TCNL: Transparent and Controllable Network Learning via Embedding Human-Guided Concepts

Zhihao, Wang  
Beijing University of Posts and Telecommunications  
wangzhihao@bupt.edu.cn

Chuang, Zhu*  
Beijing University of Posts and Telecommunications  
czhu@bupt.edu.cn

(ICCIP) (ICCIP 2023), December 14–16, 2023, Lingshui, China. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3638884.3638890

1 INTRODUCTION

Recently, the convolutional neural networks (CNN) [1–4], have achieved excellent performance in various tasks such as classification, detection, and segmentation. Besides the superior performance, the interpretability of the model is also important in safety, fairness, and scientific research. To build a trusty AI system, more and more scholars devote themselves to the study of interpretability.

Nowadays, there are two main types of interpretable algorithms. One is to improve the transparency of the CNN by adjusting the model structure, named the transparency-interpretability method [5]. The other aims at giving a reasonable explanation for the decision of the CNN, named the post-hoc interpretability method [5].

Although some progress [6] has been made in the area of interpretability, some issues remain unsolved. Many transparency-interpretability works to improve the interpretability of the CNN by changing the model structure [7–10]. However, most of these works interpret the CNN model less intuitively [8, 10]. For post-hoc interpretability methods, many methods [10–13] do the visualization by operating features from a specific convolution layer. These methods explain the model in a linear way. Post-hoc interpretability methods work on an already trained model, therefore they can not change the fact that the CNN still lacks transparency.

To ameliorate these issues, we propose Transparent and Controllable Network Learning (TCNL), a novel approach to improve the transparency-interpretability and controllability of the CNN. In TCNL, for specific tasks, we first define concepts corresponding to human understanding. Then we guide the model to learn disentangled knowledge from predefined concepts. Finally, the model accomplishes the classification task using concept-related features. Meanwhile, TCNL is able to visualize concept information using the concept mapper. We specifically design an experiment to prove that the high-quality concept visualization stems from the successful concept learning process rather than a strong concept mapper. Also, the experiments prove that TCNL can be applied to existing CNN models [1, 2, 14]. We will release all the data in our experiments to support future scientific research on interpretability.

2 RELATED WORK

Many studies focusing on transparency-interpretability have been carried out. Some works optimize the representation learning of neurons. Zhang et al. [16] try to train filters in the high convolution layer to represent an object or a part. Based on the work in [16], Shen et al. [15] divide neurons into different groups in an unsupervised way to learn disentangled representations. However,
connections between class labels and neurons are still entangled. To deal with this issue, Liang et al. [9] align each filter in the last convolution layer with a specific class. Some approaches try to improve the interpretability through structure adjustment. Nicola et al. [8] implement a novel neural network according to the visual cortex structure to represent the part-whole hierarchies and conceptual-semantic relationships. Pietro et al. [7] explain the model with First-Order Logic with an Entropy-based Network structure.

For semantic concepts, some methods pay attention to the concept found by the model. Zhou et al. [17] find that neurons in the deep layers attempt to detect a certain pattern or concept in the input image and name these neurons as detectors. To quantitatively analyze the relationship between neurons and concepts, Zhou et al. [18] propose Network Dissection. Also, there are some methods focusing on finding important concepts for the prediction. Kim et al. [19] propose TCAV, a novel framework to evaluate the importance of pre-set concepts to the model decision. However, TCAV requires additional training using pre-set concepts. To fill this gap, Amirata et al. [20] propose the ACE algorithm to find important concepts automatically leveraging the philosophy of unsupervised methods.

The concept bottleneck model is a kind of method that build connections between human understanding and model decision. This kind of method usually works in two steps. First, it extracts features from input data and predicts the concepts, then uses predicted concepts to make a final decision. The early version of the concept bottleneck model did not use neural networks, and it has been previously used for specific applications [21, 22]. More recently, the CNN technique has been merged into the concept bottleneck model for solving specific tasks [23–26]. TCNL is different from Concept Bottleneck Models as TCNL guides the model to encode visual concepts predefined by human, while Concept Bottleneck Model only use the concept label and simply classifies images as containing or not containing certain concepts.

