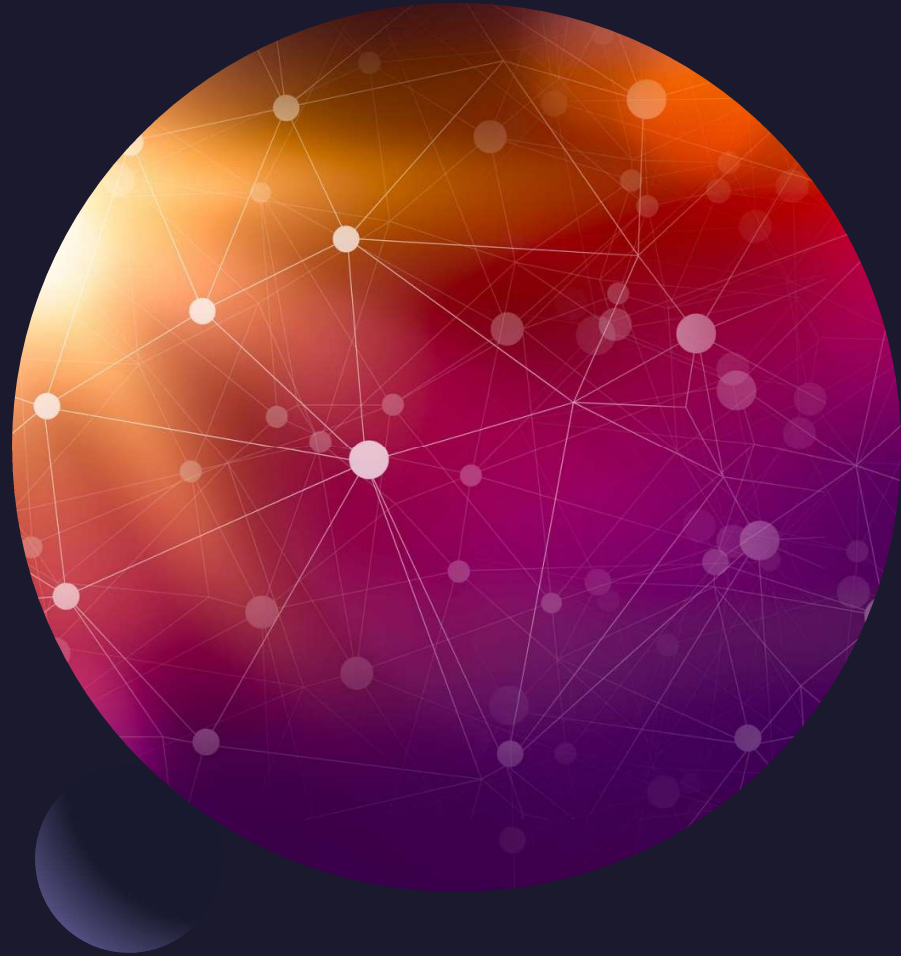


Detección de sonrisa

Proyecto final



Integrantes:

07140147 – José Hernández

21001119 – Velvet González



Resumen

- Implementación del algoritmo visto en clase con las mejoras de detección de sonrisa y ojos agregando al algoritmo el training “haarcascade_smile.xml”.

```
11 face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
12 eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')
13 smile_cascade = cv2.CascadeClassifier('haarcascade_smile.xml')
14
15 def detect(gray, frame):
16     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
17     for (x, y, w, h) in faces:
18         cv2.rectangle(frame, (x, y), ((x + w), (y + h)), (255, 0, 0), 2)
19         roi_gray = gray[y:y + h, x:x + w]
20         roi_color = frame[y:y + h, x:x + w]
21         eyes = eye_cascade.detectMultiScale(roi_gray, 1.3, 5)
22         smiles = smile_cascade.detectMultiScale(roi_color, 1.8, 20)
23
24         for (sx, sy, sw, sh) in smiles:
25             cv2.rectangle(roi_color, (sx, sy), ((sx + sw), (sy + sh)), (0, 0, 255), 2)
26         for (ex, ey, ew, eh) in eyes:
27             cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0, 0, 255), 3)
28     return frame
29
30 video_capture = cv2.VideoCapture(0)
31 while True:
32
33     _, frame = video_capture.read()
34     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
35
36     canvas = detect(gray, frame)
37
38     cv2.imshow('Video', canvas)
39
40
```

Demostración

El programa tiene la capacidad de detectar rostros, sonrisas y ojos.

