Practical 7 – Sample Quality and Data Privacy

Ole André Hauge

March 14, 2021

Sample Quality

1. We discussed the fundamentals of sample quality assessment.

   (a) [3 points] What is a fundamental expectation for the design of a quality score?

   Answer: According to the ISO/IEC 29794-1, the quality score (QS) shall produce a quantitative expression of quality that help predicting performance. The score should take three components: character, fidelity, and utility, so that it reflects, respectively, the quality to the inherent features and similarity to the source, as well as the positive or negative effect a single sample has on the overall performance of the system.

   Optimally these scores should be comparable with quality scores from another system, but this is sometimes not the case as vendors tend to have their own quality scores. The quality score can be used as feedback at enrolment to assist a decision regarding accepting or rejecting samples, demand recapture of characteristics, or perhaps terminate the enrolment process entirely e.g. declaring a failure to enroll.

   (b) [2 points] We have obtained the quality values in different ranges. In which range shall you expect a quality score according to ISO/IEC 19794-1(29794-1??)? How would you achieve this?

   Answer: According to ISO/IEC 29794-1 the quality values shall be in a range of 0 - 100. Obtaining this is done by what is referred to as "quality score normalization". It is the process of re-scaling quality scores to improve the consistency in scale and inseparability. Simply put, it takes raw quality scores and fits them on a scale of 0-100, without losing any information about the scores and how they correlate. The standard provides a couple of methods to achieve this: QSN, QAID, and QSND.

   (c) [5 points] Use you smartphone camera to capture images of your own face with different yaw angles (-90, -45, 0, 45, 90) - minimum 15 images. You can re-use the images captured as part of practical 2. To complete the sample quality analysis, use a pre-trained head pose estimation model to predict the yaw angles of your face images. You can use the following repository: https://github.com/vitoralbiero/img2pose, which is implemented in Python. Please submit your result as a table in PDF format, which documents "true yaw angle" vs. "estimated yaw angle" for each image. Note: Please do not upload any personal face images.

   Answer: Table 1 bellow shows the result of using the pre-trained model and the *test_own_images .ipynb* from the aforementioned GitHub repository. I took 15 images and linked to them in the Jupyter code.

Yaw Angle / Image	True	Estimated
000	0	3.6440116
001	45	25.0669038
002	45	41.5663688
003	-45	-8.149752
004	-90	-15.8422835
005	-45	-4.1596735
006	0	9.4136964
007	45	48.2831524
008	45	47.4924741
009	0	5.9301133
010	45	36.4916822
011	45	57.1983764
012	-45	-2.6642539
013	90	75.8768011
014	0	0.8373778

Table 1: Yaw Angle Results

The true angles are my attempt at angling my face in different yaw angles (-90, -45, 0, 45, 90). Since I am not a robot the angles might not have been 100% correct. Meaning that some of the estimated scores might be more accurate than my estimations. Further, the 90-degree images might be less accurate due to lighting conditions.

It is worth mentioning that the estimated angles are taken from the first value of the first list in the *'dofs'* key in the *res* dictionary resulting from the Jupyter code (see example below) as this corresponded the best to the actual values. Some guessing of what values correspond to what had to be done as the documentation and paper for the code were suboptimal. After reading the *img2pose* paper it became clear that the resulting values of the code were Euler angles, thus I had to convert these values to degrees.

```
'dofs': tensor([[-0.0726, ...]...])
```