Progettazione e produzione multimediale

Sistema di riconoscimento e raccolta di testo ed emozioni

Leonardo Gori
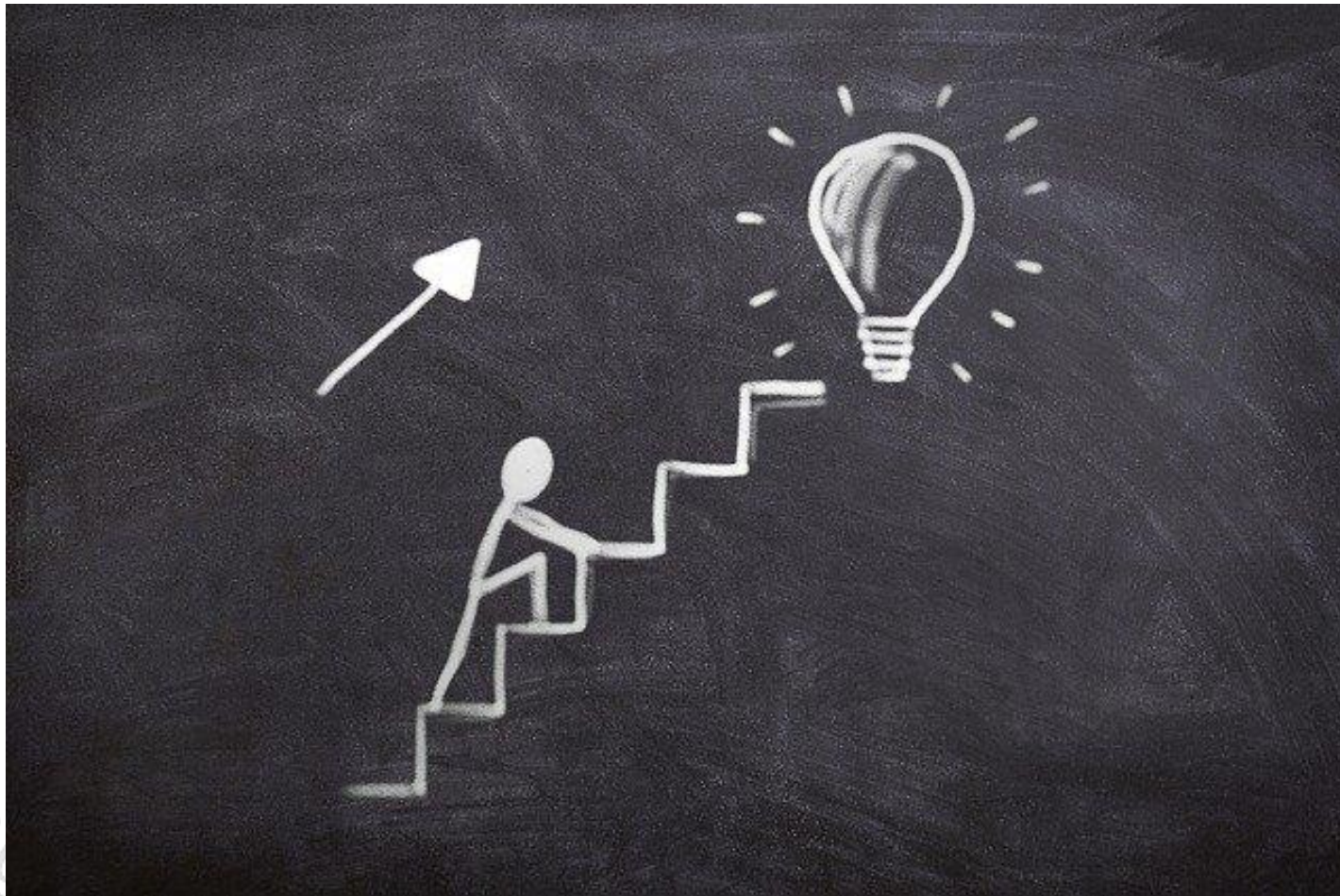**Cosimo Michelagnoli**

L'idea...
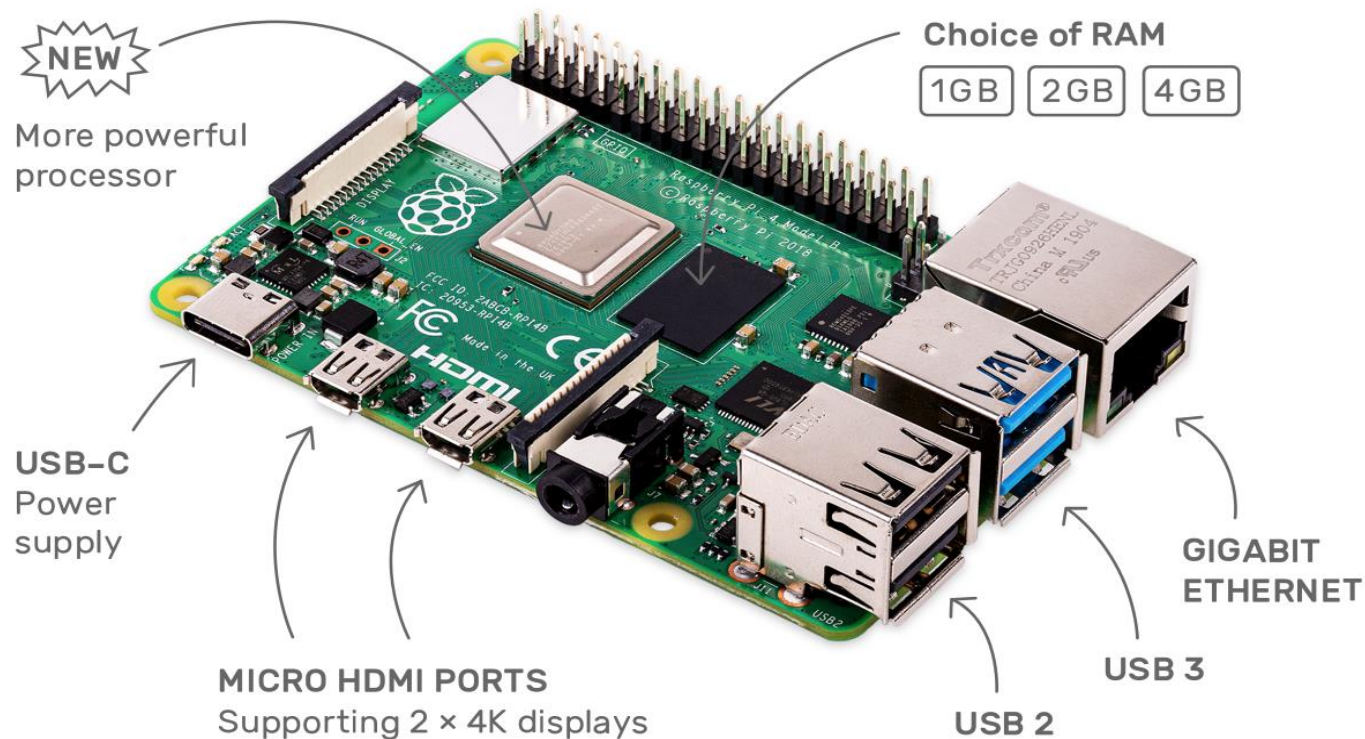
Aspetti operativi del progetto:

- Fattibilità economica del progetto: *(mappatura del flusso del valore)*

- Fattibilità in termini di dimensioni fisiche: *(è utilizzabile in un match?)*

- Fattibilità in termini di competenze progettuali: *(cosa possiamo fare?)*

L'idea per diventare progetto deve essere realizzabile…

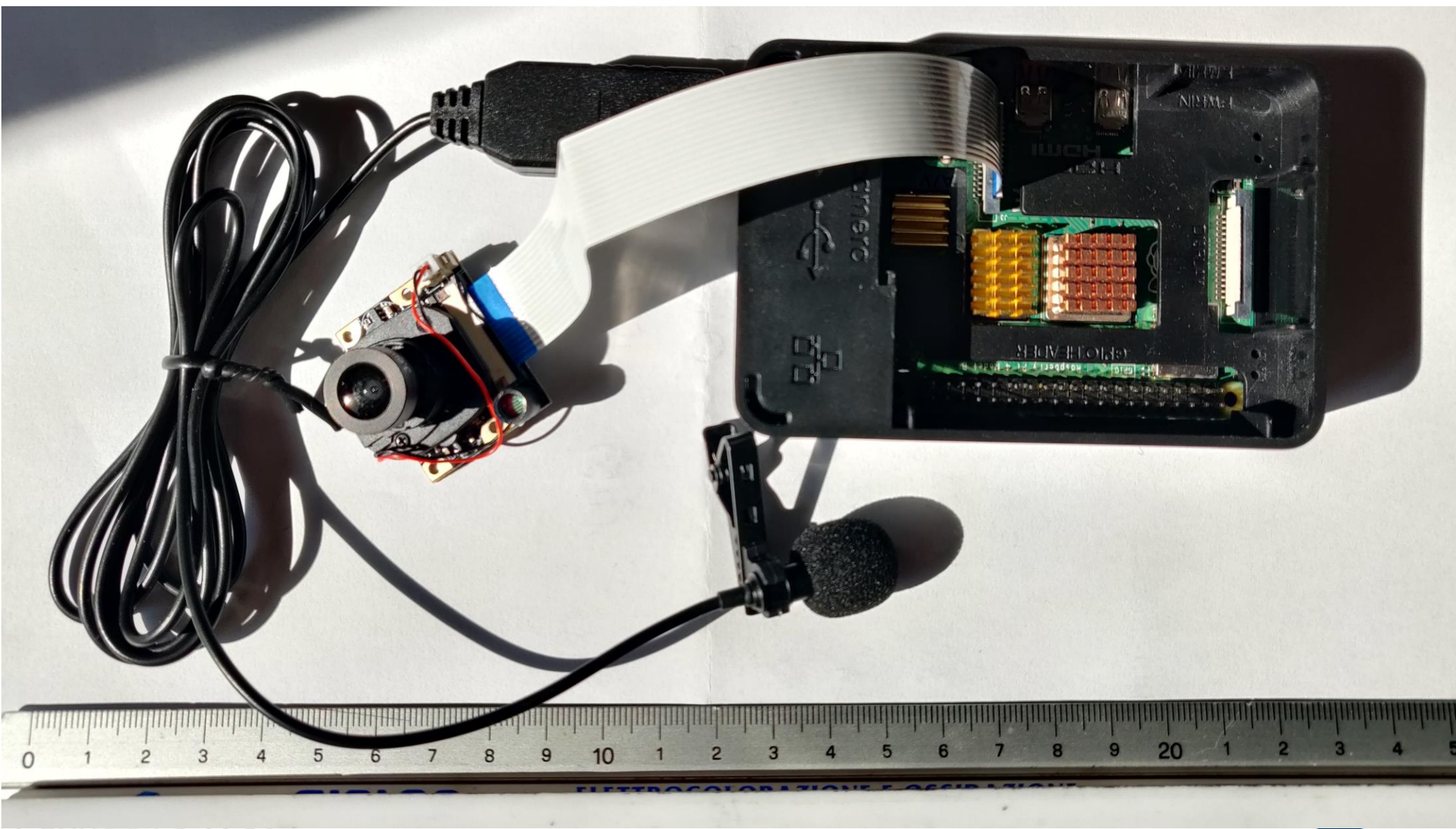# Raspberry pi 4
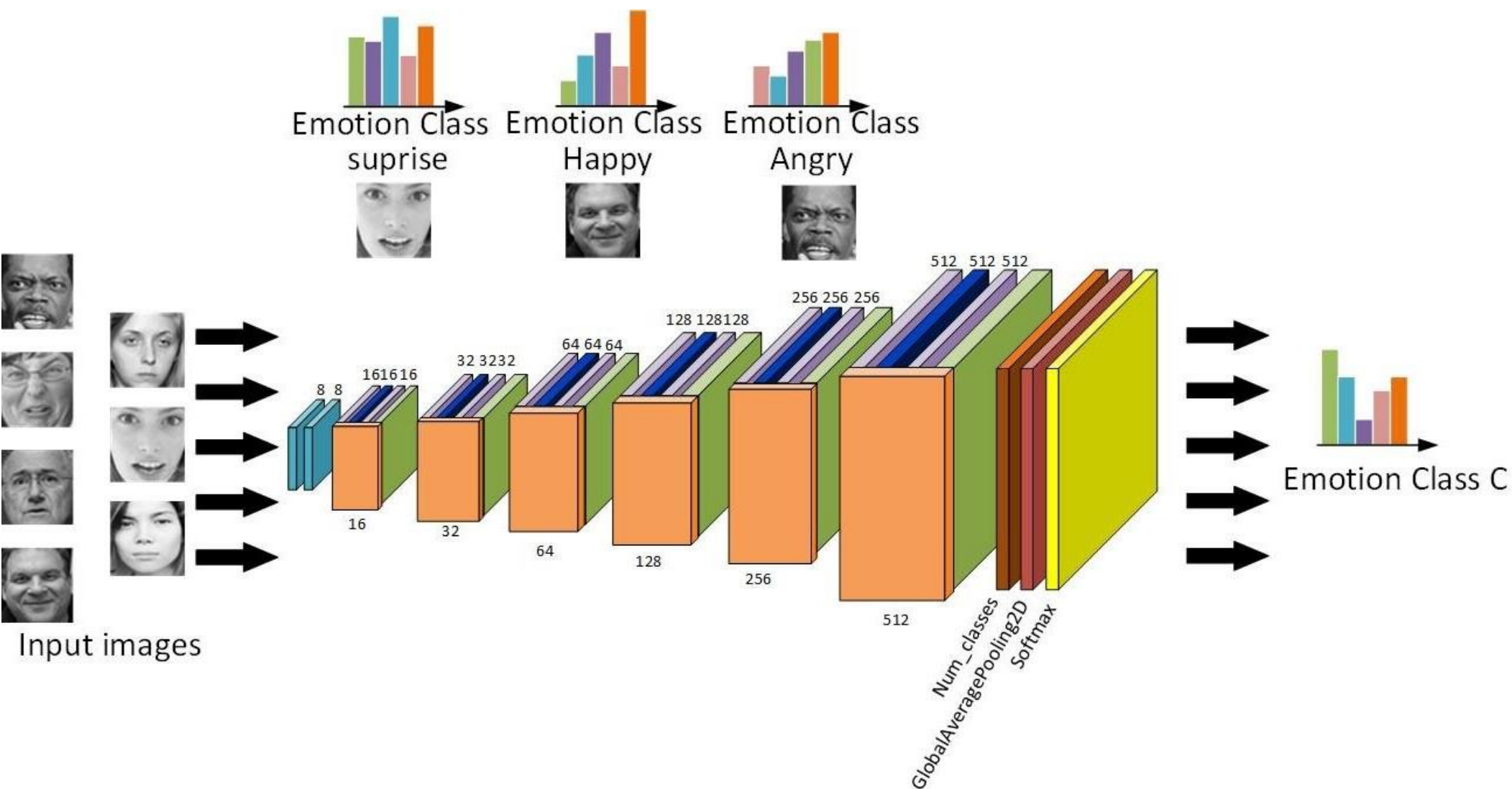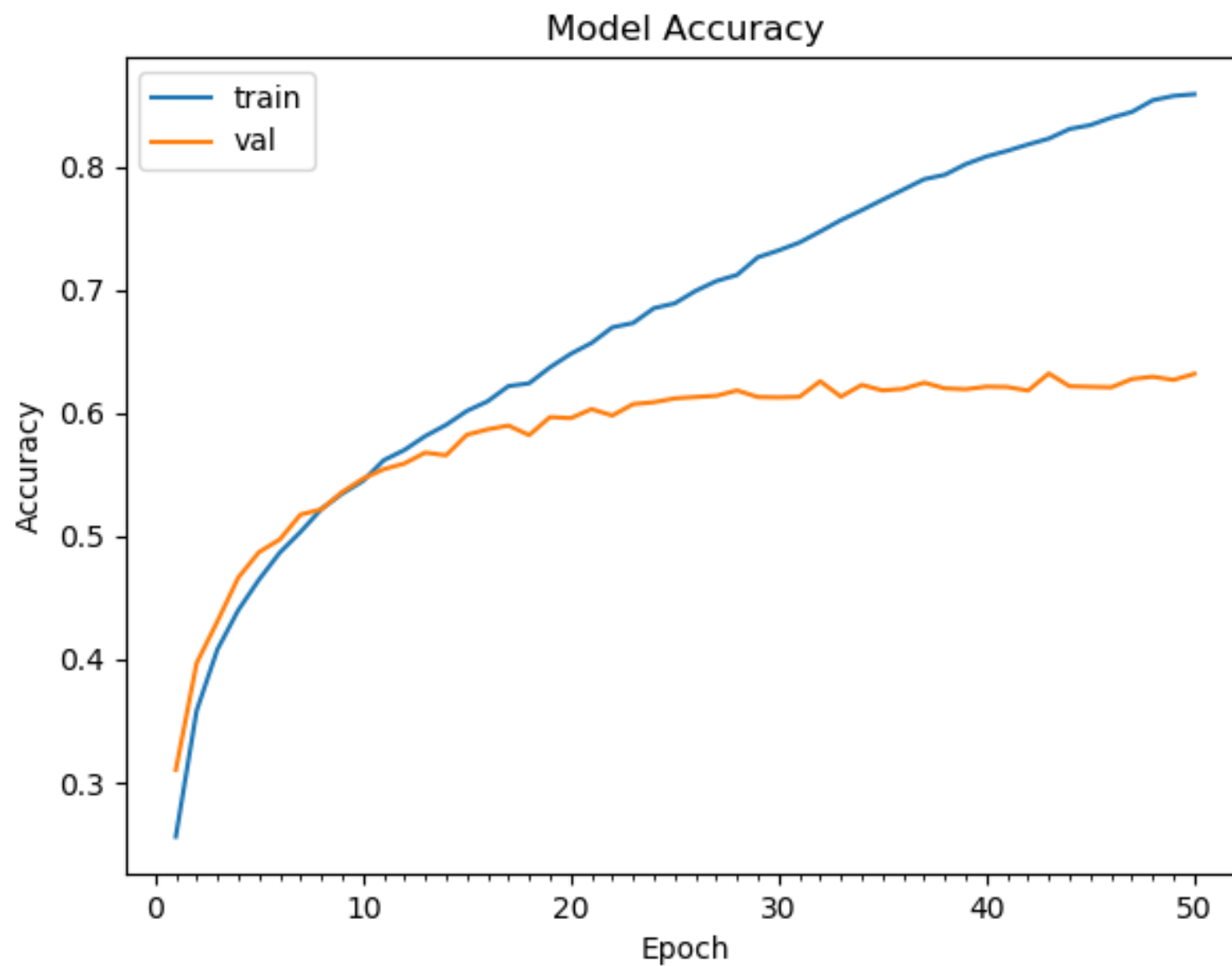
Periferiche I/O

