

Periodo	Enero-abril 2020
Cuatrimestre y grupo	8 A
Asignatura	Inteligencia artificial
Corte	3
Actividad	IA.C3.A1 key points
Fecha de asignación	12/03/2020
Fecha de entrega	12/03/2020
Matriculas	Nombre
153224	Alexis Pérez Gómez

Actividad: Encontrar el porcentaje de las correspondencias de determinadas imágenes rotadas escaladas y desplazadas.

Características de cada imagen:

Imagen rotada: como máximo 360 grados.

Imagen desplazada: debe ser en los cuatro ejes y diagonales

Imagen escalada: las imágenes deben reducir ala (1/2,1/4,1/8,1/16)

El algoritmo tiene las siguientes funciones:

Números	FUNCIONES EN EL ARCHIVO keyPoints.py	Nombre descriptivo
1	def get_image()	Obtener imagen
2	def rotation():	Rotación
3	def input_dialog(text)	Dialogo de entrada
4	def window_two()	Ventana 2
5	def input_data()	Dato de entrada
6	def resized()	Escalado
7	def method_harris_rotation()	Harris rotación
9	method_harris_displaced()	Harris desplazado
10	method_harris_escalade_up()	Harris escalado
11	def graphics	Graficas
12	Def paint_matches()	Dibujar correspondencias

Descripción de las funciones:

- 1. Obtener imagen:** este método sirve para seleccionar la imagen desde la computadora.
- 2. Rotación:** rota la imagen a través de un método llamado del archivo tranformaciones.py, con un determinado grado proporcionado por el usuario, siendo los datos de entrada menor a 360 grados, las rotaciones son hasta que el dato de entrada sea menor a o igual 360 generando n imágenes.

3. **dialogo de entrada:** ejecuta un alerta donde se ingresara el valor de la rotación inicial.
4. **Ventana dos:** genera una venta para ingresar los valores de (x, y) para el desplazamiento de la imagen.
5. **Datos de entrada:** recibe los valores de (x, y) y los transforma a enteros para mandarlos como para metro al método que desplazara la imagen.
6. **Escalado:** escala una imagen al (1/2), (1/4), (1/8), (1/16) a través de un método llamado desde el archivo transformaciones.py.
7. **Harris rotación:** recibe un arreglo de imágenes y la imagen original, posteriormente se convierten a escala de grises para poder aplicar el método Harris que nos devolverá un data set, que servirá para encontrar los keypoints para posteriormente transformarlos y encontrar la correspondencias entre las imágenes resultantes y la original para posteriormente obtener los porcentaje que pasaran graficarse.
8. **Harris desplazado:** este método hace lo mismo del punto 7 con una sola diferencia es que recibe las coordenadas de desplazamiento para hacer las combinaciones.
9. **Harris escalado:** realiza las mismas tareas que el punto 7 y 8 pero con un cambio que solo recibe las imágenes escaladas.
10. **Gráficas:** grafica el porcentaje de correspondencias de la imagen original y las resultantes.
11. **Dibujar correspondencias:** se encarga de unir las coordenadas que tuvieron correspondencias.

Números	FUNCIONES EN EL ARCHIVO transformatios.py	Nombre descriptivo
1	def rotate_image(image, angle)	Rotar imagen
2	def shifting(image_1,x,y)	Desplazar imagen
3	def displacement_image_save()	Guardar desplazado
4	def resized_image(image_1):	Escalar imagen
5	escaled_up_image_save()	Guardar escalado

1. **Rotar imagen:** rota la imagen con respecto al ángulo además de modificar el tamaño esto para no perder los valores de las esquinas.
2. **Desplazar imagen:** se encarga de desplazar las imágenes en los cuatro ejes y diagonales con respecto a la (x, y) proporcionado.
3. **Guardar desplazado:** guarda la imagen desplaza en una ruta determinada.
4. **Escalar imagen:** este método se encarga de escalar la imagen proporcionada.
5. **Guardar escalado:** guarda las imágenes escaldas en una ruta determinada

Números	FUNCIONES EN EL ARCHIVO harris.py	Nombre descriptivo
1	<code>def image_gray(image_rgb)</code>	Convertir imagen
2	<code>def dataset(image_gray)</code>	DataSet de harris
3	<code>def get_points(imagen,dst)</code>	Obtener keypoints
4	<code>def draw_points(imagen,coords)</code>	Dibujar puntos
5	<code>def rotated_coords</code>	Transformar coordenadas
6	<code>def convert_to_coordinates(kp1)</code>	Convertir a cordenadas
7	<code>def rotation_matches()</code>	Matches rotacion
8	<code>def kp_transform_deplaced_image()</code>	Tranformar desplazado
9	<code>Def displaced_matches()</code>	Matches desplazado

1. **Convertir imagen:** convierte una imagen RGB a escala de grises.
2. **Data Set:** se le aplica Harris detector de esquinas a la imagen escala de grises que nos devolverá un Data Set.
3. **Obtener key points:** se le asigna un valor a un threshold de 0.01 buscamos el máximo de valor del data set y lo multiplicamos por 0.01 se recorre el data set y lo valores que sean mayor al threshold multiplicado ahí obtenemos un key point y lo guardamos junto a sus coordenadas.
4. **Dibujar puntos:** se pintan las esquinas detectadas en las imágenes.
5. **Transformar coordenadas:** se transforman las coordenadas (keypoints) de la imagen.
6. **Convertir a coordenadas:** se conviertes a coordenadas los key points.
7. **Matches rotación:** se buscan las correspondencias entre key points para poder saber que coordenadas hacen match y poder unirlos.
8. **Transformar desplazado:** se transforman los keypoints del la imagen.
9. **Matches desplazado :** se buscan las correspondencias entre key points para poder saber que coordenadas hacen match y poder unirlos

Lenguaje utilizado y librerías:

En la parte de programación se utiliza el lenguaje de programación Python.

NumPy: agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.

Matplotlib: es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.

OpenCV: es una biblioteca que se centra principalmente hacia procesamiento de imagen.

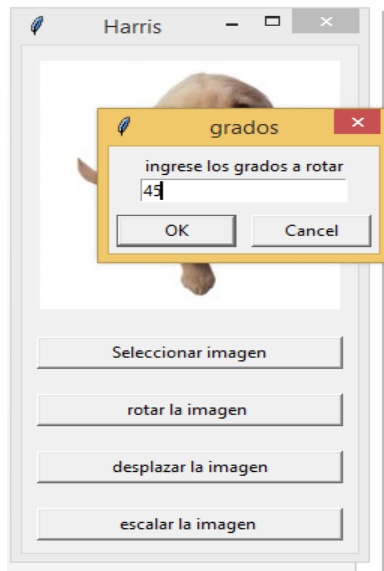
Tkinter : es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python. Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python.

RESULTADOS

Interfaz gráfica de imagen ya seleccionada.



Se elige rotación ingresamos 45 grados.



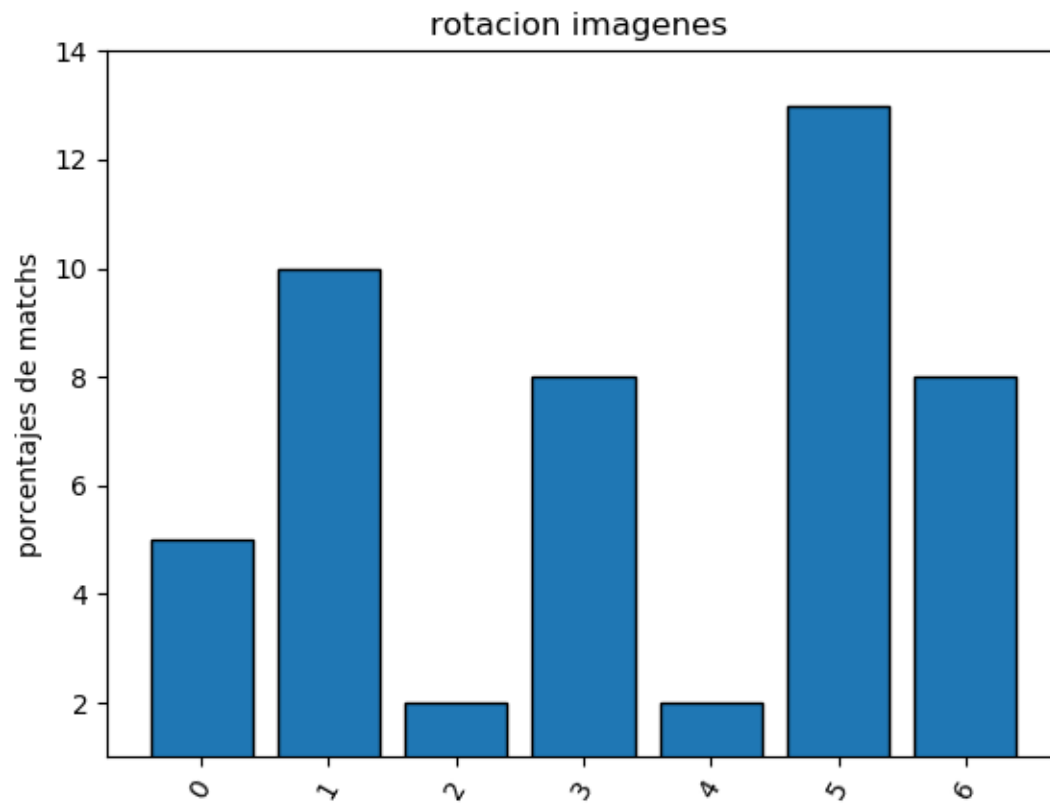
Resultados imágenes rotadas.



Imagen1 (45°), imagen2 (90°), imagen3 (135°), imagen4 (180°)

Imagen5 (225°), imagen6 (270°), imagen7 (315°)

Grafica de correspondencias



Barra 0 a (45°), barra 1 a (90°), barra 2 a (135°), barra 3 a (180°), barra 4 a (225°), barra 5 (270°), barra 6 a (315°)

