1.13MB Model for 2360 Classes Face Recognition

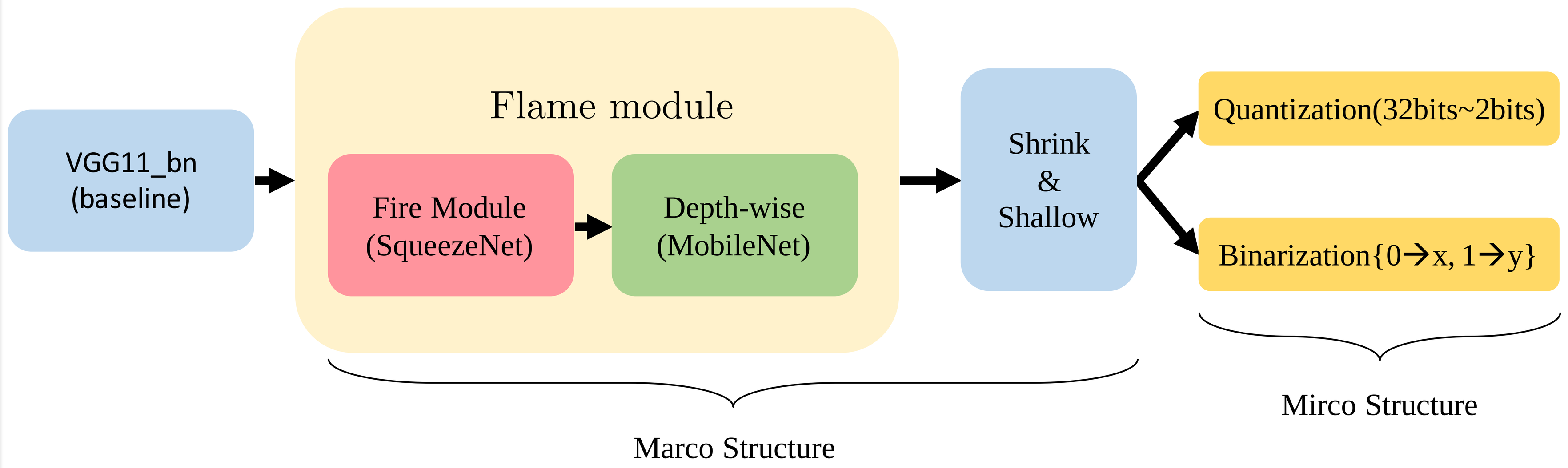
Ching-Yen Shih, Po-Hsiang Huang, Sheng-Lung Chung
National Taiwan University

Introduction

In conventional deep structure network for computer vision task, most of the parameters are distributed at the CNN structure due to the kernel size and the number of filters.
In this project, we show how we compress the model by substituting the convolutional layers with lighter structure and quantizing model parameters.
We present and analyze our method on the face recognition task on the CelebA dataset, *with best model reduce about 97% of size on convolutional structure but about only 1% performance drop compared to our baseline model.*

