Tutorial of Contourlet Convolutional Neural Networks

MIN HSUAN CHENG, P10942A05

**Abstract**

In recent work, deep-learning-based approaches make significant progress in computer vision tasks. People use the convolution neural network (CNN) approaches to achieve multiple tasks. Adding extra information during neural network training can make our model more robust. Contourlet Convolutional Neural Networks (C-CNN) **[1]** author takes an approach to fuse spatial domain approach in this paper use CNN, frequency domain uses contourlet transform, and image statistical information. CNN is a widely used feature extraction approach in the spatial domain. Contourlet can get edge information not only vertical and horizontal. Statistical can get more image information using mean and variance properties. In this tutorial, we focus on the contourlet and statistical approach used in CNN and the quick review of the CNN-based model compared to the C-CNN model different. In the last of this tutorial, we will review some contourlet-based approaches in classic image tasks with deep learning.

1. **Introduction**

In this paper, they focus on the texture classification problem. Texture has some properties like fabric or buildings repeating again and again in a single image. Texture takes an important role in computer vision tasks. Using texture can get more information from images to solve object detection problems. In previous work, texture classification problems used statistical, frequency, and spatial approaches to achieve feature extraction tasks. Then use some classic machine learning approaches like k-nearest-neighborhood (KNN), support vector machine (SVM) to classify image texture. Recently, deep learning approaches are widely used in object detection tasks on MNIST, ImageNet, and CIFAR10 datasets. However, how to get better performance than classic deep learning network structure on texture classification problems. So the author implements C-CNN in multiple classification datasets and gets higher accuracy than other deep learning classic models. C-CNN is mainly different from other CNN-based models in fusing frequency domain information and statistical information.

We often use the wavelet transform to solve image compression and edge detection problem, but wavelet transform has some disadvantages in edge detection problems. Wavelet using in image edge can detect vertical and horizontal edge easily, but in other specific degrees, the result of wavelet transform is not satisfactory. To solve this problem

1. **Background**

In this section, we will focus on building some background knowledge.

* 1. **Convolution neural network (CNN)**

Before we talk about convolution neural networks, we will introduce neural networks structure and mechanism. As a traditional optimum approach, neural networks had been proposed a long time ago. We can find some statistical software like SPSS or JMP that supports neural networks to optimum the factor of different parameters. With hidden layer structure, they called it multilayer perceptron (MLP) Figure 1. We also can call it deep neural networks.