Correspondências

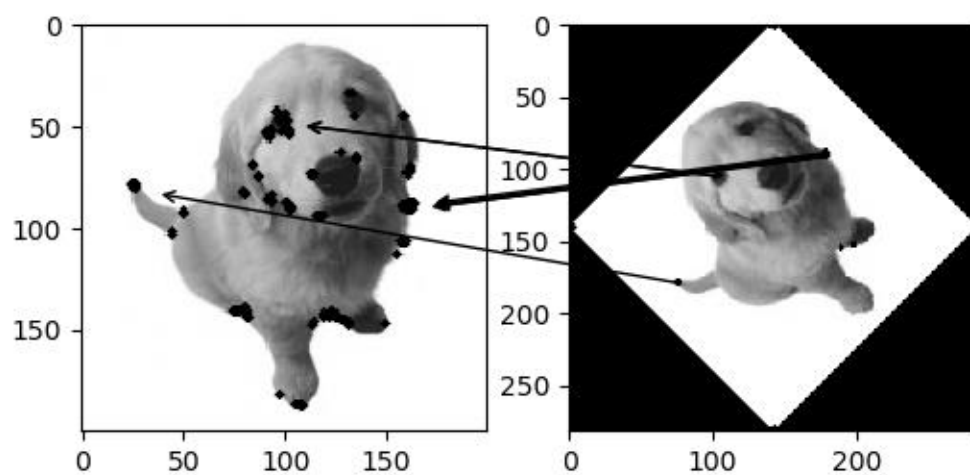


Imagem original e Imagem a 45°

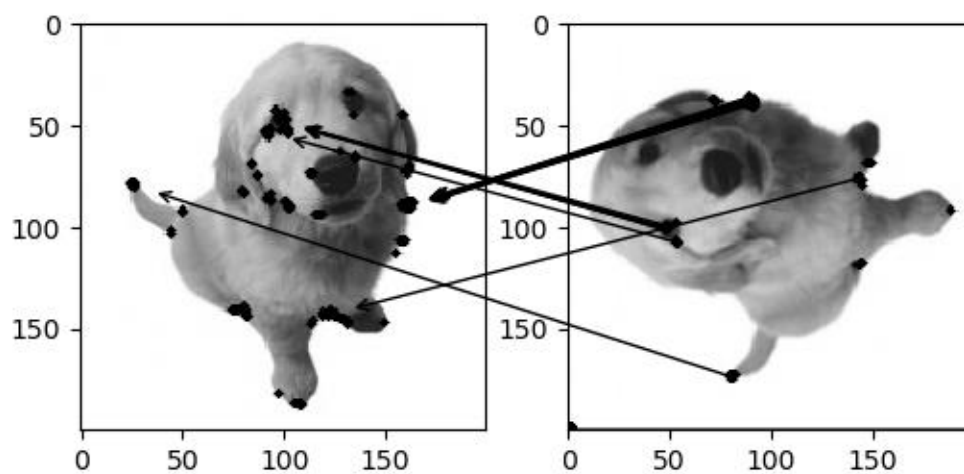


Imagem original e Imagem a 90°

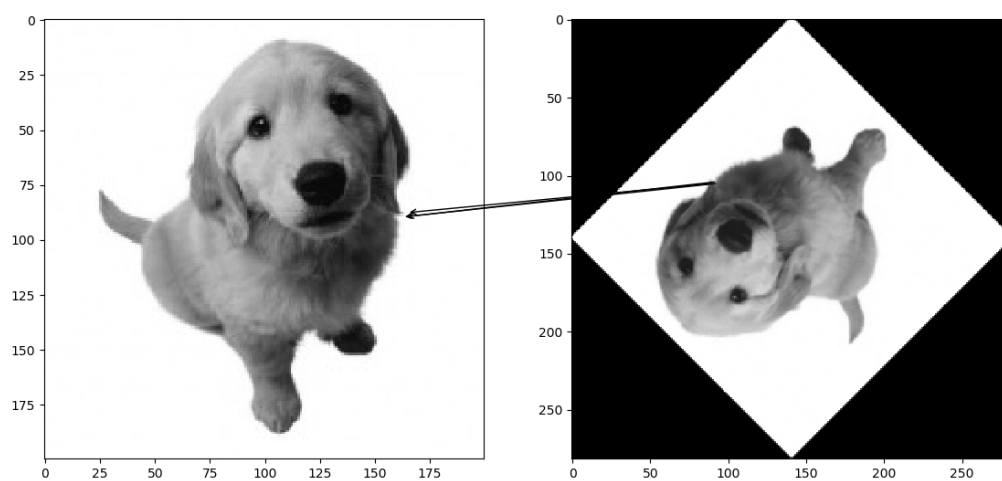


Imagen original e Imagen a 135°

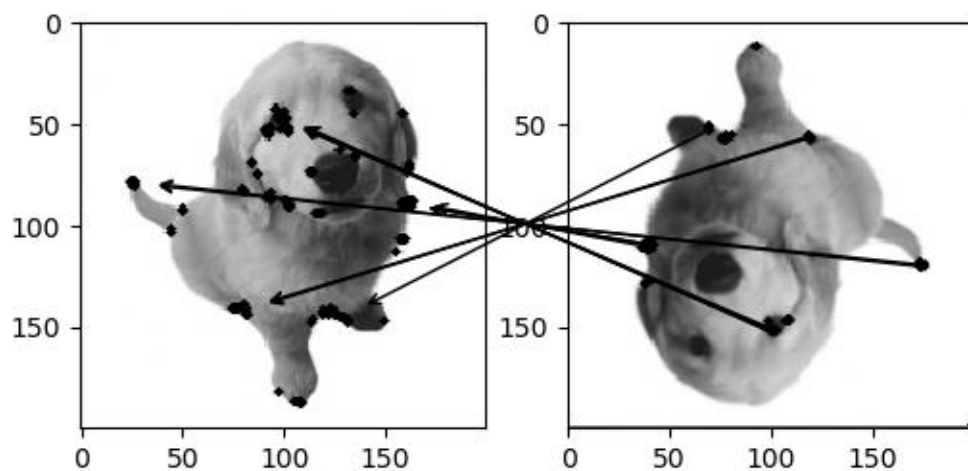


Imagen original e Imagen a 180°

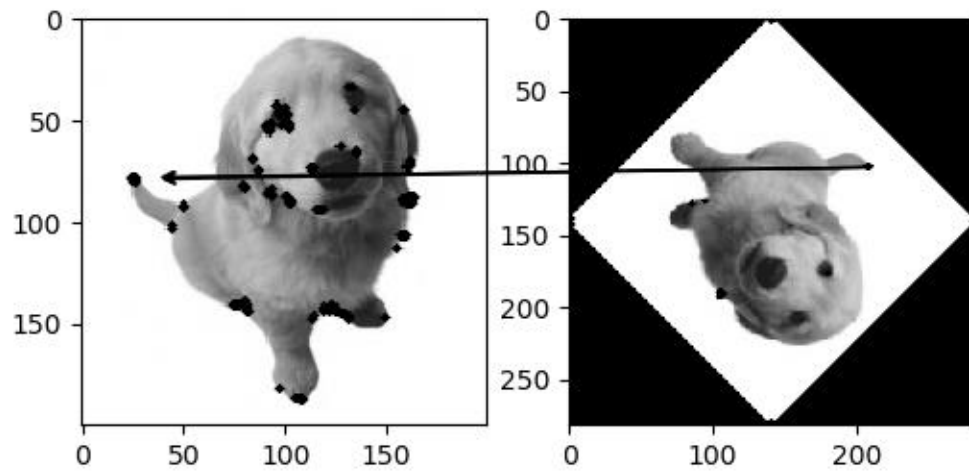


Imagen original e Imagen 225°

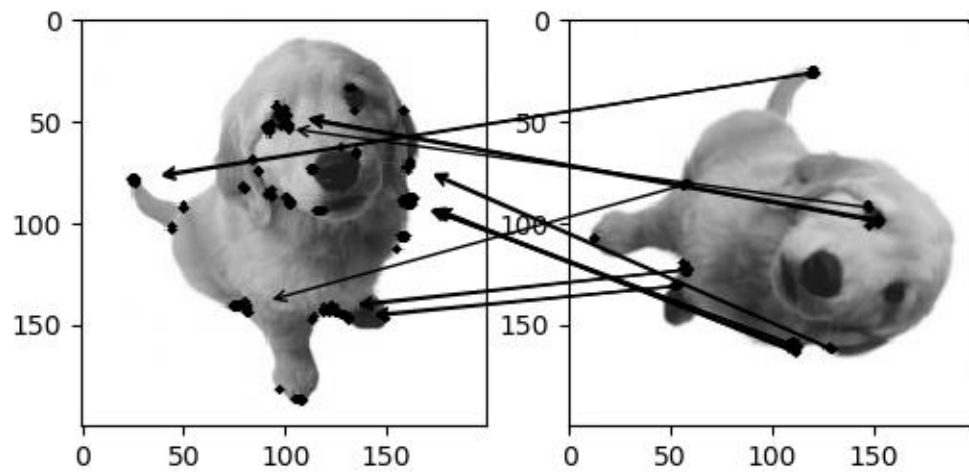


Imagen original e Imagen a 270°

