

MÁSTER EN INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL

VISIÓN ARTIFICIAL

Memoria M0: Bloque I: Visión Artificial Tradicional

Autora:
Marta Rodríguez Sampayo



24 de enero de 2020

Índice

1. Distinctive Image Features from Scale-Invariant Keypoints	3
1.1. Etapas para la generación del descriptor SIFT	3
1.2. Cumplimiento del método científico	3
1.2.1. Otros ejemplos	4
1.3. Comparativa de descriptores	5
2. Datasets	9
2.1. Uso de los descriptores	9
2.1.1. Resultados experimentales	9
2.1.2. Discusión y conclusiones	13
2.2. Dataset 1	14
2.2.1. Resultados experimentales	15
2.2.2. Discusión y conclusiones	21
2.3. Dataset 2	21
2.3.1. Resultados experimentales	22
2.3.2. Discusión y conclusiones	28
2.4. Dataset 3	28
2.4.1. Contornos	29
2.4.2. Regiones de interés	31
2.4.3. Discusión y conclusiones	33

Índice de figuras

1. Aplicación del descriptor SIFT	9
2. Aplicación del descriptor SURF	10
3. Aplicación del detector FAST	10
4. Aplicación del descriptor BRIEF	11
5. Aplicación del descriptor HOG	11
6. Aplicación del descriptor ORB	12
7. Aplicación del descriptor BRISK	12
8. Aplicación del descriptor AKAZE	13
9. Correspondencias entre puntos de interés (SIFT)	15
10. Transformaciones obtenidas con el descriptor SIFT	15
11. Correspondencias entre puntos de interés (SURF)	16
12. Transformaciones obtenidas con el descriptor SURF	16
13. Correspondencias entre puntos de interés (BRIEF)	17
14. Transformaciones obtenidas con el descriptor BRIEF	17
15. Correspondencias entre puntos de interés (ORB)	18
16. Transformaciones obtenidas con el descriptor ORB	18
17. Correspondencias entre puntos de interés (BRISK)	19
18. Transformaciones obtenidas con el descriptor BRISK	19
19. Correspondencias entre puntos de interés (AKAZE)	20
20. Transformaciones obtenidas con el descriptor AKAZE	20
21. Correspondencias entre puntos de interés (SIFT)	22
22. Transformaciones obtenidas con el descriptor SIFT	22
23. Correspondencias entre puntos de interés (SURF)	23
24. Transformaciones obtenidas con el descriptor SURF	23
25. Correspondencias entre puntos de interés (BRIEF)	24
26. Transformaciones obtenidas con el descriptor BRIEF	24
27. Correspondencias entre puntos de interés (ORB)	25
28. Transformaciones obtenidas con el descriptor ORB	25
29. Correspondencias entre puntos de interés (BRISK)	26

30.	Transformaciones obtenidas con el descriptor BRISK	26
31.	Correspondencias entre puntos de interés (AKAZE)	27
32.	Transformaciones obtenidas con el descriptor AKAZE	27
33.	Contornos	29
34.	Elección del parámetro K	29
35.	Regiones de interés	31
36.	Elección del parámetro K	32

Índice de tablas

1.	Comparativa de descriptores (1)	5
2.	Comparativa de descriptores (2)	5
3.	Comparativa de descriptores (3)	6
4.	Informe de clasificación para 3 clases (Características extraídas de los contornos)	30
5.	Informe de clasificación para 2 clases (Características extraídas de los contornos)	30
6.	Exactitud obtenida para los distintos modelos con 2 o 3 clases	30
7.	Informe de clasificación para 3 clases (Características extraídas de las regiones de interés) . . .	32
8.	Informe de clasificación para 2 clases (Características extraídas de las regiones de interés) . .	32
9.	Exactitud obtenida para los distintos modelos con 2 o 3 clases	33

1. Distinctive Image Features from Scale-Invariant Keypoints

1.1. Etapas para la generación del descriptor SIFT

1. Detección de características locales

- Detección de máximos y mínimos en la escala.
- Identificación de localizaciones que no varían con cambios de escala. Diferencia de gaussianas: aplicación de filtros gaussianos con diferentes escalas (valores de σ) a la imagen original y cálculo de la diferencia entre cada par de imágenes resultantes con escalas adyacentes.
- Re-escalado
- Búsqueda de máximos y mínimos
- Refinamiento de la detección
 - Refinar la localización exacta
 - Eliminación de puntos de contraste bajo
 - Eliminación de puntos en los contornos de la imagen

2. Descripción de características locales

- Histograma de orientaciones del gradiente
- Cálculo de la orientación del punto de interés
- Cálculo del descriptor
- Normalización del descriptor

1.2. Cumplimiento del método científico

En líneas generales, se puede afirmar que el artículo sigue el método científico, presentando una visión general del problema, planteando una solución y realizando experimentos y análisis de los resultados de los mismos para obtener los valores óptimos de los parámetros.

Tipos de justificación:

- Basada en **investigaciones previas** con resultados fiables y aplicables al problema observado. Además de citar los trabajos, se exponen los resultados y el razonamiento seguido para aplicarlos. Por ejemplo, para la elección de la diferencia de Gaussianas como método para crear el espacio de escalas.
- Basada en los resultados de la **realización de experimentos**. Partiendo de cierto escenario, se llevan a cabo un número de experimentos con diferentes parámetros sobre ciertos datos y se escogen los parámetros con los que se obtienen mejores resultados. En el artículo se presentan los resultados de los diferentes experimentos y se detalla el tipo de datos utilizados. Un ejemplo de esto es la elección del tamaño (4x4) y del número de orientaciones (8) del descriptor.
- Basada en un **razonamiento matemático**. Se plantea un problema y la resolución del mismo siguiendo un método conocido. La eficacia y/o eficiencia de dicho método se demuestra con el desarrollo matemático de dicha solución. Como ejemplo de esto se puede presentar la elección de la matriz Hessiana y el correspondiente cálculo para el cálculo de un ratio (y de su umbral correspondiente, con un valor de $r=10$) entre las curvaturas principales.

En todos los casos se expone de forma clara y razonada la elección de los métodos y los valores de los parámetros y la adecuación de los mismos a los diferentes escenarios planteados.

1.2.1. Otros ejemplos

- Elección de la función Gaussiana: basada en otras investigaciones previas.
- Diferencia de Gaussianas. Motivo: eficiencia, permite invarianza de escala. Base: investigación previa.
- Valor de $k = \sqrt{2}$. Base: experimentación práctica.
- Frecuencia de muestreo y dominios de escala. Motivo: compromiso entre eficiencia e integridad. Base: experimentación práctica (estudio de un rango de frecuencias de muestreo y usando aquellas que proporcionan los resultados más fiables en una simulación real de la tarea de matching)
- Frecuencia de muestreo en escala. Valor: 3 escalas por octava. Base: experimentación (repeticibilidad más alta, al aumentar la escala no mejora la eficacia pero si el coste computacional, así que no se aumenta)
- Frecuencia de muestreo en el dominio espacial. Valor: $\sigma = 1.6$. Base: experimentación (más cercano al valor óptimo).
- Pre-suaviazado. Motivo: descartar las frecuencias espaciales más altas.
- Expandir la imagen para crear más puntos de muestra. Conclusión: se dobla la imagen para mayor eficiencia
- Refinamiento de la localización de puntos de interés. Base: otras investigaciones. Motivo: mejora notable utilizando la expansión de Taylor.
- Asignación de orientación. Elección: histograma de gradientes orientados (HoG). Motivo: invariante a rotación de la imagen. Base: contraste con otras investigaciones y experimentación (resultados más estables).
- Reducción de los efectos en los cambios de las magnitudes de los gradientes. Decisión: valores menores que 0.2 y renormalizar. Base: evaluación experimental con imágenes de diferentes iluminaciones.
- Número de orientaciones y tamaño del descriptor. Valores: 8 y 4x4 (aumentar esto hace el descriptor demasiado sensible a distorsiones). Base: evaluación experimental.

