# Non-Local ConvLSTM for Video Compression Artifact Reduction

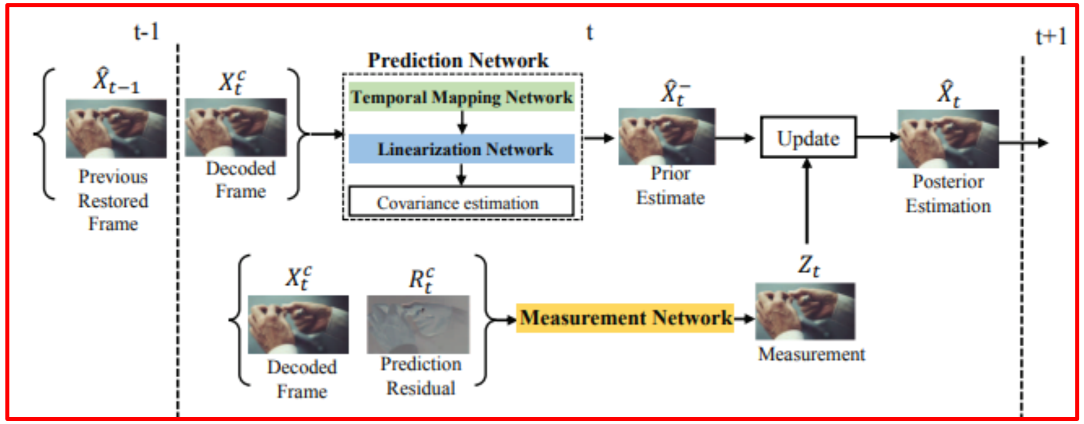
Yi Xu[1*] Longwen Gao[2] Kai Tian[1] Shuigeng Zhou[1†] Huyang Sun[2]
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, Shanghai, China
[2]Bilibili, Shanghai, China
{yxu17,ktian14,sgzhou}@fudan.edu.cn {gaolongwen,sunhuyang}@bilibili.com

