

CLiF-VQA: Enhancing Video Quality Assessment by Incorporating High-Level Semantic Information related to Human Feelings

1 DETAILS ON SUBJECTIVE EXPERIMENTS

We will detail the setup of the experiments (shown in Fig. 1 of the paper) on the effects of human feelings on the judgments of the human visual system (HVS) conducted in Sec. 1 of the paper. Specifically, we choose Honor Magic3 as the shooting device. An expert with professional shooting skills takes the video and the video resolution obtained is 1920×1080 . For the subjective experiment, we ask 10 experts to evaluate the quality of the video. While evaluating the quality of the video, the experts only need to give a rating as to which of the two videos is better in terms of quality. The better quality video gets a score of 1 while the poor quality video gets a score of 0. Then we calculate the mean of all the scores given by the experts as the quality score of the video. In addition, we design four pairs of antonyms as CLIP prompts to explore the consistency of CLIP with human feelings. Specifically, we use one pair of antonyms at a time as prompts to extract features from the video, and then we use the CLIP with prompts to perform feature extraction at 10 different locations in the video frame. Finally, the average of all feature values corresponding to a particular prompt is computed as the feature of the video.

2 DETAILS ON EXPLORING THE PERFORMANCE OF CLIP IN VQA

In Sec. 3 of the paper, the model we designed for exploring the performance of CLIP in video quality assessment is shown in Fig 1. Specifically, we apply the architecture of the classic VQA model VSFA, but we replace the feature extraction module used in VSFA with our CLIP-based semantic feature extraction module, leaving the other modules intact.

Table 1: Feature extraction models for NR-VQA.

Methods	Venus	Feature Extraction
VSFA[4]	ACMMM'19	ResNet-50
GST-VQA[1]	TCSVT'21	VGG-16
MD-VQA[11]	CVPR'23	EfficientNet-V2

Table 2: Performance comparison of different feature extraction methods. 'o' denotes using only objective description (35), 's' denotes using only subjective description (17), and 'a' denotes using both objective and subjective description.

Method	SROCC \uparrow	KROCC \uparrow	PLCC \uparrow	RMSE \downarrow
VGG-16	0.740	0.542	0.742	0.412
ResNet-50	0.771	0.575	0.775	0.398
EfficientNet-V2	0.797	0.599	0.795	0.387
CLIP _o	0.782	0.590	0.786	0.401
CLIP _s	0.499	0.343	0.531	0.528
CLIP _a	0.814	0.622	0.826	0.375

We compare the CLIP-based feature extraction method with three other feature extraction methods (ResNet-50 [2], VGG-16 [6], EfficientNet-V2 [8]) that are widely used for VQA, as shown in Tab. 1. Specifically, we compare the performance of different feature extraction methods by keeping the temporal-memory effect modeling module unchanged. In the comparison experiment, we use a total of 52 descriptions related to human objective and subjective feelings as prompts of CLIP. The results of the comparison experiments are shown in Tab. 2. The results here correspond to the bar chart in the paper. Detailed information about the prompts we use will be presented in the next section.

3 DETAILED INFORMATION ON PROMPT SETTINGS

3.1 Prompt Design on Human Feelings

A total of 52 descriptions related to human feelings are designed as prompts, including 35 descriptions related to human objective feelings and 17 descriptions related to human subjective feelings. The details of the descriptions are as follows.

Objective Descriptions: ["good image block", "bad image block", "noisy image block", "hazy image block", "dark image block", "bright image block", "blurry image block", "over exposure image block", "sharp image block", "colorful image block", "dull image block", "high contrast image block", "low contrast image block", "image block without noise", "image block without blur", "image block with additive gaussian noise", "image block with noise in color compression", "image block with spatially correlated noise", "image block with masked noise", "image block with high frequency noise", "image block with impulse noise", "image block with quantization noise", "image block with gaussian blur", "image block with motion blur", "image block with bokeh blur", "uniform color image block", "uneven color image block", "image block with chromatic aberration", "image block without chromatic aberration", "image block with distortions", "image block without distortions", "uniform illumination image block", "unevenly illuminated image block", "image block with sharpness loss", "image block without sharpness loss"]

