Measuring perceptual video quality with VMAF

Zhi Li
Video Algorithms, Netflix

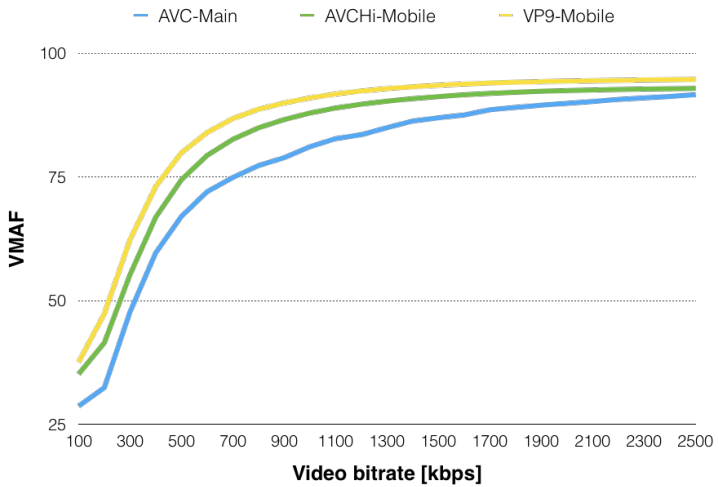
9/18/17 @ ICIP 2017

Outline

- The need for a better quality metric for video
- How VMAF works
- VMAF open-source project

Ways to measure video quality

Subjective Assessment



Automated Assessment
using PSNR, SSIM, or VMAF


PSNR 37.3 dB


PSNR 32.9 dB
0009 0'00"36

Need a better perceptual metric

- Accurately measures **human perception** of quality
- Consistent across content
- Can be run at scale
- Works well relevant to adaptive streaming
    - Compression artifacts
    - Scaling artifacts

