6-29-2021

Train the Neural Network by Abstract Images

Liqun Yang
Florida International University

Yan Liu
Qingdao Agricultural University

Wei Zeng
Xi'an Jiaotong University

Yijun Yang
Xi'an Jiaotong University

Recommended Citation

Yang, Liqun; Liu, Yan; Zeng, Wei; and Yang, Yijun, "Train the Neural Network by Abstract Images" (2021). *All Faculty*. 370.
https://digitalcommons.fiu.edu/all_faculty/370

PAPER • OPEN ACCESS

Train the Neural Network by Abstract Images

To cite this article: Liqun Yang *et al* 2021 *J. Phys.: Conf. Ser.* **1952** 022009

View the article online for updates and enhancements.

You may also like

- Deep Learning for Image Sequence Classification of Astronomical Events
  Rodrigo Carrasco-Davis, Guillermo Cabrera-Vives, Francisco Förster et al.
- A segmentation method combining probability map and boundary based on multiple fully convolutional networks and repetitive training
  Wenshe Yin, Yangsheng Hu, Sanli Yi et al.
- Target Detection of Marine Ships Based on a Cascade RCNN
  Lei Wu, Shicai Chen and Chengcheng Liang

Train the Neural Network by Abstract Images

Liqun Yang [1, a], **Yan Liu** [2, b, *], **Wei Zeng** [3, c], **Yijun Yang** [4, d]

[1] School of Information Science and Computing Florida International University Miami, United States
[2] College of Civil Engineering and Architecture Qingdao Agricultural University Qingdao, China
[3] School of Mathematics and Statistics Xi'an Jiaotong University Xi'an, China
[4] School of Computer Science and Technology Xi'an Jiaotong University Xi'an, China

[a] lyang028@fiu.edu, [*, b] Corresponding author e-mail:yanliu@qau.edu.cn,
[c] wz@xjtu.edu.cn, [d] yangyijun@xjtu.edu.cn

Abstract. Like the textbook for students' learning, the training data plays a significant role in the network's training. In most cases, people intend to use big-data to train the network, which leads to two problems. Firstly, the knowledge learned by the network is out of control. Secondly, the space occupation of big- data is huge. In this paper, we use the concepts-based knowledge visualization [33] to visualize the knowledge learned by the model. Based on the observation results and information theory, we make three conjectures about the key information provided by the dataset. Finally, we use experiments to prove that the artificial abstracted data can be used in networks' training, which can solve the problem mentioned above. The experiment is designed based on Mask-RCNN, which is used to detect and classify three typical human poses on the construction site.

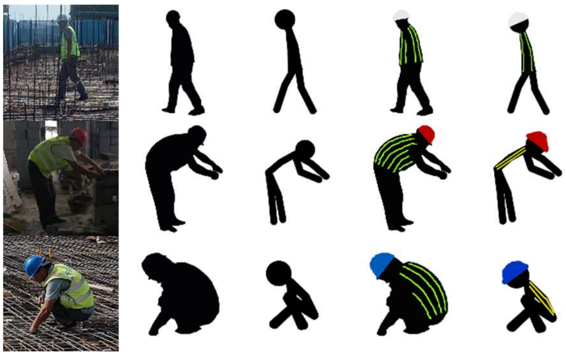
Keyword: Transfer Learning, Sim2real.

1. Introduction

Since the convolutional neural network was proposed ([14]– [17]), the technique of image recognition ushered in explosive development. However, some questions need to solve when we apply the neural network. Data dependence is one of the most serious problems. Compared with traditional machine learning methods, deep learning depends on the training data because it requires the data to reflect the underlying patterns of the data [24].

People usually intend to use big-data to solve this problem. However, big-data cannot solve everything. In [18], the author points out that with the increasing of the data scale, the accuracy of classification is increasing, but the robustness of the model is decreasing at the same time. Moreover, the accuracy improvement has an upper bound as mentioned in [13]. More important is that the noise is inevitable in a huge dataset, including irrelevant information [25], [32], error labels [23] etc. It can cause unpredictable impacts potentially. Finally, the most significant shortcoming of big-data is the great space occupation and computing cost in training.

