Politecnico di Torino

Smart Surveillance System on Raspberry-Pi

Politecnico di Torino

Machine Learning for IoT

Francesco Di Salvo
s282418@studenti.polito.it

Gianluca La Malfa
s290187@studenti.polito.it

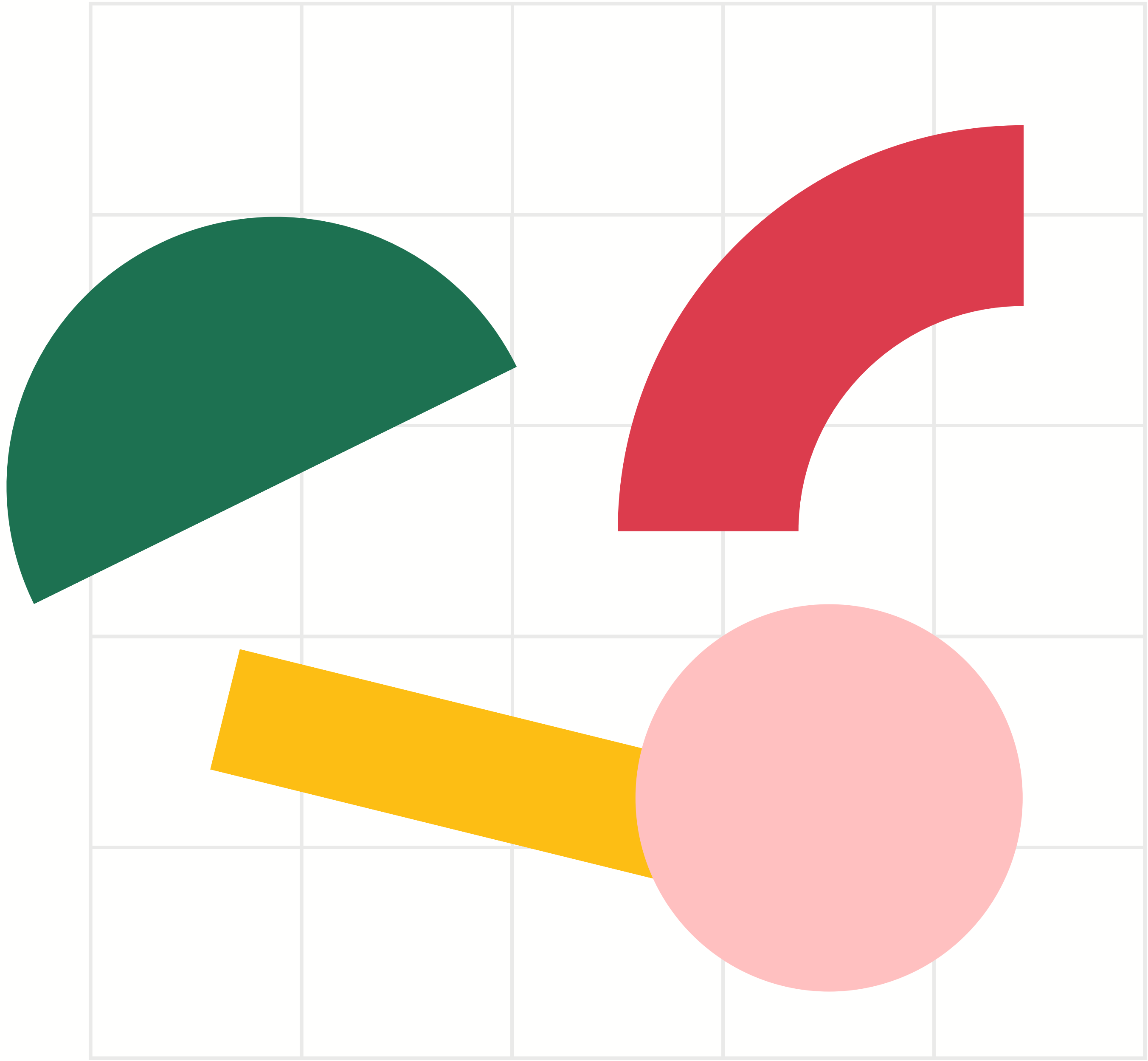
Leonardo Maggio
s292938@studenti.polito.it

Introduction

Video Surveillance is the act of monitoring the activities, behavior and changing activities in a scene for the purpose of managing, directing or identifying security threats in an automated way.

Smart Video Surveillance:

- Sound detection
- Human detection (image)



Devices on the market

Amazon's Alexa Guard

- Sound detection: footsteps, doorclosing, and glass breaking
- Play siren and turn on lights
- Alert the user through a notification on his phone

Google Nest Cam

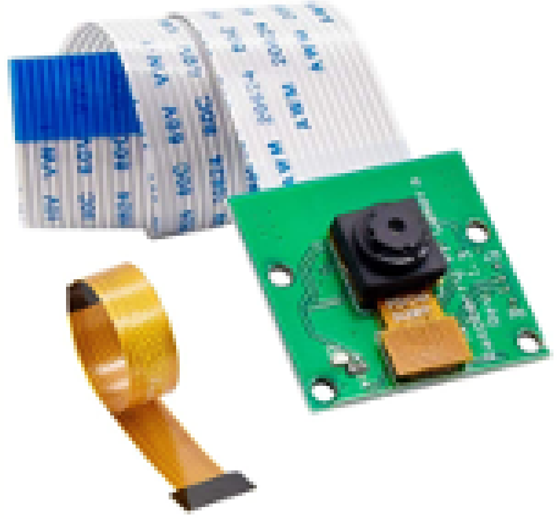
- Object recognition: person, animal, and vehicle
- Records and stores 3h video
- Alert the user through a notification on his phone

