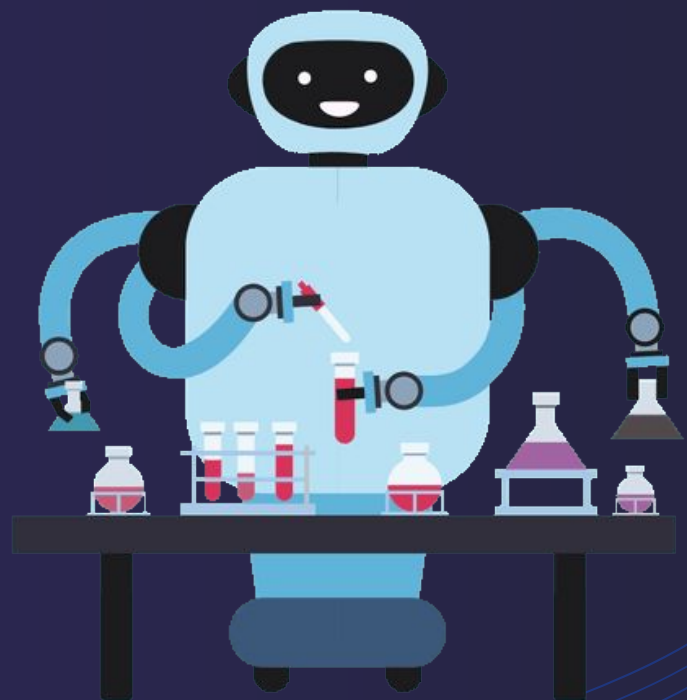


# Travaux Pratiques.

1. Pré-Traitements
2. Reconnaissance Faciale



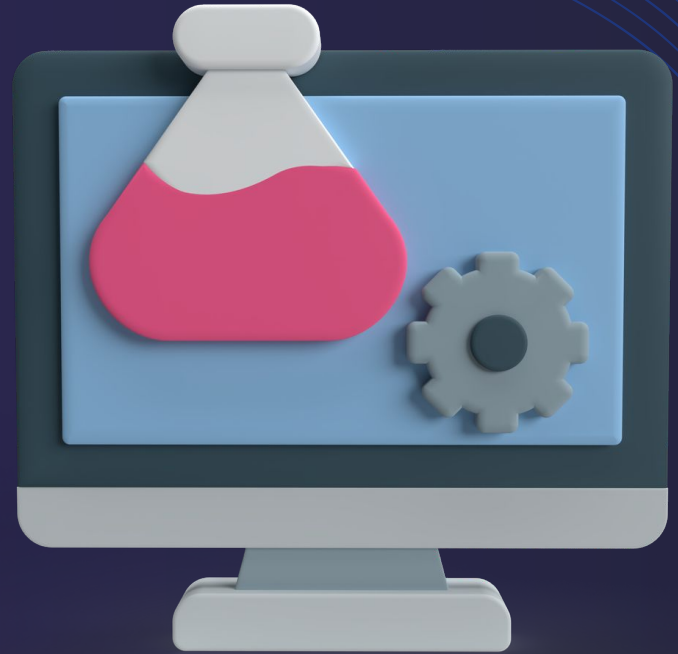
# Travaux Pratiques

## Partie 1

- Conversion en Niveaux de Gris
- Redimensionnement
- Normalisation
- Filtrage et Réduction du Bruit

## Partie 2

- Reconnaissance Faciale
- Passage en application Web (lightsail)
- Passage en API (lambda)



# Lecture d'une image.

```
import cv2

# Charger une image
image = cv2.imread('image.jpg')

# Afficher l'image
cv2.imshow('Image', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

# Conversion en Niveaux de Gris

```
# Convertir en niveaux de gris
gray_image = cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)

# Sauvegarder ou afficher
cv2.imshow('Grayscale Image', gray_image)
cv2.imwrite('gray_image.jpg', gray_image)
```

# Redimensionnement

```
# Redimensionner à une taille fixe
resized_image = cv2.resize(image, (200, 200))

# Redimensionner en maintenant le ratio
height, width = image.shape[:2]
new_width = 300
new_height = int((new_width / width) * height)
resized_aspect_image = cv2.resize(image,
                                   (new_width, new_height))

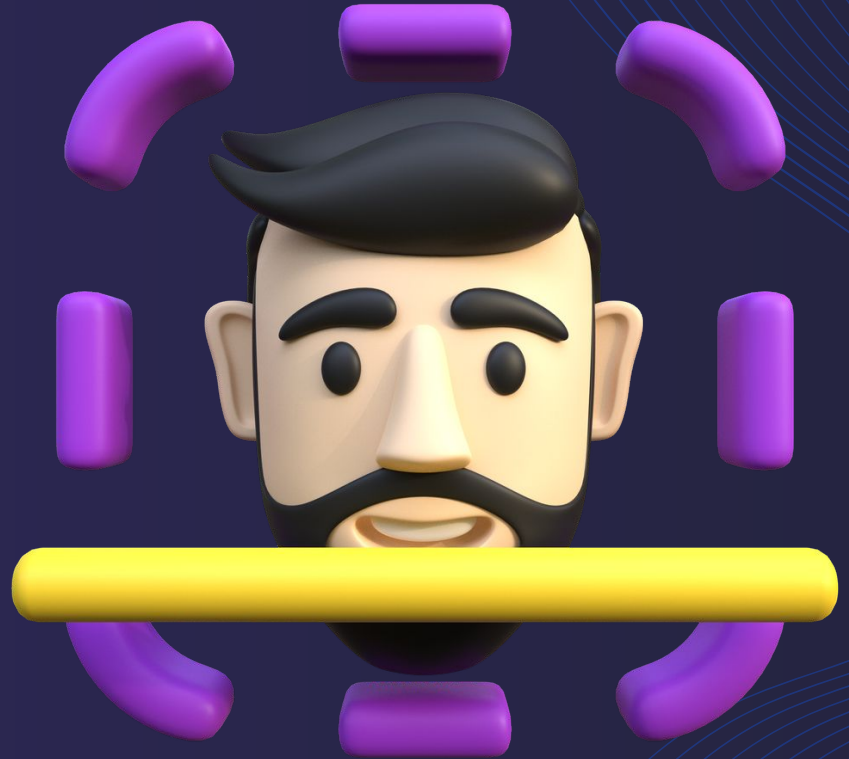
cv2.imshow('Resized Image', resized_image)
```

# Normalisation

```
# Normalisation entre 0 et 1
normalized_image = image / 255.0

# Normalisation à une plage personnalisée
norm_image = cv2.normalize(image, None,
alpha=0, beta=255, norm_type=cv2.NORM_MINMAX)
```

# Réconnaissance Faciale.



# Etapes.

- Création d'un fichier app
- Récupération des fichiers :  
`requirements.txt`
- Lancement de l'app





Fichier "app.py"

```
import face_recognition

known_image = face_recognition.load_image_file('file1.jpg')
unknown_image = face_recognition.load_image_file('file2.jpg')

biden_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]

results = face_recognition.compare_faces([biden_encoding],
unknown_encoding)

print(str(results))
```