



Politecnico
di Torino

Smart Home Vigilance System on Raspberry-Pi

Politecnico di Torino
Machine Learning for IoT

Matteo Merlo

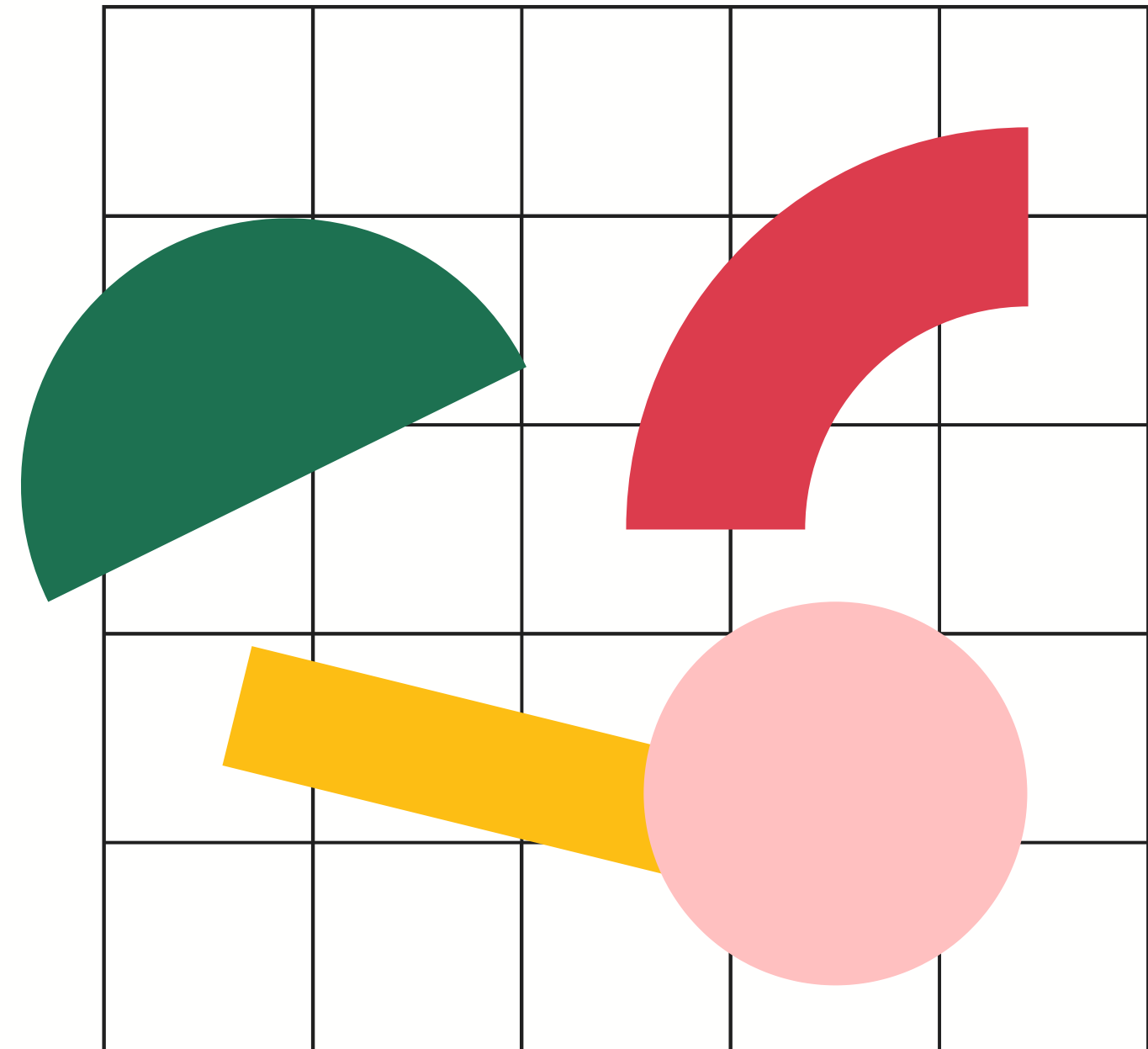
s287576

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 Home Vigilance:

- 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

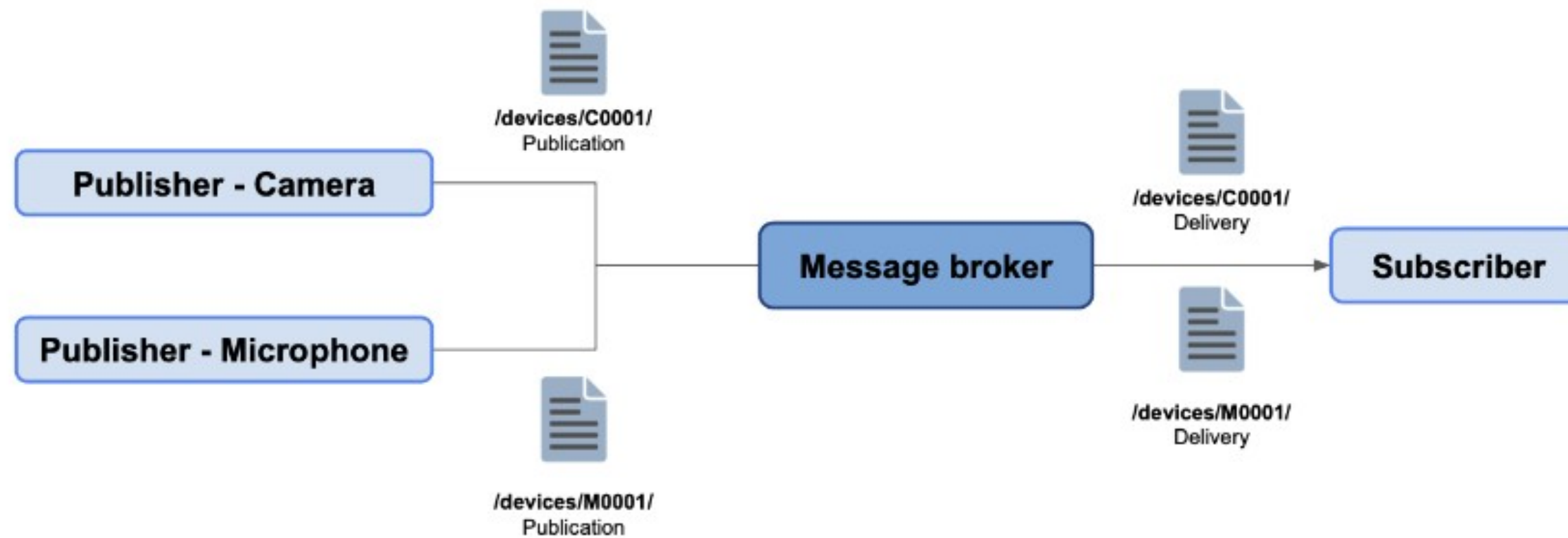


USB-Microphone



PiCamera

Communication Paradigm



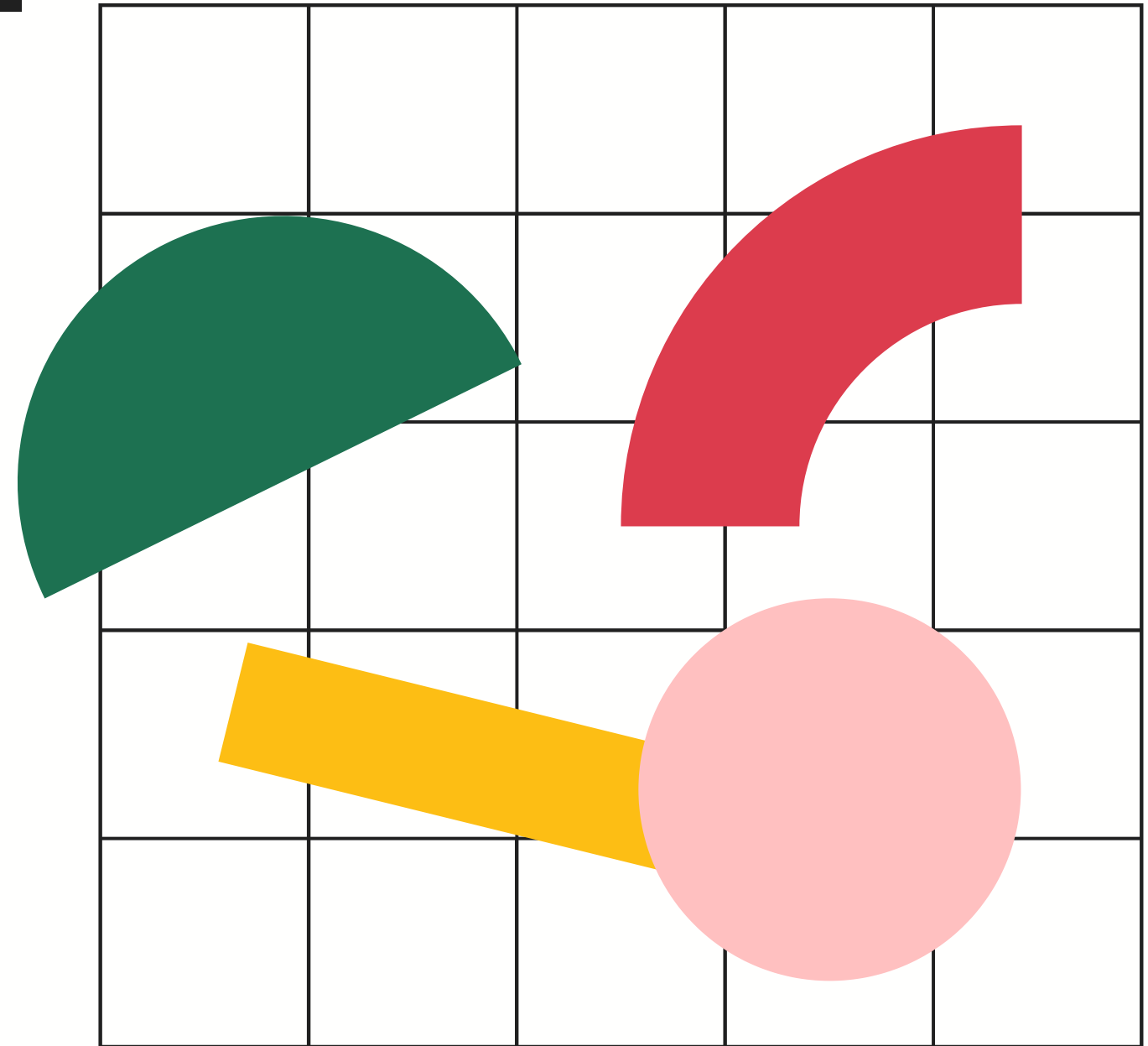
- MQTT communication paradigm
- Two publishers
 - One for the camera - /devices/C0001/
 - One for the microphone - /devices/M0001/
- One subscriber
 - Receive the notification from both devices and will communicate with the end-user
- Window of 5 minutes to avoid multiple alerts in short time

