Two-Level Approach for No-Reference Natural Video Quality Assessment

Jari Korhonen

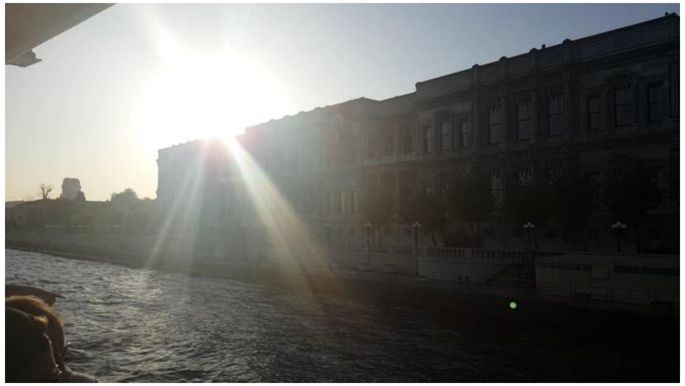
15. October, 2019

jari.t.korhonen@ieee.org

Introduction

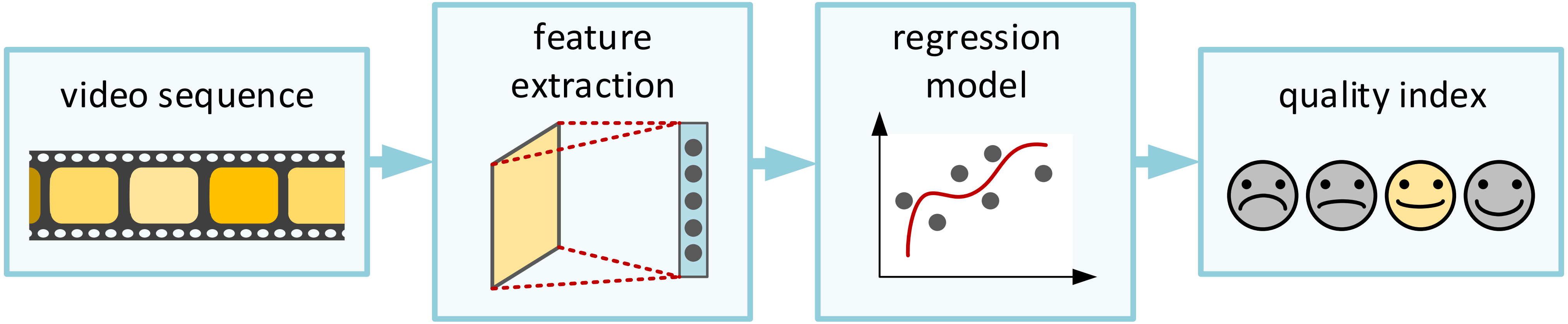
- User generated video content becoming very common
  - Smartphone cameras, wireless connections and social media platforms available for content generation and sharing for a reasonable cost
- User generated content often prone to capture artifacts
  - Sensor noise, motion blur, shakiness, over- and underexposure…





