2021

An End-to-End Face Recognition System Evaluation Framework

James Andrew Duncan
West Virginia University, jduncan8@mix.wvu.edu

An End-to-End Face Recognition System Evaluation Framework

James Andrew Duncan

Problem report submitted
to the Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Daryl S. Reynolds, Ph.D., Chair
Natalia A. Schmid, D.S.
Xin Li, Ph.D

Lane Department of Computer Science and Electrical Engineering
Morgantown, West Virginia
2021

Keywords: Face Recognition, Face Detection, Face Recognition System Evaluation

Abstract

An End-to-End Face Recognition System Evaluation Framework

James Andrew Duncan

The performance of face recognition system components is traditionally reported using metrics such as the Receiver Operating Characteristic (ROC), Cumulative Match Characteristic (CMC), and Identification Error Tradeoff (IET). Recently, new metrics have been published to take advantage of annotation-dense datasets such as IARPA Janus Benchmark-Surveillance and IARPA Janus Benchmark-Multi Domain Face to describe end-to-end face recognition system performance. Unlike traditional (component-level) analysis, end-to-end analysis of a system produces a metric proportional to the experience of a user of a face recognition system. The End-to-End Cumulative Match Characteristic (E2ECMC) summarizes detection, identity consolidation, and identity retrieval performance. The End-to-End Subject Cumulative Match Characteristic (E2ESCMC) describes the lowest rank that subjects are retrieved in identification experiments. The End-to-End Identification Error Tradeoff (E2EIET) is a measure of the false alarm and miss performance of a system.

Until now, an evaluation utility capable of reporting the performance of individual components of a system and describing the user experience of a face recognition system was unavailable to the biometric community. Along with this problem report, a software package/framework capable of evaluating the performance of the following components will be released:

- Face Detector
- Face Verification
- Face Identification
- Face Clustering
- End-to-End Face Recognition System Performance

In addition to providing a utility to researchers and system integrators, the evaluation framework is a C++17 library which may be incorporated into evolving/fine-tunable face recognition system pipelines as a means of providing performance snapshots over time.

Contents

List of Figures

List of Algorithms

List of Tables

Listings

In memory of my mother, Mary Ernestine Duncan.

Acknowledgments

- My wife, Stacy. Thank you for your never-ending support (and patience) throughout my graduate studies.
- My parents, James and Ernestine. Thank you for teaching me the importance of education and for the guidance you've given me throughout life.
- My committee members: Dr. Daryl Reynolds, Dr. Natalia Schmid, and Dr. Xin Li. Nearly every day in my professional life, I am reminded of a topic I learned in one of your classes. The lessons you've taught me are truly invaluable.
- My colleagues and mentors, Dr. Nathan Kalka and Dr. Nick Bartlow. Thank you for always being available to discuss ideas and providing opportunities to sharpen my technical skills.

Chapter 1

Face Recognition System Components

Face recognition systems are constructed from several discrete components. The following sections describe each component in a face recognition system pipeline, provide sample output cases, and describe characterization methods.

1.1 Face Detector

1.1.1 Face Detector Function

After image/frame acquisition, the first component of a face recognition system is a face detector. Face detectors estimate the locations of faces in images or video frames. To align with traditional image processing techniques, a face detector's estimates for a piece of media, m, may be summarized by a set of tuples representing x and y offsets, box width, w, height, h, frame number, f (if video), and detection confidence, c:

$$\mathcal{D}_{\Updownarrow} = \{(x_1, y_1, w_1, h_1, f_1, c_1), \ldots, (x_n, y_n, w_n, h_n, f_n, c_n)\}$$

. In addition to detecting faces, a tracking method may be implemented in this stage to aid in the construction of identity tracks (the locations of an identity from frame-to-frame) [9].

1.1.2 Face Detector Performance

Face detector performance may be expressed for a known dataset for all confidences as pairs of probabilities: probability of false detection and probability of correct detection. Realistically, this can be reported as a discrete Receiver Operating Characteristic (ROC) curve [10].

Face Detection Association

Since ground truth annotations may describe a slightly different location in space than the output of a face detector, the two quantities are not strictly comparable. A traditional method of determining box association is the Jaccard Index, or "Intersection Over Union (IOU)" which may be defined graphically as shown in Figure 1.1 and symbolically in Equation 1.1.

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{1.1}$$

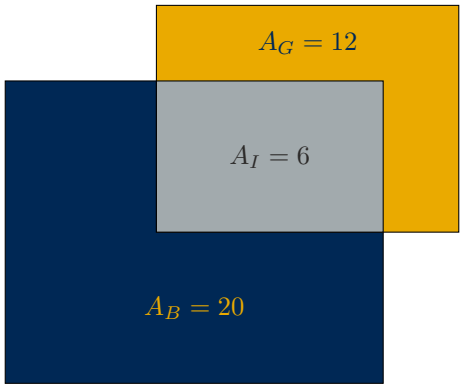

Figure 1.1: Intersection over Union Example. For this example, $J(B, G) = \frac{6}{12+20-6} = \frac{6}{26}$.

The difficulty of selecting an appropriate threshold for detection algorithm-agnostic bounding box association has been reduced by the area-normalized Jaccard Index metric [14, 11, 8]. The area-normalized Jaccard Index normalizes the area of the predicted box's shape to that of a ground truth box by nudging top-left and bottom-right coordinates in positive or negative directions. Coupled with a measure (e.g. percent difference) that describes differences between the original detected box's area and the normalized predicted box's area, false associations due to extremely small detections contained fully within a ground truth annotation can be avoided. The area-normalized Jaccard Index may be implemented as shown in Algorithm 1 and visualized in Figure 1.1.2.

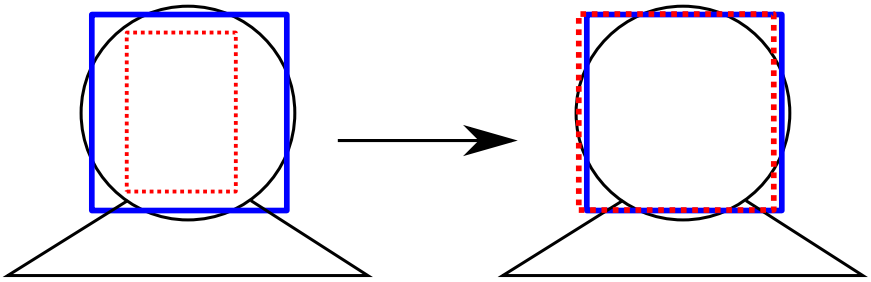

Figure 1.2: Area-Normalized Jaccard Index Visualization. The blue bounding box represents a ground truth annotation while the red bounding box represents a detected bounding box. The left figure is pre-normalization. The right figure is post-normalization. For this figure, the Area-Normalized Jaccard Index is 0.841 and the detected area percent difference (pre/post-normalization) is 78.6%.

Data: ground_truth_box, $G_{x,y,w,h}$

Data: detected_box, $D_{x,y,w,h}$

Result: area_normalized_jaccard_index, J

Result: normalized_percent_difference, P_d

initialization;

$J \leftarrow 0$;

$P_d \leftarrow 1$;

if *valid(G) and valid(D)* **then**

$\Delta h_2 \leftarrow (D_x - G_x)/2$;

$\Delta w_2 \leftarrow (D_w - G_w)/2$;

// Compute top/bottom/left/right of resized detected box

$E_t \leftarrow D_y + \Delta h_2$; $E_b \leftarrow D_y + D_h - \Delta h_2$; $E_l \leftarrow D_x + \Delta w_2$; $E_r \leftarrow D_y + D_w + \Delta w_2$;

// Compute top/bottom/left/right of ground truth box

$G_t \leftarrow G_y$; $G_b \leftarrow G_t + G_h$; $G_l \leftarrow G_x$; $G_r \leftarrow G_l + G_w$;

// Compute area of ground truth box

$A_G \leftarrow G_w\,(G_h)$;

// Compute area of original detected box

$A_{D_{\text{original}}} \leftarrow D_w\,(D_h)$;

// Compute area of normalized detected box

$A_{D_{\text{normalized}}} \leftarrow (E_r - E_l)\,(E_b - E_t)$;

// Compute x overlap

$O_x \leftarrow \max(0, \min(E_r, G_r) - \max(E_l, G_l))$;

// Compute y overlap

$O_y \leftarrow \max(0, \min(E_b, G_b) - \max(E_t, G_t))$;

// Compute Intersection

$I \leftarrow O_x\,(O_y)$;

// Compute Union

$U \leftarrow A_{D_{\text{normalized}}} + A_G - I$;

// Set outputs

$J \leftarrow I/U$;

$P_d \leftarrow 2\left|\frac{A_{D_{\text{original}}} - A_{D_{\text{normalized}}}}{A_{D_{\text{original}}} + A_{D_{\text{normalized}}}}\right|$;

end

Algorithm 1: Area-Normalized Jaccard Index

Face Detector Performance Example

Given the constraints in Table 1.1, consider the performance description in Figure 1.3. These metrics allow algorithm developers, system integrators, and even end users to select an operating point for the detector. By inspection, it is clear that:

1. There are 10% more ground truth annotations than detections. Therefore, the detector will, at best, detect 91% of faces presented to it assuming the test data is similar in all aspects to operational data.

2. It is possible for the false detects per image rate to be greater than 100% since there are more false detections than files.

3. At the selected sample threshold, 82% of faces may be detected with a a false accept detect rate of 1 face per 28 images.

Number of Ground Truth Annotations	110,000
Number of True Detections	100,000
Number of False Detections	50,000
Number of Images	4,000

Table 1.1: Hypothetical Face Detector Parameters

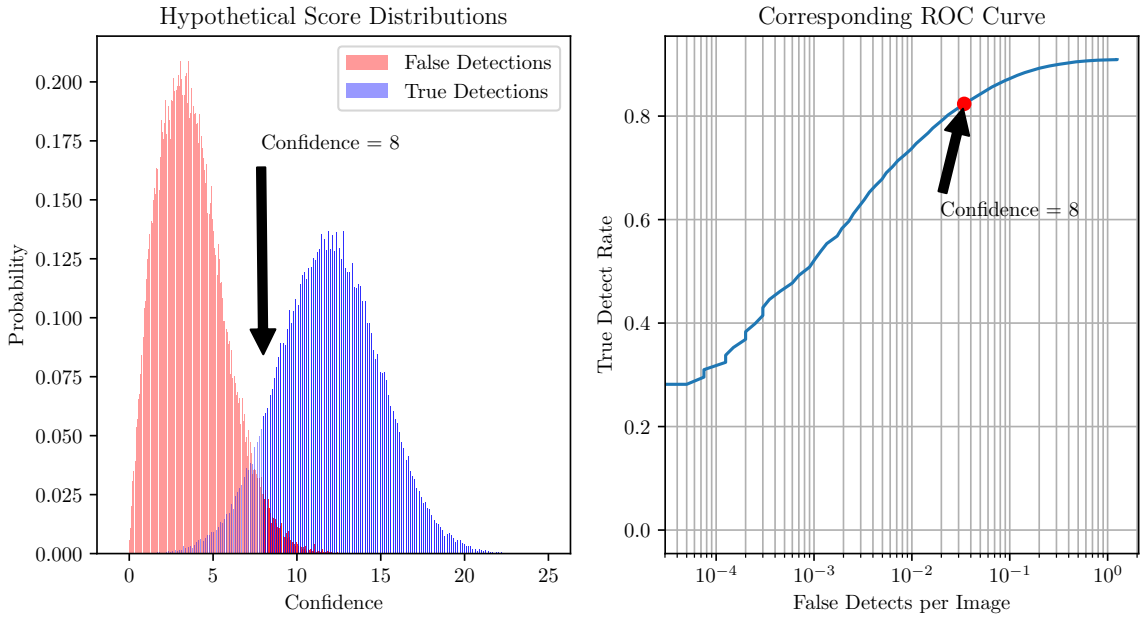

Figure 1.3: Discrete Score Distribution and Discrete ROC Curve Relationship

1.2 Feature Extractor

1.2.1 Feature Extractor Purpose

After face detection and photometric normalization, features (a lower-dimensional representation of an object than the image itself) may be extracted from the image of the face which allow for search and comparison [9].

1.2.2 Feature Extraction Techniques

Face features may be extracted using traditional techniques such as Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG) [9]. With the wide availability of affordable high-performance computing resources, networks such as VGG and FaceNet may be trained on large datasets to produce features for face recognition [13].

1.3 Comparator

1.3.1 Comparator Function

Face comparators consume two previously-extracted feature vectors, f_1 and f_2, and output a score, c. For practical purposes, $c \propto 1/d$ where $d :- \Delta(f_1, f_2)$, where Δ is a distance function. This functionality is referred to as 1:1 comparison or face verification.

1.3.2 Comparator Performance

Comparator performance may be expressed for a known dataset in a very similar manner as described previously in Section 1.1.2. *Note: NIST's Face Recognition Vendor Test reports verification performance as False Non-Match Rate/False Match Rate [5]. By traditional detection/estimation theory wisdom, FNMR* $= 1 - TPIR$. A 1:1 verification experiment is defined as a set of genuine (both faces are the same subject) and impostor (faces are two different subjects) face comparisons. In order to measure a system's performance at extremely-low false accept rates, a large number of comparisons are necessary [11]. For example, the IARPA Janus Benchmark-C dataset's 1:1 single image verification protocol contains 19,557 genuine matches and 15,638,932 impostor matches [11].

Given the constraints in Table 1.2, consider the performance description in Figure 1.4. Like the face detector, these metrics allow algorithm developers, system integrators, and even end users to select an operating point for their data and system. *Note that this example describes no failed enrollments (the inability of a feature extractor to extract a feature vector for a given face). The presence of failed enrollments will result in an ROC which does not terminate at (FAR,TAR) = (1,1). At low FAR, ROC curves may be non-smooth, indicating dataset limitations.*

Number of Impostor Comparisons	500,000
Number of Genuine Comparisons	1,000
Number of Failed Enrollments	0

Table 1.2: Hypothetical 1:1 Verification Experiment Parameters

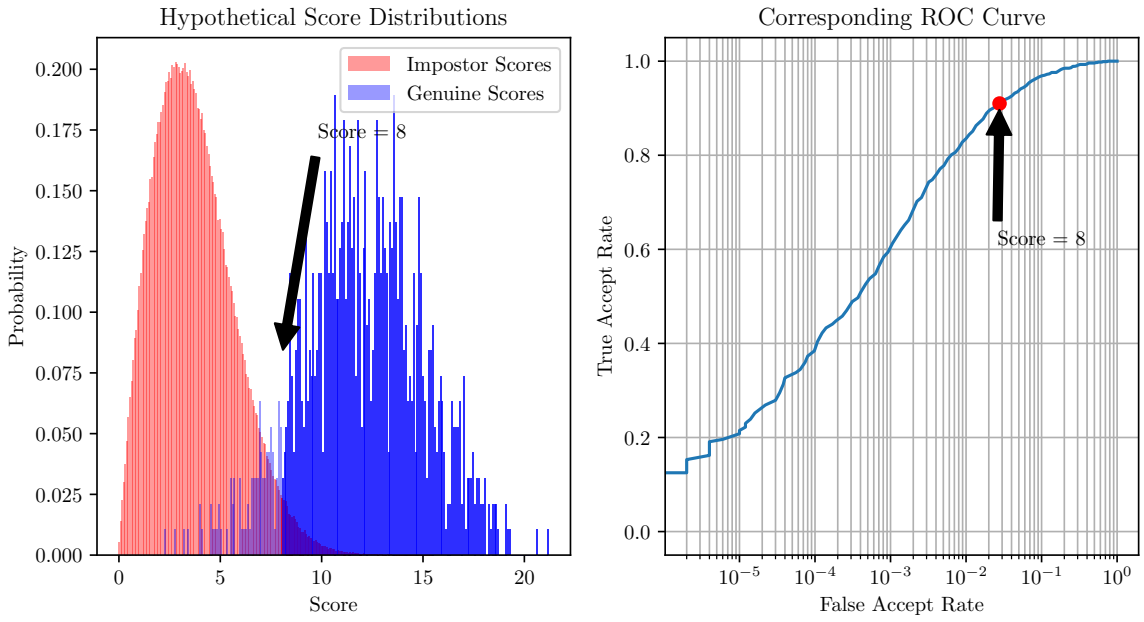

Figure 1.4: Discrete Score Distribution and Discrete ROC Curve Relationship

1.4 Matcher

1.4.1 Matcher Function

In a face recognition system, the matcher must compare a feature vector of a subject (probe template) to a set of known identities (gallery). The output of the matcher (see Table 1.3) for a given probe $\mathcal{P}_t$ is a candidate list of known subjects (denoted by $\mathcal{G}_s$, where s is a known subject's identity). While possible to generate a score for each known subject in a gallery, it may only be practical to return the top N hits, where N is determined by the end users or integrators of a face recognition system.

Probe Template	**Rank**	**Gallery Template**	**Score**
$\mathcal{P}_1$	1	$\mathcal{G}_2$	0.9077
$\mathcal{P}_1$	2	$\mathcal{G}_7$	0.8983
$\mathcal{P}_1$	3	$\mathcal{G}_1$	0.7033
...	...	...	...
$\mathcal{P}_1$	$\lvert\mathcal{G}\rvert$	$\mathcal{G}_8$	0.2924

Table 1.3: Hypothetical Candidate List for Probe Template $\mathcal{P}_1$

Some remarks:

- The correct candidate may not be present on the candidate list if the number of requested search results is less than the gallery size.
- A rank-1 candidate will always be returned, whether or not the probe subject is in the gallery.

1.4.2 Matcher Performance

Matcher performance is characterized by assessing the system's true accept rate (closed-set* performance), false positive identification rate, and false negative identification rate (open-set† performance).

Measuring Closed-Set Performance

Closed-set performance for a facial recognition system's matcher is described by the Cumulative Match Characteristic [12, 6]. The Cumulative Match Characteristic can be defined in a recursive manner as shown by Equations 1.2 and 1.3. $\mathcal{H}_r$ is the set of correct matches at rank r. $\mathcal{M}$ is the set of mated searches.

$$\mathrm{MC}\,(r) = \frac{|\mathcal{H}_r|}{|\mathcal{M}|} \tag{1.2}$$

$$\mathrm{CMC}\,(r) = \begin{cases} \mathrm{MC}\,(r) & r = 1 \\ \mathrm{CMC}\,(r-1) + \mathrm{MC}\,(r) & r > 1 \end{cases} \tag{1.3}$$

Measuring Open-Set Performance

Open-set performance describes the matcher's false positive and false negative performance [12, 6].

False Positive Performance False positive performance (False Positive Identification Rate, FPIR) describes a matcher's rank-1 candidates from nonmated searches [6]. Given a set of rank-1 nonmated candidate scores, $\mathcal{S}_\mathrm{U}$, the false positive identification rate can be defined for all thresholds, t, where $\min(\mathcal{S}_\mathrm{U}) \leq t \leq \max(\mathcal{S}_\mathrm{U})$. FPIR may be defined as shown in Equation 1.4 [6].

$$\mathrm{FPIR}\,(t) = \frac{|\{x \in \mathcal{S}_\mathrm{U} \colon x > t\}|}{|\mathcal{S}_\mathrm{U}|} \tag{1.4}$$

False Negative Performance False negative performance (False Negative Identification Rate, FNIR) describes a matcher's inability to retrieve a mated subject [6]. FNIR is defined for a set of mated searches, $\mathcal{M}$ possessing correct candidate scores S_M. *Note that in the general case,* $|\mathcal{M}| = |\mathcal{S}_M|$*, however, in practical applications (see Section 1.4.1),* $|\mathcal{M}| \geq |\mathcal{S}_M|$. FNIR may be defined as shown in Equation 1.5 [6].

$$\mathrm{FNIR}\,(t) = \frac{|\{x \in \mathcal{S}_\mathrm{M} \colon x < t\}|}{|\mathcal{M}|} \tag{1.5}$$

*A probe subject is enrolled in the gallery (mated).
†A probe subject is not enrolled in the gallery (nonmated).

1.5 Clusterer

1.5.1 Clusterer Function

Like many other machine-learning systems, a face recognition system may make use of a clustering algorithm. For example, features extracted from faces depicted in a pile of media may be clustered into K discrete bins via the K-Means clustering algorithm. In this simplistic example, the clustered features are assigned to bins which ideally represent individual identities.

1.5.2 Clusterer Performance

In biometric systems, clusterer performance may be described by three metrics: B-cubed precision, B-cubed recall, and B-cubed f-measure [1, 14, 11]. In a study comparing several clusterer performance methods [1], B-cubed metrics best represent clusterer performance when constrained by these criteria:

1. **Cluster Homogeneity:** Higher B-cubed precision may be observed when analyzing "clean" clusters.

   For face recognition systems, a clean cluster would represent a single identity.

2. **Cluster Completeness:** Higher B-cubed recall may be observed when analyzing large "clean" clusters.

   For face recognition systems, this condition corresponds to correctly clustering identities in a minimal amount of bins.

3. **Rag Bag:** "Junk" clusters may be created to contain unusual, infrequent features which would generally reduce the B-cubed precision.

   For face recognition systems, a "junk" cluster may contain features extracted from either poor quality faces (resolution, blur (focal distance or motion), extreme yaw/pitch/roll, or extreme occlusion) or non-faces (i.e. car tires, trees, dogs, etc.). B-cubed metrics do not penalize a clusterer for grouping these unusual components.

4. **Cluster Size vs. Quantity:** Higher B-cubed metrics may be observed when the amount of clean clusters are minimized.

   For face recognition systems, this condition could be observed if a clusterer produced the minimum number of clusters to represent all subjects in the dataset.

1.5.3 B-cubed Algorithm

The B-cubed algorithm may be implemented as shown in Algorithm 2.

Data: Cluster to Template Mapping

Data: Number of Clusters

Data: Cluster to (Subjects Count) Mapping

Data: Ground Truth Subjects Counts

Result: B-cubed Precision

Result: B-cubed Recall

Result: B-cubed F-Measure

initialization;

B-cubed Precision = 0;

B-cubed Recall = 0;

B-cubed F-Measure = 0;

foreach *(Cluster ID, Cluster Templates) in (Cluster to Template Mapping)* **do**

    Cluster Size = |Cluster Templates|;

    Subjects Counts = (Cluster to (Subjects Count) Mapping).at(Cluster ID);

    foreach *(Subject ID, Subject Count) in (Subjects Counts)* **do**

        Ground Truth Subject Count = (Ground Truth Subjects Counts).at(Subject ID);

        B-cubed Precision += (Subject Count) * (Subject Count) / (Cluster Size);

        B-cubed Recall += (Subject Count) * (Subject Count) / (Ground Truth Subject Count);

    end

end

// Note: normalization quantity is the number of ground truth templates. ;

B-cubed Precision /= | Ground Truth Subjects Counts | ;

B-cubed Recall /= | Ground Truth Subjects Counts | ;

B-cubed F-Measure = $\left((\text{B-cubed Precision})^{-1} + (\text{B-cubed Recall})^{-1}\right)^{-1}$

Algorithm 2: Computing B-cubed Metrics

Clustering Example Data

A visual interpretation of the B-cubed metrics (Algorithm 2) is shown below.

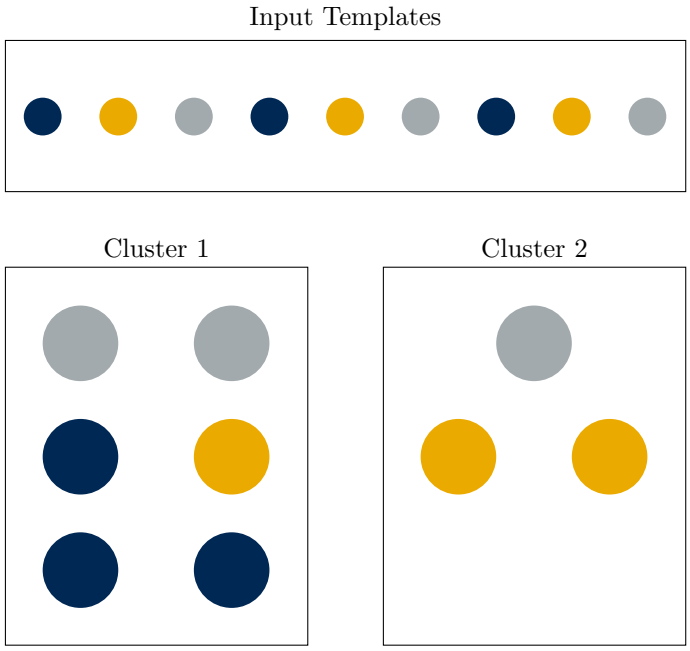

Figure 1.5: Clustering Example: Nine templates representing three classes (3 "WVU Blue", 3 "WVU Gold", 3 "WVU Neutral"). Cluster 1 contains three members of the "WVU Blue" class, one member of the "WVU Gold" class, and two members of the "WVU Neutral" class. Cluster 2 contains one member of the "WVU Neutral" class and two members of the "WVU Gold" class.

$$P_1 = \underbrace{\frac{1}{9}}_{\text{Number of Ground Truth Templates}} \left[\frac{3(3)+1(1)+2(2)}{\underbrace{6}_{\text{Cluster 1 Size}}} + \frac{2(2)+1(1)}{\underbrace{3}_{\text{Cluster 2 Size}}} \right] = \frac{24}{54}$$

$$R_1 = \frac{1}{9}\left[\frac{3(3)}{\underbrace{3}_{\text{\# Templates}}} + \frac{1(1)}{\underbrace{3}_{\text{\# Templates}}} + \frac{2(2)}{\underbrace{3}_{\text{\# Templates}}} + \frac{2(2)}{\underbrace{3}_{\text{\# Templates}}} + \frac{1(1)}{\underbrace{3}_{\text{\# Templates}}} \right] = \frac{19}{27}$$

$$F_1 = \frac{2}{\frac{54}{24} + \frac{27}{19}} = \frac{11}{39} = \frac{912}{1674}$$

Chapter 2

Face Recognition System Pipeline

For an operational system, the components described in Chapter 1 may be chained together to isolate identities depicted in photographs or video frames [3]. Analyzing the microscopic details of the end-to-end performance of a video-compatible face recognition system requires numerous spatiotemporal annotations. With the introduction of datasets like the IARPA Janus Benchmark-Surveillance (IJB-S) and IARPA Janus Benchmark Multi-Domain Face (IJB-MDF), these analyses are now possible. To illustrate this fact, consider that the IARPA Janus Benchmark-C (IJB-C) dataset contains $118,483$ annotations from video frames [11] while the IJB-S dataset contains over 10 million annotations from video frames [8].

2.1 End-to-End Facial Recognition System Examples

2.1.1 Intramedia Identity Consolidation

System Architecture

Intramedia identity consolidation operates under the assumption that users of a facial recognition system are interested in generating identity descriptions on a media-by-media basis [4]. A block diagram of this system is shown in Figure 2.1.

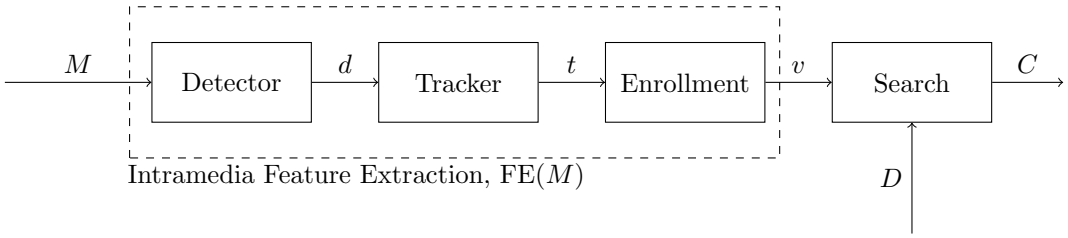

Figure 2.1: End-to-end facial recognition system architecture (intramedia identity consolidation) [4]. In this illustration, a face recognition system is presented with a piece of media, M. FE(M) detects faces, creates identity tracks, and extracts features from all detections in M. Finally, all identities are searched against database D, yielding candidates C.

Theoretical Use Cases/User Stories

- A security system which grants (or denies) visitors access
- A watchlist [4] – Are individuals of interest in this piece of media?

2.1.2 Intermedia Identity Consolidation

System Architecture

Intermedia identity consolidation operates under the assumption that users of a facial recognition system are interested in generating identity descriptions for a set of multimedia [4].

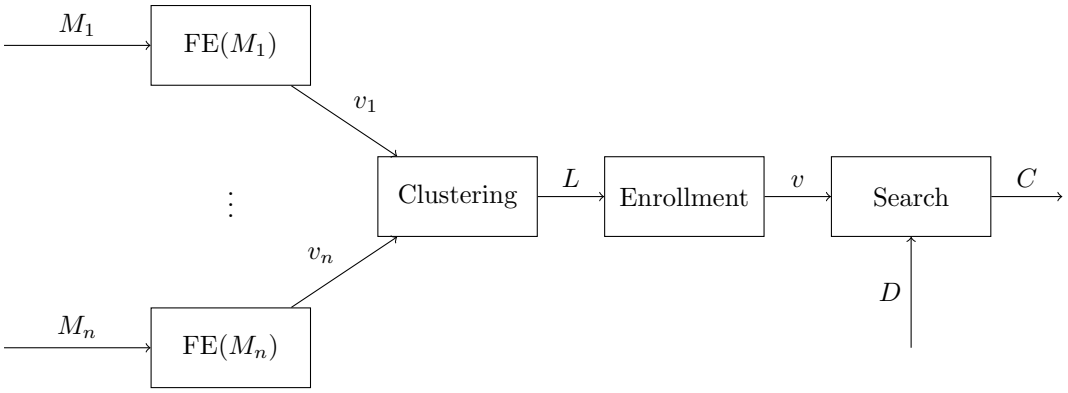

Figure 2.2: End-to-end facial recognition system architecture (intermedia identity consolidation) [4]. In this illustration, a face recognition system is presented with pieces of media, $M_0, ..., M_n$. FE(M) detects faces, creates identity tracks, and extracts features from all detections in M. Next, all extracted features are clustered and enrolled. Finally, all identities are searched against database D, yielding candidates C.

Theoretical Use Cases/User Stories

- Law enforcement leads [4] – Are the same individuals observed in multiple pieces of evidence?
- User experience – Provide the ability to ask a personal device "Show me all media containing me and my wife."

2.1.3 Description

End-to-end performance metrics of a facial recognition system describe the performance of the system as observed operating in two configurations: intramedia identity consolidation and intermedia identity consolidation.

2.1.4 End-to-End Metrics

A summary of the notation used to describe end-to-end metrics is shown below [4]:

- Multimedia (images or videos), $\mathcal{M}$
- Probe multimedia, $\mathcal{Z}$
- Subjects, $\mathcal{S}$, observed in $\mathcal{M}$.

- Subject/media occurrences (sightings), T
- Ground truth metadata database (queryable by media, m and subject, s), $\mathcal{N}_{m,s}$
- Ground truth metadata database of missed identities (queryable by media, m and subject, s), $\mathcal{B}_{m,s}$
- Detection database (queryable by media, m and subject, s), $\mathcal{D}_{m,s}$
- Candidate list (C) consisting of tuples describing: search rank, probe detections/media, search score, and gallery candidate $(c_r, c_d, c_m, c_s, c_g)$
- n galleries, $\mathcal{G}_n$, each consisting of a subset of subjects from $\mathcal{S}$

2.1.5 End-to-End False Positive Quantification

Unlike fully-defined identification experiments, end-to-end false positive performance may be reported as a raw count instead of a rate to avoid falsely-improving negative performance due to a very active detector [4]. A slight modification to this quantity from [4] has been made to account for known nonmated identities as seen in Equation 2.1.

$$\mathrm{E2E_{FP}}(t, \mathcal{G}_n) = \sum_{c \in \mathcal{C}} \sum_{\substack{c_s >= t \\ c_r = 1}} \sum_{m \in c_m} \sum_{s \in c_d \setminus \mathcal{S}} |c_d = s| + \sum_{c \in \mathcal{C}} \sum_{\substack{c_s >= t \\ c_r = 1}} \sum_{m \in c_m} \sum_{s \in c_d \cap \mathcal{S} \setminus \mathcal{G}_n} \frac{|c_{d=s}|}{|\mathcal{N}_{m,s}|} \tag{2.1}$$

2.1.6 End-to-End False Negative Identification Rate

As defined in [4], end-to-end false negative performance is shown in Equation 2.2.

$$\mathrm{E2E_{FNIR}}\,(t, \mathcal{G}_n) = \frac{1}{T} \left[\sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{G}_n} \frac{|\mathcal{B}_{m,s}|}{|\mathcal{N}_{m,s}|} + \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{G}_n \cap c_d} \sum_{c_s < t} \sum_{m \in c_m} \frac{|c_{d=s,m=m}|}{|\mathcal{N}_{m,s}|} \right] \tag{2.2}$$

2.1.7 End-to-End Cumulative Match Characteristic

As defined in [4], end-to-end cumulative match characteristic is shown in Equation 2.3.

$$\mathrm{E2E_{CMC}}(r, \mathcal{G}_n) = \frac{1}{T} \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{G}_n \cap c_d} \sum_{m \in c_m} \sum_{1 < i \leq r} \frac{|c_{d=s,r=i}|}{|\mathcal{N}_{m,s}|} \tag{2.3}$$

2.1.8 End-to-End Subject Cumulative Match Characteristic

The End-to-End Subject Cumulative Match Characteristic describes the first appearance of unique mated subjects for a set of ranks [4]. Note that unique subjects returned for a given rank and media may be represented as $\mathcal{E}_{r,m}$ [4]. Thus, End-to-End Subject Cumulative Match Characteristic is defined by [4] to be the quantity in Equation 2.4.

$$\mathrm{E2E_{SCMC}}\,(r, \mathcal{G}_n) = \frac{1}{|\mathcal{G}_n|} \sum_{m \in \mathcal{Z}} \sum_{1 \leq i \leq r} |\mathcal{E}_{r=i,m=m}| \tag{2.4}$$

2.2 End-to-End Example

A simple end-to-end experiment is described below.

2.2.1 End-to-End Probe Definition

Consider detection, feature extraction, identity consolidation, and identification given two files, `vid0.mp4` and `vid1.mp4`.

FILENAME
vid0.mp4
vid1.mp4

Table 2.1: Sample End-to-End Experiment Probe Definition

2.2.2 End-to-End Ground Truth Definition

For all listed files in the end-to-end experiment, metadata describing all observable identities must be available. In this example (see Table 2.2), one subject is observed in each file. *Note: the GT_ROW column is not an output. It exists solely for reference purposes.*

2.2.3 Detections/Tracks

A sample detection/track list is shown in Table 2.3. *Note: the DET_ROW column is not an output. It exists solely for reference purposes.*

GT ROW	SIGHTING_ID	SUBJECT_ID	FILENAME	FRAME_NUM	FACE_X	FACE_Y	FACE_WIDTH	FACE_HEIGHT
0	0	0	vid0.mp4	0	1	1	1	1
1	0	0	vid0.mp4	1	2	2	2	2
2	0	0	vid0.mp4	2	3	3	3	3
3	1	1	vid1.mp4	3	1	1	1	1
4	1	1	vid1.mp4	4	2	2	2	2
5	1	1	vid1.mp4	5	3	3	3	3
6	1	1	vid1.mp4	6	4	4	4	4

Table 2.2: Sample End-to-End Experiment Ground Truth

DET ROW	TEMPLATE_ID	FILENAME	FRAME_NUM	FACE_X	FACE_Y	FACE_WIDTH	FACE_HEIGHT	CONFIDENCE
0	0	vid0.mp4	1	2	2	2	2	10
1	1	vid0.mp4	2	3	3	3	3	11
2	1	vid0.mp4	3	4	4	4	4	12
3	2	vid1.mp4	4	2	2	2	2	13
4	2	vid1.mp4	5	3	3	3	3	14
5	3	vid1.mp4	6	4	4	4	4	16
6	3	vid1.mp4	6	5	5	5	5	17
7	3	vid1.mp4	6	6	6	6	6	18

Table 2.3: Sample End-to-End Experiment Detections

Identity associations for the detections listed in Table 2.3 and hit/false alarm annotations may be seen in Table 2.4. Failed detections are summarized in Table 2.5.

DET ROW	GT ROW	HIT	FALSE ALARM	HIT CONTRIBUTION
0	1	✓		1/3
1	2	✓		1/3
2			✓	
3	4	✓		1/4
4	5	✓		1/4
5	6	✓		1/4
6			✓	
7			✓	

Table 2.4: Sample End-to-End Experiment Association

GT ROW	DETECTED	MISS CONTRIBUTION
0	×	1/3
1	✓	
2	✓	
3	×	1/4
4	✓	
5	✓	
6	✓	

Table 2.5: Sample End-to-End Experiment Failed Detections

2.2.4 Gallery

For this sample, consider the gallery ($\mathcal{G}_0$) shown in Table 2.6. *Note that subject 1's metadata is associated (Table 2.4), but is nonmated with respect to this gallery.*

2.2.5 Candidate List

A sample search result for templates defined in Table 2.3 against the gallery defined in Table 2.6 may be seen in Table 2.7. *Note that HIT/MISS, FALSE ALARM, and RETRIEVED SUBJECTS columns are not algorithm output and exist for illustration purposes only.*

TEMPLATE_ID	SIGHTING_ID	SUBJECT_ID	FILENAME	FRAME_NUM	FACE_X	FACE_Y	FACE_WIDTH	FACE_HEIGHT
10	10	0	0.jpg	0	1	1	1	1
11	11	2	2.jpg	0	1	1	1	1
12	12	3	3.jpg	0	1	1	1	1
13	13	4	4.jpg	0	1	1	1	1
14	14	5	5.jpg	0	1	1	1	1

Table 2.6: Sample End-to-End Experiment Gallery

SEARCH_TEMPLATE_ID	GALLERY_TEMPLATE_ID	RANK	SCORE	HIT/MISS	FALSE ALARM	RETRIEVED SUBJECTS
0	11	1	5			
0	10	2	4	1/3*		0
0	12	3	3			
0	13	4	2			
1	10	1	6	1/3*	1†	0
1	11	2	5			
1	12	3	4			
1	13	4	3			
2	10	1	7		1/4‡ + 1/4‡	
2	11	2	6			
2	12	3	5			
2	13	4	4			
3	13	1	3		2§ + 1/4‡	
3	12	2	2			
3	10	3	1			
3	11	4	0			

Table 2.7: Sample End-to-End Experiment Candidate List

*1 of 3 pieces of metadata returned from Sighting ID 0. This will be a hit if score threshold is lower than candidate score – a miss, otherwise.

†1 false detection in detection Template ID 1.

‡1 of 4 pieces of nonmated metadata returned from Sighting ID 1.

§2 false detections in detection Template ID 3.

2.2.6 Performance Computation Example

End-to-End Cumulative Match Characteristic

$$\mathrm{E2E}_{\mathrm{CMC}}(r, \mathcal{G}_0) = \begin{cases} \frac{1}{3} & r = 1 \\ \frac{2}{3} & r \in \{2, 3, 4\} \end{cases} \tag{2.5}$$

End-to-End Subject Cumulative Match Characteristic

$$\mathrm{E2E}_{\mathrm{SCMC}}(r, \mathcal{G}_0) = \begin{cases} \frac{1}{5} & r \in \{1, 2, 3, 4\} \end{cases} \tag{2.6}$$

End-to-End False Positives

$$\mathrm{E2E}_{\mathrm{FP}}(t, \mathcal{G}_0) = \begin{cases} 1 + \frac{1}{4} + \frac{1}{4} + 2 + \frac{1}{4} & t < 3 \\ 1 + \frac{1}{4} + \frac{1}{4} & 3 \leq t < 6 \\ \frac{1}{4} + \frac{1}{4} & 6 \leq t < 7 \\ 0 & t \geq 7 \end{cases} \tag{2.7}$$

End-to-End False Negative Identification Rate

$$\mathrm{E2E}_{\mathrm{FNIR}}(t, \mathcal{G}_0) = \begin{cases} \frac{1}{3} & t < 4 \\ \frac{1}{3} + \frac{1}{3} & 4 \leq t < 6 \\ \frac{1}{3} + \frac{1}{3} + \frac{1}{3} & t \geq 6 \end{cases} \tag{2.8}$$

Chapter 3

Face Recognition Evaluation Framework

3.1 Protocol Format Description

Given the nature of suitable data for end-to-end analysis (IJB-S and IJB-MDF), the evaluation framework which compliments this writing is designed to consume data in the IARPA Janus Benchmark format.

3.1.1 The Ground Truth Database

The IARPA Janus Benchmark protocol format defines the following required fields [14, 11, 8, 7] for each piece of metadata:

- `SUBJECT_ID` – a unique integer ID assigned to an identity. This subject ID must be observed in a minimum of two files. Occurrences will be distributed across one gallery protocol and probe protocols.
- `SIGHTING_ID` – a unique integer ID assigned to an identity for a piece of media. The sighting ID may be used to analyze identity tracks for a piece of media.
- `FILENAME` – a relative path to a file within the dataset distribution.
- `FRAME_NUM` – the frame number in a video for which a subject is visible (a sighting).
- `FACE_X` – the x-coordinate for a subject's bounding in `FILENAME` at `FRAME_NUM`.
- `FACE_Y` – the y-coordinate for a subject's bounding in `FILENAME` at `FRAME_NUM`.
- `FACE_WIDTH` – the width component for a subject's bounding box in `FILENAME` at `FRAME_NUM`.
- `FACE_HEIGHT` – the height component for a subject's bounding box in `FILENAME` at `FRAME_NUM`.

Note: If `SUBJECT_ID` and/or `SIGHTING_ID` are not-a-number (NaN) while possessing valid `FACE_*` attributes, the bounding box may be considered suitable only for face-detection purposes as it is not associated with an identity.

Note: If `FACE_*` fields are NaN while `SUBJECT_ID` is valid, this piece of metadata is the "parent" media for files possessing a similar `SIGHTING_ID` (e.g. video frames extracted from a "parent" video).

Note: If the FACE_* fields and the SUBJECT_ID field are all NaN, this piece of metadata is a file which contains no faces. These files are used for measuring false detection rate of a face recognition system.

Additional Fields

While the aforementioned fields are required for metadata which describe faces, some datasets may contain more fields. For example, IJB-C contains fields to study covariates [11, 2] such as:

- FACIAL_HAIR – the presence or absense of facial hair.
- AGE – the approximate age of the subject for this piece of metadata.
- INDOOR_OUTDOOR – this metadata describes an indoor or outdoor scene.
- SKINTONE – an enumeration describing a subject's skin tone for this piece of metadata.
- GENDER – Male or Female.
- YAW – Yaw component of subject's pose in this metadata.
- ROLL – Roll component of subject's pose in this metadata.
- OCC1-18 – Occlusion grid of subject in this metadata.

3.1.2 The Gallery Database

The IARPA Janus benchmark protocol defines the following required fields [14, 11, 8, 7] for each piece of gallery metadata. IARPA Janus benchmark datasets contain two galleries to compare performance for all dataset subjects in an open-set manner.

- TEMPLATE_ID – the (gallery only) template ID* for this subject. A template may be composed of multiple images.
- SUBJECT_ID – the subject ID represented by this piece of metadata.
- SIGHTING_ID – the subject's sighting represented by this piece of metadata.
- FILENAME – the relative path to a file for for this piece of metadata.
- FRAME_NUM – the frame number in a video for which a subject is visible (a sighting).
- FACE_X – the x-coordinate for a subject's bounding in FILENAME at FRAME_NUM.
- FACE_Y – the y-coordinate for a subject's bounding in FILENAME at FRAME_NUM.
- FACE_WIDTH – the width component for a subject's bounding box in FILENAME at FRAME_NUM.
- FACE_HEIGHT – the height component for a subject's bounding box in FILENAME at FRAME_NUM.

Note: Each piece of gallery metadata may also be observed in the Ground Truth Database.

*Template IDs shall not be repeated.

3.1.3 The Verification Protocol

The IARPA Janus benchmark protocol specifies two "sub-protocols" for the 1:1 verification task [14, 11] – one for enrollment and one to define required comparisons.

The Enrollment Protocol

The verification enrollment protocol possesses the same fields as the gallery enrollment protocol (see Section 3.1.2).

The Matches Protocol

The matches protocol defines the required comparisons for an experiment. Thus, the required fields for this protocol are:

- `TEMPLATE_ID1` – template ID of the "left-hand-side" of the comparison.
- `TEMPLATE_ID2` – template ID of the "right-hand-side" of the comparison.

3.1.4 The Identification Protocol

The IARPA Janus benchmark protocol defines identification experiments that involve at least one gallery protocol (see Section 3.1.1) and one probe protocol.

The Probe Enrollment Protocol

The probe enrollment protocol possesses the same fields as the gallery enrollment protocol (see Section 3.1.2).

3.1.5 The Clustering Protocol

The IARPA Janus benchmark protocol defines two clustering experiments: detection + clustering and template clustering.

Clustering Templates

Clusterable templates may be created with metadata possessing the same set of fields as the gallery database (see Section 3.1.2). This experiment may be evaluated directly as demonstrated in Section 1.5.3.

Clustering Detections

Given a set of filenames, a system must detect faces, extract features, and produce clustering results. After an initial identity association process (detected templates must be mapped to a ground truth identity), this experiment may be evaluated as demonstrated in Section 1.5.3.

3.1.6 The End-to-End Protocol

The IARPA Janus benchmark protocol's definition of an end-to-end protocol is simply a list of filenames [11].

3.2 Algorithm Output Requirements

To be compatible with the face recognition evaluation utility, a face recognition system must provide appropriate metadata for each use case defined below.

3.2.1 Face Detection Results

- `FILENAME` – the file containing this detection.
- `FRAME_NUM` – the frame number containing this detection.
- `FACE_X` – the x-coordinate of a detection's bounding box.
- `FACE_Y` – the y-coordinate of a detection's bounding box.
- `FACE_WIDTH` – the width of a detection's bounding box.
- `FACE_HEIGHT` – the height of a detection's bounding box.
- `CONFIDENCE` – detection confidence for this detection.

3.2.2 Verification Comparison Scores

- `TEMPLATE_ID1` – template ID of the "left-hand-side" of the comparison.
- `TEMPLATE_ID2` – template ID of the "right-hand-side" of the comparison.
- `SCORE` – score for this comparison[†].

3.2.3 Identification Candidate List

- `SEARCH_TEMPLATE_ID` – template ID which corresponds to a probe template as defined by the identification probe enrollment protocol (Section 3.1.4).
- `GALLERY_TEMPLATE_ID` – template ID which corresponds to a gallery template as defined by the gallery database (Section 3.1.2).
- `RANK` – candidate result set position. Lower ranks should ideally correspond to "better" candidates.
- `SCORE` – candidate score. Higher scores should ideally correspond to "better" candidates.

3.2.4 Clusterer Results

- `CLUSTER_INDEX` – a cluster ID.
- `TEMPLATE_ID` – a clustered template ID.

[†]For dissimilar templates, this score should approach an algorithm-specific minimum. For similar templates, this score should approach an algorithm-specific maximum.

3.2.5 End-to-End Results

Unlike previously-described experiments, end-to-end analysis requires output from several stages of a face recognition system pipeline.

Face Detection/Tracker Output

Output from the face detector is necessary to determine contributions to a given appropriate statistical quantity (hit/miss/false alarm). In addition to the fields described in Section 3.2.1, end-to-end analysis requires a `TEMPLATE_ID` column. This column is expected to describe an identity track for a given piece of media as generated by a face recognition system.

(Optional) Clusterer Output

Clusterer results may be considered for evaluating a face recognition system's ability to consolidate and identify identities from a set of multimedia. See Section 2.1.2 for a block diagram and use cases for this mode of operation. The same fields described in Section 3.2.4 are required for this mode of analysis. *Note that `TEMPLATE_ID` now corresponds to the `TEMPLATE_ID` defined in Section 3.2.5.*

Search Results

The search results produced by a face recognition system for end-to-end analysis possess the same fields described in Section 3.2.3. *Note that if clustering output is considered, the `SEARCH_TEMPLATE_ID` column is expected to describe the `CLUSTER_INDEX` column of the clusterer output.*

3.3 Utility Description

The presenting face recognition evaluation utility is composed of the five executables whose output and usage are described in the following sections. *Note that these utilities produce comma-separated output with a header corresponding to the executed experiment.*

3.3.1 Face Detection Evaluator

Usage

Listing 3.1: Face Detection Utility Usage

```
$ ./eval_fd
Usage: ./eval_fd                          \
    /path/to/fd_ground_truth.csv          \
    /path/to/fd_performer_output.csv
```

Output

The output of the face detection evaluation utility describes the points of a discrete receiver operating characteristic curve and their corresponding confidence thresholds.

FD_ROC		
threshold	**hit_rate**	**false_alarms_per_image**
float	float	float
...	...	...
float	float	float

Table 3.1: Face Detection Evaluation Output Fields and Data Types

3.3.2 Verification Evaluator

Usage

Listing 3.2: Verification Utility Usage

```
$ ./eval_verification
Usage: ./eval_verification                  \
    /path/to/metadata_0.csv                 \
    [...]                                   \
    [/path/to/metadata_n.csv]               \
    /path/to/scores.csv
```

Output

The output of the verification evaluation utility describes points of a discrete receiver operating characteristic curve and their corresponding confidence thresholds.

VERIFICATION_ROC		
threshold	**hit_rate**	**false_alarm_rate**
float	float	float
...	...	...
float	float	float

Table 3.2: Verification Evaluation Output Fields and Data Types

3.3.3 Identification Evaluator

Usage

Listing 3.3: Identification Utility Usage

```
$ ./eval_identification
Usage: ./eval_identification               \
    /path/to/probe.csv                     \
    /path/to/gallery_0.csv                 \
```

```
/path/to/candidate_list_0.csv           \
[...]                                   \
/path/to/gallery_n.csv                  \
/path/to/candidate_list_n.csv
```

3.3.4 Output

The output of the identification evaluation utility describes points of at least one[‡] cumulative match characteristic curve and at least one identification error tradeoff curve.

IDENTIFICATION_CMC		
gallery_num	**rank**	**hit_rate**
integer	integer	float
...	...	...
integer	integer	float

Table 3.3: Identification (CMC) Evaluation Output Fields and Data Types

IDENTIFICATION_IET			
gallery_num	**threshold**	**miss_rate**	**false_alarm_rate**
integer	float	float	float
...	...	...	...
integer	float	float	float

Table 3.4: Identification (IET) Evaluation Output Fields and Data Types

[‡]This utility may evaluate performance for multiple galleries and candidate lists for a given invocation.

3.3.5 Clusterer Evaluator

Usage

Listing 3.4: Cluster Utility Usage

```
$ ./eval_clustering
Usage: ./eval_clustering                    \
     /path/to/ground_truth.csv              \
     /path/to/cluster_list_0.csv            \
     [...]                                  \
     /path/to/cluster_list_n.csv
```

Output

The output of the clusterer evaluation utility consists only of the B-cubed metrics.

CLUSTER			
cluster_list_filebcubed_precision	**bcubed_recall**	**bcubed_fmeasure**	
string	float	float	float
...	...	...	...
string	float	float	float

Table 3.5: Clusterer Evaluation Output Fields and Data Types

3.3.6 End-to-End Evaluator

Usage

Listing 3.5: End-to-End Utility Usage

```
$ ./eval_e2e
Usage: ./eval_e2e                              \
    /path/to/fd_ground_truth.csv               \
    /path/to/fd_performer_output.csv           \
    /path/to/e2e_experiment_definition.csv     \
    /path/to/gallery_0.csv                     \
    /path/to/e2e_candlist_gallery0.csv
```

Output

E2E_CMC	
rank	**hit_rate**
integer	float
...	...
integer	float

Table 3.6: End-to-End (CMC) Evaluation Output Fields and Data Types

Output

E2E_SCMC	
rank	**unique_subject_retrieval_rate**
integer	float
...	...
integer	float

Table 3.7: End-to-End (CMC) Evaluation Output Fields and Data Types

E2E_IET		
threshold	**miss_rate**	**num_false_alarms**
float	float	float
...	...	...
float	float	float

Table 3.8: End-to-End (IET) Evaluation Output Fields and Data Types

Chapter 4

Future Work

4.1 End-to-End Detection Confidence

For the use case of end-to-end analysis, the face recognition system evaluation utility discards detection confidence values. Perhaps "partial hit," "partial miss," and "partial false alarms" could be weighted further by a face detection confidence heuristic.

4.2 Covariate Analysis

As noted in Section 3.1.1, some datasets are constructed in a manner which facilitates covariate analysis for some set of variables. To perform these analyses with the face recognition system evaluation utility, data must be pre-processed before analyzed. In the future, this support could be built in to the face recognition system evaluation utility.

4.3 Performance

Some tasks performed by the face recognition system evaluation utility are embarrassingly parallel, such as the resource-intensive task of media-wise face association. In the future, much higher performance may be achieved by implementing cross-platform multi-processing techniques for these tasks.

4.4 Language Bindings

The face recognition system evaluation utility is written in C++17. I/O and evaluation logic are contained in shared libraries which may be used by face recognition system pipelines, provided they are implemented in C/C++. In the future, at least one high-level-language binding for these features may prove useful given the popularity of Python-based machine learning frameworks such as Keras, TensorFlow, and PyTorch.

4.5 Result Plotter

The face recognition system evaluation utility produces output defined in Section 3.3. Plotting, summarization, and interpretation of the results produced by the utility are currently exercises left for the user. In the future, Gnuplot, Matplotlib, or MATLAB/Octave support could be added to assist with result plotting.

Bibliography

[1] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval Journal*, 12(4):461–486, 2009.

[2] J. Anderson, C. Otto, B. Maze, N. Kalka, and J. A. Duncan. Understanding confounding factors in face detection and recognition. In *2019 International Conference on Biometrics (ICB)*, pages 1–8, 2019.

[3] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain. Unconstrained face recognition: Identifying a person of interest from a media collection. *IEEE Transactions on Information Forensics and Security*, 9(12):2144–2157, Dec 2014.

[4] J. A. Duncan, N. D. Kalka, B. Maze, and A. K. Jain. End-to-end protocols and performance metrics for unconstrained face recognition. In *2019 International Conference on Biometrics (ICB)*, pages 1–8, June 2019.

[5] Patrick Grother, Mei Ngan, and Kayee Hanaoka. Face recognition vendor test (frvt) Part 1: Verification. `https://pages.nist.gov/frvt/reports/11/frvt_11_report.pdf` Accessed: 03APR2021.

[6] Patrick Grother, Mei Ngan, and Kayee Hanaoka. Face recognition vendor test (frvt) Part 2: Identification. `https://pages.nist.gov/frvt/reports/1N/frvt_1N_report.pdf` Accessed: 03APR2021.

[7] N. D. Kalka, J. A. Duncan, J. Dawson, and C. Otto. Iarpa janus benchmark multi-domain face. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9, 2019.

[8] N. D. Kalka, B. Maze, J. A. Duncan, K. O'Connor, S. Elliott, K. Hebert, J. Bryan, and A. K. Jain. Ijb–s: Iarpa janus surveillance video benchmark. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9, 2018.

[9] Stan Z. Li and Anil K. Jain. *Introduction*, pages 1–15. Springer London, London, 2011.

[10] Zicheng Liu and Baining Guo. *Face Synthesis*, pages 277–300. Springer New York, New York, NY, 2005.

[11] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother. Iarpa janus benchmark - c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165, Feb 2018.

[12] P. Jonathon Phillips, Patrick Grother, and Ross Micheals. *Evaluation Methods in Face Recognition*, pages 329–348. Springer New York, New York, NY, 2005.

[13] Mei Wang and Weihong Deng. Deep face recognition: A survey. *CoRR*, abs/1804.06655, 2018.

[14] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother. Iarpa janus benchmark-b face dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 592–600, July 2017.