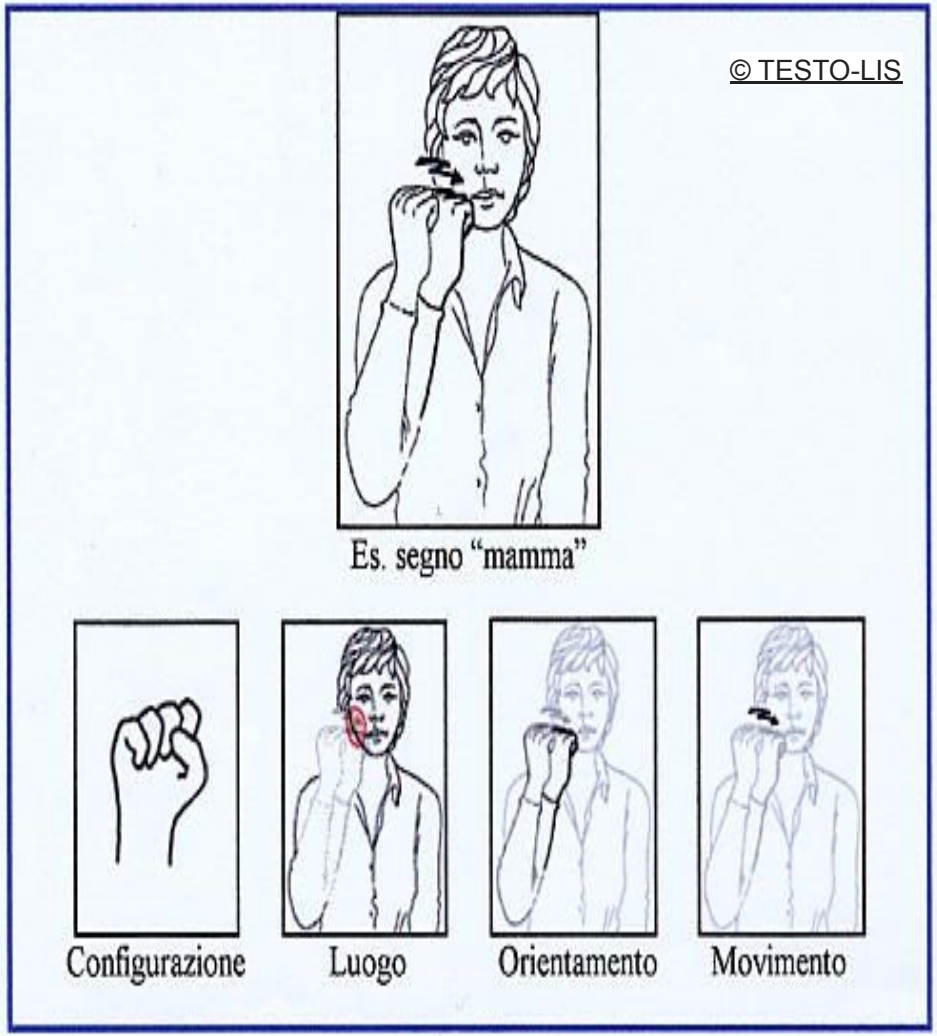
Sistema di riconoscimento della lingua dei segni per applicazioni smart home

Sofia Cerè; 0001057906
Laboratorio di Making
LM in Informatica
Alma Mater Studiorum - University of Bologna

