# GLOBAL FEATURE REPRESENTATION USING SQUEEZE, EXCITE AND AGGREGATION NETWORKS (SEANET)

Akhilesh Pandey*, S. Balasubramanian, Gunasekar D, Karam Kulung Rai, Darshan Gera

* akhilesh.sssihl@gmail.com

MIND2019, 4/03/2019

# *Outline*

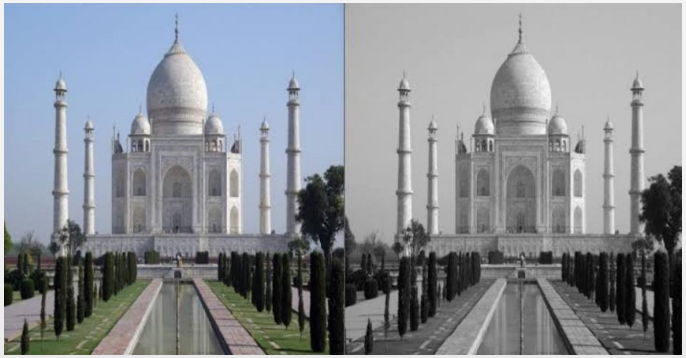
- Introduction
- Motivation
- Proposed architecture
- Results and analysis
- Conclusion

# Introduction

## Continuous evolvement in Deep Learning models

- ❑ **InceptionNet (**ILSVRC 2014**)**
  - ✓ uses a set of filters of different sizes at different layers
- ❑ **VGGNet (**1st runner up ILSVRC 2014**)**
  - ✓ fixes some hyper-parameters like #filters, activation functions etc. thereby eliminating the need of fine-tuning
- ❑ **ResNet** (ILSVRC 2015)
  - ✓ learns a residual mapping to ease training
- ❑ **SENet (**ILSVRC 2017**)**
  - ✓ emphasizes on importance of feature maps