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

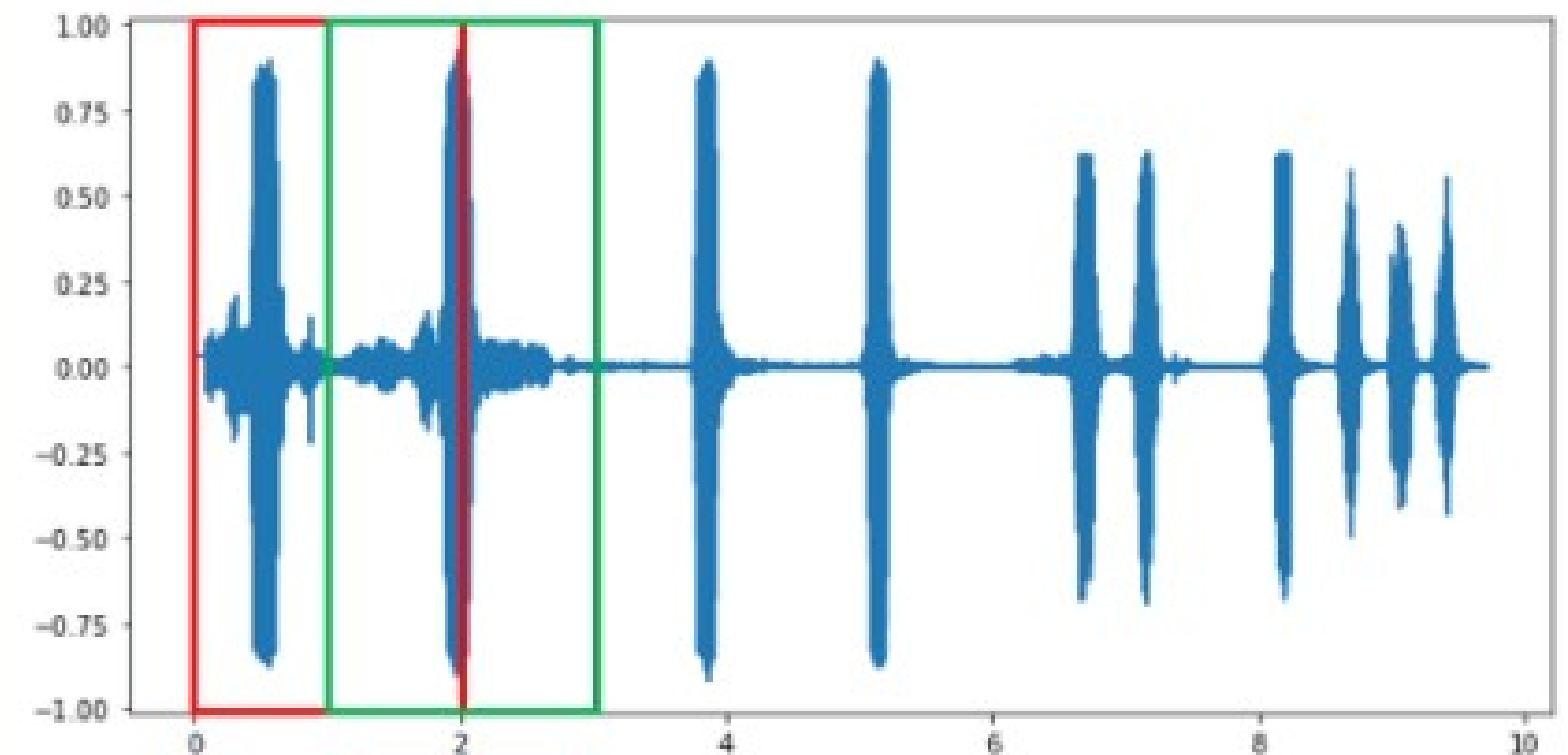
Class	Samples from the Dataset	Samples after Pre-Processing
Bark	200	686
Doorbell	180	478
Drill	200	1442 > 400 (down-sampled)
Glass breaking	190	461
Hammer	178	1219 > 400 (down-sampled)
Speech	187 + 3 IELTS Practice Listening	1088

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.

