EMERGENCY SIRENS DETECTION

Alessia Leclercq
Chiara Van der Putten

ML4IOT
A.A 2022/2023
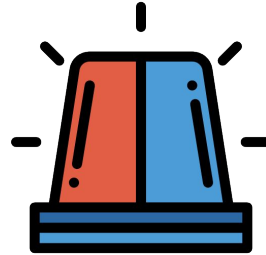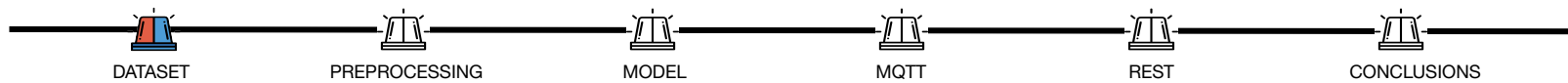
DATASET

PREPROCESSING

MODEL

MQTT

REST

CONCLUSIONS

DATASET

Two labelled classes:

- Emergency vehicle sirens (932 audio)
- Traffic noise (902 audio)

Recording at various distance points (10-45 meters)

Different Data Collection method:

- Live Camera
- Online Sources
- Placing Emergency Sirens

Politecnico
di Torino

Paper: Large-scale audio dataset for emergency vehicle sirens and road noises (nature.com)
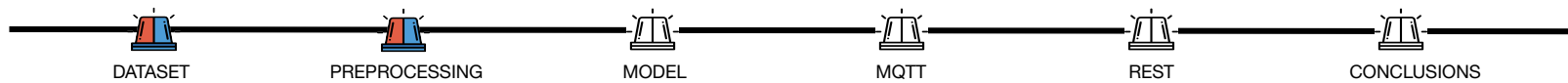
PREPROCESSING

- Audio format converted into WAV

- Up-sampling of the audio to 48 kHz

- Cut audio to 3 seconds

- Padding

- Conversion to MFCC

- Reshape 32x32

- Train/Test/Val split 65/25/10 [%]

Frame length [s]	0.016
Frame step [s]	0.016
Downsampling rate	48000
Number of mel bins	20

