FRVT Quality Assessment - Specific Image Defect Detection

Joyce Yang
Image Group
National Institute of Standards and Technology
U. S. Department of Commerce

FRVT Team: Patrick Grother, Mei Ngan, Kayee Hanaoka,
Austin Hom, Joyce Yang

International Face Performance Conference (IFPC) 2022
November 15, 2022

FRVT Quality Assessment - Specific Image Defect Detection

An ongoing evaluation of software that checks for quality problems in face images

API and Concept Document – 2022-08-19

1. FRVT 1:1 Verification	2. FRVT 1:N Search Performance	3. FRVT Morph Morphed Photo Detection	4A FRVT QA Image Quality Scalar Summary	4B FRVT QA Specific Image Defect Detection	5. FRVT Attack Presentation Attack Detection
2017...	2018...	2018...	2019...	2022 Q3 ...	2022 Q3 ...

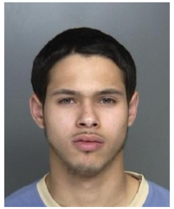
Contents

- Role, context, scope
  - Relationship to ISO/IEC 29794-5 now under development
- API
- Detailed description of quality measurements
- ISO/IEC 29794-5 Face Image Quality

FRVT Quality Tracks

NIST

TRACK A Quality Summarization



SCALAR: Q = 98

DECISION: Y, Accept

BOX 0. QUALITY BENCHMARK

- One ”visa – border” dataset
- No longer use wild
- Extend to use new ”kiosk” dataset

TRACK B Specific Image Defect Detection

BOX 1. QUALITY BENCHMARK

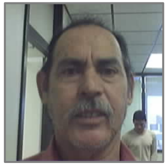
- Concept presented at the Nov Q Workshop 2021-11
- Initial API + Concept Published 2021-07-07 for public comment
- Final specifications (this document) 2022-08-19
- Algorithms to NIST 2022-09-26 [HTML]
- Align with ISO/IEC 29794-5 [PDF]

BOX 2. IMAGING VARIABLES THAT INFLUENCE ACCURACY

- Illumination adequacy + uniformity
- Exposure
- Focus, blur
- Resolution / Sp. Sampling Rate
- …

BOX 3. SUBJECT VARIABLES THAT INFLUENCE ACCURACY

- Head orientation (R, P, Y)
- Expression neutrality
- Sunglasses, face masks
- Motion blur
- No, or additional, faces
- …



Two People

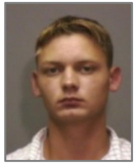
No People

Noise

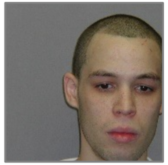
Over-exposure

Under-exposure

Hot Spots



Mis-focus



Cropped

Non-frontal

Criteria for including a component in FRVT SIDD

Required property of the quality metric

- Quantity should be related to recognition outcomes
  - Example YES: Resolution
  - Example NO: Shoulder orientation
- Quantity must be measurable from an image
  - Example YES: Yaw angle
  - Example NO: Exposure duration
- Quantity must have an available quantitative definition
  - Example YES: Mouth openness
  - Example NO: Expression neutrality
- Quantity could be (quickly) remedied in an operational setting
  - Example YES: Sun glasses present
  - Example NO: Signal to noise ratio
- Quantity should be capable of being measured on sequestered datasets (at NIST)
  - To separate developer-training from our testing

Properties not considered

- **Aspect ratio** non-square pixels - this occurs, it will undermine recognition, but an estimator seems likely to reject wide/narrow faces (how many sigma is acceptable)
- **Unnatural color**

Properties to be considered in future

- **Expression neutrality** - we don't have fine-grained expression information such as FACS or classification.
- Localized specular reflections **hot spots** - these should be part of the test, but how to specify severity? As area? Ground truth is not (readily) available.