Lingua dei segni

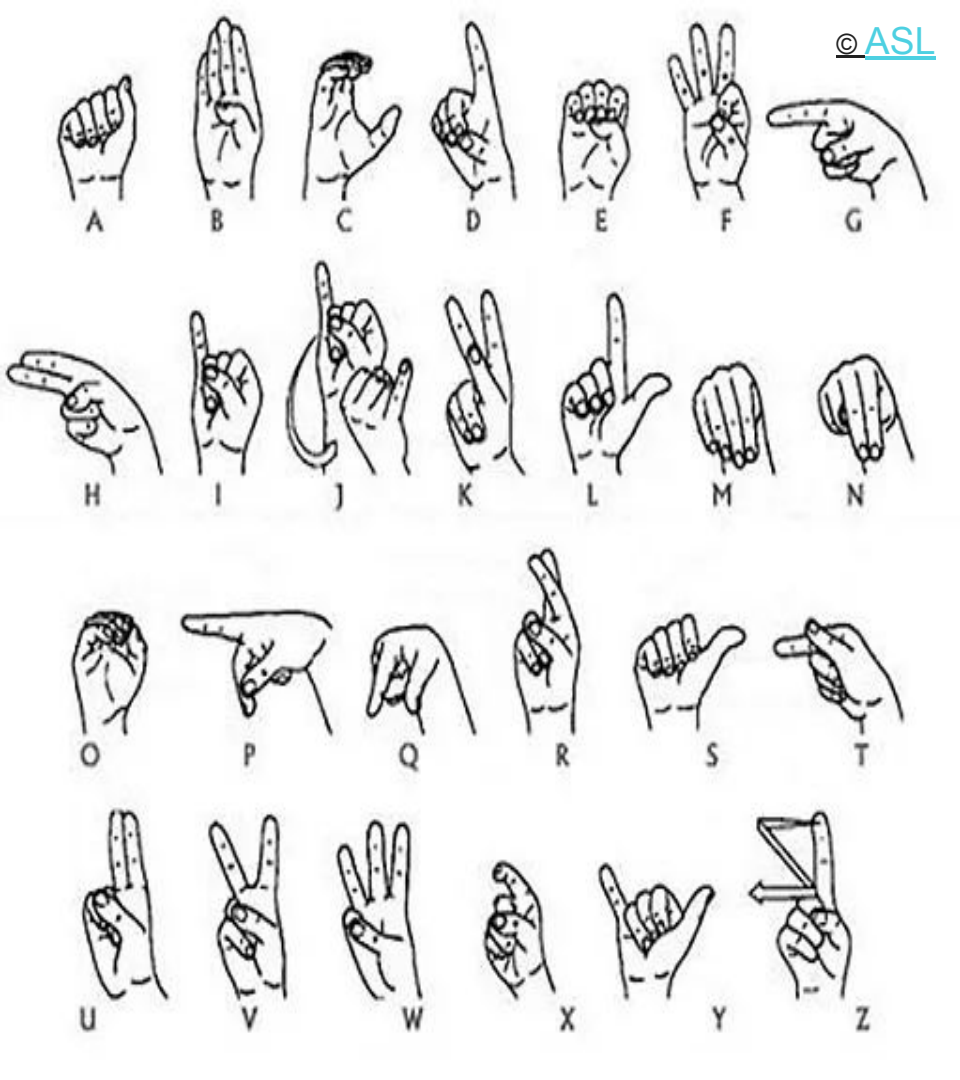
La lingua dei segni è la comunicazione manuale comunemente usata dalle persone sorde. La lingua dei segni non è universale: persone sorde di paesi diversi parlano lingue dei segni diverse. [...].

Ogni segno ha tre parti distinte: la forma della mano, la posizione delle mani e il movimento delle mani.



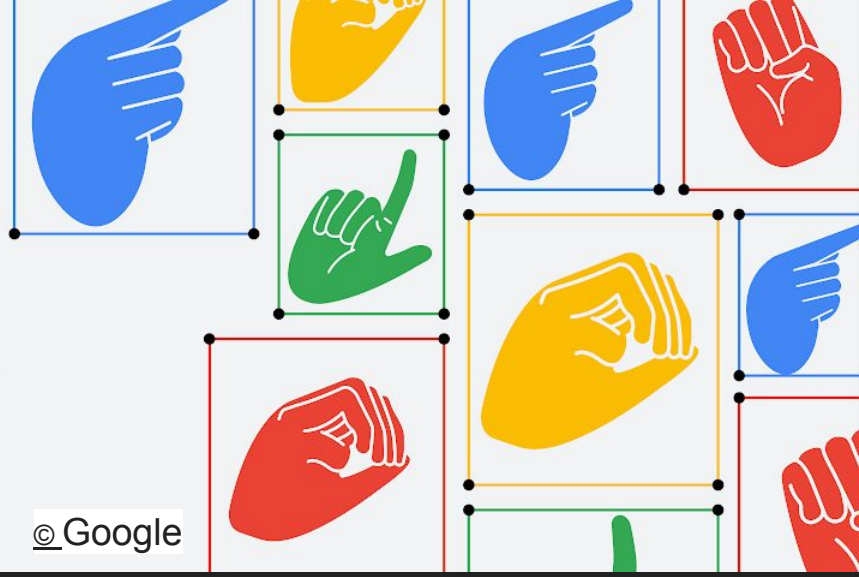
Dattilologia

Con il termine dattilologia (da 'dattilos', dito e 'logos' discorso, studio) ci si riferisce all'alfabeto. E’ fondato su configurazioni statiche delle mani, è uno dei mezzi di comunicazione visivo-gestuali più semplici: consiste, com'è noto, nel formare con le dita e la mano le lettere dell'alfabeto. L’alfabeto manuale, ora in uso, è caratterizzato dall'uso di una sola mano e dall'utilizzo esclusivo dello spazio neutro davanti al segnante come luogo di esecuzione.