VMAF: Video Multimethod Assessment Fusion


PSNR 37.1 dB, VMAF 71.1
0109 0'04"36


PSNR 32.9 dB, VMAF 70.2
0009 0'00"36


PSNR 29.1 dB, VMAF 20.4


PSNR 29.3 dB, VMAF 69.8

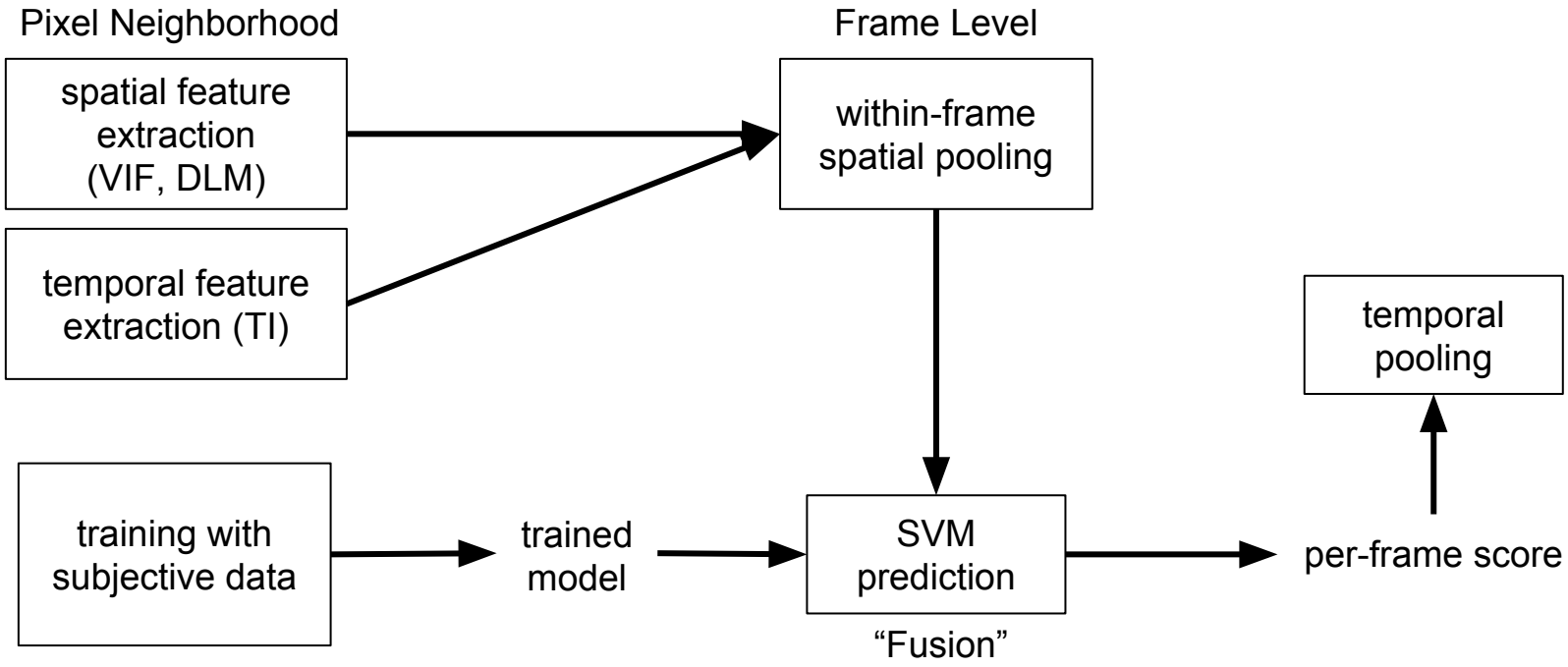
Video Multimethod Assessment Fusion

- Full-reference video quality metric
- Combines multiple elementary quality metrics
  - Visual quality fidelity (VIF*) @ 4 scales
  - Detail loss measure (DLM**)
  - Temporal information (TI) - average pixel difference between adj. frames
- Machine-learning regression to predict a final “fused” score, guided by subjective data

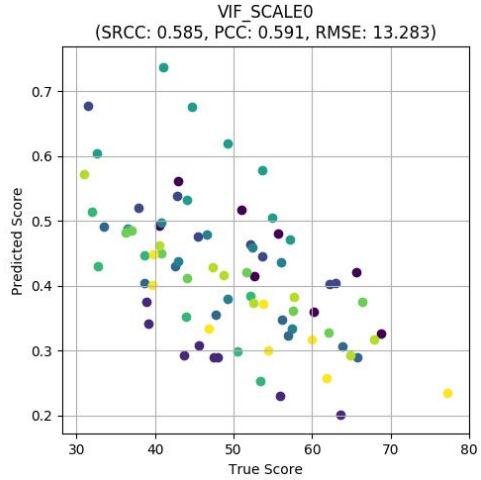
*__Visual Information Fidelity -__ H. Sheikh and A. Bovik, “Image Information and Visual Quality”.

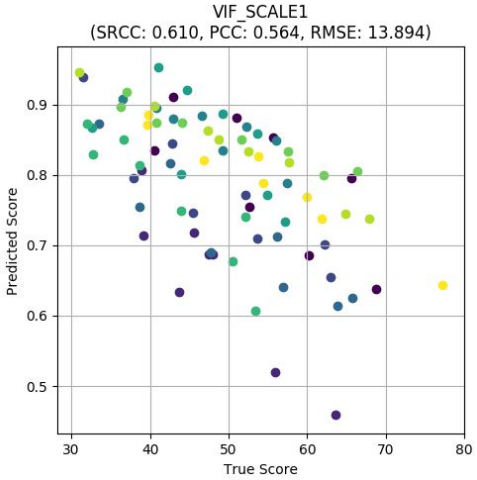
**__Detail Loss Measure -__ S. Li, F. Zhang, L. Ma, and K. Ngan, “Image Quality Assessment by Separately Evaluating Detail Losses and Additive Impairments”.