C++ API

```
/**
 * @brief
 * Data structure that stores key-value pairs, with each
 * entry representing a quality component and its value
 */
using QualityAssessments = std::map<QualityMeasure, double>;
```

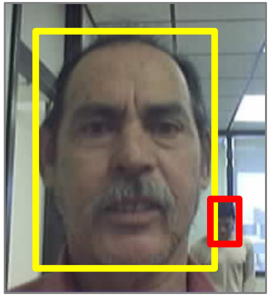
```
typedef struct ImageQualityAssessment
{
   FRVT::BoundingBox boundingBox;
   FRVT::QualityAssessments qAssessments;
 };
```

```
typedef struct BoundingBox
{
    int xleft; // leftmost point on head, typically subjects right ear
               // value must be on [0,ImageWidth-1]
    int ytop; // high point of head, typically top of hair
               // value must be on [0,ImageHeight-1]
    int width;     // box width
    int height;    // box height
 };
```

```
// Quality component labels
enum class QualityMeasure {
  Begin = 0,
  TotalFacesPresent = Begin,
  SubjectPoseRoll,
  SubjectPosePitch,
  SubjectPoseYaw,
  EyeGlassesPresent,
  SunGlassesPresent,
  Underexposure,
  Overexposure,
  BackgroundUniformity,
  MouthOpen,
  EyesOpen,
  FaceOcclusion,
  Resolution,
  InterEyeDistance,
  MotionBlur,
  CompressionArtifacts,
  PixelsFromHeadToLeftEdge,
  PixelsFromHeadToRightEdge,
  PixelsFromChinToBottom,
  PixelsFromHeadToTop,
  UnifiedQualityScore,
  End
};
```

- **Measures are optional - developers should implement one or more.**
- Others will be added in future revisions of this specification, and some may be removed.

```
/**
  * @brief  This function takes an image and outputs
  * face location and quality information. The quality assessment
  * should be performed on the largest detected face.
  *
  * @param[in] image
  * Single face image
  *
  * @param[out] assessments
  * An ImageQualityAssessments structure.
  * The implementation should populate
  * 1) the bounding box and
  * 2) those items in the QualityAssessments object that the
  *     developer chooses to implement
  */
virtual FRVT::ReturnStatus
vectorQuality(
   const FRVT::Image &image,
   FRVT::ImageQualityAssessment &assessments) = 0;
```



The quality of the second (small red) face should not be assessed, but it should be detected and counted in the Face Count component (see next slide)

Face count

Task

- Count the number of faces in the image, including those of the subject, people in the background, on T-shirts, in photos on the walls behind, even if cropped.
- Cropped partial faces should be detected (left corner in final example on this page)

Motivation

- In applications where one face is assumed, other faces can be detected instead of the intended one, leading to false negatives.
- Operationally detectors are usually configured to find faces whose size exceeds some small fraction of the image width.

Software output

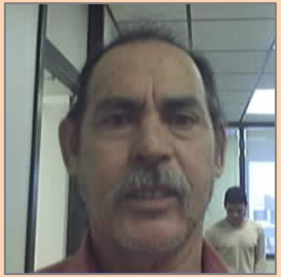
- Assign the **QualityMeasure::TotalFacesPresent** with the number of faces present in the image
- Do not count faces whose estimated IED is below 0.02W where W is the width of image

NIST will execute the code on

- sets of images with known number of faces, N = 0, 1, 2

NIST will report performance using

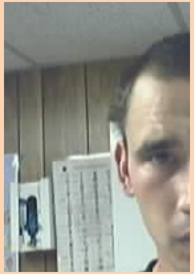
- Statistics on actual vs. reported counts, confusion matrix, overall accuracy
- Tabulate by image type (”wild”, “visa” ...) or conditioned on IED.



2

Background face has size about 3.5% of image width

0



1

1

2

Background face, if uncropped, would have size about 6% of image width

Non-frontal head orientation

Task

- Estimate the orientation of face (with respect to the camera):
- The head may not be close to the optical axis.

Motivation

- Head orientation other than ISO standard frontal can degrade accuracy

Software output

- Assign estimates of signed angles in degrees
  - **QualityMeasure::SubjectPoseRoll**
  - **QualityMeasure::SubjectPosePitch**
  - **QualityMeasure::SubjectPoseYaw**

Coordinate system as defined in ISO/IEC 39745-5

NIST will execute the code on images

- with known ground truth orientation (either by-design, or hand-coded)

NIST will report performance using

Visualizations of distribution of $\theta_{ESTIMATE}$ and θ_{TRUTH} and their difference ϕ

Penalties

- $F_{YAW}(\theta_{ESTIMATE} - \theta_{TRUTH})$
- $F_{PITCH}(\theta_{ESTIMATE} - \theta_{TRUTH})$ tolerant of unavailability of zero datum
- $F_{ROLL}(\theta_{ESTIMATE} - \theta_{TRUTH})$

With penalty e.g. $F(\phi) = 1 - \cos(a\phi)$ with scale factor “a” that is more tolerant of pitch angle errors and less tolerant of roll.

