NDNetGaming - Development of a No-Reference Deep CNN for Gaming Video Quality Prediction

Saman Zadtootaghaj

Quality and Usability Lab (TU Berlin), (in collaboration with Fraunhofer HHI)

What we are doing

- Quality assessment for gaming service
  - Cloud gaming, e.g. Stadia, Nvidia Geforce Now
  - Passive gaming streaming, e.g. Twitch tv, Youtube gaming
- Major focus on cloud gaming planning model for ITU-T (G.OMG)
  - Parametric model
- Signal based models
  - NR-metrics
    - Machine learning based

Cloud Gaming

Special encoding and network protocol

- Latency
  - Capturing RGB data from frame buffer (front buffer) without any involvement from OpenGL/Direct3D
  - Using GPU hardware accelerator engines for video encoding/decoding
  - Fixed macroblock size for fast encoding
- Packet loss (concealment)
  - Designing task-specific network protocol such as reliable UDP
- Encoding setting
  - CBR, short GoP, …

Gaming Content

Special Temporal and Spatial Information

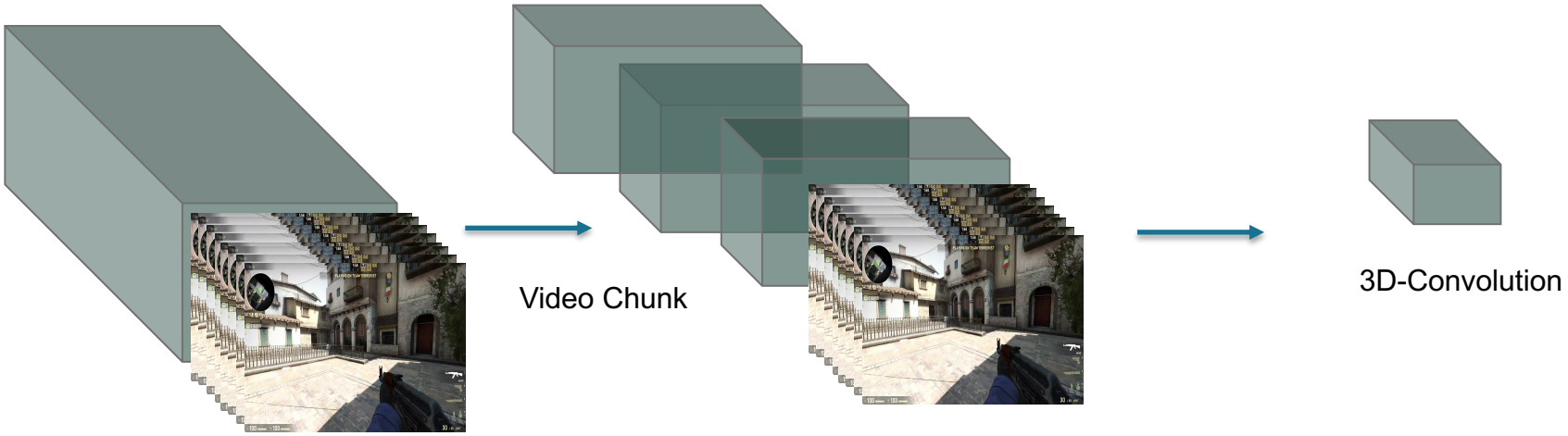
- Game is a **rule-based** system that has special characteristics.
- A game is usually constructed from a **pool of predesigned objects** which result in different level of details.
- A game has a **certain level of abtsraction**, and that does not vary much during the gameplay
- Many games have **specific motion pattern**, e.g. racing game or side scrolling games.

Spatial and temporal features

Game	Original Frame	MAD heatmap	PPSNR with threshold of 35	Heatmap of Average of SI	Heatmap of Variation of SI	Heatmap of Average of TI	Heatmap of Variation of TI
Doat 2							
LoL							
PC (Out-side car)							
PC (In-side car)							
CSGO							

Video quality assessment using CNN

- Two types of Convolution can be used
  - 2D and 3D Convolution (frame or video level)
  - How to make it work on the video level?



Frames level quality assessment

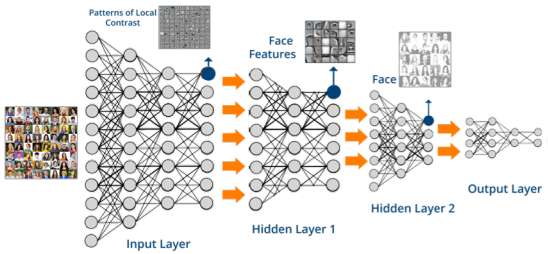
- No Dataset available for CGI content
  - We used VMAF as quality indicator of each frame (similar to DeViQ [1])
  - The idea is not to predict the VMAF but to pretrain the network on a reliable metric and retrain some layers based on the subjective results
  - There might be a difference between the perceived quality on the image level and video level
    - Employ the temporal pooling methods

Transfer Learning

Freeze all layers Train the whole network

Figure from [3]. Kaiming, et al. Deep residual learning for image recognition

The structure of the framework



Retrain network based on an objective metric

Which CNN model to use

Fine-tune the model based on subjetive data

What about other video dimensions

Which video metric to use

Blockiness, Blurr

Take different patterns of patches

Temporal pooling

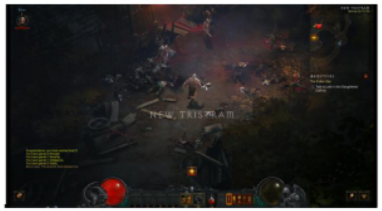
Video Quality

Sample Snapshot of Recorded Sequences

	GVSET	KUGVD
Influencing Factors	Resolution, Bitrate	Resolution, Bitrate
Preset	Veryfast	Veryfast
Number of stimuli	90	90
Encoding	FFmpeg, h264, CBR	FFmpeg, h264, CBR
Number of source sequence	24 (6 used in subjective test)	6