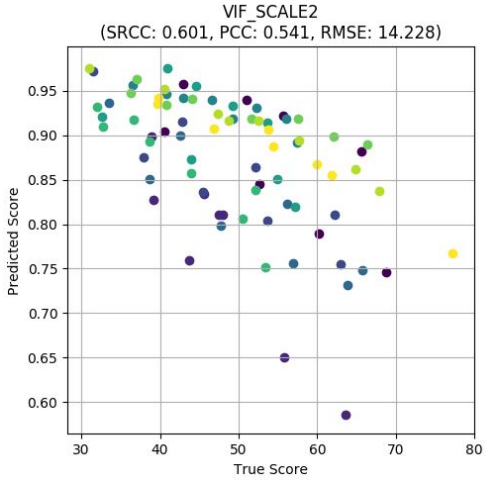
How VMAF works

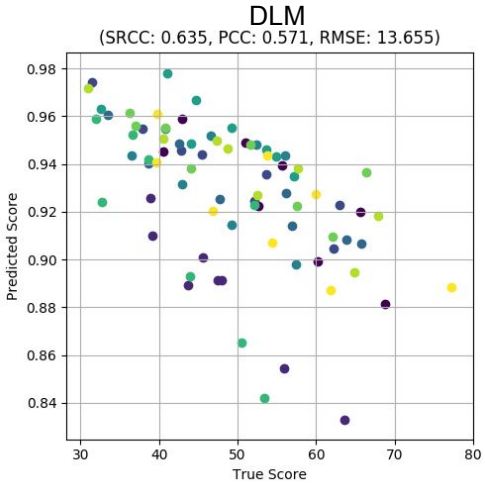


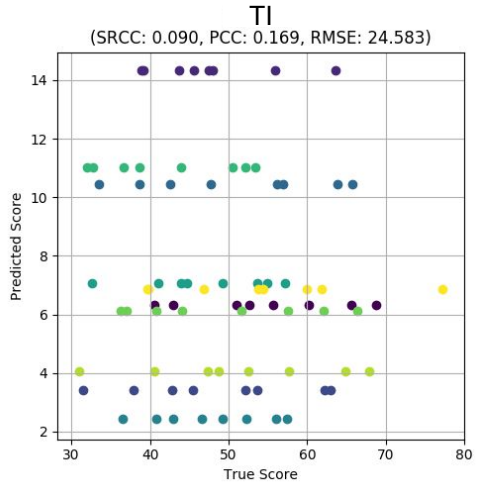
The Power Of Fusion

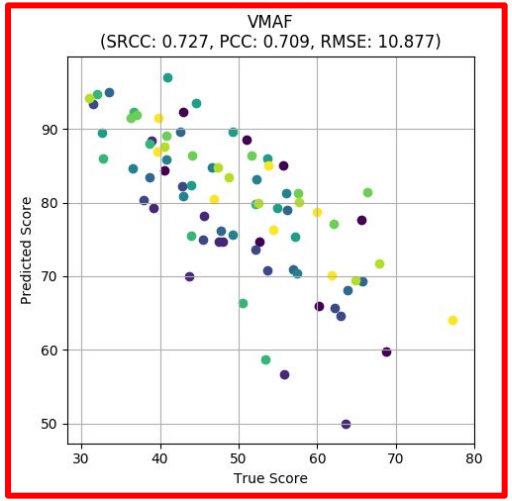










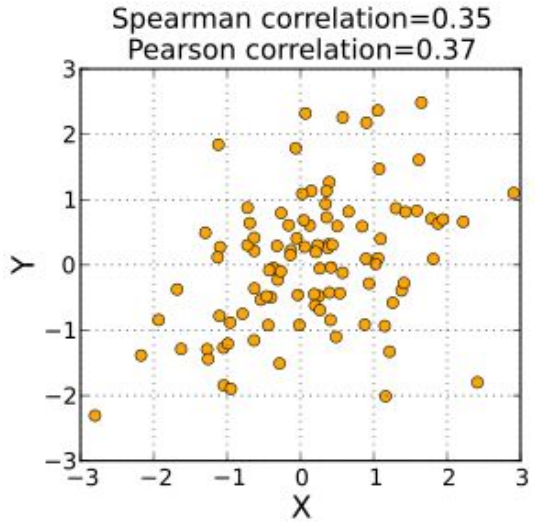


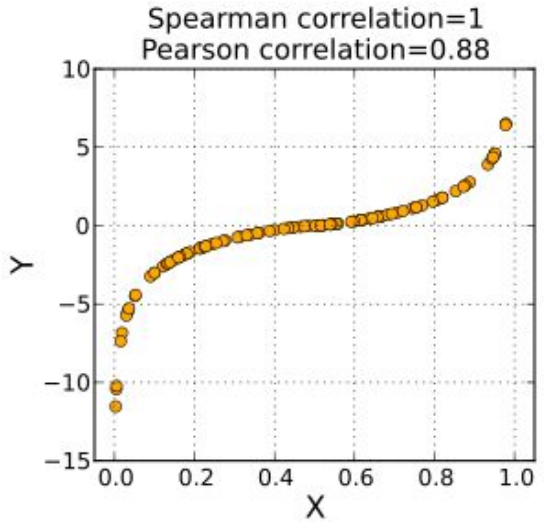
*Tested on **LIVE Video Database**

NETFLIX

Performance evaluation

- SROCC: Spearman Rank Order Correlation Coefficient
- PLCC: Pearson Linear Correlation Coefficient
- RMSE: Root Mean Squared Error [sqrt(mean((y - x)^2))]





Source: Wikipedia

Results

	SRCC	PCC	RMSE
PSNR	0.746	0.725	24.577
SSIM*	0.603	0.417	40.686
MS FastSSIM*	0.685	0.605	31.233
PSNR-HVS*	0.845	0.839	18.537
VMAF v0.6.1	**0.931**	**0.948**	**10.616**