As a model with similar working mechanisms of the human brain, rethinking the human learning process might be the best way to break the current dilemma. Teachers intend to use few representative and simplified samples in education but not many real instances to show one object's feature. The educator will take a lot of time to write the most important tool in students learning, the textbook. It is very inspiring to us. Can we use a well-designed dataset with abstract images to train the model?



Level-0 (b) Level-1 (c) Level-2 (d) Level-1f (e) Level-2f

Fig. 1 Multi-level abstract datasets.

2. Related works

To solve the low sampling efficiency and avoid security risks in the real world training, people use simulated data to train the model, called simulation to real transfer (sim2real). Among the methods of sim2real, two kinds of methods are similar to our ideas, domain adaption, and domain randomization. The purpose of the former is to map the data of different source domains and target domains into a feature space, and minimize the distance between them in the space, like [2], [3], [7], [8]. And the domain randomization intends to learn the random combinations of various characteristic variables or features, like [27], [30]. Different from our ideas, synthetic data generation requirements in these two kinds of methods are similar to the real. Essentially, they are solutions for the lack of data in networks' training.

On the contrary, in our work, we actively filter some of the real-world information. We intended to verify if the abstract data can be used in the network's training or not. Like the educator editing the textbook's content, we want to use data with a more effective form to train the network. In a word, the methods related to sim2real intend to use features to generate simulated images, whereas our work is to minimize the information provided by the image of the real world without affecting the network's final performance.

3. Motivation

Training the network with abstract data has the following advantages. First, It can eliminate the "shortcut" in the dataset and avoid networks' cheating in learning. Secondly, the abstract data takes up less space than the original data and can accelerate the network's training.

3.1. Networks' cheating

In most cases, the trained net- work's performance is the only measure in the evaluation of the training process. However, as the author mentioned in [9], some of the information in the dataset can be the "shortcut" to complete tasks, which leads to the training's failure. The model uses a simple "shortcut" to complete a complex learning task called Clever Hans Effect. In practice, this phenomenon has been observed in the early version of BERT [5] when it completes the argument reasoning comprehension task. In [22], the author shows that the model does not understand the task. It makes the judgment just based on the statistical feature of the dataset. In [10], the author points out that in some cases, the CNN-based model classifies the image just depending on the texture unexpectedly. In other words, the CNN is misled by the training dataset, it does find the best way to complete the task, but that is not what we want it to learn. In [20], the author points out that the unsupervised learning of disentangled representations is fundamentally impossible without inductive biases on both the models and the datasets. In the examples above, the network makes the correct judgment based on a hidden trick but not the logic we want it to learn. Even though we can visualize some of the knowledge learned by networks as [33]–[36] show, we still cannot impact their learning.

Therefore, to eliminate the "shortcut" hidden in the dataset, control the information provided by the training data is the last option. Using well-designed abstract data can purify the knowledge, speed up the training, and control the information learned by the neural network.

3.2. Our Contribution:

This work verifies that the abstract images can also be used in the network's training. Using the data with reasonable abstraction will not affect the performance of the model after training. It can reduce the size of the dataset and avoid models' cheating in the learning process effectively.

4. Network cheating in human pose detection

Because the learning process is out of control, the network's cheating (Clever Hans Effect) is inevitable. This section demonstrates an example of a network's cheating behavior to show the importance of eliminating the "shortcut" hidden in the dataset.

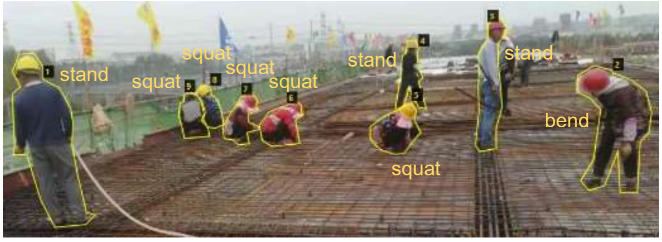
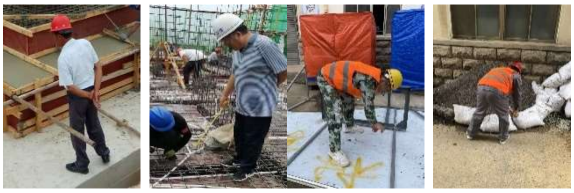


Fig. 2 Labels of the dataset.



(a) Quality supervisor (b) Concrete pouring worker

Fig. 3 Two typical works on the construction site.

4.1. Introduction of the Model

To ensure workers' health and avoid labor injury caused by long working hours, net- works for human pose detection are wildly used to monitor worker movements. Based on Mask-RCNN [12], we design a network to monitor the pose of human to prevent work-related musculoskeletal disorders. [28] shows a study of occupational mobility in a cohort of construction workers. It shows that disorders of the back and spine are one of the major causes of early retirement. Therefore, this model is designed to recognize three main poses related to workers' back and spine on the construction site, standing, bending, and squatting (see [31]). Because there are only three kinds of the pose, to simplify the computing, we do not use the network to extract human skeleton information but identify the region with a human directly, as Fig. 2 shows. VGG Image Annotator (VIA) [6] is used to mark the label.

4.2. Clues of the Network's Cheating

The model is trained by the real data (see Fig. 1(a)), and its mAP (IoU = 50%) can get to 89% on the similar data (test dataset of workers). However, when we use another test case of workers, the mAP decreases to 71% rapidly, which is caused by the accuracy decrease in classification but not the detection. This phenomenon catches our attention. After analyzing the dataset, we find some clues about the network's cheating. On the construction site, workers with different types of work are usually with different types of cloth. In Fig. 3, we present two typical work on the construction site, quality supervisor and concrete pouring worker. For the clothing, the former wear usual clothes, and the latter are required to wear the special vest for their safety. Meanwhile, the former usually stands and plays the role of a supervisor in most cases. And the latter often needs to bend or squat because of their work. There is a "shortcut", classifying the pose of human by their cloth.

To verify our hypothesis, we create a special dataset based on the original dataset shown in Fig. 10(a). In the training dataset,

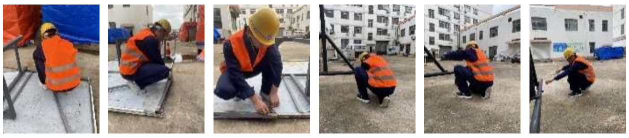


Fig. 4 Images of designed "bend" samples.

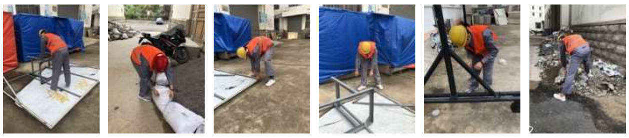


Fig. 5 Test cases used to detect network cheating.

We replace all the "bend" samples with the same amount of designed images as Fig. 4 shows. Unlike the original images, the worker is always with an orange fluorescent vest and yellow helmet in this group of pictures.

After training, we use it to identify another group of designed images (240 images) as Fig. 5 shows. In this group, the worker with the same equipment, but all their pose is "squat".

As a result, there are 63% samples are classified as "bend", 30% are detected as "squat", 3% are classified as "stand", and 4% are not detected. This result indicates that **to get a trustworthy model based on neural network, control what is learned by the network is necessary.** However, there isno effective way for us to control the learning process of the network. Controlling the information provided by the datasetis the last way we can choose. In the following part of this paper, we demonstrate our work in this aspect.

5. Basis of abstraction

To eliminate the "shortcut" in the dataset, we try to minimize the quantity of information in the input image on the premise of not affecting model training as much as possible. To generate the abstract data, we need to know what kind of information is useful.

