1 PP-LCNet: A Lightweight CPU Convolutional Neural Network

1.1 Abstract

This paper lists technologies which can improve network accuracy while the latency is almost constant. With these improvements, the accuracy of PP-LCNet can greatly surpass the previous network structure with the same inference time for classification

1.2 Introduction

As the model feature extraction capability increases and the number of model parameters and FLOPs get larger, it becomes difficult to achieve fast inference speed.

We consider the following three fundamental questions: 1.How to promote the network to learn stronger feature presentations without increasing latency. 2.What are the elements to improve the accuracy of lightweight models on CPU. 3.How to effectively combine different strategies for designing lightweight models on CPU.

We come up with several general rules for designing lightweight CNNs.

1.3 Related Works

Two types of methodologies to promote the capabilities of the models: 1.Manually-designed CNN Architecture. 2.Neural Architecture Search(NAS)

1.3.1 Manually-designed Architecture

VGG: Stacking blocks with the same dimension.

GoogLeNet: Inception block(including four parallel operation: 1×1 convolution, 3×3 convolution, 5×5 convolution and max pooling) which can make the convolution neural network light enough.

MobileNetV1: Depthwise and pointwise convolutions replace standard convolution.

MobileNetV2: Inverted block,which reduces the amount of parameters and FLOPs of the model.

ShuffleNetV1/V2: Exchange information through channel shuffle.

GhostNet: Ghost module, which can generate more feature maps with fewer parameters to improve the overall performance of the model.

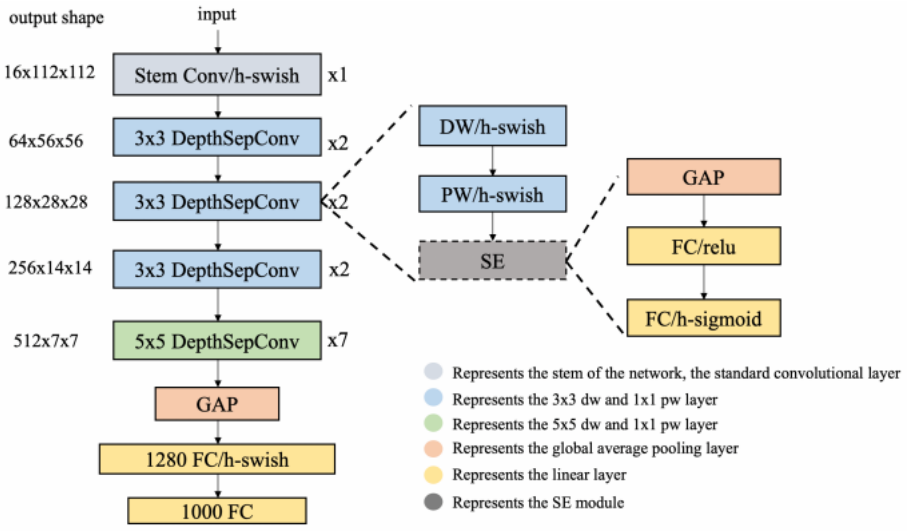
1.3.2 Neural Architecture Search

MixNet: Hybridize depthwise converlations of different kernel size in one layer.

1.4 Approach

Here we have summarized some methods that can improve the performance of the model with little increase of inference time.

Concat or elementwise-add will not only slow down the inference speed of the model, but also will not improve the accuracy on a small model.



Stem Conv uses standard 3×3 convolution.DepthSepConv means depth-wise separable convolutions, DW means depth-wise convolution, PW means point-wise convolution, GAP means Global Average Pooling.

1.4.1 Better Activation Function

1.Sigmoid 2.ReLU 3.Swish activation funvtion 4.H-Swish

We replaced the activation function from ReLU to H-Swish.The performance has been grearly improved, while the inference time has hardly changed.

1.4.2 SE Modules at Appropriate Position

It does a good jib of weighting the network channels for better features,and its speed improvement version is also used in many lightweight networks.

The SE module increases the inference time,so that we cannot use it for the whole network.SE module is located at the end of the network,it can play a better role.So we just add the SE module to the blocks near the tail of the network.

The activation functions for the two layers of the SE module are ReLU and H-Sigmoid.

1.4.3 Larger Convolution kernels

The size of the convolution kernel often affects the final performance of the network.

However, mixing different sizes of convolutional kernels in the same layers of the network slows down the inference speed of the model, so we try to use only one size of convolution kernel in the single layer, and ensure that a large convolution kernel is used in the case of low latency and high accuracy.

We replace the 3×3 convolutional kernels with only the 5×5 convolutional kernels at the tail of the network would achieve the effect of replacing almost all layers of the network.

1.4.4 Larger dimensional 1x1 conv layer after GPA

When the output dimension of the network is small,in order to give the network a stronger fitting ability, we appended a 1280-dimensional size 1×1 conv(equivalent to FC layer) after the final GAP(Global Average Pooling) layer,which would allow for more storage of the model with little increase of inference time.

1.5 Ablation Study

1.5.1 The impact of SE module in different positions

Network	SE Location	Top-1 Acc (%)	Latency (ms)
PP-LCNet-0.5x	1100000000000	61.73	2.06
	0000001100000	62.17	2.03
	0000000000011	**63.14**	**2.05**
	1111111111111	64.27	3.80

The table clearly shows that adding the last two blocks is more advantageous for almost the same inference time.

1.5.2 The impact of large-kernel in different locations

Network	Large-kernel location	Top-1 Acc (%)	Latency (ms)
PP-LCNet-0.5x	1111111111111	63.22	2.08
	1111111000000	62.70	2.07
	0000001111111	**63.14**	**2.05**

The table shows the positions added by the 5×5 depth-wise convolution.

1 means that the depth-wise convolution kernel in DepthSepConv is 5×5, and 0 means that the depth-wise convolution kernel in DepthSepConv is 3×3.

Strategy	Top-1 Acc.(%)	Latency(ms)
BaseNet	55.58	1.61
+h-swish	58.18	1.66
+large-kernel	59.09	1.70
+SE	59.91	1.85
+last-1x1 conv w/o dropout	62.50	2.05
+last-1x1 conv w/ dropout	**63.14**	**2.05**

1.5.3 The impact of different techniques

At the same time, perhaps because a relatively large matrix is involved here, the use of the dropout strategy can further improve the accuracy of the model.

1.6 Conclution and Future Work

The CNN used the approach shows stronger performance on a large number of vision tasks and has a better accuracy-speed balance.

In addition, this work reduces the search space of NAS and also offers the possibility of faster access to lightweight models for NAS.

In the future, we will also use NAS to obtain faster and stronger models.