Efficiency considerations for face recognition algorithms

International Face Performance Conference (IFPC)
29 October, 2020

Presented by:
Brendan F. Klare, Ph.D.

Co-Founder & CEO
Rank One Computing
Denver, Colorado
brendan@rankone.io
Template Generation Speed vs. Accuracy

Little correlation between speed and accuracy

One generally does not need to sacrifice speed for accuracy

Over 5x difference in speed

Amongst high accuracy vendors (< 1% error rate), there is a massive difference in template generation speed
No correlation between template size and accuracy

No need to sacrifice template size for accuracy

Over 20x difference in template sizes

There is over an order of magnitude difference in template size for top-tier accuracy vendors
The primary limitations on Facial Recognition applications and their viability come from the host machine’s processor and memory capacities.

### The Processor (CPU)

**Affects Speed**

<table>
<thead>
<tr>
<th>Template Generation</th>
<th>Comparison / Search</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FAST 200ms</strong></td>
<td><strong>FAST 1µs</strong></td>
</tr>
<tr>
<td>1 CPU core can create 5 templates per sec</td>
<td>1 CPU core can perform 1,000,000 comparisons per sec</td>
</tr>
<tr>
<td><strong>SLOW 1s</strong></td>
<td><strong>SLOW 10µs</strong></td>
</tr>
<tr>
<td>1 CPU core can create 1 template per sec</td>
<td>1 CPU core can perform 100,000 comparisons per sec</td>
</tr>
</tbody>
</table>

### The Memory (RAM)

**Affects Efficiency**

<table>
<thead>
<tr>
<th>Database / Template Size</th>
<th>Model and Binaries</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EFFICIENT 256 bytes</strong></td>
<td><strong>EFFICIENT 50 megabytes</strong></td>
</tr>
<tr>
<td>4GB of RAM can host ~15,636,000 templates</td>
<td>FR Software loads immediately &amp; doesn’t affect system resources.</td>
</tr>
<tr>
<td><strong>INEFFICIENT 4096 bytes</strong></td>
<td><strong>INEFFICIENT 2 gigabytes</strong></td>
</tr>
<tr>
<td>4GB of RAM can host ~976,000 templates</td>
<td>FR software is slow to load &amp; has a noticeable impact on system resources.</td>
</tr>
</tbody>
</table>
THE RIGHT HARDWARE FOR THE RIGHT APPLICATION

Architecture Options

Here is a description of this distribution

**Embedded Device**  
Portability & Single-Purpose

- **PROCESSOR**  
  1-8 cores  
  ARM, 1-2 GHz

- **RAM**  
  1GB - 8GB

**Desktop / Server**  
Dedicated Workstations & Workhorses

- **PROCESSOR**  
  4 - 64 cores  
  x64 with 2 to 4 GHz

- **RAM**  
  8GB - 256GB

**Scalable Cloud**  
Flexible & Service Based

- **PROCESSOR**  
  1 - 1000+ cores  
  x64 with 2 to 4 GHz

- **RAM**  
  1GB - 16GB
PRIMARY USE CASE

Identity Verification

User Presents the system with:

- **Personal Identifier**
  - i.e. ID #, Keycard, Username
- **Face Capture**
  - via Sensor (camera)

Retrieve Stored Template

Process Face
- Detection > Templatization > Assess Quality > Assess Liveness

Compare Template Similarity
- Access Granted
- Access Denied

Bank Account Access
Secure Facility Access
Phone Unlock
Tax Return Filing
**PRIMARY USE CASE**

**1:N Search**

- **Enter Probe Image**
- **Process Face**
  - Detection > Templatization > Assess Quality
- **Database Search**
  - From Identity Database
- **Identity Candidate List**
  - with Similarity Scores
- **Face Image Verification & Adjudication**
- **Investigative Report**

- Manually Performed by Trained Analyst / Examiner
- Automated by the Algorithm

---

**Identification of a Bank Robber**
from a surveillance video frame.

**Identification of an Assaulter**
from their online dating profile.

**Identification of a Hit & Run Suspect**
from a bystander's cell phone camera.
PRIMARY USE CASE

Automated Search

**Person**

Captured by Sensor
Usually a Camera

**Process Face**
Detection > Templatization > Assess Quality > Assess Liveness

**Database Search**
From Identity Database

**Similarity Threshold:**
Match | No Match

**Use Cases:**
- Airport Security Screening
- Home Security
- Terrorist Watchlisting
- Automated Photo Tagging
# Taxonomy of Bottlenecks