Number of coefficients	20
Lower frequency	20
Upper frequency	24000

Politecnico di Torino

conversion, upsampling, cut operations performed using ffmpeg

4
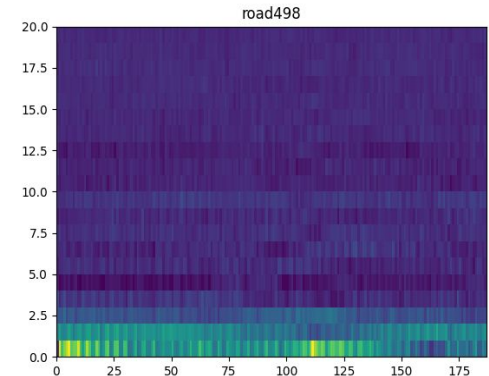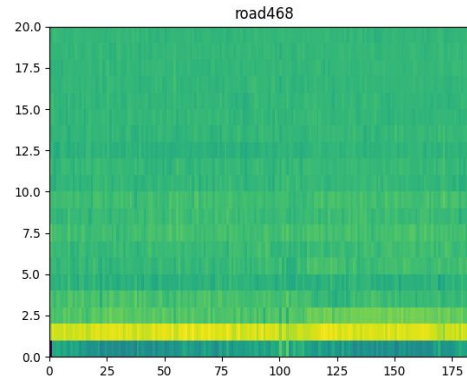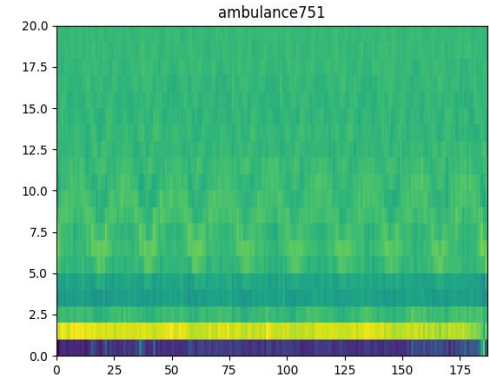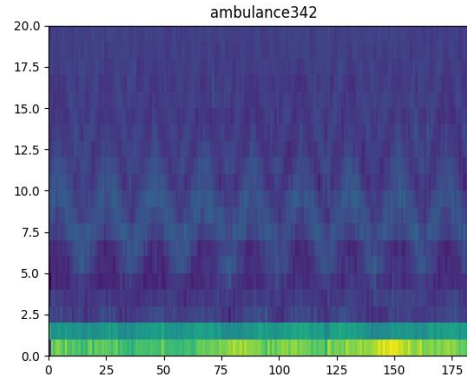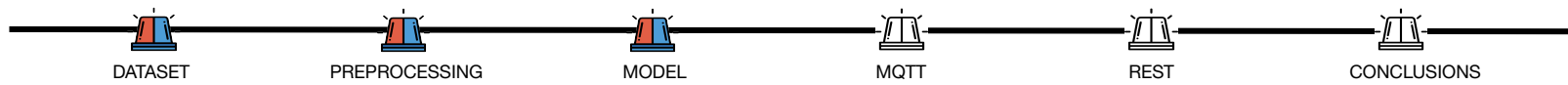
MODEL
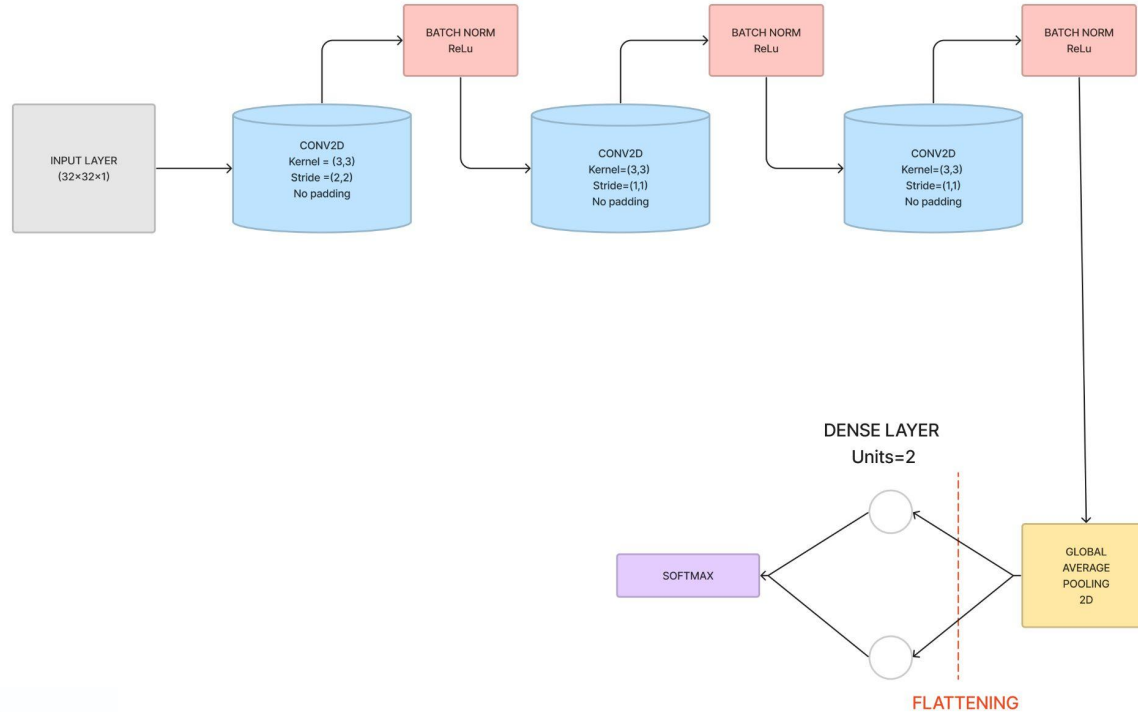
Batch size	20
Initial learning rate	0.01
End learning rate	1e-5
Epochs	20
Number of filters	64
Pruning alpha	0.2
Initial sparsity	0.2
Final sparsity	0.6

