Classification Work on Architecture Image with Hierarchy

Chao Fu
Tufts University
161 College Avenue, Medford
MA, United States
chao.fu@tufts.edu

Yuanye Chi
Tufts University
161 College Avenue, Medford
MA, United States
yuanye.chi@tufts.edu

Yingjie Jiang
Tufts University
161 College Avenue, Medford
MA, United States
yingjie.jiang@tufts.edu

Zhiyi Zhao
Tufts University
161 College Avenue, Medford
Oregon, United States
zhiyi.zhao@tufts.edu

Abstract

CNN models have become the state-of-the-art model in image tasks. It is hard for simple CNN models to achieve better results in complex image classification tasks. In architectural style classification, the model needs to learn and utilize features of different granularity. In this task, we use the HD-CNN model, which introduces the hierarchy structure, using coarse category components and fine category components to learn the features of different levels. We further discuss the accuracy and the interpret-ability of the HD-CNN results and infer that by continuously optimizing the model, HD-CNN can achieve better results in the architectural style classification task.

1 Introduction

Because of the excellent ability to learn and capture features, more and more image tasks choose CNNs as the basis in image tasks these decades, especially image classification tasks. However, the original CNN model has emerged as a limitation along with the changing objectives and increasing task complexity.

To develop methods for such applications, many researchers have changed the existing structure of the CNN models, hoping to improve the learning ability [4, 2, 5, 10, 15, 21, 14]. Google has built GoogleNet to learn more detailed knowledge by deepening the model[16]. However, optimizing a single CNN structure is not sufficient for learning complex tasks. On the contrary, the over-complex CNN cell structure has many weaknesses, such as larger parameter size, larger space, and longer learning time. In response to such a situation, some other researchers started to work on combining CNN models with other existing models to combine the advantages of each model[17, 22, 12]. Among the many combined models, some of them introduce the idea of hierarchy, which uses different CNN units for feature learning with different coarse and fine granularity[11, 23, 3]. The classifier can combine the features from different levels and use them together to improve the model's classification accuracy.

In this paper, we hope to build a model to classify 25 different architectural styles accurately for the architectural style classification task. Architectural images have feature layering and complex features. We need to consider the outline features of buildings and the detail features inside the outline. Firstly,

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

we used a simple CNN model for classification and obtained an accuracy of 37.9. Figure 6 shows that the simple CNN cannot classify all the categories better, and some of the categories appear to overlap without further distinction. To improve the classification task further, we introduced hierarchy theory and used HD-CNN to learn the classification of features at different levels to improve classification accuracy from 37.9 to 40.6.

We organize this paper as follows. In Section 2, we first review some studies about CNN optimization and CNN-based model in classification work. Then, we introduce theories and models used in this paper in Section 3. Next, we discuss the model design and report results in Section 4. We further discuss some existing problems in Section 5. Finally, we summarize our future work in Section 6.

2 Related Work

CNN based models has become the state-of-the-art model and has successful contribution in computer vision tasks including image classification, image recognition, object detection. In reality, it has been applied to diversity application in image and video data.

In the past decade, researchers have proposed numerous techniques for image classification using CNNs. These techniques can be divided into two categories based on their modified target. One category is to modify and optimize the component of CNN to improve the performance of the model; another is to redesign a CNN-based classifier.

Each part of the CNN architecture plays an essential role for the CNN to perform the learning of image features and use them for prediction. Researchers optimize the overall performance by overcoming the existing weakness in the CNN structure. After Glorot et al. [4] proposed *ReLU* to solve the gradient descent problem, the Clevert et al. [2] performance *ELUs* as the new active function. Goodfellow et al. [5] modifies the dropout layer and implements the maxout. Lee et al. [10], Springenberg et al. [15] and Zeiler and Fergus [21] enhance CNN performance from implementing new pooling methods. In addition to optimizing the internal component, some researchers focus on the overall depth and size of the CNN. The depth of the network helps the neural units to learn the features continuously. Simonyan and Zisserman [14] constructs convolution filters with the unit size of 3(3*3) and increases the depth of the network to achieve better performance. Similar networks include *GoogleNet* [16], *AlexNet* [8].