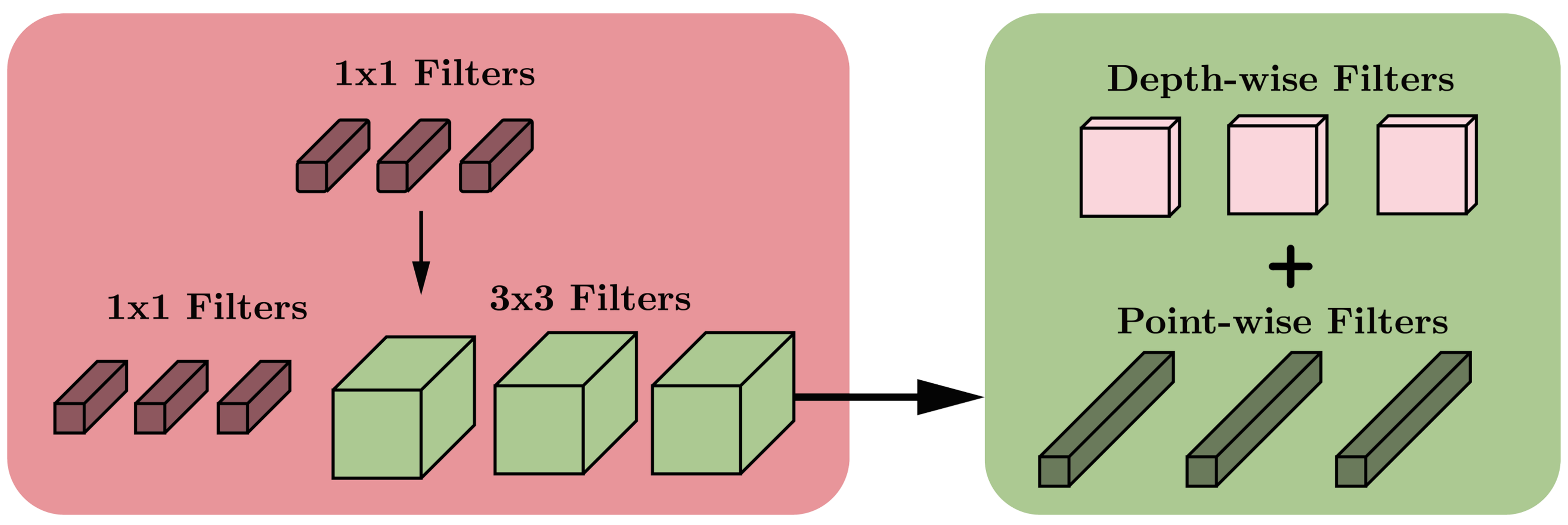
Method

• Compression Flow Chart



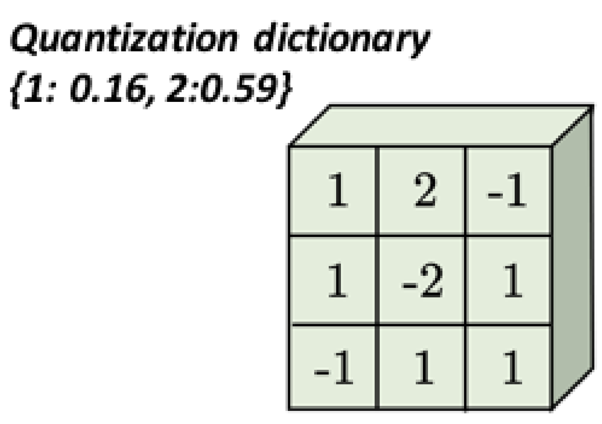
- We compress the CNN model stage by stage:
  (1) Replace the convolution layer by *"Flame module"*.
  (2) Reduce the number of filters and the depth of the network.
  (3) Quantize or binarize the weights by log min-max quantization.

• Flame module

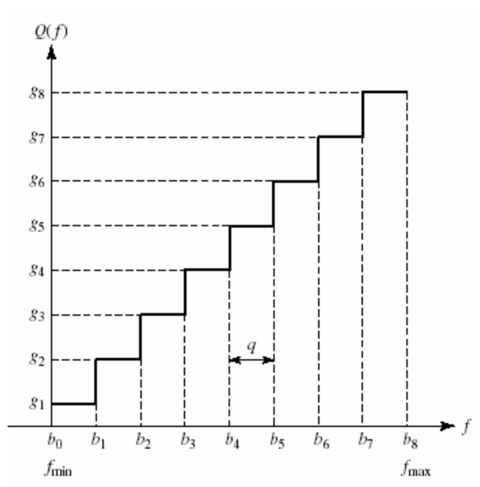


- We fuse the idea of the SqueezeNet [1] and MobileNet [2]: Merge the fire modules with depth-wise 3x3 filters and 1x1 point-wise filters. With this strategy, we can save about 95.7% of parameters every substitution compare to original convolution.

• Log min-max quantization



Quantize-to-Floating



Min-max quantization

$$f = \log|x|, \qquad s = sign(x)$$

$$f_{min} = \min(f), \qquad f_{max} = \max(f), \qquad q = \frac{(f_{max} - f_{min})}{2^b}$$

$$Q'(f) = floor\left(\frac{f - f_{min}}{q}\right) \times q + f_{min}, \qquad Q(x) = e^{Q'(f)} \times s$$

- We create a quantization dictionary containing floating numbers based on **log min-max quantization method.** Each floating number correspond to a number with fewer bits which replacing the original weightings in the convolutional layer.

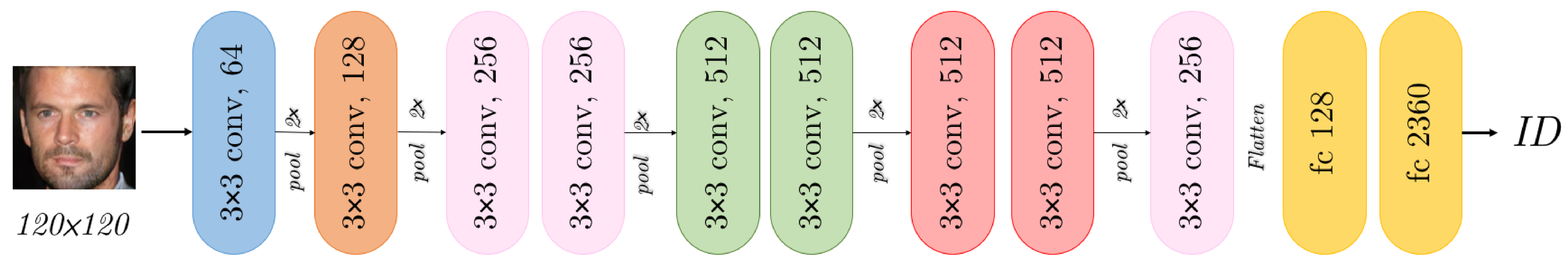
• Comparison of Different Module Parameters