Many methods have been proposed to visualize the decision of the CNN, knowledge learnt by the model, or the structure of the CNN. To explain the decision of the model, Zhou et al. [27] first propose the CAM algorithm to find and visualize the important regions of the input images that support the decision of the CNN. Along with the idea of the CAM algorithm, many CAM-based methods [11, 12, 14, 28] have been proposed for better visualization and localization. For representation visualization, Dosovitskiy et al. [29] propose the Inverting Network that can invert features to images. To visualize representations for neurons, Zhou et al. [18] propose a method based on image perturbation to visualize the Receptive Field and Activation Pattern of a single neuron.

3 METHOD

Our TCNL tries to make the process of feature extraction more understandable. In our TCNL, we first define some concepts following the logic of the human decision. Then we encourage the concept feature extractor to encode information related to predefined concepts. Based on the extracted concept features, the classifier makes decisions, and the concept mapper maps concept features to concept instances for visualizing representations of the concept feature extractor.

3.1 Predefined Concepts and Datasets

To define concepts in accordance with human understanding for specific tasks, we carry out a human-intuition study. 79 people participate in our study to define concepts for mammal classification task and scene classification task. According to the study result, we select different parts and the shape of the mammal body as the key concepts for mammal classification. For scene classification, we find that different types of scenes may have totally different concepts. For example, the concept of the bed may never appear in a theater scene. Therefore, after defining concepts for scene classification, we also invite participants to select concepts that appeared in most images. Finally, we select head, torso, leg, and shape as concepts for mammal classification. For scene classification, we select bed, sofa, shelf, and seat as concepts. Examples of concept instances are shown in Figure 1. To reduce the bias, all the concepts are defined and selected by the 79 participants.

As the concepts for mammal classification are shared between each class, we also invite people to sort the concepts we select according to the importance of these concepts in human decisions. The importance score for mammal concepts is shown in Figure 2.

In TCNL, to guide the model to learn information about predefined concepts, we propose a mammal classification dataset and a scene classification dataset. With fine-grained annotation, we build concept instance sets for both datasets mentioned above. Considering most datasets [30–32] do not match our method, we collect images from existing datasets [33] and the Internet to build our own datasets and give fine-grained annotation for every image to build concept sets. Mammal classification dataset includes 5 classes, which are cat, dog, cow, horse, and panda. Scene classification dataset includes 4 types of scenes in total, which are bedroom, living room, store, and theater.

3.2 Structure and Learning

The TCNL mainly aims at guiding the CNN to learn and encode information related to predefined concepts. An overview is shown in Figure 3. In TCNL, we first define some human-intuition concepts for specific classification tasks (Figure 3 takes mammal classification task as an example). Then, images are fed into the shallow feature extractors to compute shallow features. From these shallow features, the concept extractor encodes specific concept-related information. The features output by all the concept extractors are concatenated and fed into the classifier for classification. At the same time, the concept mapper can map concept features to concept instances. The discriminator is used to classify concept instances mapped from concept features and original concept instances, which aims at improving the quality of concept learning.

3.3 Feature Extraction

TCNL guides the concept feature extractor to learn disentangled representations about predefined concepts. The feature extraction process in TCNL can be described using following formulas.

First, in equation 1), we build the concept instance set for the specific task $T$. Concept instances are used as supervision information for concept learning.

$$T \rightarrow \left\{ \left[ c_1, \ldots, c_{l-1}, c_l \right], \ldots, \left[ c_k, \ldots, c_{l-1}, c_k \right] \right\} \in C,$$  \hspace{1cm} (1)
Figure 1: This figure shows some samples of the predefined concept instances. All the instances except the shape instances are generated by pixel-wise dense annotation. The shape instances are generated using the Laplacian operator.

Figure 2: This figure shows the importance score of each concept for the human decision on the mammal classification task. The importance score is calculated based on the human-intuition study result.

Second, in equation 2), the shallow feature extractor $f_{\text{shallow}}$ computes the shallow feature $x_{\text{shallow}}$ of the input image $I$. Finally, $x_{\text{shallow}}$ is passed to the concept feature extractor to compute the concept feature $x_{c_i}$ related to concept $c_i$.

\[
x_{c_i} = f_{c_i} (x_{\text{shallow}}) = f_{c_i} (f_{\text{shallow}} (I)),
\]

where $f_{\text{shallow}}$ and $f_{c_i}$ denote the shallow feature extractor and the concept feature extractor, respectively. $x_{\text{shallow}}$ and $x_{c_i}$ represent the output features of the shallow feature extractor and the concept feature extractor, respectively.