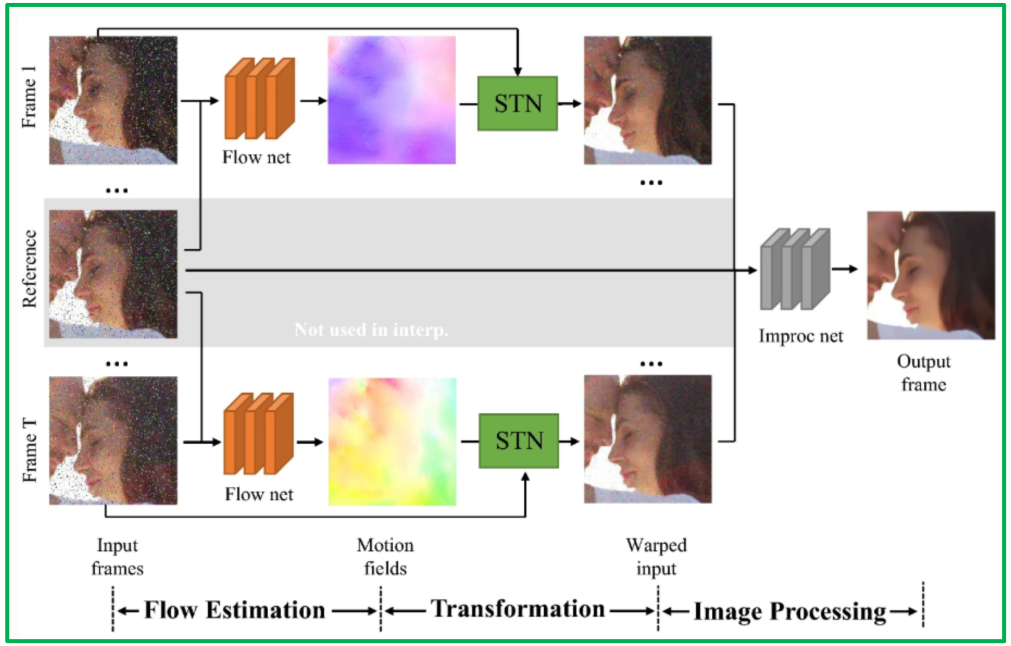
ICCV 2019

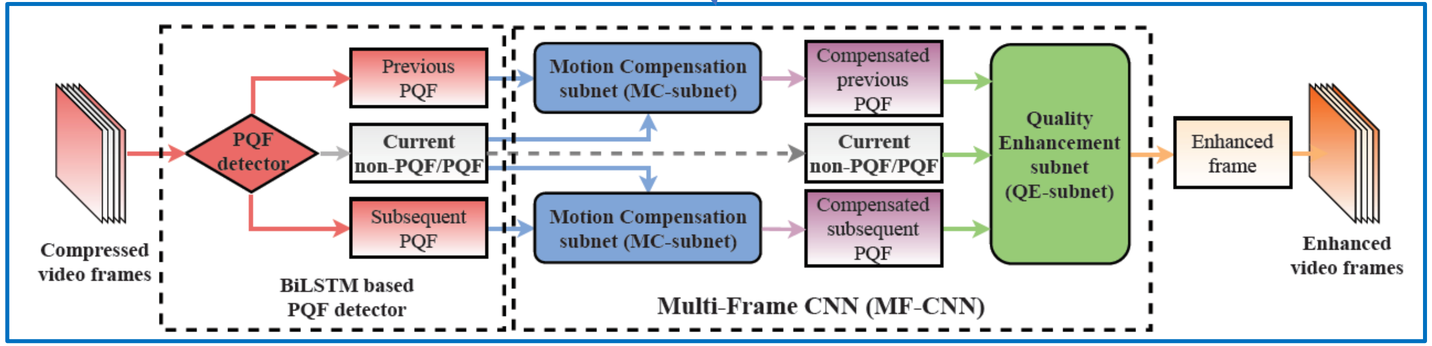
Presented by Qing Lyu on 7/22/20

# Background

- Video compression artifact reduction
  - Single image compression artifact reduction
  - Video compression artifact reduction
    - Deep Kalman filter network (ECCV 2018, DL club: July 31, 2019)
    - Task-oriented motion-based network(IJCV 2019, DL club: September 11, 2019)
    - Network using motion-compensated nearest PQFs (PAMI 2019)