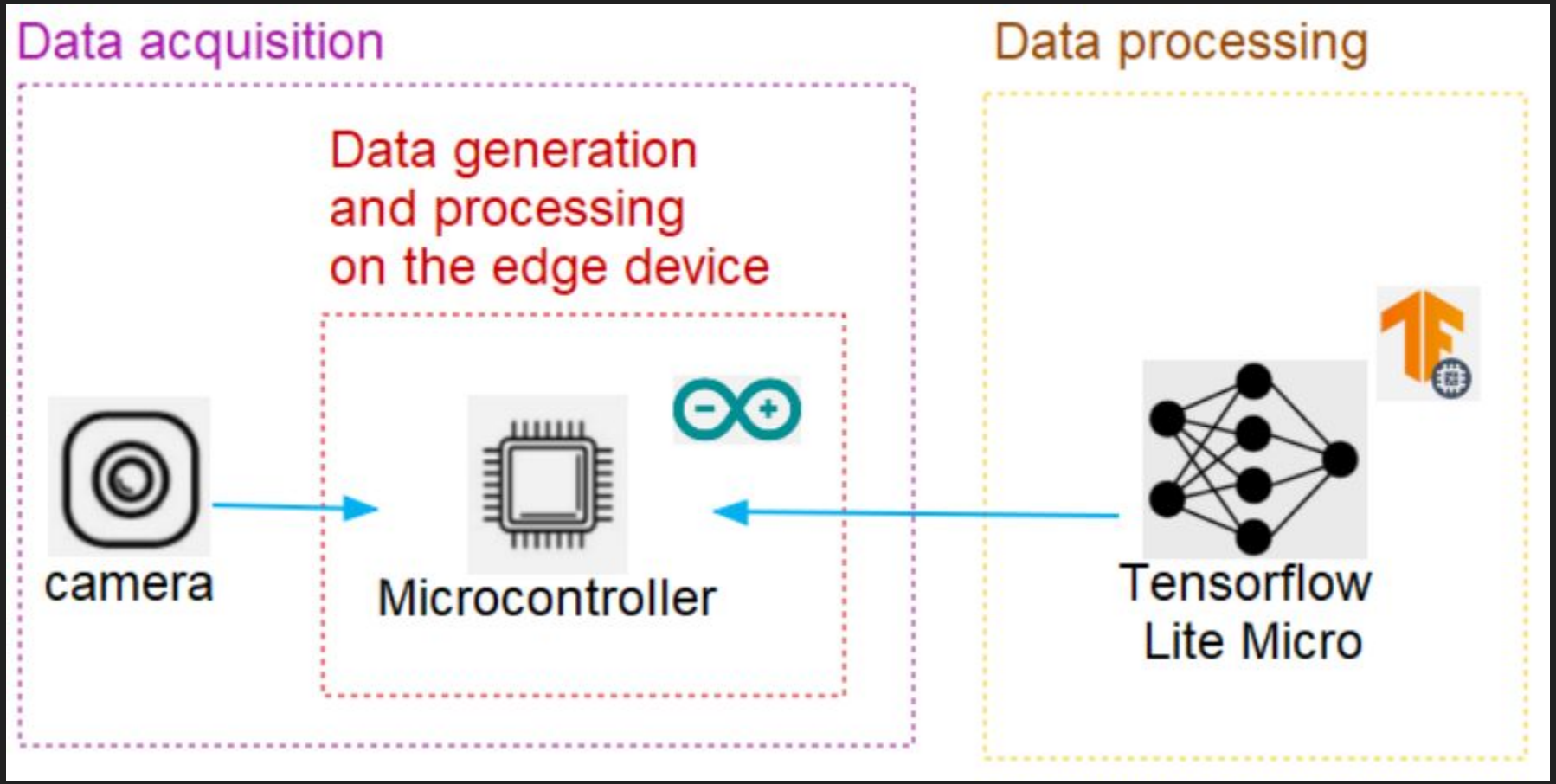


Google

American Sign Language Fingerspelling Recognition



To that end, we're excited to announce the release of one of the largest datasets of ASL fingerspelling and a Kaggle ML competition that will award $200k in prizes to ML engineers who develop the most accurate ASL fingerspelling recognition models using MediaPipe and TensorFlow Lite. The winning models will be open sourced to help developers add support for fingerspelling to their apps.


