchical classification. *IET Comput. Vis.*, 12(6):892–899, 2018. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016. 5

[12] Michael B Holte, Cuong Tran, Mohan M Trivedi, and Thomas B Moeslund. Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE J. Sel. Topics Signal Process.*, 6(5):538–552, 2012. 1

[13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 5

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 5

[15] Muhammad Usama Islam, Hasan Mahmud, Faisal Bin Ashraf, Iqbal Hossain, and Md Kamrul Hasan. Yoga posture recognition by detecting human joint points in real time using microsoft kinect. In *R10 HTC*, pages 668–673, 2017. 2

[16] Bellur Krishnamukar Sundara Iyengar. *Light on yoga*. New York: Schocken Books, 1965. 2, 3

[17] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, volume 2, page 5, 2010. 1, 2

[18] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, pages 1465–1472. IEEE, 2011. 2

[19] Leslie Kaminoff and Amy Matthews. *Yoga anatomy*. Human Kinetics, 2011. 2, 3

[20] Manuel Martin, Stephan Stuehmer, Michael Voit, and Rainer Stiefelhagen. Real time driver body pose estimation for novel assistance systems. In *ITSC*, pages 1–7. IEEE, 2017. 1

[21] Sen Qiao, Yilin Wang, and Jian Li. Real-time human gesture grading based on openpose. In *CISP-BMEI*, pages 1–6, 2017. 2

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 5

[23] Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, pages 3674–3681, 2013. 1, 2

[24] Mark Singleton. *Yoga body: The origins of modern posture practice*. Oxford University Press, 2010. 3

[25] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 5

[26] Santosh Kumar Yadav, Amitojdeep Singh, Abhishek Gupta, and Jagdish Lal Raheja. Real-time yoga recognition using deep learning. *Neural Comput. Appl.*, pages 1–13, 2019. 2

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 7

[28] Xinqi Zhu and Michael Bain. B-CNN: branch convolutional neural network for hierarchical classification. *arXiv preprint arXiv:1709.09890*, 2017. 3, 6