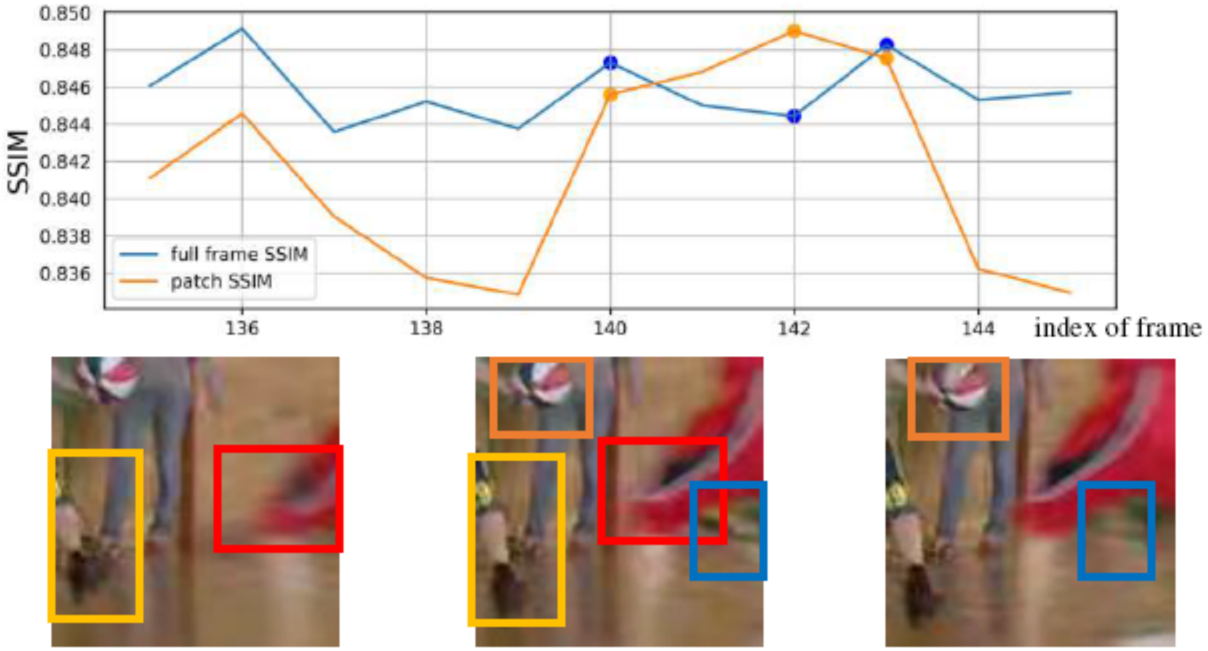


# Shortcoming of existing methods

- Existing methods used a pair of neighboring frames, may miss high-quality details of some other neighbor frames




patch from $140^{th}$ frame  patch from $142^{th}$ frame  patch from $143^{th}$ frame

# Advantageous

- No accurate motion estimation and compensation is explicitly needed
- It is applicable to videos compressed by various commonly-used compression algorithms such as H.264/AVC and H.265/HEVC
- The proposed method outperforms the existing methods

# Method: network

- End-to-end framework with three modules
  - Encoder
  - NL-ConvLSTM
  - Decoder

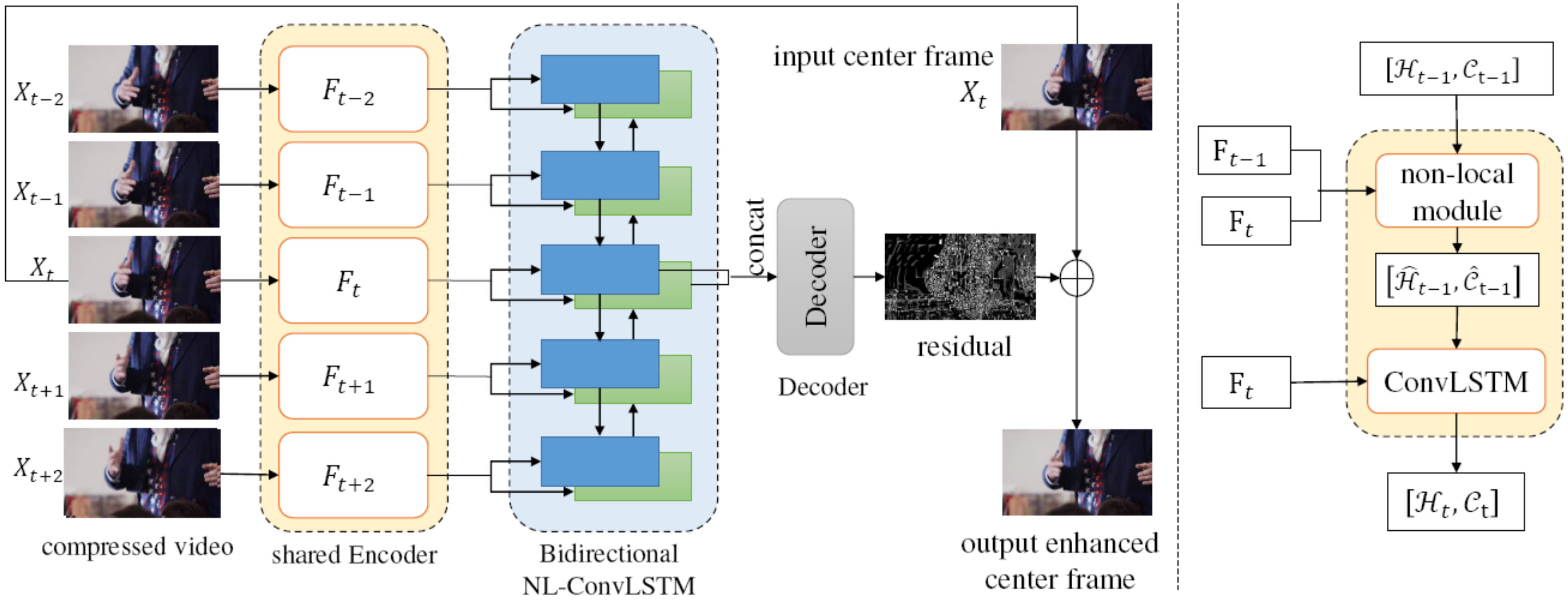



Figure 2. The framework of our method (left) and the architecture of NL-ConvLSTM (right)