1.3. Comparativa de descriptores

A continuación se presentan las tablas con el resumen de las características principales de la construcción de los distintos descriptores. Con el objetivo de simplificar la información de las tablas 1 y 2, se define posteriormente el funcionamiento de cada descriptor.

	SIFT	SURF	FAST
Espacio-escala	DoG	Box filters	X
Detección de puntos de interés	Máximos y mínimos en cada escala	Detector BLOB	Comparación de intensidades entre píxeles cercanos
Asignación de orientación	Histograma de orientaciones del gradiente	Respuestas Haar-Wavelet	X
Generación del descriptor	Vector con todas las entradas del histograma	Respuestas Haar-Wavelet	Algoritmo ID3
Matching	Distancia euclídea	Distancia euclídea	X
Tamaño del descriptor (bytes)	128	64 o 128	X
Invarianza	Escala y rotación Robusto frente a cambios en iluminación, ruido y punto de vista (parcialmente)	Escala y rotación Robusto frente a ruido y deformaciones geométricas y fotométricas	X
Descriptor binario	No	No	No

Tabla 1: Comparativa de descriptores (1)

	BRIEF	HOG	ORB
Espacio-escala	X	X	X
Detección de puntos de interés	X	Cálculo del gradiente	FAST + Harris corner measure
Asignación de orientación	X	Orientación del gradiente	Centroide de Intensidad (oFAST)
Generación del descriptor	Comparación entre intensidades de píxeles y creación de una cadena binaria.	Concatenación de los histogramas de cada celda	rBRIEF
Matching	Distancia Hamming	X	Distancia Hamming
Tamaño del descriptor (bytes)	16, 32 o 64	X	32
Invarianza	X	X	Rotación Robusto frente a ruido y cambios de iluminación y de perspectiva
Descriptor binario	Sí	No	Sí

Tabla 2: Comparativa de descriptores (2)

	BRISK	AKAZE
Espacio-escala	Muestreo	FED
Detección de puntos de interés	FAST en cada escala y octava	Determinante de la Hessiana + filtros Scharr
Asignación de orientación	Patrón de muestreo + gradiente	Respuestas de las derivadas de primer orden
Generación del descriptor	Rotación de los pares de píxeles del patrón	MLDB
Matching	Distancia Hamming	Distancia Hamming
Tamaño del descriptor (bytes)	64	8, 32 (bytes) o 486 bits
Invarianza	Escala y rotación Robustez limitada frente a transformaciones afines	Escala y rotación Robustez limitada frente a transformaciones afines
Descriptor binario	Sí	Sí

Tabla 3: Comparativa de descriptores (3)

■ Speeded-Up Robust Features (SURF) [1]

- Construcción del espacio-escala: se utilizan filtros de aproximación (box filters) en lugar de gaussianas, de forma que aumenta la rapidez del algoritmo respecto a SIFT.
- Detección de puntos de interés: detector BLOB (Binary Large Object) basado en la matriz Hessiana.
- Cálculo de la orientación: las respuestas Wavelet de Haar, realizadas con un Gaussiano centrado en el punto de interés, se representan como puntos en el espacio. Una vez calculados para todos los vecinos, se estima la orientación dominante calculando la suma de todos los resultados dentro de una ventana deslizante que cubre un ángulo de $\pi/3$. Finalmente se suman la respuesta horizontal y vertical con la ventana y el vector más largo de las ventanas define la orientación del punto de interés.
- Generación del descriptor: se construye una región cuadrada centrada en el punto de interés, esta se divide en 4x4 regiones y, para cada una de ellas, se calculan el Wavelet de Haar para x e y y se suavizan los resultados mediante un filtro Gaussiano. Se forma un vector con las sumas de estos resultados y sus valores absolutos y el descriptor es la unión de los vectores de las subregiones.

- Features from Accelerated Segment Test (FAST) [2]

FAST es un detector de bordes, por tanto no sigue exactamente el mismo esquema que SIFT a la hora de calcular el descriptor.

- Detección de puntos de interés: se considera un círculo de 16 píxeles alrededor del píxel candidato, si existen n píxeles contiguos en dicho círculo cuyo valor de intensidad sea mayor que el del candidato más cierto umbral, este se considera perteneciente a un borde (análogamente para valores mínimos de intensidad).
- Machine learning: Se almacenan estos 16 píxeles en un vector de características que se divide en 3 subregiones (más oscuro, similar o más luminoso que el punto de interés) y se crea un árbol de decisión (algoritmo ID3) utilizando para ello la entropía de la variable de tipo booleano que define si es un borde o no. El objetivo es conseguir la máxima ganancia de información (entropía mínima), escogiendo a través de un proceso recursivo los puntos que lo alcanzan.

- Binary Robust Independent Elementary Features (BRIEF) [3]

BRIEF es un descriptor de características, pero no implementa un mecanismo para encontrar los puntos de interés.

- Preprocesado de la imagen: filtro gaussiano para suavizar la imagen.
- Generación del descriptor: para construir el vector binario que representa una región alrededor de un punto de interés se recorren cierto número de pares (x,y) y se compara la intensidad de x e y, almacenando un 1 cuando la intensidad de 1 primero es mayor que la del segundo o un 0 en caso contrario. Puesto que BRIEF no implementa un mecanismo de detección, se proponen 5 métodos diferentes para determinar los pares (x,y) dentro de la región siguiendo diferentes distribuciones de probabilidad.

- Histogram of Oriented Gradients (HOG)

Este método cuenta el número de ocurrencias de orientación de gradiente en áreas locales de una imagen y difiere de otros descriptores como SIFT en que se calcula sobre una malla densa de celdas espaciadas uniformemente. Funcionamiento:

- Se divide la imagen en celdas y para cada celda se calcula un histograma de direcciones de gradiente u orientaciones de borde para los píxeles.
- Se discretiza cada celda en intervalos angulares de acuerdo con la orientación del gradiente.
- El píxel de cada celda contribuye de forma ponderada al gradiente de su correspondiente intervalo angular.
- Los grupos de celdas adyacentes se consideran regiones espaciales llamadas bloques. La agrupación de celdas en un bloque es la base para la agrupación y normalización de histogramas.
- El grupo normalizado de histogramas representa el histograma del bloque. El conjunto de estos histogramas de bloque representa el descriptor.