Camera

Microfono

Dimensioni

Software utilizzato:

- Emotion detection using deep learning

- Google Speech Recognition API

Emotion detection

Accuratezza

```python
13   class Emotions(Thread):
14
15       def __init__(self):
16           super().__init__()
17           self.vectEmotion = []
18           os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
19
20           # Create the model
21           self.model = Sequential()
22
23           self.model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48, 48, 1)))
24           self.model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
25           self.model.add(MaxPooling2D(pool_size=(2, 2)))
26           self.model.add(Dropout(0.25))
27
28           self.model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
29           self.model.add(MaxPooling2D(pool_size=(2, 2)))
30           self.model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
31           self.model.add(MaxPooling2D(pool_size=(2, 2)))
32           self.model.add(Dropout(0.25))
33
34           self.model.add(Flatten())
35           self.model.add(Dense(1024, activation='relu'))
36           self.model.add(Dropout(0.5))
37           self.model.add(Dense(7, activation='softmax'))
38
39           # emotions will be displayed on your face from the webcam feed
40
41           self.model.load_weights('model.h5')
42
43           # prevents openCL usage and unnecessary logging messages
44           cv2.ocl.setUseOpenCL(False)
45           # dictionary which assigns each label an emotion (alphabetical order)
46           self.emotion_dict = {0: "Arrabbiato", 1: "Disgustato", 2: "Impaurito", 3: "Felice", 4: "Neutrale", 5: "Triste",
47                               6: "Sorpreso"}
48
49           # start the webcam feed
50           self.cap = cv2.VideoCapture(0)
```

```python
def run(self):

    while True:

        # Find haar cascade to draw bounding box around face
        ret, frame = self.cap.read()
        if not ret:
            break
        facecasc = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        color = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        faces = facecasc.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y - 50), (x + w, y + h + 10), (255, 0, 0), 2)
            roi_gray = gray[y:y + h, x:x + w]
            roi_color = color[y:y + h, x:x + w]
            self.cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray, (48, 48)), -1), 0)
            self.image = np.expand_dims(np.expand_dims(cv2.resize(roi_color, (305, 305)), -1), 0)
            prediction = self.model.predict(self.cropped_img)
            maxindex = int(np.argmax(prediction))
            self.vectEmotion.append(maxindex)
            cv2.putText(frame, self.emotion_dict[maxindex], (x + 20, y - 60), cv2.FONT_HERSHEY_SIMPLEX, 1,
                        (255, 255, 255),
                        2, cv2.LINE_AA)

        cv2.imshow('Video', cv2.resize(frame, (1600, 960), interpolation=cv2.INTER_CUBIC))
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    self.cap.release()
    cv2.destroyAllWindows()

def getEmotion(self):
    return self.emotion_dict[self.most_common(self.vectEmotion)]

def getCropeImage(self):
    return self.image

def setReady(self):
    self.vectEmotion.clear()
```
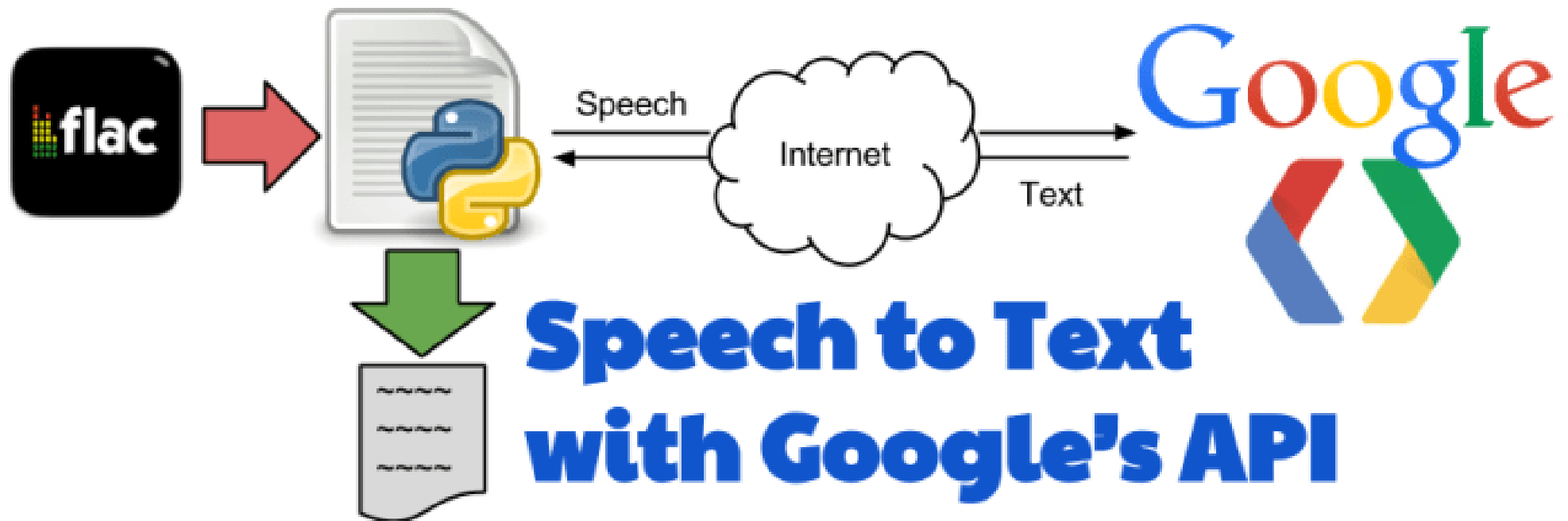
1