One of the abstract basis is based on the entropy analysis method. In [11], [21], the author proposes a method to quantify the input information that is encoded in a specific intermediate layer of a DNN. The entropy measures how much input information is neglected when the DNN extracted the feature of this layer, and we can use low entropy part to visualize the region with more information. The information discarding is formulated as the conditional entropy $H(X')$ of the input, given the intermediate-layer feature $f = f(x)$, as Eq. 1 shows

$$H(X') \text{ s.t. } \forall \mathrm{x}' \in X', f(x') - f^* \leq \tau. \tag{1}$$

where X' denotes a set of images which corresponding to the concept of a specific object instance, $\tau$is a small positive value which represent the tolerance. $x'$is assumed following an $i.i.d$ Gaussian distribution $x' \sim \mathrm{N}(x, \Sigma)$where Σrrepresents the covariance matrix. To reduce the computing complexity, Σ is simplified as diag $(\sigma_1^2, \ldots, \sigma_n^2)$, where n is the pixel amount of input image and $\sigma_i^2 = E[(x_i - E[x_i])^2]$. In this way, the assumption of the Gaussian distribution ensures that the entropy $H(X')$ of the entire image can be decomposed into pixel-level entropies $\{H_i\}$ as the following equation,

$$H(X') = \sum_{i=0}^{n} H_i \tag{2}$$

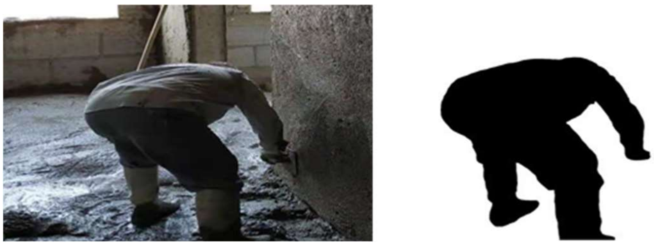

(a) Colorful Image (b) Binary Image

Fig. 6 Comparing with the colorful image, the binary images has lower 1d entropy expectation.

where $H_i = \log \sigma_i + \frac{1}{2} \log 2\pi e$. This entropy is called 1d entropy when H_i is the entropy of one pixel. For a colorful image as the left one shown in Fig. 6, the entropy is high, which can provide more features. However, for a binarized image, the entropy is much lower, which can provide fewer features. Therefore, for the images with the same content, **the binarized image contains less information than the colorful one** (Cor. 1) as Fig. 6 shows.

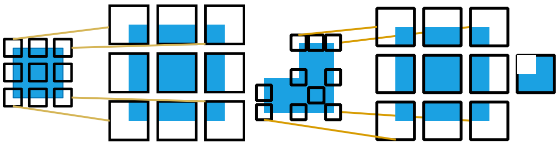

(a) Simple Boundary (b) Complex Boundary

Fig. 7 Comparing with the shape with complex boundary, the shape with simple boundary contains less information.

Corollary 1: for the images with the same content, the binarized image contains less information than the colorful one.

One step further, we can expand the definition of Eq. 2 by the concept of the "super pixel". Here, the pixel is not a point but a set contains one pixel and its neighbors, which can be defined by the radium. We can use a tuple (i, j) to briefly describe a super pixel, where i is the pixel value of the center pixel, and j is the average value of all its neighbors. For Eq. 2, if the object is super pixel, the result is called 2d entropy. Based on the definition, we can infer that for two images with the same content, **the simple boundary contains less information than the complex one** (Cor. 2). For example, in Fig. 7, we use the box with four pixels to sample the shape. For an image with simple boundary, like the left one in Fig. 7, We can get nine kinds of samples from it. For an image with complex boundary, like the right one in Fig. 7, we can get ten kinds of samples from it.

Corollary 2: For the image with the same content, the simpleboundary contains less information than the complex one.

Another abstract basis is based on the results of knowledgevisualization of CNN. In [34], the author provides a method tovisualize the pattern learned by the CNN. We use it to analyzeour dataset and draw the heat map of inference score (see Fig.8). Comparing with the details of the image, the model seemsmore sensitive to the boundary of the region, which indicatesthat **the information related to the boundary of the regionplays an important role in the network's training** (Cor. 3).

