Access Control Method for the Offline Home Automation System

Mikołaj Pabiszczak, Monika Grajzer, Łukasz Sawicki

Mikołaj Pabiszczak
Gido Labs
m.pabiszczak@gidolabs.eu
Resume of presenter

- BEng in Materials’ Engineering obtained at Poznań University of Technology, Poland.
- MA in Theoretical Mathematics obtained at Adam Mickiewicz University in Poznań, Poland.
- Machine Learning Engineer at Gido Labs.
Outline

- Scope of the paper
- Problem statement
- Related work
- The proposed solution
- Experiments and results
- Conclusion
Scope of the paper

- Home Automation Systems using small, embedded control devices

- Voice-based interface: access to the main speech recognition system (ASR) is controlled by an Access Control Decision Support System:
  - Voice Activity Detection (VAD) system for recognizing audio only when voice is detected AND
  - **Keyword Spotting Module (KWS)** — full ASR functionality is turned on after detecting proper keyphrase
  - Speaker Recognition - voice biometrics to grant access to the system only to authorized users
Problem statement

- Home automation systems with speech-based interfaces become increasingly popular.
- BUT: speech recognition is a resource-consuming task typically performed in the cloud => privacy concerns
- Offline systems working fully locally are desirable but challenging on small embedded devices
- Additional challenges:
  - support for non-English languages
  - relatively small dataset of examples with the recordings of a selected keyword
Related work

- KWS is a core part of an Access Control DSS
- Convolutional Neural Networks and Residual Neural Networks (ResNets) used for KWS:
  - State of the art KWS systems reach accuracy of 95% with False Positive Rate (FPR) of 2%.
  - BUT: this makes those solutions inapplicable "as is" in commercial set-ups - if a system makes prediction every second, with FPR=2% there will be ~72 false alarms in an hour.

- ResNets for Speaker Recognition
Solution

- In typical reference system only KWS and SR modules would be present
- Our solution: several other modules proposed (light gray)
- GOAL: reduce FPR of the AccessControl DSS
Solution - details

- **Loudness Checking** - allow only audio with the loudness above certain threshold to be processed further.

- **Timer** - after minimal loudness was reached, process only first 1.2s of audio (keyword is approx. 1s long).
Solution - details

- KWS module: a small ResNet (110k parameters) using 40 MFCCs as input
- Transfer learning:
  - model trained for recognizing 10 English keywords as a basis
  - adapted for recognizing single keyword from Polish language (by using smaller dataset of examples)
Solution - details

- **Score Smoothing** - calculates mean score of the last $n$ predictions and checks if it is above a certain threshold:
  - comparison between averaging over 3 or 4 predictions
Solution - details

- Speaker Recognition - takes a single frame with the strongest trigger.
- Utilises different ResNet as a classifier.
Evaluation

- Evaluation of the 3 main components proposed for Access Control DSS and their combinations:
  - Loudness Checking
  - Timer
  - Score smoothing with averaging over last 3 or 4 predictions

- **AIM:** estimate their influence on both False Positive Rate (FPR) and True Positive Rate (TPR)

- Reference system: only KWS block present

- The DSS system was implemented on a RaspberryPi 3B (CPU: 1,2 GHz quad-core; 1 GB RAM) with a custom-made microphone matrix (5 independent microphones)
Results

- FPR measurements:
  - analysing long audio recording with no keyword present
  - counting the number of falsely positive system activations (i.e. when the system has wrongfully detected the keyword)
  - FPR = ratio of the number of false alarms to the number of all analysed audio frames

<table>
<thead>
<tr>
<th>Design</th>
<th>FPR [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference system (only KWS)</td>
<td>2.23</td>
</tr>
<tr>
<td>Loudness checking [LC]</td>
<td>0.90</td>
</tr>
<tr>
<td>LC + Timer [T]</td>
<td>0.64</td>
</tr>
<tr>
<td>Score smoothing (avg. last 3)</td>
<td>1.43</td>
</tr>
<tr>
<td>[SS3]</td>
<td></td>
</tr>
<tr>
<td>Score smoothing (avg. last 4)</td>
<td>1.30</td>
</tr>
<tr>
<td>[SS4]</td>
<td></td>
</tr>
<tr>
<td>LC + T + SS3</td>
<td>0</td>
</tr>
<tr>
<td>LC + T + SS4</td>
<td>0</td>
</tr>
</tbody>
</table>
Results

- Performance of the best set-ups — analysing audio samples recorded live from 13 users:
  - 30 repetitions of the keyword
  - 10 other words 3 times each (both phonetically similar and very different from the keyword)

<table>
<thead>
<tr>
<th>Design</th>
<th>TPR [%]</th>
<th>FPR [%]</th>
<th>Acc. [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference system - KWS only</td>
<td>90.77</td>
<td>5.90</td>
<td>92.44</td>
</tr>
<tr>
<td>KWS + SS3</td>
<td>86.41</td>
<td>4.87</td>
<td>90.77</td>
</tr>
<tr>
<td>KWS + SS4</td>
<td>84.10</td>
<td>4.87</td>
<td>89.62</td>
</tr>
</tbody>
</table>
Conclusion

- Proposed Access Control Decision Support System allows to decrease FPR to an acceptable level while retaining high TPR:
  - overall accuracy above 90%
  - the proposed solution allowed to entirely suppress false alarms caused by background radio voices, while the reference set-up generated approx. 122 unwanted activations per 5471 analysed frames
- The proposed design is computationally lightweight — works on an embedded device => commercially applicable Access Control DSS
Thank you for your attention!

The presented research has been supported by the National Centre for Research and Development in Poland under the grant no. POIR.01.01-00-0044/17