Optimization Techniques of Deep Learning Models for Visual Quality Improvement

Lorenzo Palloni

UNIVERSITÀ DEGLI STUDI FIRENZE

School of Mathematics, Physics and Natural Science
Master of Science Degree in Computer Science

Supervisor: Marco Bertini
Co-supervisor: Leonardo Galteri
Co-supervisor: Donatella Merlini

April 21, 2023

Introduction

(a) Original

(b) Low-resolution

(c) Super-resolution

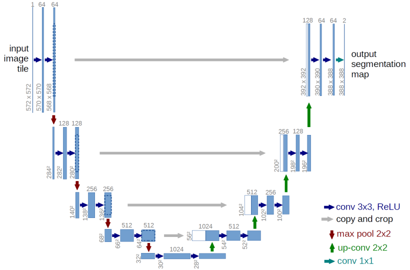
- Goal: Improve video restoration with deep learning models
- Challenge: Computational complexity of deep learning models
- Solution: Apply quantization techniques for faster inference and reduced memory usage

- Focus: Artefact removal and super-resolution tasks
- Method: Post-training quantization using TensorRT
- Impact: Enable practical deployment on resource-constrained devices

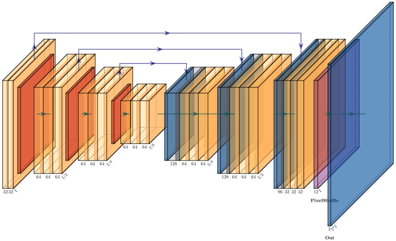
Quality Metrics for Video Restoration

- Traditional Metrics:
  - Evaluate image quality based on numerical comparisons
  - An example: SSIM (Structural Similarity)
- Perceptual Metrics:
  - Evaluate image quality based on human perception
  - An example: LPIPS (Learned Perceptual Image Patch Similarity)
- No-Reference Metrics:
  - Evaluate image quality without reference images
  - An example: BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator)
- Video Quality Metrics:
  - Take into account temporal aspects of quality degradation
  - An example: VMAF (Video Multi-Method Assessment Fusion)

- UNet: encoder-decoder structure with skip connections
- SRUNet: modified UNet for super-resolution and artefact removal
- Training setup: Generative Adversarial Network (GAN) framework
- Generator loss: combination of LPIPS and SSIM metrics



(a) UNet architecture



(b) SRUNet architecture

Quantization

- Quantization in deep learning: process of reducing the precision of weights and activations of a neural network
- Two main approaches to quantize a deep learning model:
  - Quantization-Aware Training (QAT)
    - Quantization incorporated during training
    - Model learns to be more robust to quantization effects
  - Post-Training Quantization (PTQ)
    - Quantization applied after training the model
    - Model accuracy may be affected

Dataset and Training Setup

- BVI-DVC dataset:
  - Designed for deep video compression tasks
  - Includes 200 frame sequences truncated at the 64th frame
  - Various content types: natural scenes, man-made objects, cityscapes
- Training setup:
  - Models trained on frame patches of 96×96 pixels randomly cropped
  - Model training required ∼72 hours on a single GPU NVIDIA Titan Xp
  - Custom data-loader: to speed up training reducing GPU bottlenecks

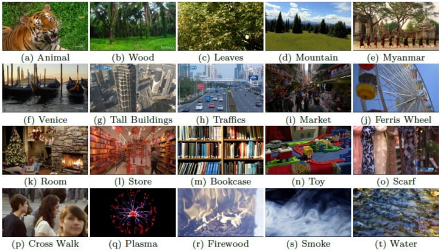


Figure: Some frame examples from the BVI-DVC dataset.

Experiments

- Post-Training Quantization (PTQ) implemented using TensorRT
- Comparison of UNet and SRUNet models with TensorRT-optimized versions (FP32, FP16, INT8)
- Evaluation on 60 test frames using perceptual and traditional metrics
- Quantitative results: Perceptual and traditional metrics show minor differences between optimized and non-optimized models.
- Analysis of inference speed and memory consumption

Quantitative Results: Perceptual Metrics

	LPIPS ↓	DISTS ↓	BRISQUE ↓
UNet	**0.2897 ± 0.0138**	**0.1222 ± 0.0067**	31.1372 ± 1.1690
UNet-FP32	**0.2897 ± 0.0138**	**0.1222 ± 0.0067**	31.1373 ± 1.1684
UNet-FP16	0.2898 ± 0.0138	0.1223 ± 0.0067	31.1383 ± 1.1691
UNet-INT8	0.3041 ± 0.0137	0.1283 ± 0.0066	**29.6849 ± 1.0138**
SRUNet	0.3111 ± 0.0151	**0.1717 ± 0.0047**	27.3738 ± 3.5705
SRUNet-FP32	0.3111 ± 0.0151	**0.1717 ± 0.0047**	27.3736 ± 3.5697
SRUNet-FP16	0.3111 ± 0.0151	**0.1717 ± 0.0047**	27.3790 ± 3.5739
SRUNet-INT8	**0.3068 ± 0.0137**	0.1722 ± 0.0044	**26.1546 ± 3.2273**

Table: Evaluations on perceptual metrics on 60 test frames (mean ± standard deviation).

