iCNN: A Convolutional Neural Network for Fractional Interpolation in Video Coding

Chi Do-Kim Pham\textsuperscript{1} and Jinjia Zhou\textsuperscript{1,2}
\textsuperscript{1}Graduate School of Science and Engineering, Hosei University, Japan
\textsuperscript{2}JST, PRESTO, Tokyo, Japan

Presented by: Chi Do-Kim Pham
chi.kim.pham.do.94@stu.hosei.ac.jp
Outline

1. Introduction to Convolutional Neural Network-based fractional interpolation

2. Proposal: interpolation Convolutional Neural Network

3. Experimental Results

4. Conclusion and Future Work
Motion Compensated Prediction

MCP searches for the **best matching block** from the list of reference (reconstructed) frames and encode:

- Motion vector.
- Residual.
Fractional interpolation in video coding

Recent video coding standards interpolate reconstructed frame to get fractional pixels

High Efficiency Video Coding (HEVC) uses DCTIF for fractional interpolation

Discrete Cosine Transform Interpolation Filter (DCTIF) in HEVC

Y fractional samples in HEVC

### Position | Filter coefficients
--- | ---
1/4 | -1, 4, 10, 58, 17, -5, 1, 0
2/4 | -1, 4, -11, 40, -11, 4, -1
3/4 | 0, 1, -5, 17, 58, -10, -1

15 h×w fractional Luma samples by DCTIF

Reconstructed image (integer sample)
Drawbacks of HEVC filters

- HEVC interpolation filters have drawbacks:
  - Designed filters are not flexible enough
  - Applied area is small

→ Apply CNN for Fractional interpolation in video coding

However,

- CNN-based fractional interpolation tends to change integer pixels
- Have no training set for fractional interpolation because fractional pixels do not really exist.
Key ideas of CNN-based fractional interpolation in the Related works

• *Keep integer pixels*: generate only fractional samples or a mask to restrict integer samples

  ![Diagram of integer and fractional samples]

  - DCTIF
  - ... (fractional samples)

• *Training set*: Assume and extract integer and fractional samples in each image

  ![Diagram of training set]

  - Need to train many models for the fractional samples, or
  - Support only half pixels
Proposals

• Proposal 1: Design (and prove) a training set for learning fractional pixels in video coding.

• Proposal 2: iCNN (interpolation CNN) for fractional interpolation.
  • Support half and quarter pixel
  • One model for all fractional samples

• Proposal 3: an RDO-based interpolation selection to further improve coding performance
Proposal 1: Training Data Selection (prove)

We archive a 7.1% bitrate saving compare to standard HEVC at the same image quality with the first 3 frames over sequence class B, C.
Proposal 1: Training data preparation

1. Assume and extract integer and fractional samples
2. Encode integer video
3. Extract Y component from reconstructed integer video
4. Extract Y component from fractional videos
Proposal 2 & 3: Interpolation CNN architecture and RDO-based interpolation method selection

- **Fractional Motion Search/Motion Compensation**

  - **Reconstructed frame** (integer samples)
  - **DCTIF**: 15 DCTIF fractional frames
  - **iCNN architecture**: 15 iCNN fractional frames

- **CU Interpolation Method Selection**
  - Best predicted CU by DCTIF
  - Smaller RDO-cost?
  - Best predicted CU by iCNN

**Support**: all the fractional samples

**Interpolation flag is coded in bypass mode**
Experimental Settings

• Training data:
  • Extract and process from YUV 4:2:0 standard test sequences from class A:

  Traffic

  PeopleOnTheStreet

  Pedestrian

  • Low Delay P configuration with QP 22, 27, 32, and 37.

• HEVC Test Model (HM) version 16.18

• Test on 20 standard sequences [1] from Class B to F

HEVC standard test sequences

Class A
2560x1600
For training

Traffic

Class B
1920x1080
Kimono

BasketballDrill

Class D
416x240
BlowingBubbles

Class C
832x480

Class E
1280x720
Johnny

Screen-content video

Class F
SlideEditing
Visualize coding performance

(a) RD-curve of Y component on Kimono

Evaluated by Bjontegaard (BD-rate): How many bits a method can save compared to an anchor method.
Overall Results

BD-rate (%): How many bits a method can save compared to an anchor. *Note: the lower negative BD-rate, the better result than HEVC

<table>
<thead>
<tr>
<th>Class</th>
<th>Without iCNN/DCTIF selection</th>
<th>With iCNN/DCTIF selection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Without extra bits</td>
<td>With extra bits</td>
</tr>
<tr>
<td>Average B</td>
<td>-0.7</td>
<td>-6.3</td>
</tr>
<tr>
<td>Average C</td>
<td>0.1</td>
<td>-4.0</td>
</tr>
<tr>
<td>Average D</td>
<td>1.4</td>
<td>-5.2</td>
</tr>
<tr>
<td>Average E</td>
<td>-1.9</td>
<td>-5.9</td>
</tr>
<tr>
<td>Average F</td>
<td>-0.4</td>
<td>-0.6</td>
</tr>
<tr>
<td>Average all sequences</td>
<td>-0.2</td>
<td>-4.4</td>
</tr>
</tbody>
</table>

BD-rate (%) of our model compared to the anchor HM on Y component
Visualize iCNN/DCITF selection results

Visualization iCNN/DCTIF interpolation selection on POC7 of sequence BQ Mall. Ratio of iCNN over the standard test sequences: 43.27%

* Note: Blue, red blocks represent for CU that choose DCTIF, iCNN, respectively. Other parts are block coded in intra coding or has integer motion vector
Conclusion and Future Works

• Conclusion
  • Design training set for fractional interpolation in video coding
  • CNN-based fractional interpolation
    • 2.6% BD-rate reduction compared to HEVC
    • Support half and quarter pixels in one model

• Future work
  • Support CNN-based Chroma interpolation
  • Encode interpolation flag to further improve coding efficiency
References


iCNN: A Convolutional Neural Network for Fractional Interpolation in Video Coding

Chi Do-Kim Pham\textsuperscript{1} and Jinjia Zhou\textsuperscript{1,2}
1 Graduate School of Science and Engineering, Hosei University, Japan
2 JST, PRESTO, Tokyo, Japan

Presented by: Chi Do-Kim Pham
chi.kim.pham.do.94@stu.hosei.ac.jp
Target of video coding: try to reduce the size of video (for storage or transmission) while keeping the video quality
Typical Interpolation (or Super Resolution) vs. Fractional Interpolation in video coding

(1) No ground truth for training fractional interpolation in video coding

(A) Typical interpolation

Down-sampled image → Original image

(B) Fractional interpolation in video coding

Reference frame Frame t

Fractional frames

... interpolate

Current block to be encoded Frame t+1

Fractional pixels do not exist

(2) If we use the typical super resolution training set for training fractional interpolation in video coding, integer pixels may change after convolution.
Typical Fractional interpolation

We do not generate the interpolated image for fractional motion search
Integer sample

Current block to be encoded size $h \times w$

15 $h \times w$ fractional Luma samples by DCTIF
**Related work (1)**

*Keep integer pixels*: use 3 CNNs model for half-pel interpolation

*Training set:*
- Blur images
- Extract integer and half pixels
- Encode integer image to be input, half pixels are ground truth

Keep integer pixels: use 15 CNN models for interpolating half- and quarter-pel

Training set: extract real reference blocks from encoded bitstream files.

Keep integer pixels: use special mask

Training set:
- Extract integer image
- Encode integer image to be input, original frame is set as ground truth