Subjective Descriptions: ["light-hearted image block", "depressing image block", "comfortable image block", "uncomfortable image block", "sad image block", "sentimental image block", "fearful image block", "exciting image block", "satisfactory image block", "calming image block", "fascinating image block", "interesting image block", "impatient image block", "tense image block", "puzzling image block", "delightful image block", "outrageous image block", "disgusting image block"]

3.2 Experiments on Different Prompt Words

As you can see the prompt word we use is `<image block>`, which we chose after a lot of comparison experiments. Since the performance of CLIP is affected by the prompt word [5], choosing a suitable prompt word can make CLIP perform better in our task. Therefore, we design six alternative prompt words (`<photo>`, `<photo block>`, `<video frame>`, `<video frame block>`, `<image>`, `<image block>`).

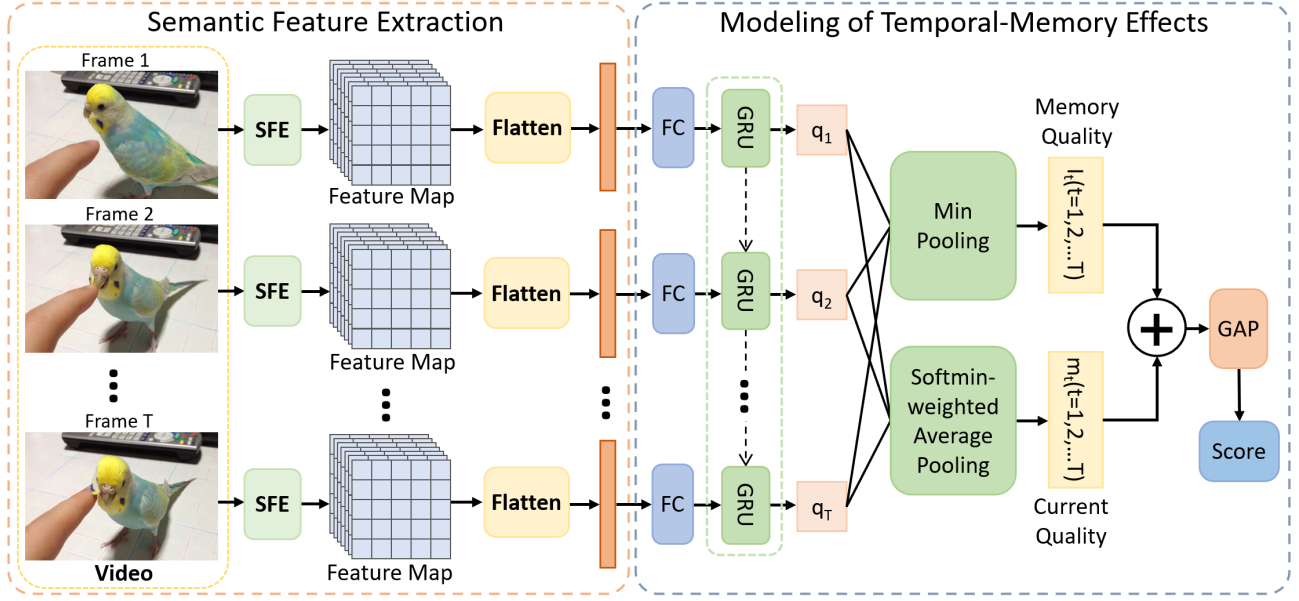


Figure 1: Model architecture (mentioned in Sec. 3) used when exploring performance of CLIP in VQA. More details about the model can be found in model VSFA[4].

Of these, *<photo>*, *<video frame>* and *<image>* are widely used prompt words for CLIP. Since we use CLIP to extract semantic features from multiple region blocks in a video frame, we consider that adding *<block>* to prompt words might improve the performance of CLIP, and thus design prompt words *<photo block>*, *<video frame block>* and *<image block>*.

Table 3: Performance using different prompt words.