# Motivation

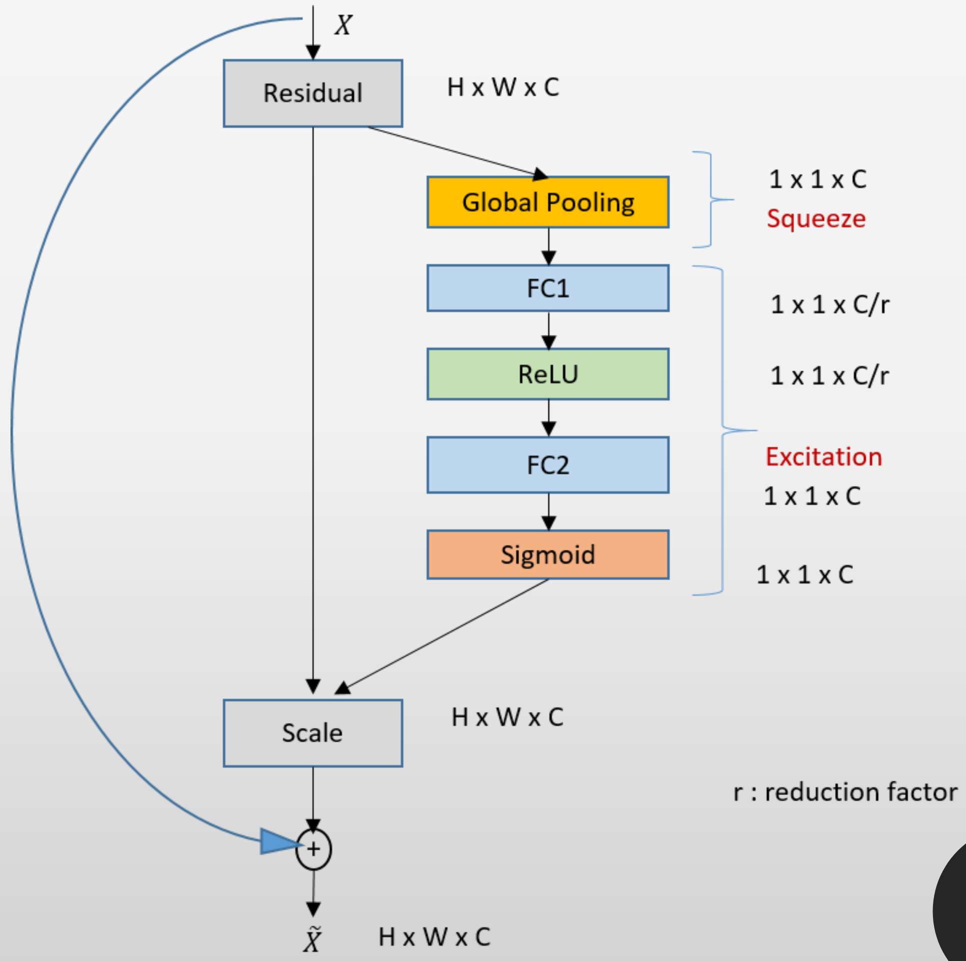


This is a fundamental work on architectural design.

- ❑ Previous architectures does not focus on redundancies across feature maps
- ❑ Aggregating RGB to grayscale reduces the redundancies across feature maps at the same time restores important ones
- ❑ Extending the idea of batch-normalization
- ❑ We add aggregate layer to SE-block to get SEANet

# *Preliminary: SENet*

- Uses Squeeze-Excitation(SE) block
- **Squeeze**: Extracting global information using global avg pooling
- **Excite**: Learns weights corresponding to channels and multiply the output with the learned weights



# *Proposed Architecture*

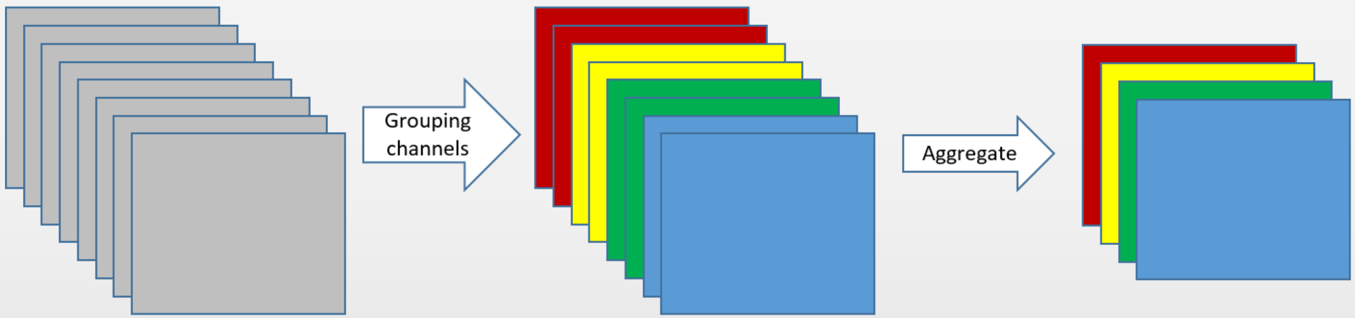
## SEANet

- ❑ Based on SENet (SENet is based on ResNet-20)
- ❑ Uses Aggregate block after SE-block of each Res-block
  - ➢ Incoming feature maps are divided into groups of k features where k is the aggregation factor
  - ➢ Each group is down sampled to a single feature map  using aggregate operation
- ❑ Aggregation operation – sum, mean, max etc