After the feature extraction, the concept mapper maps the concept feature to the concept instance $\hat{c}_i$ for visualization and the classifier makes the final decision.

### 3.4 Concept Learning

TCNL encourages the model to encode concept-related information while keeping the outstanding performance on classification using the constraint in equation 3). $\text{Loss}_{\text{gan}}$ and $\text{Loss}_{\text{similarity}}$ aim at concept learning. $\text{Loss}_{\text{classification}}(\hat{y}, y)$ is a cross entropy loss to keep the classification performance.

\[
\text{Loss} = \lambda \text{Loss}_{\text{gan}} + \mu \text{Loss}_{\text{similarity}} + \eta \text{Loss}_{\text{classification}}(\hat{y}, y). \tag{3}
\]

Towards the goal of guiding the model to learn knowledge from predefined concepts, we use $\text{Loss}_{\text{similarity}}$ and $\text{Loss}_{\text{gan}}$ in equation 3) to constrain the learning process. For $\text{Loss}_{\text{similarity}}$, it can be described as equation 4). $\text{Loss}_{\text{similarity}}$ is calculated at the end of the concept mapper, and affects the concept feature extractor and the concept mapper. It measures the pixel-wise mean square error between the original concept instance $c_i$ and the visualized concept instance $\hat{c}_i$.

\[
\frac{1}{w \times h} \sum_{n}^{w} \sum_{m}^{h} (c_{nm} - \hat{c}_{nm})^2,
\tag{4}
\]

where $w$ and $h$ denote the width and height of the concept instance, respectively. $c_{nm}$ and $\hat{c}_{nm}$ represent the pixel at coordinates $(n, m)$ in concept instance $c_i$ and visualized concept $\hat{c}_i$.

We leverage the advantages of GAN in our TCNL to further enhance the ability of the model to learn predefined concepts. $\text{Loss}_{\text{gan}}$ in equation 3) can be described as equation 4). Consistent with the philosophy of GAN, a discriminator is used to classify the original
Figure 3: This figure shows the framework of the TCNL method.

Figure 4: This figure shows the backward propagation process in our TCNL during training.

concept instance $c_i$ and visualized concept instance $\tilde{c}_i$. Under the influence of the discriminator, the concept feature extractor and the concept mapper can have better performance in encoding and mapping concept features.

$$\text{Loss}_{\text{gan}} = E_{\tilde{c}_i} [\log D(\tilde{c}_i)] + E_{c_i, \tilde{c}_i} [\log (1 - D(c_i, \tilde{c}_i))],$$

where $D$ denotes the discriminator. Discriminator $D$ tries to maximize this function while other parts of the model with TCNL tries to minimize it.

The backward process is presented in Figure 4. $\text{Loss}_{\text{similarity}}$ is back-propagated to the concept feature extractor and the concept mapper. $\text{Loss}_{\text{gan}}$ is back-propagated to the concept feature extractor, the concept mapper, and the discriminator. $\text{Loss}_{\text{classification}}(\hat{y}, y)$ is back-propagated to the shallow feature extractor, the concept feature extractor, and the classifier. During the backward propagation, the discriminator and other parts of the model are optimized separately.

4 EXPERIMENTS

4.1 Datasets and Implementation

All the experiments are performed on mammal classification dataset and scene classification dataset. Mammal classification dataset contains 5 classes, 2500 mammal images (500 images for each class), and 10000 concept instances (2000 instances for each class). Scene classification dataset contains 4 classes, 2000 scene images (500 images for each class), and 8000 concept instances (2000 instances for each class). Some samples of the concept instances are shown in Figure 1.

As has been done in other studies [7, 16, 18, 29], we apply our TCNL to three traditional CNN models (VGG, AlexNet, ResNet). For hyper-parameters of the training process, we set the learning rate to 0.001, and the batch size to 8 for both datasets.