Sample Snapshot of Recorded Sequences

(a) Counter Strike: Global Offensive



(b) Diablo III

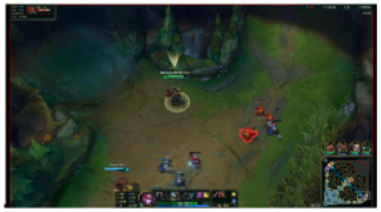
(c) Dota 2

(d) FIFA 2017

(e) H1Z1: Just Kill

(f) Hearthstone

(g) Heroes of the Storm



(h) League of Legends

(i) Project Cars

(j) PlayerUnknown's Battleground

(k) Starcraft 2

(l) World of Warcraft

The structure of the framework

Training | Validation | Goal | Output

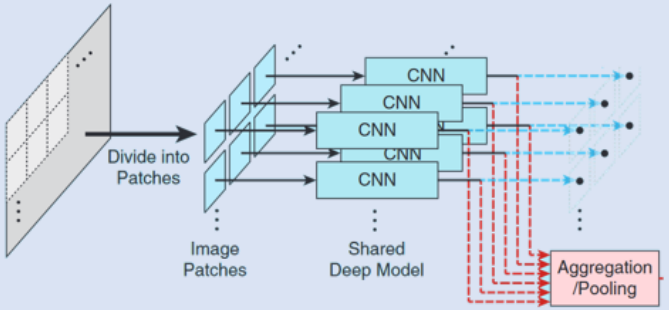
1. Fundamental Design Phase

Find best network design by testing four different architectures, and varying the number of frames and

243.000 images with VMAF scores (GVSET)

210.600 images with VMAF scores (GVSET + KUVGD)

initial design of network (frame-level)



2. Fine-Tuning Phase

Subjective ratings of 164 images (GISET)

Custom Cross-validation

Improve model performance by applying transfer learning based on subjective image ratings

final frame-level quality metric

3. Video Quality Prediction Phase

Subjective ratings of 90 videos (GVSET)

Subjective ratings of 90 videos (KUGVD)

testing different pooling methods for best prediction of video quality

final NDNetGaming metric

Training

Based on VMAF

- We retrained only 25 %, 50 % or 75 % of total trainable weights for four CNNs
- Training set: 243.000 frames

	MobileNetV2	DenseNet-121	Xception	ResNet50
25 %	9.59	7.58	7.33	7.60
50 %	7.98	6.84	7.25	7.34
75 %	7.34	6.74	7.29	6.71
		8,062,504		25,636,712

Required number of layers

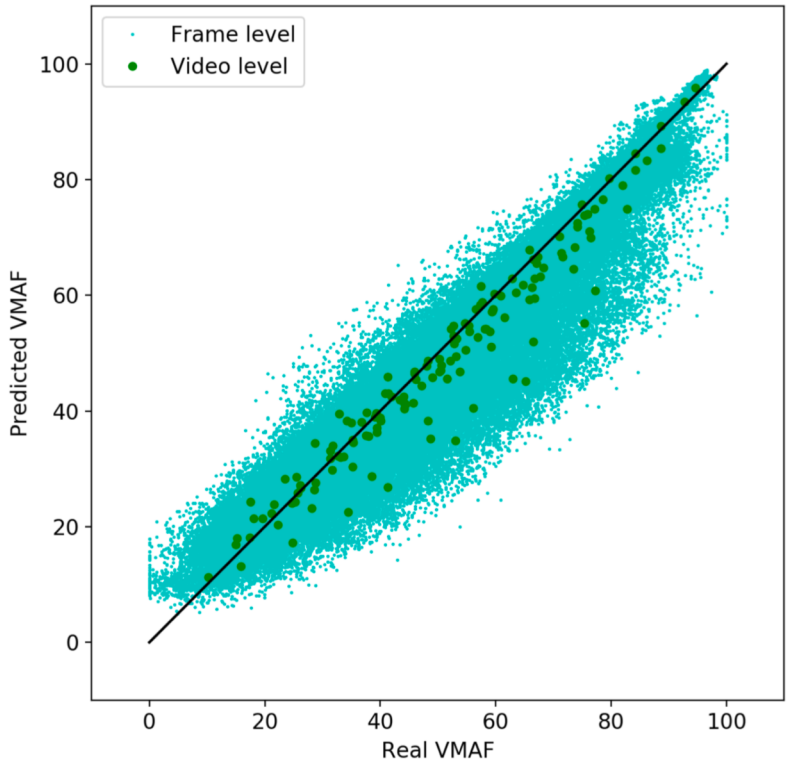
- The DenseNet-121 architecture consists of four blocks, each containing between 12 and 48 convolutional layers

Dense Blocks	Number of layers	Number of weights	RMSE	SRCC
4	120	7039 k	8.11	0.925
3 ½	113	6878 k	7.02	0.942
3	107	6657 k	6.74	0.945
2 ½	94	6268 k	6.77	0.946
2	82	5594 k	6.84	0.942
1 ½	57	4461 k	6.82	0.946
1	33	2191 k	7.22	0.939
½	16	1233 k	7.39	0.936
0	0	1 k	10.60	0.870

Best model trained for VMAF Prediction

Scatter plot of actual VMAF and predicted VMAF values on frame and video level of KUGVD dataset

RMSE: 7.07 in frame level
RMSE: 5.47 in video level



Frame Level: R^2: 0.88 RMSE: 7.07 PCC: 0.946 SRCC: 0.945
Video Level: R^2: 0.92 RMSE: 5.47 PCC: 0.967 SRCC: 0.965

