TCNL: TRANSPARENT AND CONTROLLABLE NETWORK LEARNING VIA EMBEDDING HUMAN-GUIDED CONCEPTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Explaining deep learning models is of vital importance for understanding artificial intelligence systems, improving safety, and evaluating fairness. To better understand and control the CNN model, many methods for transparency-interpretability have been proposed. However, most of these works are less intuitive for human understanding and have insufficient human control over the CNN model. We propose a novel method, Transparent and Controllable Network Learning (TCNL), to overcome such challenges. Towards the goal of improving transparency-interpretability, in TCNL, we define some concepts for specific classification tasks through scientific human-intuition study and incorporate concept information into the CNN model. In TCNL, the shallow feature extractor gets preliminary features first. Then several concept feature extractors are built right after the shallow feature extractor to learn high-dimensional concept representations. The concept feature extractor is encouraged to encode information related to the predefined concepts. We also build the concept mapper to visualize features extracted by the concept extractor in a human-intuitive way. TCNL provides a generalizable approach to transparency-interpretability. Researchers can define concepts corresponding to certain classification tasks and encourage the model to encode specific concept information, which to a certain extent improves transparency-interpretability and the controllability of the CNN model. The datasets (with concept sets) for our experiments will also be released.

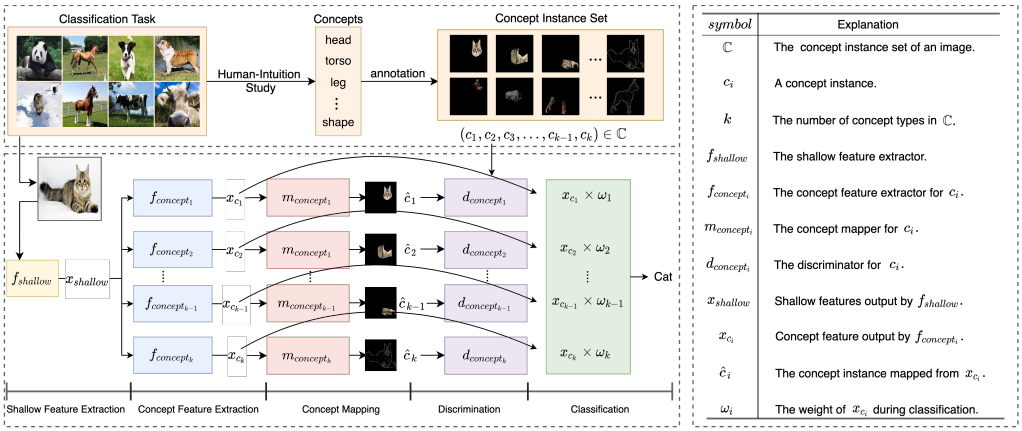
1 INTRODUCTION

1.1 BACKGROUND

Recently, the convolutional neural networks (CNN) (LeCun et al., 1998; Krizhevsky et al., 2012; He et al., 2016; Szegedy et al., 2016), have achieved excellent performance in various computer vision tasks such as image classification, object detection, and semantic segmentation. Besides the superior performance, the interpretability of the model plays a critical role in safety, fairness, and scientific research. Towards the goal of building the trusty artificial intelligence system, more and more scholars devote themselves to the study of the interpretability of the CNN.

Nowadays, there are two main types of algorithms. One is designed to improve the transparency of the CNN by adjusting the structure of the model, named as the transparency-interpretability method (Lipton, 2018). The other aims at giving a reasonable explanation for the decision of the CNN, named as the post-hoc interpretability method (Lipton, 2018).

Although some progress (Došilović et al., 2018) has been made in the area of the interpretability of the CNN, some issues remain unsolved. Many transparency-interpretability works focus on improving the interpretability of the CNN by changing the structure of the model (Barbiero et al., 2022; Garau et al., 2022; Liang et al., 2020; Wang et al., 2021). However, most of these works interpret the CNN model in a way that is less intuitive for human understanding (Garau et al., 2022; Wang et al., 2021). For post-hoc interpretability methods that give visual explanations of the CNN, many methods (Zhou et al., 2016a; Selvaraju et al., 2017; Wang et al., 2020; Lee et al., 2021) do the visualization by operating feature maps from a specific convolution layer. These methods try to explain



symbol	Explanation
$\mathbb{C}$	The concept instance set of an image.
c_i	A concept instance.
k	The number of concept types in $\mathbb{C}$.
$f_{shallow}$	The shallow feature extractor.
$f_{concept_i}$	The concept feature extractor for c_i.
$m_{concept_i}$	The concept mapper for c_i.
$d_{concept_i}$	The discriminator for c_i.
$x_{shallow}$	Shallow features output by $f_{shallow}$.
x_{c_i}	Concept feature output by $f_{concept_i}$.
$\hat{c}_i$	The concept instance mapped from x_{c_i}.
ω_i	The weight of x_{c_i} during classification.