Corollary 3: The information related to the boundary issignificant in networks' training.

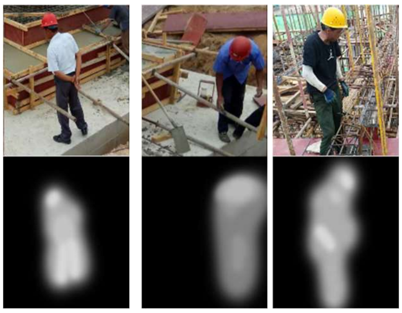


Fig. 8 The boundary of the region plays a significant role in the network's training.

6. Multi-level abstraction

Based on corollaries mentioned above, we abstract the samples at five levels as Fig. 1 shows. To measure the information of images, we calculate 1d entropy and use the pixel and its 8-neighborhoods to calculate 2d entropy based on Eq. 2. In our experiments, (H1d, H2d) is used to represent the information contains by the images.

Level-0: Level-0 data is the images collected fromthe construction site directly without changing, whose averageentropy is (5.76, 11.85) (see Fig. 1(a)).

Level-1: Based on Cor. 1, removing the information hidden in the colorful pixels can reduce the information provided by the image. And based on Cor. 3, the boundary of the region need to be kept. Therefore, we use the silhouette of the original image as the level-1 abstraction whose average entropy is (0.56, 1.28) (see Fig. 1(b)).

Level-2: Based on Cor. 2, we further simplify the information stored in the boundary. Human pose detection requires a model to represent the human pose, and we want simplify the boundary of the model as much as possible. There are two kinds of components in the stick-man model, as Fig. 9 shows, and its boundary is much simpler than the silhouette (circular border and straight border). Therefore, we use the stick-man to represent the human pose in the images as level- 2 abstraction whose average entropy is (0.48, 0.96) (see Fig. 1(c)).

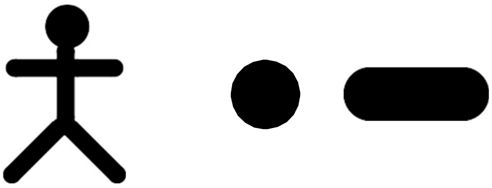


Fig. 9 Stick-man model and its components.

Level-1f and level-2f: As Fig. 8 shows, the hat and cloth whose color contrasting with the background are obvious in the heat map of inference score, which means they play a more important role in the classification of the human pose. Moreover, [10] point out that CNN-based models are more interested in the texture. To explore the importance of these features in network training, we add these feature on level-1 and level-2 data to get level-1f and level-2f whose average entropy are (0.95, 2.12) and (0.76, 1.38) (see Fig. 1(d) and Fig. 1(e)).

7. Experiment

7.1. Network and Dataset

The experiments are based on a personalized Mask-RCNN based on [1]. To make our experiments easily reproducible, we change the default settings of Mask-RCNN provided by [1] as little as possible and use Colab [4] with GPU acceleration as our experiment environment. For the dataset, the details of the distribution are shown in Tab I. Some of the samples of these five datasets are shown in Fig. 7-A.

