Video Analytics Framework with Multilevel Security

Dr. Patrick McDaniel
Zachary Lassman
Fall 2015
Video Analytics Network

- Distributed video database that can be queried on video metadata and feature classifications
- “Just-In-Time” video processing for feature classification
- Computational offloading from mobile devices to MicroClouds
Network Structure

Central Control Server

Mobile MicroClouds

Android Mobile Devices
Video Processing

- Frame extraction
- Frame classification
- Compilation of frame classification probabilities

- Tests conducted on 1080p mp4 video at approx. 30 fps
• OpenCV on server
  ▸ Bottleneck of server-side video processing
  ▸ Approx. 50 ms / frame
• FFmpeg on mobile devices
  ▸ Approx. 500 ms / frame
Classification

• Caffe deep learning framework using neural networks developed by Berkeley Vision and Learning Center
• Using models trained at ARL
• Slow on mobile devices
  ▸ Approx. 2000 ms / frame for 1080p mp4
Hardware Acceleration

- NVIDIA GeForce GTX Titan X GPU
- Caffe built using NVIDIA cuDNN
- Orders of magnitude faster
  - Approx. 7 ms / frame
Google Protocol Buffers

- Serialize and parse data represented by objects
- Efficient encoding
- Backwards compatible
- Code compiled from .proto file

Protobuf messages generated and prefixed with message size using varint encoding
# Timing Data

<table>
<thead>
<tr>
<th>size</th>
<th>gpu ext</th>
<th>gpu pred</th>
<th>mobile ext</th>
<th>mobile pred</th>
<th>thumbnail</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.4 MB</td>
<td>5154</td>
<td>438</td>
<td>36637</td>
<td>102643</td>
<td>desk w hat and perrier</td>
</tr>
<tr>
<td>85.6 MB</td>
<td>3302</td>
<td>300</td>
<td>25898</td>
<td>68013</td>
<td>laptop display</td>
</tr>
<tr>
<td>63.8 MB</td>
<td>2558</td>
<td>215</td>
<td>18249</td>
<td>50002</td>
<td>supreme hat</td>
</tr>
<tr>
<td>42.9 MB</td>
<td>1535</td>
<td>144</td>
<td>12634</td>
<td>33964</td>
<td>kleenex</td>
</tr>
<tr>
<td>21.2 MB</td>
<td>754</td>
<td>72</td>
<td>6828</td>
<td>16964</td>
<td>laptop</td>
</tr>
<tr>
<td>11 MB</td>
<td>390</td>
<td>36</td>
<td>4134</td>
<td>9655</td>
<td>perrier</td>
</tr>
<tr>
<td>6.2 MB</td>
<td>243</td>
<td>22</td>
<td>2990</td>
<td>6345</td>
<td>starbucks cup</td>
</tr>
<tr>
<td>4.6 MB</td>
<td>156</td>
<td>14</td>
<td>2344</td>
<td>4687</td>
<td>zongqing name</td>
</tr>
<tr>
<td>2.05 MB</td>
<td>120</td>
<td>7</td>
<td>1404</td>
<td>2852</td>
<td>green bag</td>
</tr>
</tbody>
</table>
Timing Data

- Frame extraction time (ms) vs video (MB)
- Prediction time (ms) vs video (MB)
Query Initiation

**Server**
- Query command from user or central server
- Nmap mobile device discovery
- Send query message to discovered mobile devices
- Run scheduling algorithm on unprocessed videos
- Send process directive message to mobile devices

**Mobile**
- Generate list of videos and corresponding metadata
- Send video info
- Generate lists of mobile and server side processing
Distributed Processing
Future Work (non-security)

• Further parallelization
• Query propagation from central command server and mobile devices
• Multiple GPU’s / MicroClouds
• General optimization
  ▸ Frame extraction
  ▸ Network communication
  ▸ Database caching
MLS

- Application of computer system to process information with incompatible classifications
- Based on military access control model
Military Access Control

- Classifications:
  - Top Secret
  - Secret
  - Confidential
  - Unclassified

- Information may only flow upwards through classifications
  - One can only view documents classified at or below their clearance

- Compartmented need-to-know access
Bell-LaPadula Model

• Model of computer security formulated in context of government classification

• Enforces two properties:
  ■ Simple security property (no read up): no process may read data at a higher level
  ■ *-property (no write down): no process may write data to a lower level

• Does not allow for approved interactions across classifications or changes to classification

• Deals only with confidentiality
Alternatives

- **Noninterference**: High’s actions have no effect on what Low can see

- **Nondeducibility**: Low cannot deduce anything with 100 percent certainty about High’s input

- **Harrison-Russo-Ullman model**: handles creation and deletion of files; operates on access matrices

- **Type enforcement**: used in SELinux
  - Subjects assigned *domains*, objects assigned *types*
  - Matrices defining permitted domain-domain and domain-type interactions

- **Role-based access control**: access depends on user’s role in organization
Biba Model

- Deals only with data integrity and ignores confidentiality
- Read up and write down
- NO read down and write up as high integrity objects could become contaminated with low
- Used in many modern computer systems: system files as high and network as low
- Does not allow trusted subjects to override security model
MLS Applications

• SCOMP
• Blacker
• MLS Unix
• NRL Pump
• Logistics Systems
• Sybardi Suite
• Wiretap Systems
Covert Channels

• Unintentional channel that can be abused to allow data flow from high to low confidentiality

• If high and low processes run on single system without partitioned resources, high process can signal low process to initiate data transfer
Application to Project

- MLS scheme for videos and video metadata
- Restricted access of certain classifications/locations
- Compartmentalized for collaboration among organizations
- Eliminate covert channels to prevent information leakage (obviously)