# Method: Non-Local ConvLSTM

- ConvLSTM

$$[\mathcal{H}_t, \mathcal{C}_t] = ConvLSTM(F_t, [\mathcal{H}_{t-1}, \mathcal{C}_{t-1}])$$

- NL-ConvLSTM

$$\begin{aligned} S_t &= NL\left(F_{t-1}, F_t\right), \\ \left[\hat{\mathcal{H}}_{t-1}, \hat{\mathcal{C}}_{t-1}\right] &= NLWarp([\mathcal{H}_{t-1}, \mathcal{C}_{t-1}], S_t), \\ [H_t, C_t] &= ConvLSTM(F_t, \left[\hat{\mathcal{H}}_{t-1}, \hat{\mathcal{C}}_{t-1}\right]), \end{aligned}$$

$$\begin{aligned} D_t\left(i, j\right) &= \left\| F_{t-1}\left(i\right) - F_t\left(j\right) \right\|_2, \\ S_t\left(i, j\right) &= \frac{\exp\left(-D_t\left(i, j\right)/\beta\right)}{\sum_{\forall i} \exp\left(-D_t\left(i, j\right)/\beta\right)}, \\ \left[\hat{\mathcal{H}}_{t-1}, \hat{\mathcal{C}}_{t-1}\right] &= \left[\mathcal{H}_t \cdot S_t, \mathcal{C}_t \cdot S_t\right], \end{aligned}$$

# Calculation simplification

- Directly compute S is the warping operation will incur extremely high computation and memory cost

- To simplify the calculation, proposing a two stage NL approximation method
  - Use average pooling to downsample the feature map from the Encoder

$$D_t(i,j) = \|F_{t-1}(i) - F_t(j)\|_2, \longrightarrow D_t^p(i,j) = \left\|F_{t-1}^p(i) - F_t^p(j)\right\|_2$$

  - Compute and store the similarities between each pixel of $F_t^p$ and the corresponding $k \times p^2$ pixels of $F_{t-1}^p$. While for the other pixels in the preceding frame, the elements of $D_t$ and $S_t$ are set to infinity and 0 respectively.

