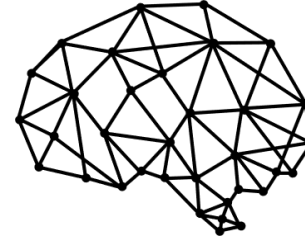


# Il Progetto

**Realizzare un software atto a riconoscere le espressioni facciali e identificarne l'emozione correlata attraverso un algoritmo di machine learning. Il risultato finale dovrà essere sufficientemente accurato da essere utilizzabile nei vari campi che lo richiedono (videogiochi, giustizia o assistenza ai non vedenti).**

**Il software dovrà essere testato all'interno di un virtual environment isolato dal sistema operativo su HW locale.**





CHIARA FRANCO - DOMENICO POLICASTRO

# Emotive

Emotion and identity recognition software



IIS Blaise Pascal 2020-21

# Il Team



**Chiara Franco**

Emotion Recognition

Face Recognition

Flask



**Domenico Policastro**

Emotion Recognition

Face Recognition

Web app Style



## SEZIONE 1:

# Studio di Fattibilità



### Obiettivo

Realizzazione di un software che riconosca le emozioni umane attraverso l'analisi del volto



### Idea progettuale

Software realizzato in Python, integrato in un framework Flask per creare un applicazione web



### Destinatari del progetto

Il progetto è rivolto ad aziende che si occupano dello sviluppo di videogiochi o di sicurezza





## SECTION 1:

# Abstract

### What is **Emotive**?

Emotive is an emotion and identity recognizing software. It is written in Python and integrated in a web app using the flask template.

### How does it **Work**?

It exploits a Tensorflow neural network, trained on our data set in order to recognize users' emotions.

It also uses the face\_recognition library (a pre-trained algorithm) to recognize the user's identity.

### Final **Result**

We embedded the two algorithms in a web app available online. Everyone can register to our website and freely use our services: emotion and identity recognition from both video and photo.