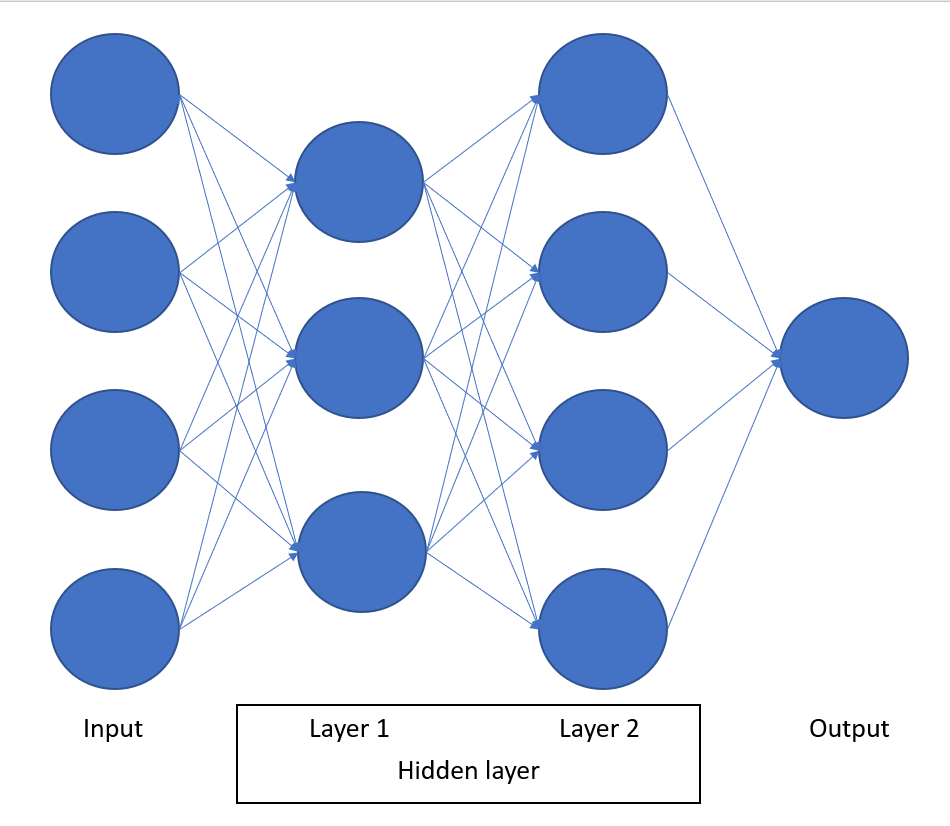


Figure 1: Simple multilayer perceptron (MLP) structure.

The main mechanism in neural networks is forward and backpropagation. Forward is calculating next layer result mechanism, but neural networks are leaning based algorithm we have to use backpropagation to update the weight and bias factor to close our target. So neural networks approach has some properties below.

* A learning-based approach can reduce design parameters’ time, but how to define a proper loss function will be a critical problem.
* Hard to explain the mechanism in hidden layers with different activation functions and deeper structures in neural networks.

A convolution is a common approach to preprocessed frequency and image. Figure 2. In digital image processing, we can use convolution to get the edge and object location information. But to find the object correctly, we have to design convolution filters for different objects.

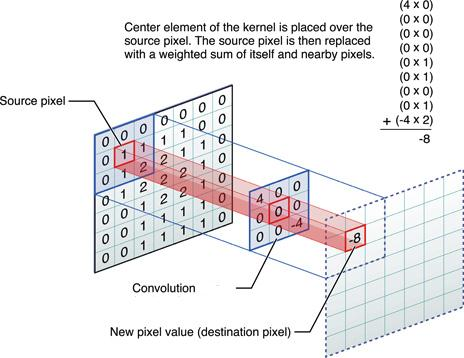
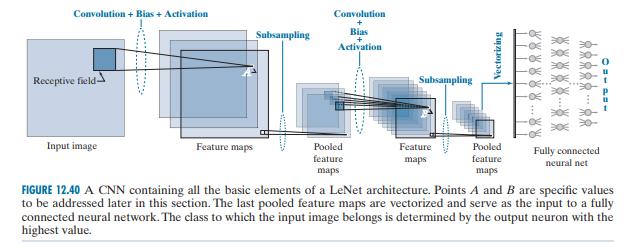


Figure 2: The process of 2D convolution. **[2]**

To solve this problem, a combined deep neural networks and convolution approach had been proposed and we called it convolution neural networks (CNN). We use the LeNet model to explain some basic modules in CNN architecture. In Figure 3 we can observe that the activation function is added after the convolution filter, the activation function can simulate the neural mechanism to prevent largely negative impact. To predict regression or classification problems we tend to flatten model architecture in the last of the model architecture and use multilayers fully connected networks to predict results with continuous value or discrete type.

Figure 3: LeNet architecture **[3]**

* 1. **Contourlet transform**

It is a multiresolution analysis-based method, which describes the energy distribution in spectral. Like wavelet, bandlet, and curvelet also are multiresolution analysis-based methods. Wavelet transform is a more common approach in image compressing and edge detection. But wavelet has some disadvantages in processing multiple direction edge problems. In Figure 4 we use the wavelet transform in the doge image. This picture shows that wavelet transform results have better performance in the vertical and horizontal edge.