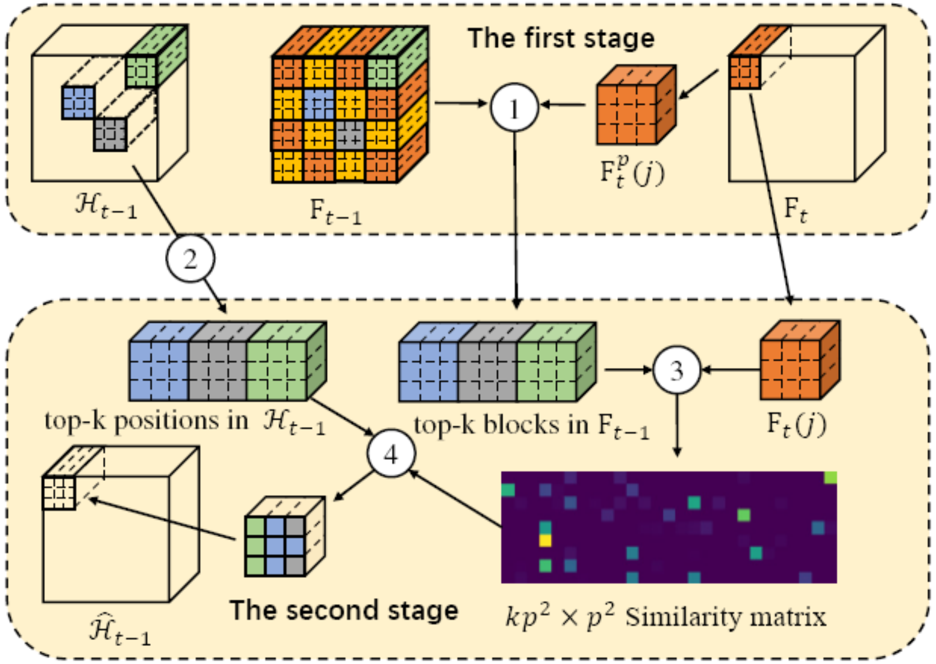



Figure 3. The workflow of two-stage similarity approximation. ① finding the top-$k$ most similar blocks in $F_{t-1}$ with respect to block $F_t^p(j)$ from $F_t$; ② extracting blocks in $\mathcal{H}_{t-1}$ from the corresponding positions of the top-$k$ most similar blocks in $F_{t-1}$; ③ calculating pixel-wise similarity between the selected blocks from $F_{t-1}$ and $F_t^p(j)$; ④ NLWarp operation for $\mathcal{H}_t$.

# Complexity analysis

Table 1. Complexity comparison of the original non-local approach and ours. Here, $N$ and $C$ are the numbers of positions and channels, $k$ and $p$ are the number of pre-filtered blocks and the downsampling scale. By setting $k$=4 and $p$=10, our method cuts the time and space to about 1/1000 of that consumed by the original non-local method in 1080P videos.

| | Original non-local | NL-ConvLSTM |
|---|---|---|
| Time | $\mathcal{O}(2N^2C)$ | $\mathcal{O}((N/p^2)^2(C + \log k) + 2kNCp^2)$ |
| Space | $\mathcal{O}(2N^2)$ | $\mathcal{O}((N/p^2)^2 + kN/p^2 + 2kNp^2)$ |

to $\mathcal{O}((N/p^2)^2C + 2kNCp^2)$. By properly choosing the values of $k$ and $p$ so that $kp^2 \ll N$, we have $\phi/\psi = 1/(2p^4) + kp^2/N \ll 1$, which means that our method dramatically reduces the computation cost of the original method. And for a given $k$, $\phi/\psi$ achieves the minimum $1.5(k/N)^{2/3}$ with $p$=$(N/k)^{1/6}$. Similar conclusion can be

# Experiment

- Datasets
  - Vimeo-90K
    - 89,800 video sequences
    - 448x256 resolution
    - Compression algorithm: x265 in FFmpeg with QP=32 or 37
  - Yang’s dataset
    - 70 video sequences
    - Resolution vary from 352x240 to 2560x1600
    - Compression algorithm: HEVC LDP

# Ablation study

Table 2. Ablation study of the proposed NL-ConvLSTM on Yang *et al.*'s dataset with $QP$=37. The results of PSNR improvement $\Delta$PSNR (db) are reported in the $1^{st}$ row. The results of SSIM improvement $\Delta$SSIM ($\times 10^{-2}$) are listed in the $2^{nd}$ row.

| | Encoder-Decoder with 1 frame | ConvLSTM with 7 frames | ME-ConvLSTM with 7 frames | Our method with 7 frames |
|---|---|---|---|---|
| $\Delta$PSNR | 0.395 | 0.456 | 0.503 | **0.601** |
| $\Delta$SSIM | 0.684 | 0.723 | 0.827 | **0.897** |

# Quantitative comparison

Table 3. Average PSNR/SSIM on Vimeo-90K.

| QP | 32 | 37 |
|---|---|---|
| HEVC [34] | 34.19 / 0.950 | 31.98 / 0.923 |
| ARCNN [11] | 34.87 / 0.954 | 32.54 / 0.930 |
| DnCNN [49] | 35.58 / 0.961 | 33.01 / 0.936 |
| DSCNN [44] | 35.61 / 0.960 | 32.99 / 0.938 |
| DKFN [26] | 35.81 / 0.962 | 33.23 / 0.939 |
| 3D CNN | 35.81 / 0.961 | 33.25 / 0.938 |
| Our method | **35.95 / 0.965** | **33.39 / 0.943** |