The structure of the framework

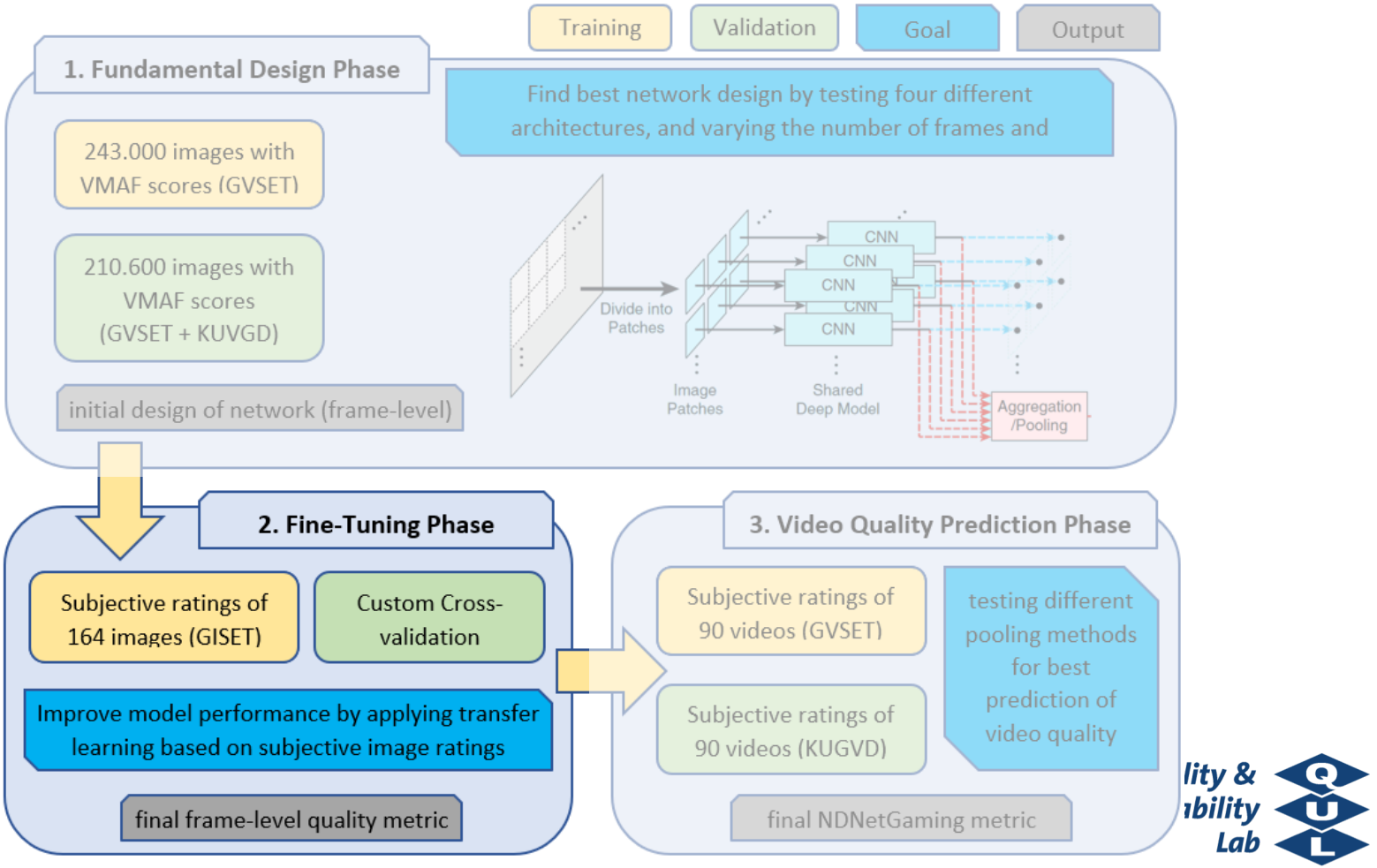


Image quality dataset - GISET

- We selected 164 frames from 18 different video sequences
  - 3 resolution (Unlcearness) and 10 bitrates (Fragmentation)
  - Selected multiple source frames (together with 3 distorted) from each game (41 reference frames)
  - Minimum 2 source frame from each game
  - Distribution of quality levels
    - Selection of frames was based on VMAF values ~ ranges from 90 to 25

Image Quality Dataset - GISET

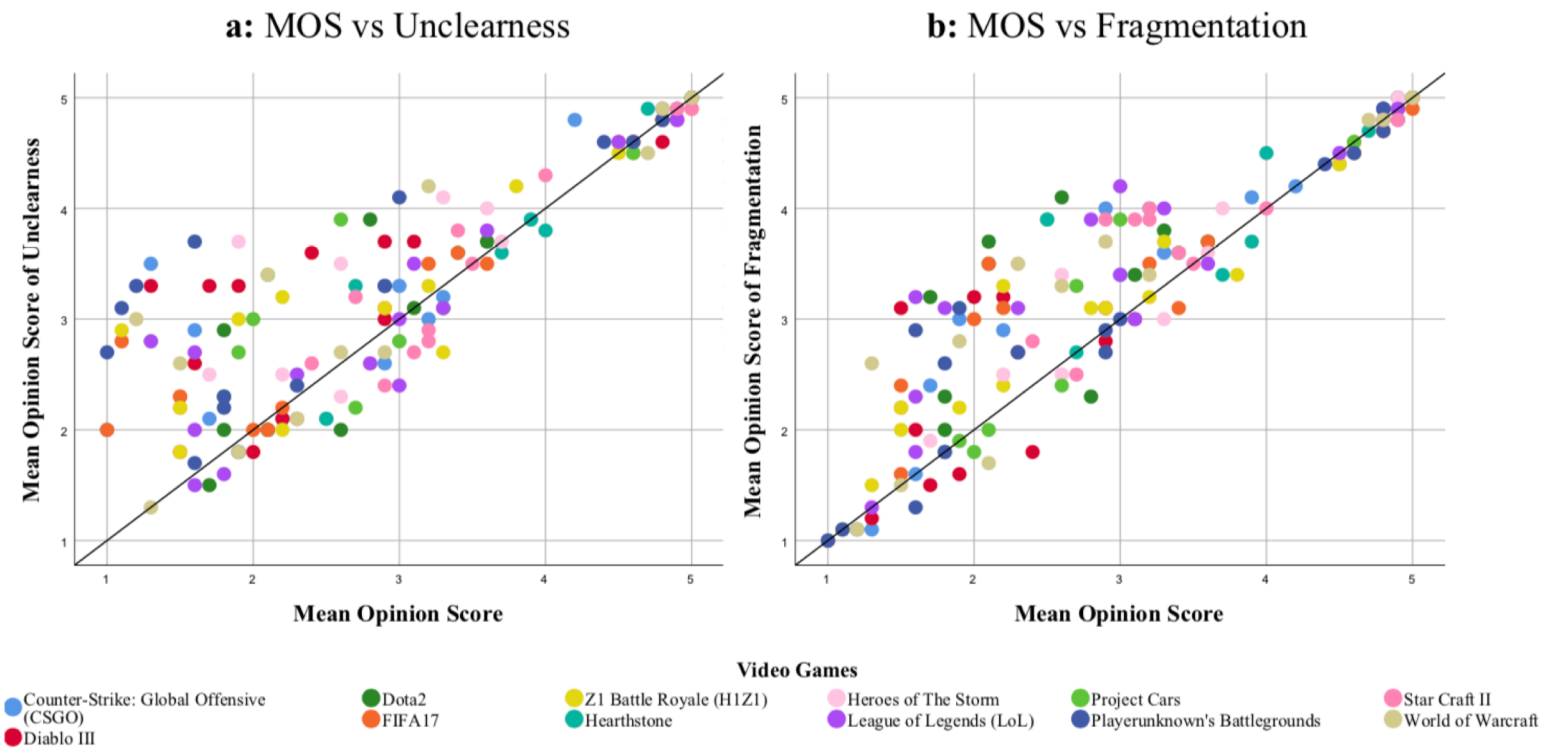
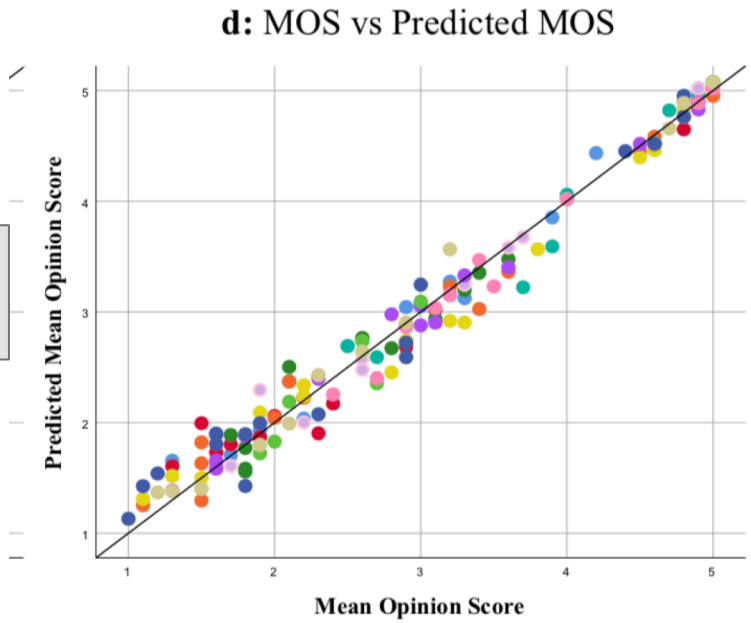