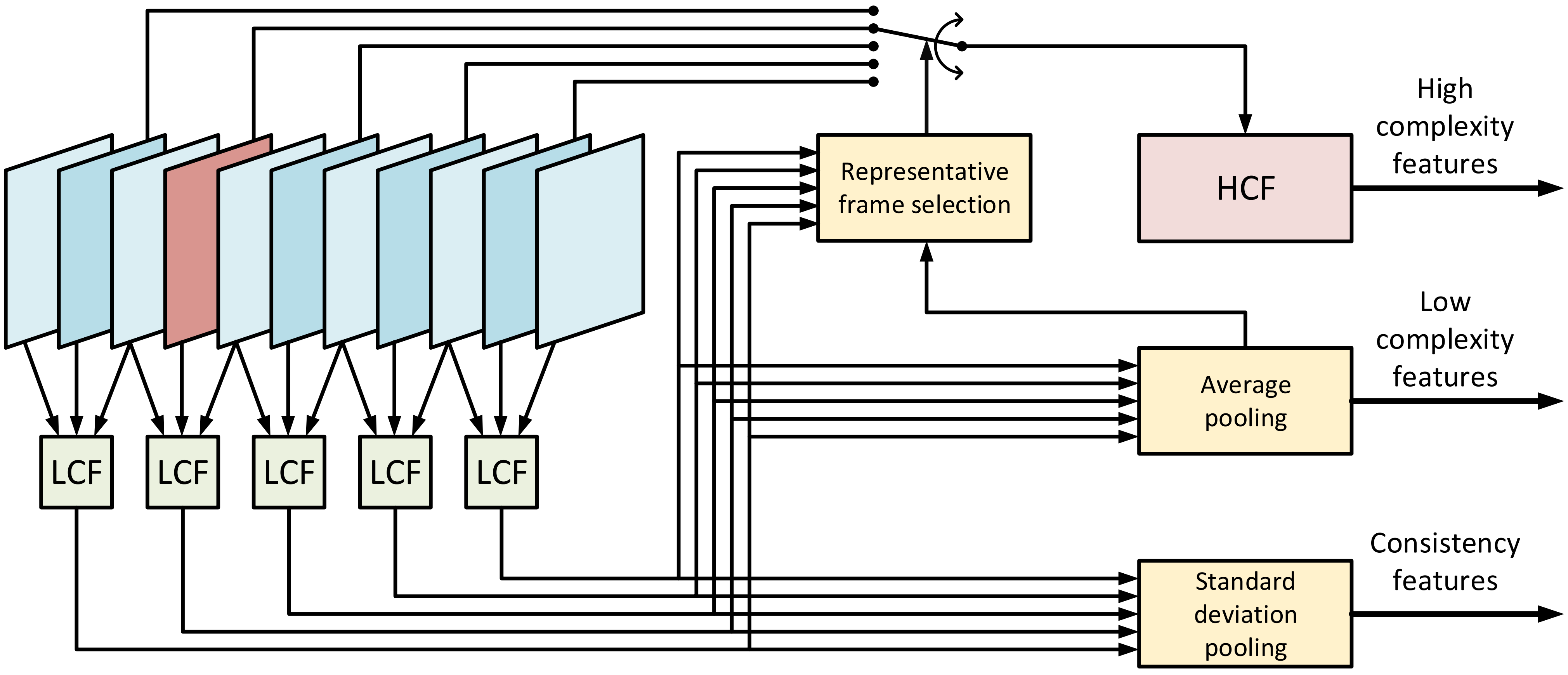
Example images from LIVE Video Quality Challenge database, http://live.ece.utexas.edu/research/LIVEVQC/

Motivation



- Several no-reference video quality metrics (NR-VQMs) have been proposed already
  - However, only few learning-based models with implementations available
  - Mostly focused on compression and transmission artifacts, not natural video with capture artifacts
  - Proposed techniques typically too complex for practical applications

Proposed two-level NR-VQA model



Low complexity features (LCF)

- Hand-crafted features (22 in total) with two main purposes

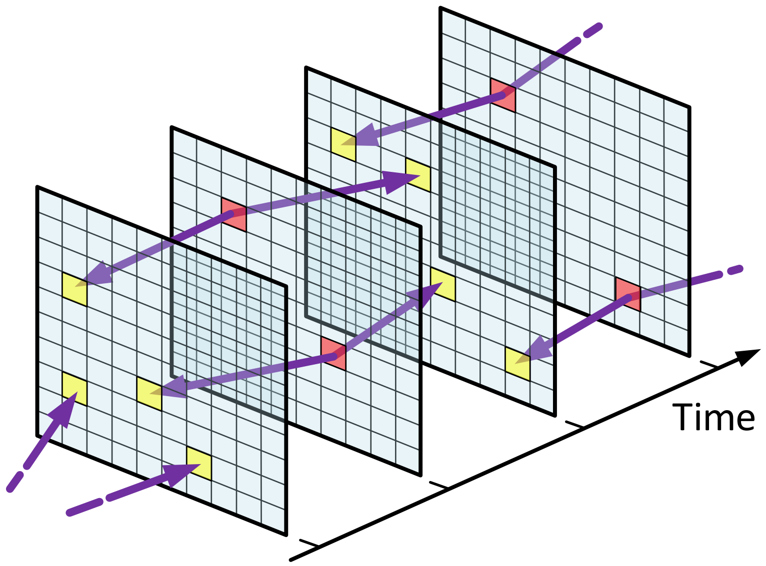
  1) collect information about local temporal characteristics and motion consistency

