An End-to-End Compression Framework Based on Convolutional Neural Networks

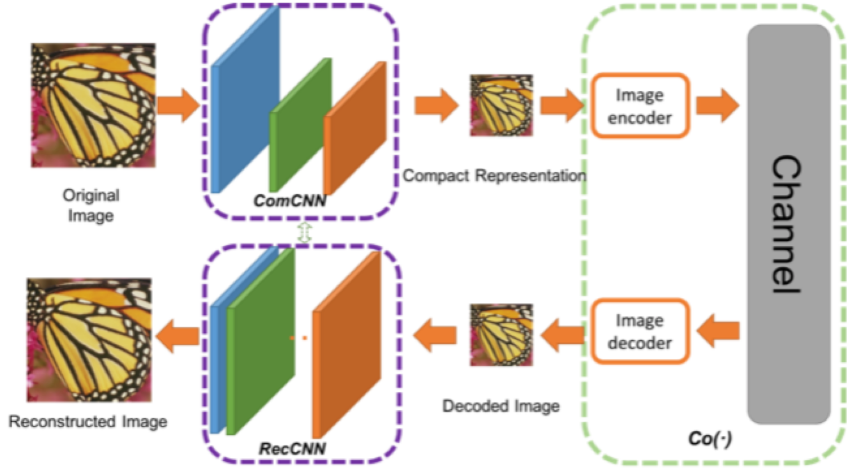
Feng Jiang, Wen Tao, Shaohui Liu, Jie Ren, Xun Guo, Debin Zhao, *Member, IEEE*

Traditional image compression algorithms

- JPEG attempt to distribute available bits
- Deblocking

New solution

- The first CNN, named ComCNN, learns an optimal compact representation from an input image, which is then encoded using an image codec (e.g., JPEG, JPEG2000 or BPG).

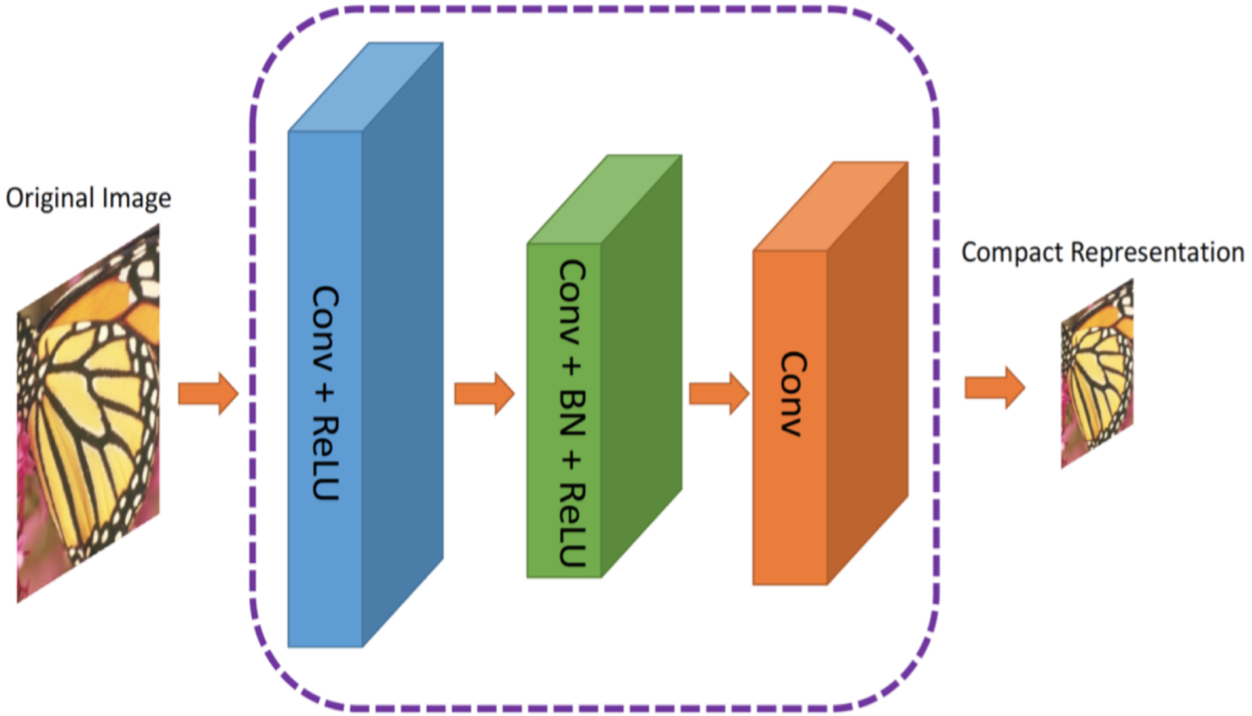


- The second CNN, named RecCNN, is used to reconstruct the decoded image with high quality in the decoding end.