- Oriented FAST and Rotated BRIEF (ORB) [4]

- Detección de puntos de interés: utiliza FAST para encontrar los puntos de interés y el detector de bordes de Harris para encontrar los N puntos mejores.
- Asignación de orientación: FAST no calcula la orientación y por tanto no es invariante a la rotación, pero sí calcula el centroide ponderado de intensidad de la sección con el borde localizado en el centro. La dirección del vector desde ese punto del borde al centroide da la orientación. Se calculan los momentos para mejorar la invarianza de rotación.
- Generación del descriptor: se calcula una matriz de rotación y se utilizan descriptores BRIEF dirigidos según la orientación de los puntos de interés.

- Binary Robust Invariant Scalable Keypoints (BRISK) [5]

- Construcción del espacio-escala: las imágenes de cada octava son de la mitad del tamaño de las correspondientes a la octava anterior. Dentro de una misma octava, cada imagen es muestreada de forma que se encuentre entre las octavas inmediatamente anterior y posterior.
- Detección de puntos de interés: FAST-9 para cada escala y octava.
- Cálculo de la orientación: se configura un patrón de muestreo (60 muestras) de píxeles suavizados con un filtro Gaussiano alrededor del punto de interés, se separan los pares de píxeles en dos subconjuntos (corta y larga distancia) y se calcula el gradiente local entre los pares de larga distancia, a partir de la suma de los gradientes se determina la orientación del punto de interés.
- Generación del descriptor: se rotan los pares de corta distancia usando la orientación y se construye un descriptor binario a partir de ellos.

- Accelerated-KAZE (AKAZE) [6]

- Construcción del espacio-escala: FED (*Fast Explicit Diffusion*)
- Detección de puntos de interés: puntos donde la respuesta del determinante normalizado de la Hessiana en múltiples escalas es máxima.
- Cálculo de la orientación: estimación de la orientación dominante en un vecindario local centrado en la ubicación del punto de interés. Para cada una de las muestras en el área circular, se ponderan las derivadas de primer orden con una Gaussiana centrada en el punto de interés. Posteriormente, las respuestas de las derivadas se representan como puntos en el espacio vectorial y la orientación dominante se define sumando las respuestas dentro de un segmento circular deslizante.
- Generación del descriptor: MLDB *Modified-Local Difference Binary*

2. Datasets

En esta sección se presentan los resultados obtenidos al aplicar los descriptores de características explícadas anteriormente a los conjuntos de imágenes proporcionados.

En el primer apartado, a modo de ejemplo del uso de cada descriptor, se muestran los puntos detectados en el modelo para cada uno de ellos. En los dos siguientes se realiza un análisis de la solución implementada para encontrar las transformaciones aplicadas a las imágenes de los dos primeros datasets. Finalmente, para el tercer conjunto de datos, se propone una solución para distinguir entre las distintas categorías.

2.1. Uso de los descriptores

Con el fin de llevar a cabo una comparación práctica de los descriptores, se han aplicado al modelo original de la figura del REY, recopilando el tiempo de ejecución y el número de puntos de interés extraídos.

La implementación íntegra del operador SIFT se ha llevado a cabo y se adjunta su código junto a esta memoria, sin embargo no se utilizará en el resto de tareas, debido a que el tiempo de ejecución es muy elevado y el número y calidad de los puntos de interés son mucho menores en comparación al resto de descriptores, por lo que los resultados no resultarían relevantes.

Por otro lado, el operador BRIEF es un método de descripción de características que no ofrece la funcionalidad de obtener los puntos de interés antes de generar los descriptores. Por ello, se utiliza en conjunto con el detector FAST.

2.1.1. Resultados experimentales

1. SIFT

- Número de puntos de interés: 453
- Tiempo: 0.0398 segundos

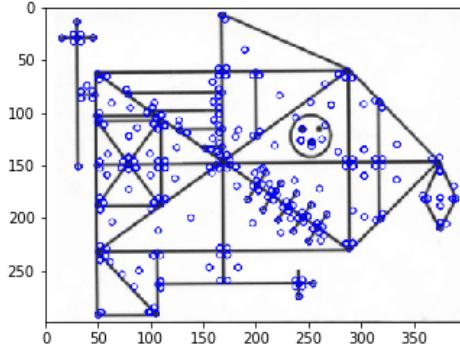


Figura 1: Aplicación del descriptor SIFT

2. SURF

- Número de puntos de interés: 1175
- Tiempo: 0.0767 segundos

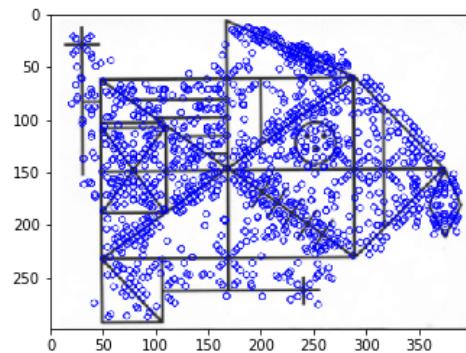


Figura 2: Aplicación del descriptor SURF

3. FAST

- Número de puntos de interés: 569
- Tiempo: 0.00036 segundos

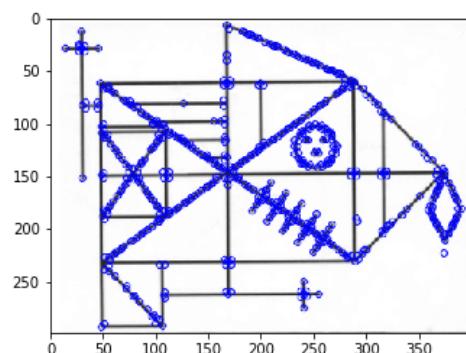


Figura 3: Aplicación del detector FAST

4. BRIEF

- Número de puntos de interés extraídos: 513
- Tiempo: 0.00083 segundos

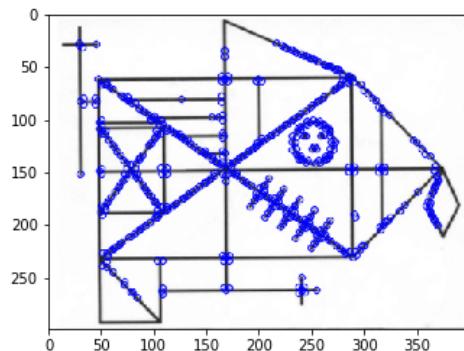


Figura 4: Aplicación del descriptor BRIEF

5. HOG

Puesto que se ha utilizado una librería distinta (skicit en lugar de OpenCV) para esta implementación, los resultados de tiempo de ejecución no son comparables con el resto de métodos y por ello no se han incluido.