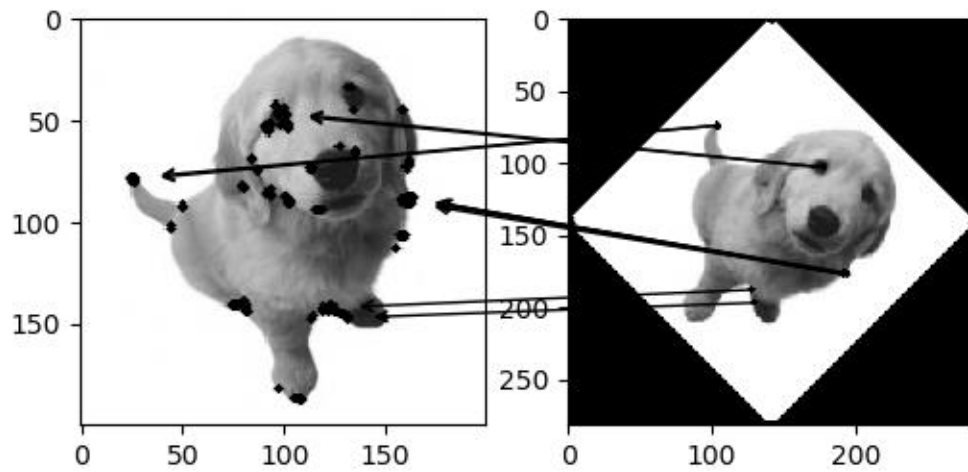
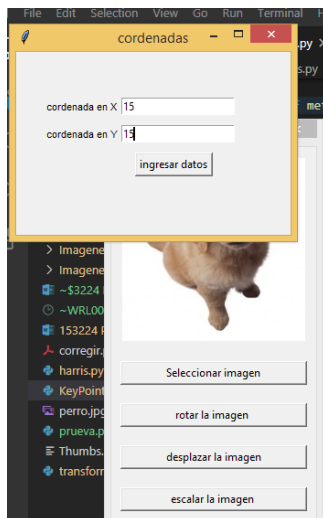


Imagen original e Imagen a 315°

Imágenes desplazadas se ingresa las coordenadas en x y.



Resultados desplazamiento $x=15$ $y=15$ sus combinaciones en los ejes y diagonales.

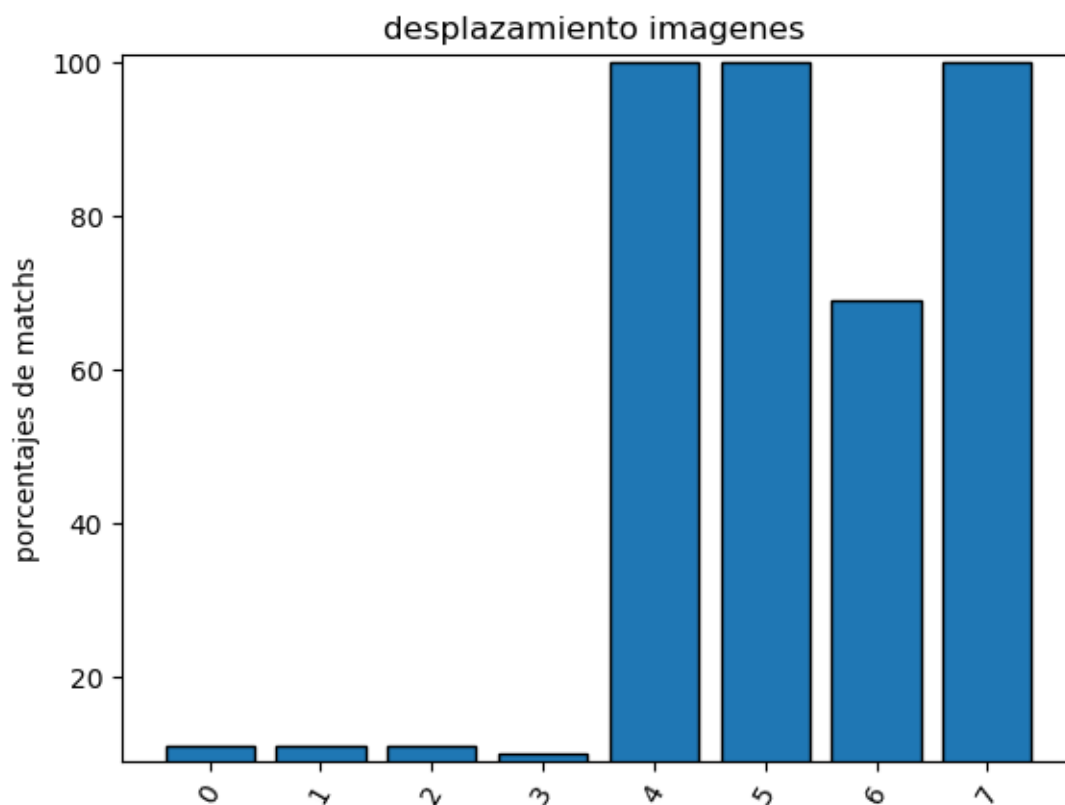


Imagen0(x=15, y=15), imagen1(x=-15, y=-15), imagen2(x=15, y=-15)

Imagen3(x=-15, y=15), imagen4(x=15, y=0), imagen5(x=-15, y=0)

Imagen6(x=0, y=15), imagen7(x=0, y=-15).