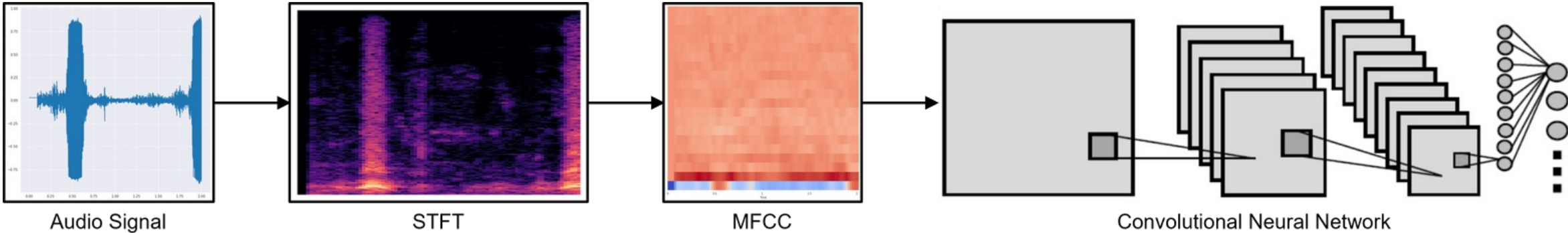
Pre-Processing

- 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



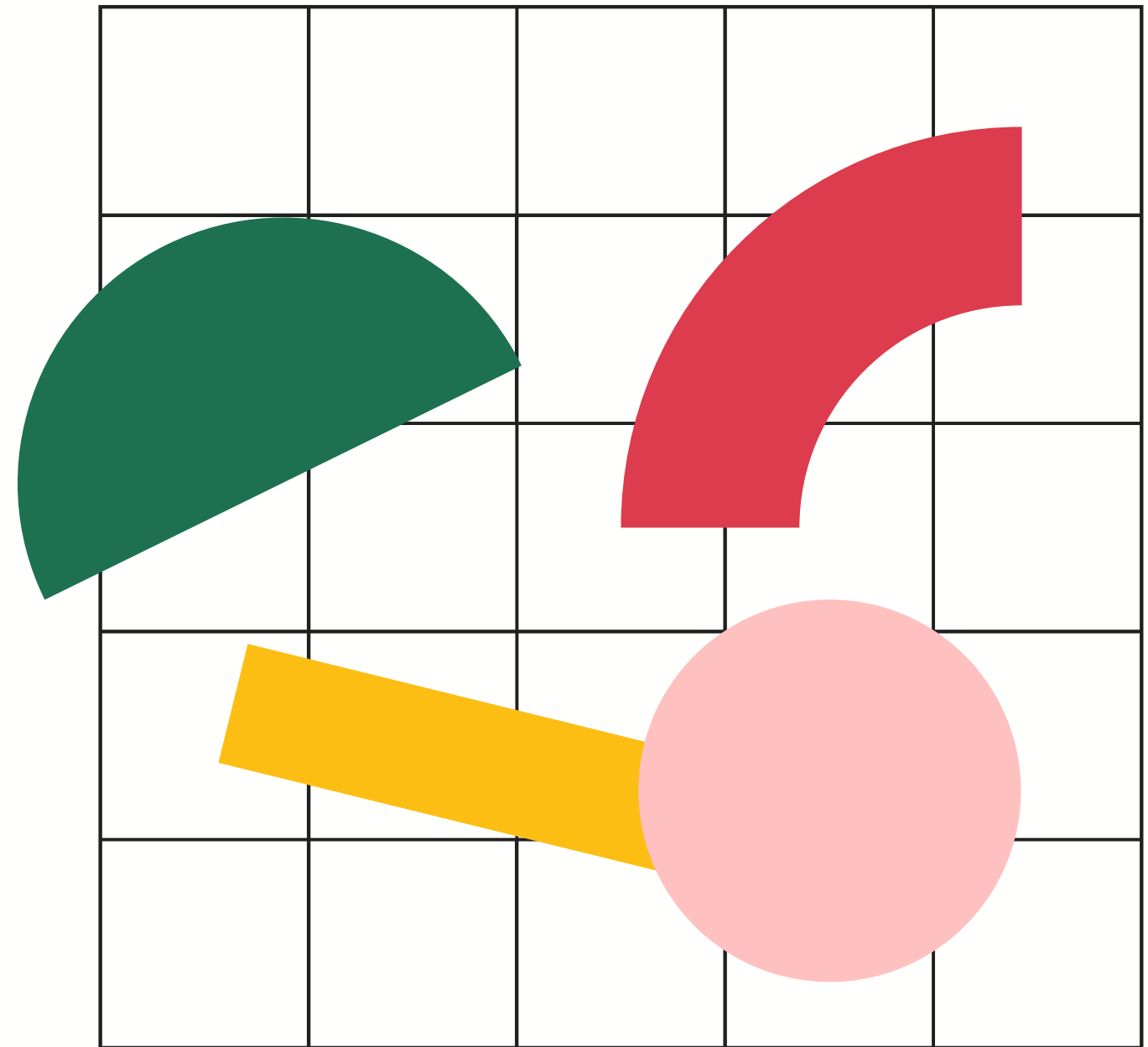
Benchmarks

- 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