💰 Device + monthly subscription

Our equipment

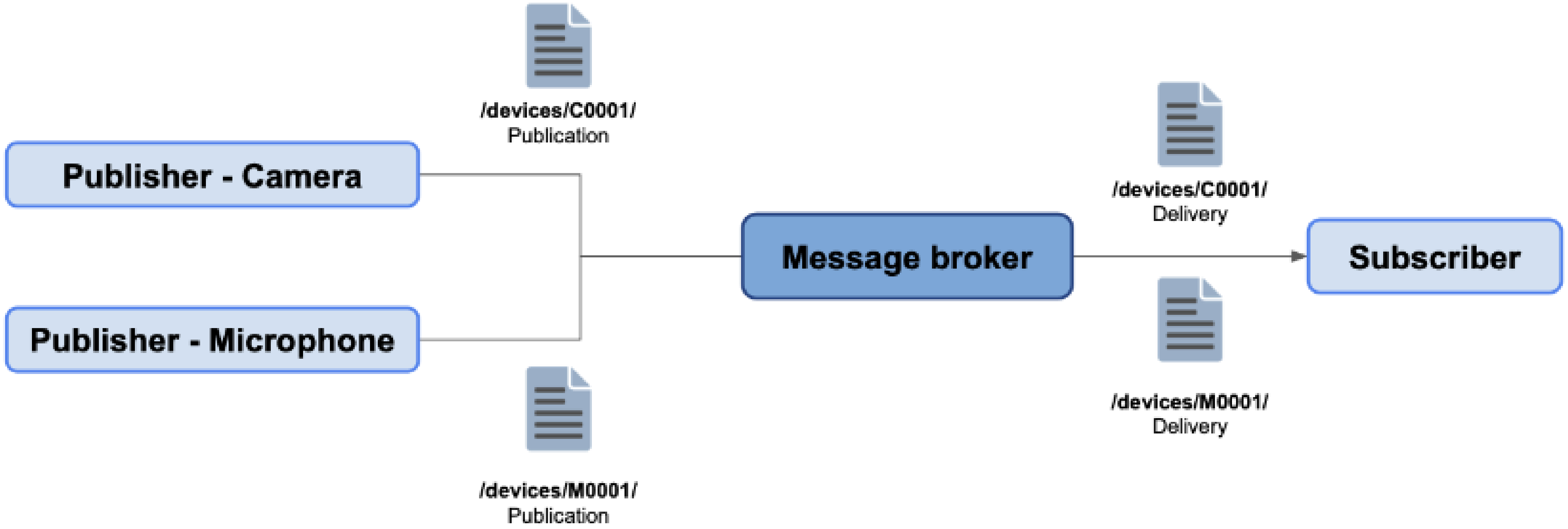
Raspberry Pi 4B 8Gb

USB-Microphone



PiCamera

Communication Paradigm



- **MQTT** communication paradigm
- Two publishers
  - One for the **camera** – /devices/C0001/
  - One for the **microphone** - /devices/M0001/

- One subscriber
  - Receive the notification from both devices and will communicate with the end-user

- Window of 5 minutes to avoid multiple alerts in short time

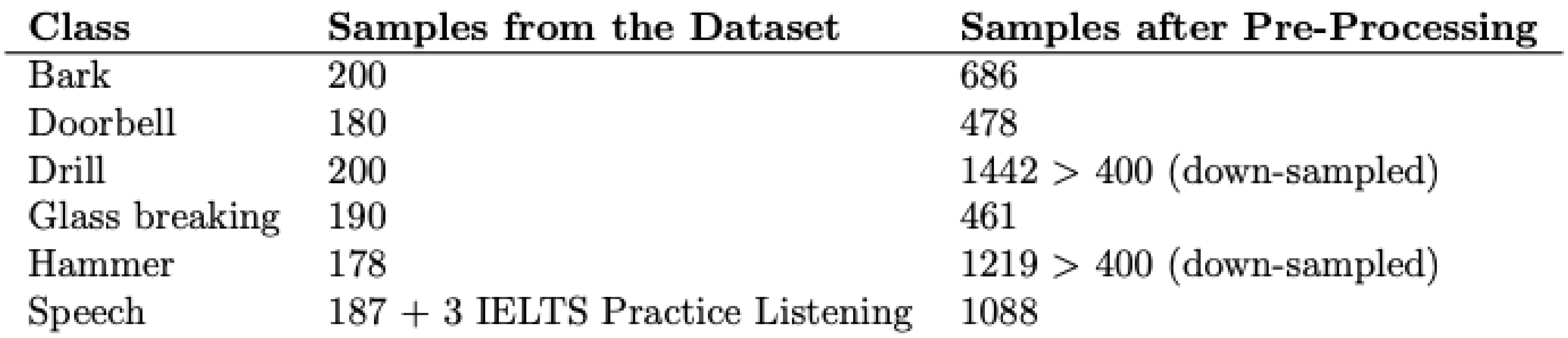
Audio Classification

This task is required in the context of a surveillance system to understand if the perceived sound is due to an intrusion or not.

Trained labels:

- Bark
- Doorbell
- Drill
- Glass breaking
- Hammer
- Speech

Dataset



Class	Samples from the Dataset	Samples after Pre-Processing
Bark	200	686
Doorbell	180	478
Drill	200	1442 > 400 (down-sampled)
Glass breaking	190	461
Hammer	178	1219 > 400 (down-sampled)
Speech	187 + 3 IELTS Practice Listening	1088