Figure 1: In TCNL, we first define some human-intuition concepts for specific classification tasks (this figure 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.

decisions of 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 interpretability.

1.2 Our work

In order to ameliorate the issues mentioned above, we propose Transparent and Controllable Network Learning (TCNL), a novel approach to improve the transparency-interpretability and controllability of the CNN model. 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 features related to predefined concepts.

TCNL can be applied to several existing models such as VGG, ResNet, and AlexNet (Simonyan & Zisserman, 2015; Krizhevsky et al., 2012; He et al., 2016). TCNL improves interpretability and controllability by defining and learning concepts in accordance with human understanding for specific tasks. Meanwhile, TCNL is able to visualize concept information extracted by the model through 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. We will release all the concept instance sets and datasets in our experiments to support future scientific research on transparency-interpretability.

2 Related Work

Interpretable Models. Many studies focusing on transparency-interpretability have been carried out. Some works try to optimize the representation learning of neurons. Zhang et al. (2018) try to train each filter in the high convolution layer to represent an object or a part. Based on Zhang et al. (2018), Shen et al. (2021) 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. (2020) try to align each filter in the last convolution layer with a specific class during the learning process. Some approaches also try to improve the interpretability through structure adjustment. Garau et al. (2022) implement a neural network with a novel structure according to the visual cortex structure to represent the part-whole hierarchies

and conceptual-semantic relationships. Barbiero et al. (2022) propose an Entropy-based Network structure, trying to explain the model with First-Order Logic.

Semantic Concepts. For semantic concepts, some methods pay attention to the concept found by the model during feature extraction. Zhou et al. (2015) find that neurons in the deep layers attempt to detect a certain pattern or concept in the input image and they name these neurons as detectors. To quantitatively analyze the relationship between neurons and concepts, Bau et al. (2017) propose Network Dissection. Also, there are some methods focusing on finding important concepts for the prediction of the model. Kim et al. (2018) propose TCAV, a novel framework to evaluate the importance of pre-set concepts to the decision of the model. However, TCAV requires additional training using pre-set concepts. To fill this gap, Ghorbani et al. (2019) propose the ACE algorithm to find important concepts for the decision of the model automatically leveraging the philosophy of unsupervised methods.

Visualization. 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. (2016a) 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 (Selvaraju et al., 2017; Wang et al., 2020; Lee et al., 2021; Jalwana et al., 2021) have been proposed for better visualization and localization. For representation visualization, Dosovitskiy & Brox (2016) propose the Inverting Network that can invert features to images. To visualize representations for neurons, Bau et al. (2017) 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

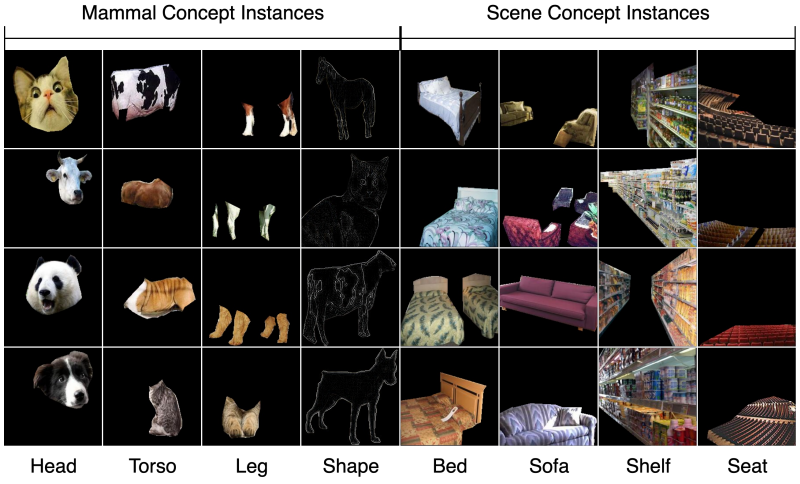


Figure 2: This figure shows some samples of the predefined concept instances. All the instances except the shape instances are generated by pixel-wise dense annotation. The shape instances are generated using the Laplacian operator.

