NIST
National Institute of Standards and Technology
U.S. Department of Commerce

secunet

Federal Office for Information Security

NFIQ 2.0
Open Source Distribution

Michael Schwaiger
secunet Security Networks AG

Elham Tabassi
NIST

Agenda

- Development kit
- Operational software

Distribution packages

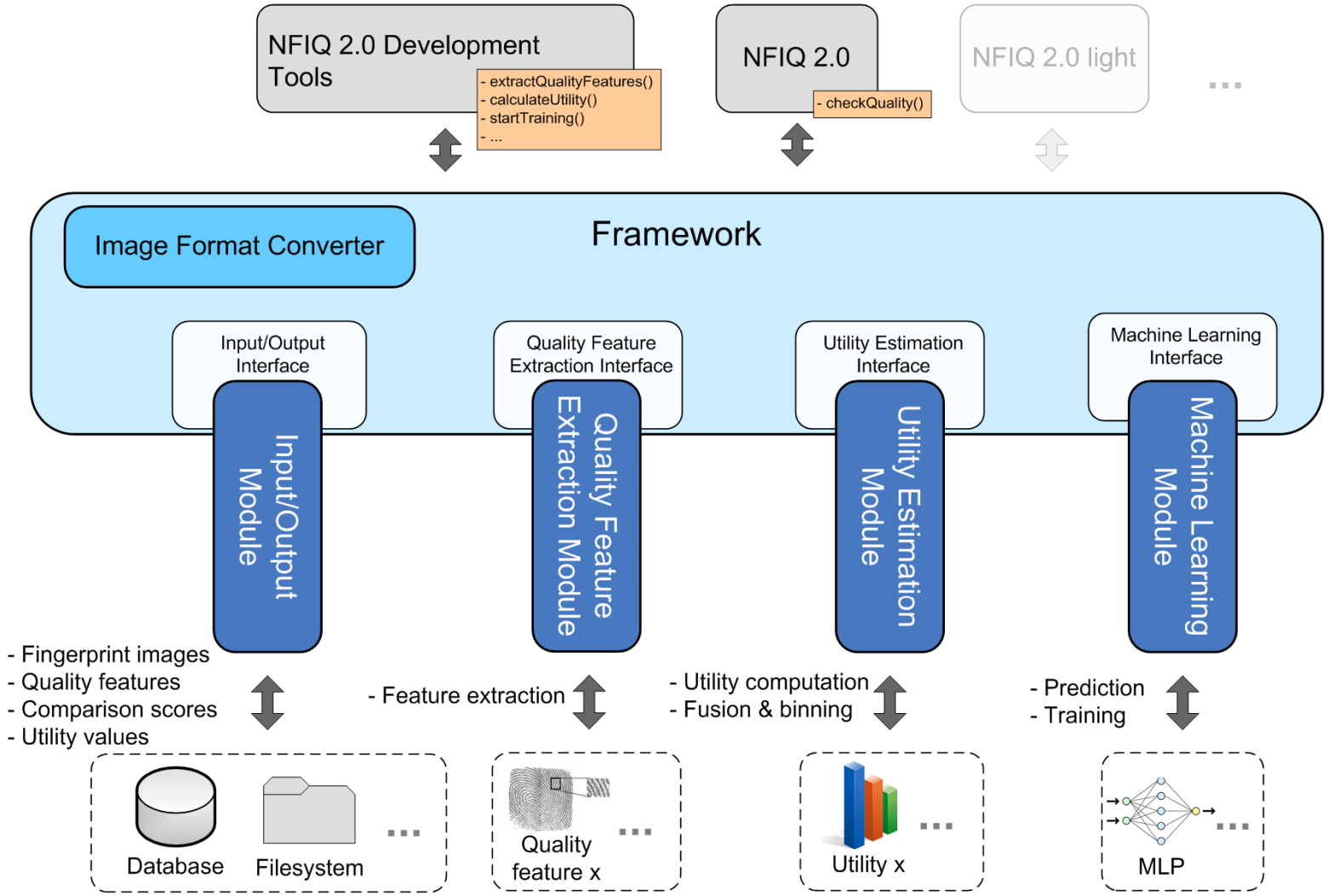
- Development kit
  - Aimed for developers and researchers
  - Includes NFIQ 2.0 Framework
  - Plug and play of different combinations of quality features and machine learining techniques
- Operational software
  - Aimed for operational use
  - Includes command line tool