Image Quality Dataset - GISET

$$VQ_{Estimated} = -1.073 + 0.657 \times VF + 0.573 \times VU$$

PCC: 0.98 - RMSE: 0.154

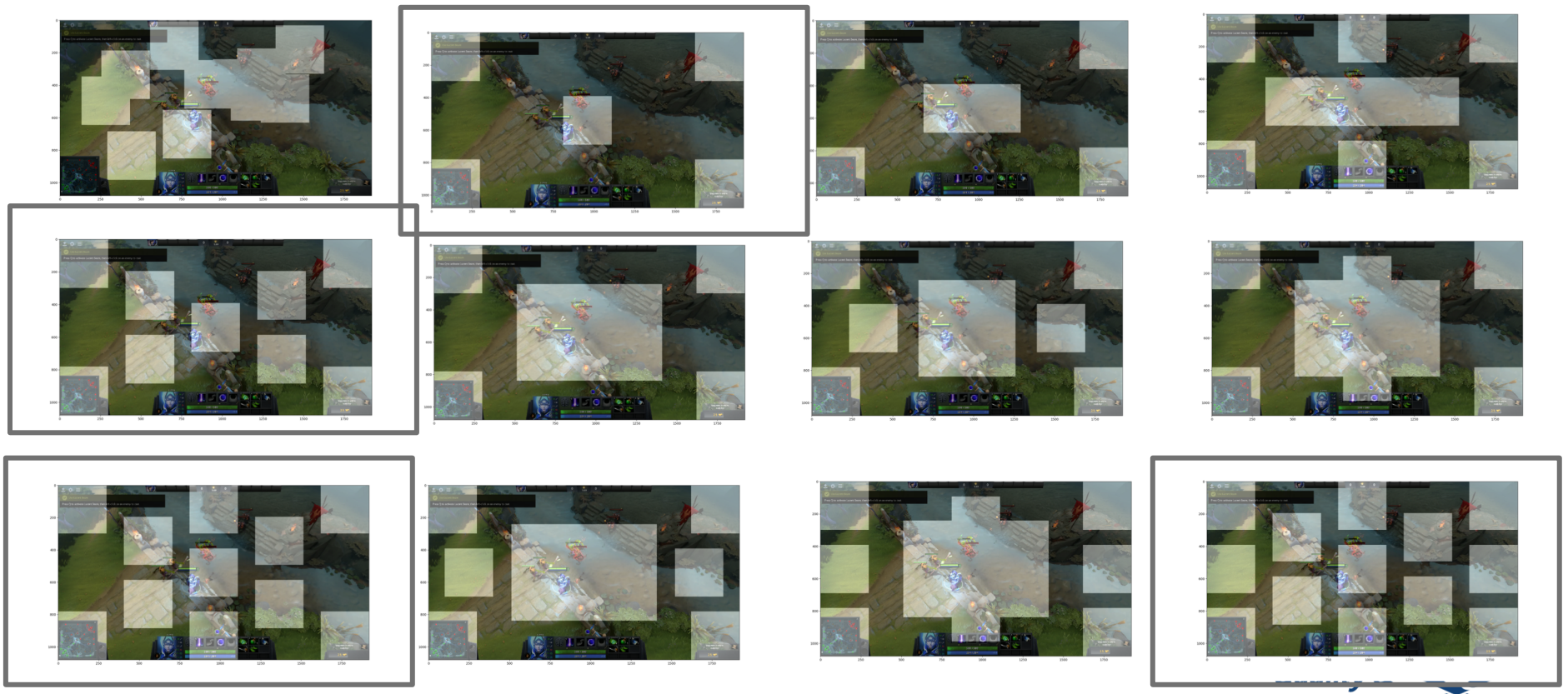


Fine-tuning Phase

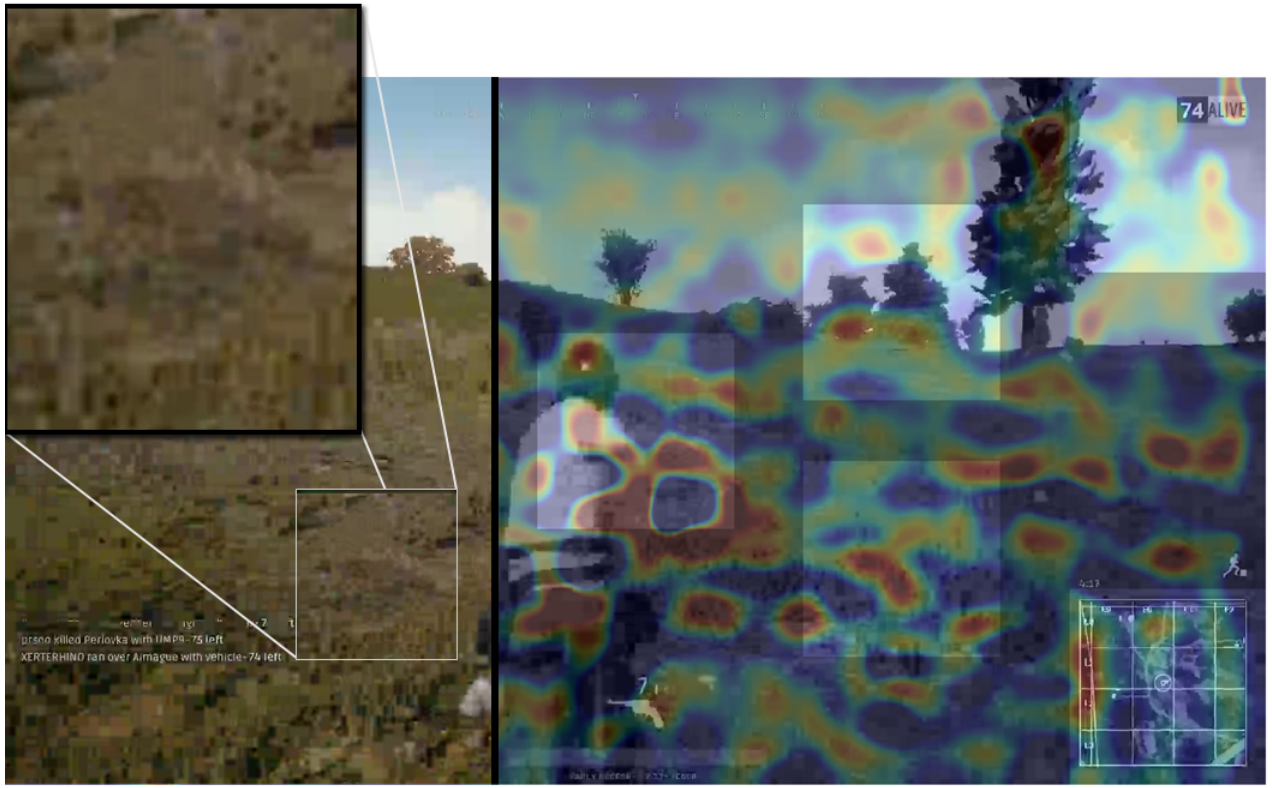
- DMOS was used in training process
- Leave-one-out cross-validation was employed where for every iteration of training the network, we kept one game completely out of training process
- RMSE and SRCC for different numbers of patches used for testing the model:

Number of Patches	RMSE	SRCC
5	0.390	0.953
7	0.374	0.957
9	0.380	0.954
11	0.381	0.958
13	0.377	0.953

Different Patch Patterns Selection (fine-tuning)