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 stage may never appear in a bedroom scene. Therefore we select some common concepts that appear in images from each class of scenes. 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 2. As the concepts for mammal classification are shared between each class, we 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 3.

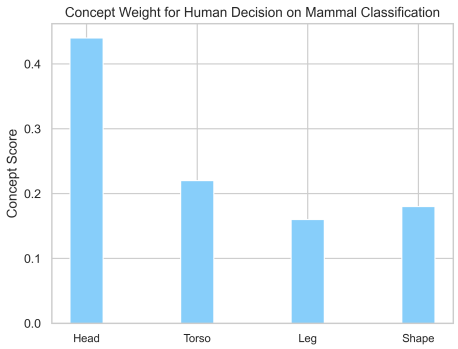


Figure 3: This figure shows the importance score of each concept for the human decision on the mammal classification task. The importance score is calculated based on the human-intuition study result.

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 (Deng et al., 2009; Lin et al., 2014; Everingham et al., 2010) do not match our method, we collect images from existing datasets (Zhou et al., 2016b) 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 1.

Structure of TCNL. In contrast to the traditional CNN structure, TCNL divides the CNN model into the shallow feature extractor, the concept feature extractor, the concept mapper, and the discriminator. Different parts of the model have different structures[1] and perform different functions. The shallow feature extractor consists of some shallow convolutional layers and pooling layers, aiming at learning some simple features such as texture and color (Bau et al., 2017). The concept feature extractor contains some deep convolutional layers and it is encouraged to learn information about the predefined concepts from simple features. The concept mapper consists of some transposed convoluational layers. Based on concept features output by the concept feature extractor, the concept mapper maps features to images for visualizing concept representation learnt by the model. In addition, we use the discriminator to improve the performance of the model on concept learning.

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, and it is also shown in Figure 1.

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.

$$T \rightarrow \{(c_1^1, c_1^2, \ldots, c_1^{l-1}, c_1^l), \ldots, (c_k^1, c_k^2, \ldots, c_k^{l-1}, c_k^l)\} \in \mathbb{C}, \tag{1}$$

where c_i^j denotes a certain concept instance from concept c_i and $\mathbb{C}$ denotes a concept instance set including $k \times l$ instances (k denotes the number of the concepts and l denotes the number of instances from a certain concept). For the clear expression of the formulas, we use c_i to denote an instance from a specific concept in the following content.

[1]The specific structure of each part can be adjusted according to the CNN model applied by TCNL

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

$$x_{c_i} = f_{c_i}(x_{shallow}) = f_{c_i}(f_{shallow}(I)), \tag{2}$$

where $f_{shallow}$ and f_{c_i} denote the shallow feature extractor and the concept feature extractor, respectively. $x_{shallow}$ 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.

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

$$Loss = \lambda Loss_{gan} + \mu Loss_{similarity} + \eta Loss_{classification}(\hat{y}, y). \tag{3}$$

Towards the goal of guiding the model to learn knowledge from predefined concepts, we use $Loss_{similarity}$ and $Loss_{gan}$ in Equation 3 to constrain the learning process. For $Loss_{similarity}$, it can be described as Equation 4. $Loss_{similarity}$ 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 $\hat{c}_i$. With this constraint, the concept feature extractor is guided to encode concept-related information and the concept mapper is encouraged to map concept features to concept instances.

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

where w and h denote the width and height of the concept instance, respectively. c_{nm} and $\hat{c}_{nm}$ represents 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. $Loss_{gan}$ in Equation 3 can be described as Equation 5. Consistent with the philosophy of GAN, a discriminator is used to classify the original concept instance c_i and visualized concept instance $\hat{c}_i$. Under the influence of the discriminator, the concept feature extractor and the concept mapper can have better performance in encoding and mapping concept features.

$$Loss_{gan} = \mathbb{E}_{\hat{c}_i}[logD(\hat{c}_i)] + \mathbb{E}_{c_i,\hat{c}_i}[log(1 - D(c_i, \hat{c}_i))], \tag{5}$$