Yaw = +69 degrees
Pitch = 0 degrees
Roll = 0 degrees

Yaw = -37 degrees
Pitch = +4 degrees
Roll = +1 degrees

Yaw = -90 degrees
Pitch = 0 degrees
Roll = 0 degrees

Yaw = -22 degrees
Pitch = +3 degrees
Roll = -18 degrees

Eyes open

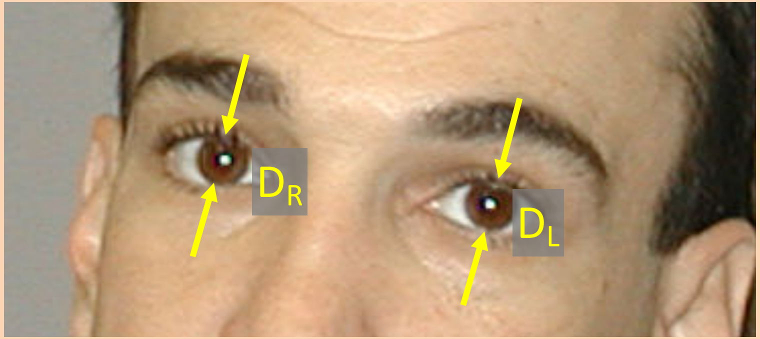
Task
- Determine if the eyes are required in standards
- Measure the palpebral aperture in left and right eyes, find the minimum of the two, and normalize by IED

Motivation
- Closed eyes can undermine localization and alignment, thereby contributing to FNMR

Software output
- Assign **QualityMeasure::EyesOpen** the measured minimum separation of eyelids divided by inter-eye distance = min(D_L, D_R) / IED

NIST will execute the code on
- images with eyes closed
- images with eyes variously open

NIST will report performance using
- Visualizations of joint distribution of estimated ratio and known ratio



See also [news story]on an incorrect rejection

Eye glasses present

Task

- Detect if eye glasses are present – include both transparent and sunglasses

Motivation

- Photography specification documents often include a policy for glasses
- False positives can occur because similar glasses' frames can increase non-mate score
- False negatives from change of style or presence of glasses
- ISO/IEC 39794-5 Annex D.2 guides that the thickness of frames of glasses should not exceed 5% of the estimated inter-eye distance (IED)

Software output

- Assign **QualityMeasure::EyeGlassesPresent** a value on [0,1] giving probability that eye glasses are present
- If no glasses are present, this should be zero.
- Caution: In future, because frame thickness matters, we may seek to change this variable to measure frame thickness as a proportion of estimated inter-eye distance. This approach would better relate to the effect on recognition.

NIST will execute the code on

- sets of images with and without glasses

NIST will report performance using

- Confusion matrix, error tradeoff between false negatives (failed detection) and false positive (erroneous detections)
- Summary measure: FNR at FPR = 0.01

Sunglasses present

Task

- Detect sunglasses (but not transparent eye glasses)

Motivation

- False negatives associated with occlusion of periocular detail
- This component is included separately to eye glasses because policy may dictate different actions for glasses vs. sunglasses

Software output

- Assign **QualityMeasure::SunGlassesPresent** a value on [0,1] giving probability sunglasses are present (1.0 for certainty)

NIST will execute the code on

- Sets of images with and without sunglasses

NIST will report performance using

- Confusion matrix, error tradeoff between false negatives (failed detection) and false positive (erroneous detections)
- Summary measure: FNR at FPR = 0.01

Mouth open

Task

- Measure how much the mouth is open
- Normalize lip separation by IED (which will require eye-finding)

Motivation

- Reduced mate comparison scores and increased false negatives due to the change in appearance relative to a reference photo