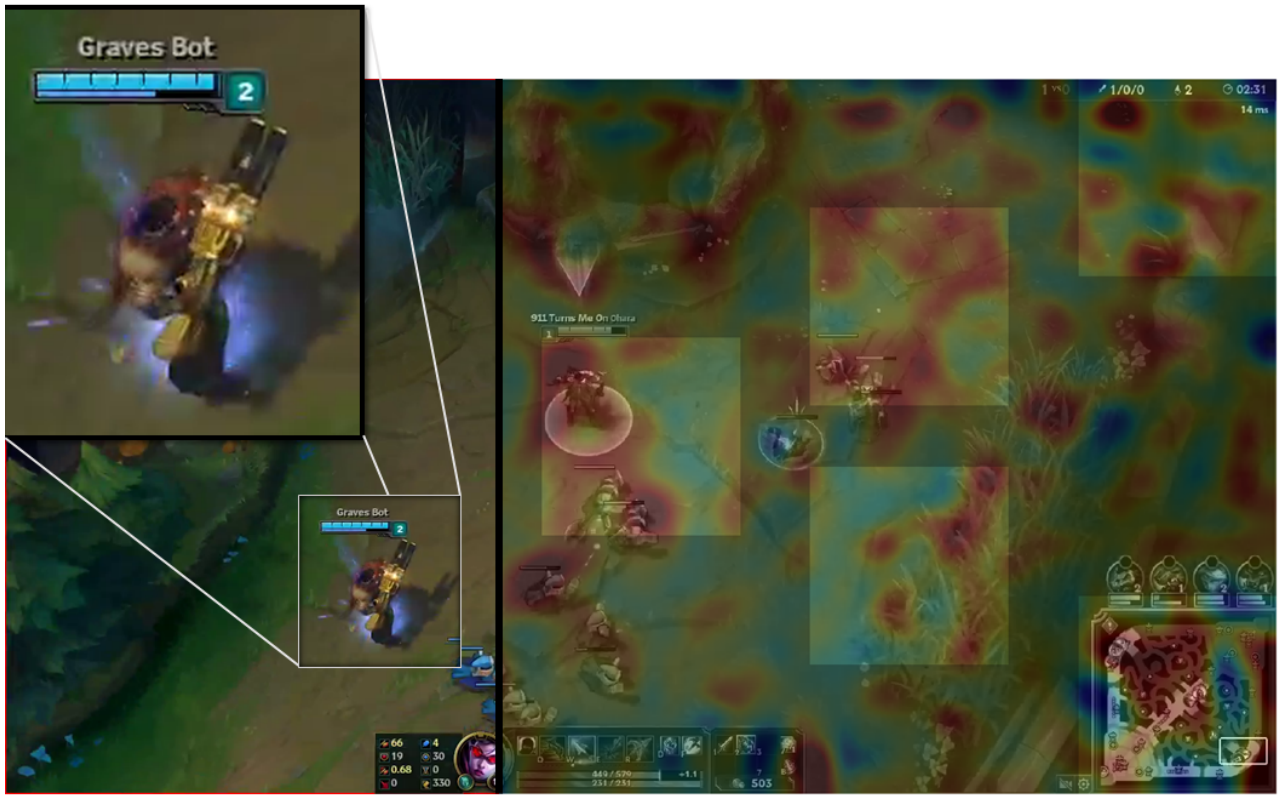


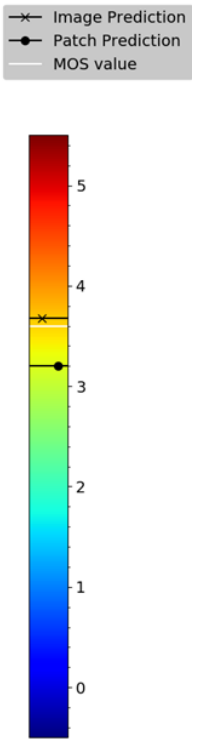
Local Quality Predictions



Quality & Usability Lab

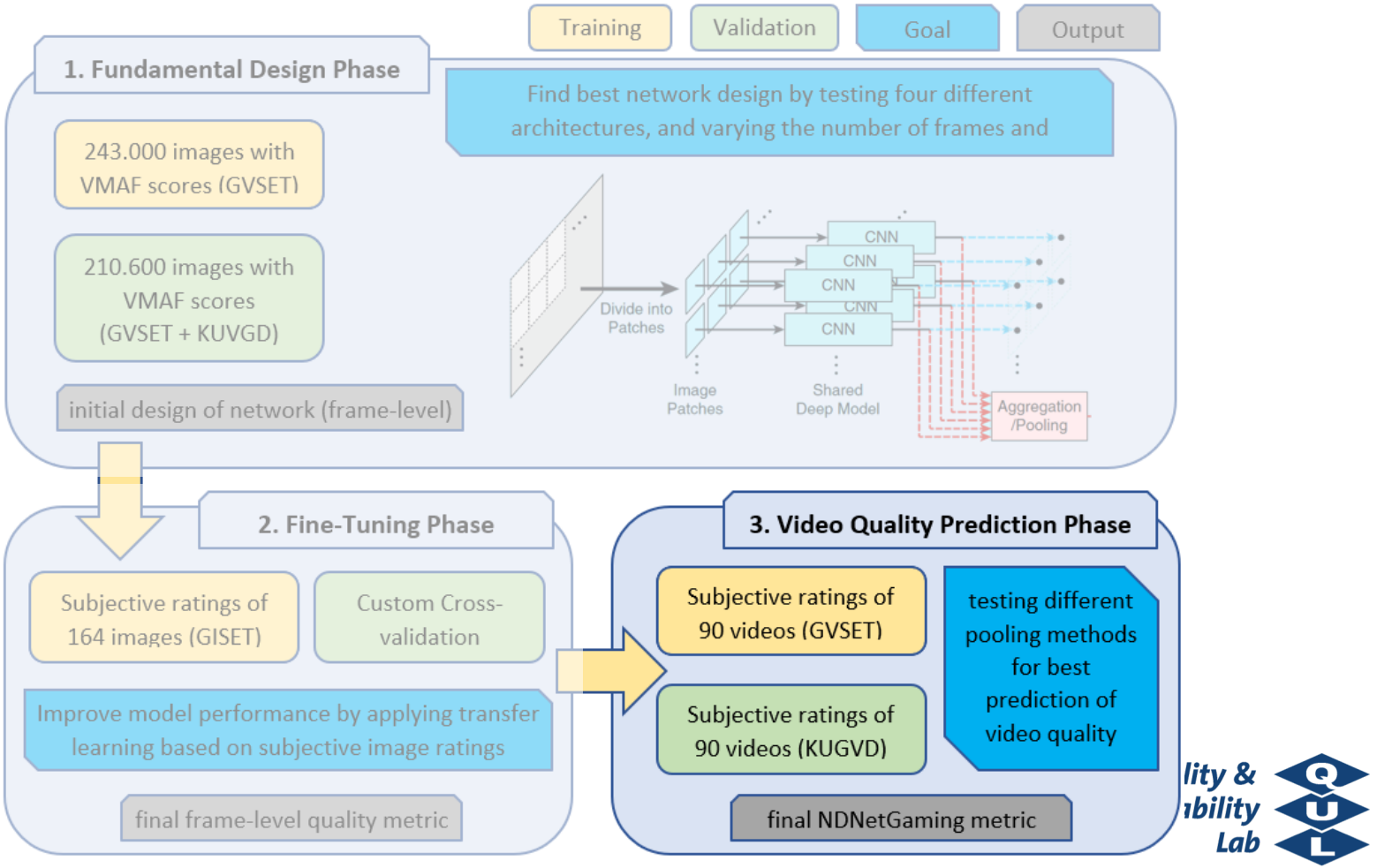
Local Quality Predictions





Quality & Usability Lab

The structure of the framework



Video Quality Prediction Phase