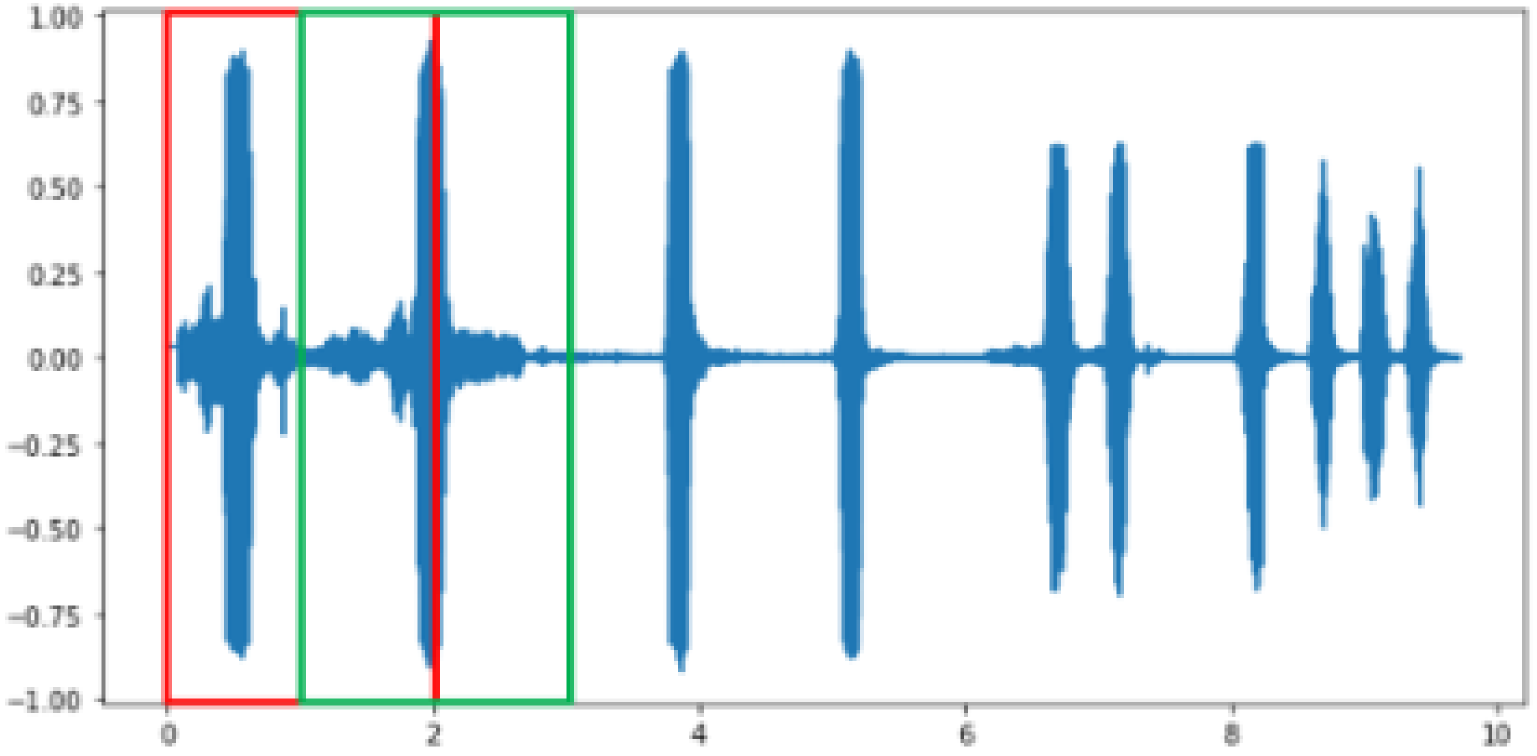
From **FSD50K**, the cleanest and most representative samples of each class have been **manually selected** to train our model. The dataset obtained was **not balanced** after pre-processing to preserve as much information as possible.

Since the audios from this Dataset belonging to the "**Speech**" class are mostly **noisy**, we have added three 8-minute **IELTS** practice listenings, which consist of fairly realistic, good-quality, and noise-free dialogues.

Pre-Processing

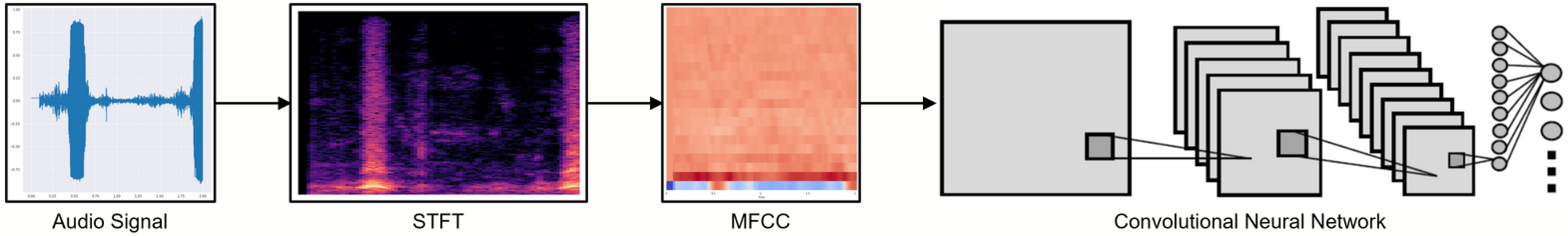
- Generate **2-seconds samples** from the original file with a step of 1 second
  - More samples
  - Augmentation (time-shift)

- Other data augmentation techniques carried out through **audiomentation** did not provide particular benefits