Graficas de correspondencias



Barra 0(x=15, y=15), barra 1(x=-15, y=-15), barra 2(x=15, y=-15)

Barra 3(x=-15, y=15), barra 4(x=15, y=0), barra 5(x=-15, y=0)

Barra 6(x=0, y=15), barra 7(x=0, y=-15)

Correspondencias

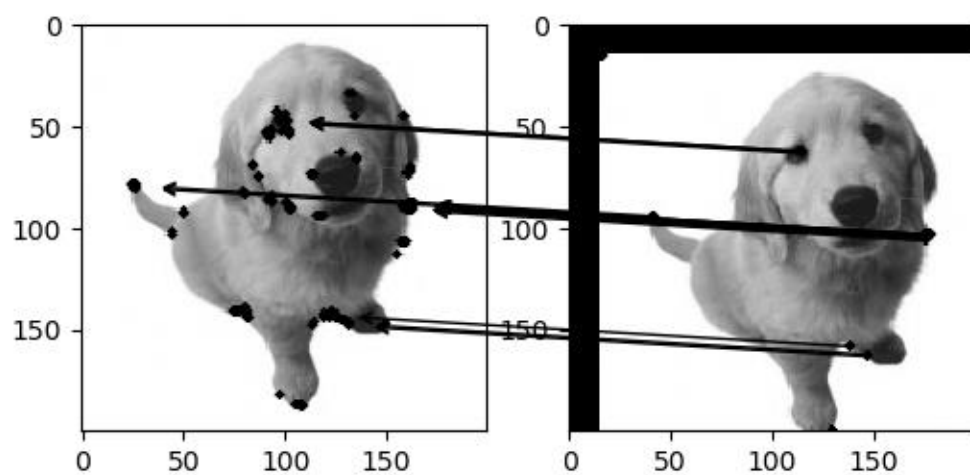


Imagen original e Imagen0 ($x=15$, $y=15$)

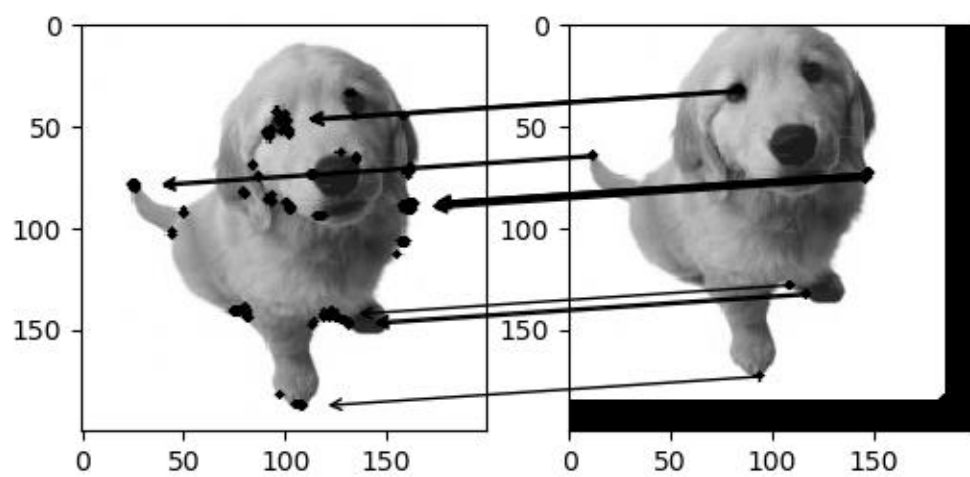


Imagen original e imagen1 ($x=-15$, $y=-15$)

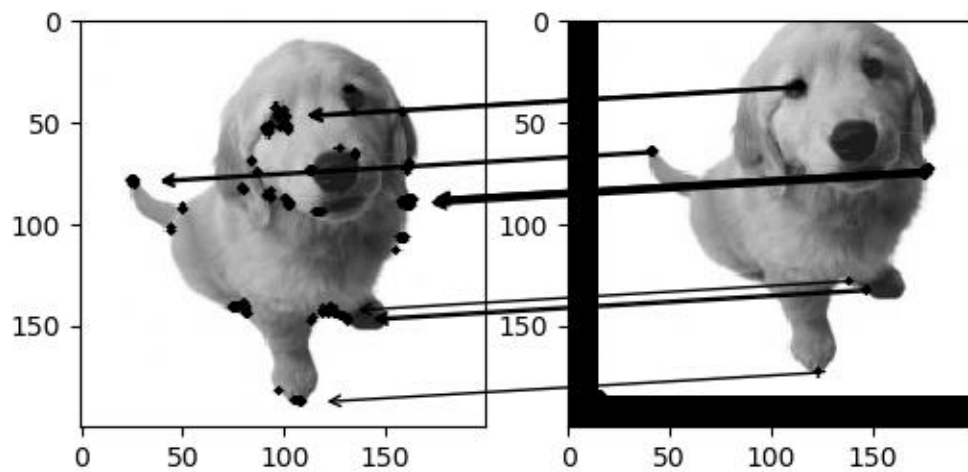


Imagen original e imagen2(x=15, y=-15)

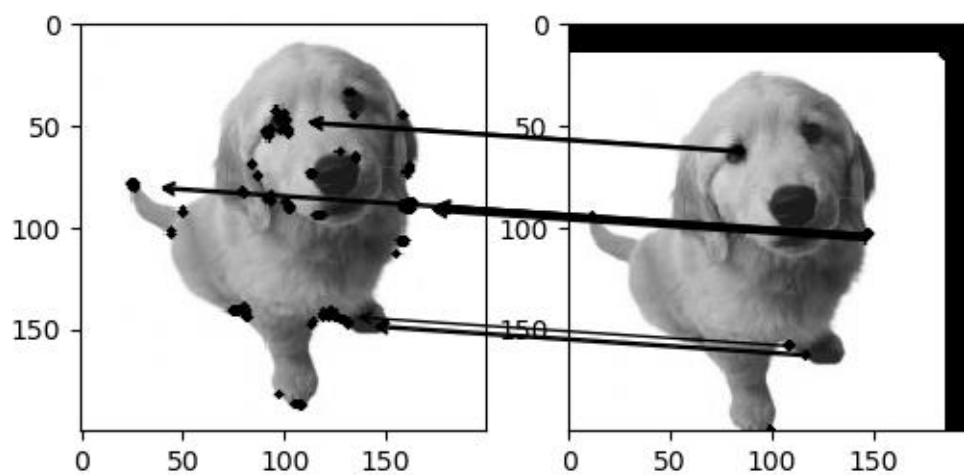


Imagen original e Imagen3(x=-15, y=15)