Software output

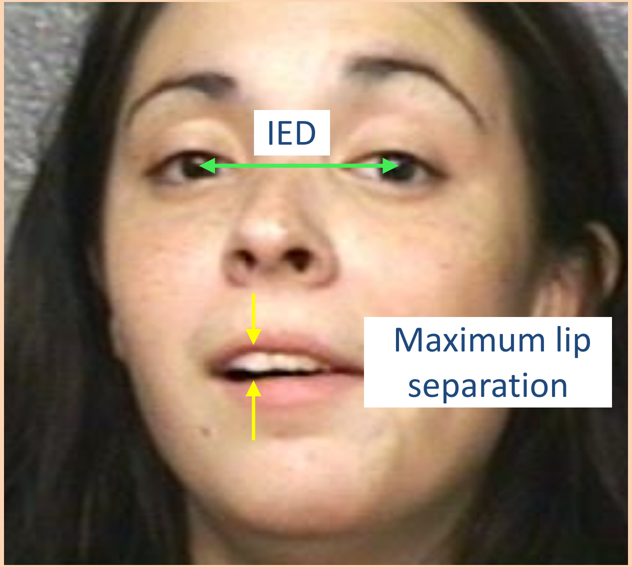
- Populate **QualityMeasure::MouthOpen** with the ratio: measured maximum separation of lips divided by inter-eye distance (IED). Limit range to [0,1] even if mouth is very wide open.

NIST will execute the code on

- images with mouth closed
- images with mouth open for which lip separation and IED are known

NIST will report performance using

- Visualizations of joint distribution of estimated ratio and known ratio



Normalization by IED because it is well-defined and ubiquitously computed. Alternatives such as normalization by lip thickness gives higher fractional error, possible age and ethnicity linkage.

Face occlusion

Task
- Quantify the area of the face that is occluded (by objects such as masks, hands, microphones, lecterns)
- The face region extends from top of forehead to chin, and from ear to ear.
- Ignore transparent eye-glasses and frames

Motivation
- Occlusion can impede detection and elevate FNMR

Software output
- Populate **QualityMeasure::Occlusion** with proportion of area that is occluded [0,1]

Evaluation
- Runs on sets of images with various levels of occlusion

NIST will report performance using
- Report pairwise statistics of ground-truth and measured value


33%


28%


0%


22%

Face cropping and margin

Task

- Determine if the face is cropped, or close to the image edge

Motivation

- Cropping can cause detection or recognition failure

Software output: estimate of proximity to edge of image

- **QualityMeasure::PixelsFromHeadToLeftEdge**
- **QualityMeasure::PixelsFromHeadToRightEdge**
- **QualityMeasure::PixelsFromChinToBottom**
- **QualityMeasure::PixelsFromHeadToTop**
- Negative values when face is cropped, giving estimate of how much is cropped
- Positive values give distance of closest part of the face to the edge

This formulation allows for head rotation, and avoids possible confusion arising from left side of face being in the right hand side of the image.

Evaluation

- Runs on sets images with various placements, yaw angles, crops

NIST will report performance using

- Report pairwise statistics on estimated vs. ground truth

(30,185,230,36)

(-25,-25,-36,-30)

(-15,105,48,20)

(15,1,48,12)

Background uniformity

Task

- Quantify how uniform the background is

Motivation

- Sufficient illumination non-uniformity will produce false negatives
- Possible false detection (i.e. of other people or non-faces in the background)

Software output

- Populate **QualityMeasure::BackgroundUniformity** with a value on [0,1] giving degree of non-uniformity of region behind the subject. Higher is worse.

NIST will execute the code on

- With uniform background
- The shadows from the subject head
- With cluttered background

NIST will report performance using

- Some statistics or visualization of actual vs. estimated
- Perhaps an error tradeoff characteristic

Spatial sampling rate

NIST

Task

- Compute the inter-eye distance (IED) in pixels
- Use the ISO/IEC {1,3}9794-5 definition (distance between canthi midpoints)
- For images where eyes are not visible due to occlusion or head rotation, produce an IED estimate based on some (anatomical) model – e.g. see example at right.