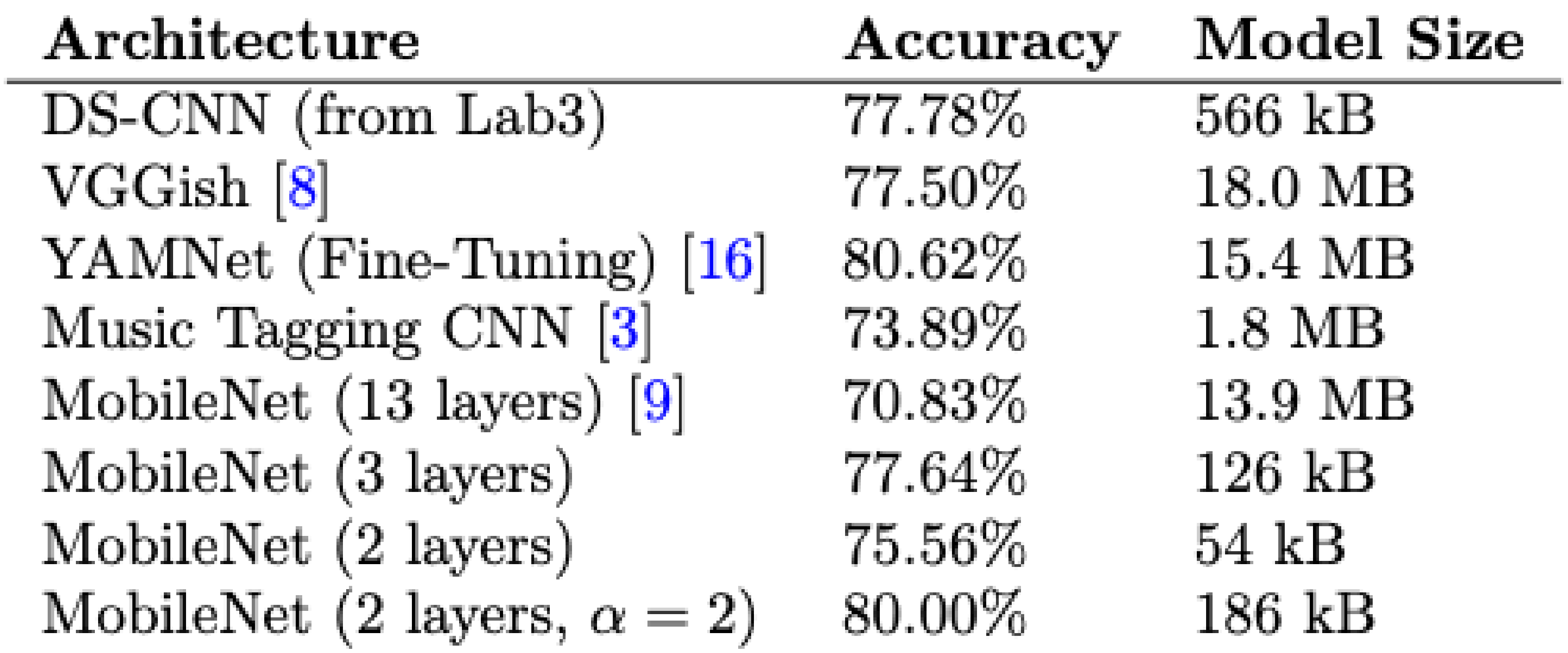
Feature Extraction

- MFCCs
- Settings
  - Frequency : 44.1kHz
  - Frame Length : 80ms
  - Frame step : 40 ms
  - Coefficients : 20
  - Mel bins : 32



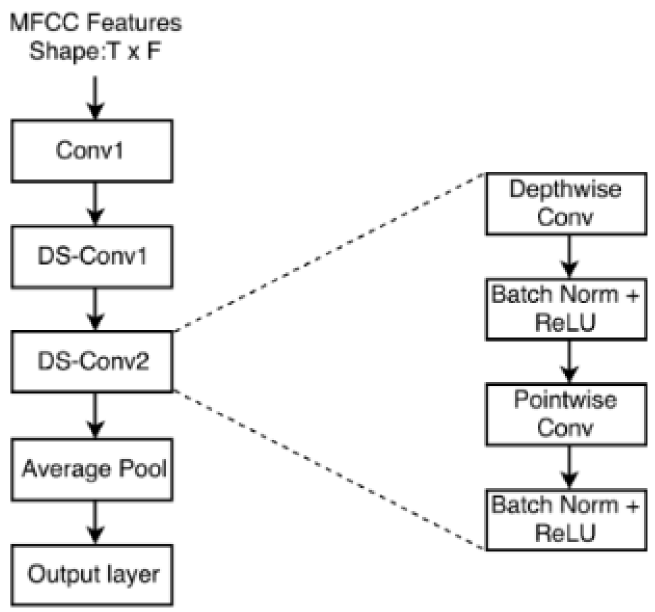
Benchmarks

- Benchmarks are performed on an external, balanced, validation set
  - Manually recorded samples from YouTube
- Testing is performed in real time
- YAMNet is our upper bound (trained on 2M audios)
- MobileNet 2 Layers (alpha=2) is the chosen one