## SEZIONE 2:

# Project Charter



Analisi dei Rischi



Deliverables Principali



Milestones



Risorse Principali



Tempistica

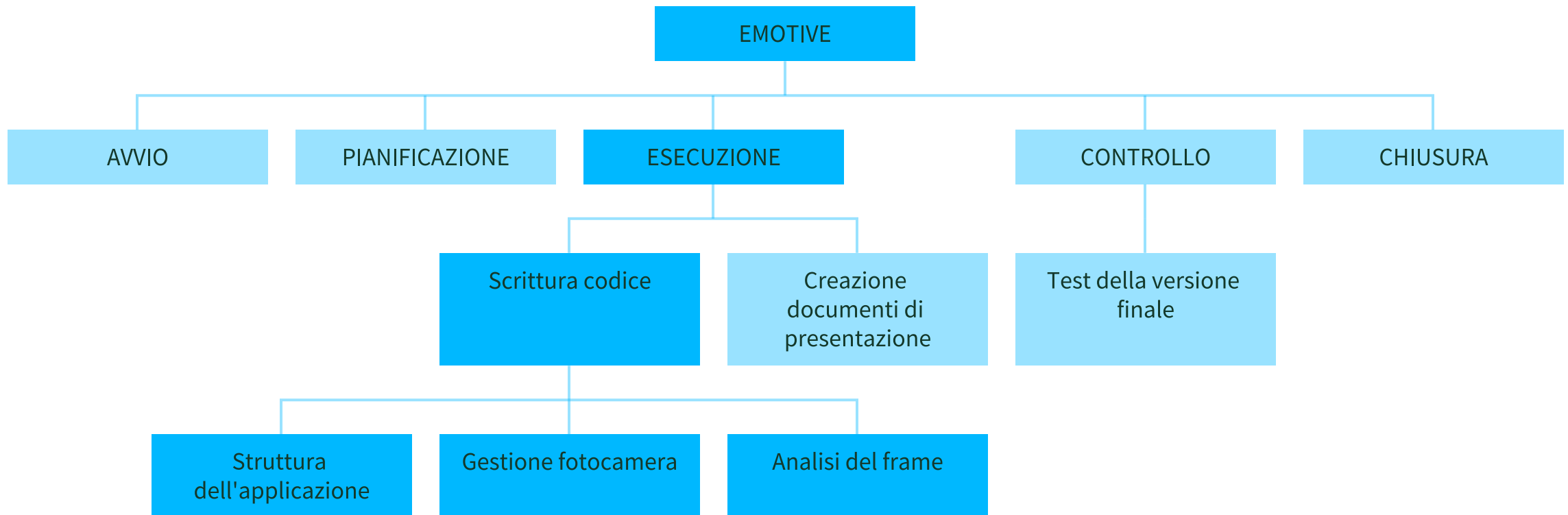


Autorizzazione



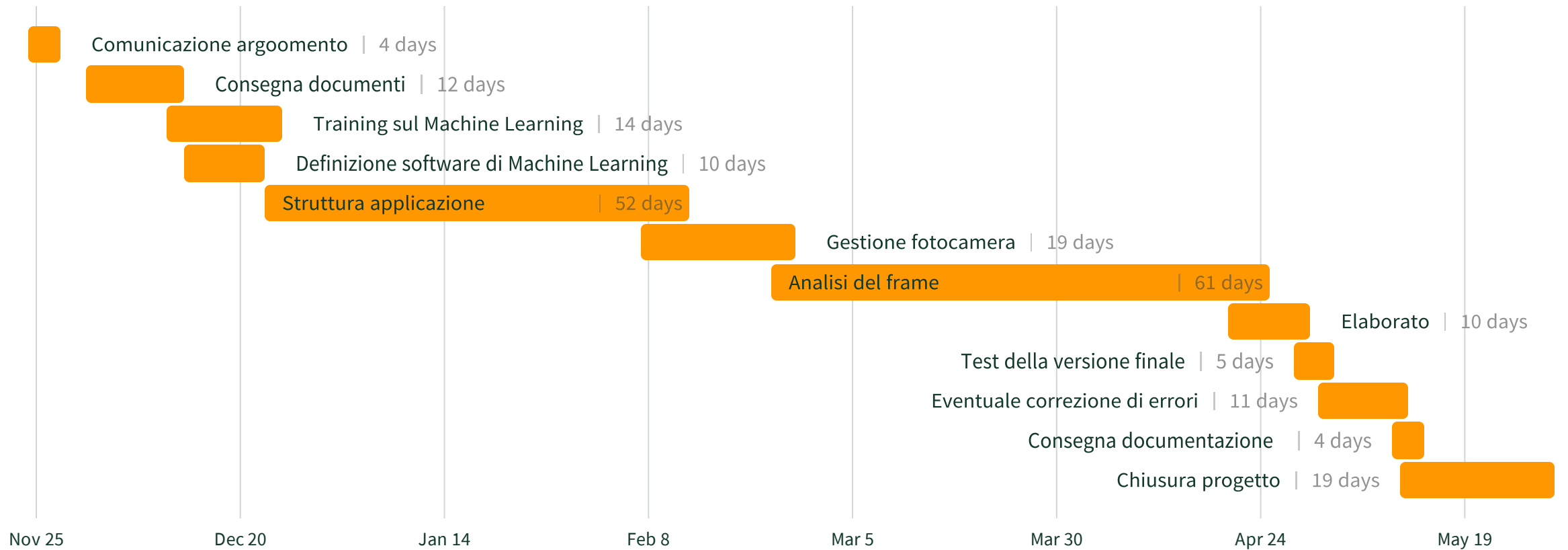
## SEZIONE 3:

# WBS



## SEZIONE 4:

# Gantt

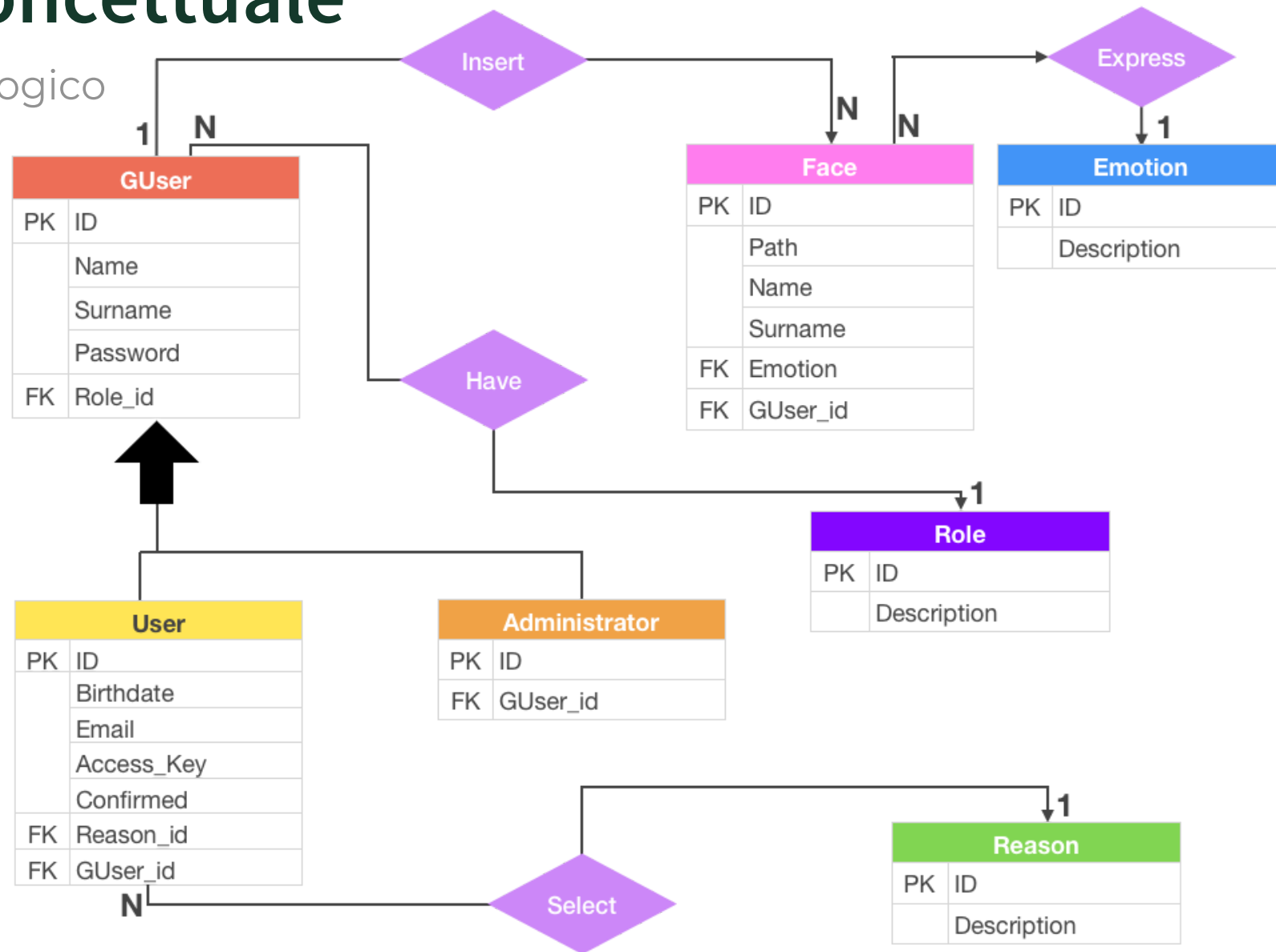




## SEZIONE 5:

# Modello Concettuale

Include modello Logico



## SEZIONE 6:

# Codice Significativo

**Tensorflow:** addestramento dell'algoritmo di riconoscimento delle emozioni

```
In [20]: new_model = keras.Model(inputs = base_input, outputs = final_output)
```

```
In [21]: new_model.compile(loss="sparse_categorical_crossentropy", optimizer = "adam", metrics = ["accuracy"])
```

```
In [ ]:
```

```
In [23]: new_model.fit(X,Y,epochs=15)
```