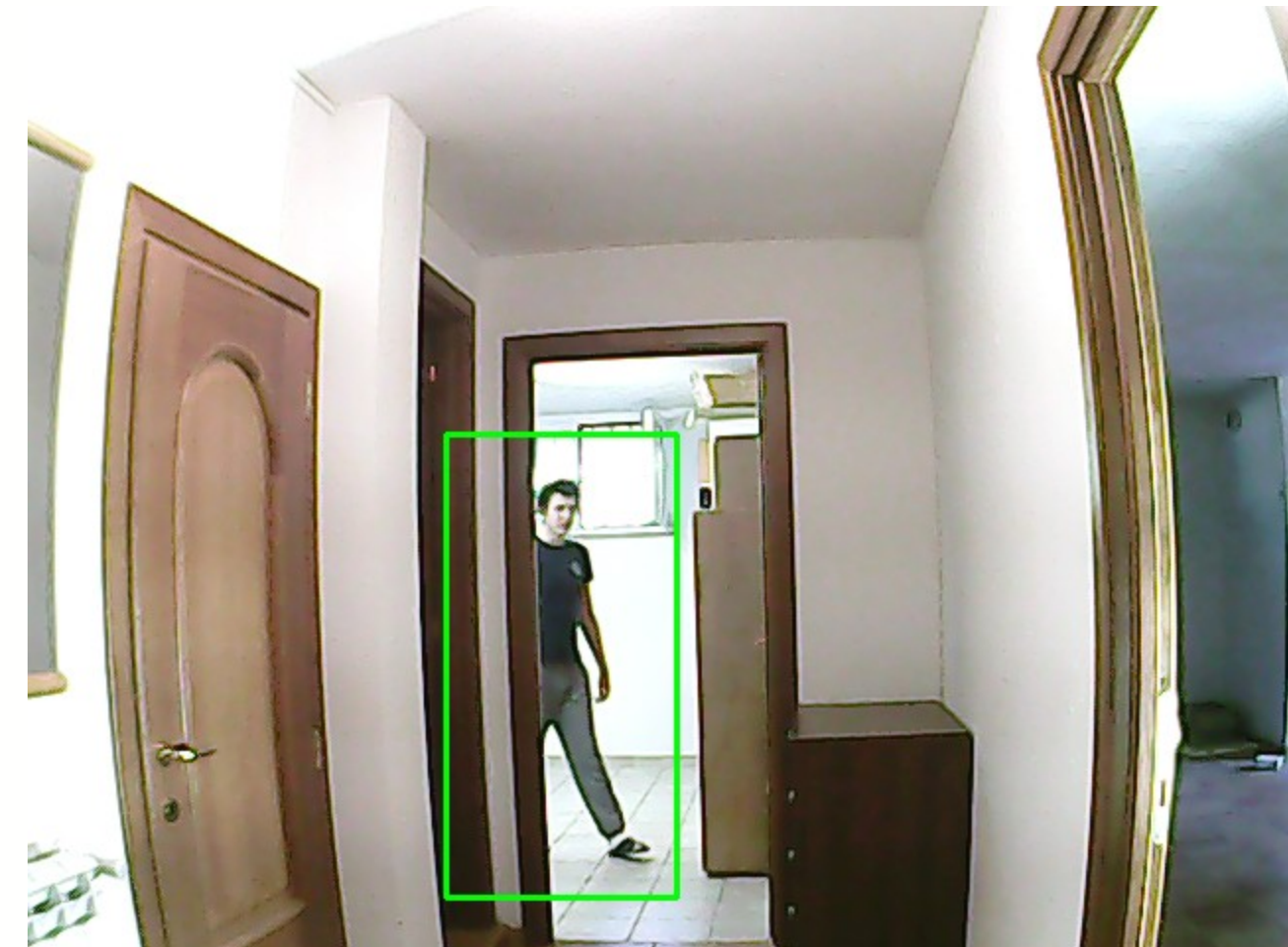
Human Detection

- 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 return 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.

