RNNPool : Efficient Non-linear Pooling for RAM Constrained Inference

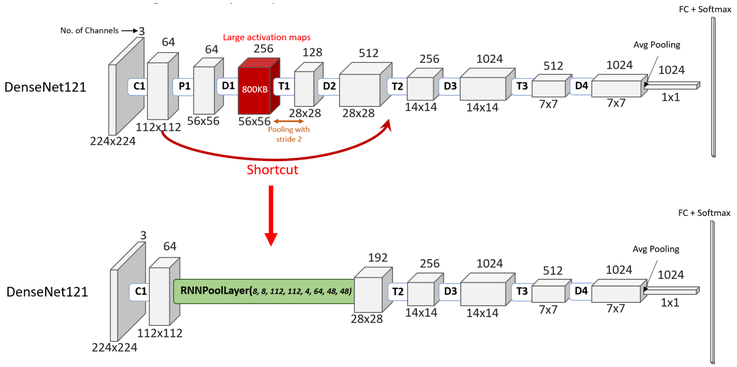
Code: https://github.com/Microsoft/EdgeML

Oindrila Saha[1], Aditya Kusupati[2], Harsha Vardhan Simhadri[1], Manik Varma[1] and Prateek Jain[1]

[1]Microsoft Research [2]University of Washington

Our goal: Accurate computer vision models that can be deployed on tiny devices

Barriers:
- CNN models have many layers with large activations
- Large memory footprint, the most constrained resource on microcontrollers



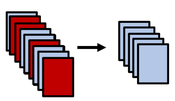
- Standard pooling operators (e.g. max pool) are gross aggregators, thus are only used with a maximum stride of 2
- RNNPool can reduce intermediate feature maps significantly (up to 16x) with small loss in accuracy

  ➔ Ensures heavy convolution blocks run on smaller activation maps

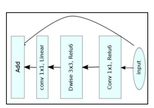
Existing Works

Model Compression

Decreasing Model Size
- Pruning
- Sparsification

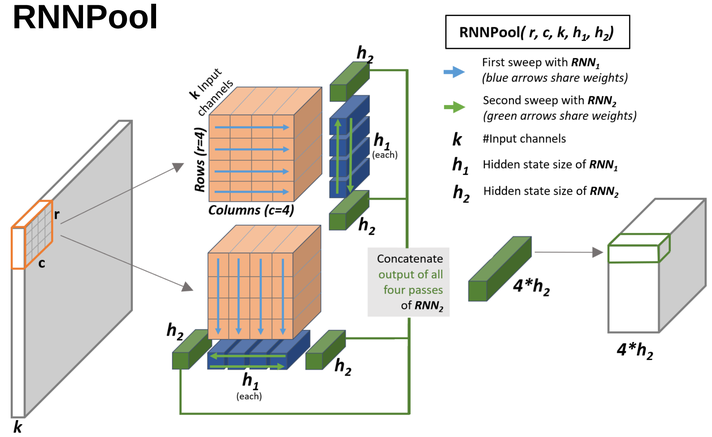


Decreasing Compute
- Depthwise Separable Convolutions
- MobileNets , EfficientNets



However, peak RAM requirement still remains high

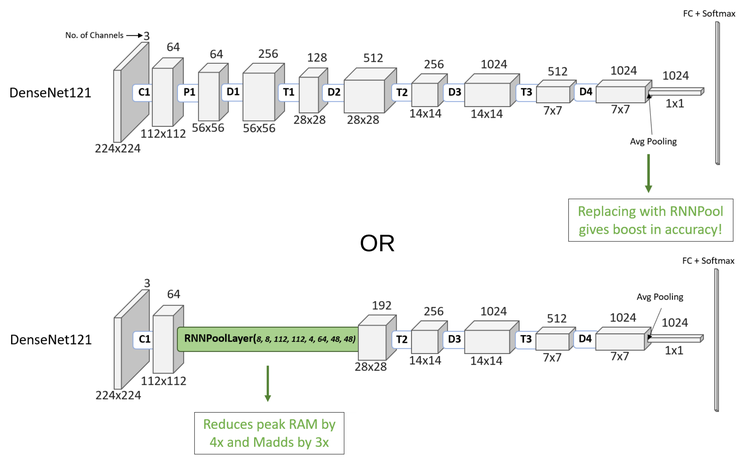
RNNPool



- Takes a patch of the input and produces a 1x1 summary
- Patches having overlap of size = PatchSize – Stride
- For each patch, 4 RNN runs produce the pooled feature vector

Usage

- Semantically equivalent to pooling, so can be used to replace any pooling operator
- But key usage in reducing image size in beginning of network to save RAM requirement



Evaluation

Task 1: Image Classification

Results on ImageNet-10 dataset formed by subsampling 10 classes from ImageNet1K and MobileNetV2 as the base architecture

Model	Accuracy	MAdds	#Params	RAM
Base Network	94.2	0.300G	2.2M	2.29MB
Average Pooling	90.8	0.334G	2.0M	0.24MB
Max Pooling	92.8	0.200G	2.0M	
Strided Conv	93.0	**0.200G**	2.1M	
ReNet	92.2	0.296G	2.3M	
RNNPool	**94.4**	0.226G	**2.0M**	

Standard Pooling Techniques

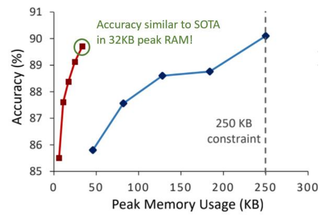
10x Lower RAM !

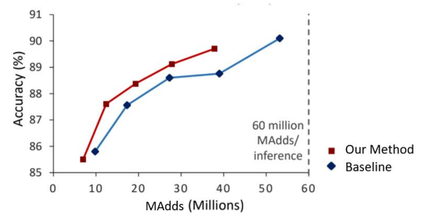
Task 2: Visual Wakeword ➔ Predict whether a person is present

- 8x less RAM
- 40% less compute


No person Present


Person Present





Task 3: Face Detection

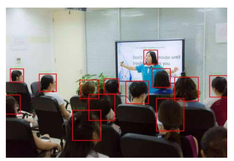
Comparison with SOTA for very low MAdds category on WIDER FACE validation subset

Method	RAM	#Params	MAdds	MAP		
				E	M	H
EagleEye	1.17MB	0.23M	0.1G	0.74	0.70	0.44
Rpool-Face-Quant	**225KB**	**0.07M**	0.1G	**0.80**	**0.78**	**0.53**

Face Detection M4 Deployment ➔

- 188KB peak RAM
- 70M MAdds
- 160KB Model Size

➔ 10.45 sec/image on STM32F439-M4 device clocked at 168 MHz



SCUT Head Dataset: https://github.com/HCIILAB/SCUT-HEAD-Dataset-Release

Image from https://researchdesignlab.com/stm32-arm-cortex-m4-development-board-stm32f407vet6.html