Raspberry Pi-based video filtration system

A novel approach to reversible PII anonymization in videostreams using commodity hardware

C.H.J Kuipers & S. Scholtes
February 6, 2018

Research Project 1
Master of System and Network Engineering
Institute of Informatics
University of Amsterdam
Introduction
Research question

How can commodity hardware be used to filter PII from video streams?
Research question

How can \textit{commodity hardware} be used to filter \textit{PII} from video streams?

1. What types of PII?
How can commodity hardware be used to filter PII from video streams?

1. What types of PII?
2. What anonymization techniques?
Research question

How can commodity hardware be used to filter PII from video streams?

1. What types of PII?
2. What anonymization techniques?
3. Tailoring to commodity hardware?
Figure 1: Video processing overview
Personally Identifiable Information
Any information related to a natural person, that can be used to directly or indirectly identify the person¹.

Filtration
Figure 2: The filtration process
Reversibility of the filtration process

- Contextual
- Container
Reversibility of the filtration process

- **Contextual**
  Image manipulation

- **Container**
  Encryption in transit & at rest
Filtration - anonymization techniques

Reversible methods:

- Pixel relocation
- Warping
- Chaos Cryptography
Filtration - anonymization techniques

Reversible methods:

- Pixelrelocation
- Warping
- Chaos Cryptography

One-way methods:

- Masking
- Blurring (a.k.a. Normalized Blurring)
Figure 3: Blur
Filtration - techniques

Reversible methods:

• Pixelrelocation
• Warping
• Chaos Cryptography

One-way methods:

• Masking
• Blurring (a.k.a. Normalized Blurring)
• Gaussian blurring
Figure 4: Gaussian blur
Proof of Concept
Figure 5: Overview

Proof of Concept
Components used for the Proof of Concept

Hardware:

• IP Camera
• Interception device
• Router
Proof of Concept

Components used for the Proof of Concept

Hardware:

• IP Camera
• Interception device
• Router

Software:

• Ubuntu 16.0.4.3 LTS
• Stretch 9.3
• Python 3.5.1-3
• OpenCV 3.3.0
• Caffenet Caffemodel
Test overview
Test overview

• Baseline
Figure 6: Baseline
Proof of Concept

Test overview

• Baseline
Test overview

• Baseline
• Detection
Proof of Concept

Test overview

- Baseline
- Detection
- Draw boxes
Figure 7: Draw boxes
Proof of Concept

Test overview

• Baseline
• Detection
• Draw boxes
Test overview

• Baseline
• Detection
• Draw boxes
• Labels
Proof of Concept

Figure 8: Label detections
Proof of Concept

Test overview

• Baseline
• Detection
• Draw boxes
• Labels
Proof of Concept

Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
Proof of Concept

Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
- Re-stream anonymized stream
Proof of Concept

Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
- Re-stream anonymized stream
- Blurring
Figure 9: Blurring detections
Proof of Concept

Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
- Re-stream anonymized stream
- Blurring
Proof of Concept

Test overview

• Baseline
• Detection
• Draw boxes
• Labels
• Save anonymized stream
• Save original stream
• Encrypt original stream
• Re-stream anonymized stream
• Blurring
• Blurring + padding
Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
- Re-stream anonymized stream
- Blurring
- Blurring + padding
- Gaussian blurring
Figure 10: Gaussian blurring
Proof of Concept

Test overview

• Baseline
• Detection
• Draw boxes
• Labels
• Save anonymized stream
• Save original stream
• Encrypt original stream
• Re-stream anonymized stream
• Blurring
• Blurring + padding
• Gaussian blurring
Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
- Re-stream anonymized stream
- Blurring
- Blurring + padding
- Gaussian blurring
- Gaussian blurring + padding
Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
- Re-stream anonymized stream
- Blurring
- Blurring + padding
- Gaussian blurring
- Gaussian blurring + padding
- Masking
Proof of Concept

**Figure 11:** Masking detections
Proof of Concept

Test overview

- Baseline
- Detection
- Draw boxes
- Labels
- Save anonymized stream
- Save original stream
- Encrypt original stream
- Re-stream anonymized stream
- Blurring
- Blurring + padding
- Gaussian blurring
- Gaussian blurring + padding
- Masking
Proof of Concept

Test overview

• Baseline
• Detection
• Draw boxes
• Labels
• Save anonymized stream
• Save original stream
• Encrypt original stream
• Re-stream anonymized stream
• Blurring
• Blurring + padding
• Gaussian blurring
• Gaussian blurring + padding
• Masking
• Masking + padding
Proof of Concept

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Detection</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Drawing boxes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Labeling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Save anonymized stream</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Save original stream</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AES Encrypting original stream</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Re-stream</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Blur</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blur + padding</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaussian blur</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaussian blur + padding</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Masking</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Masking + padding</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1:** Shows how the different tests are constructed
## FPS overview