## Architecture vs. Use Case

<table>
<thead>
<tr>
<th>Identity Verification 1:1</th>
<th>Manual Search 1:N</th>
<th>Automated Search 1:N+1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Persistent Server/Desktop</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Slow template generation speed will reduce throughput/system response time.</td>
<td>- Large template size will require significant memory resources</td>
<td>- High template generation speed will reduce throughput (e.g., video processing)</td>
</tr>
<tr>
<td>- Large binary size will impact system restart speed</td>
<td>- High template generation speed will delay search results</td>
<td>- Large template size will exasperate memory resources</td>
</tr>
<tr>
<td><strong>Embedded Device</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Slow template generation speed will cause major latency (&gt; 3 seconds).</td>
<td>- Template size must be very small due to memory limits</td>
<td>- High template generation speed will render video processing impossible</td>
</tr>
<tr>
<td>- Large binary size will occupy a high percentage of available memory.</td>
<td>- High template generation speed will significantly delay search results</td>
<td>- Template size must be very small due to memory limits</td>
</tr>
<tr>
<td><strong>Scalable Cloud</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Large binary size will slow container instantiation time.</td>
<td>- NOT ADVISED TYPICALLY</td>
<td>- Poor network bandwidth will prevent video transmission</td>
</tr>
<tr>
<td>- Poor network bandwidth will delay image transmission.</td>
<td>- Large template size or large number of templates will make container instantiation very slow</td>
<td>- High template generation speed make video processing expensive</td>
</tr>
<tr>
<td>- Slow template generation speed will reduce throughput / system response time.</td>
<td>- Gallery size is typically too large to instantiate containers in less than 30 seconds</td>
<td>- Large template size, large number of templates, and/or large binary size will make container instantiation very slow</td>
</tr>
</tbody>
</table>
Identity Verification 1:1
Persistent Server / Desktop

- Slow template generation speed will reduce throughput/system response time
- Large binary size will impact system restart speed
Identity Verification 1:1
Embedded Device

- Slow template generation speed will cause major latency (>3 seconds).
- Large binary size will occupy a high percentage of available memory.
Identity Verification 1:1
Scalable Cloud

- Large binary size will slow container instantiation time
- Poor network bandwidth will delay image transmission
- Slow template generation speed will reduce throughput / system response time
Manual Search 1:N
Persistent Server / Desktop

- Large template size will require significant memory resources
- High template generation speed will delay search results
- High comparison speed will delay search results
Manual Search 1:N
Embedded Device

- Template size must be **very** small due to memory limits
- High template generation speed will significantly delay search results
- High comparison speed will significantly delay search results
- Large binary size will occupy a high percentage of available memory
Manual Search 1:N
Scalable Cloud

- NOT ADVISED TYPICALLY
- Large template size or large number of templates will make container instantiation very slow
- Gallery size is typically too large to instantiate containers in less than 30 seconds
Automated Search 1:N+1
Persistent Server / Desktop

- High template generation speed will reduce throughput (e.g., video processing)
- Large template and binary sizes will exasperate memory resources
Automated Search 1:N+1
Embedded Device

- Slow template generation speed will render video processing impossible
- Template size must be very small due to memory limits
- Large binary sizes will exasperate memory resources
Automated Search $1:N+1$

Scalable Cloud

- Slow template generation speed make video processing expensive
- Large template size, large number of templates, and/or large binary size will make container instantiation very slow
- Poor network bandwidth will prevent video transmission
FRVT Wish List

- Require vendors to certify the algorithm they submit with their operationally available version
- Enable submission of an additional algorithm or two to help ensure operationally deployed algorithms can be NIST validated
  - E.g., some vendors deploy two different algorithms
- Benchmark speeds on ARM devices
- Be involved in crafting legislation / regulation
  - E.g., help require only NIST validated algorithms are used operationally
    - Amazon AWS and Clearview AI have never submitted their algorithms to FRVT
    - NEC and Microsoft have never submitted their algorithms to FRVT Ongoing
Presented by:
Brendan F. Klare, Ph.D.
Co-Founder & CEO
brendan@rankone.io

BUILT BY ENGINEERS
IN DENVER, COLORADO.

Read our blog for more information:
blog.rankone.io

Partnership Inquiries:
bd@rankone.io