Development kit
Motivation for NFIQ 2.0 Framework

- Lessons learned from NFIQ re-training in 2009/2010
  - NBIS source code changes necessary for adaptation of
    - machine learning algorithm
    - quality features
- Modular approach for NFIQ 2.0 development is desired
  - to be flexible regarding the implementation
  - to have a common basis of functionality needed for NFIQ 2.0 development which might then be extended by exchange of certain modules
  - because project team is distributed and located all over the world
  - because only certain project partners have access to certain fingerprint databases
  - to allow sharing and re-using of results
  - to simplify the development process

Development kit

Architecture of NFIQ 2.0 Framework



Development kit
NFIQ 2.0 development tools and interchange file format

- Implemented on top of the NFIQ 2.0 Framework
  - ComputeQualityFeatureData, ComputeQualityVector
  - ComputeUtilityValues
  - StartTraining
  - ComputeQuality (final or intermediate NFIQ 2.0 algorithm)
  - XMLExportImport
- XML interchange file format defined
  - exchange of training and evaluation data among project partners
    - fingerprint comparison scores of several databases
    - utility values and quality scores
    - quality features
  - referenced by unique IDs

Development kit

Example: How to perform training with the NFIQ 2.0 Framework I

- Assume that necessary data for training is stored in database X
- Compute quality features
  `computeQualityFeatureData(X, <featureID_1>, …, <featureID_M>)`
- Compute and fuse utility values
  `computeUtilityValues(X, <providerID_1>, <utility_ID>)`
  `…`
  `computeUtilityValues(X, <providerID_N>, <utility_ID>)`
  `fuseUtilityValues(X, <providerID_1>, …, <providerID_N>, <utility_ID>)`
- Select images for training
  `defineImagesForTraining(X, <trainingSet_X>, <testSet_X>)`
  and/or
  `partitionDataForTraining(X, <seed>, <N_train_X>, <N_test_X>)`
- Start training
  `train(<providerID_1>, …, <providerID_N>, <utility_ID>, <featureID_1>, …, <featureID_M>, useWeights)`

Development kit

Example: How to perform training with the NFIQ 2.0 Framework II

- Select images for evaluation
  `defineImagesForEvaluation(X, <evaluationSet_X>)`
  and/or
  `partitionDataForEvaluation(X, <seed>, <N_eval_X>)`
- Start evaluation
  `evaluate(<providerID_1>, …, <providerID_N>, <utility_ID>, <featureID_1>, …, <featureID_M>)`

Development kit
Open source libraries

- NFIQ 2.0 Framework will be open source

- External library dependencies
  - OpenCV for image processing
  - FingerJetFX minutiae extraction
  - RapidXML for XML parsing
  - NIST Biometric Data Interchange (BIOMDI)

Development kit
Input / output modules

- Modules for
  - PostgreSQL DB
  - NIST Record-Store format
  - File system

- Everyone can add new modules to adapt the NFIQ 2.0 Framework to their existing infrastructure!

Development kit
Quality feature modules

- More than 100 features integrated
  - NFIQ 1 features
  - FJFX minutiae count and minutiae quality features
  - Orientation Certainty Level, Ridge Valley Uniformity, Radial Power Spectrum, Local Clarity Score, …
  - ROI area features
  - Contrast features (Mu, Mu Mu Block, Sigma, …)
  - Quality map features
  - Gabor features

- Adding new features is easy!

Development kit
Machine learning modules

- Random Forest implemented
- Self Organizing Maps to be added for NFIQ 2.0 lite

Operational software
Why two different software distributions?

- Operational software package is what will be used in applications
- NFIQ 2.0 has overhead that is not needed in applications
  - Input/output modules
  - Utility modules
  - Features that were not selected for the NFIQ 2.0
- Operational software provides optimized code (not features itself but the usage of them)
- Command line tool will be provided
  - Input: Fingerprint image
  - Output:
    - Quality score
    - Actionable quality feedback
    - Quality feature values (optional)
    - Performance numbers (optional)

Operational software
NFIQ 2.0 command line tool

```
NFIQ2 <fingerprintImage> <imageFormat> <outputFeatureData> <outputSpeed>

    <fingerprintImage>: path and filename to a fingerprint image

    <imageFormat>: one of following values describing the fingerprint image
                   format: BMP, WSQ

    <outputFeatureData>: if to print computed quality feature values
                         (true|false)

    <outputSpeed>: if to print speed of quality feature computation
                   (true|false)
```

- Command line tool calls internal library that can be used to easily integrate NFIQ2 algorithm into applications

Operational software
NFIQ 2.0 examples

- Development version with 29 quality features

```
NFIQ2: Achieved quality score: 97
       Time needed for quality score computation: 213.000 ms
       Actionable quality (EmptyImageOrContrastTooLow):
              165.580 -> HIGH actionable quality
```

Operational software
NFIQ 2.0 examples

- Development version with 29 quality features

```
NFIQ2: Achieved quality score: 7
       Time needed for quality score computation: 128.364 ms
       Actionable quality (EmptyImageOrContrastTooLow):
              199.397 -> HIGH actionable quality
```

Operational software
NFIQ 2.0 examples

- Development version with 29 quality features

```
NFIQ2: Achieved quality score: 0
       Time needed for quality score computation: 0.515 ms
       Actionable quality (EmptyImageOrContrastTooLow):
              253.108 -> LOW actionable quality
```

Operational software
NFIQ 2.0 examples

- Output with feature values

```
FingerJetFX_MinutiaeCount: 57.000
FingerJetFX_MinCount_COMMinRect300x200: 39.000
FingerJetFX_MinCount_COMMinCircle200: 24.000
FingerJetFX_ROIBlockArea: 0.280
FJFXPos_Mu_MinutiaeQuality_0: 0.000
FJFXPos_Mu_MinutiaeQuality_1: 0.088
FJFXPos_Mu_MinutiaeQuality_2: 0.421
FJFXPos_Mu_MinutiaeQuality_3: 0.491
FJFXPos_COMMin_MMB_224: 127.428
FJFXPos_OCL_MinutiaeQuality_0: 0.000
FJFXPos_OCL_MinutiaeQuality_20: 0.018
FJFXPos_OCL_MinutiaeQuality_40: 0.053
FJFXPos_OCL_MinutiaeQuality_60: 0.421
FJFXPos_OCL_MinutiaeQuality_80: 0.509
Mu: 165.580
MMB: 165.580
OCL: 0.803
OCL_CD: 0.821
ImgProcROIPixelAbs: 105166.000
ImgProcROIPixelArea: 0.685
ImgProcROIArea_Mean: 127.076
OrientationMap_ROIFilter_CoherenceSum: 308.950
OrientationMap_ROIFilter_CoherenceRel: 0.687
LowFlowMap16_ROIArea_HighFlowBlocks: 443.000
RVU_P: 0.485
RVU_NP: 0.494
RPS_ROIArea: 5189.663
LCS: 0.825
OF: 0.864
```

Operational software
NFIQ 2.0 examples

- Output with feature speed

```
Contrast features (Mu, MMB): 0.454 ms
FJFX features (FingerJetFX_MinutiaeCount, FingerJetFX_MinCount_COMMinRect300x200,
       FingerJetFX_MinCount_COMMinCircle200, FingerJetFX_ROIBlockArea): 25.113 ms
FJFX minutiae quality features (FJFXPos_Mu_MinutiaeQuality_*): 0.381 ms
FJFX minutiae quality features (FJFXPos_COMMin_MMB_224): 0.090 ms
FJFX minutiae quality features (FJFXPos_OCL_MinutiaeQuality_*): 0.597 ms
OCL features (OCL): 1.568 ms
OCL features (OCL_CD): 10.233 ms
ROI features (ImgProcROIPixelAbs, ImgProcROIPixelArea, ImgProcROIArea_Mean): 12.259 ms
Quality map features (OrientationMap_ROIFilter_CoherenceSum,
       OrientationMap_ROIFilter_CoherenceRel): 2.737 ms
Quality map features (LowFlowMap16_ROIArea_HighFlowBlocks): 26.711 ms
RVU features (RVU_P): 12.518 ms
RVU features (RVU_NP): 12.247 ms
RPS features (RPS_ROIArea): 86.592 ms
LCS features (LCS): 16.166 ms
OF features (OF): 13.611 ms
```

Summary

- Development kit
  - Provides flexible integration and development for future versions and improvements
  - Design of dedicated versions possible (e.g. NFIQ 2.0 lite)
  - Large collection of quality features
- Operational software
  - Optimization done for use in applications
  - Unnecessary data and code removed
- Both will be distributed as open source!

Contact

secunet Security Networks AG
Michael Schwaiger
michael.schwaiger@secunet.com

NIST
Elham Tabassi
elham.tabassi@nist.gov