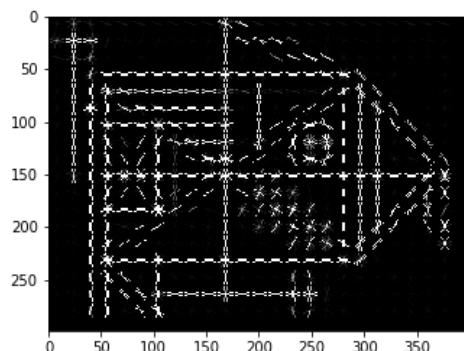


Figura 5: Aplicación del descriptor HOG

6. ORB

- Número de puntos de interés: 1447
- Tiempo: 0.0062 segundos

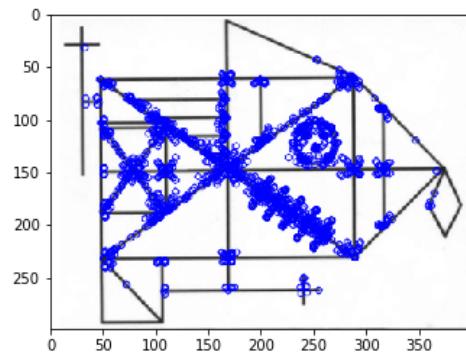


Figura 6: Aplicación del descriptor ORB

7. BRISK

- Número de puntos de interés: 707
- Tiempo: 0.0269 segundos

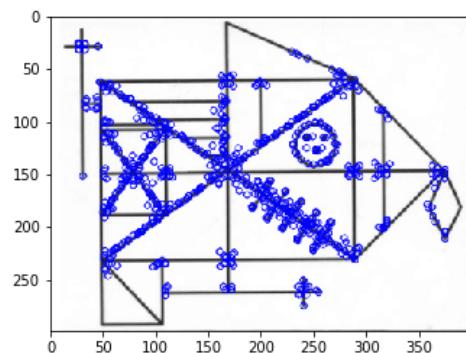


Figura 7: Aplicación del descriptor BRISK

8. AKAZE

- Número de puntos de interés: 389
- Tiempo: 0.0554 segundos

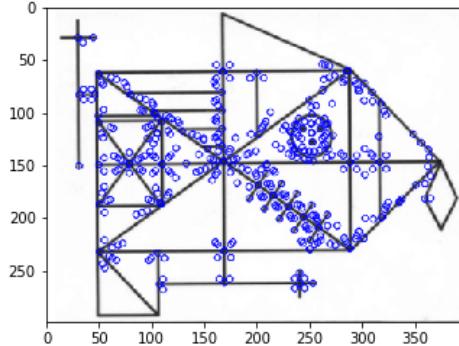


Figura 8: Aplicación del descriptor AKAZE

2.1.2. Discusión y conclusiones

El número de puntos de interés extraídos por cada descriptor es muy variable, obteniéndose el mayor valor, 1447, con ORB en contraposición a los 389 encontrados por AKAZE. En cuanto a la distribución de estos puntos, la mayor dispersión se aprecia en la figura 2, los puntos de interés obtenidos empleando SURF se distribuyen sobre la superficie del dibujo, mientras que con el resto de descriptores se concentran sobre las líneas del mismo. En general, los puntos detectados se encuentran en las intersecciones de las líneas del modelo y sus extremos, proporcionando una descripción que se puede considerar adecuada a simple vista, puesto que permitirá realizar las comparaciones necesarias entre las imágenes de los siguientes apartados, identificando los distintos elementos presentes en el dibujo. En las figuras 3, 4, 6 y 7 (respectivamente FAST, BRIEF, ORB y BRISK) se pueden apreciar muchas similitudes, además de puntos aislados encontramos líneas totalmente cubiertas (p.ej.: las diagonales principales). Teniendo en cuenta que FAST se utiliza junto a BRIEF y en las implementaciones de ORB y BRISK, es lógico que los resultados obtenidos empleando estos métodos sean similares. Por último, añadir que, aunque el número de puntos de interés calculados con SIFT y AKAZE es menor en comparación al resto de descriptores, (figuras 1 y 8), se aprecia una menor redundancia en la distribución de los mismos.

En cuanto a velocidad de ejecución, como era de esperar los descriptores binarios consiguen tiempos mucho menores con respecto a SIFT y SURF, siendo BRIEF con FAST el más rápido. En cualquier caso, al tratarse de tiempos tan bajos, en ningún caso afectan al uso que se hará de estos descriptores en las siguientes secciones.

En lo referente al descriptor HOG, no se ha comparado con el resto de descriptores puesto que su funcionamiento no se basa en la detección de puntos de interés, si no en el cálculo de los gradientes dentro de cada celda de la imagen, es decir el resultado depende, entre otros parámetros, del tamaño escogido para las celdas. La figura 5 es el resultado de aplicar HoG a la imagen con celdas de 16x16 píxeles y 8 intervalos de orientación, como se puede observar el modelo se reproduce de forma bastante fiel al original. Puesto que este método no extrae puntos de interés, no se empleará en los siguientes apartados para hallar las transformaciones de las imágenes.

Puesto que FAST es un detector de bordes, es decir no genera un descriptor de características en sí mismo como hacen otros métodos y ya que se utiliza para detectar los puntos de interés utilizados posteriormente en el cálculo del vector de características de BRIEF, no se utilizará este método por separado en los siguientes apartados.

2.2. Dataset 1

Este primer conjunto de datos (directorio dataset1_reyDistorsiones/) está formado por 7 imágenes obtenidas al aplicar ciertas transformaciones a la imagen modelo. Entre estas transformaciones se encuentran rotaciones a distintos grados, cambios de perspectiva y cambios de escalado.

La solución implementada, basada en https://docs.OpenCV.org/master/d1/de0/tutorial_py_feature_homography.html y https://OpenCV-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_matcher/py_matcher.html, consta de las siguientes etapas:

- Cálculo de los descriptores y puntos de interés de la imagen transformada y del modelo.
- Proceso de *matching* para encontrar coincidencias entre ambos conjuntos de puntos de interés y filtrado del resultado para conservar solo las mejores coincidencias. El *matcher* utilizado está basado en FLANN (*Fast Approximate Nearest Neighbor*) que permite escoger diferentes algoritmos para el cálculo de las correspondencias. Para los descriptores binarios se utiliza el algoritmo LSH multi-sonda (*multi-probe LSH*) [7] y en el resto de casos, árboles k-dimensionales (kd-trees) [8].
- Cálculo de la matriz de homografía para encontrar la transformación realizada.
- Una vez encontrada la transformación se aplica su inversa a la imagen transformada para observar si se consigue obtener el modelo original.

Con el objetivo de evaluar esta solución, se presentan para cada descriptor:

- Las correspondencias entre los puntos clave del modelo y dos imágenes.
- El resultado de aplicar la inversa de la transformación encontrada a la imagen inicial para todo el conjunto de datos.

2.2.1. Resultados experimentales

Con el fin de proporcionar una mayor legibilidad y facilitar con ello la comprensión de los resultados, se muestran las inversas de las transformaciones de todas las imágenes del dataset, pero solo las correspondencias de puntos de interés de la primera (t1.png) y la última (t7.png) para cada método de descripción de características.