4.2 Metrics

For evaluating interpretability, we propose Concept-Related Neuron Proportion (CRNP is shown in Section 4.3.1), which represents the proportion of neurons that are sensitive to a certain concept. Higher CRNP means more neurons tend to encode information from a certain concept. We also use Accuracy (ACC) to measure the performance on classification tasks.

Mean Squared Error (MSE) and Structural Similarity (SSIM) as the evaluation indicators for the performance of the concept mapper. MSE measures the pixel-wise similarity between original concept instances and visualized concept instances, and SSIM comprehensively measures the differences in image brightness, contrast, and structure. For MSE metric, lower is better. For SSIM metric, higher is better.
4.3 Results

4.3.1 Concept-Related Neurons Analysis. Existing research [17, 18] shows that neurons in deep layers tend to detect high-level concepts like objects and parts in the image. When concepts are removed from the image, the activation value usually drops. Zhou et al. [17] use the numerical drop of the activation of neurons to define Detectors. Similarly, we define Concept-Related Neuron. Taking the head concept as an example, we first calculate the activation value of each neuron in the last layer of the head concept feature extractor using the full image as input. Then we remove the head part from the image and calculate the activation value again using the new image as input. Finally, calculate the average numerical drop of all the neurons in the last layer of the head concept extractor. Neurons whose activation value decreases more than the average numerical drop are defined as Concept-Related Neurons. The proportion of the Concept-Related Neurons is named as CRNP.

To analyze the transparency-interpretability of our TCNL, we calculate the proportion of concept-related neurons in the last layer of the concept feature extractor on mammal classification dataset. As the result in Figure 5 shows, our TCNL has a better performance on CRNP. On every concept in the mammal classification task, all the models with TCNL outperform traditional CNNs. For the scene classification task, VGG with TCNL and AlexNet with TCNL both outperform traditional CNNs on every concept. However, for ResNet with TCNL, it does not perform as well as the traditional ResNet on the shelf concept and the seat concept. This may be because, in addition to the concept constraints, we also use the classification constraint to ensure the discriminating ability. The classification constraint may affect the ability of the model to learn specific concepts.

4.3.2 Visualizations. To evaluate the performance of the visualization, we first train three types of models (VGG, AlexNet, ResNet) with TCNL on mammal classification dataset and scene classification dataset. Then, we collect the concept visualization results of each image on the two datasets. For each concept, we calculated MSE and SSIM to evaluate the performance of the concept mapper. The visualization result presented in Figure 6 proves that the concept mapper can successfully map concept features to concept instances based on the concept representation of the model.

4.3.3 Validating the Concept Learning. To demonstrate that the high-quality concept visualization stems from concept knowledge learned by the model, rather than a powerful concept mapper, we specifically design this contrast experiment. We applied our TCNL on two same VGG models. The first model does not have the concept-related constraint for encoding concept information while other parts of the model are the same as we have proposed in Section 3. The second model is trained with complete TCNL method. These two models are trained on our mammal classification dataset with the same training hyper-parameters (Batch size set to 8, learning rate set to 0.001). Then we evaluate the visualization performance of the concept mapper using MSE and SSIM.

As the result in Table 1 shows, the model with the concept-related constraint gives a better performance. We also present visualization result of these two models in Figure 7. It is clear that the concept constraint in TCNL helps the model better learn knowledge about predefined concepts.

5 CONCLUSION

In this paper, we propose TCNL to guide the model to learn knowledge about the predefined concepts. Therefore, transparency-interpretability of the model is improved. In our method, concepts (such as head, leg, bed, sofa and so on) that fit the logic of the human decision can be defined artificially. In TCNL, the model is divided into the shallow feature extractor, the concept feature extractor, the concept mapper, the discriminator, and the classifier. Concept instances used for concept learning can be easily accessed through artificial annotation. With the concept-related constraint in TCNL, the concept feature extractor is guided to encode information related to predefined concepts and the concept mapper is encouraged to map concept features to concept instance images. Referring to
Figure 6: This figure shows the comparison between instances visualized from the learnt concept features and the original concept instances. For each image pair, the left one is the original concept instance accessed through annotation and the right one is the output of a concept mapper.