The improvement of the individual components can help a CNN unit achieve better results while redesigning the CNN-based model becomes another optimization direction for image classifiers in image classification tasks. A single CNN unit can use its structure to capture and learn features, including outline, color, and luminance. However, there are a variety of complex scenes in real life. A single CNN is insufficient to support the implementation of the task. Therefore, researchers implemented different CNN-based classifiers by combining CNN units with other models in different image classification tasks. Since CNN gets great success in single-label classification work, Wang et al. [17] combines CNN and RNN to solve multi-label classification work. Zhang et al. [22] implements the *CV-CNN* by extending the domain of input to the complex dimension to get better results. Sermanet et al. [12] accumulate the predicted bounding boxes and used for simultaneous prediction. It helps avoid learning background knowledge, reduce learning cost, and improves accuracy. In order to learn the features at different levels, some researchers introduced the idea of hierarchy into CNN-based models [11], [23], [3]. Generally speaking, whether improving the CNN design or combining other models with CNNs, the CNN-based image classifiers attempt to capture and learn the images' features and achieve higher accuracy.

3 Model

In image classification work, a single CNN can change its ability to capture features by changing its components, including activation functions, pooling methods, number of convolution kernels, network depth, etc. However, if the classifier requires to learn features at different levels, a single CNN is not sufficient in feature extraction. Therefore, combining hierarchical theory and CNN models allows us to use hierarchical features to learn coarse and fine levels in images separately and employ them for predictive classification to achieve better prediction.

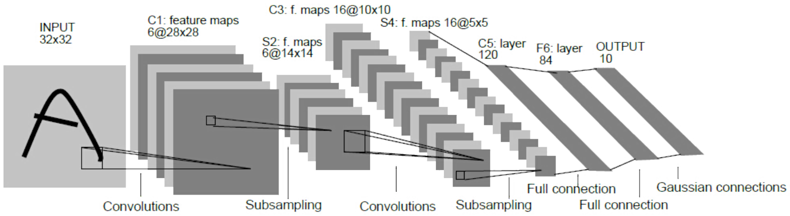

Figure 1: The basic CNN model.[9]

This section will first describe the single CNN model in Section 3.1. Then we will introduce the concept of hierarchy theory in Section 3.2. The CNN model combining the hierarchy theory will be discussed in Section 3.3.

3.1 Convolution Neural Network

The prototype of CNNs, LeNet-5 [9], network was born in 1998. With the continuous improvement of computing power, large CNN networks have shown excellent performance in image tasks and have become the dominant model in image tasks.

The CNN model structure comprises five layers: the input layer, convolution layer, pooling layer, fully connected layer, and final output layer. Figure 1 represents the basic model structure.

Input Layer The input layer is responsible for processing the image into pixel values as input. The input will be fed into the corresponding channels according to the RGB color. The image will be input to a single channel for grayscale image tasks. Each channel uses a pixel range, [0, 255], to represent the color.

Convolution Layer The convolution layer acts as a ’filter’ that can set the convolution values to perform different missions, like no change to the original image, edge detection, sharpening process, Gaussian blur, etc. In addition, the settings of stride and padding parameters in the convolution layer can determine the distance of features and maintain the image shape after convolution.

Pooling Layer The pooling layer uses pooling functions to reduce the output size and the number of parameters to improve computational efficiency without changing the feature representation.

Fully Connected Layer The fully connected layer consolidates the output from the pooling layer into the final output. With the fully-connected layer, the final output will include necessary features. At the same time, the influence of feature position on the final learning can be ignored.

3.2 Hierarchy Structure

Hierarchy Theory is not a theory limited to machine learning tasks, it appears widely in social systems, biological structures, and biological taxonomy. In systems theory, the up-down relationships in hierarchy enable the establishment of bounded but asymmetric relationships. The criteria of the hierarchy can be customized, and each hierarchy is populated by entities whose characteristics are the properties of the hierarchy [1].