No significant improvement compared to average pooling has been observed

We tried to reduce the effect of temporal masking in two steps:

Step 1

$$ewma_{TI} = smooth_{ewma}(std_{space}[M_n(i,j)])$$

$$weights_{frame} = ewma_{TI} / sum_{time}[ewma_{TI}]$$

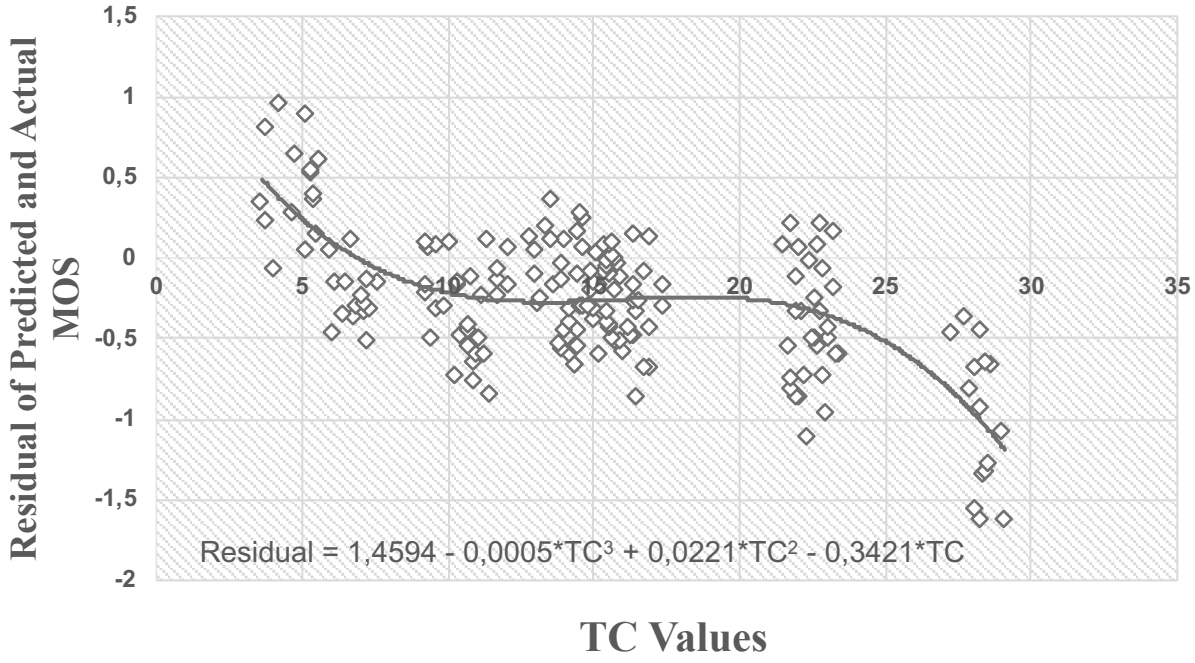
$$inverse_{weights} = \frac{(1 - P(F = 1))}{1 - P(F = 1|W = w)}$$