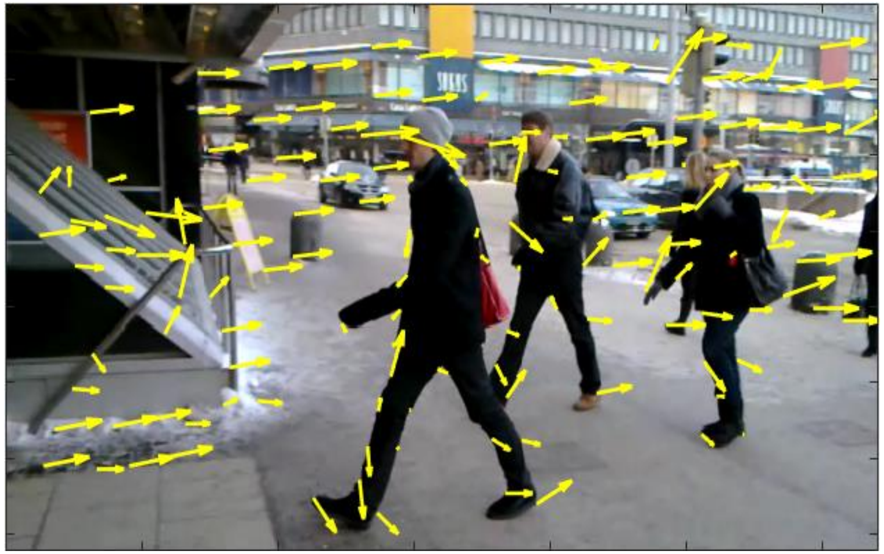
  2) select the most representative frame in a segment for computing high complexity features

- Mostly based on statistical characteristics of motion
  - Derived from motion vectors
  - Represent motion intensity, consistency, jerkiness…
- Some LCFs also represent spatial characteristics
  - Simple features assessing spatial activity, sharpness, blockiness and interlacing

Motion estimation for LCFs

- Convolution filter to find key pixels
  - Simpler than e.g. SIFT, but sufficient to find points statistically accurate enough
- Motion estimation only for 3x3 blocks around key pixels
  - Much lower complexity than normal block-based motion estimation





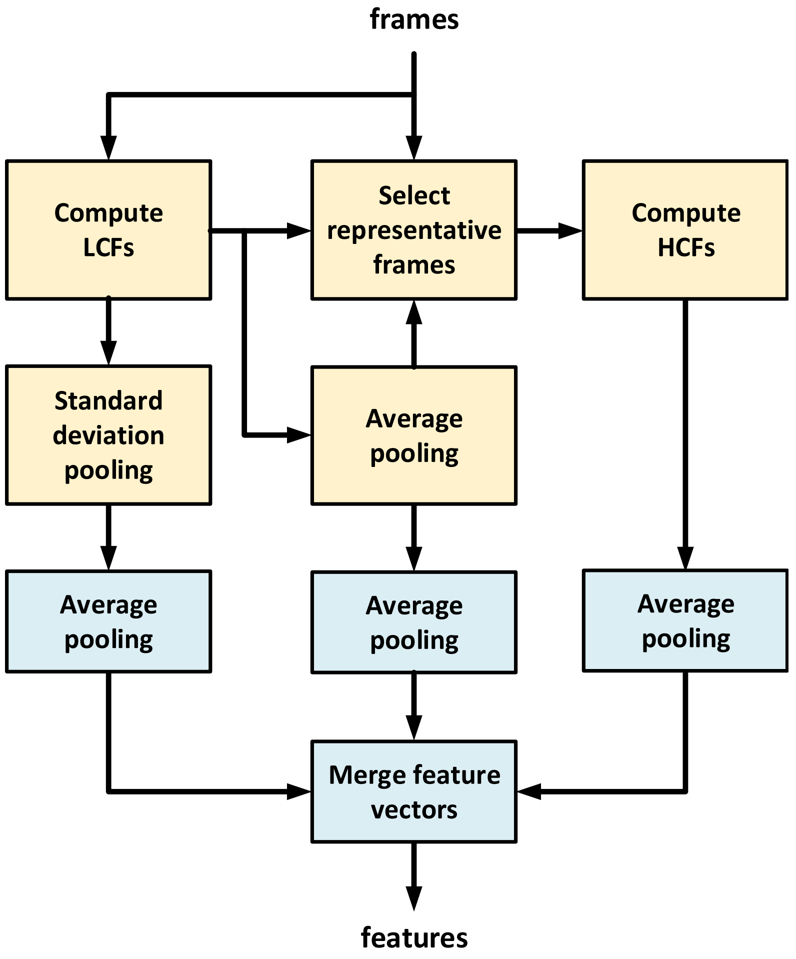
High complexity features (HCF)