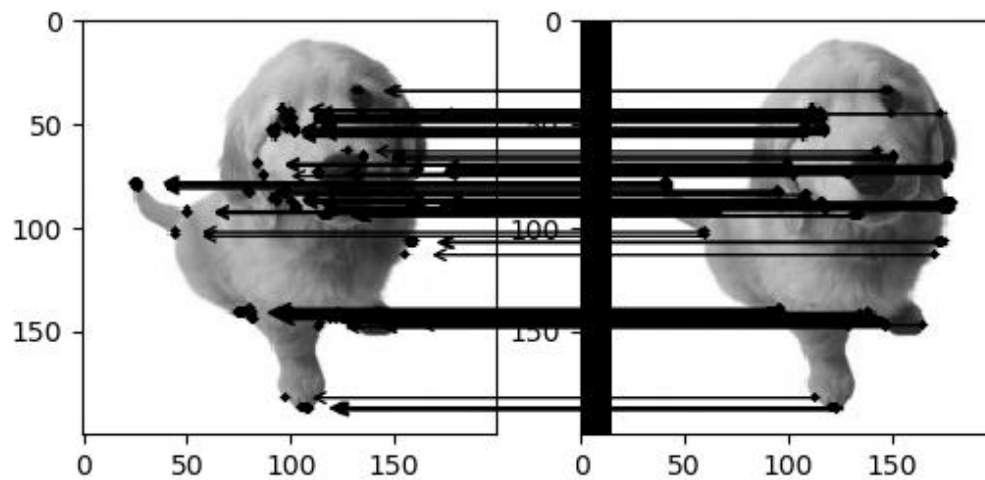


Imagen original e imagen4(x=15, y=0)

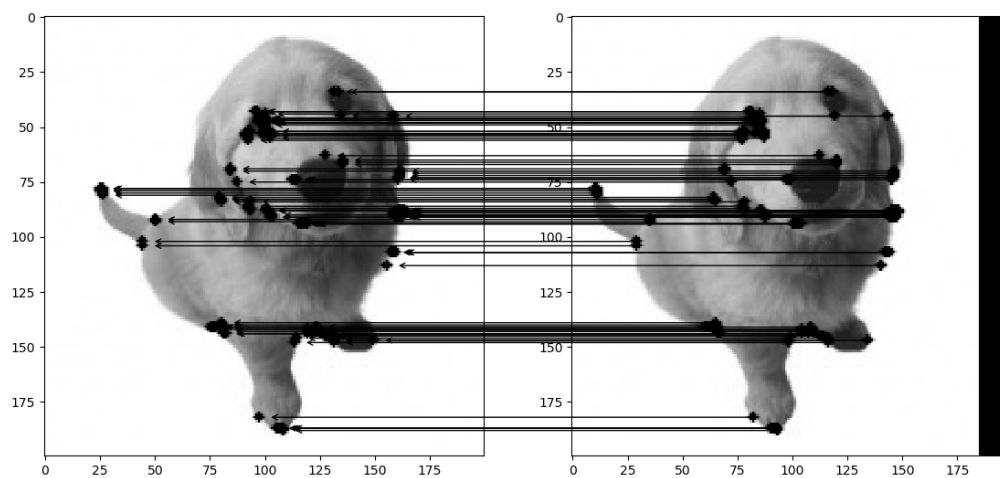


Imagen original e imagen5(x=-15, y=0)

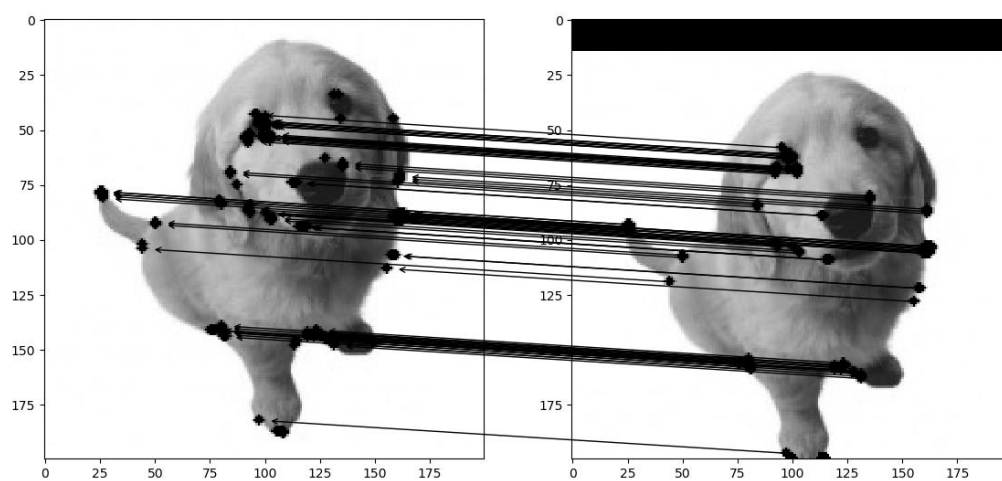


Imagen original e Imagen6(x=0, y=15)