1. SIFT

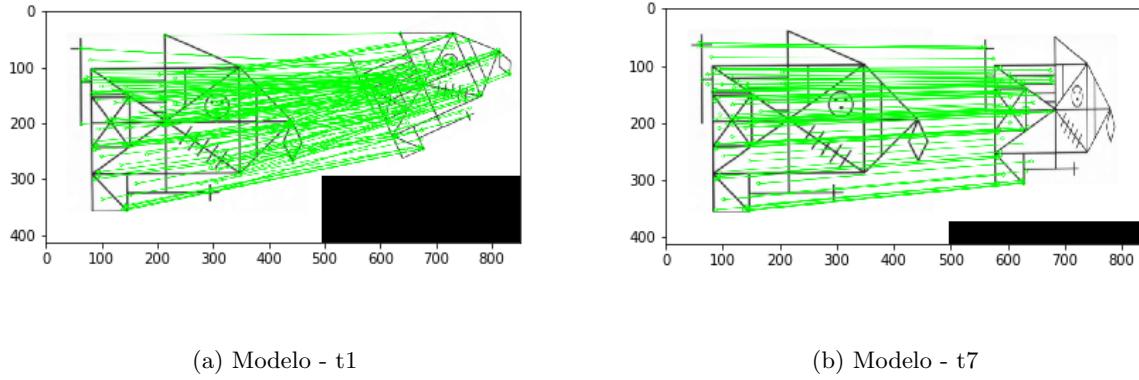


Figura 9: Correspondencias entre puntos de interés (SIFT)

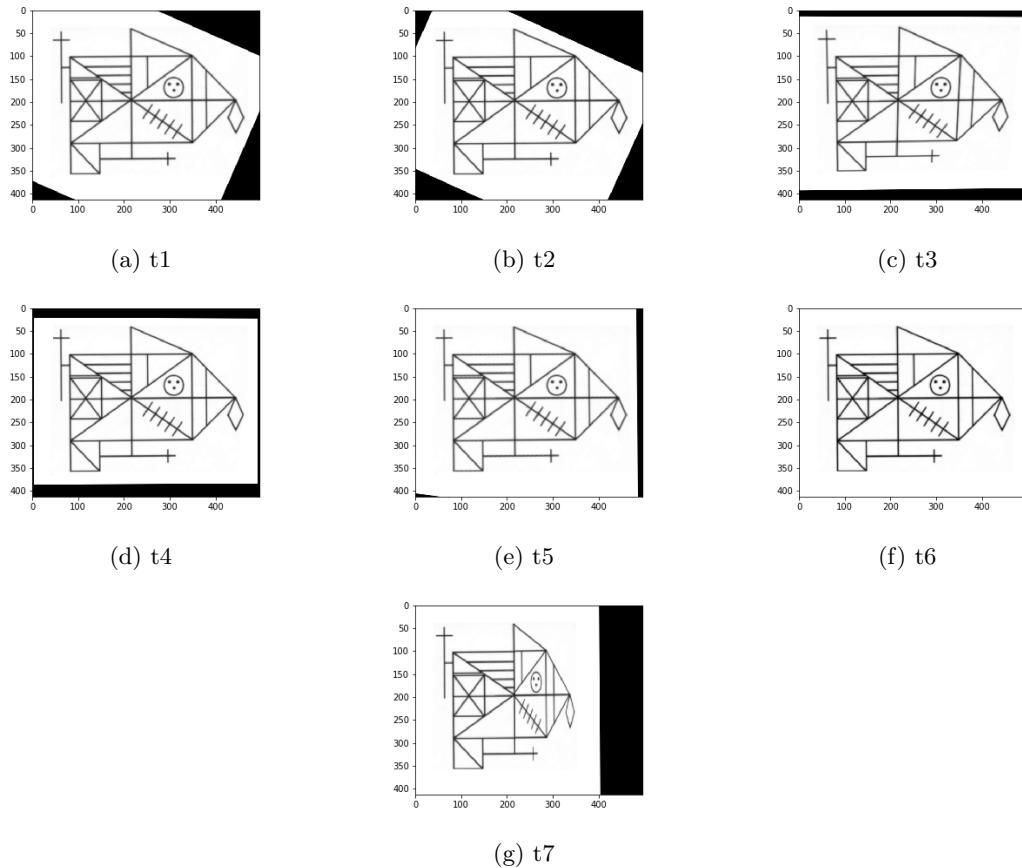


Figura 10: Transformaciones obtenidas con el descriptor SIFT

2. SURF

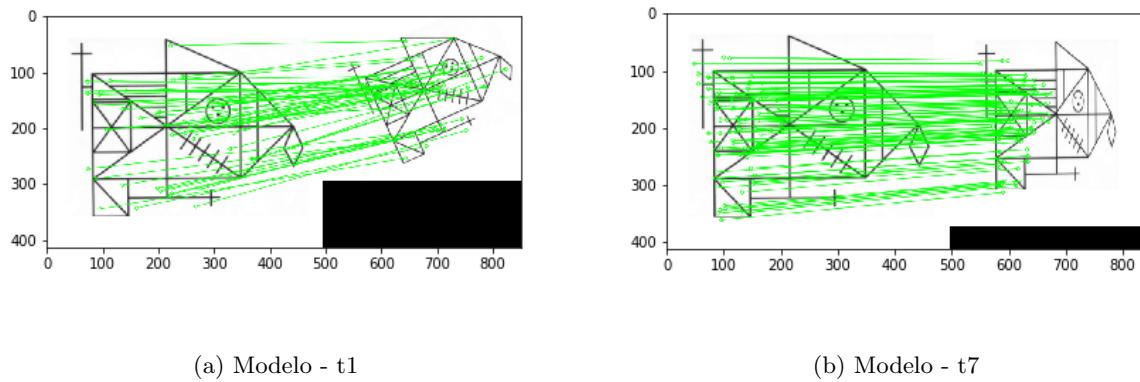


Figura 11: Correspondencias entre puntos de interés (SURF)