Architecture of compression framework

comCNN:

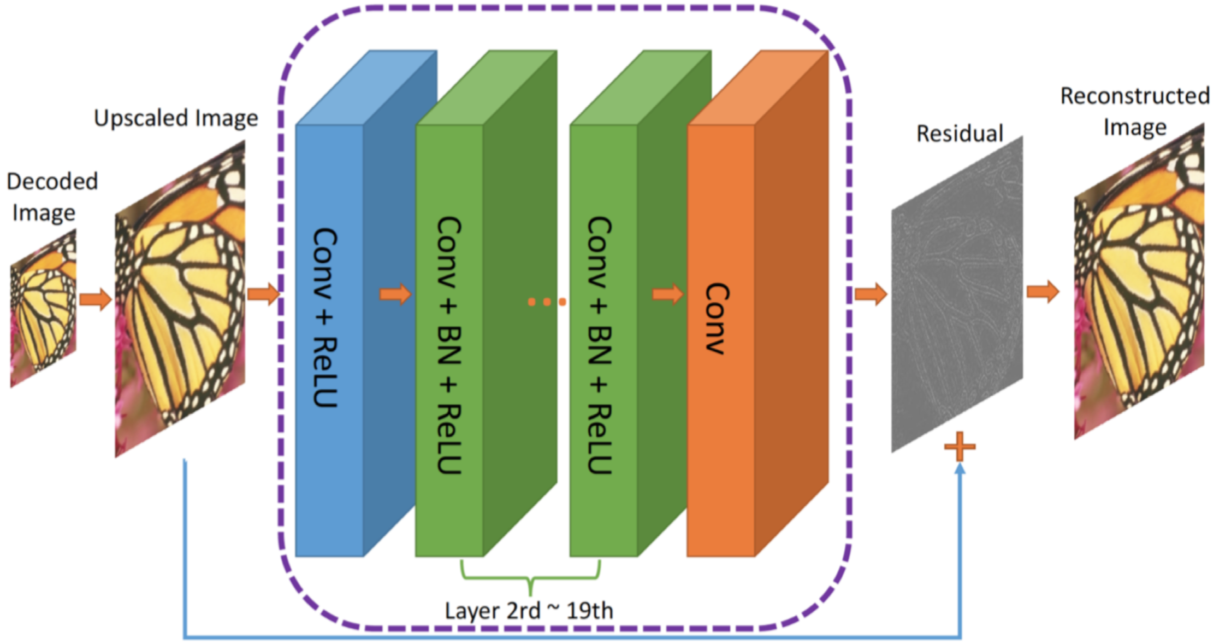
- combination of conv and ReLU
- 1st layer : 64 filters of size 3 x 3 x c with ReLU
- 2nd layer: 64 filters of size 3 x 3 x 64 with ReLU stride = 2.
- 3rd layer: c filters of size 3 x 3 x 64



Architecture of compression framework

- RecCNN :

  - 20 layers
  - **Layer 1** : 64 filters of 3 x 3 x c (Convolution + ReLU)
  - **Layer 2 to 19** : 64 filters of size 3 x 3 x 64 (Convolution + batch normalization + ReLU)
  - **Layer 20** : c filters of size 3 x 3 x 64
  - The compressed image is unsampled to the original image size using bicubic interpolation



Learning Algorithm

- Optimization goal is to find –

$$\left(\hat{\theta_1}, \hat{\theta_2}\right) = \arg\min_{\theta_1, \theta_2} \left\| Re\left(\theta_2, Co\left(Cr\left(\theta_1, x\right)\right)\right) - x \right\|^2$$

Algorithm 1 The Proposed Compression Framework for Training Sub-Networks

---

Input: The original image x
Initialization: Random initial $\hat{\theta}_1^0$ and $\hat{\theta}_2^0$
for $t = 1 \rightarrow T$ **do**
  Update $\hat{x}_m^t$ by computing Eq.(2)
  for $x_m = \hat{x}_m^t$ **do**
    Update $\hat{\theta}_2^t$ by training RecCNN to compute Eq.(5)
  end for
  for $\theta_2 = \hat{\theta}_2^t$ **do**
    Update $\hat{\theta}_1^t$ by training ComCNN to compute Eq.(13)
  end for
end for
Return: $\hat{\theta}_1^t$, $\hat{\theta}_2^t$ and $\hat{x}_m^t$

$$\hat{\theta}_1 = \arg\min_{\theta_1} \left\| Re\left(\hat{\theta}_2, Co\left(Cr\left(\theta_1, x\right)\right)\right) - x \right\|^2, \qquad (2)$$

$$\hat{\theta}_2 = \arg\min_{\theta_2} \left\| Re\left(\theta_2, \hat{x}_m\right) - x \right\|^2. \qquad (5)$$

$$\hat{\theta}_1 = \arg\min_{\theta_1} \left\| Re\left(\hat{\theta}_2, Cr\left(\theta_1, x\right)\right) - x \right\|^2. \qquad (13)$$

Loss functions (MSE)

- For comCNN:

$$L_1(\theta_1) = \frac{1}{2N} \sum_{k=1}^{N} \left\| Re\left(\hat{\theta}_2, Cr\left(\theta_1, x_k\right)\right) - x_k \right\|^2$$

- For recCNN:

$$L_2(\theta_2) = \frac{1}{2N} \sum_{k=1}^{N} \left\| res\left(Co\left(\hat{x}_{m_k}\right), \theta_2\right) - \left(Co\left(\hat{x}_{m_k}\right) - x_k\right) \right\|^2$$

Reading

- https://en.wikipedia.org/wiki/Discrete_cosine_transform