Motivation

- IED is a universally understood and widely specified in photography for biometrics, either with a direct value, or implied by the image dimensions (and a known geometry e.g. IED = W/4)
- Low or high values of IED are often immediately actionable
- While high IED is no guarantee of high resolution, low IED necessarily implies low resolution

Software output

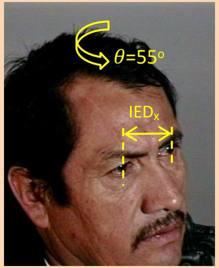
- Assign **QualityMeasure::InterEyeDistance** a higher-is-better value on [0, Inf] measured in pixels
- Do not round fractional estimates to integer

NIST will execute the code on

- Frontal images with various estimated IEDs.
- Highly non-frontal images (for which we have a frontal image from the same session)

NIST will report performance using

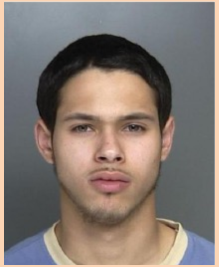
- Error statistics relative to estimated ground truth
- Condition the statistics on IED and on yaw angle



$IED_{FRONTAL} = IED_x \sec \theta$

EXAMPLE: 80.2 = 46 sec 55

NOTE: This method becomes inaccurate for large angles and fails with divide-by-zero error for a profile-view image.



IED = 120

IED = 70

Resolution

Task

- Quantify resolution (blind, without a calibration target). Produce a scalar value that expresses how far from perfect an image is with respect to absence of fine detail of the human face. This factors in all of the following de-focus, low spatial sampling rate, other homogeneous blur kernels.
- The software should operate on all images, but should assign highest values to an uncompressed image with IED of 256 pixels or higher that is perfectly focused and in all respects pristine.

Motivation

- Very low resolution gives elevated false negative rates in automated FR, and impedes human review

Software output

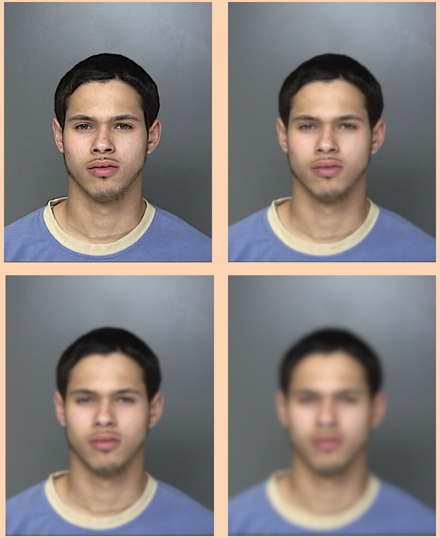
- Assign **QualityMeasure::Resolution** a value on [0,1] expressing how detailed and sharp the face in the image is.

NIST will execute the code on

- Sets of images considered to be ideal
- Sets of images with various reductions in resolution applied synthetically
- Sets of images with clearly low resolution

NIST will report performance using

- Calibration of the component against mate comparison scores
- Checks of correct ordering for progressively damaged images.



The four images have the same IED but much different resolutions

Motion blur

Task

- Quantify the extent to which motion blur affects the face in an image.
- The software should not report motion blur for an image affected by solely de-focus, or high compression.

Motivation

- Motion blur is one mechanism by which resolution is reduced. It can often be quickly remediated by asking the subject to be still, or by guiding the photographer to use shorter integration times and more light.

Software output

- Assign **QualityMeasure::MotionBlur** with an estimated displacement of the head from the beginning to end of the motion, measured in pixels
- The value should be zero when there is no motion, even for an out-of-focus camera

NIST will execute the code on

- sets of images considered to be ideal
- sets of images with various amounts of linear motion blur
- sets of images with various amounts of blur due to motion along a path

NIST will report performance using

- Measures of difference in estimated vs. known displacement

Compression artifacts

Task

- Quantify the presence of lossy compression artifacts: For JPEG these exist on an 8x8 grid. Note that in operations, this computation can be skipped if the input is a never-compressed image received from a sensor

Motivation

- Lossy compression is necessary in many applications but it permanently removes information that may be useful for recognition, thereby elevating comparison error rates.
- It is common for too much compression to be applied – this (particularly) impedes human review of images.

