Part 1 Media and Systems

2024年4月28日 23:47

EE6403 Lecture Part 1 AY2324

color space: RGB, HSI, CMY, YUV, YCbCr

Part 2 Compression

2024年4月29日 20:02

EE6403 Lecture Part 2 AY2324

Text compression:

Huffman coding: VLC, average number of bits reduced.

Solution to Exercise 2-1 (Text)

Solution to Exercise 2-2 (Huffman) becareful compression ratio, find most original bits.

Image Video Compression:

Human is less sensitive to noise or distortion in high frequency components and vice versa

Human is more sensitive to luminance (brightness) components than chrominance (color) components.

Quainter: An irreversible many-to-one mapping, causing information loss.

DCT

$$S_{uv} = \alpha(u)\alpha(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij} \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N} \qquad u, v = 0, \ldots, N-1$$

$$\alpha(k) = \begin{cases} \sqrt{\dfrac{1}{N}} & \text{for } k = 0 \\ \sqrt{\dfrac{2}{N}} & \text{for } k = 1, 2, \ldots, N-1 \end{cases}$$

It can offer the following:

Energy compaction for transform coefficients

Redundancy reduction amongst transform coefficients

Pro: good compression results, basis functions are fixed and not image-dependent.

Con: compression is not as effective as some other transforms, e.g., Karhunen Loeve Transform.

when trying IDCT, formula can be a Matrix, it is so called basis function, it is the fundamental bricks of picture.

matrix implementation

$$F(u, v) = \mathbf{T} \cdot f(i, j) \cdot \mathbf{T}^T.$$

We will name $\mathbf{T}$ the *DCT-matrix*.

$$\mathbf{T}[i, j] = \begin{cases} \frac{1}{\sqrt{N}}, & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cdot \cos\frac{(2j+1)\cdot i\pi}{\ldots} & \text{if } i > 0 \end{cases}$$

$$F(u,v) = \mathbf{T} \cdot f(i,j) \cdot \mathbf{T}^T.$$

We will name $\mathbf{T}$ the *DCT-matrix*.

$$\mathbf{T}[i,j] = \begin{cases} \frac{1}{\sqrt{N}}, & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cdot \cos\frac{(2j+1)\cdot i\pi}{2N}, & \text{if } i > 0 \end{cases}$$

The 2D IDCT matrix implementation is simply:

$$f(i,j) = \mathbf{T}^T \cdot F(u,v) \cdot \mathbf{T}.$$

The DCT-matrix is orthogonal, hence,

$$\mathbf{T}^T = \mathbf{T}^{-1}.$$

solution to Exercise 2-3 (2D-DCT)(1)

KLT

PCA,EVD

same to DCT, compact energy, decorrelation, dim reduction,

but, image dependent, eigen decomposition.

$$\boldsymbol{C} = \frac{1}{N_T - 1}\sum_{i=1}^{N_T}(\boldsymbol{x}_{ni} - \bar{\boldsymbol{x}}_n)(\boldsymbol{x}_{ni} - \bar{\boldsymbol{x}}_n)^T$$

$$\bar{\boldsymbol{x}}_n = \frac{1}{N_T}\sum_{i=1}^{N_T}\boldsymbol{x}_{ni}$$

when reconstructing, left multiply the axis transpose to get a scalar that is coordinate.

- Synthesis

$$\boldsymbol{x} = \sum_{i=1}^{M} c_i \boldsymbol{p}_i$$

- Analysis

$$c_i = \boldsymbol{p}_i^T \boldsymbol{x}, \qquad i = 1,\ 2,\ \ldots,\ M$$

- Note:

$$E[c_i c_j] = \begin{cases} \lambda_i & i = j \\ 0 & i \neq j \end{cases}$$

solution to Exercise 2-4 (KLT)

Differential/ Predictive coding:

Vector Quantization:

solution to Exercise 2-5 (VQ)

SVD:

- SVD of an mxn matrix $\mathbf{A}$:

$$\mathbf{A} = \mathbf{U\Sigma V}^T$$

U is an mxm orthogonal matrix

V is an nxn orthogonal matrix

$\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$, an $m \times n$ matrix

with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$ and r is the rank of the matrix

U = normalized(decreasing(eigen(AA^T)))

Σ = diag(decreasing(sqrt(λ)))

V = normalized(decreasing(eigen(A^TA)))

reconstruct by:

$$\hat{\mathbf{A}} = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \qquad \text{where } p < r$$

solution to Exercise 2-6 (SVD)
solution to Exercise 2-7 (SVD)

JPEG:

JPEG Encoder

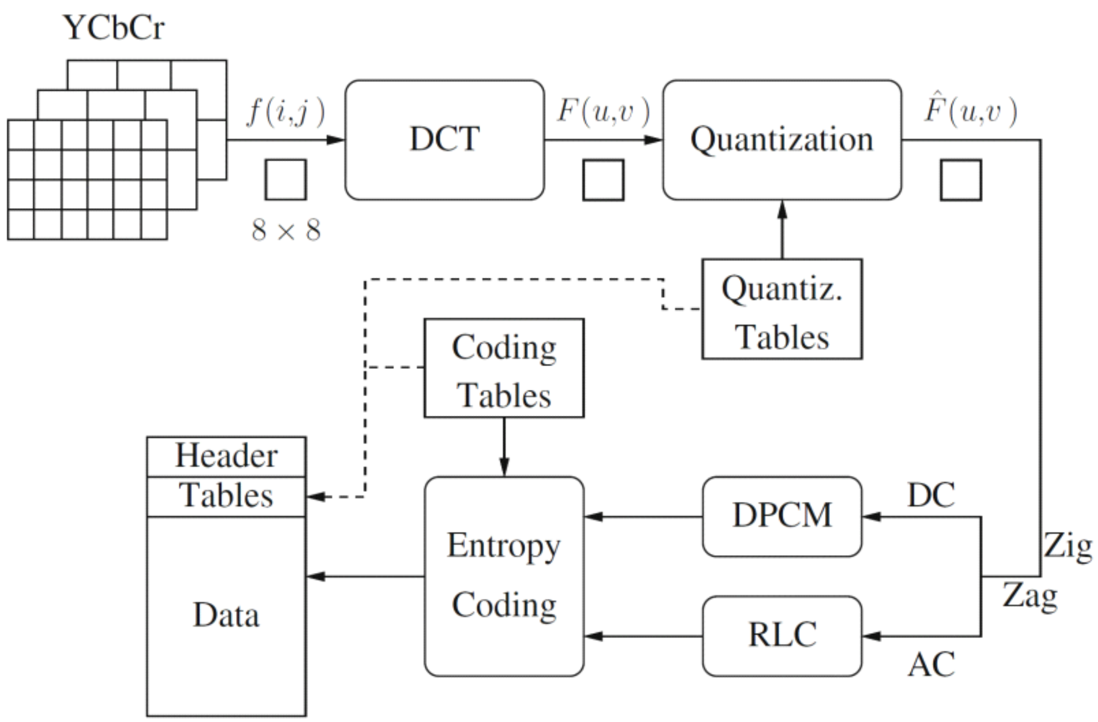

Fig. 9.1: Block diagram for JPEG encoder.

Differential coding is used as average intensity between 2 consecutive blocks is similar.

solution to Exercise 2-8 (JPEG)

Part 3 Video Compression

2024年4月29日 20:02

EE6403 Lecture Part 3 AY2324
Part3

Part 4 Media Trans/ QoS

2024年4月29日 20:03

EE6403 Lecture Part 4 AY2324
Part4.135

Part 5 Multimedia App

2024年4月29日 20:03

[EE6403 Lecture Part 5 AY2324]

Benchmark Datasets:

MNIST: handwritten digits, many years ago.
CIFAR-10, CIFAR-100
ImageNet: 1000 cls
Coco
Google Open Image.
LAION: vision and text.

DNN:

CNN, RNN, Transformer, GAN, GNN, Diffusion, LLM

Linear Classifiers:

Multilayer perception: MLP. FF, dense network
loss func:

- Square loss:

$$L(x,y) = \sum_i (y_i - f(x_i))^2$$

- Mean Square Error (MSE):

$$MSE = \frac{1}{N}\sum_i (y_i - f(x_i))^2$$

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_i |y_i - f(x_i)|$$

- Softmax loss:
  - Cross-entropy loss with Softmax normalization.

$$p_j = \frac{e^{z_j}}{\sum_k e^{z_k}}, \text{ where } z_j = f(x_j)$$

$$L = -\sum_j y_j \log_e p_j$$

natural log namely ln.

CNN:

- With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.

[Solution to Part 5 Exercise (CNN)]

CNN Training/ Optimization:

hyperparameters: learning rate.
optimizer: SGD, Adam

Architecture:

Alexnet
VGG
GoogleNet
ResNet

DenseNet
SENet
EfficientNet

Key metric: Acc, mem, flops

Transfer Learning:

Solution to Part 5 Exercise (Transfer Learning)

APP:

image captioning: CNN->LSTM

Emerging:

transformer
ViT: partition, flatten, embedding, learnable class embedding, encoder, MLP

GAI:

GAN:
Stable Diffusion: