Learning-Based Image/Video Coding

Lu Yu
Zhejiang University
Outlines

- **System architecture of learning based image/video coding**
  - Learning based modules embedded into traditional hybrid coding frameworks
    - In-loop filter, Intra prediction, Inter prediction, Entropy coding, etc.
    - Transform, quantization
    - Encoder optimization
  - End-to-end image and video coding

- **Coding for human vision vs. coding for machine intelligence**
Theory of Source Coding and Hybrid Coding Framework

- Two threads of image/video coding
  - Characteristics of source signal
    - Spatial-temporal correlation
      - Intra and inter prediction
      - Transform
    - Statistical correlation
      - Symbols: stationary random process
      - Entropy coding
  - Characteristics of human vision
    - Limited sensitivity
      - Quantization
- Balance between cost and performance
  - Rate-distortion theory
In-Loop Filter

Filtering

- **Network input**
  - Current compressed frame

- **Network output**
  - Filtered frame

- **Network structure**
  - 22-layer CNN with inception structure

- **Integration into coding system**
  - Same model for Luma and chroma component
  - Different model for different QP
  - For I-frame: replace Deblocking filter (DB) and Sample Adaptive Offset (SAO)
  - For B/P-frame: added between DB and SAO, switchable at CTU-level

- **Performance (anchor: HM16.0)**

<table>
<thead>
<tr>
<th>Class</th>
<th>All-Intra</th>
<th>Low-Delay B</th>
<th>Random-Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y (%)</td>
<td>U (%)</td>
<td>V (%)</td>
<td>Y (%)</td>
</tr>
<tr>
<td>Class A</td>
<td>-5.4</td>
<td>-6.2</td>
<td>-5.5</td>
</tr>
<tr>
<td>Class B</td>
<td>-7.3</td>
<td>-7.5</td>
<td>-9.0</td>
</tr>
<tr>
<td>Class C</td>
<td>-9.9</td>
<td>-10.4</td>
<td>-13.4</td>
</tr>
<tr>
<td>Class D</td>
<td>-10.0</td>
<td>-10.4</td>
<td>-13.4</td>
</tr>
<tr>
<td>Class E</td>
<td>-13.4</td>
<td>-10.0</td>
<td>-9.5</td>
</tr>
<tr>
<td>Overall</td>
<td>-9.2</td>
<td>-8.9</td>
<td>-10.2</td>
</tr>
<tr>
<td>Enc. Time</td>
<td>47.0%</td>
<td>1818%</td>
<td>1835%</td>
</tr>
<tr>
<td>Dec. Time</td>
<td>267686%</td>
<td>756074%</td>
<td>727860%</td>
</tr>
</tbody>
</table>

In-Loop Filter

Filtering with spatial and temporal information

- **Network input**
  - Current compressed frame
  - Previous reconstructed frame

- **Network output**
  - Filtered frame

- **Network structure**
  - 4-layer CNN

- **Integration into coding system**
  - Same model for Luma and chroma component
  - Different model for different QP
  - Used in I/P/B frames
  - After DB and SAO
  - Switchable at CTU-level

- **Performance** (anchor: RA, HM16.15)

In-Loop Filter

Filtering with quantization information

- **Network input**
  - Current compressed frame
  - Normalized QP map

- **Network output**
  - Filtered frame

- **Network structure**
  - 8-layer CNN

- **Integration into coding system**
  - Same model for Luma and chroma component
  - Same model for all QPs
  - Replace bi-lateral filter, DB and SAO, and before ALF
  - Only used on I frames
  - No RDO

- **Network compression**
  - Pruning:
    - Operate during training
    - Filters pruned based on absolute value of the scale parameter in its corresponding BN layer
  - Loss function: additional regularizers for efficient compression
  - Low rank approximation:
    - Operate after pruning
  - Dynamic fixed point adoption

- **Performance** (anchor: RA, JEM7.0)

---

In-Loop Filter

Filtering with high-frequency information

- **Network input**
  - Current compressed frame
  - Reconstructed residual values

- **Network output**
  - Filtered frame

- **Network structure**
  - 4-layer CNN

Integration into coding system

- Same model for Luma and chroma component
- Different model for different QP
- Replace DB and SAO
- Only used on I frames
- No RDO

Performance (anchor: HM16.15)

In-Loop Filter

Filtering with block partition information

- **Network input**
  - Current compressed frame
  - Block partition information: CU size

- **Network output**
  - Filtered frame

- **Network structure**
  - deep CNN

- **Integration into coding system**
  - Different model for different video content in an Exhaustive search way
  - Different model for different QP
  - Used on I/P/B frames
  - After DB and SAO
  - CTU-level switchable

- **Performance** (anchor: HM16.0)

---

In-Loop Filter

- **Content adaptive filtering**
  - Filtering for reconstructed pixels
    - Inserted into diff. position of in-loop filtering chain: deblocking $\rightarrow$ SAO $\rightarrow$ ALF
    - Replace some filters in the chain
  - Information utilized
    - Reconstructed pixels in current frame
    - Temporal neighboring pixels
    - QP map, blocksize, prediction residuals, ...
  - Network
    - From 4-layer to deep
Spatial-Temporal Prediction: Intra

Network input
- 8x8 PU and its three nearest 8x8 reconstruction blocks

Network output
- Refined PU

Network Structure: composed of 10 weight layers
- Conv+ ReLU: for the first layer, 64 filters of size 3x3x\(c\)
- Conv + BN + ReLU: for layers 2 ~ 9, 64 filters of size 3x3x64
- Conv: for the last layer, \(c\) filters of size 3x3x64

\(*c: c\) represents the number of image channels

Integration into coding system
- Replace all existing intra modes
- Fixed block size

Performance (anchor: AI, HM14.0)

<table>
<thead>
<tr>
<th>Sequences</th>
<th>BD-rate</th>
<th>Sequences</th>
<th>BD-rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic</td>
<td>-0.9%</td>
<td>PartyScene</td>
<td>-0.5%</td>
</tr>
<tr>
<td>PeopleOnStreet</td>
<td>-1.2%</td>
<td>RaceHorses</td>
<td>-0.7%</td>
</tr>
<tr>
<td>Kimono</td>
<td>-0.2%</td>
<td>BasketballPass</td>
<td>-0.4%</td>
</tr>
<tr>
<td>ParkScene</td>
<td>-0.8%</td>
<td>BQSquare</td>
<td>-0.1%</td>
</tr>
<tr>
<td>Cactus</td>
<td>-0.8%</td>
<td>BlowingBubbles</td>
<td>-0.7%</td>
</tr>
<tr>
<td>BasketballDrive</td>
<td>-0.6%</td>
<td>RaceHorses</td>
<td>-0.7%</td>
</tr>
<tr>
<td>BQTerrace</td>
<td>-0.8%</td>
<td>FourPeople</td>
<td>-0.3%</td>
</tr>
<tr>
<td>BasketballDrill</td>
<td>-0.5%</td>
<td>Johnny</td>
<td>-1.0%</td>
</tr>
<tr>
<td>BQMall</td>
<td>-0.6%</td>
<td>KristenAndSara</td>
<td>-0.8%</td>
</tr>
<tr>
<td>All average</td>
<td>-0.70%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Spatial-Temporal Prediction: Intra

Prediction Block Generation Using CNN

- **Network input**
  - 8 rows and 8 columns reference pixels

- **Network output**
  - prediction block

- **Network Structure:**
  - 4 fully connected networks with PReLU

Integration into coding system

- As an additional intra mode
- CU-level selective
- Different models for all TU size in HEVC : 4x4, 8x8, 16x16, 32x32

Performance (anchor: Al, HM16.9)

<table>
<thead>
<tr>
<th>Sequence</th>
<th>IPFCN-D</th>
<th>IPFCN-S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small QPs</td>
<td>Normal QPs</td>
</tr>
<tr>
<td>Class A (4K)</td>
<td>-2.2%</td>
<td>-4.2%</td>
</tr>
<tr>
<td>Class B (4K)</td>
<td>-1.9%</td>
<td>-3.1%</td>
</tr>
<tr>
<td>Class C (WYOGA)</td>
<td>-1.1%</td>
<td>-2.1%</td>
</tr>
<tr>
<td>Class D (WQGA)</td>
<td>-0.9%</td>
<td>-1.6%</td>
</tr>
<tr>
<td>Class E (TCP)</td>
<td>-2.3%</td>
<td>-4.2%</td>
</tr>
<tr>
<td>Average of All Classes</td>
<td>-1.1%</td>
<td>-3.6%</td>
</tr>
</tbody>
</table>

IPFCN-D: different model for angular intra modes and non-angular intra modes, respectively

IPFCN-S: same model for angular intra modes and non-angular intra modes

---

Spatial-Temporal Prediction: Intra

Prediction Block Generation Using RNN

- **Network input**
  - neighboring reconstructed pixels and current PU

- **Network output**
  - prediction block

- **Training strategy:**
  - Loss Function: MSE/SATD

**Network Structure:**

- **Overall structure:** CNN + RNN
  - using CNN to extract local features of the input context block and transform the image to feature space.
  - using PS-RNN units to generate the prediction of the feature vectors.

**PS-RNN:**

- Structure of a PS-RNN unit. It splits a stack of feature maps into vertical and horizontal planes. Each plane represents a feature map of a vertical line or a horizontal line in the original gray-scale image. After the progressive prediction, these planes are concatenated to reconstruct the feature maps. A convolutional layer is used to fuse the predictions from the vertical and horizontal feature maps.

---

Spatial-Temporal Prediction: Intra

Prediction block generation using RNN

- Performance (anchor: AI, HM16.15)

<table>
<thead>
<tr>
<th>Class</th>
<th>Sequence</th>
<th>PS-RNN-SATD</th>
<th>PS-RNN-MSE</th>
<th>FC-SATD</th>
<th>Li [16]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>Traffic</td>
<td>-3.8%</td>
<td>-2.3%</td>
<td>-3.1%</td>
<td>-1.0%</td>
</tr>
<tr>
<td></td>
<td>PeopleOnStreet</td>
<td>-3.8%</td>
<td>-2.2%</td>
<td>-3.1%</td>
<td>-1.3%</td>
</tr>
<tr>
<td></td>
<td>Nebula(10bit)</td>
<td>-1.9%</td>
<td>-1.9%</td>
<td>-1.9%</td>
<td>-1.6%</td>
</tr>
<tr>
<td></td>
<td>Steaml compromised(10bit)</td>
<td>-3.2%</td>
<td>-2.8%</td>
<td>-3.2%</td>
<td>-1.7%</td>
</tr>
<tr>
<td></td>
<td>Class A Average</td>
<td>-3.2%</td>
<td>-2.3%</td>
<td>-2.8%</td>
<td>-1.4%</td>
</tr>
<tr>
<td>Class B</td>
<td>Kimono</td>
<td>-6.6%</td>
<td>-3.6%</td>
<td>-6.4%</td>
<td>-3.2%</td>
</tr>
<tr>
<td></td>
<td>ParkScene</td>
<td>-3.4%</td>
<td>-1.9%</td>
<td>-2.9%</td>
<td>-1.1%</td>
</tr>
<tr>
<td></td>
<td>Cactus</td>
<td>-3.3%</td>
<td>-1.8%</td>
<td>-2.2%</td>
<td>-0.9%</td>
</tr>
<tr>
<td></td>
<td>BasketballDrive</td>
<td>-7.8%</td>
<td>-3.2%</td>
<td>-3.7%</td>
<td>-0.9%</td>
</tr>
<tr>
<td></td>
<td>BQTerrace</td>
<td>-2.6%</td>
<td>-1.8%</td>
<td>-1.6%</td>
<td>-0.5%</td>
</tr>
<tr>
<td></td>
<td>Class B Average</td>
<td>-4.7%</td>
<td>-2.5%</td>
<td>-3.4%</td>
<td>-1.3%</td>
</tr>
<tr>
<td>Class C</td>
<td>BasketballDrill</td>
<td>-2.9%</td>
<td>-1.5%</td>
<td>-1.9%</td>
<td>-0.3%</td>
</tr>
<tr>
<td></td>
<td>BQMall</td>
<td>-2.9%</td>
<td>-1.9%</td>
<td>-1.4%</td>
<td>-0.3%</td>
</tr>
<tr>
<td></td>
<td>PartyScene</td>
<td>-2.3%</td>
<td>-1.8%</td>
<td>-1.1%</td>
<td>-0.4%</td>
</tr>
<tr>
<td></td>
<td>RaceHorses</td>
<td>-2.8%</td>
<td>-2.1%</td>
<td>-2.3%</td>
<td>-0.8%</td>
</tr>
<tr>
<td></td>
<td>Class C Average</td>
<td>-2.7%</td>
<td>-1.8%</td>
<td>-1.7%</td>
<td>-0.5%</td>
</tr>
<tr>
<td>Class D</td>
<td>BasketballPass</td>
<td>-2.5%</td>
<td>-1.7%</td>
<td>-1.4%</td>
<td>-0.4%</td>
</tr>
<tr>
<td></td>
<td>BQSquare</td>
<td>-1.8%</td>
<td>-1.2%</td>
<td>-0.8%</td>
<td>-0.2%</td>
</tr>
<tr>
<td></td>
<td>BlowingBubbles</td>
<td>-2.3%</td>
<td>-1.6%</td>
<td>-1.7%</td>
<td>-0.6%</td>
</tr>
<tr>
<td></td>
<td>RaceHorses</td>
<td>-2.6%</td>
<td>-2.5%</td>
<td>-2.2%</td>
<td>-0.6%</td>
</tr>
<tr>
<td></td>
<td>Class D Average</td>
<td>-2.3%</td>
<td>-1.8%</td>
<td>-1.5%</td>
<td>-0.5%</td>
</tr>
<tr>
<td>Class E</td>
<td>Johnney</td>
<td>-6.8%</td>
<td>-3.8%</td>
<td>-4.7%</td>
<td>-1.0%</td>
</tr>
<tr>
<td></td>
<td>FourPeople</td>
<td>-5.6%</td>
<td>-2.8%</td>
<td>-4.1%</td>
<td>-0.8%</td>
</tr>
<tr>
<td></td>
<td>KristenAndSara</td>
<td>-6.6%</td>
<td>-2.9%</td>
<td>-4.0%</td>
<td>-0.8%</td>
</tr>
<tr>
<td></td>
<td>Class E Average</td>
<td>-6.3%</td>
<td>-3.2%</td>
<td>-4.3%</td>
<td>-0.9%</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>-3.8%</td>
<td>-2.3%</td>
<td>-2.7%</td>
<td>-0.9%</td>
</tr>
</tbody>
</table>

TABLE I
Quantitative analysis of selected methods. The results are shown in BD-Rate using HEVC (HM 16.15) as the anchor. PU Size is set to 8 x 8 in both the proposed model and the anchor.

Spatial-Temporal Prediction: Intra

Prediction Block Generation Using Single Layer Network

- **Network input**
  - R rows and R columns reference pixels
  - Height/width of current block smaller than 32: R = 2
  - Otherwise: R = 1
  - Mode:
    - Height/width of current block smaller than 32: 35 modes
    - Otherwise: 11 modes

- **Network output**
  - prediction block

- **Network Structure:**
  - 2-layer neural network during training
  - Layer1: feature extraction, same for all modes
  - Layer2: prediction, different for different modes

\[
 f(x)_i = \max(-1,x_i), \quad R: \text{reference samples}
\]
\[
 t_i = f(A_1 r + b_1) \quad i = \text{network layer index}, k = \text{mode index}
\]
\[
 p_k(r) = A_2 r + b_2 k. \quad P_k(r) = \text{output prediction results}
\]

- **Network Simplification:**
  - Pruning: compare the predictor network and the zero predictor in terms of loss function in frequency domain. If loss decrease is smaller than threshold, use zero predictor instead.
  - Affine linear predictors: removing the activation function, using a single matrix multiplication and bias instead.

![Diagram](Diagram.png)

Figure 1. Prediction of MxN intra block from reconstructed samples using a neural network.

Spatial-Temporal Prediction: Intra

Prediction Block Generation Using Single Layer Network

- **Signaling mode index**
  - Use a two-layer network to predict the conditional probability of each mode
  - The outputs from step#1 are sorted to obtain an MPM-list and an index is signaled in the same way as a conventional intra prediction mode index.

- **Integration into coding system**
  - Network generated prediction as an additional intra mode
  - RDO to choose intra mode

Performance (anchor: AI, VTM1.0)

<table>
<thead>
<tr>
<th>Sequence class</th>
<th>Sequence name</th>
<th>BD-Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>A1</td>
<td>Tango2</td>
<td>-5.20</td>
</tr>
<tr>
<td></td>
<td>FoodMarket4</td>
<td>-5.07</td>
</tr>
<tr>
<td></td>
<td>Campfire</td>
<td>-1.44</td>
</tr>
<tr>
<td></td>
<td>CatRobot1</td>
<td>-3.66</td>
</tr>
<tr>
<td></td>
<td>DaylightRoad2</td>
<td>-4.01</td>
</tr>
<tr>
<td></td>
<td>ParkRunning3</td>
<td>-1.93</td>
</tr>
<tr>
<td>B</td>
<td>MarketPlace</td>
<td>-3.11</td>
</tr>
<tr>
<td></td>
<td>RitualDance</td>
<td>-5.49</td>
</tr>
<tr>
<td></td>
<td>Cactus</td>
<td>-3.88</td>
</tr>
<tr>
<td></td>
<td>BasketballDrive</td>
<td>-2.92</td>
</tr>
<tr>
<td></td>
<td>BQTerrace</td>
<td>-2.60</td>
</tr>
<tr>
<td>C</td>
<td>RaceHorses</td>
<td>-2.78</td>
</tr>
<tr>
<td></td>
<td>BQMall</td>
<td>-4.40</td>
</tr>
<tr>
<td></td>
<td>PartyScene</td>
<td>-2.89</td>
</tr>
<tr>
<td></td>
<td>BasketballDrill</td>
<td>-2.51</td>
</tr>
<tr>
<td>D</td>
<td>RaceHorses</td>
<td>-3.74</td>
</tr>
<tr>
<td></td>
<td>BQSquare</td>
<td>-2.84</td>
</tr>
<tr>
<td></td>
<td>BlowingBubbles</td>
<td>-2.71</td>
</tr>
<tr>
<td></td>
<td>BasketballPass</td>
<td>-3.43</td>
</tr>
<tr>
<td>E</td>
<td>FourPeople</td>
<td>-6.23</td>
</tr>
<tr>
<td></td>
<td>Johnny</td>
<td>-5.80</td>
</tr>
<tr>
<td></td>
<td>KristenAndSara</td>
<td>-6.12</td>
</tr>
<tr>
<td></td>
<td>average</td>
<td>-3.70</td>
</tr>
</tbody>
</table>

Figure 2: Prediction of mode probabilities from reconstructed samples using a neural network.

Spatial-Temporal Prediction: Intra

- **Prediction for block of pixel values**
  - Refinement of traditional prediction: content adaptive filtering
  - Prediction by extrapolation
    - Prediction domain: spatial domain, frequency domain
    - Supplement or replace to traditional modes
    - Network architecture: CNN, RNN, FCN and their combinations
    - Reference pixels: one or multiple raw(s)/column(s)
    - Loss function: Energy of residuals in spatial domain (MSE), Hardmad transform domain (SATD), DCT domain

- **Prediction of intra mode**
  - Probability estimation for all modes: Most Probability Modes list
Spatial-Temporal Prediction: Inter

Subpixel Interpolation

- **Network input**
  - Integer-pixel frame
- **Network output**
  - Half-pixel Interpolated frame
- **Network Structure:**
  - SRCNN: 4-layer CNN

- **Integration into coding system**
  - Different model for different QP
  - Directly replace ½ DCTIF

- **Performance** (anchor: LDP, HM16.7)

Spatial-Temporal Prediction: Inter

Subpixel Interpolation

- **Network input**
  - Integer-pixel position samples

- **Network output**
  - Half-pixel position samples of each sub-pixel position

- **Network Structure:**
  - Different FRCNN for different half-pixel position
  - FRCNN: 4-layer CNN with Inception structure

- **Integration into coding system**
  - Different model for different QP, different half-pixel position and different inter-prediction direction
  - Use as an additional interpolation filter: CU-level selection between CNN, $\frac{1}{2}$ DCTIF and $\frac{1}{4}$ DCTIF

- **Performance (anchor: HM16.7)**
Spatial-Temporal Prediction: Inter

Subpixel Interpolation
- **Network input**
  - Integer-pixel position samples
- **Network output**
  - Quarter/half-pixel position samples of each sub-pixel position
- **Network Structure:**
  - Grouped variation neural network:
    - one model can generate all sub-pixel positions at one sub-pixel level and deal with frames coded with different QPs.
    - Shared feature map is generated and then used to infer sub-pixel samples at different locations.

Integration into coding system
- Different model for different sub-pixel level
- Use as an additional interpolation filter: CU-level selection between CNN, ½ DCTIF and ¼ DCTIF

Performance (anchor: HM16.4)

---

Spatial-Temporal Prediction: Inter

Subpixel Interpolation

- Network input
  - Integer-pixel position samples
- Network output
  - Half-pixel position samples of each sub-pixel position
- Network Structure:
  - 4-layer CNN

Integration into coding system

- Different model for different QP, different sub-pixel position
- Additional mode and replacement mode are studied

Performance (anchor: HM16.7)

Training Scheme:

- Interpolate sub-pixel samples from integer-pixel samples
- Recover integer-pixels samples from sub-pixel samples

Network input

- Integer-pixel position samples

Network output

- Half-pixel position samples of each sub-pixel position

Network Structure:

- 4-layer CNN

Spatial-Temporal Prediction: Inter

Block Refinement of Uni-Prediction

- **Network input**
  - Predicted CU by conventional methods
  - L-shape neighboring pixels of current CU

- **Integration into coding system**
  - Different model for different QP
  - Switchable at CU-level

- **Performance** (anchor: LDP, HM12.0)

**Network output**
- Refined predicted block

**Network Structure:**
- VRCNN: 4-layer CNN

---

Spatial-Temporal Prediction: Inter

Block Refinement of Uni-Prediction

- **Network input**
  - Prediction CU of conventional methods
  - L-shape neighboring reconstructed pixels of both current predicted block and temporal reference block

- **Network output**
  - Refined predicted block

- **Network Structure**
  - Fully connected network + CNN

- **Integration into coding system**
  - Different model for different QP and different blocksize
  - Switchable at CU-level

- **Performance** (anchor: LDP, HM16.9)

---

Table 2. The BD-rate of NNIP for luma component compared to HM16.9

<table>
<thead>
<tr>
<th>Class</th>
<th>Resolution</th>
<th>Sequence</th>
<th>BD-Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>2560×1600</td>
<td>Traffic</td>
<td>+1.5%</td>
</tr>
<tr>
<td>Rlamo</td>
<td></td>
<td>People/Street</td>
<td>-0.6%</td>
</tr>
<tr>
<td>Class B</td>
<td>1920×1080</td>
<td>Cheetah</td>
<td>+2.3%</td>
</tr>
<tr>
<td>BasketballDrive*</td>
<td></td>
<td>BasketballDrive*</td>
<td>-3.6%</td>
</tr>
<tr>
<td>BigBench</td>
<td></td>
<td>AddedScene</td>
<td>-3.3%</td>
</tr>
<tr>
<td>Class C</td>
<td>832×480</td>
<td>AOMedia*</td>
<td>-2.2%</td>
</tr>
<tr>
<td>BasketballDrive*</td>
<td></td>
<td>BasketballDrive*</td>
<td>-1.3%</td>
</tr>
<tr>
<td>BMConway</td>
<td></td>
<td>AddedScene</td>
<td>-0.6%</td>
</tr>
<tr>
<td>Class D</td>
<td>416×240</td>
<td>BMConway</td>
<td>-1.4%</td>
</tr>
<tr>
<td>BasketballDrive*</td>
<td></td>
<td>AddedScene</td>
<td>-0.7%</td>
</tr>
<tr>
<td>Class E</td>
<td>128×720</td>
<td>BMConway</td>
<td>-0.6%</td>
</tr>
<tr>
<td>FivePeople</td>
<td></td>
<td>AddedScene</td>
<td>-1.5%</td>
</tr>
<tr>
<td>Class F</td>
<td>64×360</td>
<td>BMConway</td>
<td>-2.0%</td>
</tr>
<tr>
<td>KinesAndSurf</td>
<td></td>
<td>AddedScene</td>
<td>-2.1%</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td>-1.7%</td>
</tr>
</tbody>
</table>

Table 3. The computational complexity of NNIP

<table>
<thead>
<tr>
<th>Class</th>
<th>∆T_enc</th>
<th>∆T_dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>3273%</td>
<td>1700%</td>
</tr>
<tr>
<td>Class B</td>
<td>3314%</td>
<td>3301%</td>
</tr>
<tr>
<td>Class C</td>
<td>2479%</td>
<td>2416%</td>
</tr>
<tr>
<td>Class D</td>
<td>2842%</td>
<td>1578%</td>
</tr>
<tr>
<td>Class E</td>
<td>5310%</td>
<td>1113%</td>
</tr>
<tr>
<td>Average</td>
<td>3444%</td>
<td>2022%</td>
</tr>
</tbody>
</table>

---

Spatial-Temporal Prediction: Inter

Bi-prediction Block Generation

- **Network input**
  - 2 reference blocks

- **Network output**
  - Bi-directional prediction block

- **Network Structure**
  - CNN

- **Integration into coding system**
  - Different model for different QP and different block size
  - Directly replace the traditional simple average of bi-prediction reference blocks

- **Performance** (anchor: RA, HM16.15)

![Diagram of Bi-prediction Block Generation]

---

Spatial-Temporal Prediction: Inter

Refinement of Bi-prediction Block

- **Network input**
  - 2 reference blocks, together with L-shape neighboring pixels of the 2 reference blocks
  - Predicted block by averaging of 2 reference blocks, together with L-shape neighboring pixels of current block
  - Temporal distances between each reference block and current block

- **Network output**
  - Current bi-predicted block

- **Network Structure**

Integration into coding system

- Different model for different QP and different block size
- Replace traditional averaging bi-prediction in AMVP mode
- Switchable in Merge mode

Performance

(anchor: RA, HM16.15)

![Performance Chart](image)

Refinement of Bi-prediction Block

![Refinement Diagram](image)

Spatial-Temporal Prediction: Inter

- Prediction of block of pixel values
  - Fractional pixel interpolation
    - Super-resolution: position-aware model
  - Refinement of traditional prediction or directly generation of prediction
    - Content adaptive temporal filtering to replace simple average
    - Generalize of bi-hypothesis uni-directional and bi-directional by introduce temporal distances: temporal interpolation and extrapolation
    - With/without motion vector
  - As supplementing inter modes or replacing to traditional ones

- Prediction of motion/optical flow
Transform

Network structure:
- CNN Layers: feature analysis
- Fully Connection Layer: fulfill the transform

Training method:
- Initialization: FC Layer is initialized by transform matrix of DCT/IDCT
- Joint training of FC and CNN
- Loss: joint rate-distortion cost
  - Rate estimated by the l1-norm of the quantized coefficients
  - Distortion estimated by MSE

How good will it be for prediction residuals?

Performance

<table>
<thead>
<tr>
<th>Network</th>
<th>Rate</th>
<th>Distortion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours vs. DCT</td>
<td>32 x 32</td>
<td>28.43%</td>
</tr>
<tr>
<td>Ours vs. JPEG</td>
<td></td>
<td>36.7%</td>
</tr>
<tr>
<td>Ours vs. Tedeschi et al.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kodim01</td>
<td>-17.74%</td>
<td>-28.43%</td>
</tr>
<tr>
<td>kodim02</td>
<td>-5.15%</td>
<td>-56.38%</td>
</tr>
<tr>
<td>kodim03</td>
<td>-10.28%</td>
<td>-47.22%</td>
</tr>
<tr>
<td>kodim04</td>
<td>-2.86%</td>
<td>-50.96%</td>
</tr>
<tr>
<td>kodim05</td>
<td>-18.31%</td>
<td>-24.65%</td>
</tr>
<tr>
<td>kodim06</td>
<td>-13.29%</td>
<td>-35.05%</td>
</tr>
<tr>
<td>kodim07</td>
<td>-11.10%</td>
<td>-39.13%</td>
</tr>
<tr>
<td>kodim08</td>
<td>-11.63%</td>
<td>-24.42%</td>
</tr>
<tr>
<td>kodim09</td>
<td>-8.09%</td>
<td>-41.15%</td>
</tr>
<tr>
<td>kodim10</td>
<td>-5.48%</td>
<td>-62.18%</td>
</tr>
<tr>
<td>kodim11</td>
<td>-10.91%</td>
<td>-33.09%</td>
</tr>
<tr>
<td>kodim12</td>
<td>-8.84%</td>
<td>-43.60%</td>
</tr>
<tr>
<td>kodim13</td>
<td>-13.35%</td>
<td>-23.27%</td>
</tr>
<tr>
<td>kodim14</td>
<td>-15.91%</td>
<td>-50.20%</td>
</tr>
<tr>
<td>kodim15</td>
<td>7.17%</td>
<td>-57.82%</td>
</tr>
<tr>
<td>kodim16</td>
<td>-7.95%</td>
<td>-66.79%</td>
</tr>
<tr>
<td>kodim17</td>
<td>-11.85%</td>
<td>-55.23%</td>
</tr>
<tr>
<td>kodim18</td>
<td>-15.79%</td>
<td>-54.46%</td>
</tr>
<tr>
<td>kodim19</td>
<td>-9.84%</td>
<td>-47.52%</td>
</tr>
<tr>
<td>kodim20</td>
<td>-8.26%</td>
<td>-58.89%</td>
</tr>
<tr>
<td>kodim21</td>
<td>-17.32%</td>
<td>-34.89%</td>
</tr>
<tr>
<td>kodim22</td>
<td>-12.52%</td>
<td>-39.39%</td>
</tr>
<tr>
<td>kodim23</td>
<td>-3.48%</td>
<td>-54.09%</td>
</tr>
<tr>
<td>kodim24</td>
<td>-10.00%</td>
<td>-26.80%</td>
</tr>
<tr>
<td>Average</td>
<td>-8.83%</td>
<td>-38.03%</td>
</tr>
</tbody>
</table>

Quantisation

Content-adaptive QP selection

- Local visibility threshold prediction - VNet-2
  - Convolution layer: 362 trainable parameters (19×19 kernel + 1 bias)
  - Subsampling layer: scale=2, 2 trainable parameters( 1 weight + 1 bias)
  - Full connection layer: 530 trainable parameters( 23×23 weight + 1 bias)

- Quantization steps derivation for CTU
  \[
  \log(Q_{step}) = \alpha C^2 + \beta C + \gamma
  \]
  - \( C \): predicted local visibility threshold
  - \( \{\alpha, \beta, \gamma\} \): model coefficients depend on patch features, predicted from 3 separate NNs.

- Performance
  - 11% bitrate saving for luma channel against HEVC at same SSIM.

Entropy coding

Probability Estimation of Intra Prediction Mode

- **Network inputs**
  - Reconstructed pixels: above-left, above and left blocks with the same size of current coding block
  - Prediction modes of 3 neighboring blocks: one 35-D one-hot binary vector for each neighboring block

- **Network output**
  - 35-D probability vector of 35 intra prediction modes

- **Network structure**
  - Based on LeNet-5

- **Integration into coding system**

- **Performance** (anchor: AI, HM12.0)

---

Entropy coding

Probability Estimation of Transform Kernel Index

- **Network input**
  - Transform coefficients block

- **Network output**
  - Probability vector of transform kernel indexes

- **Network structure**
  - Convolution layer
  - Subsampling layer: scale=2
  - Fully connected layer

- **Integration into coding system**
  - Utilize the probability to reorder transform kernel indexes
  - Binarize the index with truncated unary code

![Diagram of network structure and integration into coding system]

**Performance** (anchor: AI, HM15.0)

<table>
<thead>
<tr>
<th>Class</th>
<th>Sequence</th>
<th>EP</th>
<th>CTX1</th>
<th>CNN</th>
<th>NolIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>BasketballDrive</td>
<td>-0.42</td>
<td>-0.42</td>
<td>-0.48</td>
<td>-0.48</td>
</tr>
<tr>
<td>C</td>
<td>BasketballDrive</td>
<td>-3.14</td>
<td>-3.14</td>
<td>-3.34</td>
<td>-3.34</td>
</tr>
</tbody>
</table>

Entropy coding

- **Probability estimation**
  - For different syntaxes
    - Mode indexes, coefficients values, ...
  - Using correlated information
    - Reconstructed pixels, intermediate reconstructed pixels
    - Decoded neighboring modes
  - Labels
    - Happened or not – POSSIBILITY instead of probability
      - *Possibility* describes the likelihood of a value happening in one symbol while *probability* describe the frequency of a value happening in an infinite string of symbols
      - *Possibility* is a more suitable descriptor for non-stationary process

- **Possibility estimation**

---

**Performance**

Performance (anchor: HEVC)

- **Intra Prediction**: 0.7[1], 3.4[2], 3.8[3], 3.8[4]
- **Inter Prediction**: 0.9[1], 1.3[2], 0.9[3], 3.6[4], 1.7[6], 5.2[5], 3.0[7], 4.8[8]
- **Entropy Coding**: 0.2[1], 1.2[2], 4.7[3], 7.4[1]
- **Filter**: 2.3[4], 1.3[2], 3.6[3], 8.6[5]
- **Transform**: 38[1]
- **Quantization**: 11[1]

*: compared to JPEG
#: evaluated in BD-rate with MS-SSIM (%)
Hybrid or End-to-End?

Input video

Spatial redundancy
Transform

Perceptual redundancy
Quantization

Dequantization
Inv. Transform

Bitstream

Entropy coding

Statistical redundancy

In-loop filter

Inter prediction

Intra prediction

Spatial redundancy

Temporal redundancy

Perceptual redundancy
Spatial redundancy
Statistical redundancy
End-to-End Video Coding

- **Overall Network Structure**
  - limited temporal information utilization

- **End-to-End Video Compression Framework**
  - Residual Encoder Net & Decoder Net
  - Motion Compensation Network
  - Optical Flow Net
  - Bit Rate Estimation Net

- **Loss function**

\[
L = \lambda D + R = \lambda d(x_t, \tilde{x}_t) + R(\tilde{m}_t) + R(\tilde{y}_t)
\]

- **References**

End-to-End Video Coding

Overall Network Structure

- Intra Coding & Residual Coding
  ✓ An end-to-end image compression network

- Inter Coding
  ✓ One-stage Unsupervised Flow Learning:

Optical flow estimation and compression realized in one stage

✓ Context Adaptive Flow Compression

For entropy model, besides using spatial features and hyperpriors, temporal priors generated by ConvLSTM are used.

Loss function

\[ L = \lambda D + R = \lambda d(x_t, \tilde{x}_t) + R(\tilde{m}_t) + R(\tilde{y}_t) \]

End-to-End Video Coding

Performance

Figure 8: Visual Comparison. Reconstructed frames of our method, H.265/HEVC and H.264/AVC. We avoid blocky artifacts and provide better quality of reconstructed frame at low bit rate.

Conclusion

- **All roads lead to Rome**
  - NN modules embedded into hybrid video coding frameworks can bring significant coding gains
  - End-to-end image and video coding – still follow the source coding theory
  - **Training**: separately or jointly

- Performance of learning based coding comes from
  - Re-organization of information: non-linear transform to independent symbol
  - Quantization: scalar vs. vector quantization
  - Entropy coding: hyperprior to estimate of possibility + arithmetic coding
Latest Publications on Learning-based Coding

- **SPECIAL SECTION ON LEARNING-BASED IMAGE AND VIDEO CODING, IEEE TCSVT 2020. Jul**

12 papers:

- End-to-end image compression (1)
- Intra prediction (3)
- Inter prediction (2)
- Filtering (2)
- Arithmetic coding (1)
- Encoder optimization (3)
Deep Neural Network Based Video Coding

- AhG on DNNVC established in 130\textsuperscript{th} MPEG meeting in Apr. 2020

- Mandates
  - Evaluate and quantify \textit{performance improvement potential of DNN based video coding technologies} (including hybrid video coding system with DNN modules and end-to-end DNN coding systems) compared to existing MPEG standards such as HEVC and VVC, considering various quality metrics;
  - Study \textit{quality metrics for DNN based video coding};
  - Solicit input contributions on DNN based video coding technologies;
  - Analyze the \textit{encoding and decoding complexity} of NN based video coding technologies by considering software and hardware implementations, including impact on power consumption;
  - Investigate technical aspects specific to NN-based video coding, such as design network representation, operation, tensor, on-the-fly network adaption (e.g. updating during encoding) etc

Subscribe mailing list:

https://lists.aau.at/mailman/listinfo/mpeg-dnnvc
Image/Video Coding for ...

- Reconstruction image/video for **human vision** -- yes, but not the only target

- Coding image/video for **machine understanding**
Video Coding for Machine: Use Cases

• 6 major application areas
  • Smart Industry
  • Intelligent Transportation
  • Smart Retailer
  • Smart City
  • Smart Sensors Networks
  • Immersive Video / HD Entertainment
  • Smart Media Editing and Creation

• Use Cases:
  • machine-oriented analysis
  • hybrid machine/human representation
Video Coding for Machine: Potential Pipelines

1. Video Encoder → Bitstream → Video Decoder → Video → Machine Analysis (Part1) (Part2) → Inference Results

2. Video → Feature → Feature Conversion → Video Encoder → Bitstream → Video Decoder → Feature Inverse Conversion → Feature → Machine Analysis (Part2) → Inference Results

3. Video → Feature → Machine Analysis (Part1) → Feature → Feature Encoder → Bitstream → Feature Decoder → Machine Analysis (Part2) → Inference Results

4. Video → Feature → Machine Analysis (Part1) → Feature → Video Encoder → Bitstream → Video Decoder → Video
Video Coding for Machine

- **AhG on VCM established in 127th MPEG meeting in July, 2019**

- **Mandates**
  - To create and evaluate anchors for object detection, object segmentation and object tracking
  - To collect data sets, ground truth
  - To define metrics for object detection, object segmentation and object tracking
  - To compare performance of analysis using original data vs. analysis using compressed features at different bit rates in the typical cases of object detection
  - To collect evidence on the level of achievability of combined human/machine-oriented video representation and compression
  - To encourage experts to provide feature stream codecs
  - To encourage experts to provide uncompressed bitstream from feature extractor

- **Preliminary Timeline**
  - 2019.07 Establish VCM, set up mailing list, release use cases
  - 2020.01 Release requirements, provide evidences on Mandate 5 and 6
  - **2020.07 Call for evidence**
Thanks!

Contact me: yul@zju.edu.cn