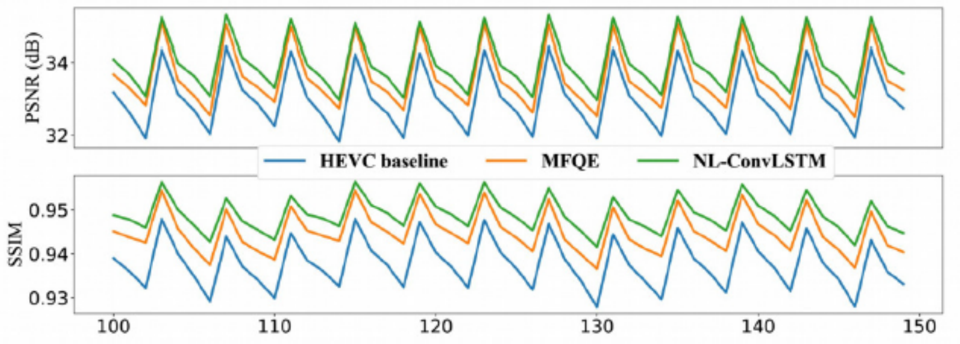


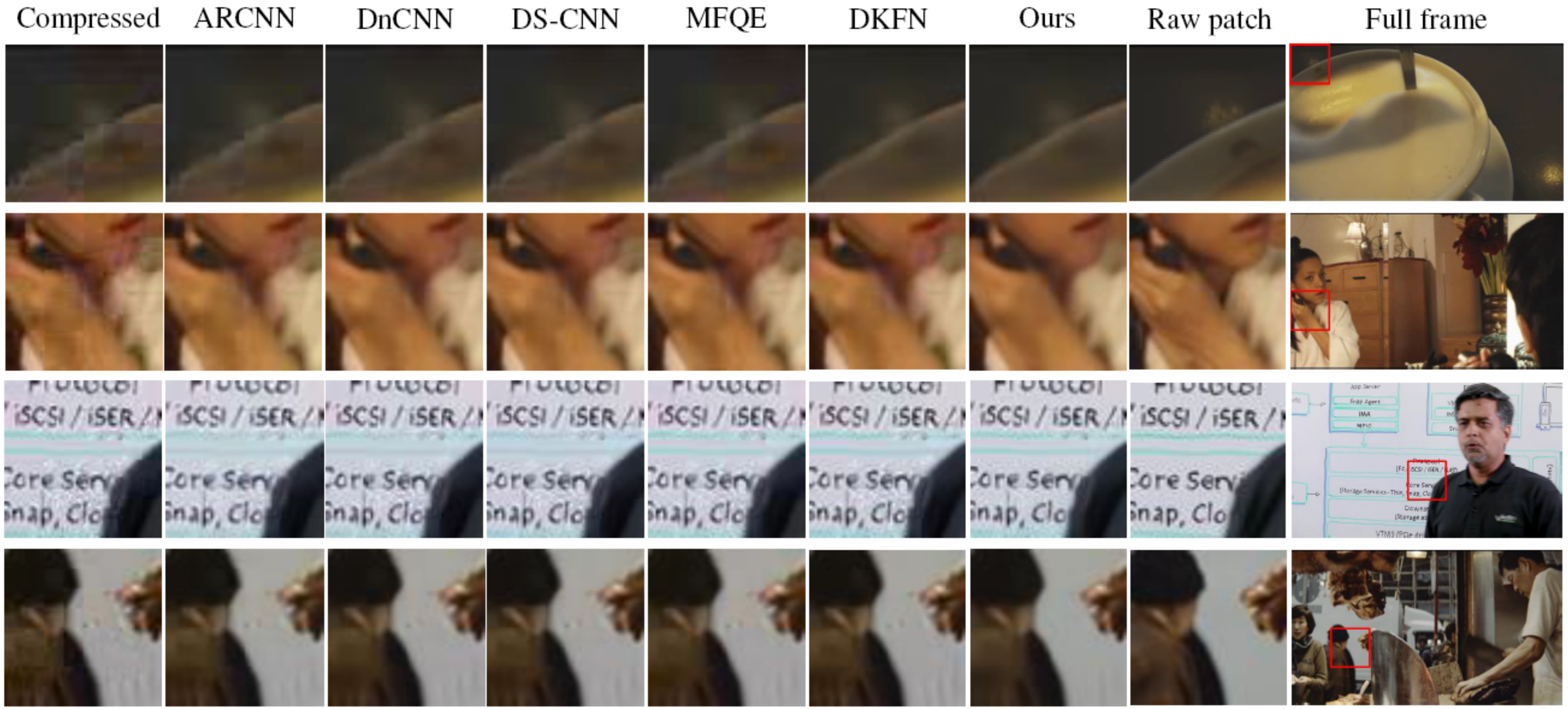

Figure 5. PSNR/SSIM curves of HEVC baseline, MFQE and NL-ConvLSTM on the sequence *TunnelFlag* with $QP$=37.