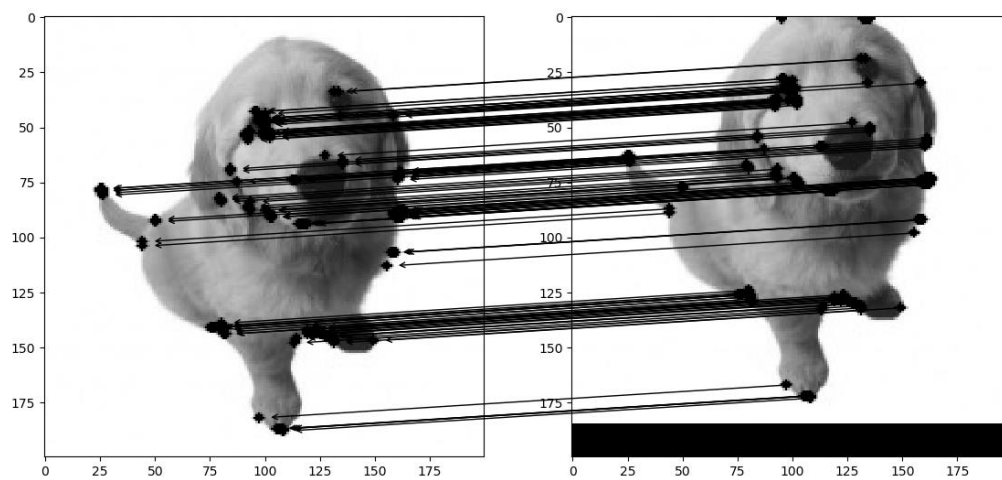


Imagen original e imagen7(x=0, y=-15).