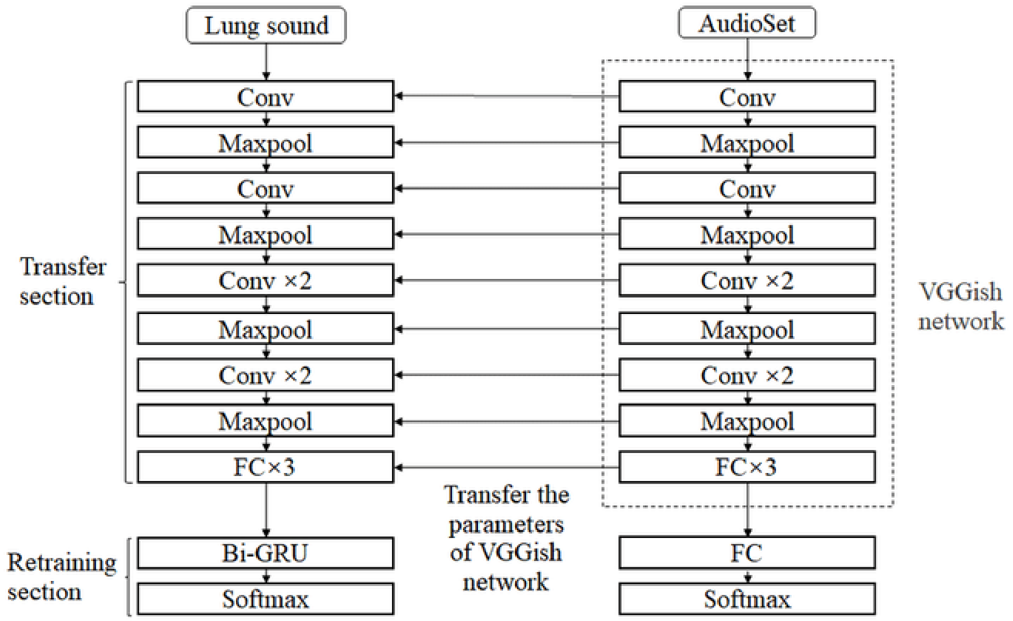


Architecture	Accuracy	Model Size
DS-CNN (from Lab3)	77.78%	566 kB
VGGish [8]	77.50%	18.0 MB
YAMNet (Fine-Tuning) [16]	80.62%	15.4 MB
Music Tagging CNN [3]	73.89%	1.8 MB
MobileNet (13 layers) [9]	70.83%	13.9 MB
MobileNet (3 layers)	77.64%	126 kB
MobileNet (2 layers)	75.56%	54 kB
MobileNet (2 layers, $\alpha = 2$)	80.00%	186 kB

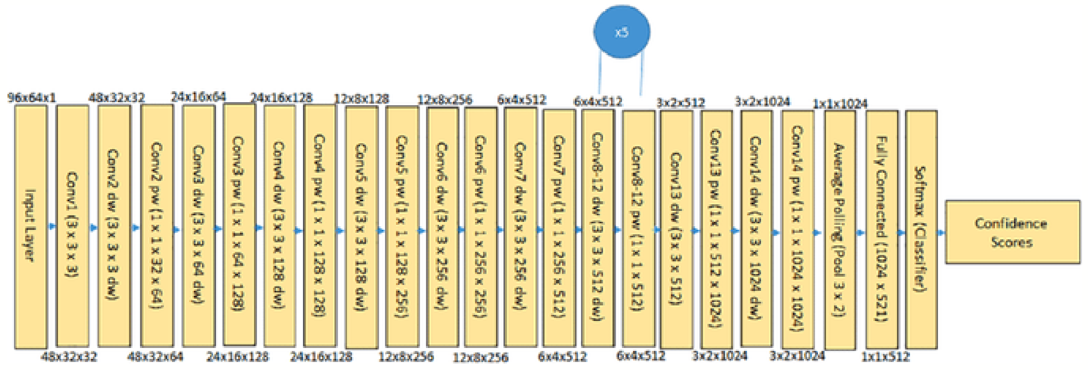
Architectures



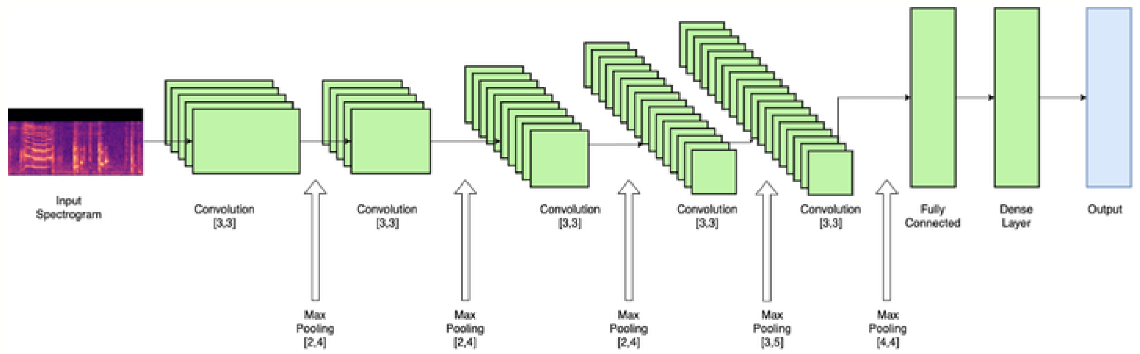
MobileNet (2 layers)