3.3 HD-CNN

In image classification tasks, the basic CNN model is better at capturing the distinct features in the dataset and classifying them. For instance, a CNN can easily distinguish a cat from a person because their outline features are entirely different. However, it is challenging to distinct between Maine Coon and Norwegian Forest Cat. Like the indistinguishable cat breeds, many real-life tasks need to capture coarse and fine features in one image and combine them for prediction work. Therefore, we discuss HDCNN [20], which combines the hierarchy theory and CNN structure. HD-CNN architecture is represented in Figure 2.

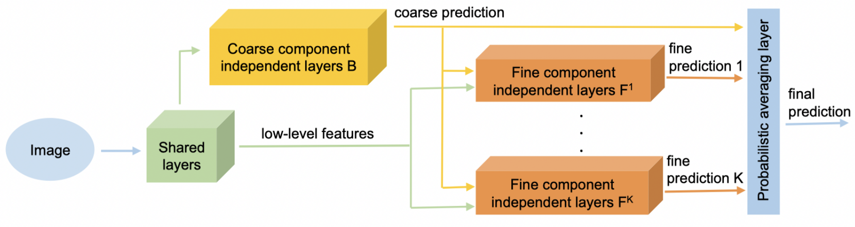
coarse prediction
Coarse component independent layers B
Fine component independent layers F¹
fine prediction 1
Image
Shared layers
low-level features
Fine component independent layers Fᴷ
fine prediction K
Probabilistic averaging layer
final prediction

Figure 2: HD-CNN architecture[20]

Hierarchical Deep Convolutional Neural Network (HD-CNN) simulates the hierarchy by combining different CNN units. The model combines hierarchy's strength to exploit the common feature sharing aspect of images. In the model, coarse-grained features and fine-grained features are trained separately, and outputs are combined in the probabilistic average layer to produce the final prediction results.

The model consists of four main bodies: shared layers, a single coarse category component, multiple fine category components and a single probabilistic averaging layer. The configuration of the shared layer is set to be the same as the previous layers in the building block CNN. Each category component has the same design as CNNs. The coarse and fine category components will learn their features at their respective levels. The coarse category component enables the model to learn the upper level features and perform coarse classification. In this process, the model does not learn the fine-grained features and will not be affected. Instead, a fine category component will perform lower-level feature capture for a coarse category component under a coarse category component. At this point, the fine category components already have the knowledge from the coarse classifier, so they only need to focus on the fine-grained features under the same coarse category. Eventually, the probabilistic averaging layer receives the output from both the coarse category component as well as the fine category component. It uses the weighted average for the final classification probability.

$$p(x_i) = \frac{\sum_{k \in K} B_{ik} p_k(x_i)}{\sum_{k \in K} B_{ik}} \tag{1}$$

where B_{ik} is the probability of image x_i in coarse catefgory k predicted by coarse category component B. $p_k(X_i)$ is the probability predicted by the fine category component F_k.

Pretraining Since our overall input is unchanged, when we combine fine category components with coarse category components, the number of parameters increases linearly with coarse category, which subsequently causes increasing complexity and overfitting after training. Second, training many fine category components increases the memory footprint and slows down the training. To address these two problems, we do not train the CNN as a whole but split to the training and pretraining.

In pretraining work, we train the coarse category component using the coarse category segmented dataset. For the fine category component, we only use datasets belonging to the same coarse category for training. We use its pretrain parameter values for the fine category component configuration to initialize the model except for the final convolutional layer.

4 Experiments

Our goal is to implement the classification of architectural style dataset in this paper. We use CNN and HD-CNN to accomplish this classification task respectively. The effectiveness of the hierarchical structure introduced is demonstrated by comparing CNN and HD-CNN.