Data acquisition
Data processing
Data generation
and processing
on the edge device
camera
Microcontroller
Tensorflow
Lite Micro

Arduino Tiny Machine Learning Kit

- Arduino Nano 33 BLE Sense board
- OV7675 Camera
- Arduino Tiny Machine Learning Shield
- USB A to Micro USB Cable

ASL Dataset

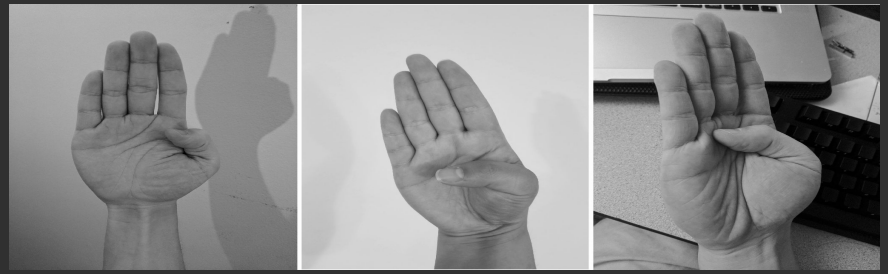
David Lee. (2022). American Sign Language Letters Dataset

"Joseph Nelson." Roboflow Blog, Oct 19, 2020. https://blog.roboflow.com/computer-vision-american-sign-language/

Creazione del dataset

- Estrazione delle lettere “A” e “B” dal dataset ASL
- Aggiunta lettere “A” e “B” fotografate con la camera di Arduino
- Scala di grigi
- 6’084 immagini
- Split 80/20

Esempio: lettera “B”



Two-class Classification

Sign = 1 = “B”

Not-Sign = 0 = “A”