2

3

Speech Recognition

```python
17   class SpeechRecognition(Thread):
18       def __init__(self, emotion):
19           super().__init__()
20           self.emotion = emotion
21
22       def run(self):
23           # obtain audio from the microphone
24
25           while True:
26               r = sr.Recognizer()
27               with sr.Microphone() as source:
28                   print("Say something!")
29                   self.emotion.setReady()
30                   audio = r.listen(source)
31
32               try:
33                   # for testing purposes, we're just using the default API key
34                   # to use another API key, use `r.recognize_google(audio, key="GOOGLE_SPEECH_RECOGNITION_API_KEY")`
35                   # instead of `r.recognize_google(audio)`
36                   text = r.recognize_google(audio, language="it-IT")
37                   IMIR = self.emotion.getCropeImage().reshape(305, 305, 3)
38                   img = Image.fromarray(IMIR)
39                   img.save('prova.png')
40                   with open("prova.png", "rb") as file:
41                       img = base64.b64encode(file.read())
42                   # img = Image.open(io.BytesIO(base64.b64decode(img)))
43                   print("Google Speech Recognition thinks you said: " + text
44                       + " while your mood was " + self.emotion.getEmotion())
45                   str_img = str(img)
46                   str_img = str_img[2:len(str_img) - 1]
47                   today = date.today()
48                   my_date = today.strftime("%Y-%m-%d")
49                   t = Table(date=my_date, speech_text=text, emotion=self.emotion.getEmotion(), image=str_img)
50                   t.save()
51
52               except sr.UnknownValueError:
53                   print("Google Speech Recognition could not understand audio")
54               except sr.RequestError as e:
55                   print("Could not request results from Google Speech Recognition service; {0}".format(e))
```
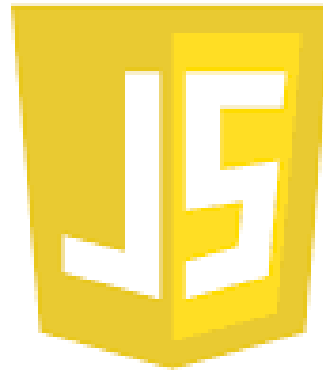
14