In Section 4.1, we will introduce our dataset, Architectural Style, and discuss our task preparation. The configuration of the model is presented in Section 4.2. Then, we describe the classification results of CNN and HD-CNN in Section 4.3 and Section 4.4.

4.1 Dataset: Architectural Styles

To ensure the varied architectural style and sufficient training samples, we chose the Architectural style dataset used in Xu et al. [19]. The dataset consists of 10,113 images of buildings in a wide

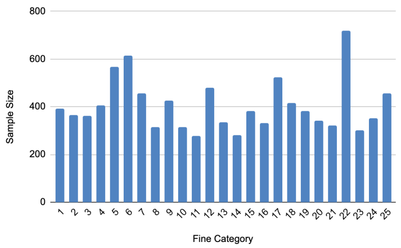

Figure 3: Architectural Dataset

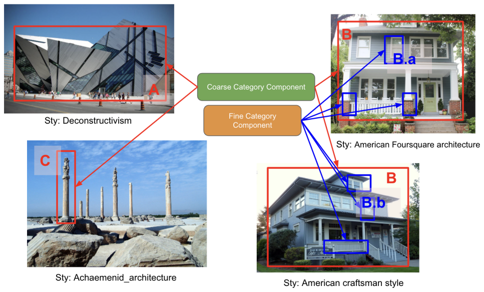
A
Sty: Deconstructivism
Coarse Category Component
Fine Category Component
B
B.a
Sty: American Foursquare architecture
C
Sty: Achaemenid_architecture
B
B.b
Sty: American craftsman style

Figure 4: The Overview of Architectural Image

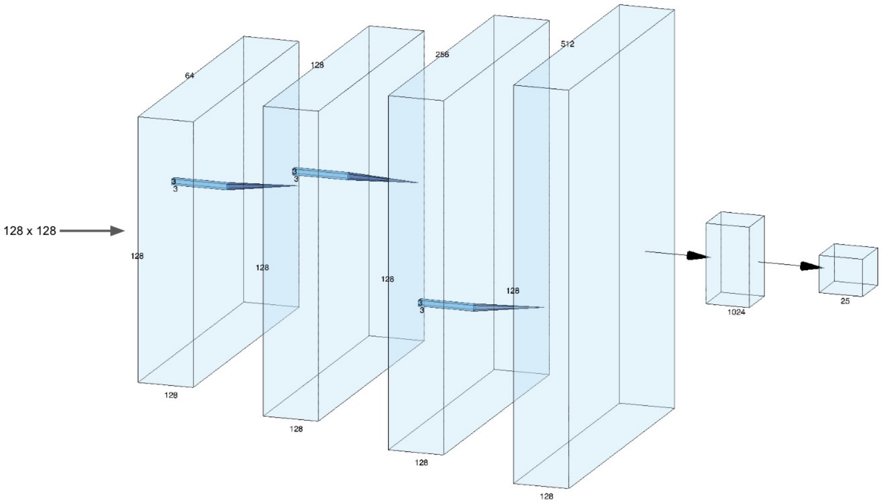

Figure 5: CNN unit architecture

range of styles, covering 25 architectural styles. The details of the dataset is shown in Table I. We can find that among the 25 architectural styles, some of them are difficult to distinguish from each other by their appearance (Figure 4). For these architectural styles, image features are not limited to their shape but also the architectural details. Thus, image classification of architectural styles needs a model to capture overall features, e.g., house shape, and detailed features, e.g., window styles, at different levels.

Unlike the original HD-CNN paper, we customize coarse categories for 25 classes of architectural styles to change the features learned by the coarse category component.

4.2 Training

As a basic unit of model composition in HD-CNN, the design within the CNN unit is also related to the overall learning ability of the classifier. The internal design of the model is shown in Figure 5. HD-CNNs share the same CNN units, except that the last layer of the coarse category component uses a full connected layer of size 11.

In the CNN design, we use padding to keep the result of each convolution having the same size as the input. The model finally uses Adam [6] as the optimization algorithm to ensure smooth parameter iterations.