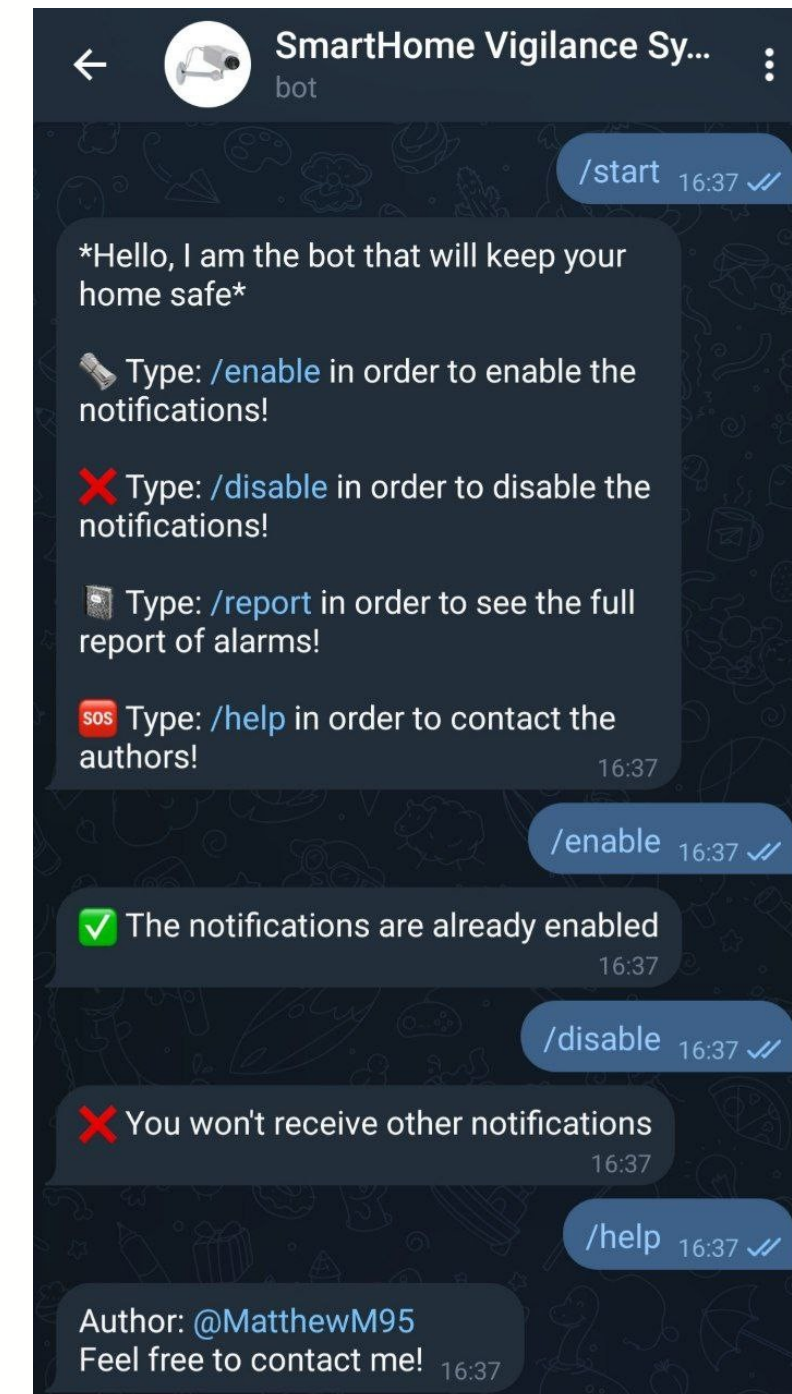
User Interface

Telegram Bot

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

Implementation

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

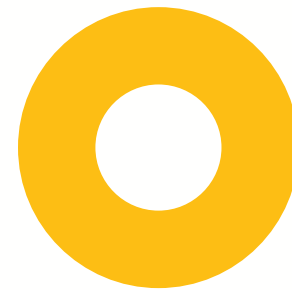


Limitations and further improvements



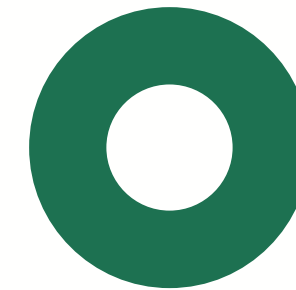
Classification task