- Hand-crafted features representing spatial characteristics of the representative frames (30 in total)

Type	Description	#
Spatial activity	Based on Sobel filter, mean and standard deviation	4
Exposure	Segmentation to find over- and underexposed areas	4
Blockiness	Sobel filter and vertical/horizontal autocorrelation	3
Contrast and colorfulness	Histogram comparison, CIELAB	4
Noise	Local maximum/minimum, strength and intensity	3
Sharpness	2D autocorrelation of 16x16 pixel blocks	9
DCT-based	Features derived from DCT coefficients	3

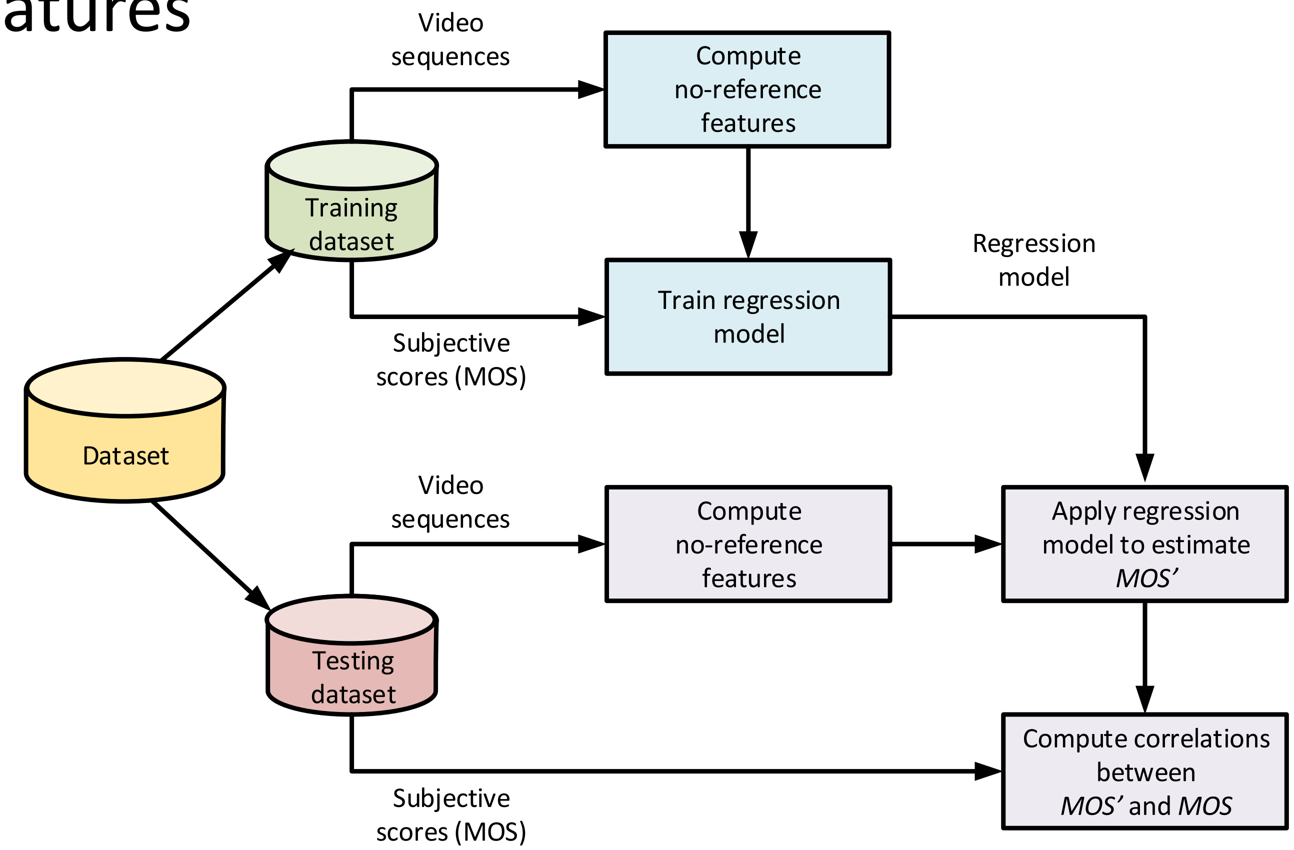
Pooling of features

- Low complexity features for each segment (1 sec) pooled by average and standard deviation pooling
  - Referred as motion consistency features
- High complexity features and pooled LCFs average pooled and concatenated to form the final feature vector
  - Different temporal pooling strategies and scene change detection out of the scope of this work