4.3 Whole Test in CNN

Without considering the image feature hierarchy, we used a single CNN model to predict 25 classes of building categories, and the model's prediction accuracy ended up being 37.9. By graphing the prediction results, we obtain Figure 6.

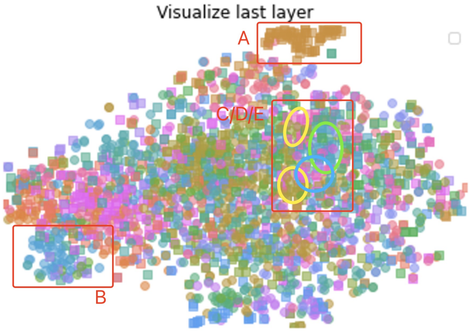

Figure 6: CNN classification result

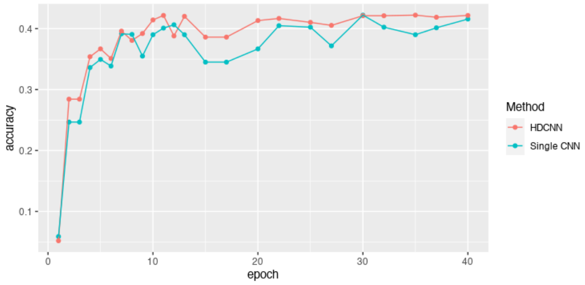

Figure 7: Accuracy between CNN and HD-CNN

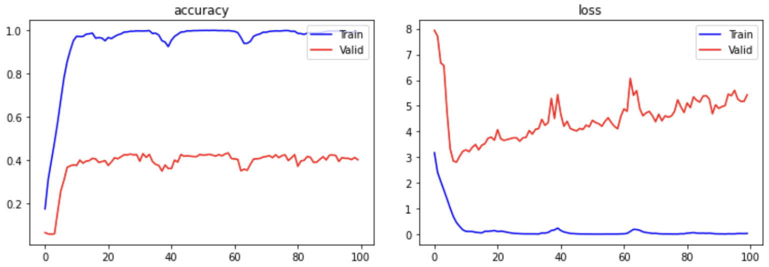

Figure 8: HD-CNN classification result

Figure 6 shows the distribution after prediction using CNN. We learn that the CNN can correctly classify some fine categories, such as A and B regions. However, for some fine categories, including regions C, D, and E in the figure, the overlap is formed on the figure, indicating that the model cannot classify them further after separating them from other categories. To further learn the overlapping classifications, we introduced hierarchy and built the HD-CNN model.

4.4 Whole Test in HD-CNN

By comparing the types of the overlapping areas with the architectural styles, we discovered that the building styles in the overlapping areas are mostly continuously distributed in the historical timeline. Thus, we divide 25 categories according to the popularity time and construct HD-CNN with 11 coarse categories.

The Classification Result Figure 7 represents that the overall accuracy of HD-CNN classification is higher than CNN.

In HD-CNN, the accuracy score improved from 37.9 to 40.6 with parameter decrease and time decrease, reflecting the effectiveness of the hierarchy. Figure 8 shows the accuracy and loss of HD-CNN. From the acc plot, we find that the accuracy of the training set becomes flat after improving to the highest, while the accuracy of the test set cannot rise further after reaching about 0.4. In the loss plot, the training set shows a gradually decreasing trend. However, the test set shows a weak increasing trend after a substantial decrease. The above performance we will discuss further in the next section.

Different Coarse Value Figure 9 shows the coarse class sample size, coarse class fine size, and coarse class loss distribution. The distribution of the three classes tends to be consistent. With the known imbalanced sample size knowledge, it shows that the number of classifications and sample size under coarse category work together on the model loss.

