How to Exploit the Transferability of Learned Image Compression to Conventional Codecs

Supplementary Material

Jan P. Klopp
National Taiwan University
kloppjp@gmail.com

Keng-Chi Liu
Taiwan AI Labs
calvin89029@gmail.com

Liang-Gee Chen Shao-Yi Chien
National Taiwan University
lgchen@ntu.edu.tw sychien@ntu.edu.tw

1. Algorithm Listing

The training procedure is listed in Algorithm 1. Note that there is one listing for all three objectives used. Different branches, depending on the chosen objective o, determine the actual implementation for that particular objective.

Result: Filter parameters θ_F
Input: Objective $o \in \{\text{MS-SSIM}, \text{Semantic}, \text{GAN}\}$; Pretrained surrogate $g_R(\cdot;\theta_C), g_D(\cdot;\theta_C)$ from [9]; Target filter h_R, h_D; Pretrained ResNet$(\cdot)$; weights γ_{GAN}, γ_{VGG}, γ_{MSE}
Initialize filter $f_F(\cdot;\theta_F)$
Copy weights for rate estimation from g_R to f_F
if $o =$ *GAN* **then**
  Initialize GAN's discriminator $y_{\text{GAN}}(\cdot;\theta_{\text{GAN}})$
end
for *Iteration* $0 < i < N$ **do**
  $I_i \leftarrow$ Input image i
  $\hat{I}_i \leftarrow f_F(I_i;\theta_F)$ # Filter the image
  $\hat{r}_i \leftarrow g_R(\hat{I}_i;\theta_C)$ # Differentiable rate
  if $o =$ *Semantic* **then**
    $\hat{I}_{i,\text{dec}} \leftarrow h_D(\hat{I}_i;\theta_C)$
  else
    $\hat{I}_{i,\text{dec}} \leftarrow g_D(\hat{I}_i;\theta_C)$
  end
  if $o =$ *GAN* **then**
    $\hat{y} \leftarrow y_{\text{GAN}}(\hat{I}_{i,\text{dec}};\theta_{\text{GAN}})$
    if $i \mod 2 = 0$ **then**
      # Update the Filter (GAN)
      $\mathcal{L}_{\text{MSE}} \leftarrow \|\hat{I}_{i,\text{dec}} - I_i\|^2$
      $\mathcal{L}_{\text{VGG}} \leftarrow \sum_{l=1}^{L} \mathbb{E}[|f_{\text{VGG},l}(\hat{I}_{i,\text{dec}}) - f_{\text{VGG},l}(I_i)|]$
      $\mathcal{L}_{\text{GAN}} \leftarrow \mathbb{E}\left[(1-\hat{y})^2\right]$
      $\mathcal{L}_{\text{Filter}} \leftarrow \gamma_{\text{GAN}}\mathcal{L}_{\text{GAN}} + \gamma_{\text{VGG}}\mathcal{L}_{\text{VGG}} + \gamma_{\text{MSE}}\mathcal{L}_{\text{MSE}} + \hat{r}_i$
      $\theta_F \leftarrow \text{Adam}\left(\frac{\partial \mathcal{L}_{\text{Filter}}}{\partial \theta_F}\right)$
    else
      # Update Discriminator
      $y \leftarrow y_{\text{GAN}}(I_i;\theta_{\text{GAN}})$
      $\mathcal{L}_{\text{Discriminator}} \leftarrow \mathbb{E}\left[\hat{y}^2\right] + \mathbb{E}\left[(1-y)^2\right]$
      $\theta_{\text{GAN}} \leftarrow \text{Adam}\left(\frac{\partial \mathcal{L}_{\text{Discriminator}}}{\partial \theta_{\text{GAN}}}\right)$
    end
  else
    if $o =$ *Semantic* **then**
      $l_i \leftarrow$ Label i # Ground truth
      $\hat{l}_i \leftarrow \text{ResNet}(\hat{I}_{i,\text{dec}})$ # Prediction
      $\mathcal{L}_{\text{Filter}} \leftarrow \lambda_T \text{CrossEnt}(l_i, \hat{l}_i) + \hat{r}_i$
    else
      $\mathcal{L}_{\text{Filter}} \leftarrow \lambda_T \text{MSSSIM}(I_i, \hat{I}_{i,\text{dec}}) + \hat{r}_i$
    end
    $\theta_F \leftarrow \text{Adam}\left(\frac{\partial \mathcal{L}_{\text{Filter}}}{\partial \theta_F}\right)$
  end
end

Algorithm 1: Filter training procedure.