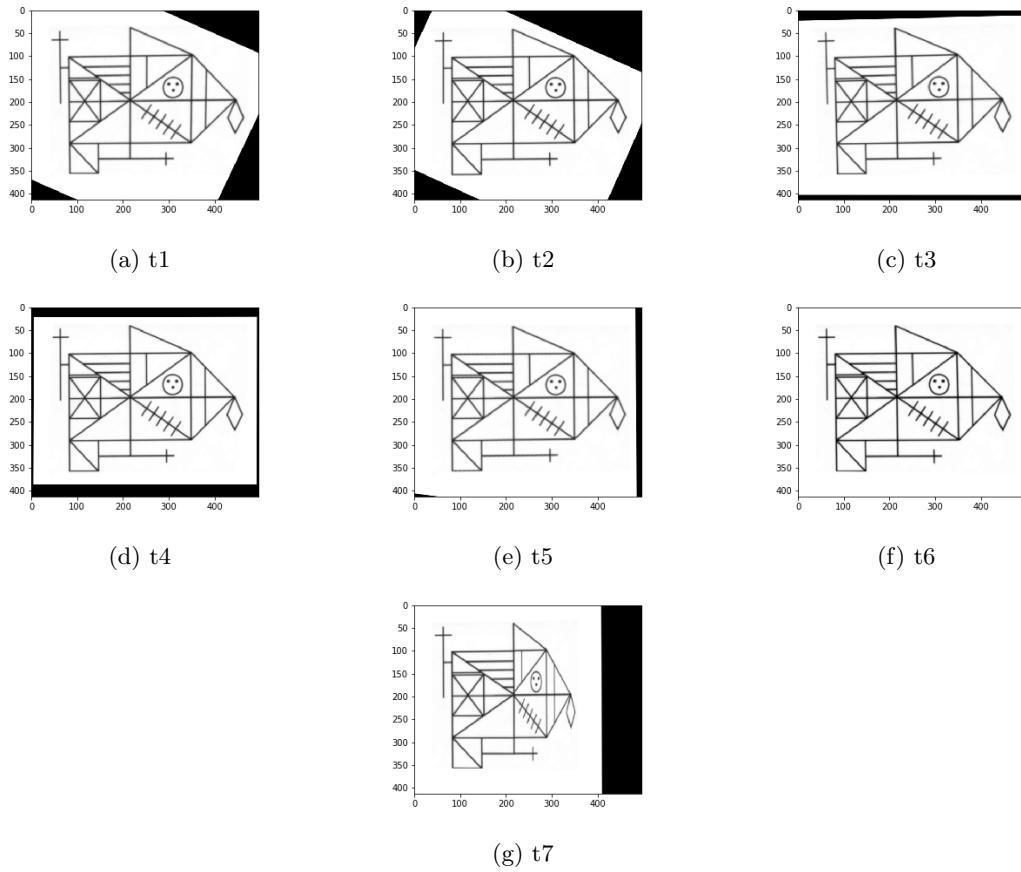


Figura 12: Transformaciones obtenidas con el descriptor SURF

3. BRIEF

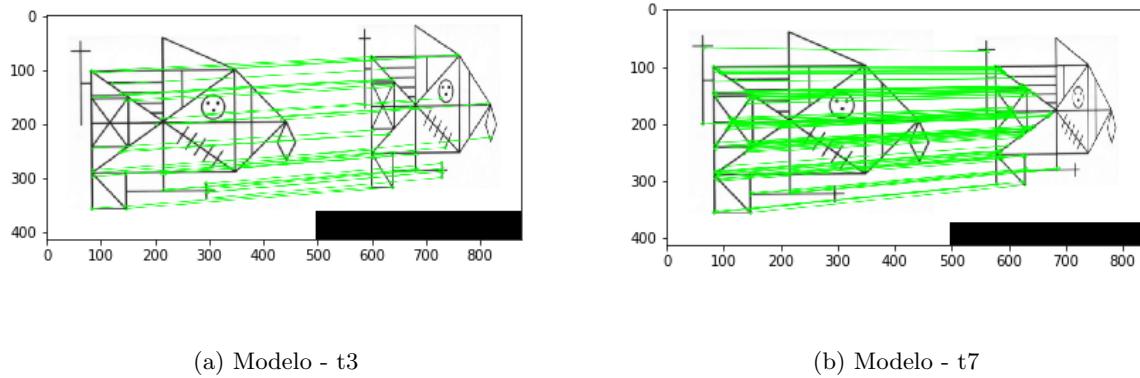


Figura 13: Correspondencias entre puntos de interés (BRIEF)

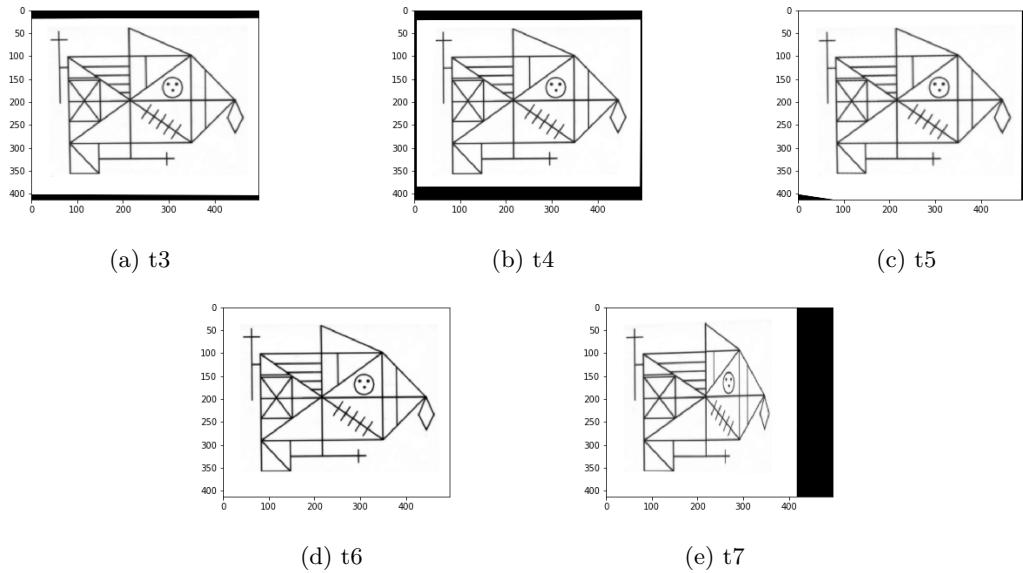


Figura 14: Transformaciones obtenidas con el descriptor BRIEF

4. ORB

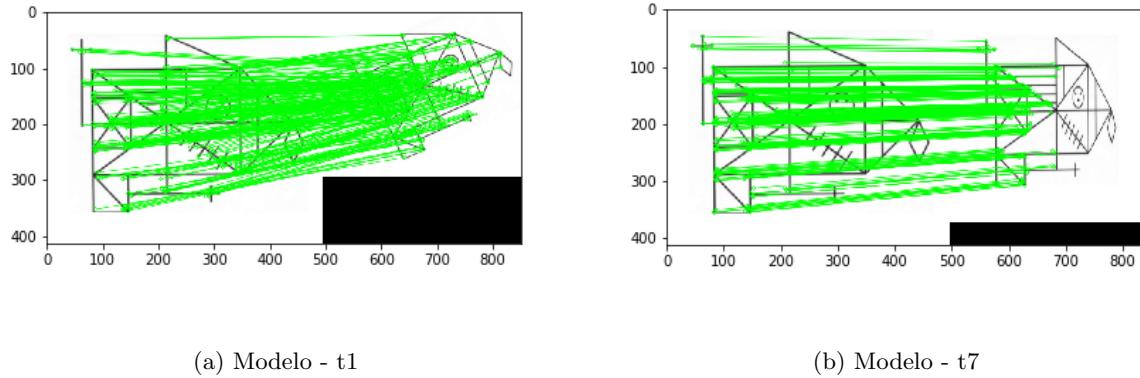


Figura 15: Correspondencias entre puntos de interés (ORB)