Software output

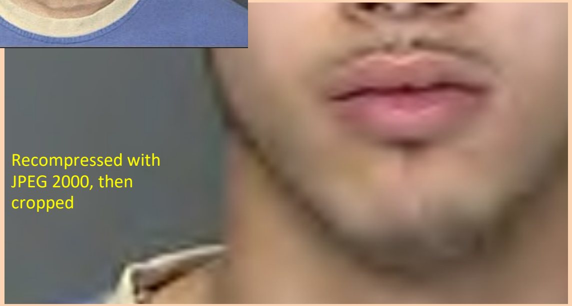
- Assign **QualityMeasure::CompressionArtifacts** a value on [0,1] that states how prominent compression artifacts are. A value of zero means no compression loss.
- One implementation would be to report a quantity related to encoded bits per pixel on the face region (e.g. by iteratively applying a compressor to the cropped uncompressed input until new loss is observed)

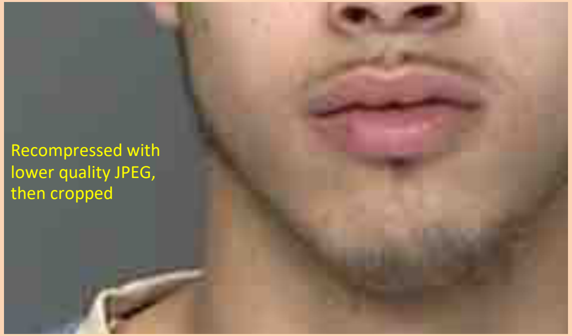
NIST will execute the code on

- sets of images with zero or very little compression
- sets of images with varying amounts of JPEG compression
- We will initially only consider ISO/IEC 10918-1 JPEG.
- Future: We may consider JPEG XL

NIST will report performance using

- Measures of difference in estimated vs. known displacement





Underexposure

Task

- Quantify underexposure of the face region in an image

Motivation

- Under exposure drives higher false negative rates
- Underexposure of ethnicities with lower skin reflectance induces a demographic differential in false negative rates (FNMR, FNIR)

Software output

- Assign **QualityMeasure::Underexposure** a value on [0,1] with higher values indicating poor exposure

NIST will execute the code on

- Hand-selected close-to perfect images and
- Images with a wide range of under-exposure

NIST will report performance using

- Joint distribution measures (e.g. QQ plot) of developer underexposure component with mated similarity scores produced by several mid-level accuracy FR algorithms comparing the underexposed images with good images.
- Summary statistics (explore rank correlation, partial).

Source: NIST Special Database 32 aka "MEDS", subject S171

0.9 0.8 0.6 0.2

NIST's will relate quality components to mate comparison scores. The alternative, for NIST to establish an automatically assigned ground-truth measure (e.g. entropy, or fraction of area that is "dark"), would lead developers into just re-implementing what NIST did. We seek prediction of continuous mated scores, not binary false negative decisions.

Overexposure

Task

- Quantify overexposure of the face region in an image.

Motivation

- Overexposure drives higher false negative rates
- Overexposure of ethnicities with high skin reflectance induces a demographic differential in false negative rates (FNMR, FNIR)

Software output

- Assign **QualityMeasure::Overexposure** a value on [0,1] with higher values indicating poor exposure

NIST will execute the code on

- Hand-selected close-to perfect images and
- Images with a wide range of overexposure

NIST will report performance using

- Joint distribution measures of developer overexposure measure with mated similarity scores produced by several mid-level accuracy FR algorithms comparing the overexposed images with good images.
- Summary statistics (explore rank correlation, partial).

NIST's proposal is to relate quality measurements to mate comparison scores. The alternative, for NIST to establish a ground-truth measure (e.g. entropy or fraction of area that is "light"), would lead developers into just re-implementing what NIST did. We seek prediction of continuous mated scores, particularly low scores, not binary false negative decisions.

Source: NIST Special Database 32 aka "MEDS"
Modified in powerpoint.

Source: NIST Special Database 32 aka "MEDS"
Modified in powerpoint.