Epoch 1/15

67/898 [=>.....] - ETA: 1:12:08 - loss: 1.7954 - accuracy: 0.3050

## SEZIONE 6:

# Codice Significativo

**image\_classifier**: utilizzo dell'algoritmo addestrato per riconoscere l'emozione espressa dal viso che appare nel frame

```
@app.route('/image_classifier')
def image_classifier(frame):
    ... gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    ... faces = faceCascade.detectMultiScale(gray, 1.1, 4)
    ... Predictions = []
    ... face_roi = []
    ... x, y, w, h = 0, 0, 0, 0
    ... for x, y, w, h in faces:
    ...     ... roi_gray = gray[y:y+h, x:x+w]
    ...     ... roi_color = frame[y:y+h, x:x+w]
    ...     ... #cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
    ...     ... facess = faceCascade.detectMultiScale(roi_gray, scaleFactor=1.1, minNeighbors=4)
    ...     ... if len(facess) == 0:
    ...     ...     print("Face not detected")
    ...     ... else:
    ...     ...     for (ex, ey, ew, eh) in facess:
    ...     ...     ... face_roi = roi_color[ey:ey+eh, ex:ex+ew]
    ...
    ... if len(face_roi) > 0:
    ...     ... final_image = cv2.resize(face_roi, (224, 224))
    ...     ... final_image = np.expand_dims(final_image, axis=0)
    ...     ... final_image = final_image/255.0
    ...     ... Predictions = new_model.predict(final_image)
    ... else:
    ...     ... Predictions.insert(0, -1)
```

## SEZIONE 6:

# Codice Significativo

**recognize**: funzione della web app che effettua il riconoscimento facciale

```
def recognize(frame): # identifica la persona presente nel frame passato come parametro
    ... global target_name
    ... global target_encoding
    ... global count
    ... global label

    ... small_frame = cv2.resize(frame, None, fx=0.20, fy=0.20)
    ... rgb_small_frame = cv2.cvtColor(small_frame, 4)

    ... face_location = face_recognition.face_locations(rgb_small_frame)
    ... frame_encodings = face_recognition.face_encodings(rgb_small_frame)

    ... if frame_encodings:
    ...     ... frame_face_encoding = frame_encodings[0]
    ...     ... match = face_recognition.compare_faces([target_encoding], frame_face_encoding)[0]
    ...     ... label = target_name if match else "Unknown"
```

## SEZIONE 6:

# Codice Significativo

**Flask:** inserimento dei frame e del risultato dell'algoritmo all'interno della pagina web

```
<div class="mt-6 bg-gray-400 antialiased text-gray-900">
...<div>
...  <div>
...    {% if close %}
...    
...    {% else %}
...    
...    {% endif %}
...  </div>
...  <div class="relative px-4">
...    <div class="bg-white p-6 rounded-lg shadow-lg">
...    <h4 class="text-blue-600 mt-1 text-xl font-semibold uppercase leading-tight truncate">Emotive says that you are</h4>
...
...    <div class="mt-1">
...    {{ session['var'][4] }}
...    </div>
...  </div>
...</div>
...</div>
</div>
```

SEZIONE 7:

# Conclusioni

- Competenze acquisite
- Obiettivi raggiunti
- Difficoltà incontrate
- Sviluppi futuri



CONCLUSIONI:

# Competenze Informatiche



Python



Jupyter



Tensorflow



TailwindCSS

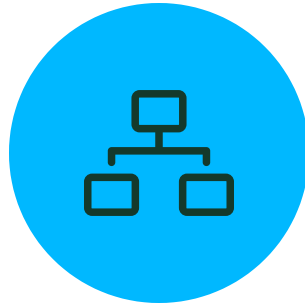


CONCLUSIONI:

# Competenze Trasversali



Approcciarsi a strumenti  
sconosciuti



Organizzazione di Progetto



TeamWork





## CONCLUSIONI:

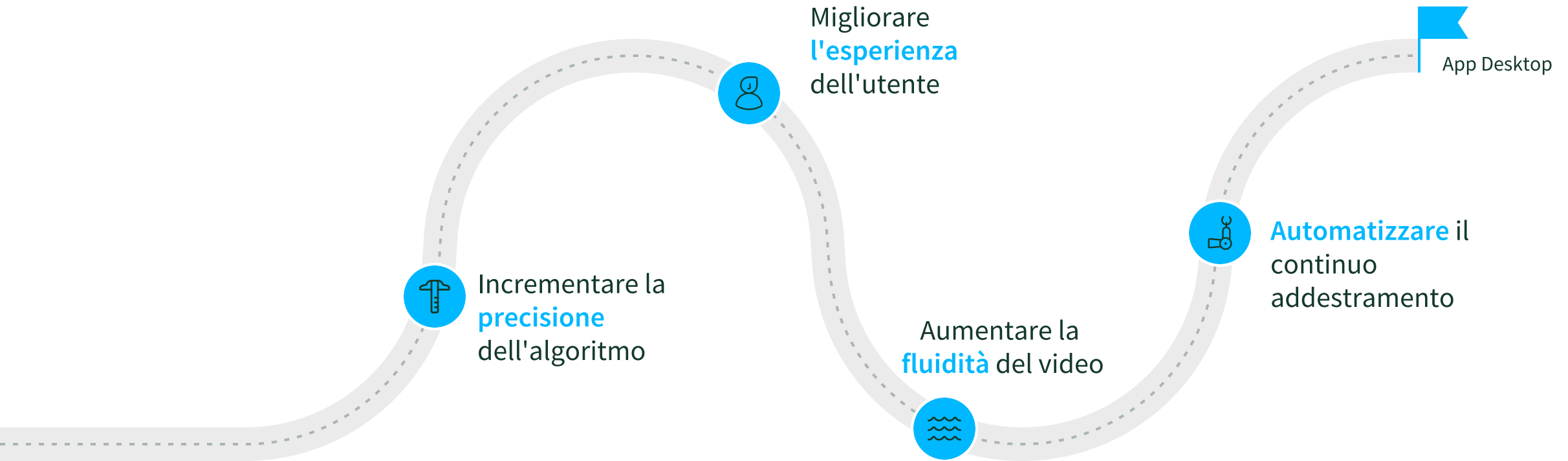
# Difficoltà Incontrate

- 1 Trovare l'algoritmo più adatto
- 2 Addestramento corretto dell'algoritmo
- 3 Installazione librerie di Python (Tensorflow)
- 4 Utilizzo di strumenti sconosciuti



CONCLUSIONI:

# Sviluppi Futuri



## CONCLUSIONI:

# Obiettivi raggiunti

- Algoritmo di riconoscimento facciale
- Algoritmo di riconoscimento delle emozioni
- Web app



# Sitografia

- **Riconoscimento Emozioni:** DeepLearning\_by\_PhDScholar,  
<https://www.youtube.com/watch?v=avv9GQ3b6Qg&t=3878s>
- **Riconoscimento Facciale:** PyMike,  
<https://www.youtube.com/watch?v=5moAA31iTmM>
- **Documentazione Flask:**  
<https://flask.palletsprojects.com/en/2.0.x/>
- **Documentazione Face Recognition:**  
<https://pypi.org/project/face-recognition/>