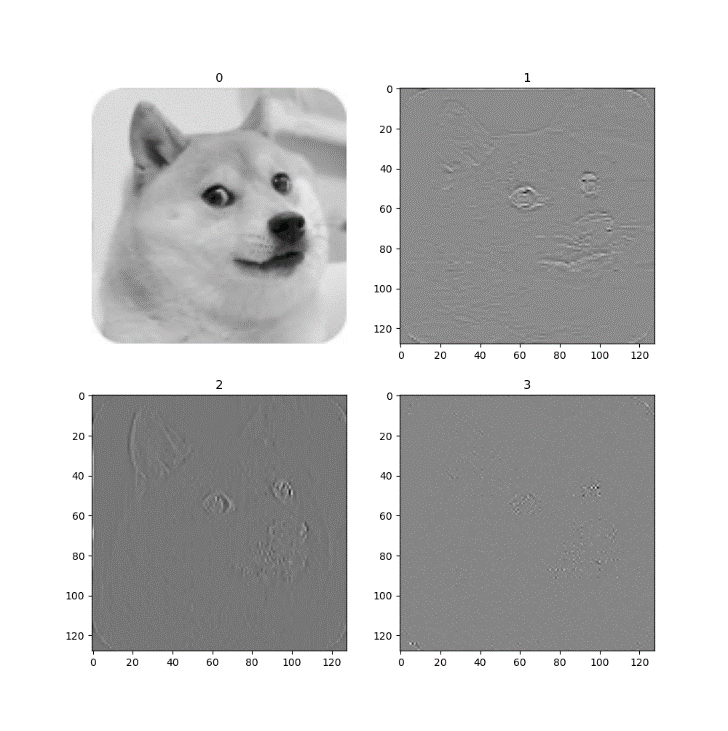


Figure 4: Wavelet transform in doge image

Compare with wavelet transform contourlet transform has better properties in directionality, anisotropy, spatial locality, and bandpass property. It helps our model to capture the geometry of contours with sparsity representation and close to our human visual system (HVS).

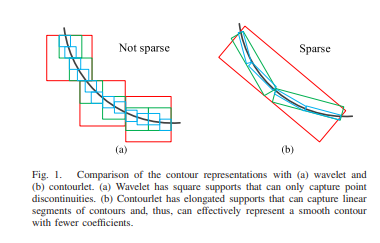


Figure 5: The comparison of contour representation (a) is wavelet transform (b) is contours transform compare with wavelet transform it can use more sparse representation to describe the contour and get better performance. **[1]**

Like our HVS, wavelet transform tends to use fewer visual neurons. HVS includes three features below:

* Multiresolution
* Locality
* Directionality

Next, we will explain why contourlet transform is effectively represented way. First, we can observe Figures 5(a) and 5(b) different and we can approximate the Figure 5(a) contour use below function.

we set as a basis in Fourier or wavelet transform and use nonlinear approximation (NLA) to measure expansion efficiency

is the best M-term approximation of function f with the basis B, is the M-term index and have above function. We can calculate approximation error decay rate in different contour approximation approach like below and more strictly prove can access this paper. []

Contourlet transform has better performance in approximation rate than the other two frequency domain approaches.

How to implement contourlet transform in this paper will be a critical issue because the spatial-spectral feature fusion and statistical feature fusion mechanism use contourlet transform to acquire corresponding parameters. Let us talk about how to implement the contourlet transform. The contourlet transform process like below Figure 6.

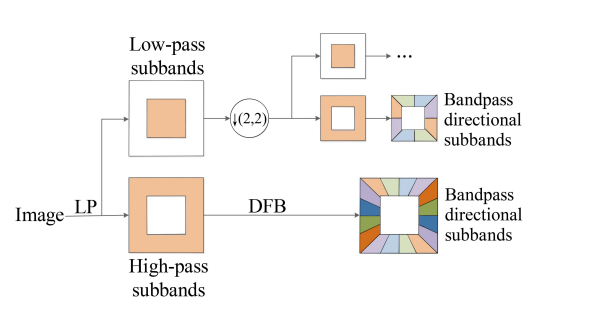


Figure 6: The process of contourlet transformation. **[1]**

The first step is using the Laplacian pyramid filter image to low pass and high pass subbands. Image in high pass subbands contain high-frequency information, it includes edge and noise. Low pass subbands have blurred images with a smooth edge and they will be subsampled before entry next levels. Contourlet transform is an iterated process from coarse to fine, it separated image into high and low frequency and use a directional band filter to get multi-direction edge information. The Laplacian pyramid mechanism is the same as wavelet transform, so the difference between contourlet and wavelet transform is only in the directional filter bands module. The directional filter bands numbers depend on the coefficient we set up, and the filter bands number will be . The directional filter bands location can be represented as below function.

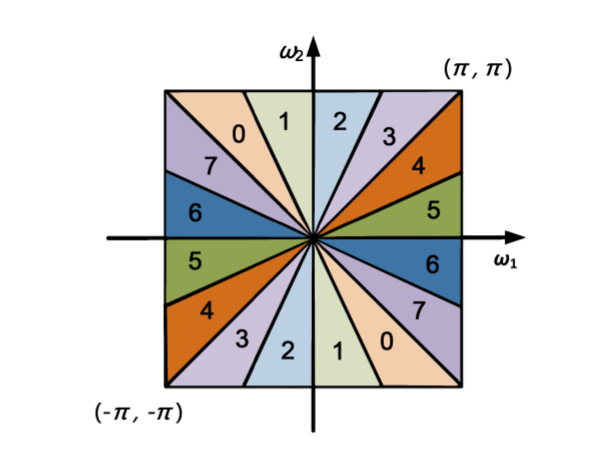


Figure 7: Frequency partitioning when the numbers of directional filter bands are eight. **[1]**

In Figure 7 we show the eight directional filter bands case depending on the above function we can know the corresponding frequency partitioning in the different indexes of filter bands. With different directional filter bands, they can focus on different angle edges not only vertical and horizontal edges. In a traditional image processing algorithm, we will observe image structure and design a corresponding frequency filter to filter or extract the region of interest in the frequency domain and inverse it to spatial domain image.

To achieve directional filter banks, wavelet transform use the shearing operation to simplify the process. The edge will deform after the image shearing operation, and we can use horizontal and vertical directional filters to detect the edge. Then we restore the edge detection result to its original shape it is the process of the directional filter bank.

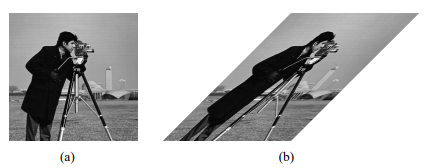


Figure 8: Shearing operation.

In Figure 8 is the process of shearing operation, and picture edge angle deform with this operation. **[4]**

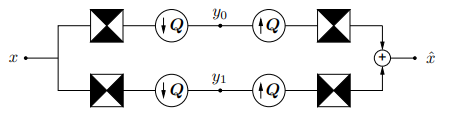


Figure 9: Two-dimensional spectrum filter using quincunx filter banks **[4]**

Figure 9 is the two-dimensional spectrum filter that can detect horizontal and vertical edge-like wavelet transform. The black region represents the ideal frequency in each filter. Q is a quincunx sampling matrix.

We add the shearing operator before the two-dimensional spectrum filter and add the reverse shearing operator after it. It’s the process of directional band filter.