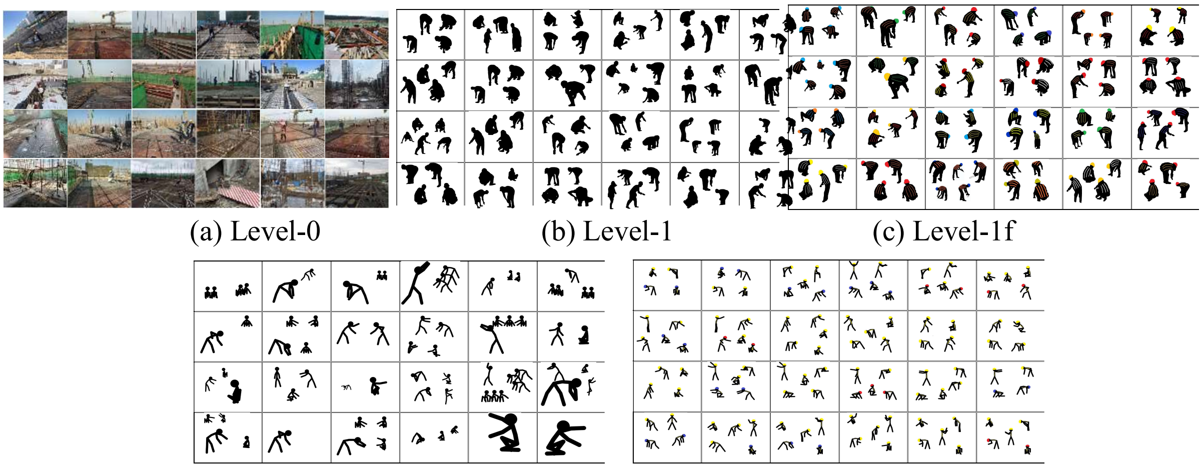
Table 1. Content of the datasets.

	Dataset	Bend	Squat	Stand	Scenes
Train	Level-0	88	209	582	240
	Level-1	269	282	434	240
	Level-2	282	318	274	240
	Level-1f	269	282	434	240
	Level-2f	282	318	274	240
Test	Level-0	26	54	216	80
	Level-1	85	64	191	80
	Level-2	69	97	110	80
	Level-1f	85	64	191	80
	Level-2f	69	97	110	80

We select pre-trained ResNet-101 as the backbone. There are two image dataset, ImageNet [26] and COCO [19] which are used to pre-train the backbone. The accuracy of the COCO- based model is 89.48%, and the accuracy of the ImageNet- based model is 47.8%. In the following part, to control the variable, all the model's backbone is pre-trained by the COCO.

7.2. Experiment and analysis

In the experiment, we train five models from scratch to fine-tuned with the same configuration as Tab. II shows and test their performance on a test dataset of workers. As Fig.



(a) Level-0 (b) Level-1 (c) Level-1f

(d) Level-2 (e) Level-2f

Fig. 10 Datasets used in the experiment.

Table 2. Basic training configura tion.

Learning rate		Optimizer	epochs	batch size
0.01 (with decay)		SGD [29]	150	2

Table 3. The accuracy decreasing.

Dataset	Accuracy Decreasing
Level-0	0.1821
Level-1f	0.1801
Level-1	0.1792
Level-2f	0.1159
Level-2	0.1107

12(a) shows, the models' performances trained by level-1 and level-1f are very close to the model trained by the real data (level-0), which means the level-1 based abstraction is effective.Compared with the level-1 data, the model trained by level-1f is a bit worse, which means that the feature cannot help the network improve its performance in this level of abstraction. On the contrary, compared with the level-2, the model trained by level-2f performs better, which means the feature helps the model classify the human pose. We can infer that for a specifickind of feature, its effectiveness could be different in different situation.

As a reference, we test the models' performances on a test dataset based on a dataset with athletes as Fig. 11 shows, and the result is shown in Fig. 12(b). On this dataset, the model's performance trained by level-1 data is almost the same as the model trained by level-0 data, which verifies again for the effectiveness of level-1 abstraction.

We calculate the accuracy difference between the models' performances on the worker dataset and the player dataset.As Tab. III shows. the performances of models trained by datasets with features (level-1f and level-2f) decrease more dramatically, which means this part of information hinder the network's recognition of human pose on the player dataset. Moreover, the accuracy decreases are positively correlated to the information quantity of the dataset. The more information the model learns, the more dramatically accuracy decreases. It indicates that there are some kinds of unknown but essential differences between the two datasets (worker dataset and player dataset), which makes not all the knowledge learned from the datasets based on the former can be used in the classificationof the latter. In other aspects, it means the model trained by the abstract data is easier to transfer, which is another advantageof the abstract data.