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

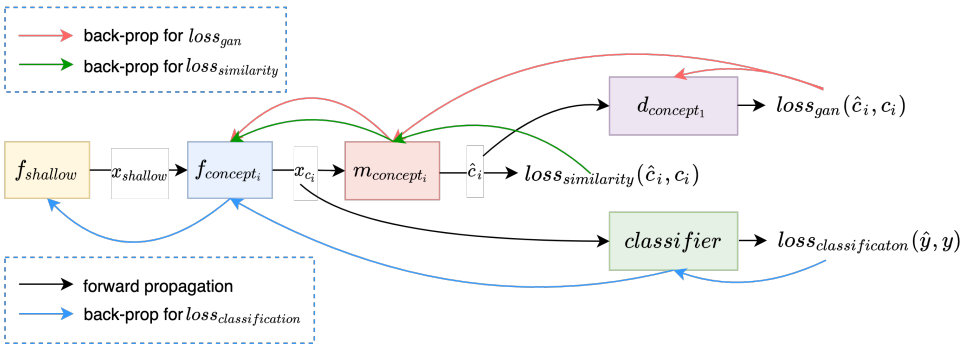


Figure 4: This figure shows the backward propagation process in our TCNL during training.

The backward process is presented in Figure 4. $Loss_{similarity}$ is back-propagated to the concept feature extractor and the concept mapper. $Loss_{gan}$ is back-propagated to the concept feature extractor, the concept mapper, and the discriminator. $Loss_{classification}(\hat{y}, y)$ is back-propagated 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.

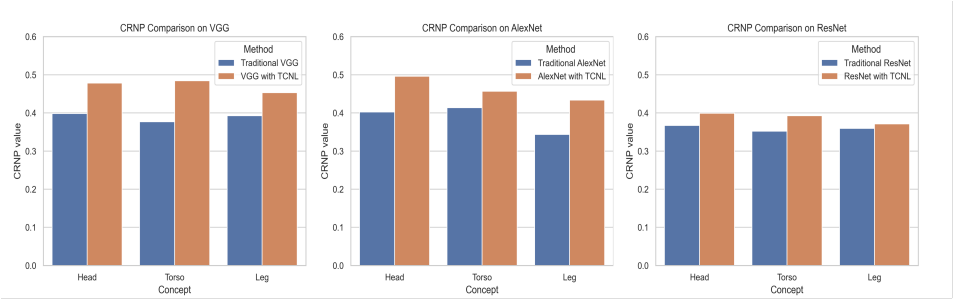


Figure 5: This figure shows the CRNP result on each concept in mammal classification dataset. We compare the result between traditional CNNs (VGG, ResNet, and AlexNet) and models with TCNL.

4 EXPERIMENTS

4.1 DATASETS AND IMPLEMENTATION

Datasets: 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 2.

Implementation. As has been done in other studies (Barbiero et al., 2022; Zhang et al., 2018; Bau et al., 2017; Dosovitskiy & Brox, 2016), 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[2]), 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 use Mean Squared Error (MSE) and Structural Similarity (SSIM) (Wang et al., 2004) 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. We also use Accuracy (ACC) to measure the performance on classification tasks.

4.3 RESULTS

4.3.1 CONCEPT-RELATED NEURONS ANALYSIS

Existing research (Zhou et al., 2015; Bau et al., 2017) 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. Based on the research of (Zhou et al., 2015; Bau et al., 2017), 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.

[2]The definition of CRNP is shown in subsubsection 4.3.1

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. Models with TCNL outperform traditional CNNs on every concept in mammal classification dataset

4.3.2 CONCEPT WEIGHT ANALYSIS

To further analyze the transparency-interpretability of TCNL, we quantitatively measure the importance of the concepts during the decision process of the model on mammal classfication datasets. To calculate weight for these concepts, we use gradient back-propagated to the corresponding concept feature as the concept weight, which is in accordance with the weight calculating method in Selvaraju et al. (2017). The gradient value measures the sensitivity of a concept feature to the decision of the model. A higher gradient value represents a higher weight in classification.

According to the result in Table 1, for human decision, the importance rank of concepts is head, torso, shape, and leg. However, in our TCNL, four types of concepts (head, torso, leg, and shape) have similar weights and the shape concept gets the biggest weight in the decision of the model. To control the weight of concepts in the decision of the model will be our future work.