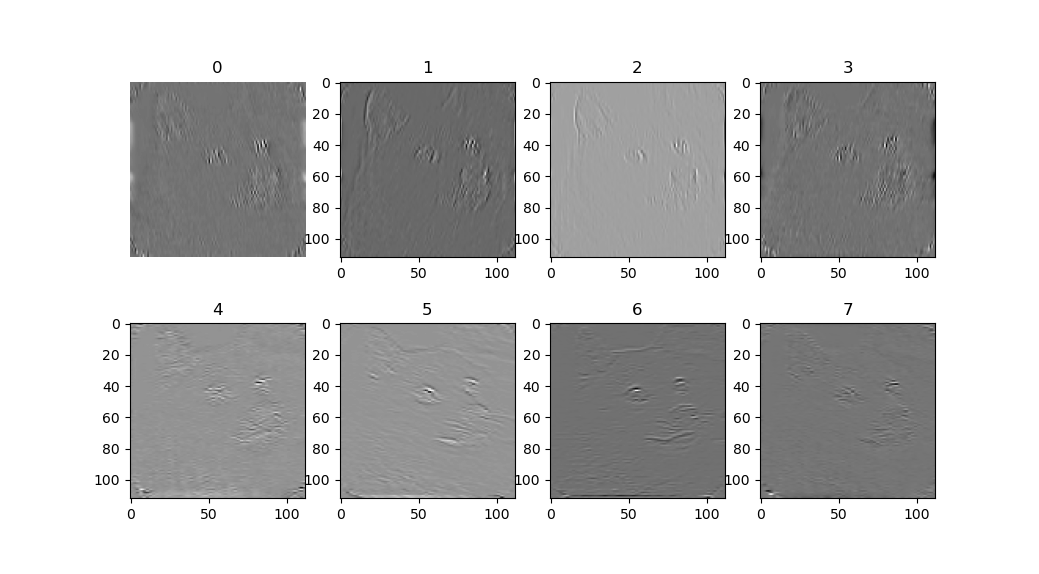


Figure 10: Contourlet transform in doge image.

1. **Contourlet Convolutional Neural Networks**

The contourlet convolution neural network (C-CNN) combines convolution neural network and contourlet information in one. The details of the C-CNN architecture are in Figure 11. We can observe input images will through the Laplacian pyramid to high passband and low passband and use the directional filter bands (DFB) to process high passband images. Low passband through two-dimension convolution from 3 channel to 64 channel and use kernel size 3x3 matrix with 1x1 padding and 2 strides instead of the pooling operation.

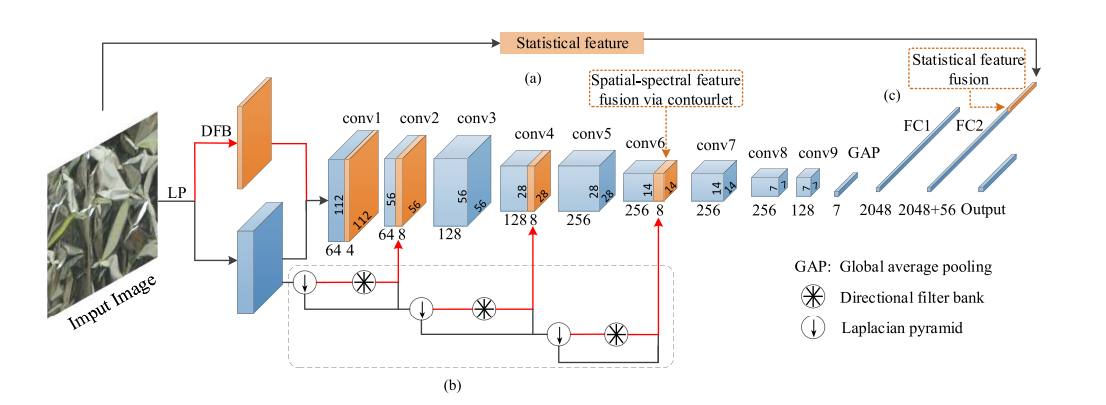


Figure 11: C-CNN network architecture **[1]**

CNN is an efficient method in spatial feature extraction, but texture analysis needs more spectral information to enhance its performance. We concatenate the DFB result in the conv1 module to give our network more spectral data.

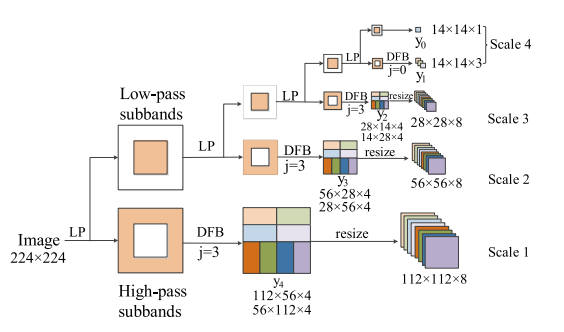


Figure 12: LPDFB number [0, 3, 3, 3] from coarse to fine **[1]**

The details of Laplacian pyramid directional filter bands show in Figure 12. First, we use the Laplacian pyramid to filter high-pass and low-pass subbands. Next use different level directional filter bands to get different direction spectral features. If the level of directional filter bands is 3 we will get coefficients images. The first 4 images are horizontal and the rest of the images are vertical we resize the elongated images into squares and concatenate them.

However, when I read this section I had confused about the DFB numbers. The author provides a picture to explain DFB numbers [0, 3, 3, 3] from coarse to fine in Figure 12. Following the DFB numbers [0, 3, 3, 3] the concatenated channel size will be 8, so we concatenate 8 channels in the process of conv1 and 4 channels in the process of conv6.