We use sum as aggregation operation and k = 4

# *Proposed Architecture*

Note: channels with same colour belong to same group




Output from SE-Layer

Grouping channels

Result after Aggregation block

# Aggregate operation

# *Advantages of Aggregation*

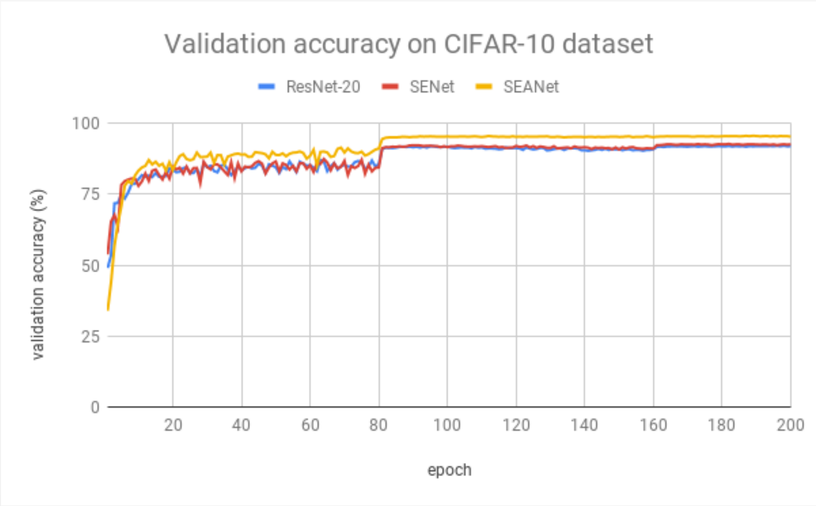
- Minimized redundancies in feature representations
- A global representation across feature maps is obtained
- With sum as aggregate operator significant flow of gradient takes place
- Significant improvement in performance

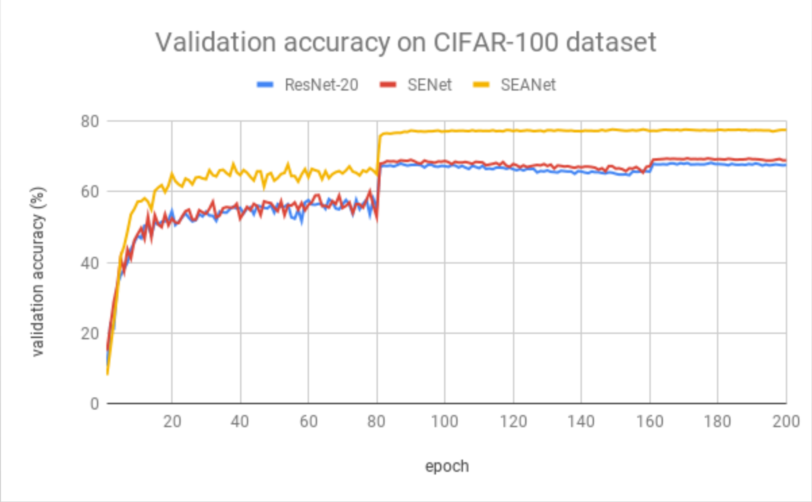
# *Implementation details*

- Trained on Titan-X GPU
- Implemented using PyTorch 0.4 framework
- Codes are available at https://github.com/akhilesh-pandey/SEANet-pytorch

# Results and Analysis

Validation accuracy on
CIFAR-10 and CIFAR-100 datasets



# *Results & Analysis cont.*

1. Comparison with ResNet
(Won ILSVRC in 2015)

Classification error(%)

| Architecture | CIFAR-10 | CIFAR-100 |
|---|---|---|
| **SEANet** | **4.3** | **21.33** |
| Res-20 | 8.6 | 32.63 |
| Res-32 | 7.68 | 30.2 |
| Res-44 | 6.43 | 26.85 |
| Res-56 | 6.84 | 26.2 |
| Res-110 | 6.34 | 26.67 |

# *Results & Analysis cont.*

2. Comparison with SENet
(Won ILSVRC 2017)

Classification error(%)

| Architecture | CIFAR-10 | CIFAR-100 |
|---|---|---|
| **SEANet** | **4.3** | **21.33** |
| SENet | 7.17 | 30.45 |

# *Result & Analysis cont.*

3. Comparison with other state-of-the-art EncapNet (2018)

Classification error(%)

| Architecture | CIFAR-10 | CIFAR-100 |
|---|---|---|
| **SEANet** | **4.3** | **21.33** |
| EncapNet | 4.55 | 26.77 |
| EncapNet+ | 3.13 | 24.01 |
| EncapNet++ | 3.10 | 24.18 |

# *Conclusion*

- ✓ Proposed a new architecture SEANet
- ✓ Reduces feature redundancies
- ✓ Uses aggregate layer after SE-block
- ✓ Reduces error by 2% and 3% more compared to ResNet and SENet respectively on CIFAR-10
- ✓ Reduces error by 6% and 9% more compared to ResNet and SENet respectively on CIFAR-100
- ✓ Outperforms the latest EncapNet and both its variants EncapNet+ and EncapNet++ on CIFAR-100 dataset by around 2%.


Thank you!!! ...