Step 2

$$TC = mean_{time}[std_{space}[M_n(i,j)]]$$

$$NDNG_{Temporal} = C_1 + C_2 \times NDNG + C_3 \times TC^3 - C_4 \times TC^2 + C_5 \times TC$$

Video Quality Prediction Phase



$$NDNG_{Temporal} = C_1 + C_2 \times NDNG + C_3 \times TC^3 - C_4 \times TC^2 + C_5 \times TC$$

Video Quality Prediction Phase

$$NDNG_{Temporal} = C_1 + C_2 \times NDNG + C_3 \times TC^3 - C_4 \times TC^2 + C_5 \times TC$$

	C_1	C_2	C_3	C_4	C_5
eq_1	−1.99	1.097	0.000 69	−0.031	0.43
eq_2	−0.532	1.116	0.000 11	−0.0043	0.084
eq_3	−1.71	1.107	0.000 53	−0.024	0.353

Coefficients of temporal pooling methods, eq1, eq2 and eq3 are trained based on GVSET, KUGVD and both respectively.

Image Quality Assessment

- **L**IVE **M**ultiply **D**istorted Image Quality Data**set** and **L**IVE **P**ublic-**D**omain Subjective Image Quality Data**set** (the first release)

Metrics		LMDSET		LPDSET	
		PCC	SRCC	PCC	SRCC
FR Metrics	PSNR	-0.69	-0.64	0.80	0.93
	SSIM	-0.58	-0.61	0.92	**0.94**
NR Metrics	BRISQUE	0.57	0.43	-0.93	-0.92
	NIQE	**0.87**	-0.62	-0.92	-0.89
	PIQE	0.82	**0.77**	-0.90	-0.87
	NDNetGaming	-0.77	-0.68	**0.95**	0.92

Video Quality Assessment

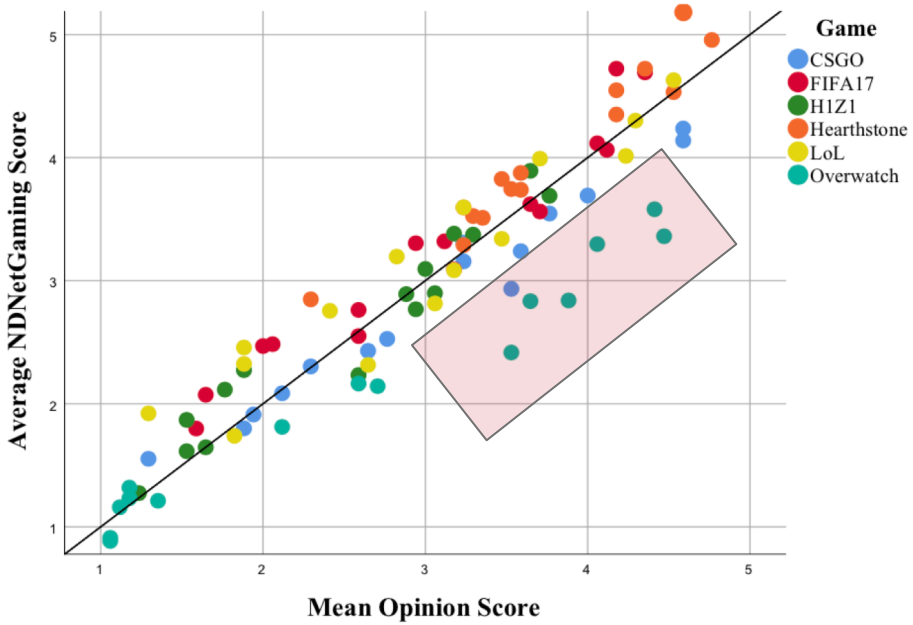
Metrics		Netflix Public Dataset		LIVE-NFLX-I	
		PCC	SRCC	PCC	SRCC
FR Metrics	PSNR	0.64	0.66	0.49	0.27
	SSIM	0.69	0.76	0.24	-0.10
	VMAF	**0.93**	**0.91**	0.78	0.24
NR Metrics	BRISQUE	-0.77	-0.76	-0.65	-0.68
	NIQE	-0.83	-0.81	-0.67	-0.28
	PIQE	-0.78	-0.80	**-0.85**	**-0.83**
	NDNetGaming	0.89	0.85	0.82	0.71

Video Gaming Quality Assessment

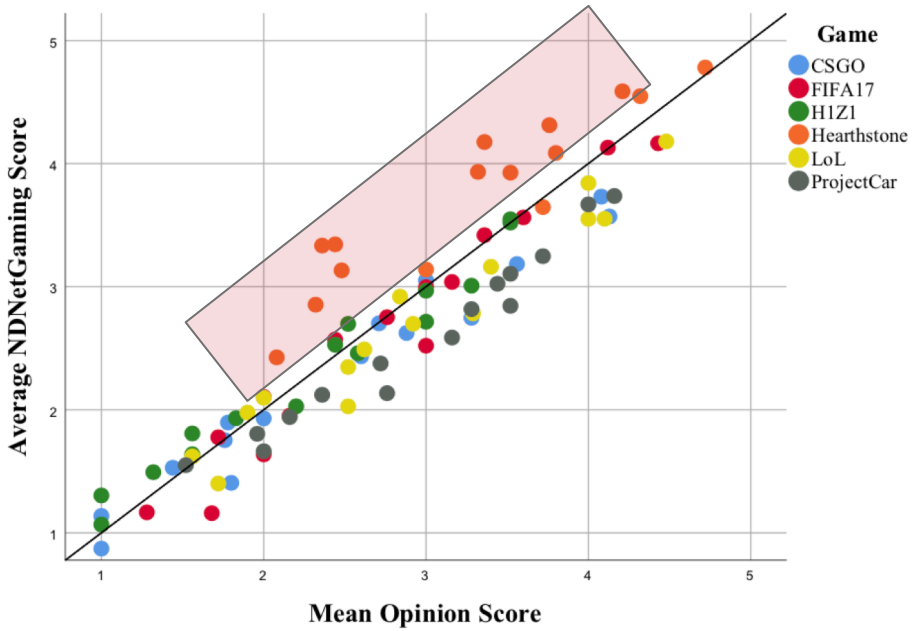
Metrics		GVSET		KUGVD	
		PCC	SRCC	PCC	SRCC
FR Metrics	PSNR	0.75	0.74	0.80	0.78
	SSIM	0.80	0.80	0.89	0.88
	VMAF	0.87	0.87	0.92	0.92
RR Metrics	ST-RREDOpt	-0.75	-0.77	-0.73	-0.72
	SpEEDQA	-0.75	-0.77	-0.70	-0.70
NR Metrics	BRISQUE	-0.44	-0.46	-0.62	-0.60
	BIQI	-0.42	-0.45	-0.60	-0.59
	NIQE	-0.72	-0.71	-0.85	-0.84
	MEON	-0.35	-0.30	-0.43	-0.39
	NR-GVQM	0.89	0.87	0.91	0.91
	NR-GVSQI	0.87	0.86	0.89	0.88
	nofu	0.91	0.91	-	-
	NDNetGaming	**0.934**	**0.933**	**0.934**	**0.929**