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



Hardware

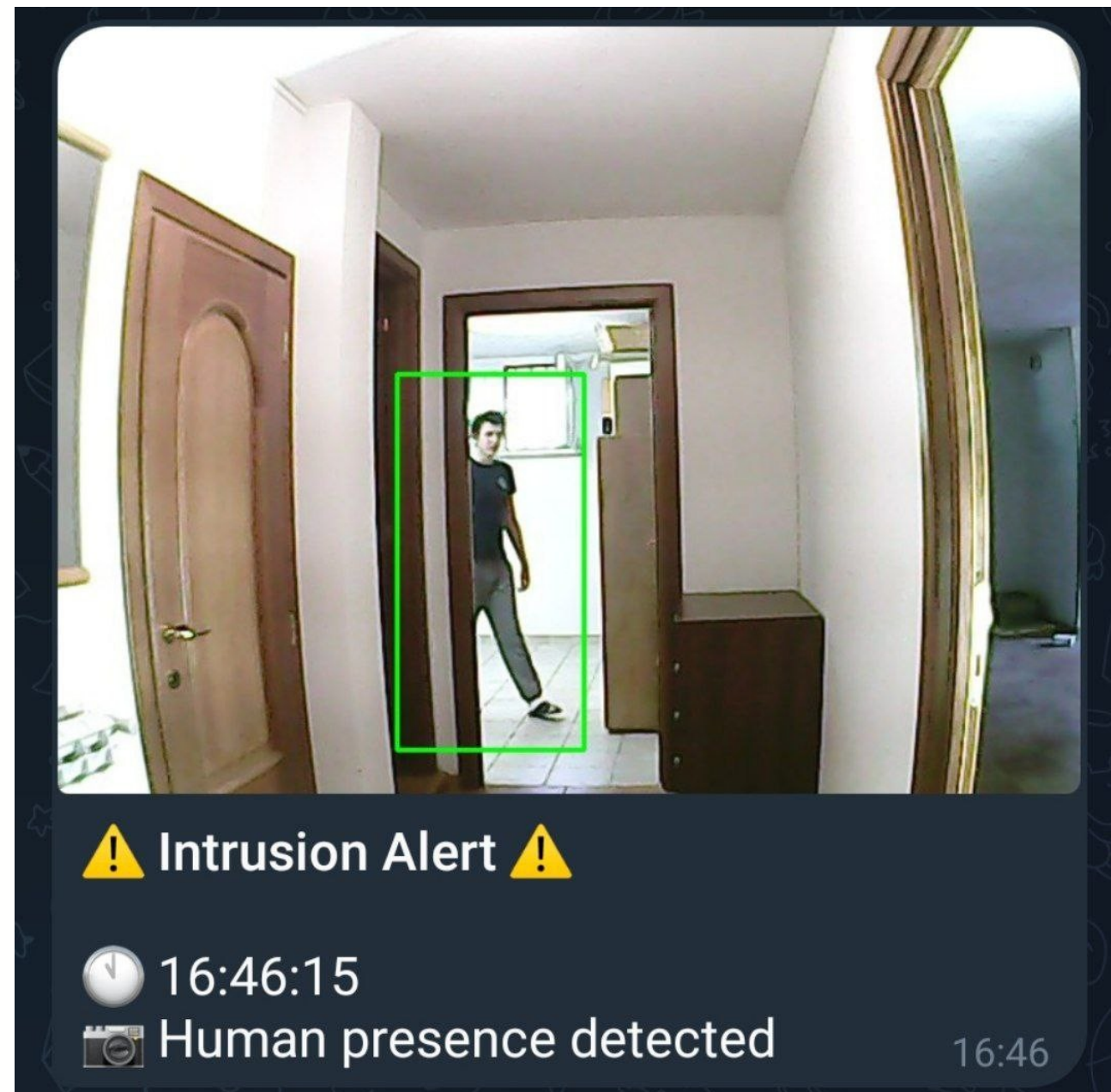
- Better microphone
- Night vision on camera
- Ecosystem
 - Speaker
 - Lights



Storage

- Periodic transfer of the images

Human Detection



Audio Recognition



Thank you for your attention

Report & Code:

<https://github.com/MatteoM95/Smart-Home-Vigilance-System>