CNN structure	Parameters (10 thousands)	Compression ratio
Standard CNN	1179.648	--
Fire Module	180.224	84.72 %
Depth-wise + Point-wise	133.376	88.69 %
Flame CNN	*49.728*	*95.78 %*

*Filter parameters with input channel=256, output channel=512, kernel size = 3×3

Experimental Results

• Baseline model: VGG11_bn



• Training Details

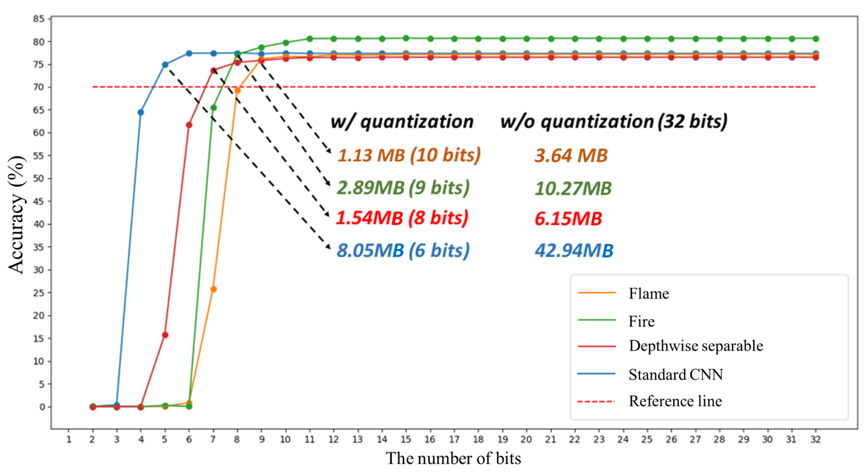
Data Preprocessing:
crop 120×120, random horizontal flip, random rotation
Hyper Parameters:
batch size 32, lr 1e-5, s_ratio 0.125, Adam optimizer
Training tips:
Distillation, L2_norm on feature

• Stage1: Comparison of layer substitution

Model	Network Size/ Compression ratio	FLOPs	Validation Accuracy(%)
VGG11_bn	42MB / --	2120 M	77.33
Fire module	6.1MB / 85%	280 M (↓ 86.7%)	79.13(+1.8)
Depth-wise & Point-wise	10.4MB / 75.2%	260 M (↓ 87.7%)	79.66(+2.33)
Flame module	3.7MB / 91.2%	110 M (↓ 94.8%)	76.42(-0.91)
+Shrink*	***3.2MB / 92.4%***	***31.7 M** (↓ 98.5%)*	***78.02(+0.69)***
+Shallow*	3.2MB / 92.3%	71.7 M (↓ 96.6%)	74.8(-2.53)
+Shrink & Shallow	2.7MB / 93.5%	27.1 M (↓ 98.7%)	63.29(-14.04)

*Shrink: Halve the # of filters except for last two CNN.
*Shallow: Remove layer 3,5,7 of the original model.

• Stage2: Comparison after weight quantization



Methods	# of bits / Size	Compression ratio	Validation Accuracy
Vanilla VGG11_bn	32 / 42.94 MB	-	77.33%
VGG11_bn	6 / 8.05 MB	81.25%	74.89% (-2.44%)
Depthwise & pointwise	9 / 2.89 MB	93.27%	77.12% (-0.12%)
Fire	8 / 1.54 MB	96.41%	73.67% (-3.66%)
Flame	10 / 1.13 MB	**97.37%**	76.11% (-1.22%)

• Comparison of different quantization method

Quantization methods	# of bits	Network Size/ Compression ratio	Validation Accuracy (%)
w/o quantization	32	42MB / --	77.33
log min-max (CNN only)	6	9.15 MB (↓78.69%)	76.29(↓1.04)
log min-max (whole network)	6	8.05 MB(↓81.25%)	74.89(↓2.44)
Binarization training	**2**	**3.97 MB(↓90.75%)**	**76.34(↓0.99)**

Conclusion

We deal with the problem of model compression from two aspect:
(1) Proposed more efficient CNN structure "Flame module" to reduce about 91% of computation steps.
(2) Quantize the weights of model to save about 95% of storage space.

Finally, we get our best model with 97.37% of compression ratio (1.13MB) compared to baseline model but can get 76.11% accuracy on 2360 classes face recognition task.

Reference: *[1] "Squeeznet: Alexnet-level Accuracy with 50X fewer parameters and < 0.5 MB model size", Forrest N. Iandola et al., ICLR 2017;*
[2] "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", Andrew G. Howard et al., CoRR, abs/1704.04861, 2017