Edge Impulse Project

Sign Language Detection A&B

NN Architecture

Layers supported by TF Lite Micro ops

On Device Performance

Conversione del modello nel formato TFLM

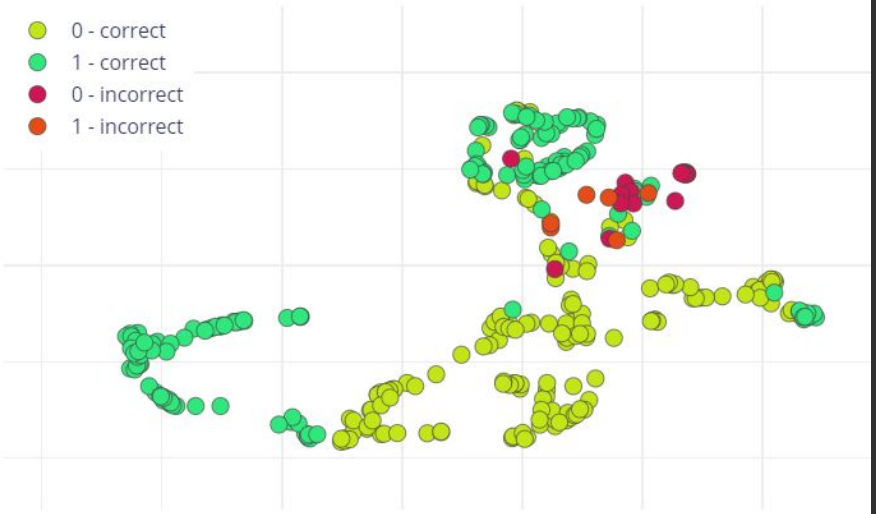
© Medium

Risultati del test del modello

ACCURACY
96.55%

	0	1	UNCERTAIN
0	**94.3%**	3.5%	2.2%
1	1.2%	**98.8%**	0%
F1 SCORE	0.97	0.98	

Feature explorer

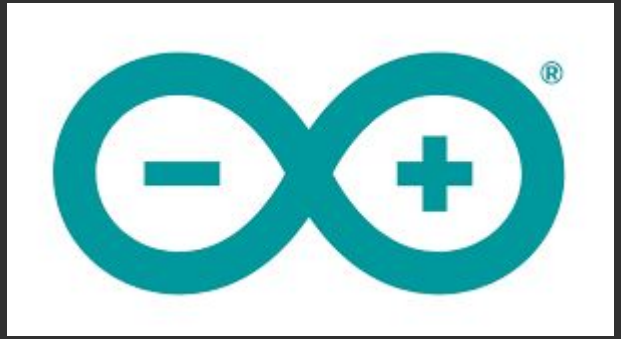


Inference

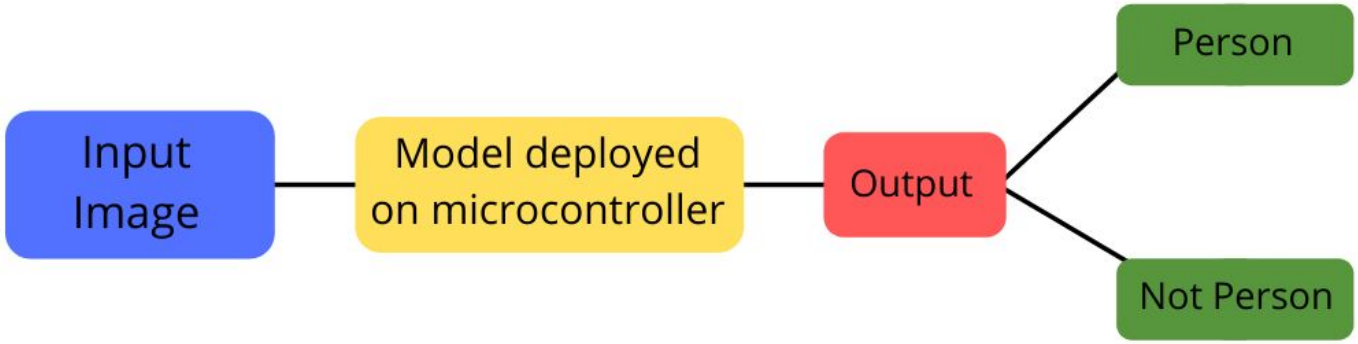
```
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 1 ms., Classification: 66 ms., Anomaly: 0 ms.):
    0: 0.08594
    1: 0.91406
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 1 ms., Classification: 66 ms., Anomaly: 0 ms.):
    0: 0.01953
    1: 0.98047
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 1 ms., Classification: 66 ms., Anomaly: 0 ms.):
    0: 0.82031
    1: 0.17969
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 1 ms., Classification: 66 ms., Anomaly: 0 ms.):
    0: 0.23438
    1: 0.76562
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 1 ms., Classification: 66 ms., Anomaly: 0 ms.):
    0: 0.30078
    1: 0.69922
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 1 ms., Classification: 66 ms., Anomaly: 0 ms.):
    0: 0.07422
    1: 0.92578
```

Sketch di Arduino

basato su Harvard_TinyMLx Arduino Library



Visual Wake Words with TensorFlow Lite Micro



Conversione del modello nel formato cc

```
  0x19, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x19,
  0xc8, 0xff, 0xff, 0xff, 0x09, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x09, 0x0c, 0x00, 0x0c, 0x00, 0x0b, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x04, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x16, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x16, 0xf0, 0xff, 0xff, 0xff, 0x11, 0x00, 0x00, 0x00,
  0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x11, 0x0c, 0x00, 0x10, 0x00,
  0x0f, 0x00, 0x00, 0x00, 0x08, 0x00, 0x04, 0x00, 0x0c, 0x00, 0x00, 0x00,
  0x03, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03
};

const int g_sign_detect_model_data_len = 12360;
```

source	arduino_detection_responder.cpp	set LED behaviour
source	arduino_image_provider.cpp	call to image_provider.h
source	arduino_main.cpp	call to main_functions.h
header	detection_responder.h	set of behaviour when the results of a sign detection run are available
header	image_provider.h	set camera behaviour
header	main_functions.h	loop() and setup() cycle
header	model_settings.h	setting of model (constant)
source	model_settings.cpp	call to model_setting.h, set label of class
header	sign_detect_model_data.h	call to model
source	sign_detect_model_data.cpp	model in cc format, call to sign_detect_model_data.h
header	sign_detection.h	TF Lite Interpreter and its utilities

Licenza