Table 1: Concept weight on mammal classification task

Subject of decision-making	Head	Torso	Leg	Shape
VGG with TCNL	0.22	0.22	0.23	0.33
Human	0.44	0.22	0.16	0.18

4.3.3 VISUALIZATION ANALYSIS

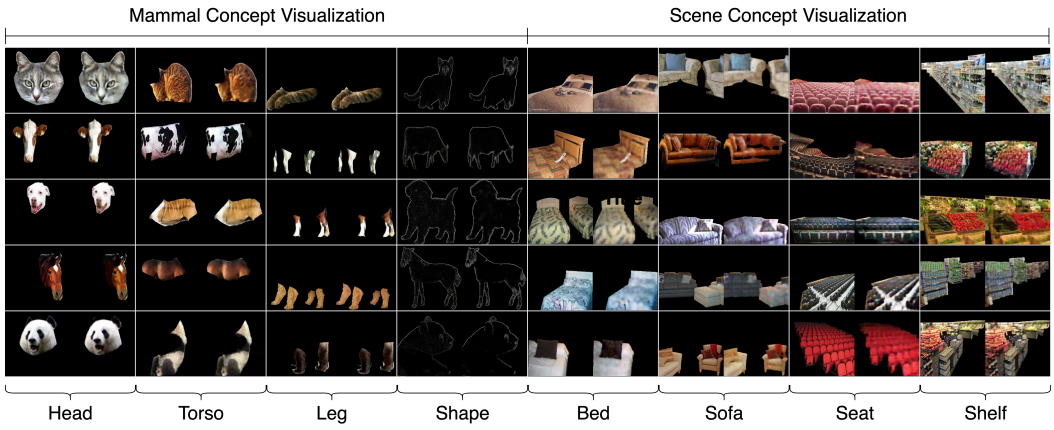


Figure 6: This figure shows the comparison between instances visualized from the concept features and the original concept instances. For each image pair, the left one is the original concept instance accessed through annotation and the right one is the output of a concept mapper.

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 results presented in Table 2 and Figure 6 prove that the concept mapper can successfully map concept features to concept instances based on the concept representation of the model.

We also concatenate concept instances visualized by the concept mapper to analyze the positional association of the concept instances. The result in Figure 7 shows that TCNL is also able to help the model learn position information among concept instances.

Table 2: Evaluation of the visualization quality on each concept using MSE and SSIM

Model and Metric		Head	Torso	Leg	Shape	Bed	Sofa	Shelf	Seat
ResNet-50 with TCNL	MSE	44.44	36.52	31.34	276.49	93.38	50.61	136.75	64.44
	SSIM	0.96	0.96	0.98	0.74	0.95	0.94	0.92	0.95
VGG-11 with TCNL	MSE	54.82	48.03	26.47	205.06	44.79	35.97	105.58	69.28
	SSIM	0.96	0.95	0.98	0.77	0.96	0.97	0.94	0.94
AlexNet with TCNL	MSE	98.98	111.56	115.04	505.09	126.26	192.80	152.45	230.87
	SSIM	0.92	0.92	0.93	0.69	0.96	0.93	0.92	0.92

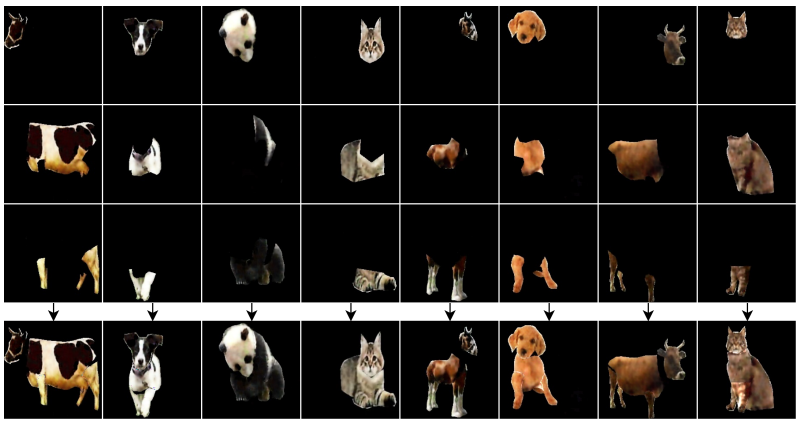


Figure 7: This figure shows the positional association of the concept instances. Images above the arrow are concept instances visualized by the concept mapper. Images below the arrow are concatenated from individual concept instances.