RESULT

Confusion Matrix

Training Loss and Accuracy
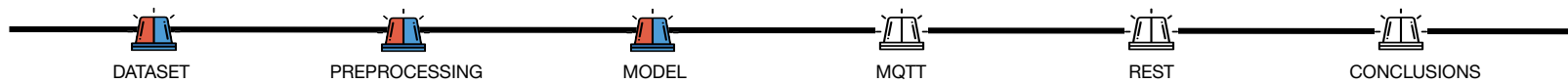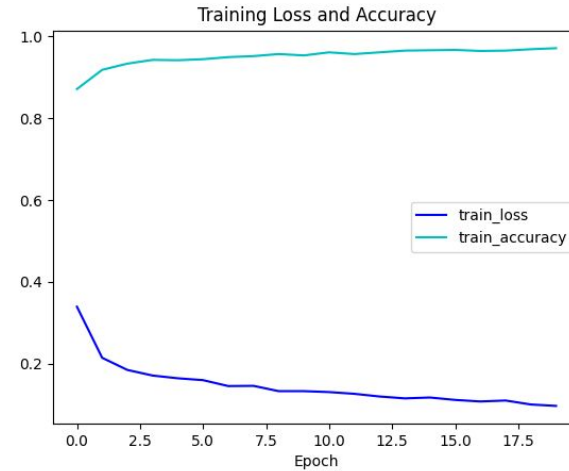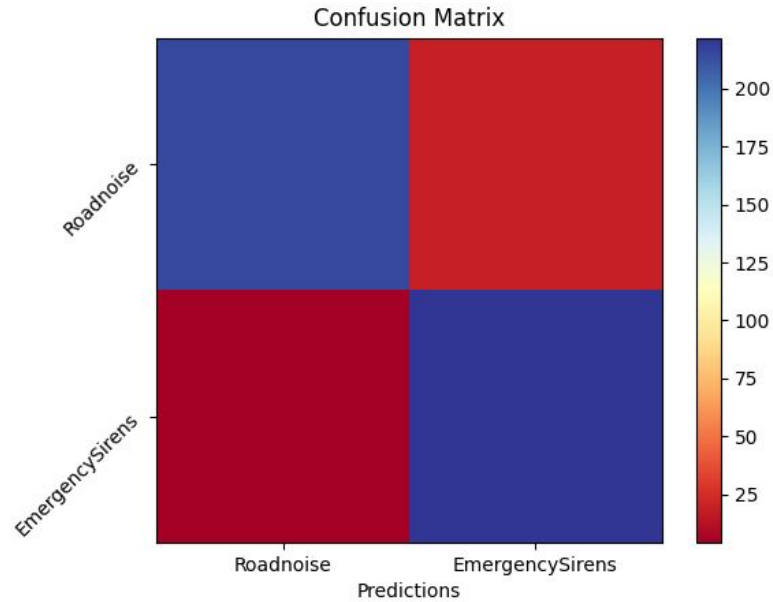
Accuracy	95.000%
F1	0.95
Tflite size	31.3 [KB]
Tflite size (.zip)	9.36 [KB]
Median latency	31.4 [ms]
Model parameters	5610 (2798 trainable)

|	RoadNoises	EmergencySirens
Precision	0.98	0.92
Recall	0.92	0.98

Politecnico di Torino

MQTT

The *publisher* gets the prediction (i.e., it detects an emergency siren) and it sends the MQTT broker the following:

- Sensor name
- Timestamp
- "1"

The *subscriber* stores the received data into a Redis time series. The series are named according to the following rule: "*{sensor_name}:emergencySiren*". When a new* {*sensor_name*} is received, then the related Redis time series is also created.

*new means that no Redis time series is associated to {sensor_name}:emergencySiren
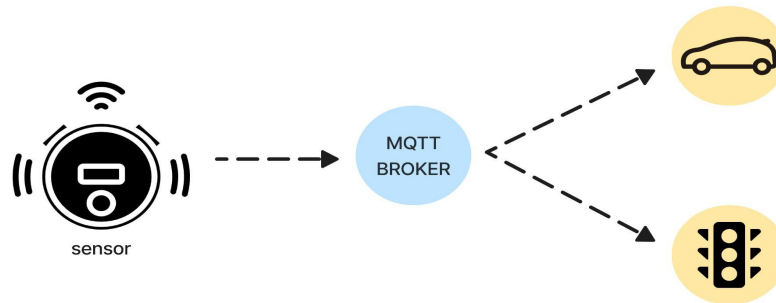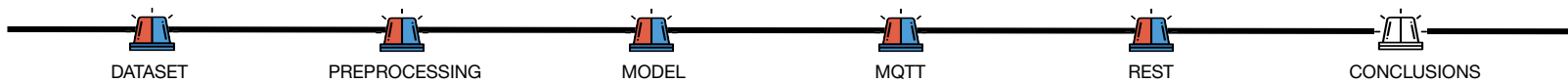
REST API

The *server* implements three endpoints:
- DeleteSensorHistory
- ListSensors
- SensorHistory

Endpoint	Path	Query
DeleteSensorHistory	optional	no
ListSensors	no	no
Sensor History	optional	optional

The *client* implements the following actions:
- *Geolocation* of a sensor detecting an emergency siren using folium
- *Computation of statistics* at different granularity levels (sensor or district / year, month, day, day of the week, etc.).
- *Deletion* of time series
- *List* of sensors associated to a time serie

CLIENT   REST API   REDIS

GET / DELETE

JSON
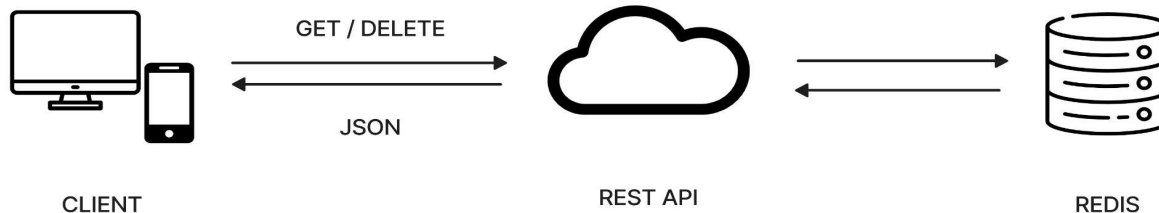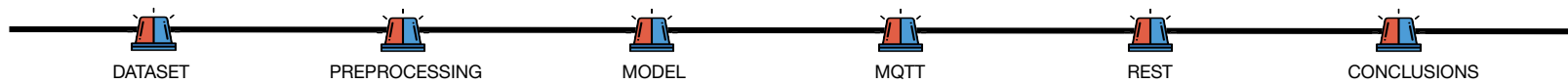
POSSIBLE FUTURE APPLICATION

- Self-driving Car

- Smart Traffic Light

- Road Traffic Monitoring

- Driver Notification

Politecnico
di Torino

LIMITATIONS AND INTEGRATIONS

Limitations:

- Presence of clacsons
- Multiple Emergency sirens together

Integrations:

- Classification of different emergency sirens
- Increase the dataset using also audio recorded inside the vehicles
- Increment variability in the road noise dataset

Politecnico
di Torino

THANK YOU FOR YOUR ATTENTION :)