Convolutional DCT Image Super-resolution

Progress Report & Next Step

Department of Electrical Engineering

Pennsylvania State University

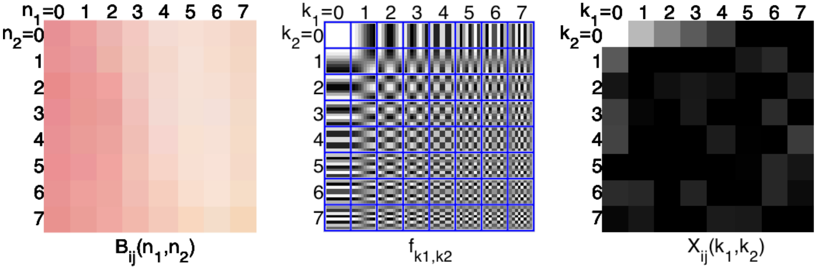
Information Processing
& Algorithms Laboratory
Everything can be optimized

2018

Outline

1. Convolutional DCT Network
   - 2D DCT and representation process
   - 2D DCT and 2D IDCT by transpose convolutional neural network
   - Orthogonality constrains
   - Training process
2. Preliminary Results
   - SSIM, PSNR, IFC
   - CDCT layer: threshold and orthogonality
3. Next Step
   - Complexity order constrains
   - De-noising and SR

2D DCT & Representation

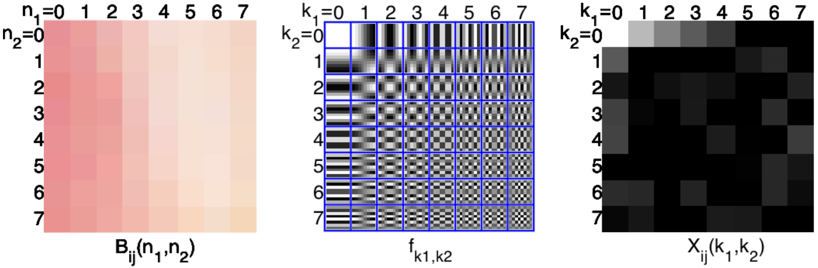


Compute DCT coefficients block X_{ij} for x_{ij}

$$X_{ij}(k_1,k_2) = \sum_{n_2=0}^{N-1} \sum_{n_1=0}^{N-1} x_{ij}(n_1,n_2) \times f_{k_1,k_2}(n_1,n_2)$$

Note that x_{ij} and X_{ij} is the same size 8×8.
Coefficient $X_{ij}(k_1,k_2)$ represents the significance of basis $f_{k_1k_2}$ embedded in x_{ij}.

2D IDCT & Representation

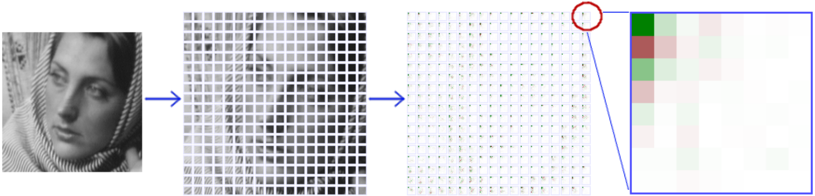


Compute $x_{(i,j)}$ from DCT coefficients block X_{ij}

$$x_{ij}(n_1, n_2) = \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{N-1} X_{ij}(k_1, k_2) \times f_{k_1,k_2}(n_1, n_2)$$

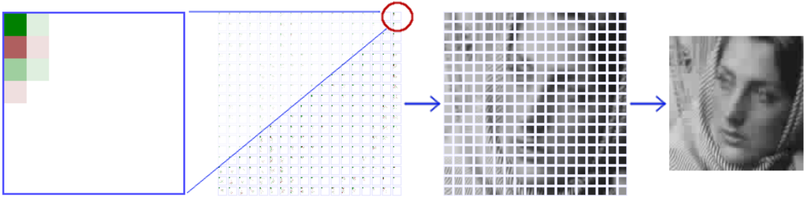
Coefficient $X_{ij}(k_1, k_2)$ represents the significance of basis $f_{k_1 k_2}$ embedded in x_{ij}.

2D DCT & IDCT



Original image *Pixel blocks* *DCT coefficient blocks* *Single coefficient block*

Process the whole image and inverse procedure:

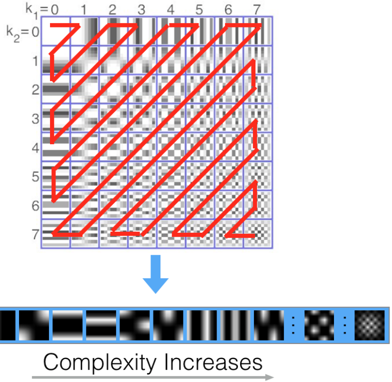


Single quantized block *Quantized coefficient blocks* *Reconstructed pixel blocks* *Reconstructed image*

Each block only contains its own spatial information.

DCT by neural network

- Treat DCT basis functions as filters and organize in zig-zag order:



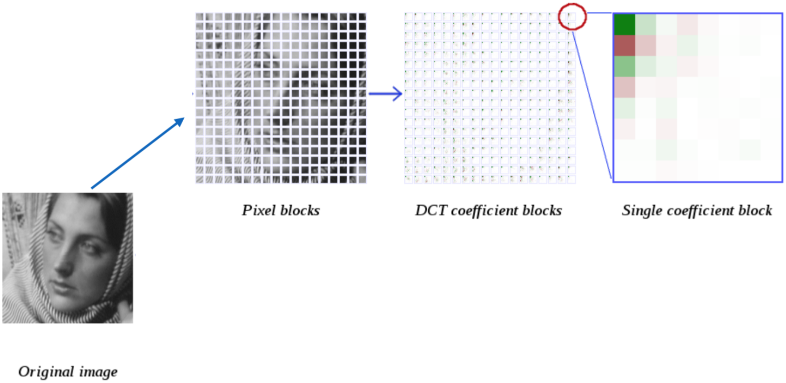
- Re-index $f_{k1,k2}$ to f_k with zig-zag mapping function Zig:
  $Zig\{(k_1, k_2)\} = k$ where
  $(k_1, k_2) \in (0, \ldots, N-1) \times (0, \ldots, N-1) \rightarrow k \in (1, \ldots, N \times N)$
- $Zig\{(k_1, k_2)\} = k$ and $Zig^{-1}\{k\} = (k_1, k_2)$
- Now here are $\{f_k\}_{k=1}^{N \times N}$ DCT basis filters, each of size $N \times N$

DCT by neural network

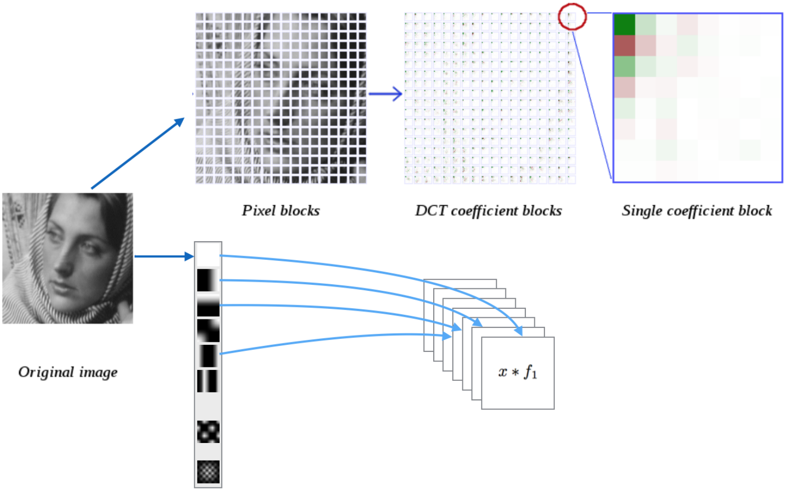
Convolve image $x \in \mathbb{R}^{(W \times H)}$ with $\{f_k\}_{k=1}^{N \times N}$ DCT basis filters:

- Convolve without overlapping, with shift of N.
- For a fixed k, $x * f_k$ gives DCT coefficients $X_{ij}(k)$ of all x's x_{ij}
- $x * f_k = X_k$, where $X_k \in \mathbb{R}^{\frac{W}{N} \times \frac{H}{N}}$
- As k increases, smaller details are captured by $x * f_1$

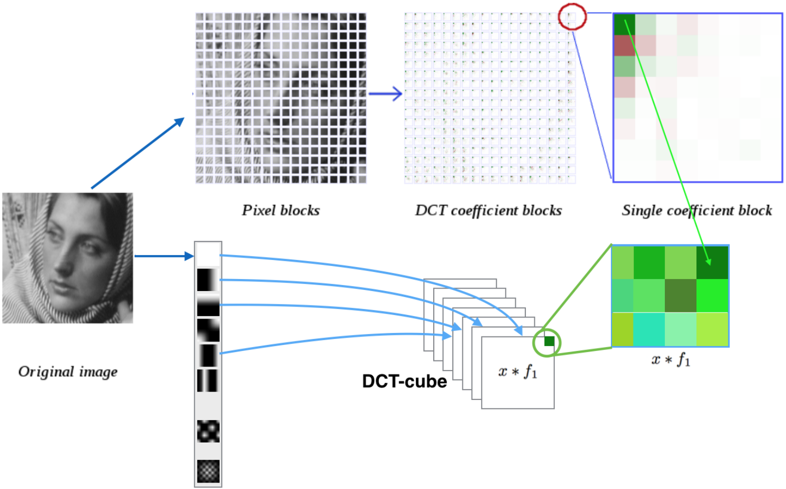
DCT by neural network



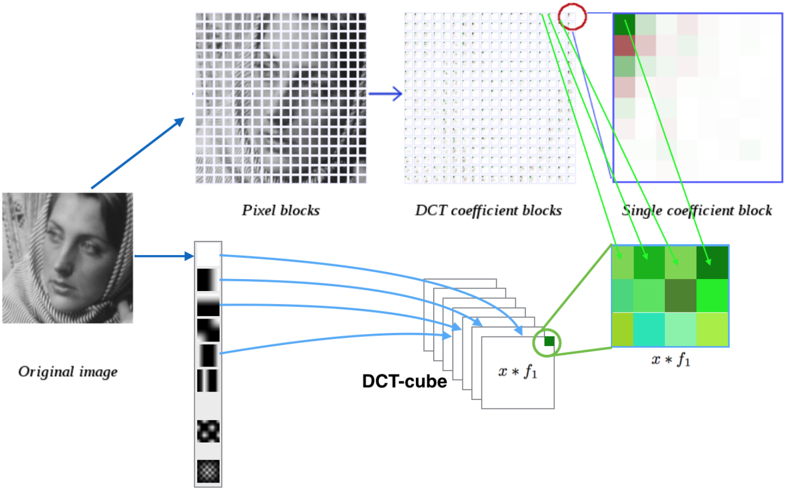
DCT by neural network



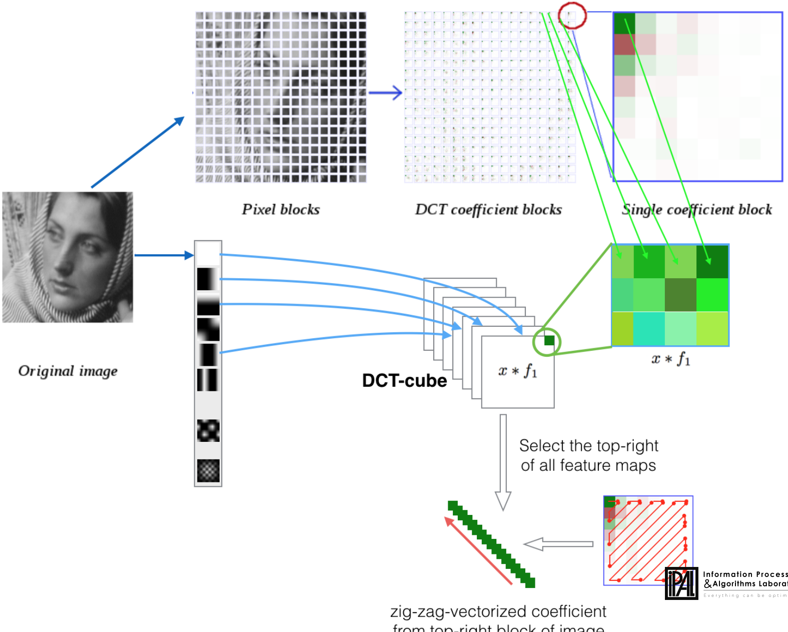
DCT by neural network



DCT by neural network



DCT by neural network



DCT by neural network: proof

Prove: DCT by neural network generates the same DCT coefficients X_{ij} for x_{ij}, ij denotes one 8×8 block

For a fixed block ij, $k_1, k_2, n_1, n_2 = 1, \dots, N$

DCT:

$$X_{ij}(k_1, k_2) = \sum_{n_2=0}^{N-1} \sum_{n_1=0}^{N-1} x_{ij}(n_1, n_2) \times f_{k_1,k_2}(n_1, n_2)$$

DCT by neural network:

- $X(i,j)_k = \sum_{n_2=i\times N}^{(i+1)\times N} \sum_{n_1=j\times N}^{(j+1)\times N} x(n_1, n_2) \times f_k(n_1, n_2)$
- For fixed ij, the $X(i,j)_k$ can be re-indexed as: $X_{ij}(k) = \sum_{n_2=0}^{N} \sum_{0}^{N} x_{ij}(n_1, n_2) \times f_k(n_1, n_2)$
- Since $Zig^{-1}(f_k) = f_{k_1,k_2}$, we have:

$$\begin{aligned} Zig^{-1}(X_{ij}(k)) &= \sum_{n_2=0}^{N} \sum_{n_1=0}^{N} x_{ij}(n_1, n_2) \times Zig^{-1}(f_k(n_1, n_2)) \\ &= \sum_{n_2=0}^{N} \sum_{n_1=0}^{N} x_{ij}(n_1, n_2) \times f_{k_1,k_2}(n_1, n_2) = X_{ij}(k_1, k_2) \end{aligned}$$

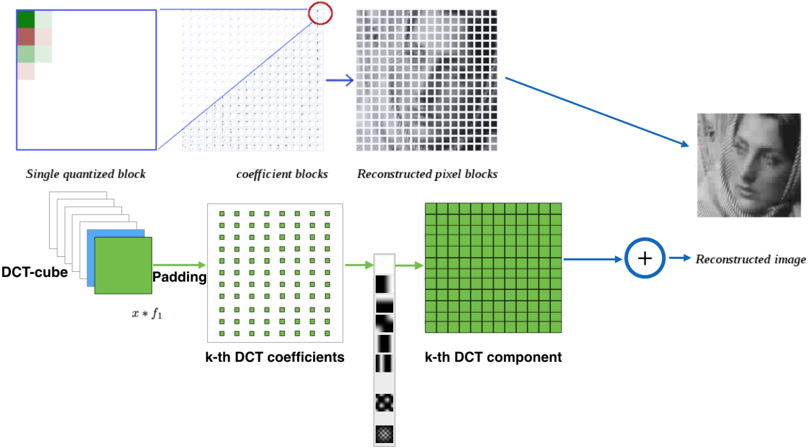
Thus, DCT by neural network generates zig-zag arranged blocked DCT.

Inverse DCT by neural network

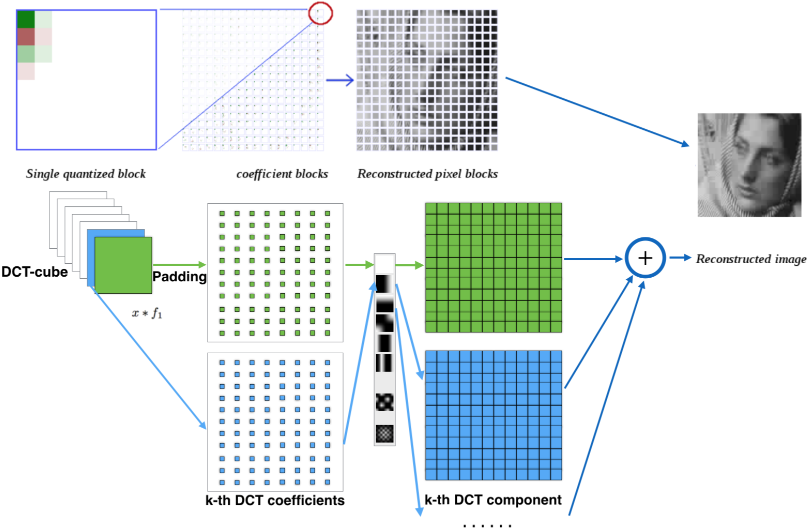
Transpose-convolve features $X \in \mathbb{R}^{N^2 \times \frac{W}{N} \times \frac{H}{N}}$ with $\{f_k\}_{k=1}^{N \times N}$ DCT basis:

- Padding: $\bar{X}_k(i,j) = \begin{cases} X_k(k,l), & \text{if } i = 8k \text{ and } j = 8l \\ 0, & \text{Otherwise} \end{cases}$, $k = 1, \dots \frac{W}{N}, l = 1, \dots \frac{H}{N}$,
- Transpose convolve: convolution with shifting 1
- For a fixed k, $\bar{X}_k * f_k$ gives all x_{ij}'s k-th spatial component
- $\bar{X}_k * f_k = x_k$, where $x_k \in \mathbb{R}^{W \times H}$
- The final results: $x = \sum_{k=1}^{N^2} \bar{X}_k * f_k$

Inverse DCT by neural network



Inverse DCT by neural network



iDCT by neural network: proof

Prove: iDCT by neural network generates the same x_{ij} from DCT coefficients X_{ij}, ij denotes one 8×8 block
For a fixed block ij, $k_1, k_2, n_1, n_2 = 1, \ldots, N$
iDCT:

$$x_{ij}(n_1, n_2) = \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{N-1} X_{ij}(k_1, k_2) \times f_{k_1,k_2}(n_1, n_2)$$

iDCT by neural network:

- $x_{ij}(n_1, n_2) = \sum_{k=1}^{N \times N} \sum_{l=i \times N}^{(i+1) \times N} \sum_{m=j \times N}^{(j+1) \times N} \bar{X}_k(i \times N + n_1 - l, j \times N + n_2 - m) \times f_k(l, m)$
- $\bar{X}_k(i \times N + n_1 - l, j \times N + n_2 - m) \neq 0$ while $n_1 - l + i \times N = i \times N$, reordering the index we have: $x_{ij}(n_1, n_2) = \sum_{k}^{N \times N} X_k(i, j) \times f_k(n_1, n_2)$

iDCT by neural network: proof

Prove: iDCT by neural network generates the same x_{ij} from DCT coefficients X_{ij}, ij denotes one 8×8 block

For a fixed block ij, $k_1, k_2, n_1, n_2 = 1, \ldots, N$

iDCT:

$$x_{ij}(n_1, n_2) = \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{N-1} X_{ij}(k_1, k_2) \times f_{k_1,k_2}(n_1, n_2)$$

iDCT by neural network:

- Since $Zig^{-1}(f_k) = f_{k_1,k_2}$ and $Zig^{-1}(X_k) = X_{k_1,k_2}$, we have:

$$\begin{aligned} x_{ij}(n_1, n_2) &= \sum_{k}^{N \times N} X_k(i, j) \times f_k(n_1, n_2) \\ &= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} X_{k_1,k_2}(i, j) \times f_{k_1,k_2}(n_1, n_2) \\ &= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} X_{ij}(k_1, k_2) \times f_{k_1,k_2}(n_1, n_2) \end{aligned}$$

- iDCT by neural network generates same $x_{ij}(n_1, n_2)$ for a given block.

Information Processing
& Algorithms Laboratory
Everything can be optimized

visit us at: `http://signal.ee.psu.edu/`