The variation of loss shown in Figure 10 demonstrates the effect of the number of coarse categories in classification work. The figure shows that more coarse categories are not better, so when using

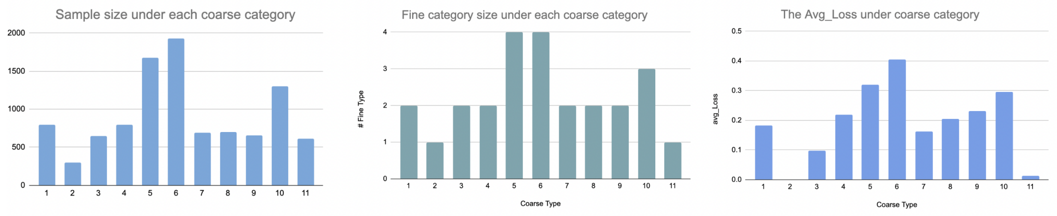

Figure 9: Loss Impact Factors

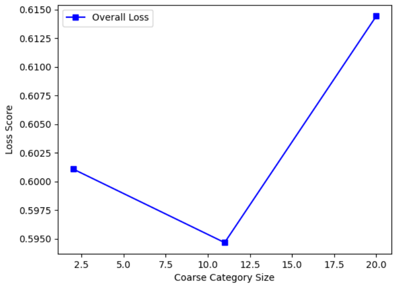

Figure 10: Loss Score with Different Coarse Size

HD-CNN for classification tasks, the number of coarse categories needs to be carefully chosen in conjunction with the characteristics of the classification task.

5 Discussion

From the results in Section 4, we can see that the introduction of hierarchy affects CNN learning and combines different levels of features applied to the classification. However, the accuracy of the prediction has not been improved substantially. We analyze this result in the context of the whole task.

Over-training and Dropout In the previous section, we discussed that the validation set in HD-CNN shows a weak upward trend, which is contrary to the network learning goal. Combined with our model settings, we speculate two causes: the Over-training and drop-out settings.

Since the training time of the network is a crucial factor affecting the result, too short or too long training time will hurt the prediction effect of the model. In our model, the model undergoes over-training, which makes the model not stop after causing over-fitting, resulting in increasing losses later on. Therefore, we consider adding 'Early Stopping' so that the model stops itself when performance on a validation dataset degrades to avoid over-training.

Referring to the original HD-CNN paper, we believe that the drop-out design is another reason for the rising trend of loss. In the original paper, the model has the drop-out design after each convolution layer, and the drop-out gets larger with the depth of the model. Such a design, which is not available in our drop-out layer, is presumed to cause the increasing loss.

Coarse Category In the experiment, we manually categorized the 25 architectural style types into customized coarse categories by building period and used them as input for the coarse category component. Therefore, the manual definition of coarse category becomes one possible reason that makes hierarchy less practical.

In HD-CNN learning, each component learns the common features of the input classes. The coarse category component defaults all fine categories under the same coarse category to share similar upper-level features. After that, a fine category component learns the difference between lower-level

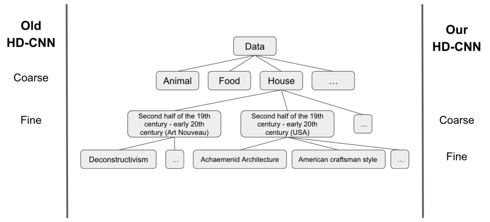

Figure 11: Hierarchy Difference

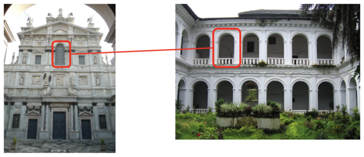

Figure 12: Luminance Learning Sample

features of each fine category under one coarse category if the upper-level feature is similar. Sample in Figure 4, the coarse category component learns the outline difference between A, B, and C. Then the fine category component learns the difference between B.a and B.b.

The manual definition of coarse category may cause the inconsistency of upper-level features of the same coarse category, and the decrease of predicted probability of fine category component, which leads to the decrease in the final probability of HD-CNN.

The Balance in Dataset From Figure 3, we can see that the number of samples under different fine categories is imbalanced. The number of samples in some categories is significantly small, which makes the HD-CNN not learn enough features with missing training samples to support the classification task during the learning process.

