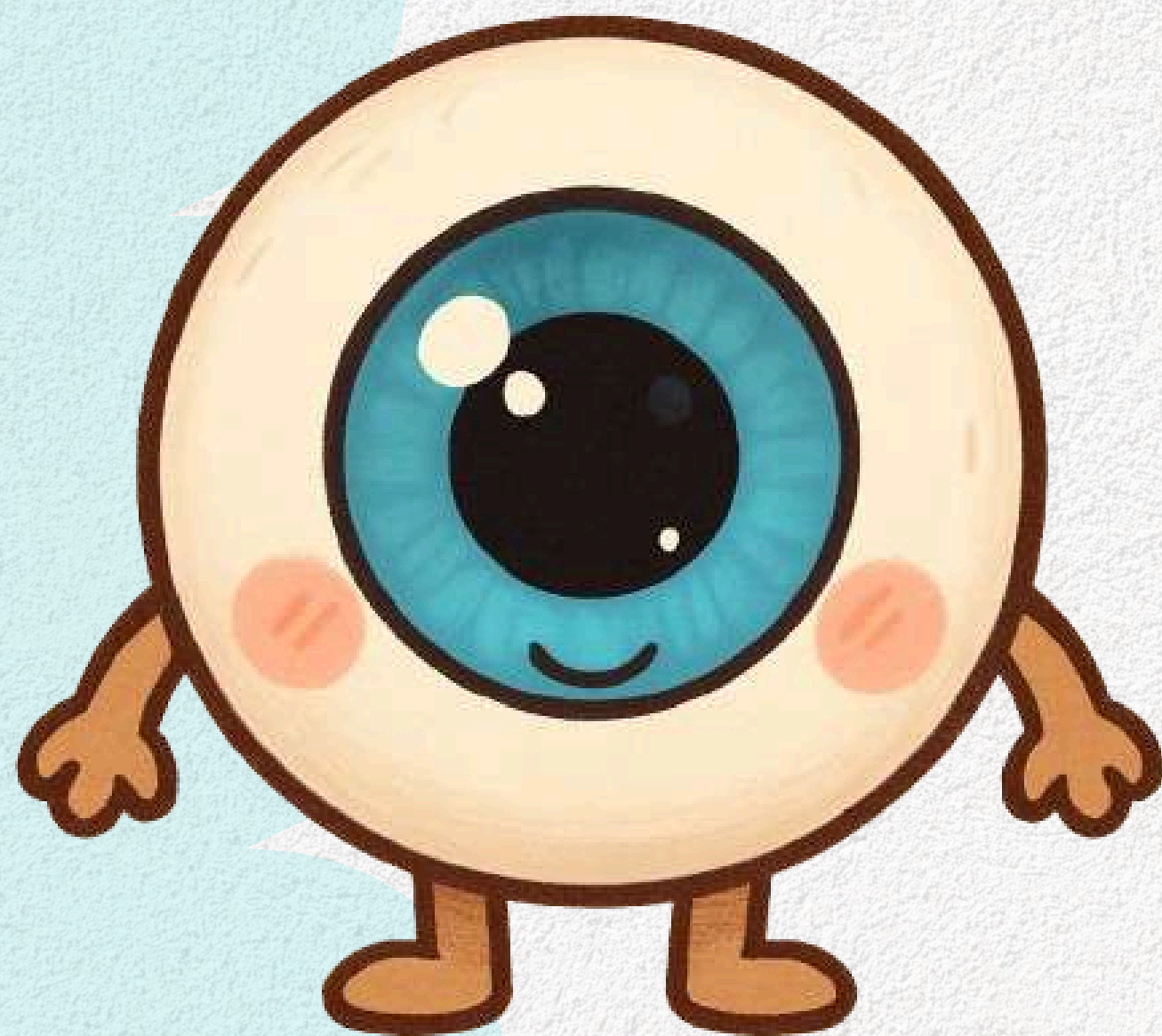
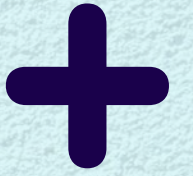


EQUIPO 3

AVANCE PROYECTO CLASIFEYE



Gabriela Guzmán
Juan Esteban Cardenas
Juan David Ardila
Manuel Eduardo Díaz
Gabriela Gallegos



RECAPITULACIÓN

Anteriormente se realizó el recorte y redimensionamiento de las imágenes con el objetivo de normalizar las dimensiones en ellas para su procesamiento

SIN EMBARGO...

- Calidad de las imágenes al momento del redimensionamiento (píxeles).
- Filtración de imágenes que en el recorte pueden haber perdido la relevancia (rostros).



CAMBIOS



- Error inicial con face-torch mtcnn; cambio a mtcnn independiente para detección facial.
- Recorte y alineación facial implementados con MTCNN y Dlib.

AVANCES

Implementación Correcta de Librería

- Problema: Fallo con face-torch mtcnn (integrada en PyTorch).
- Solución: Migración a mtcnn independiente.
- Impacto: Detección facial más estable.
- Código:

```
[ ] from mtcnn import MTCNN  
    detector_mtcnn = MTCNN()
```



AVANCES

Reducción de Ruido (Non-Local Means):

- Filtro para luminancia (mayor = más suavizado)
- Filtro para los canales de color

Corrección de Iluminación (CLAHE):

- Convertir a espacio de color LAB (Luminancia, A, B)
- Crear objeto CLAHE (limita el contraste para evitar artefactos)
- Aplicar CLAHE solo al canal de luminancia (L)
- Recomponer la imagen LAB con la luminancia corregida
- Convertir de nuevo a BGR



AVANCES

Detección de Rostro y Landmarks (MTCNN):

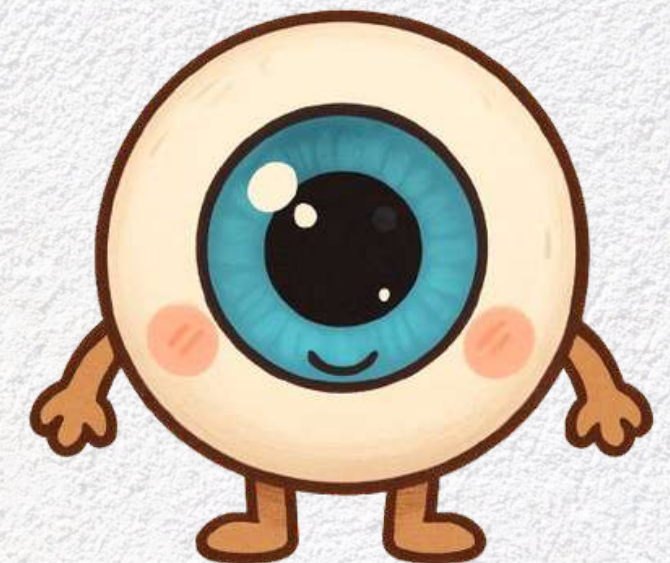
```
# Hacemos la detección
resultado = detector_mtcnn.detect_faces(imagen_corregida)
# Sacamos los resultados del recorte y recortamos
x, y, w, h = resultado[0]['box']
```



Alineación Facial + Resize a 224x224 (Dlib)



- Calcular ángulo
- Calcular centro entre los ojos
- Distancia entre ojos en la imagen original
- Distancia deseada entre ojos
- Obtener matriz de rotación + escala
- Mover los ojos a la posición deseada



Lider de la próxima semana: Manuel Eduardo Díaz