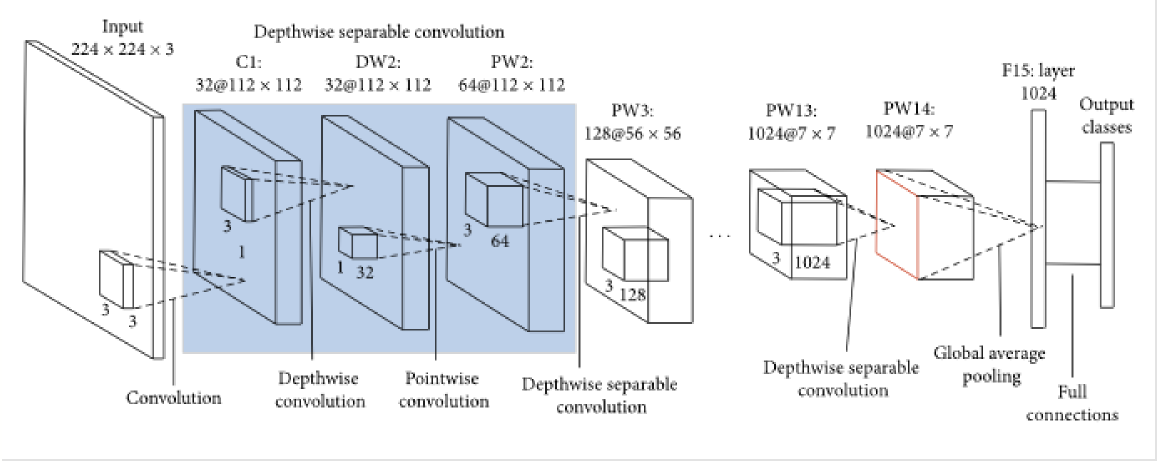
VGGish


x5
96x64x1
48x32x32
24x16x64
24x16x128
12x8x128
12x8x256
6x4x512
6x4x512
3x2x512
3x2x1024
1x1x1024
Input Layer
Conv1 (3 x 3 x 3)
Conv2 dw (3 x 3 x 3 dw)
Conv2 pw (1 x 1 x 32 x 64)
Conv3 dw (3 x 3 x 64 dw)
Conv3 pw (1 x 1 x 64 x 128)
Conv4 dw (3 x 3 x 128 dw)
Conv4 pw (1 x 1 x 128 x 128)
Conv5 dw (3 x 3 x 128 dw)
Conv5 pw (1 x 1 x 128 x 256)
Conv6 dw (3 x 3 x 256 dw)
Conv6 pw (1 x 1 x 256 x 256)
Conv7 dw (3 x 3 x 256 dw)
Conv7 pw (1 x 1 x 256 x 512)
Conv8-12 dw (3 x 3 x 512 dw)
Conv8-12 pw (1 x 1 x 512)
Conv13 dw (3 x 3 x 512)
Conv13 pw (1 x 1 x 512 x 1024)
Conv14 dw (3 x 3 x 1024 dw)
Conv14 pw (1 x 1 x 1024 x 1024)
Average Polling (Pool 3 x 2)
Fully Connected (1024 x 521)
Softmax (Classifier)
Confidence Scores
48x32x32
48x32x64
24x16x128
24x16x128
12x8x256
12x8x256
6x4x512
6x4x512
3x2x1024
3x2x1024
1x1x512

YAMNet



Music Tagging CNN



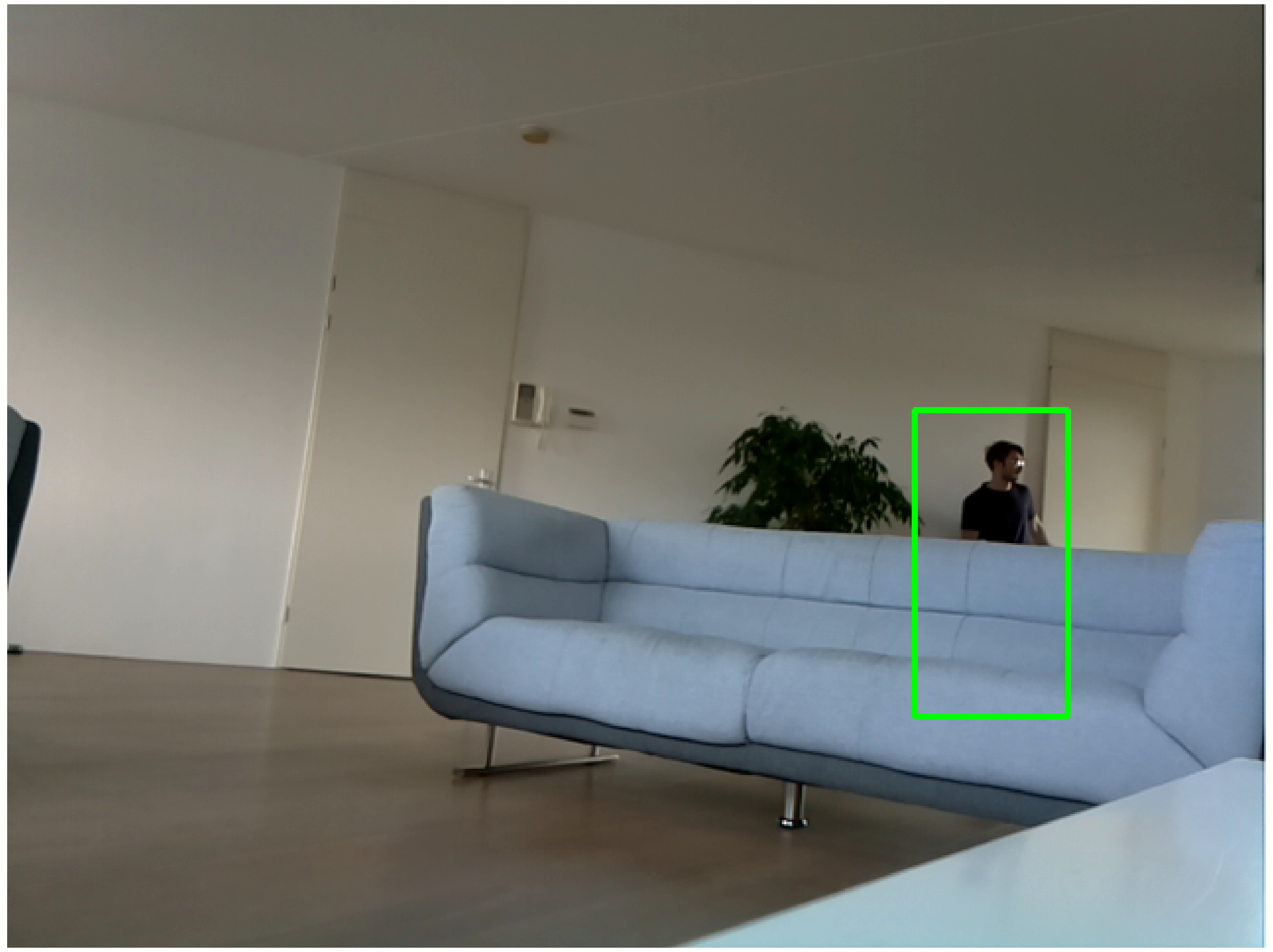
MobileNet (13 layers)

Human Detection

- PiCamera is constantly available
- Check frame by frame the presence of a human
- Used pre trained models
  - Histogram of Gradients for features extraction
  - Linear SVM

Algorithm

- The inference is performed through OpenCV's 'detectMultiScale' method
  - detects objects of different sizes in the input image and therefore it will returns a list of rectangles
- Two hyperparameters have been tuned:
  - winStride (10,10) : sliding window for SVM
  - scale (1.01) : scale factor



Post processing

- Non-Maximum Suppression
  - The algorithm will select the predictions with the maximum confidence and suppress all the other predictions having overlap with the selected predictions greater than the given threshold, that is 1.00 on our case.