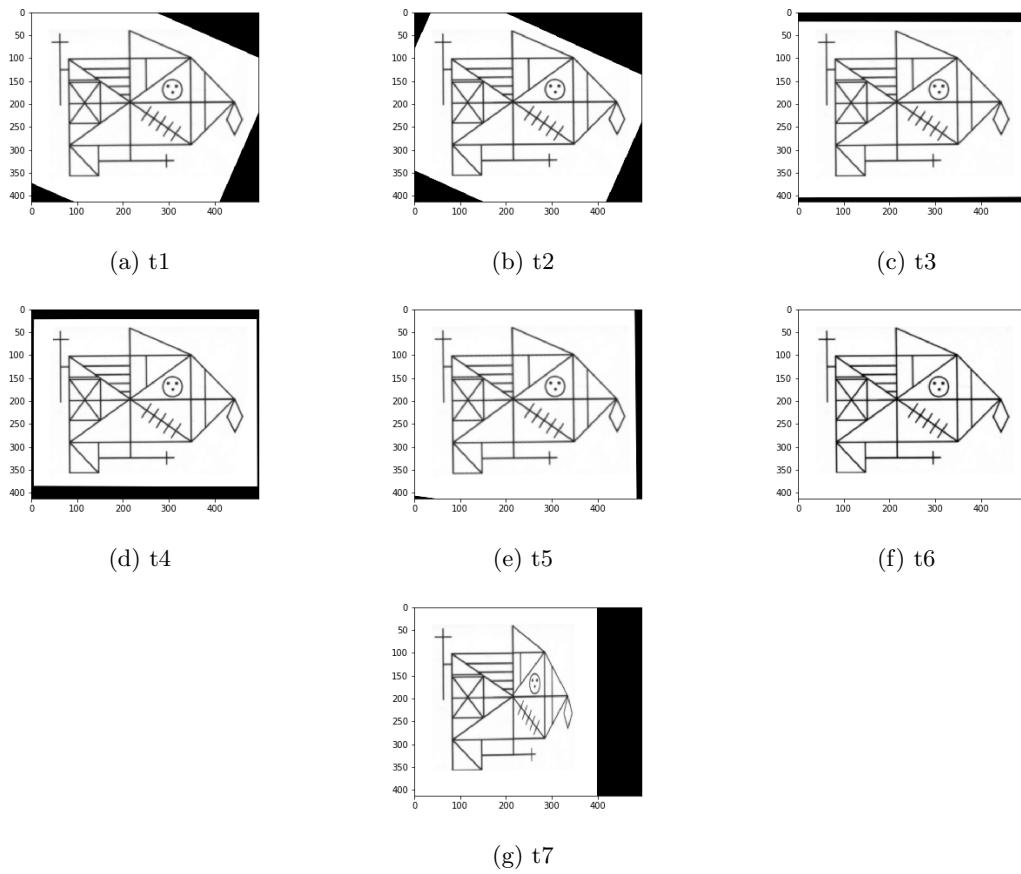


Figura 16: Transformaciones obtenidas con el descriptor ORB

5. BRISK

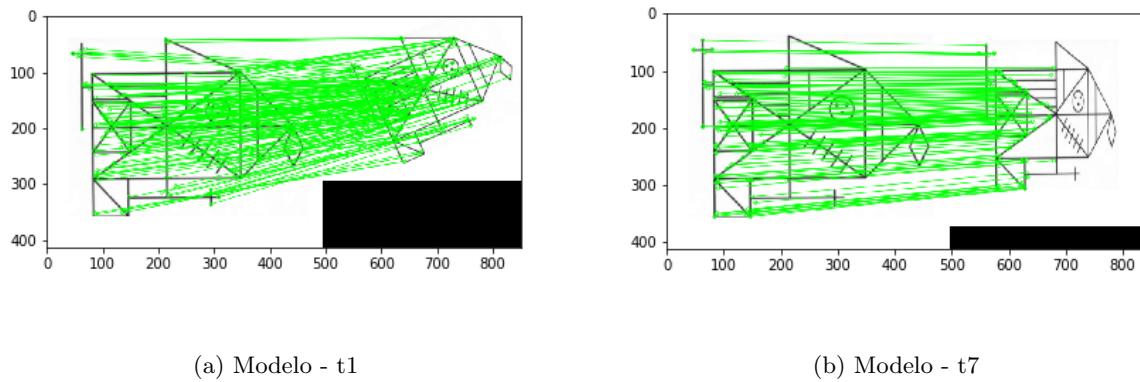


Figura 17: Correspondencias entre puntos de interés (BRISK)