Unified quality score

Task

- Summarize utility of an image for recognition as a scalar quality score.
- This can be implemented by ML-derived mapping of image to a score, or by mapping the specific defect quality components of this report to a score

Motivation

- Various use-cases seek a single number
  - That can be thresholded for yes/no acceptance decisions
  - Used to select a best image (of several available)
  - Used to summarize quality over some large collections

Software output

- Assign **QualityMeasure::UnifiedQualityScore** a value on [0,100] with higher values indicating an image is more likely to match a prior mate

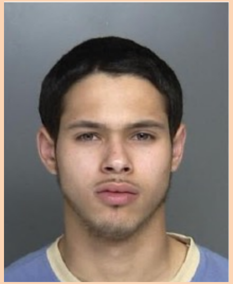
NIST will execute the code on

- Images that yield false negatives when compared with ISO-like reference images
- Images that do not yield false negatives

NIST will report performance using

- Statistics that associate low quality with higher likelihood of FNMR, including FNMR vs. QS; FNMR vs low QS rejection proportion; relationship of QS values and mated comparison scores. See [FRVT Track 4A Quality Summarization]

QS = 30



QS = 97

ISO/IEC 29794-5 Face Image Quality

NIST

0. Development

1. In Working Group 3 of SC 37, formally ISO/IEC JTC 1 SC 37 Biometrics
2. Latest draft 2022-09 [PDF]
3. To participate email patrick.grother AT nist.gov
4. The standard is defining
   - Specific tests (image processing operations) to be performed on an image; test results can be used to give actionable feedback to a photographer or subject
   - Numeric values (penalties) and data-types for the results of tests, and
   - An interpretable interoperable container for the results

1. Likely progression

1. 2022-08 WD 5
2. 2023-01 WD 6
3. 2023-04 CD 1 (copyright restrictions)
4. 2023-07 DIS 1 (copyright restrictions)
5. 2023-12-23 DIS 2 to ISO for publication
6. 2024 … PDF available for purchase

2. Capture-device related quality checks

- 6.3.2 Background uniformity
- 6.3.3 Illumination uniformity
- 6.3.4 Moments of the luminance distribution
- 6.3.5 Under-exposure
- 6.3.6 Over-exposure
- 6.3.7 Dynamic range
- 6.3.8 De-focus
- 6.3.9 Motion blur
- 6.3.10 Compression ratio
- 6.3.11 Unnatural colour
- 6.3.12 Radial distortion
- 6.3.13 Pixel aspect ratio
- 6.3.14 Camera to subject distance

3. Subject related quality checks

- 6.4.2 Single face present
- 6.4.3 Eyes visible
- 6.4.4 Eyes open
- 6.4.5 Mouth occlusion
- 6.4.6 Mouth closed
- 6.4.7 Nose occlusion
- 6.4.8 Inter-eye distance
- 6.4.9 Horizontal position of the face
- 6.4.10 Vertical position of the face
- 6.4.11 Pose
- 6.4.12 Shoulder presentation
- 6.4.13 Expression neutrality

4. Origin: Many clauses exist because Annex D in ISO/IEC 39794-5 establishes requirements

1. Reference face image for Machine Readable Travel Documents
2. General purpose face images.

5. New August 2022 public draft freely available here:
https://isotc.iso.org/livelink/livelink?func=ll&objId=22304355&objAction=Open&viewType=1

FRVT SIDD: Two roles

Support Quality Algorithm Development

- Assess capability of algorithms to quantify specific properties of faces in images that are associated with degraded face recognition performance
  - e.g. blur, non-frontal view

Support ISO/IEC 29794-5 Face Image Quality

- FRVT will support development by
  - Testing whether implementations of 29794-5 are accurate:
    - e.g. can pose be measured accurately
    - e.g. can an open-mouth be detected correctly
  - Testing whether a 29794-5 quality component expresses something that has influence on face recognition accuracy
  - Inform how to penalize a quality problem e.g. how should underexposure, or yaw angle, be penalized
- The draft of 29794-5 may include quantities not tested here.
  - Whether those quantities should be in the standard is beyond our scope here.
  - For example, the orientation of the shoulders and torso

29794-5 Terminology

Quality component is the scalar result of some image processing operation applied to the image.

Component values are on native intervals [a,b] and can have higher-is-better or lower-is-better semantics.