Table 4. Average $\Delta$PSNR (dB) and $\Delta$SSIM ($\times 10^{-2}$) on Yang *et al.*'s dataset.

| QP | Seq. | ARCNN [11] | DnCNN [49] | DSCNN [44] | MFQE [45] | Our method |
|---|---|---|---|---|---|---|
| 37 | 1 | 0.241 / 0.51 | 0.448 / 0.83 | 0.492 / 0.87 | 0.772 / 1.15 | **0.827 / 1.21** |
| | 2 | 0.115 / 0.30 | 0.439 / 0.52 | 0.458 / 0.58 | 0.604 / 0.63 | **0.971 / 0.92** |
| | 3 | 0.161 / 0.49 | 0.276 / 0.76 | 0.271 / 0.74 | 0.472 / 0.91 | **0.483 / 0.99** |
| | 4 | 0.183 / 0.35 | 0.377 / 0.55 | 0.393 / 0.54 | 0.438 / 0.48 | **0.576 / 0.66** |
| | 5 | 0.150 / 0.30 | 0.333 / 0.48 | 0.356 / 0.53 | 0.550 / 0.52 | **0.598 / 0.74** |
| | 6 | 0.161 / 0.23 | 0.415 / 0.50 | 0.435 / 0.49 | 0.598 / 0.51 | **0.658 / 0.67** |
| | 7 | 0.128 / 0.29 | 0.284 / 0.44 | 0.277 / 0.45 | 0.390 / 0.45 | **0.394 / 0.58** |
| | 8 | 0.125 / 0.37 | 0.276 / 0.61 | 0.230 / 0.63 | 0.484 / 1.01 | **0.563 / 1.18** |
| | 9 | 0.149 / 0.38 | 0.299 / 0.71 | 0.271 / 0.66 | 0.394 / 0.92 | **0.439 / 1.03** |
| | 10 | 0.146 / 0.24 | 0.289 / 0.58 | 0.274 / 0.54 | 0.402 / 0.80 | **0.501 / 0.99** |
| | Ave. | 0.156 / 0.35 | 0.344 / 0.59 | 0.346 / 0.60 | 0.510 / 0.74 | **0.601 / 0.90** |
| 42 | Ave. | 0.252 / 0.83 | 0.301 / 0.96 | 0.364 / 1.06 | 0.461 / — | **0.614 / 1.47** |

1: *PeopleOnStreet* 2: *TunnelFlag* 3: *Kimono* 4: *BarScene* 5: *Vidyo1*
6: *Vidyo3* 7: *Vidyo4* 8: *BasketballPass* 9: *RaceHorses* 10: *MaD*

# Qualitative comparison



# Run time comparison

Table 5. Run-time (*ms per frame*) comparison among six methods.

| Resolution | 180x180 | 416x240 | 640x360 | 1280x720 | 1920x1080 |
|---|---|---|---|---|---|
| ARCNN [11] | 1.73 | 4.58 | 9.19 | 36.06 | 80.70 |
| DnCNN [49] | 6.30 | 15.84 | 35.51 | 139.77 | 315.83 |
| DSCNN [44] | 15.26 | 36.88 | 82.31 | 322.92 | 731.21 |
| MFQE[4] [45] | 20.28+ | 51.01+ | 112.87+ | 443.82+ | 1009.00+ |
| original NL | 4391.75 | - | - | - | - |
| ours | 102.13 | 304.11 | 621.94 | 2607.60 | 6738.00 |

# Contribution

- Propose a new idea for video compression artifact reduction by exploiting multiple preceding and following frames of the target frame, without explicitly computing and compensating motion between frames
- Develop an end-to-end deep neural network called non-local ConvLSTM to learn the spatiotemporal information from multiple neighboring frames
- Design an approximate method to compute the inter-frame pixel-wise similarity
- Conduct extensive experiments over two datasets to evaluate the proposed method, which achieves state-of-the-art performance for video compression artifact reduction