Image processing background

Monday, November 4, 2024 6:13 AM

Image Processing

- Intersection of classic ML w/ SciML
  eg. medical imaging, tomography, astronomy, microscopy ...
  ultrasound, MRI ...

- Many tasks are linked, eg. compression
  denoising
  inverse problems
  which we'll explore

Image (lossy) Compression: JPEG (1992)

Rough idea: ① split image into 8×8 patches, and from now on, operate on patches separately
(this guarantees linear complexity and makes it fast)

② perform a 2D (8×8) DCT (Discrete Cosine Transform) to transform to frequency space
(no compression/loss yet)
(DCT via integer arithmetic!!)

③ set small coefficients to 0 (user defined threshold)

④ possibly quantize remaining coefficients
possibly also entropy-encode them] lossless
"arithmetic code"
save these to file

Why the DCT to "induce sparsity"?

to paraphrase George Box, "All (image) models are wrong, some are useful"

Let's go back even further

How to model compression:

Suppose we wish to compress $\{X_i\}_{i=1}^{n}$, $X_i \in \mathcal{X}$
discrete set of symbols like $\{a, b, c, \dots, z\}$

and we assume ① these are random,
② these are iid, w/ probability $p(X)$

ex: $p(X=\text{"a"}) \approx \frac{1}{26}$

$p(X=\text{"e"}) \approx 1/10$

$p(X=\text{"z"}) \approx 1/50$

Image processing background (p. 2)

Monday, November 4, 2024 6:30 AM

let our alphabet $\mathcal{X}$ have m symbols, so we can characterize $x_i \in \mathcal{X}$ using an integer $0, 1, \ldots, m-1$ (eg. $0, 1, \ldots, 25$)

Baseline:

encode this integer directly. Requires $\lceil \log_2(m) \rceil$ bits per symbol. To send a message $(x_1, x_2, \ldots, x_n)$ we need $n \cdot \lceil \log_2(m) \rceil$ bits, and the average per symbol is $\lceil \log_2(m) \rceil$

Doing better:

If $p(X = \text{"z"}) = 0$, no need to encode it. If it's almost 0, can we exploit? Yes!

Aside:

Entropy of a distribution p, with $X \sim p$ a r.v., is
$$H(X) := -\sum_{x \in \mathcal{X}} p(x) \log_2(p(x))$$

Ex: if p is uniform (aka uninformative) over $\mathcal{X} = \{1, 2, \ldots, m\}$ then $p(x) = \frac{1}{m} \quad \forall x \in \mathcal{X}$,
$$H(X) = \sum p(x) \cdot \log_2\left(\frac{1}{p(x)}\right) = \sum \frac{1}{m} \log_2(m)$$
$$= \log_2(m)$$
Same as baseline

All other distributions have lower entropy.

We can encode w/ **Huffman codes** or other entropy codes.

Theorem (Shannon)

It is possible to encode $(x_1, \ldots, x_n)$ (if iid) using $n \cdot (H(X) + \varepsilon)$ bits, for any $\varepsilon > 0$, and recover it w/ probability $1 - \varepsilon$.

Image processing background (p. 3)

Monday, November 4, 2024 6:40 AM

Problem What if $\{X_1, \ldots, X_n\}$ aren't iid?

Ex: $\mathcal{X} = \{a, b, \ldots, z\}$

message "Helloworld"

X_2 is not independent of X_1 — think of wordle

(it's more likely to be a vowel)

So, entropy encoding is inefficient

Strategy 1: larger messages, $\mathcal{X} = \{\text{all words}\}$

intractable

Strategy 2: make it iid

↑ actually impossible... but we can whiten it

(i.e. make the correlations zero)

Karhunen-Loève transformation

like PCA but when we treat data as random.

Ex

Suppose $\mathrm{Cov}(X_i, X_j) = \sigma^2 \cdot \rho^{|i-j|}$ for some $\rho \in (-1, 1)$

then the DCT is the KL transform.

should be familiar to those of you who took timeseries

So...

If X_i is value of image at pixel i,

and $\mathrm{Cov}(X_i, X_j) \approx \rho^{|i-j|}$ (and extend to 2D)

then DCT whitens our signal, so we can better exploit entropy encoding.

*.

Image processing background (p. 4)

Monday, November 4, 2024 8:15 AM

Evaluation metrics

- The OG: Mean Squared Error (MSE)

$$MSE = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} (\hat{X}_{ij} - X_{ij})^2$$

i,jth pixel

MSE ≥ 0
lower better

- well understood
- many nice mathematical properties

- Variant: Peak Signal-to-Noise Ratio (PSNR)

$$PSNR = 10 \cdot \log_{10} \left(\frac{max^2}{MSE} \right)$$

max = 255 for 8-bit images

PSNR $\in \mathbb{R}$
higher better
28dB+ good

in decibels (dB) a relative scale
10·log(power units)
20 log (amplitude units)

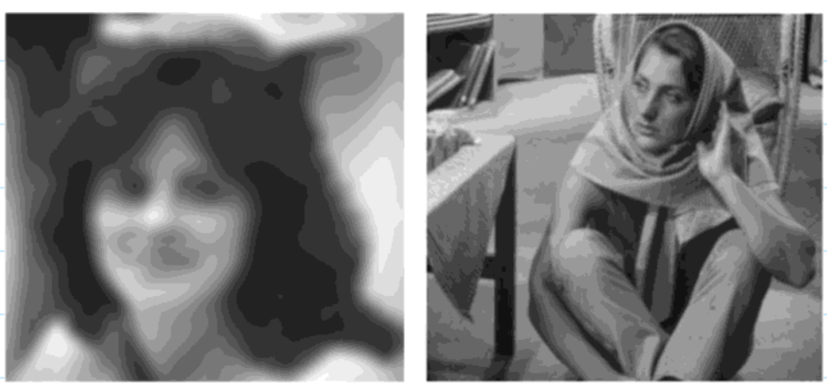
- Structural Similarity Index Measure (SSIM) '01, '04

Wang et al.
50k+ citations

Also has an easy-to-use formula ... but attempts to align w/ human perception

SSIM $\in (0,1]$
near 1 is better

ex: Failure of PSNR



PSNR=25.11 dB, IQA=0.0292   PSNR=25.12 dB, IQA=0.5574

Bezhadpour and Ghanbari, 2021,
https://www.researchgate.net/figure/Subjective-quality-of-two-different-contents-with-the-same-PSNR_fig2_362323114

SSIM attempts to capture luminance masking and feature masking
... like MP3's

- Learned Perceptual Image Patch Similarity (LPIPS)

lower better

Uses a trained neural net