Universal Video Quality (UVQ) in YouTube
open source and production deployment insights

Yilin Wang (yilin@google.com)
Staff Software Engineer
YouTube Media Algorithms Team
YouTube's Requirements for Video Quality Assessment

- **Handling UGC contents**
  - to reflect various quality expectation and sensitivity

- **Supporting no-reference**
  - to support no-ref applications, e.g. monitoring uploads and live streaming
  - also work well in reference-based use cases, and be reliable to non-pristine reference

- **Interpreting quality score**
  - to help people better understand and solve quality issues

---

- high quality expectation v.s. low sensitivity on quality
- original (non-pristine) v.s. transcoded version
- compression artifacts v.s. codec/transmission error
UVQ: a No-Ref based **Interpretable** quality model for UGC

**UVQ Quality Report:**

**Overall quality score in [1, 5]**

- Interpretation of UVQ scores
  - [1, 3.5): relatively low
  - [3.5, 4.2]: medium/fair
  - [4.2, 5]: relatively high
- Noticeable diff: 0.05~0.1 UVQ DMOS
- Score for this example: 3.15 (low quality)

**Quality labels**

- From high level (semantic) to low level (pixel difference)
- Labels for this example
  - Strategy video game,
  - Gaussian blur, Pixelate
  - Medium high compression
UVQ Framework

Inputs:
- ContentNet
- DistortionNet
- CompressionNet

Outputs:
- Video Quality Indicators:
  - content labels
  - distortion types
  - compression level
- Quality Conclusions:
  - quality score

① Self-supervised Learning with Millions of training videos (no ground truth quality scores)

② Supervised Learning
YouTube UGC Dataset
- 1500 sampled from 1.5M videos
- 15 popular content categories

Yilin Wang et al., "Rich features for perceptual quality assessment of UGC videos", CVPR 2021
UVQ has been open sourced!

- **Public link:** [github.com/google/uvq](https://github.com/google/uvq)
- **In the folder**
  - UVQ models + runnable scripts
- **Input**
  - "video_id,length/filepath"
- **Outputs**
  - overall scores + labels + raw features
UVQ Applications in YouTube

Monitor
Production quality
- VOD, Shorts
- Live, TV
- Launch Evaluation

Detector
Specific quality issues
- Release Validation
- Ingestion
- Codec Evaluation

Feature
Useful signals for quality-related services
- Search & Recommendation

Optimizer
Making decision or optimization based on quality
- Transcoding Optimization
- Video Enhancement

Model efficiency becomes critical for large scale applications.
Revisiting the Efficiency of UGC Video Quality Assessment

Yilin Wang, Joong Gon Yim, Neil Birkbeck, Junjie Ke, Hossein Talebi, Xi Chen, Feng Yang, Balu Adsumilli

ICIP 2022
## Current UGC-VQA Research

<table>
<thead>
<tr>
<th><strong>Small Training Set</strong></th>
<th><strong>Huge Model</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dataset</strong></td>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>LIVE-VQC</td>
<td>VSFA</td>
</tr>
<tr>
<td>KonVid-1k</td>
<td>PVQ</td>
</tr>
<tr>
<td>YouTube-UGC</td>
<td>UVQ (CoINVQ)</td>
</tr>
<tr>
<td>LSVQ</td>
<td>Fast-VQA</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Do we really need such high complexity models, given the limited scale of UGC data?
Possibility to Reduce Model Complexity

- **UVQ-lite**: replacing UVQ's backbones with smaller ones
  - D3D -> MoViNet
  - EfficientNet -> MobileNet
- **Significant complexity reduction**
  - Model parameters is reduced by **83.1%**
  - Flops is reduced by **92.1%**

<table>
<thead>
<tr>
<th>Backbone</th>
<th>UVQ</th>
<th>UVQ-lite</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CompressionNet</td>
<td>Content Net</td>
</tr>
<tr>
<td></td>
<td>D3D</td>
<td>Efficient Net-b0</td>
</tr>
<tr>
<td>Parameters(M)</td>
<td>16.382</td>
<td>12.177</td>
</tr>
<tr>
<td>Flops(G)</td>
<td>12.792</td>
<td>4.229</td>
</tr>
</tbody>
</table>

How does the small model UVQ-lite perform?
Model Performance

- UVQ-lite outperforms larger models (VSFA and TLVQM) on LSVQ test set
- UVQ-lite also achieves highest correlations on LSVQ Test 1080p set

<table>
<thead>
<tr>
<th>Model</th>
<th>LSVQ Test</th>
<th>LSVQ Test-1080p</th>
<th>Model Params</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLCC</td>
<td>SRCC</td>
<td>PLCC</td>
</tr>
<tr>
<td>TLVQM</td>
<td>0.774</td>
<td>0.772</td>
<td>0.616</td>
</tr>
<tr>
<td>VSFA</td>
<td>0.796</td>
<td>0.801</td>
<td>0.704</td>
</tr>
<tr>
<td>PVQ (w/o v-patch)</td>
<td>0.816</td>
<td>0.814</td>
<td>0.708</td>
</tr>
<tr>
<td>UVQ</td>
<td>0.809</td>
<td><strong>0.815</strong></td>
<td>0.717</td>
</tr>
<tr>
<td>UVQ-lite</td>
<td>0.798</td>
<td>0.806</td>
<td><strong>0.718</strong></td>
</tr>
</tbody>
</table>

t-SNE Visualized UVQ Features (Distortion Features for LSVQ)

LSVQ dataset

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>28.1K</td>
</tr>
<tr>
<td>Test</td>
<td>7.4K</td>
</tr>
<tr>
<td>Test-1080p</td>
<td>3.5K</td>
</tr>
<tr>
<td>Total</td>
<td>40K</td>
</tr>
</tbody>
</table>

The coverage of existing dataset still needs improvement.
Summary

- UVQ is open source now!
  - [github.com/google/uvq](https://github.com/google/uvq)

- Efficiency of UGC-VQA is important and insufficiently addressed
  - existing datasets may not fully represent the complexity of UGC video quality
  - we can design more efficient models with better generalizability for UGC-VQA tasks

- [Frontiers' research topic](https://www.frontiersin.org/) on image/video quality
  - deadline: Aug 1st, 2023
Thanks!