Resultados de imágenes escaladas



Imagen original

200X200



imagen $\frac{1}{4}$

50x50



imagen $\frac{1}{2}$

100x100

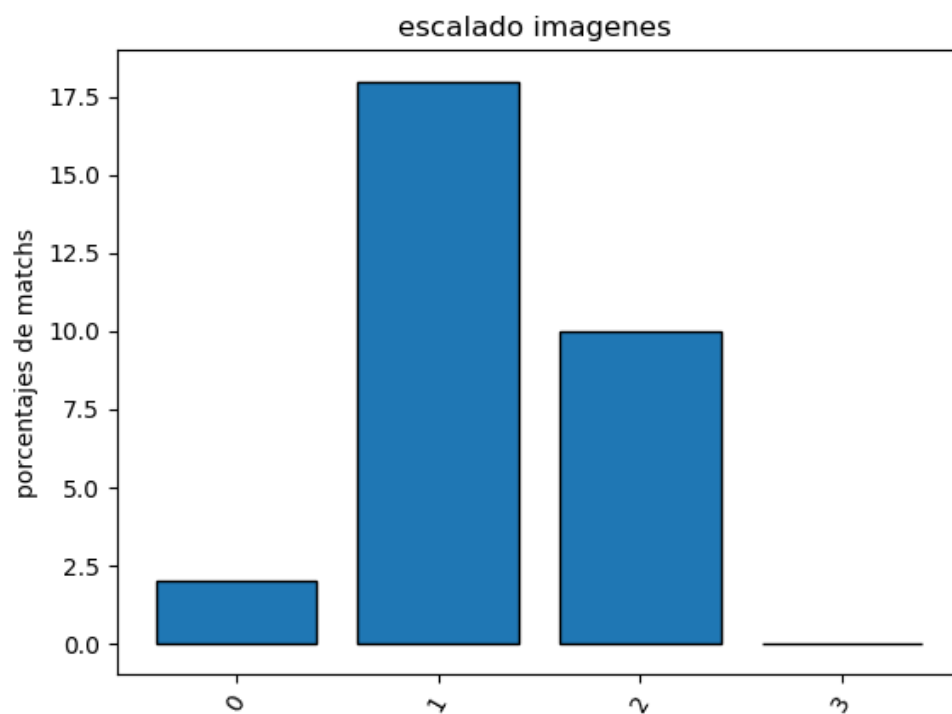


Imagen 400x400



Imagen 800x800

GRFIACAS DE CORRESPONDENCIAS



Correspondencias

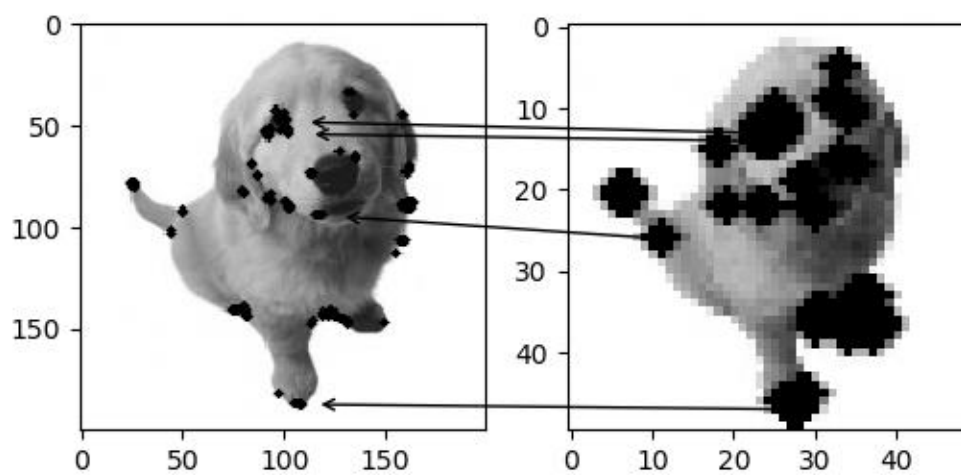


Imagen 200X200 e Imagen $\frac{1}{2}$

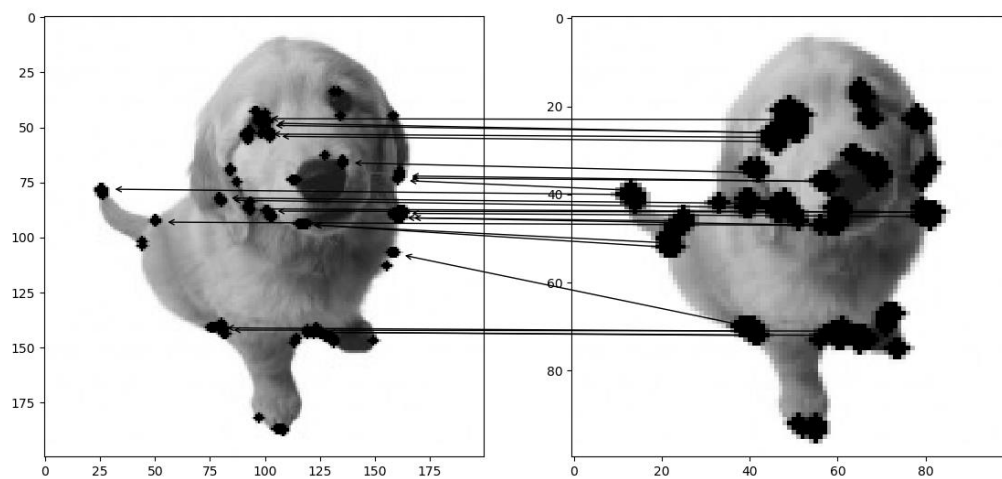


Imagen 200x200 e Imagen $\frac{1}{2}$

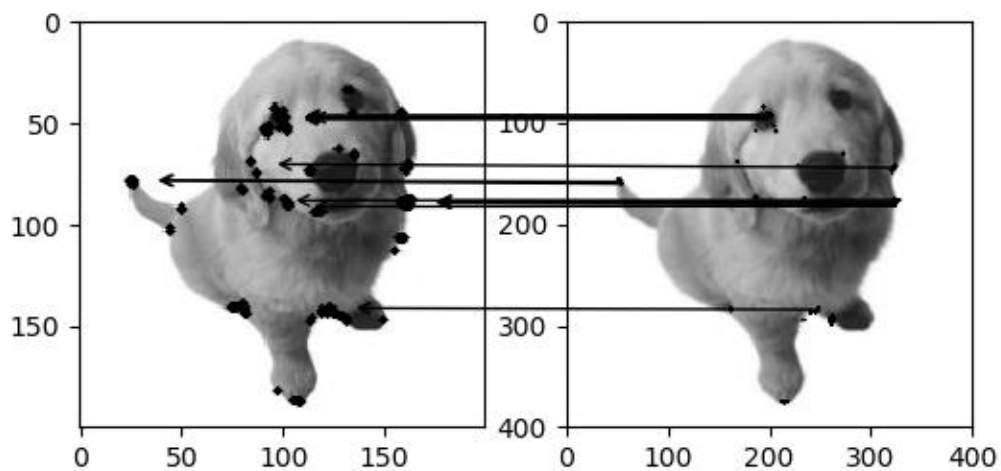


Imagen 200x200 e Imagen 400X400

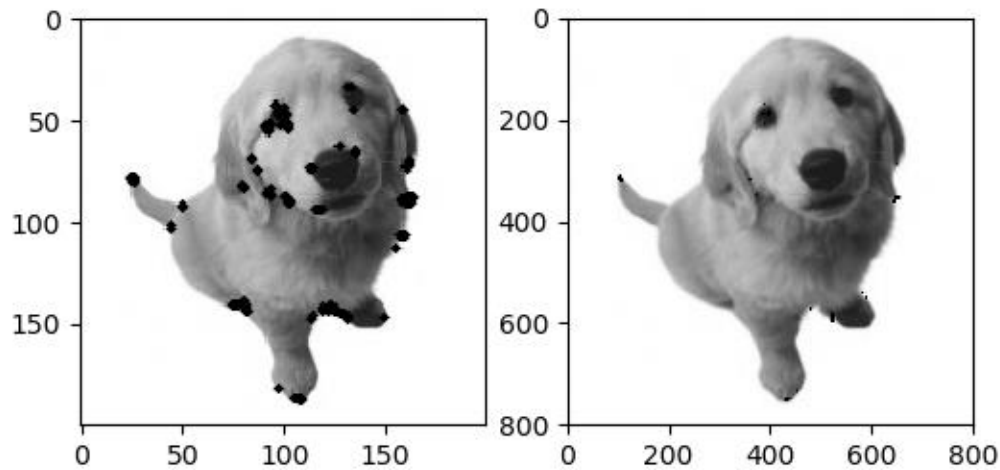


Imagen 200X200 e Imagen 800x800

Bibliografías

Autor	Tutor de Programación
Fecha de publicación	julio 11, 2017
Título del articulo	Detección de esquinas con OpenCV
Fuente	Pagina web
Url	http://acodigo.blogspot.com/2017/07/deteccion-de-esquinas-con-opencv.html
Fecha de recuperación	3/3/2020

Autor	Coding game
Fecha de publicación	Sin fecha
Título del articulo	Basic Image Manipulation
Fuente	Página web
Url	https://www.codingame.com/playgrounds/2524/basic-image-manipulation/transformation
Fecha de recuperación	4/3/2020

Autor	Cristian Pérez Brokate
Fecha de publicación	May 13, 2017
Título del articulo	Image rotation using OpenCV
Fuente	Página web
Url	https://cristianpb.github.io/blog/image-rotation-opencv
Fecha de recuperación	29/3/2020

Autor	Sourabh Sinha
Fecha de publicación	Sin fecha
Título del artículo	Image Translation using OpenCV
Fuente	Página web
Url	https://www.geeksforgeeks.org/image-translation-using-opencv-python/
Fecha de recuperación	25/03/2020

Autor	Professor Peter Corke
Fecha de publicación	Sin fecha
Título del artículo	Introduction to Corner Features (Harris)
Fuente	Página web
Url	https://robotacademy.net.au/lesson/introduction-to-corner-features-harris/
Fecha de recuperación	20/03/2020

Autor	Vino Mahendran
Fecha de publicación	Jan 13 · 6 min read
Título del artículo	Feature detection and matching with OpenCV
Fuente	Página web
Url	https://blog.francium.tech/@vinom
Fecha de recuperación	18/03/2020

Autor	Uni Python
Fecha de publicación	abril 5, 2018
Título del artículo	transformations geometrics of image
Fuente	Página web
Url	https://unipython.com/transformaciones-geometricas-de-imagenes-con-opencv/
Fecha de recuperación	17/03/2020

Autor	Alexander Mordvintsev & Abid K. Revision
Fecha de publicación	2013
Título del artículo	Feature Matching
Fuente	Página web
Url	https://opencv-python-tutroals.readthedocs.io/
Fecha de recuperación	15/03/2020