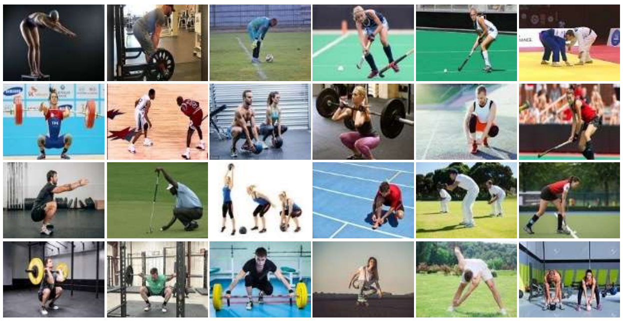
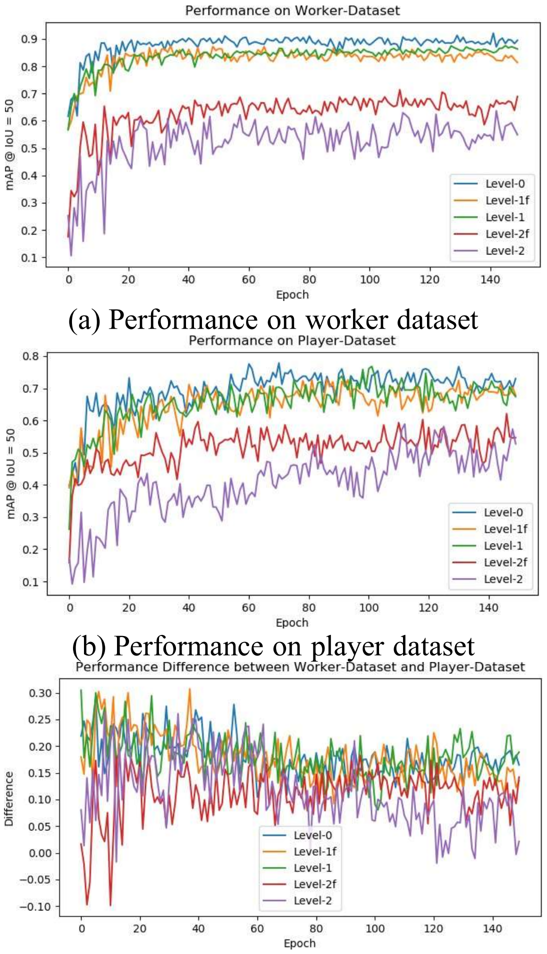


Fig. 11 Test dataset of athletes.

Finally, we compare the scale and time cost of the two datasets with similar effectiveness, level-1 and level-0 as Tab. 4 shows. In these two aspects, the level-1 abstract data has clear advantages.

Tab. 4 Dataset scale comparing.

	Level-1	Level-0
Time cost per step (s)	0.772	2.11
Scale (Mb)	10.1	335



(a) Performance on worker dataset

(b) Performance on player dataset

(c) Performance decrease

Fig. 12 Performance of level-0, level-1, level-1f, level-2, level-2f on the worker dataset and player dataset.

8. Summary and discussion

This paper verifies that the data with meaningful abstrac- tion can be used in training. However, there are still some limitations.

8.1. Question about the abstract level

In this paper, we use five levels of abstraction (level-0, level-1, level-1f, level-2, and level-2f). Someone may question it because there is no mathematical model for the abstraction process. We do not deny that this is a shortcoming of this paper, although we list the abstract theoretical basis. However, as a preliminary exploration in this field, our paper proves the feasibility of using abstract data to train neural networks, laying the foundation for further exploration in the future.

8.2. Question about human pose label

In this paper, we do not use the traditional human pose representation (skeleton information) but use the region with mark directly. Someone

would doubt that the result in this paper is not representativeof human pose detection and classification. Above all, this strategy satisfies the application's requirement and reduces the computing complexity, which is significant for the potential application platform device. Indeed, this nontraditional strategymight cause some differences between our models' performance and traditional pose detection models. However, this paper's main contribution is not in the human pose detection but the verification of the utilization of training based on abstract data.All models are based on the same strategy. Therefore, the conclusion in this paper is meaningful and trustworthy.

