Learning to Compress Color Images and Videos

Li Cheng  S. V. N. Vishwanathan

Statistical Machine Learning Program
NICTA

International Conference on Machine Learning
Corvallis OR, June 2007
Where we are

- **Coder**
- **Decoder**
- **Internet**

Input: image/video

Output: image/video
Decoder: Colorization
Encoder: *automatically* select *representative* pixel labels.
Our ideas

- We compress by exploiting color redundancy in images (and similarly for videos). It works as follows:
  - In the decoder component, given an intensity image output from the existing decoder, and few representative color pixels, we colorize the full image.
  - In the encoder component, given a color image input, we feed its intensity image into the existing encoder, and pick the few color pixels.
- existing compression methods are still used
- How to colorize, and how to pick the representative labeled pixels?
- What if we have a few similar images but the labeled pixels are from only one image?
Colorization by Semi-supervised Learning

- We show that the algorithm by Levin et al. (2004) is a transductive learning algorithm

\[ J(f) = \frac{\lambda}{n^2} \|f\|^2_G + \frac{1}{m} \sum_{i=1}^{m} l(x_i, y_i, f), \]

where

\[ \|f\|^2_G = f^T L^2 f = \sum_{i=1}^{n} \left( f(x_i) - \sum_{i \sim j} \omega_{ij} f(x_j) \right)^2, \]

\[ \sum_{i=1}^{m} l(x_i, y_i, f) = \sum_{i=1}^{m} \delta(f(x_i), y_i). \]

- We use graph based inductive algorithm LapRLS
Graph based Semi-supervised Learning

- construct a feature graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, define $L = D - W$, $\Delta = D^{-1/2}LD^{-1/2}$
- minimize the regularized risk

$$J(f) = c ||f||^2_H + \frac{\lambda}{n^2} ||f||^2_G + \frac{1}{m} \sum_{i=1}^{m} (f(x_i) - y_i)^2. \quad (1)$$

- The regularizer $\|f\|_G^2 = f^\top \Delta f$, where $f = [f(x_1), \ldots, f(x_n)]$.
- The minimizer is $f(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$, and $\alpha = (I_m K + cm I + \frac{\lambda m}{n^2} \Delta K)^{-1} y$.
- Inductive vs. transductive
What we do

- Encoder: A simple active learning strategy to query few representative pixel colors
  1. Start with a few randomly chosen labeled pixels
  2. Learn a model
  3. Evaluate the prediction using current model, stop if meet certain criterion
  4. Identify pixels with big errors, and sample several new representations from these pixels, go to step 2

- Decoder: colorization using inductive SSL
Implementation Details

- Following Levin et al. (2004), we work in YUV space. We predict $U$, $V$ values independently.
- To construct the feature graph. For each pixel, we use a $5 \times 5$ window around it, and its spatial location to compose a feature vector.
- We use Peak Signal to Noise Ratio (PSNR) score to measure the prediction quality

$$\text{PSNR} = 20 \log_{10} \frac{255}{\sqrt{\text{MSE}}}$$  \hspace{1cm} (2)
Exp 1: Human Assisted Image Colorization

Figure: A $683 \times 512$ gray-scale image with the human labeled pixels, and our result.
Exp 2: Image Compression

Figure: An image of size $640 \times 853$. PSNR values are 27.00 for human labels vs. 31.49 for active learning, using 8558 vs. 2534 labeled pixels, respectively.
Exp 3: Image Compression

Figure: An image of size $512 \times 683$. PSNR values are 38.41 for random labels vs. 40.95 for active learning, using 2976 vs. 2766 labeled pixels, respectively.
Exp 4: Image Compression

- Compression ratio of size of (grayscale jpg + labeled pixels) / size of color jpg: 0.744 for the bees image and 0.781 for the girl image.

**Figure:** Encoder: the PSNR scores for (a) bees and (b) girl vs number of iterations of the active learning algorithm.
Exp 5: Human Assisted Video Colorization

Figure: A video of 146 frames, image size $240 \times 130$. After manually scriboring 1542 color labels on the first frame, we present results of the 1st and the 32nd frames.
Video Compression

Figure: A video of 302 frames, image size $240 \times 130$. 7005 labeled color pixels are selected to ensure $\text{PSNR} \geq 38$, compression ratio 0.899 over color video using H.264.
We present a machine learning approach that facilitates image and video compression by exploiting color redundancy.

A reward for forgetting labels?