Used in 29794-5 and FRVT SIDD

Quality score is some transformation of the quality *component* to:
1. be an integer on [0,100];
2. have higher-is-better semantics.
3. have an English name that reflects higher-is-better semantics

Used in 29794-5 but not in FRVT SIDD.
FRVT SIDD may inform selection of these functions

Examples

Head pose: Yaw		θ_{YAW}		round(100 cos $\boldsymbol{\theta}_{\mathrm{YAW}}$) Name: Pose angle yaw frontal alignment
Eye openness: Eyelid aperture / Inter-eye distance		$\omega = DPAL\ /\ DIOD$		round(100 sigmoid(ω, 0.02, 0.01))
Background uniformity: Entropy measure in that region		$H = \sum p_i \log p_i$		round(100(1 − sigmoid(H,4,0.7))
Under-exposure: Luminance histogram weight in low 8 greylevels		$v = \sum_0^7 h_i$		round(100(1 − sigmoid(v,0.1,0.01)) Name: Non-underexposure

sigmoid(x, a, b) = $(1 + \exp(-(x-a)/b))^{-1}$ where a and b set position and gradient of the step

FRVT SIDD support of ISO/IEC 29794-5

2. Capture-device related quality checks	SIDD Support
6.3.2 Background uniformity	Background uniformity
6.3.3 Illumination uniformity	-
6.3.4 Moments of the luminance distribution	-
6.3.5 Under-exposure	Under-exposure
6.3.6 Over-exposure	Over-exposure
6.3.7 Dynamic range	-
6.3.8 De-focus	Resolution
6.3.9 Motion blur	Motion blur
6.3.10 Compression ratio	Compression artifacts
6.3.11 Unnatural colour	-
6.3.12 Radial distortion	-
6.3.13 Pixel aspect ratio	-
6.3.14 Camera to subject distance	-

3. Subject related quality checks	SIDD Support
6.4.2 Single face present	Face count
6.4.3 Eyes visible	Sunglasses + eyeglasses
6.4.4 Eyes open	Eyes open
6.4.5 Mouth occlusion	Face occlusion
6.4.6 Mouth closed	Mouth open
6.4.7 Nose occlusion	Face occlusion
6.4.8 Inter-eye distance	Spatial sampling rate
6.4.9 Horizontal position of the face	Face cropping and margin
6.4.10 Vertical position of the face	Face cropping and margin
6.4.11 Pose	Pose
6.4.12 Shoulder presentation	-
6.4.13 Expression neutrality	-

\- means not implemented in FRVT SIDD yet

Questions?

For more information:
Contact frvt@nist.gov
Visit pages.nist.gov/frvt/html/frvt_quality.html
Download slides from
pages.nist.gov/frvt/api/FRVT_ongoing_quality_sidd_api.pdf

FRVT SIDD: How a developer can participate

- Read this document
- Read the API
- Read the participation agreement; agree to it, sign it, scan it to PDF.
- Implement one or more image quality components enumerated in the API, and described below
- Download the FRVT quality validation package; compile, link, run, check output
- tar (or zip) the combined software and validation output; sign and encrypt the tar.gz
- Email frvt@nist.gov with
  - A download link to the encrypted package tar.gz.gpg
  - A PDF of the scan of the paper participation agreement
    - Do not mail a paper copy for this track of FRVT
  - Your public key (that was used to sign the tar.gz file)
- Subscribe to FRVT news
- …
- Consult https://pages.nist.gov/frvt/html/frvt_quality.html

Timeline:

1. 2022-07-05: First draft
2. 2022-08-18: Comments due
3. 2022-08-19: Final API published
4. 2022-09-26: Implementations can be submitted

Software API + implementation

NIST

API

» Quality interface and main function call
  - https://github.com/usnistgov/frvt/blob/master/quality/src/include/frvt_quality.h

» Supporting data types and enumerations
  - https://github.com/usnistgov/frvt/blob/master/common/src/include/frvt_structs.h

Supporting code

» A toy implementation of the API with random number outputs
  - https://github.com/usnistgov/frvt/blob/master/quality/src/nullImpl/nullimpl_frvtquality.cpp

» Public validation code, exercising the API
  - https://github.com/usnistgov/frvt/blob/master/quality/src/testdriver/validate_quality.cpp
  - This code must be executed by developers, and the outputs of the algorithm sent to NIST. NIST will check we can exactly reproduce the outputs on the same input images.
  - We distribute some unusual images (tiny, white, black, textured) in order to stress your code and elicit crashes before you send the code to us. The images are not supposed to represent our main testing images.