NFLX-TEST Dataset

	SRCC	PCC	RMSE
PSNR	0.416	0.394	16.934
SSIM*	0.658	0.618	12.340
MS FastSSIM*	0.566	0.561	13.691
PSNR-HVS*	0.589	0.595	13.213
VMAF v0.6.1	**0.727**	**0.709**	**10.877**

LIVE Video Database
(Compression-relevant impairments)

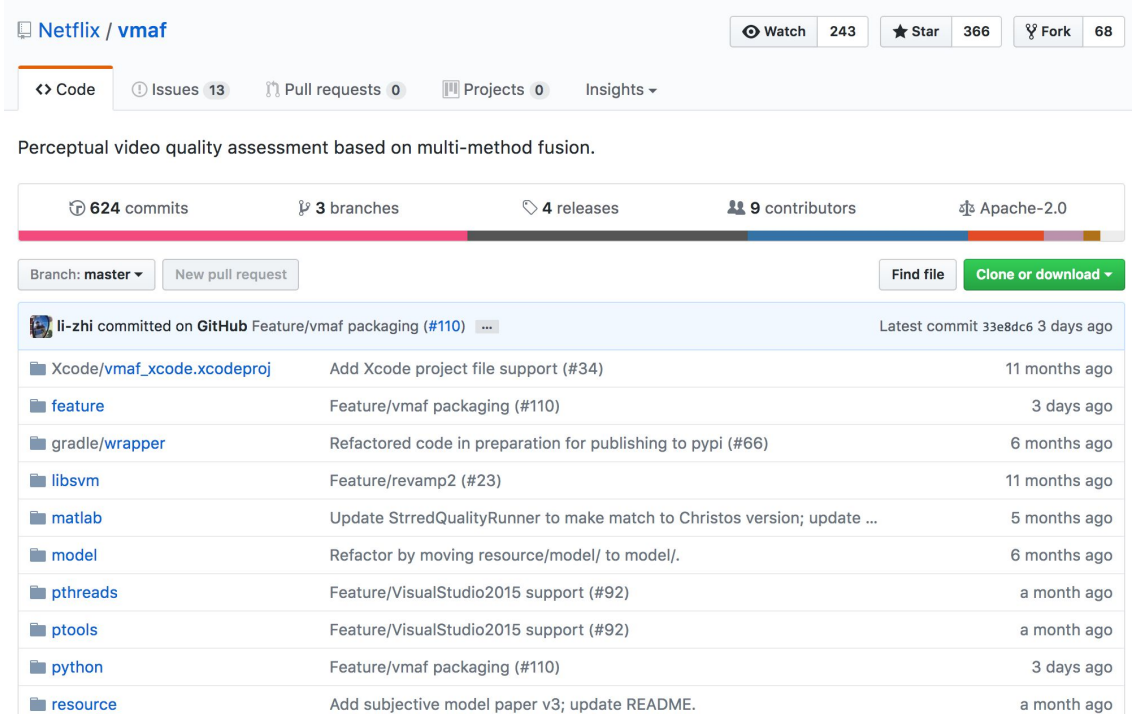
*https://github.com/xiph/daala/tree/master/tools

VMAF: advantages and limitations

- 😀 Evolvability: can easily incorporate new metrics for better accuracy
- 😕 Limited applicability: accuracy and scope are as good as training data
  - Generalization is not guaranteed
  - Default VMAF model: 1080p pristine source from Netflix catalog, living room viewing condition (3*height)
- 😀 Customizability: metrics/training data can be tailored
  - Examples: content, artifacts, viewing conditions
  - Build model for your specific application