Django
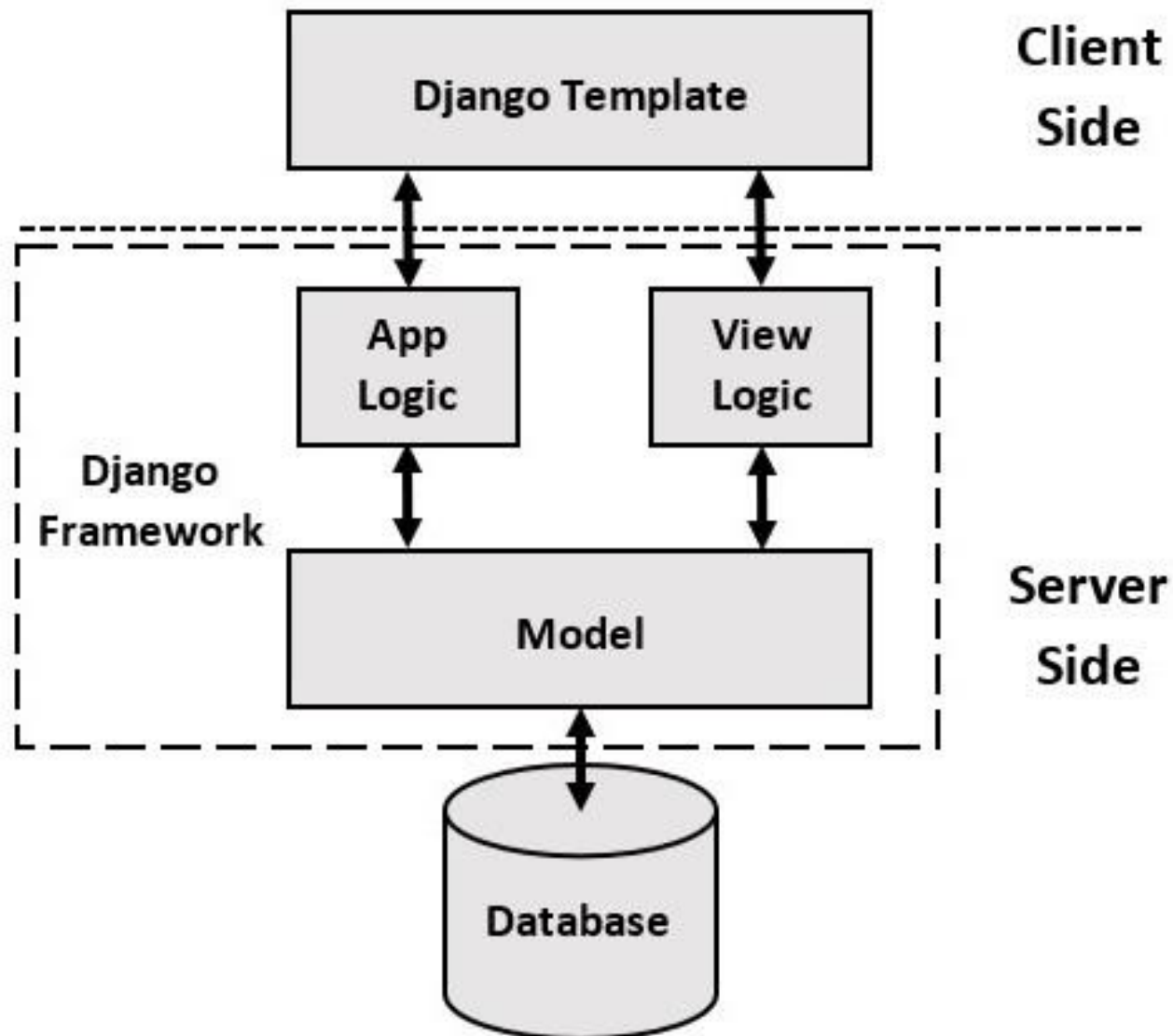
Interfaccia grafica

Bootstrap

Architettura progetto

Raspberry Pi Raspberry Pi Camera Module
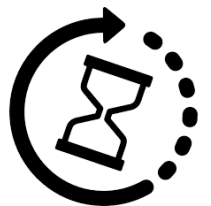

Microfono USB

Update Data

Database

Query data

Internet Enabled Mobile

Through the site it is possible to query the database

- Fattibilità economica del progetto

- Fattibilità in termini di dimensioni fisiche

- Fattibilità in termini di competenze progettuali

Sitografia:

Link Riconoscimento delle emozioni:

https://fablab.ruc.dk/facial-expression-recognition-on-a-raspberry-pi/

https://github.com/atulapra/Emotion-detection

Link Riconoscimento del testo:

https://github.com/Uberi/speech_recognition