Regression and testing procedure

- Different regression methods can be used obtain quality estimate from the features



Comparison study

- Feature extraction in Matlab, regression in Python
- Three different public datasets used for validation

Dataset	CVD2014 (Univ Helsinki)	KoNViD-1k (Univ Konstanz)	LIVE-Qualcomm (Univ Texas)
Videos	234	1200	208
Dimensions	640x480, 1280x720	960x540	1920x1080
Method	Lab-based, scale 1-100	Crowdsourcing, scale 1-5	Lab-based, scale 1-100
Test subjects	27-33 (6 experiments)	642 (min 50 per video)	39
Main strength	Realistic content, several devices and impairments	Very large database, a lot of contents and users	Realistic content with smartphones, Full HD reso
Main weakness	Small number of scenes, inconsistent methods	Exotic contents, method prone to outliers	Different scene types not well balanced, only smartphones

Results for CVD2014

- 100 test runs, 80:20 random split to training/testing sets

	Support Vector Regression			Random Forest Regression		
	PCC	SRCC	RMSE	PCC	SRCC	RMSE
V-CORNIA	0.71 (±0.08)	0.68 (±0.09)	15.2 (±1.6)	0.63 (±0.10)	0.61 (±0.10)	16.9 (±1.5)
V-BLIINDS	0.71 (±0.09)	0.70 (±0.09)	15.2 (±2.2)	0.74 (±0.07)	0.73 (±0.08)	14.6 (±1.6)
HIGRADE	0.76 (±0.08)	0.74 (±0.06)	14.2 (±1.5)	0.73 (±0.07)	0.72 (±0.08)	14.8 (±1.6)
FRIQUEE	0.83 (±0.04)	0.82 (±0.05)	12.0 (±1.2)	0.77 (±0.07)	0.74 (±0.07)	13.9 (±1.6)
Proposed	0.85 (±0.04)	0.84 (±0.04)	11.3 (±1.3)	0.81 (±0.05)	0.79 (±0.05)	12.8 (±1.5)

Results for KoNViD-1k

- 100 test runs, 80:20 random split to training/testing sets

	Support Vector Regression			Random Forest Regression		
	PCC	SRCC	RMSE	PCC	SRCC	RMSE
V-CORNIA	0.51 (±0.04)	0.51 (±0.04)	0.560 (±0.042)	0.46 (±0.09)	0.46 (±0.09)	0.546 (±0.038)
V-BLIINDS	0.60 (±0.04)	0.63 (±0.04)	0.513 (±0.027)	0.64 (±0.04)	0.65 (±0.04)	0.490 (±0.022)
HIGRADE	0.72 (±0.03)	0.73 (±0.03)	0.444 (±0.023)	0.62 (±0.04)	0.61 (±0.04)	0.501 (±0.022)
FRIQUEE	0.74 (±0.03)	0.74 (±0.03)	0.432 (±0.022)	0.73 (±0.03)	0.73 (±0.03)	0.441 (±0.021)
Proposed	0.77 (±0.02)	0.78 (±0.02)	0.406 (±0.018)	0.74 (±0.03)	0.74 (±0.03)	0.433 (±0.020)