VMAF open-source project

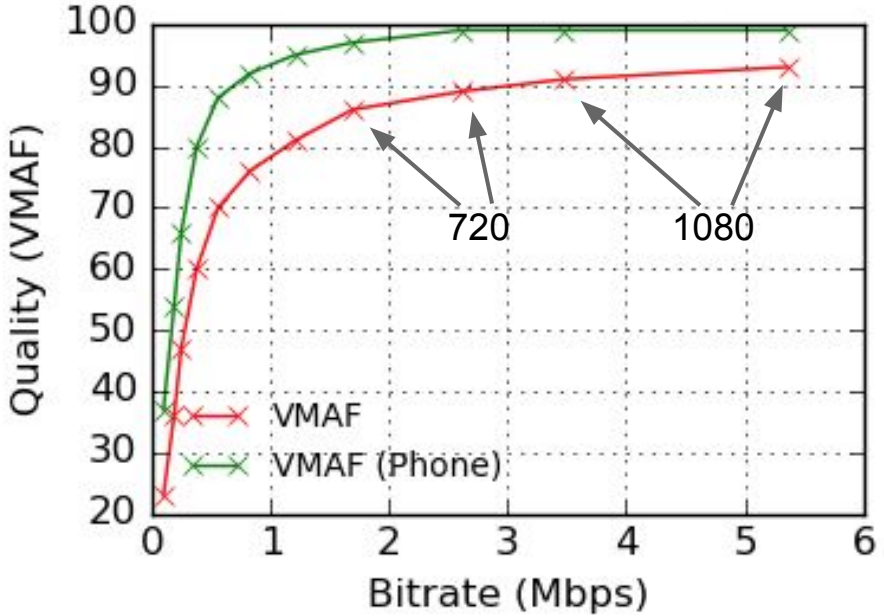
https://github.com/Netflix/vmaf



Usages

- Basic
  - ./run_vmaf: python wrapper calling c executable
  - wrapper/vmafossexec: c++ wrapper
  - ./ffmpeg2vmaf: piping FFmpeg with VMAF
- Advanced
  - ./run_vmaf_training: train a new VMAF model
  - ./run_testing: validate VMAF model on a dataset

VMAF phone model



Predict how the quality of a video is perceived when viewed on a mobile device

Adoption and external contributions

- Adoption
  - Alliance for Open Media (AOM)
  - http://arewecompressedyet.com
  - Academic papers start evaluating/using VMAF
  - ...
- External contributions
  - libvmaf library
  - FFmpeg integration
  - Docker support
  - Windows/Visual Studio support
  - ...

How you can contribute

- Report bugs, request features, implement features
- Integrate new metrics
- Share subjective dataset
- Share trained models
- ... and many more

Backup Slides

How to train a VMAF model

To begin with: run a subjective test

- Example: subjective test for VMAF 0.6.1 (1080p model)
  - **Source**: 23 videos, each 10-sec long, selected from Netflix catalog
  - **Distortion**: each source video is encoded with 6 resolutions up to 1080p, and 3 quality parameters (in total 18 impaired per source)
  - **Subjects**: ~55
  - **Selective sampling**: not all videos were viewed by each subject
  - **Test methodology**: absolute category rating (ACR)
    - Subject is instructed to watch an impaired video and give a rating on a continuous scale from bad to excellent

Collect data in a dataset file

example_raw_dataset.py

```
dataset_name = 'example'

yuv_fmt = 'yuv420p'
width = 1920
height = 1080
ref_score = 100.0

from vmaf.config import VmafConfig

ref_videos = [
    {'content_id': 0, 'content_name': 'checkerboard', 'path': VmafConfig.test_resource_path('yuv', 'checkerboard_1920_1080_10_3_0_0.yuv')},

    {'content_id': 1, 'content_name': 'flat', 'path': VmafConfig.test_resource_path('yuv', 'flat_1920_1080_0.yuv')},
]

dis_videos = [
    {'content_id': 0, 'asset_id': 0, 'os': [100, 100, 100, 100, 100], 'path': VmafConfig.test_resource_path('yuv', 'checkerboard_1920_1080_10_3_0_0.yuv')}, # ref
    {'content_id': 0, 'asset_id': 1, 'os': [40, 45, 50, 55, 60],  'path': VmafConfig.test_resource_path('yuv', 'checkerboard_1920_1080_10_3_1_0.yuv')},

    {'content_id': 1, 'asset_id': 2, 'os': [90, 90, 90, 90, 90],  'path': VmafConfig.test_resource_path('yuv', 'flat_1920_1080_0.yuv')}, # ref
    {'content_id': 1, 'asset_id': 3, 'os': [70, 75, 80, 85, 90],  'path': VmafConfig.test_resource_path('yuv', 'flat_1920_1080_10.yuv')},
]
```

Dataset validation

- ./run_testing PSNR NFLX_dataset_raw.py --cache-result

Train a new model

- Training:
  - ./run_vmaf_training NFLX_dataset_raw.py resource/feature_param/vmaf_feature_v3.py resource/model_param/libsvmnusvr_v3.py **test_model.pkl** --cache-result
- Testing:
  - ./run_testing VMAF LIVEVideo_dataset.py **--vmaf-model test_model.pkl** --cache-result
- Single run:
  - ./run_vmaf yuv420p 576 324 python/test/resource/yuv/src01_hrc00_576x324.yuv python/test/resource/yuv/src01_hrc01_576x324.yuv **--model test_model.pkl** --out-fmt xml