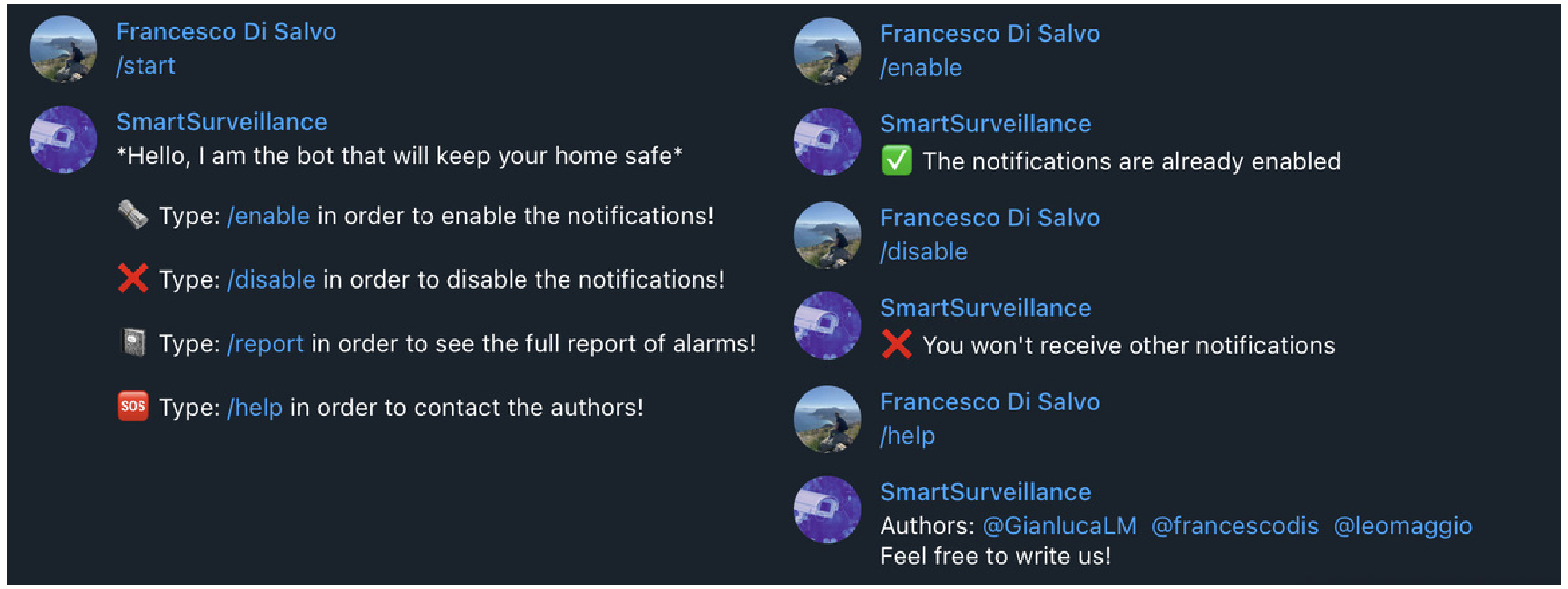
User Interface

Telegram Bot

- Activation
- Functionalities and commands:
  - intrusion notifications
  - enable
  - disable
  - report
  - help

Implementation

- telegram.ext package
- The Bot API
  - sendPhoto(chat_id,photo,caption="")



Francesco Di Salvo
/start

SmartSurveillance
Hello, I am the bot that will keep your home safe

🗞 Type: /enable in order to enable the notifications!

❌ Type: /disable in order to disable the notifications!

📓 Type: /report in order to see the full report of alarms!

🆘 Type: /help in order to contact the authors!

Francesco Di Salvo
/enable

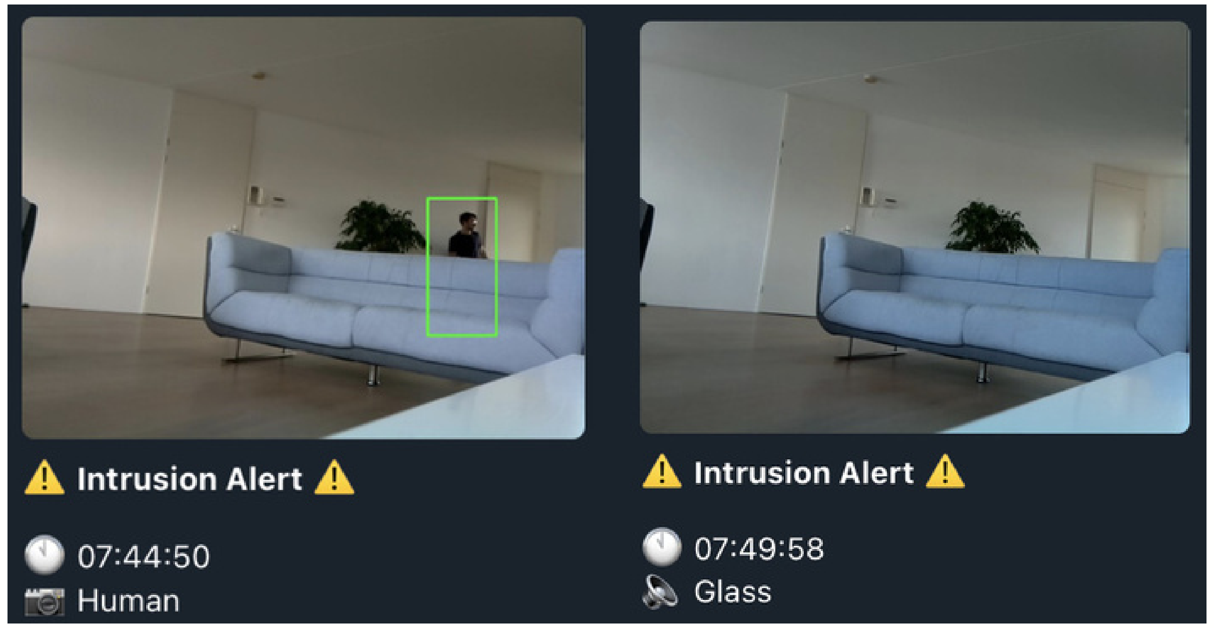
SmartSurveillance
✅ The notifications are already enabled