Table 1: Evaluation of the visualization quality

| Model and Metric | Head  | Torso | Leg    | Shape  |
|------------------|-------|-------|--------|--------|
| VGG-11 without   | MSE   | 222.67| 113.50 | 95.07  | 357.95 |
| Concept Constrain| SSIM  | 0.90  | 0.91   | 0.94   | 0.74   |
| VGG-11 with      | MSE   | 54.82 | 48.03  | 26.47  | 205.06 |
| Concept Constrain| SSIM  | 0.96  | 0.95   | 0.98   | 0.77   |

Figure 7: In this figure, we compare the visualization result between the complete TCNL method and TCNL without the concept-related constraint for concept feature extractors.

We should notice that our TCNL relies on annotated data to some extent. To support related research, we annotated animal data and indoor scene data. All the data (including concept instance sets), code and following work will be released for scientific research, and all the suggestions and contribution are welcomed.

REFERENCES

[1] 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, 770–778.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems.

[3] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 2278–2324.

[4] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2818–2826.

[5] Zachary C Lipton. 2018. The mythos of model interpretability. In machine learning, the concept of interpretability is both important and slippery, 31–57.

[6] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. 2018. Explainable artificial intelligence: A survey. In 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), 0210–0215.

[7] Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Pietro Lio, Marco Gori, and Stefano Melacci. 2022. Entropy-based logic explanations of neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, 6046–6054.

[8] Nicola Garau, Nicoletta Bisagno, Zeno Sambugaro, and Nicola Conci. 2022. Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 13689–13698.

[9] Haoyu Liang, Zihao Ouyang, Yuyuan Zeng, Hang Su, Zhaoh He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. 2020. Training interpretable convolutional neural networks by differentiating class-specific filters. In European conference on computer vision, 622–638.

[10] Rui Wang, Xingqian Wang, and David Inouye. 2021. Shapley explanation networks. In International Conference on Learning Representations.

[11] Jeong Ryong Lee, Sewon Kim, Inyong Park, Taejoon Eo, and Dosik Hwang. 2021. Relevance-cam: Your model already knows where to look. In Proceedings of the IEEE conference on computer vision and pattern recognition, 14944–14953.

[12] Ramprasath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, 618–626.

[13] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardzian, and Xia Hu. 2020. Score-cam: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 24–25.
[14] K. Simonyan and A. Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 1–14.

[15] Wen Shen, Zhihua Wei, Shikun Huang, Binbin Zhang, Jiaqi Fan, Ping Zhao, and Quanshi Zhang. 2021. Interpretable compositional convolutional neural networks. In IJCAI.

[16] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpretable convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 8827–8836.

[17] Bolei Zhou, Aditya Khosla, A’gata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene cnns. In International Conference on Learning Representations.

[18] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In Proceedings of the IEEE conference on computer vision and pattern recognition, 6541–6549.

[19] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wesler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In International conference on machine learning, 2668–2677.

[20] Amirata Ghorbani, James Wesler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. Advances in neural information processing systems.

[21] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. 2009. Attribute and simile classifiers for face verification. In Proceedings of the IEEE international conference on computer vision, 365–372.

[22] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. 2009. Learning to detect unseen object classes by between class attribute transfer. In Proceedings of the IEEE conference on computer vision and pattern recognition, 951–958.

[23] Zhi Chen, Tijte Bej, and Cynthia Rudin. 2020. Concept whitening for interpretable image recognition. In Nature Machine Intelligence, 772–782.

[24] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O’Donoghue, Daniel Visentin, et al. 2018. Clinically applicable deep learning for diagnosis and referral in retinal disease. In Nature medicine, 1342–1350.

[25] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Piersson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In International Conference on Machine Learning, 5338–5348.

[26] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. Advances in neural information processing systems.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2921–2929.

[28] Mohammad AAK Jalwana, Naveed Akhtar, Mohammed Bennamoun, and Ajmal Mian. 2021. Cameras: Enhanced resolution and sanity preserving class activation mapping for image saliency. In Proceedings of the IEEE conference on computer vision and pattern recognition, 16327–16336.

[29] Alexey Dosovitskiy and Thomas Brox. 2016. Inverting visual representations with convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4829–4837.

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

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

[32] Mark Everingham, Luc Van Gool, ChristopherKI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. In International journal of computer vision, 303–338.

[33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva. 2016. Places: An image database for deep scene understanding. arXiv preprint arXiv:1610.02055.