Another difference in C-CNN architecture is using statistical features concatenate on the fully connected layer to fuse more information. In the above function, l represents the number of decomposition levels and we can calculate the corresponding statistical feature then concatenate on a fully connected layer.

In this implementation, C-CNN loss uses logistic loss with softmax function, it also implements as CrossEntropyLoss function in PyTorch **[5]** package.

represent the input with T number of classes.

The optimizer uses adaptive moment estimation (Adam) which combines RMSProp and Momentum to achieve faster convergence. The details of the Adam optimizer algorithm are in Figure 13. Have a little bit different from the C-CNN paper so we use the original optimizer function in Adam paper

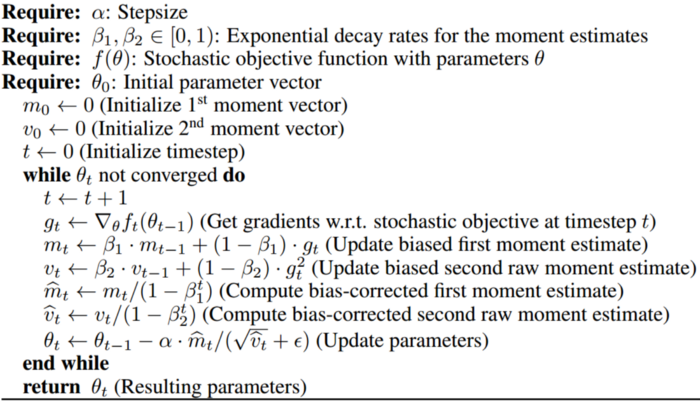


Figure 13: Optimizer pseudo code from Adam paper. **[6]**

1. **Implementation**

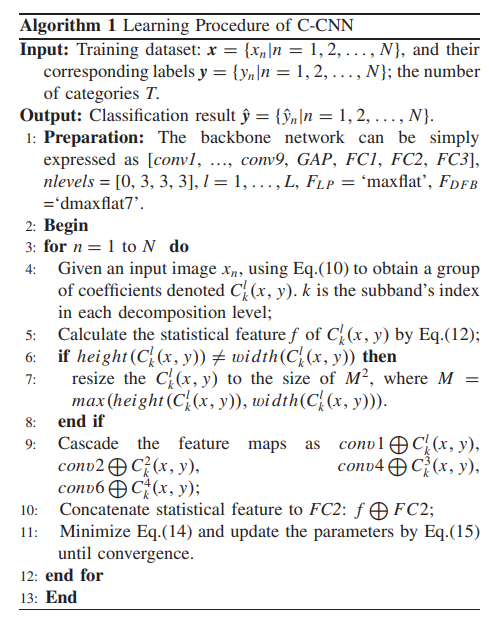


Figure 14: Training pseudo code from C-CNN paper **[1]**

Figure 14 shows the training process of C-CNN. First, we select the UC Merced data set (UCM) **[7]** which has 21 classes and per class has 100 images with 256x256 pixels. For our training, we resize images to 224x224 pixels. To accelerate our training process we save our LPDFB result with level numbers [0, 3, 3, 3] local. The C-CNN model architecture derived from the AlexNet **[8]** model uses nine convolution layers one global average pooling (GAP) layer and three fully connected layers. Replace average pooling operation using 3x3 matrix, padding 1x1, and two stride lengths in the convolution layer 2,4,6,8. Concatenate LPDFB spectral information in convolution 1,2,4,6 with 8,8,8,4 channels. In the Laplacian pyramid, we use the “max flat” filter and DFB uses the “dmaxflat7” filter. We use the LPDFB result and resize it to the square image to get statistical features ((8+8+8+4) \* 2) = 56 then concatenate the result to the FC2 layer. After building our model structure we use PyTorch CrossEntropyLoss as our loss function and use Adam optimizer to train our model until convergence. However, when we implement this model encounter some difficulties in hyperparameters and image preprocess setting like the dropout rate setting and images transform process. The author also mentioned they use some regular operations such as batch normalization and dropout, but they don’t mention the exact value and location in their C-CNN model.