Video Gaming Quality Assessment

Averaged Pooled



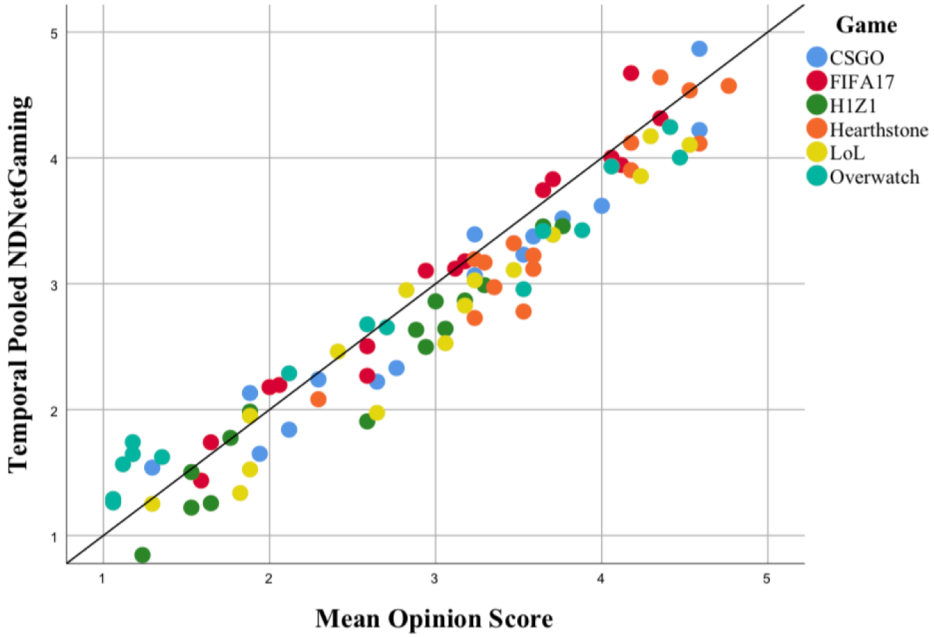
KUGVD PCC 0.934 (rmse = 0.464)



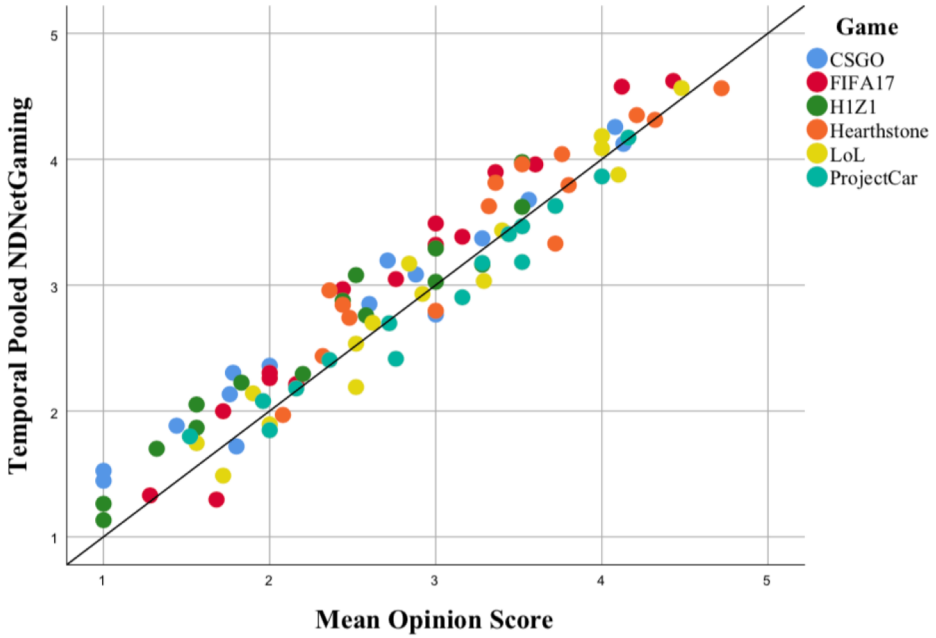
GVSET PCC 0.934 (rmse = 0.347)

Video Gaming Quality Assessment

Temporal Pooled



KUGVD PCC 0.965 (rmse = 0.28)



GVSET PCC 0.963 (rmse = 0.27)

Summary

- The plan is to make no-reference quality metric using CNN for gaming content
- The main aim is not only to predict quality but also measure the the type of distortion
- We used pretrained CNN models and fine-tune them based on the VMAF and MOS in two steps
- Investigate the reduction of computation cost
  - Lightweight CNN did not perform good

Points for Discussion

- We are biased to our dataset and condition we selected
- Prediction from sequences of the same game we had in training result in very high performance regardless of the distortion type
  - We can go with game specific metric
- 3D convolution can be seen as a good alternative
  - We did not get good result so far with similar method
  - It increases the computation cost a lot
- It seems to be difficult to get generalizable deep CNN metric

Points for Discussion

- Better result achieved for blur and noise than blockiness
- Training with more image distortion resulted in lower performance
  - Better to train the model for a specific purpose
- Huge dataset with content diversity might help to train whole network
- Correct patch quality scores may help to improve performance
  - With partial PSNR we did not achieve higher performance
  - Maybe partial VMAF!

Quality & Usability Lab

Thank you for your Attention!!

Any Question?

NDNetGaming

Saman Zadtootaghaj

saman.zadtootaghaj@qu.tu-berlin.de

Visit www.qu.tu-berlin.de for more information.