Francesco Di Salvo
/disable

SmartSurveillance
❌ You won't receive other notifications

Francesco Di Salvo
/help

SmartSurveillance
Authors: @GianlucaLM @francescodis @leomaggio
Feel free to write us!



⚠ Intrusion Alert ⚠

🕐 07:44:50
📷 Human

⚠ Intrusion Alert ⚠

🕐 07:49:58
📡 Glass

Limitations and further improvements

Classification task

- More classes
  - Audio : thunder, footsteps, bell ...
  - Video : animals, house owners ...

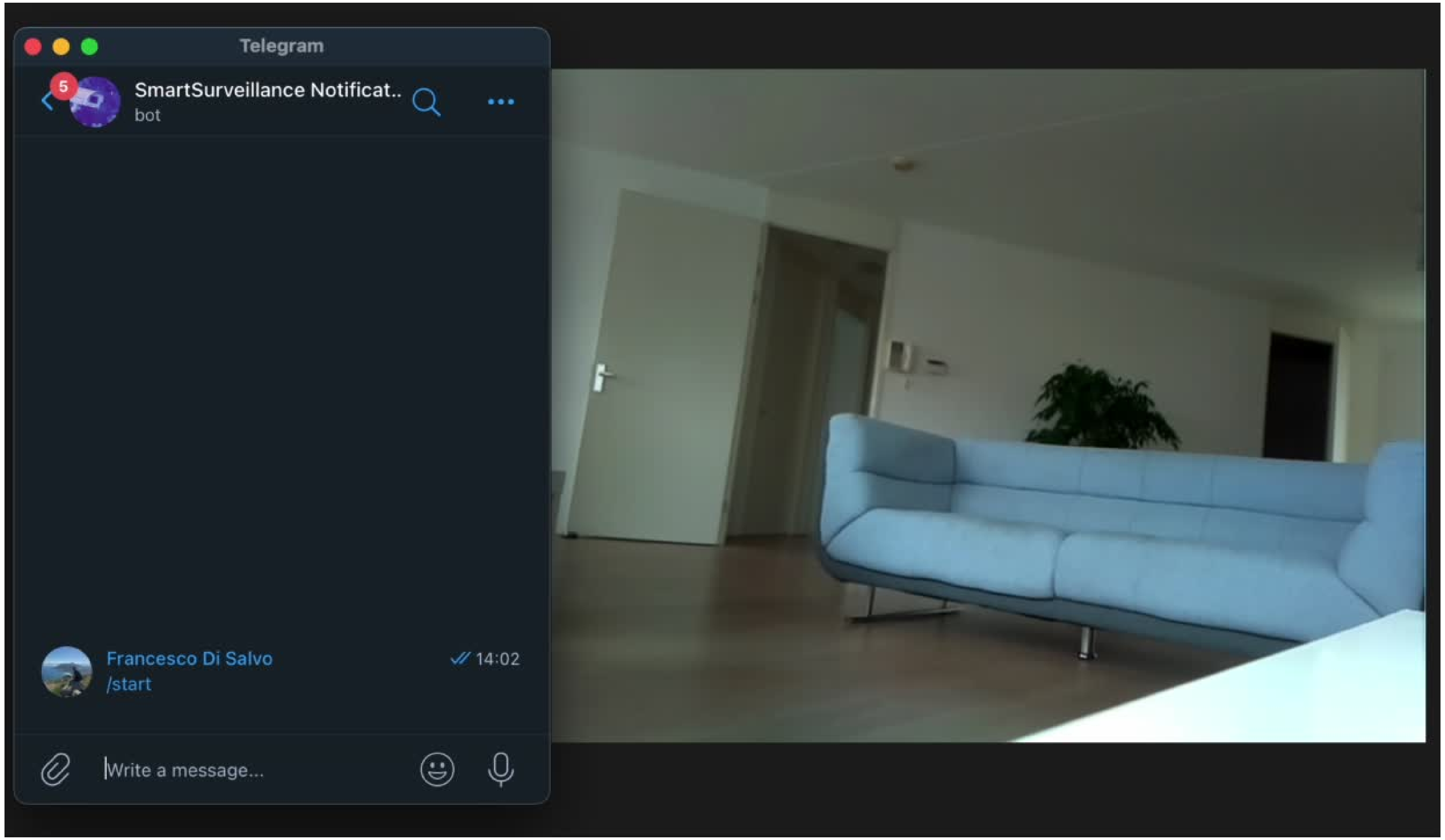
Hardware

- Better microphone
- Night vision on camera
- Ecosystem
  - Speaker
  - Lights

Storage

- Periodic transfer of the images

Human Detection



Telegram

SmartSurveillance Notificat..
bot

Francesco Di Salvo 14:02
/start

Audio recognition

Francesco Di Salvo 14:24
/start

Glass

Francesco Di Salvo 14:41
/start

Drill

Francesco Di Salvo 14:32
/start

Speech

Francesco Di Salvo 14:43
/start

Hammer

Thank you for your attention

Report & Code:

https://github.com/francescodisalvo05/smart-surveillance-raspberrypi