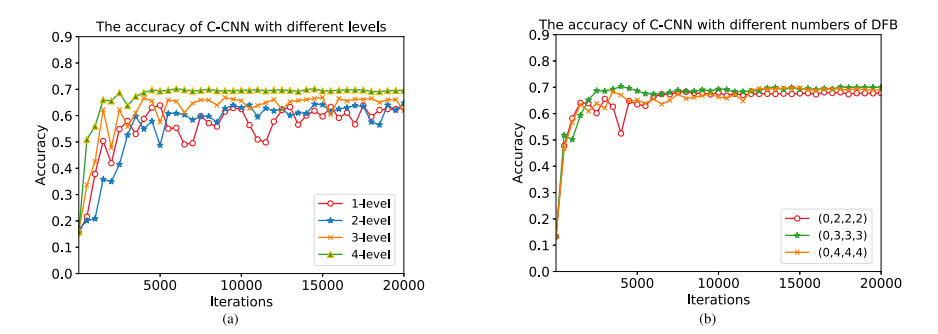


Figure 15: LPDFB parameters selected analysis **[1]**

We use the paper result to explain why they select four-level LPDFB and DFB numbers are [0, 3, 3, 3] from coarse to fine. In their experiments, they compare with different level decomposition result in Figure 15(a) and select the best result four levels decomposition. After selecting four levels of LPDFB decomposition, they compare different numbers of DFB in Figure 15(b)

[0, 3, 3, 3] get the best accuracy performance.

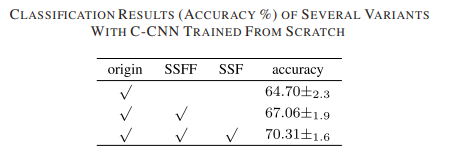


Figure 16: Classification result with different feature fuse methods. **[1]**

They also experiment with spatial-spectral feature fusion (SSFF) methods and statistical feature fusion (SFF) impact in the C-CNN model on the kth-tps2b data set. In Figure 16 “origin” means splice the elongated decomposed images. “SSFF” we resize the elongated decomposed images into corresponding sizes. “SSF” is concatenating statistical features on a fully connected layer. The model with SSFF and SSF methods has the best performance.

Following the paper’s pseudo code we build our model and data loader pipeline for training. In our code, we preprocess images and save their values to a NumPy array list with four dimensions with different levels of DFB. In the data loader, we calculate mean and standard deviation values per image and return DFB information and statistical features. Using the above methods can accelerate model forward training.

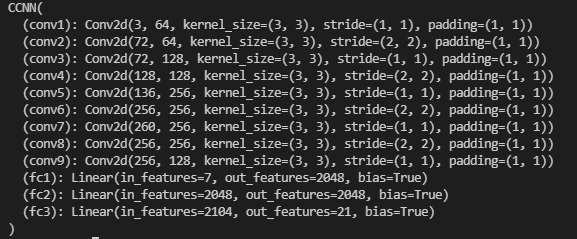


Figure 17: The details model in our implementation

Figure 17 is our model structure implemented on PyTorch. However, with some unknown parameters and details, we didn’t have high accuracy in the UCM remote sensing dataset. Figure 18 shows our validation set result accuracy in 60 epochs.

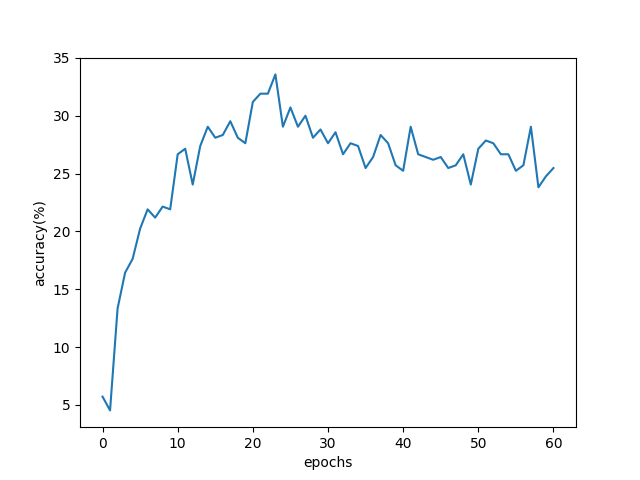


Figure 18: The details model in our implementation

But we provide code with these tutorials it includes model, train, and images preprocess to four levels of DFB code. We also try to use EfficientNet **[9]** for our training the validation dataset accuracy log in Figure 19.