Prompt word	SROCC↑	KROCC↑	PLCC↑	RMSE↓
photo	0.796	0.601	0.799	0.402
photo block	0.806	0.597	0.806	0.388
video frame	0.789	0.568	0.781	0.440
video frame block	0.788	0.573	0.779	0.435
image	0.803	0.596	0.805	0.392
image block	0.814	0.622	0.826	0.375

We perform our experiments on the dataset KoNViD-1K [3] containing 1200 videos with a resolution of 960×540 . And we perform 85 semantic feature extractions at different locations on each frame of the video. The experimental results are shown in Tab 3. The results indicate that using *<image block>* as the prompt word has the best performance, so we choose to use it as the prompt word.

3.3 Experiments on the Number of Prompts

Although we design 52 prompts, we also explore the performance of the model (shown in Fig. 1) in VQA when using fewer prompts. We experiment with five additional sets of prompts with different numbers, with the same experimental setup as in Sec. 3.2, as shown in Tab. 4. The results show that increasing the number of prompts

Table 4: Performance using different number of prompts. 'ob' denotes objective description, 'sub' denotes subjective description, 'f' denotes randomly using the half of descriptions, 'a' denotes all descriptions.

Description	SROCC↑	KROCC↑	PLCC↑	RMSE↓
only-ob-f	0.751	0.550	0.755	0.422
only-ob-a	0.782	0.590	0.786	0.401
only-sub-f	0.358	0.242	0.378	0.576
only-sub-a	0.499	0.343	0.531	0.528
obj-sub-f	0.803	0.602	0.798	0.392
obj-sub-a	0.814	0.622	0.826	0.375

can improve the performance of CLIP in the VQA task. This suggests that using more prompts might enable CLIP to capture more semantic features of the video. This may mean that using more cues than the 52 prompts we designed might further improve the performance of CLIP, which remains to be verified. We next consider designing more prompts to test this hypothesis.

4 MORE INFORMATION ON MODEL TRAINING

During pre-training, we train on the LSVQ [10] dataset for 30 epochs. After each epoch of training, we validate the performance of the model on multiple small datasets (KoNViD-1k [3], LIVE-VQC [7], YouTube-UGC [9]). We save the parameters from that time when the model performed best on a small dataset, and then use the parameters as initial parameters when fine-tuning on that dataset.

REFERENCES

- [1] Baoliang Chen, Lingyu Zhu, Guo Li, Fangbo Lu, Hongfei Fan, and Shiqi Wang. 2021. Learning generalized spatial-temporal deep feature representation for no-reference video quality assessment. *IEEE TCSVT* 32, 4 (2021), 1903–1916.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.
- [3] Vlad Hosu, Franz Hahn, Mohsen Jenadeleh, Hanhe Lin, Hui Men, Tamás Szirányi, Shujun Li, and Dietmar Saupe. 2017. The Konstanz natural video database (KoNViD-1k). In *QoMEX*. IEEE, 1–6.
- [4] Dingquan Li, Tingting Jiang, and Ming Jiang. 2019. Quality assessment of in-the-wild videos. In *ACM MM*. 2351–2359.
- [5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*. PMLR, 8748–8763.
- [6] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [7] Zeina Sinno and Alan Conrad Bovik. 2018. Large-scale study of perceptual video quality. *IEEE TIP* 28, 2 (2018), 612–627.
- [8] Mingxing Tan and Quoc Le. 2021. Efficientnetv2: Smaller models and faster training. In *ICML*. PMLR, 10096–10106.
- [9] Yilin Wang, Sasi Inguva, and Balu Adsumilli. 2019. YouTube UGC dataset for video compression research. In *MMSP*. IEEE, 1–5.
- [10] Zhenqiang Ying, Maniratnam Mandal, Deepti Ghadiyaram, and Alan Bovik. 2021. Patch-VQ: Patching Up the video quality problem. In *CVPR*. 14019–14029.
- [11] Zicheng Zhang, Wei Wu, Wei Sun, Danyang Tu, Wei Lu, Xiongkuo Min, Ying Chen, and Guangtao Zhai. 2023. MD-VQA: Multi-dimensional quality assessment for UGC live videos. In *CVPR*. 1746–1755.