Quantitative Results: Traditional Metrics

	SSIM $\uparrow$	MS-SSIM $\uparrow$	PSNR $\uparrow$
UNet	**0.8952 $\pm$ 0.0084**	**0.8517 $\pm$ 0.0067**	**21.6506 $\pm$ 0.1269**
UNet-FP32	**0.8952 $\pm$ 0.0084**	**0.8517 $\pm$ 0.0067**	**21.6506 $\pm$ 0.1269**
UNet-FP16	**0.8952 $\pm$ 0.0084**	**0.8517 $\pm$ 0.0067**	**21.6506 $\pm$ 0.1269**
UNet-INT8	0.8941 $\pm$ 0.0084	0.8508 $\pm$ 0.0067	21.6388 $\pm$ 0.1286
SRUNet	**0.8894 $\pm$ 0.0084**	**0.8457 $\pm$ 0.0062**	21.3670 $\pm$ 0.1248
SRUNet-FP32	**0.8894 $\pm$ 0.0084**	**0.8457 $\pm$ 0.0062**	21.3670 $\pm$ 0.1248
SRUNet-FP16	**0.8894 $\pm$ 0.0084**	**0.8457 $\pm$ 0.0062**	**21.3671 $\pm$ 0.1248**
SRUNet-INT8	0.8882 $\pm$ 0.0084	0.8442 $\pm$ 0.0062	21.3320 $\pm$ 0.1224

Table: Evaluations on traditional metrics on 60 test frames (mean $\pm$ standard deviation).

Quantitative Results: Inference Speed

	times [s] $\downarrow$	speedup $\uparrow$
UNet	0.0348 ± 0.0004	
UNet-FP32	0.0279 ± 0.0004	1.25X
UNet-FP16	0.0279 ± 0.0004	1.25X
UNet-INT8	**0.0146 ± 0.0006**	**2.38X**
SRUNet	0.0123 ± 0.0001	
SRUNet-FP32	0.0087 ± 0.0005	1.41X
SRUNet-FP16	0.0087 ± 0.0004	1.41X
SRUNet-INT8	**0.0054 ± 0.0006**	**2.27X**

Table: Evaluation times over 300 runs (mean $\pm$ standard deviation).

Qualitative Results: Buildings

(a) Original Image (384×384)

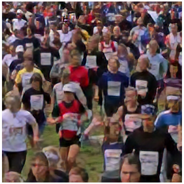
(b) Compressed and Down-scaled (96×96)

(c) Up-scaled using Bilinear Interpolation (384×384)

(d) Up-scaled using SRUNet (384×384)

(e) Up-scaled using SRUNet-INT8 (384×384)

Qualitative Results: Crowd



(a) Original Image (384×384)

(b) Compressed and Down-scaled (96×96)

(c) Up-scaled using Bilinear Interpolation (384×384)

(d) Up-scaled using SRUNet (384×384)

(e) Up-scaled using SRUNet-INT8 (384×384)

Conclusions

- Aim to optimize deep learning models for video quality improvement
- Investigated post-training quantization techniques using TensorRT
- Achieved up to 2.38X speedup and 64.3% size reduction without compromising performance
- Future research:
  - designing efficient NN model architectures
  - co-designing NN architecture and hardware together
  - pruning
  - knowledge distillation
  - exploring different quantization techniques

Thank you, for your attention!

Do you have any questions?

UNIVERSITÀ
DEGLI STUDI
FIRENZE

Appendix

Quantitative Results: VMAF

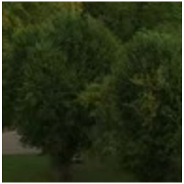
	VMAF (mean) ↑	VMAF (harmonic mean) ↑
UNet	47.71	47.20
UNet-FP32	**47.72**	**47.21**
UNet-FP16	47.71	47.20
UNet-INT8	47.47	46.96
SRUNet	**47.20**	**46.65**
SRUNet-FP32	47.19	46.64
SRUNet-FP16	47.19	**46.65**
SRUNet-INT8	47.18	46.64

Table: VMAF scores on a 120-second-test video.

Qualitative Results: Trees

(a) Original Image (384×384)

(b) Compressed and Down-scaled (96×96)



(c) Up-scaled using Bilinear Interpolation (384×384)

(d) Up-scaled using SRUNet (384×384)

(e) Up-scaled using SRUNet-INT8 (384×384)

LPIPS-Comp: Introduction (1/5)

- LPIPS-Comp: perceptual similarity metric based on deep neural networks
- Trained on a compression-specific perceptual similarity dataset
- Better alignment with human judgement on general compression tasks

VGG-16 and Feed-Forward Process (2/5)

- LPIPS-Comp uses VGG-16 architecture
- ReLU activations after each conv block in the first five layers
- Batch-normalization applied
- Feed-forward performed on VGG-16 for both original (y) and reconstructed image ($\hat{y}$)

Feature Activations and Normalization (3/5)

- $F(y)$ and $F(\hat{y})$ return stacks of feature activations for all layers L
- Unit-normalized in the channel dimension: $z_y^l, z_{\hat{y}}^l \in R^{H_l \times W_l \times C_l}$ where $l \in L$
- H_l, W_l are the spatial dimensions; C_l is the number of channels

Scaling and L2 Distance (4/5)

- $z_y^l, z_{\hat{y}}^l$ are scaled channel-wise with the vector $w_l \in R^{C_l}$
- L2 distance computed and averaged over spatial dimensions
- Channel-wise sum performed

LPIPS-Comp Equation (5/5)

$$\text{LPIPS-Comp}(y, \hat{y}) = \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \left|\left| w_l \odot \left(z^l_{\hat{y},h,w} - z^l_{y,h,w} \right) \right|\right|_2^2$$

- Weights in F learned for image classification and kept fixed
- w are linear weights learned on a compression-specific similarity dataset