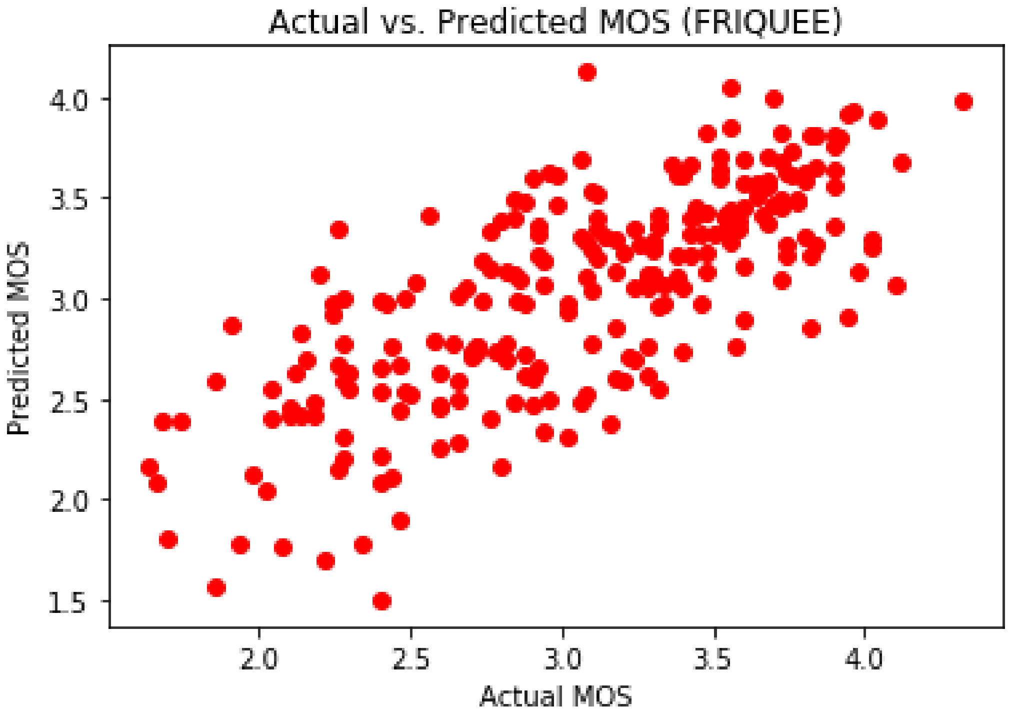
Results for LIVE-Qualcomm

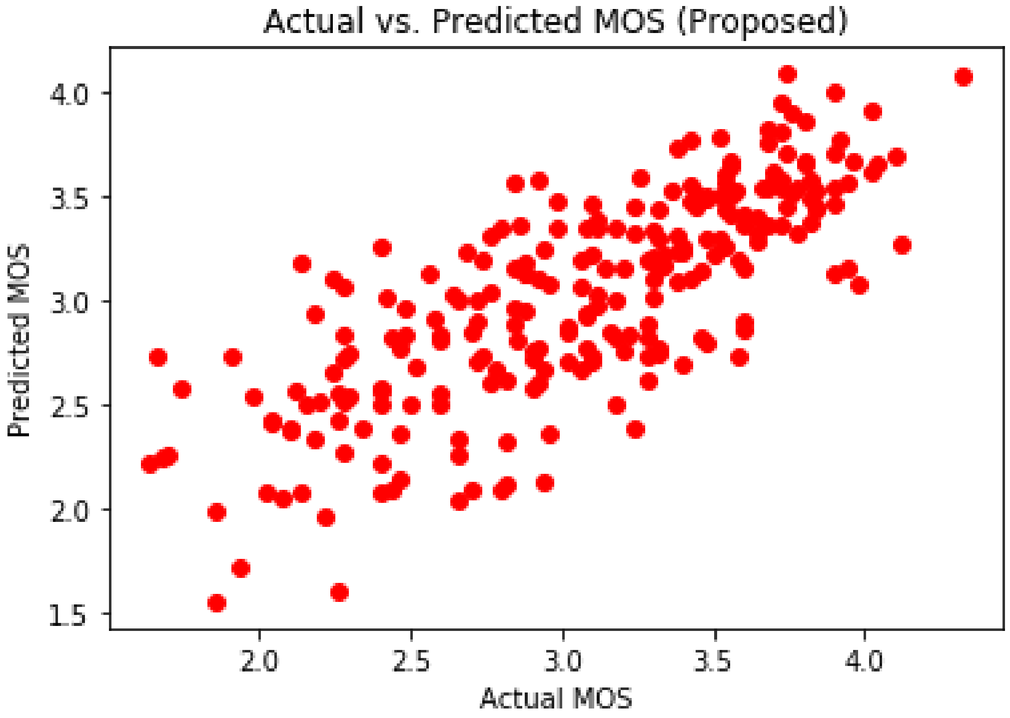
- 100 test runs, 80:20 random split to training/testing sets