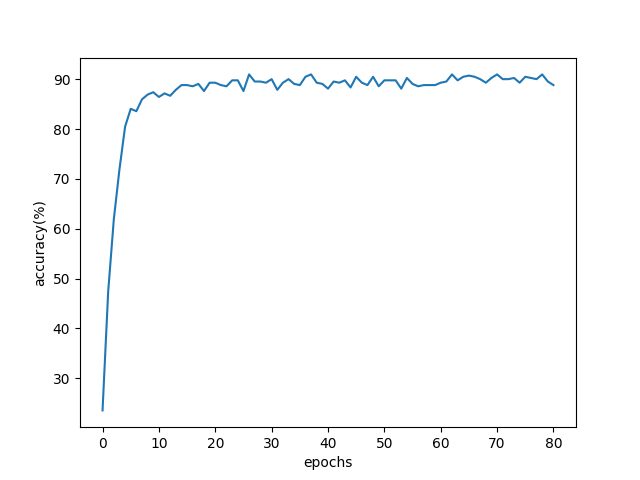


Figure 19: The details model in our implementation

EfficientNet with pretrained weight converges in the location about 90% accuracy.

1. **Conclusion**

We implement the C-CNN model and train it. Although the result of our implementation has some gap with the paper result. To improve our implementation result, we think can pretrained our network on the ImageNet dataset and fine-tune our model using regular methods like batch normalization and dropout.

However, we learn the contourlet transform method to acquire multi-direction high-frequency information, and the contourlet transform’s directionality and locality can be used in other computer vision tasks. In recent work, the fuse mathematical and deep learning method is more popular than before. We hope this tutorial can help the reader to fuse features and deep learning structures in future work.

We also find some interesting related work using contourlet transform and deep learning methods. “**ContourletNet: A Generalized Rain Removal Architecture Using Multi-Direction Hierarchical Representation**” **[10]** focus on rain removal task. In the rain removal problems, processing multi-direction raindrops and extracting them is a critical issue. The contourlet transform has a better representation in directionality and locality to address both heavy rain and moderate rain scenarios effectively.

1. **References**

[1] M. Liu, L. Jiao, X. Liu, L. Li, F. Liu and S. Yang, "C-CNN: Contourlet Convolutional Neural Networks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 6, pp. 2636-2649, June 2021, doi: 10.1109/TNNLS.2020.3007412.

[2] <https://developer.apple.com/library/archive/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html> Copyright © 2016 Apple Inc.

[3] Gonzalez, Rafael C, and Richard E. Woods. Digital Image Processing., 2018. Print.

[4] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," in IEEE Transactions on Image Processing, vol. 14, no. 12, pp. 2091-2106, Dec. 2005, doi: 10.1109/TIP.2005.859376.

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Brad-bury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary De-Vito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner,Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperativestyle, high-performance deep learning library. In H. Wallach, H. Larochelle,A. Beygelzimer, F. d'Alch ́e-Buc, E. Fox, and R. Garnett, editors, Advancesin Neural Information Processing Systems 32, pages 8024–8035. Curran As sociates, Inc., 2019.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017

[7] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in Proc. 18th Int. Conf. Adv. Geographic Inf. Syst. (SIGSPATIAL GIS), 2010, pp. 270–279.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12). Curran Associates Inc., Red Hook, NY, USA, 1097–1105.

[9] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. CoRR, abs/1905.11946, 2019

[10] Wei-Ting Chen, Cheng-Che Tsai, Hao-Yu Fang, I-Hsiang Chen, Jian-Jiun Ding, and Sy-Yen Kuo. Contourletnet: A generalized rain removal architecture using multi-direction hierarchical representation. In Proceedings of the British Machine Vision Conference, 2021.