<table>
<thead>
<tr>
<th></th>
<th>Dell desktop</th>
<th>Raspberry Pi 3</th>
<th>Atom Server</th>
<th>Atom Laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>120.1</td>
<td>9.2</td>
<td>7.9</td>
<td>3.46</td>
</tr>
<tr>
<td>Test 2</td>
<td>118.4</td>
<td>9.1</td>
<td>7.8</td>
<td>3.42</td>
</tr>
<tr>
<td>Test 3</td>
<td>117</td>
<td>9.1</td>
<td>7.8</td>
<td>3.45</td>
</tr>
<tr>
<td>Test 4</td>
<td>115.1</td>
<td>8.8</td>
<td>7.6</td>
<td>3.42</td>
</tr>
<tr>
<td>Test 5</td>
<td>101.1</td>
<td>7.3</td>
<td>6.2</td>
<td>3.06</td>
</tr>
<tr>
<td>Test 6</td>
<td>82.5</td>
<td>6.4</td>
<td>5.3</td>
<td>2.91</td>
</tr>
<tr>
<td>Test 7</td>
<td>85.8</td>
<td>6.4</td>
<td>5.3</td>
<td>2.92</td>
</tr>
<tr>
<td>Test 8</td>
<td>85.3</td>
<td>6.4</td>
<td>5.3</td>
<td>2.96</td>
</tr>
<tr>
<td>Test 9</td>
<td>76.6</td>
<td>6.0</td>
<td>5.4</td>
<td>2.87</td>
</tr>
<tr>
<td>Test 10</td>
<td>78.6</td>
<td>6.2</td>
<td>5.3</td>
<td>2.82</td>
</tr>
<tr>
<td>Test 11</td>
<td>57.9</td>
<td>4.1</td>
<td>4.78</td>
<td>2.5</td>
</tr>
<tr>
<td>Test 12</td>
<td>53.6</td>
<td>3.8</td>
<td>4.79</td>
<td>2.58</td>
</tr>
<tr>
<td>Test 13</td>
<td>82.0</td>
<td>6.3</td>
<td>5.4</td>
<td>2.88</td>
</tr>
<tr>
<td>Test 14</td>
<td>82.0</td>
<td>6.3</td>
<td>5.3</td>
<td>2.91</td>
</tr>
</tbody>
</table>

**Table 2:** Shows the relation between the baseline measurement and the performance hit in frames per seconds
Results Dell Desktop

![Graph showing framerates (FPS) and test numbers. The graph displays box plots for different tests, with the x-axis labeled 'Test number' and the y-axis labeled 'Framerate (FPS)'.]
Results Raspberry Pi 3

The chart displays the framerates (FPS) for different test numbers. Each test number is represented by a box plot, indicating the distribution of framerates across multiple trials. The x-axis represents the test number, while the y-axis represents the framerates (FPS).
Results Intel Atom Server

![Graph showing framerates (FPS) for different test numbers. The graph displays box plots for each test number, indicating the distribution of framerates with median, quartiles, and outliers.](image-url)
Results Intel Atom laptop

Test number

Framerate (FPS)
### FPS overview (percentage)

<table>
<thead>
<tr>
<th>Test</th>
<th>Dell desktop</th>
<th>Raspberry Pi 3</th>
<th>Atom Server</th>
<th>Atom Laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Test 2</td>
<td>98.58%</td>
<td>98.91%</td>
<td>98.73%</td>
<td>98.84%</td>
</tr>
<tr>
<td>Test 3</td>
<td>97.42%</td>
<td>98.91%</td>
<td>98.73%</td>
<td>99.71%</td>
</tr>
<tr>
<td>Test 4</td>
<td>95.84%</td>
<td>95.65%</td>
<td>96.20%</td>
<td>98.84%</td>
</tr>
<tr>
<td>Test 5</td>
<td>84.18%</td>
<td>79.35%</td>
<td>78.48%</td>
<td>88.44%</td>
</tr>
<tr>
<td>Test 6</td>
<td>68.70%</td>
<td>69.56%</td>
<td>67.09%</td>
<td>84.10%</td>
</tr>
<tr>
<td>Test 7</td>
<td>71.44%</td>
<td>69.56%</td>
<td>67.09%</td>
<td>84.39%</td>
</tr>
<tr>
<td>Test 8</td>
<td>71.02%</td>
<td>69.56%</td>
<td>67.09%</td>
<td>85.55%</td>
</tr>
<tr>
<td>Test 9</td>
<td>63.78%</td>
<td>65.22%</td>
<td>68.35%</td>
<td>82.95%</td>
</tr>
<tr>
<td>Test 10</td>
<td>65.44%</td>
<td>67.39%</td>
<td>67.09%</td>
<td>81.50%</td>
</tr>
<tr>
<td>Test 11</td>
<td>48.21%</td>
<td>44.56%</td>
<td>60.50%</td>
<td>72.25%</td>
</tr>
<tr>
<td>Test 12</td>
<td>44.63%</td>
<td>41.30%</td>
<td>60.63%</td>
<td>74.57%</td>
</tr>
<tr>
<td>Test 13</td>
<td>68.28%</td>
<td>68.48%</td>
<td>68.35%</td>
<td>83.24%</td>
</tr>
<tr>
<td>Test 14</td>
<td>68.28%</td>
<td>68.48%</td>
<td>67.09%</td>
<td>84.10%</td>
</tr>
</tbody>
</table>

**Table 3:** Shows the relation between the baseline measurement and the performance hit relative to the baseline measurement in percentages
Conclusion & Discussion
How can commodity hardware be used to filter PII from video streams?
Discussion

- Effectiveness anonymization techniques
- Edge detection
- False positives
- False negatives
- Raspberry Pi stability
- Hardware temperatures
Future Work

- Encryption
- Codecs
- Automation
- Distributing & clustering
- Anonymization techniques
- Overclocking the Pi
Questions?
Additional slides
Raspberry Pi setup

Figure 12: Raspberry Pi cooling issues
Raspberry Pi setup

Figure 13: Netbook and Raspberry Pi
DNN Accuracy

Snapshot of iteration 310.000:\(^2\)
Best performance at iteration 313.000
Values at iteration 313.000:

- Validation accuracy: 57.412\%
- Loss: 1,82328

Our snapshot:

- Top-1 accuracy: 57.4\%
- Top-5 accuracy: 80.4\%

---

\(^2\)Source: Jeff Donahue @jeffdonahue