4.3.4 VALIDATING THE CONCEPT LEARNING

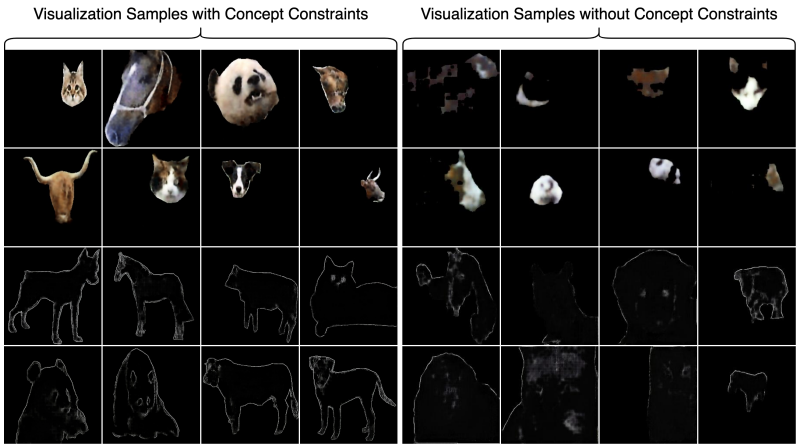


Figure 8: In this figure, we compare the visualization result between the complete TCNL method and TCNL without the concept-related constraint for concept feature extractors.

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.

Table 3: Visualization comparison between the complete TCNL method and TCNL without concept-related constraint.

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

As the result in Table 3 shows, the model with the concept-related constraint gives a better preformance. We also present visualization result of these two models in Figure 8. It is clear that the concept constrain in TCNL helps the model better learn knowledge about predefined concepts.

4.3.5 CLASSIFICATION PERFORMANCE

In addition, we also evaluate the classification performance of the model with TCNL. We train three types of CNN models (VGG, ResNet, AlexNet) with TCNL on mammal classification dataset and scene classification dataset. At the same time, we also train traditional VGG, ResNet, and AlexNet, which are used as baseline methods. ACC is used to measure classification performance. The result in Table 4 shows the model with TCNL maintains a strong discriminating ability as the traditional CNN.

Table 4: Classification performance comparison between models with TCNL and traditional CNNs on mammal classification dataset and scene classification dataset using ACC.

Dataset and Method		VGG-11	ResNet-50	AlexNet
Mammal	Original Method	0.68	0.70	0.76
	TCNL Method	**0.82**	**0.74**	**0.82**
Scene	Original Method	**0.82**	**0.80**	0.78
	TCNL Method	0.80	0.75	**0.83**

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 the successful utility of our method, we expect that TCNL has the potential to help people understand and gain more control on the CNN in more areas than the classification task.

REFERENCES

Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Pietro Lió, Marco Gori, and Stefano Melacci. Entropy-based logic explanations of neural networks. In *AAAI*, volume 36, pp. 6046–6054, 2022.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, pp. 6541–6549, 2017.

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

Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pp. 0210–0215, 2018.

Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *CVPR*, pp. 4829–4837, 2016.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Nicola Garau, Niccolò Bisagno, Zeno Sambugaro, and Nicola Conci. Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. In *CVPR*, pp. 13689–13698, 2022.

Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *NeurIPS*, 32, 2019.

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

Mohammad AAK Jalwana, Naveed Akhtar, Mohammed Bennamoun, and Ajmal Mian. Cameras: Enhanced resolution and sanity preserving class activation mapping for image saliency. In *CVPR*, pp. 16327–16336, 2021.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jeong Ryong Lee, Sewon Kim, Inyong Park, Taejoon Eo, and Dosik Hwang. Relevance-cam: Your model already knows where to look. In *CVPR*, pp. 14944–14953, 2021.

Haoyu Liang, Zhihao Ouyang, Yuyuan Zeng, Hang Su, Zihao He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. Training interpretable convolutional neural networks by differentiating class-specific filters. In *ECCV*, pp. 622–638. Springer, 2020.

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

Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pp. 618–626, 2017.

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

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, pp. 1–14, 2015.

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

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *CVPR workshops*, pp. 24–25, 2020.

Rui Wang, Xiaoqian Wang, and David Inouye. Shapley explanation networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *CVPR*, pp. 8827–8836, 2018.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pp. 2921–2929, 2016a.

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