In conclusion, this paper verifies that the data with meaning-ful abstracts can train the network. It has two main advantages.First, it eliminates the “shortcut” hidden in the dataset, which guarantees the training result is trustworthy. Second, it has a clear advantage on the space occupation comparing with the original data. Moreover, our experiments verify the correctness of the visualization method mentioned in [34]. We believethat the well-designed abstract dataset will replace the big-dataused in the neural networks’ training in the future.

References

[1] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https ://github.com/matterport/MaskRCNN , 2017.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. Machine learning, 79(1):151–175, 2010.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. Advances in neural information processing systems, 19:137, 2007.

[4] Ekaba Bisong. Google colaboratory. In Building Machine Learning and Deep Learning Models on Google Cloud Platform, pages 59–64. Springer, 2019.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[6] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In Proceedings of the 27th ACM International Conference on Multimedia, MM ’19, New York, NY, USA, 2019. ACM.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In International conference on machine learning, pages 1180–1189. PMLR, 2015.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Franc¸ois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. The journal of machine learning research, 17(1):2096–2030, 2016.

[9] Robert Geirhos, Jo¨rn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11):665– 673, 2020.

[10] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231, 2018.

[11] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. Towards a deep and unified understanding of deep neural models in nlp. In International conference on machine learning, pages 2454–2463. PMLR, 2019.

[12] Kaiming He, Georgia Gkioxari, Piotr Dolla´r, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[13] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In European Conference on Computer Vision, pages 67–84. Springer, 2016.

[14] Yann Le Cun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In Proceedings of the 2nd International Conference on Neural Information Processing Systems, pages 396–404, 1989.

[15] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backprop- agation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989.

[16] Yann LeCun, Le´on Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[17] Yann LeCun et al. Generalization and network design strategies. Connectionism in perspective, 19:143–155, 1989.

[18] Suhua Lei, Huan Zhang, Ke Wang, and Zhendong Su. How training data affect the accuracy and robustness of neural networks for image classification. 2018.

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

[20] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Scho¨ lkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. arXiv preprint arXiv:1811.12359, 2018.

[21] Haotian Ma, Yinqing Zhang, Fan Zhou, and Quanshi Zhang. Quantifying layerwise information discarding of neural networks. arXiv preprint arXiv:1906.04109, 2019.

[22] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. arXiv preprint arXiv:1907.07355, 2019.

[23] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels, 2019.

[24] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009.

[25] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694, 2017.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015.

[27] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. arXiv preprint arXiv:1611.04201, 2016.

[28] Uwe Siebert, D Rothenbacher, U Daniel, and Hermann Brenner. Demon- stration of the healthy worker survivor effect in a cohort of workers in the construction industry. Occupational and environmental medicine, 58(12):774–779, 2001.

[29] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In International conference on machine learning, pages 1139–1147. PMLR, 2013.

[30] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 23–30. IEEE, 2017.

[31] Liqun Yang et al. mask rcnn data augmentatuion. https//doi.org/10.6084/m9.figshare. 11857956.v2.

[32] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530, 2016.

[33] Hao Zhang, Sen Li, Yinchao Ma, Mingjie Li, Yichen Xie, and Quanshi Zhang. Interpreting and boosting dropout from a game-theoretic view. arXiv preprint arXiv:2009.11729, 2020.

[34] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song- Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[35] Quanshi Zhang, Xin Wang, Ruiming Cao, Ying Nian Wu, Feng Shi, and Song-Chun Zhu. Extracting an explanatory graph to interpret a cnn. IEEE transactions on pattern analysis and machine intelligence, 2020.

[36] Quanshi Zhang, Xin Wang, Ying Nian Wu, Huilin Zhou, and Song- Chun Zhu. Interpretable cnns for object classification. arXiv preprint arXiv:1901.02413, 2019.