	Support Vector Regression			Random Forest Regression		
	PCC	SRCC	RMSE	PCC	SRCC	RMSE
CORNIA	0.61 (±0.09)	0.56 (±0.09)	9.7 (±0.9)	0.43 (±0.13)	0.40 (±0.13)	10.6 (±1.1)
V-BLIINDS	0.67 (±0.09)	0.60 (±0.10)	9.2 (±0.9)	0.63 (±0.10)	0.59 (±0.10)	9.4 (±0.9)
HIGRADE	0.71 (±0.08)	0.68 (±0.08)	8.6 (±1.1)	0.68 (±0.07)	0.65 (±0.10)	8.9 (±1.0)
FRIQUEE	0.78 (±0.06)	0.74 (±0.07)	7.6 (±0.8)	0.64 (±0.09)	0.62 (±0.10)	9.3 (±1.0)
Proposed	0.81 (±0.06)	0.78 (±0.06)	7.1 (±1.0)	0.71 (±0.10)	0.68 (±0.09)	8.8 (±1.1)

Example scatterplots (KoNViD-1k)

- FRIQUEE vs. proposed model (representative example splits)





Complexity comparison

- Running times for Matlab (same computer and settings)
  - Average time of decoding sequences from CVD2014 dataset (five sequences for two different resolutions each)

Method	Low resolution	High resolution
FRIQUEE (1 frame/sec)	466.7 s	1355.9 s
V-BLIINDS	455.6 s	1050.2 s
Proposed	69.4 s	222.2 s
V-CORNIA (1 frame/sec)	15.3 s	24.9 s
HIGRADE	7.4 s	20.9 s

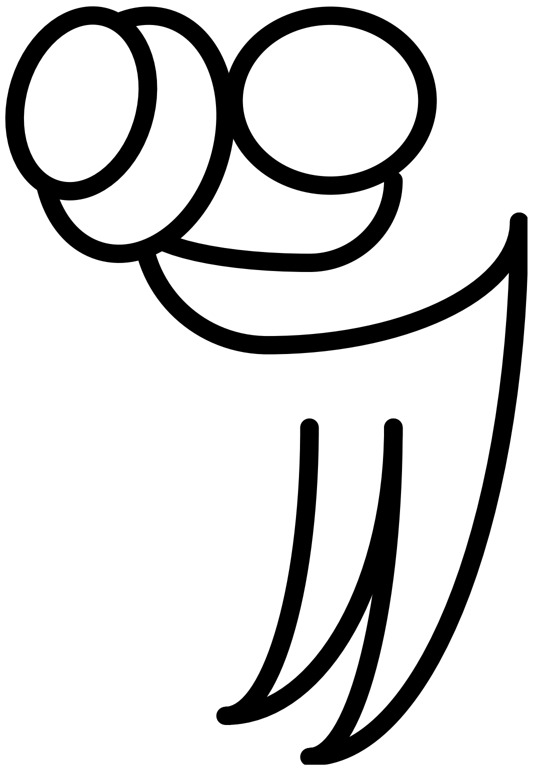
Improvement possibilities

- Matlab / Python implementation still slow
  - C++/OpenCV version would be substantially faster
- Optimizing the features
  - Possibly three-level hierarchy, developing better features
  - Using Convolutional Neural Network (CNN) for spatial features
- Optimizing pooling
  - Content change aware temporal pooling strategies
- Using larger datasets for training and testing
  - The availability of large public databases is still relatively limited

Summary

- No-Reference video quality model proposed
  - Hand-crafted features, hierarchical computation of frame level features (high complexity features only computed for a representative subset of frames)
  - Learning-based regression to combine features into quality score
- Better performance than state-of-the-art quality models
  - More accurate prediction of subjective quality score
  - Lower complexity than the best performing other models
- Possibilities for further development
  - Real-time implementation, better features, better pooling
  - Replacing HCFs with CNN-based features

Thank you!



Publication:
J. Korhonen: “Two-Level Approach for No-Reference Consumer Video Quality Assessment,” IEEE Trans. Image Processing, 28(12), 5923-5938.

You can download the implementation from
https://github.com/jarikorhonen/nr-vqa-consumervideo

Contact: jari.t.korhonen@ieee.org