The knowledge of coarse category also shows imbalanced. Figure 9 shows the effect of the number of fine categories in the coarse category and the sample size in the coarse category on the loss score of the classification task. The imbalance in category size and sample size affects HD-CNN's classification ability.

"Fine" to "Coarse" In Yan et al. [20] work, the origin HD-CNN uses the CIFAR [7] dataset test to check the model's validity. Unlike the architectural style dataset in our work, the CIFAR dataset covers multiple general categories. In the original paper, the general category is used as the coarse category, and fine categories are the class under the general category. Overall, our classification task is one level more refined than the HD-CNN. Figure 11 gives a schematic diagram of the two models' hierarchy of samples learned.

6 Future Work

We have demonstrated the effectiveness of the introduction of hierarchy for improving the effectiveness of different architectural style classification tasks using CNNs and discussed the experimental weaknesses. This section will discuss our future work on model enhancement. Section 6.1 will discuss about the improvement direction of the existing model. In Section 6.2 we will propose optimizations for the architectural style image classification task.

6.1 In Model

Based on the idea of 'Clustering', we could classify fine categories to coarse categories based on their common upper-level feature. In addition, we can use the Turning model to turn false-positive samples back to the correct category and improve the final probability. In addition, we can set a threshold for the loss score of the coarse category component, T. When the loss score exceeds T, we stop training to control the model's accuracy and change the size of coarse categories.

Regarding the imbalance of the dataset, we can expand the imbalanced data categories by converting the angle of existing images, learning to generate images, etc., in order to improve the ability of fine category component.

6.2 Out Model

We imagine some ideas for the architectural style image classification task to help CNNs learn architectural features better.

Luminance Learning Inspired by Wang et al. [18], we believe that the learning ability of image brightness can enhance CNN learning, making the model better capture the detailed features. When we use our eyes daily, we can recognize the spatial structure by combining luminance and contour. Therefore, we envision that letting the model learn the shadows of the buildings and the lines of the buildings will likewise help the model make a better classification. In Figure 12, it is easy to distinguish between windows and columns by combining lines and shadows.

Relative Position Encoding Relative position encoding(RPE) Shaw et al. [13] in Transformer can model the position relationship between two tokens to learn the relative position or distance between tokens. In recent years, RPE techniques have been demonstrated in natural language processing. More RPE techniques for imaging tasks are also worked out. Therefore, we consider whether we can introduce the RPE technique to classify architectural images to learn the relative position of each building composition in architectural style, making the model better learn.

References

[1] Daniel Stephen Brooks. *The concept of levels of organization in the biological sciences*. PhD thesis, Bielefeld, Universität Bielefeld, Diss., 2014, 2016.

[2] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[3] Yifeng Ding, Zhanyu Ma, Shaoguo Wen, Jiyang Xie, Dongliang Chang, Zhongwei Si, Ming Wu, and Haibin Ling. Ap-cnn: Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification. *IEEE Transactions on Image Processing*, 30:2826–2836, 2021.

[4] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[5] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International conference on machine learning*, pages 1319–1327. PMLR, 2013.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[9] 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.

[10] Chen-Yu Lee, Patrick W Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial intelligence and statistics*, pages 464–472. PMLR, 2016.

[11] Swalpa Kumar Roy, Gopal Krishna, Shiv Ram Dubey, and Bidyut B Chaudhuri. Hybridsn: Exploring 3-d–2-d cnn feature hierarchy for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 17(2):277–281, 2019.

[12] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[13] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[17] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[18] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5239–5247, 2017.

[19] Zhe Xu, Dacheng Tao, Ya Zhang, J. Wu, and Ah Chung Tsoi. Architectural style classification using multinomial latent logistic regression. In *ECCV*, 2014.

[20] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2740–2748, 2015.

[21] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[22] Zhimian Zhang, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. Complex-valued convolutional neural network and its application in polarimetric sar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7177–7188, 2017.

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