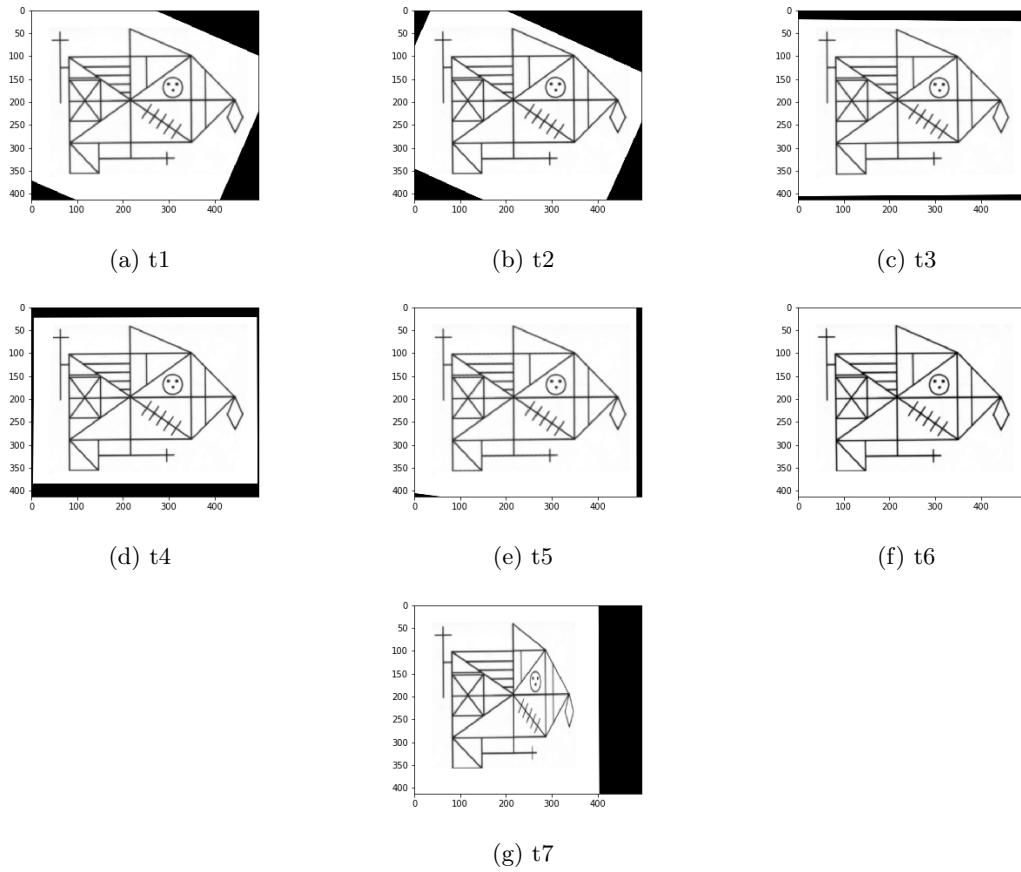


Figura 18: Transformaciones obtenidas con el descriptor BRISK

6. AKAZE

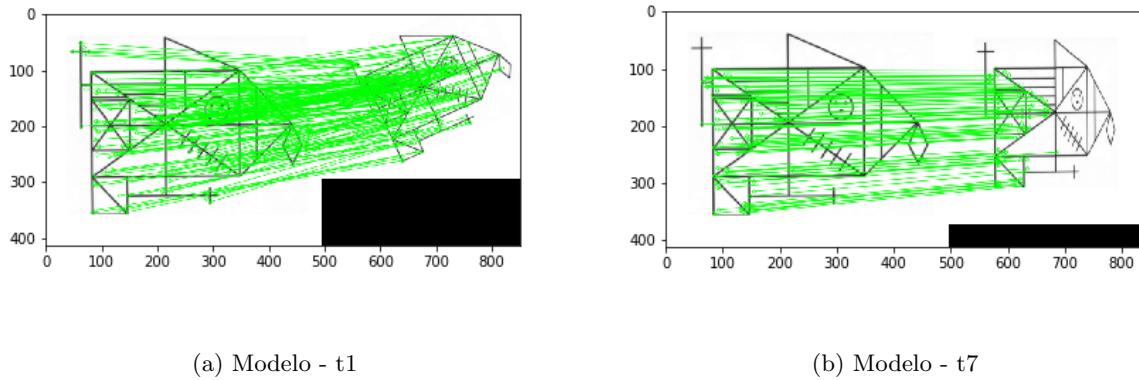


Figura 19: Correspondencias entre puntos de interés (AKAZE)

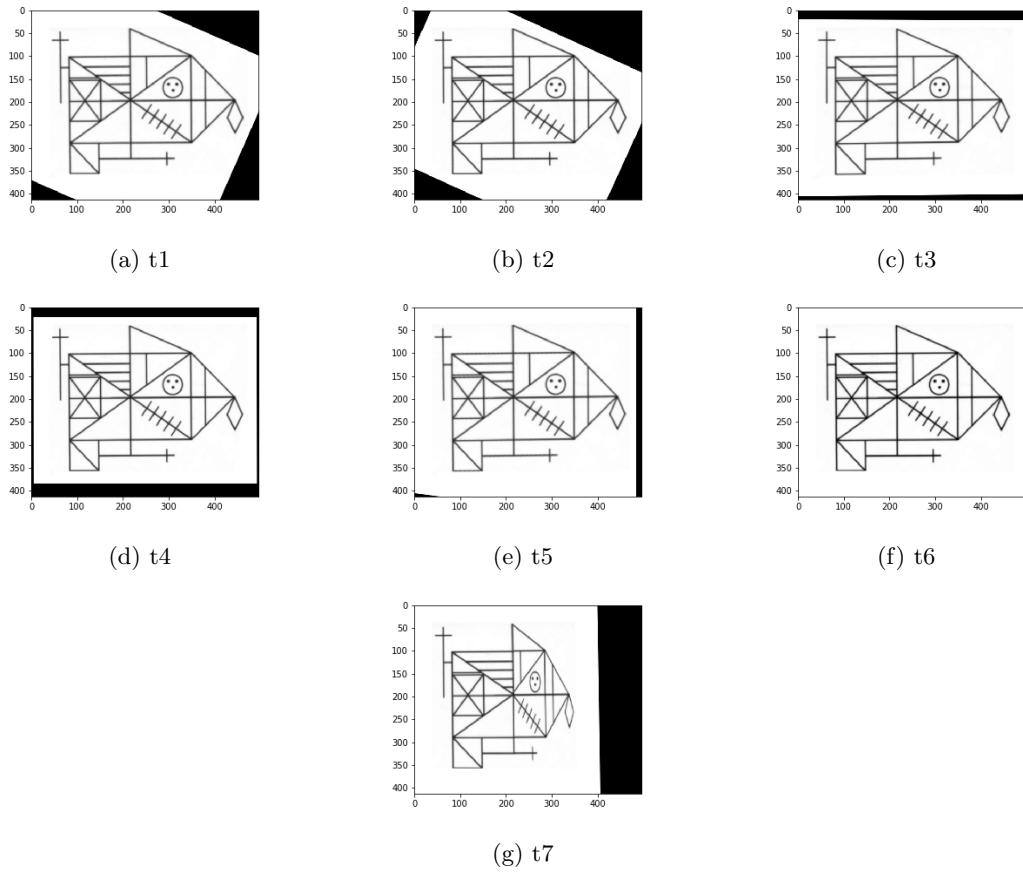


Figura 20: Transformaciones obtenidas con el descriptor AKAZE

2.2.2. Discusión y conclusiones

En las figuras 9, 11, 13, 15, 17 y 19 se puede apreciar que la cantidad de puntos de interés a los que se le encuentra correspondencia en ambas imágenes (modelo y transformación) es elevada. Como se comenta al inicio de la sección de resultados solo se presenta la correspondencia de dos imágenes. En el caso del descriptor BRIEF en lugar de la primera imagen del dataset (t1.png) se muestra la tercera (t3.png), esto se debe a que BRIEF no es invariante a rotación, por lo que no se encuentran suficientes correspondencias como para calcular la matriz de homografía de las dos primeras imágenes.

Las inversas de las transformaciones para las 6 primeras imágenes se realizan con total éxito para todos los descriptores, salvando la diferencia ya comentada con BRIEF, esto se puede comprobar en las figuras 10, 12, 14, 16, 18 y 20 en las que se observa el modelo recuperado de la transformación aplicada inicialmente. Sin embargo, ninguno de los descriptores empleados ha obtenido buenos resultados en la séptima imagen. En la subfigura b de las correspondencias se encuentra la explicación a este fenómeno, en ninguno de los casos se encuentran coincidencias entre los puntos de interés de la parte frontal del dibujo (el "pico" o triángulo delantero) entre modelo e imagen transformada.

Estos resultados confirman las invariantias descritas en la sección 1.3 y muestran la robustez de los descriptores frente a pequeñas transformaciones geométricas. Se pueden destacar SIFT, ORB, BRISK y AKAZE como los métodos que encuentran un mayor número de correspondencias entre modelo y transformación, pero las transformaciones obtenidas son igual de buenas para todos los descriptores.

2.3. Dataset 2

En este caso las imágenes con las que se trabaja son 20 copias manuscritas del modelo original (directorio dataset2_reyCopia/reyCopia/), en algunos casos también sometidas a una rotación de 90 o -90º. Puesto que el objetivo es el mismo que el del apartado anterior, se ha utilizado la misma solución, permitiendo de esta forma comparar los resultados obtenidos en ambas.

2.3.1. Resultados experimentales

Al tratarse de un dataset de mayor tamaño que el primero, no se muestran las transformaciones obtenidas para todas las imágenes, solo las necesarias para poder realizar la evaluación de cada uno de los descriptores de forma representativa. De la misma forma, en cuanto a las figuras de las correspondencias de puntos, se presentan una con la que se obtienen buenos resultados y otra con la que los resultados son peores.

1. SIFT

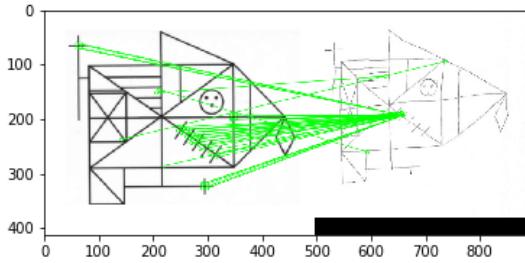


Figura 21: Correspondencias entre puntos de interés (SIFT)

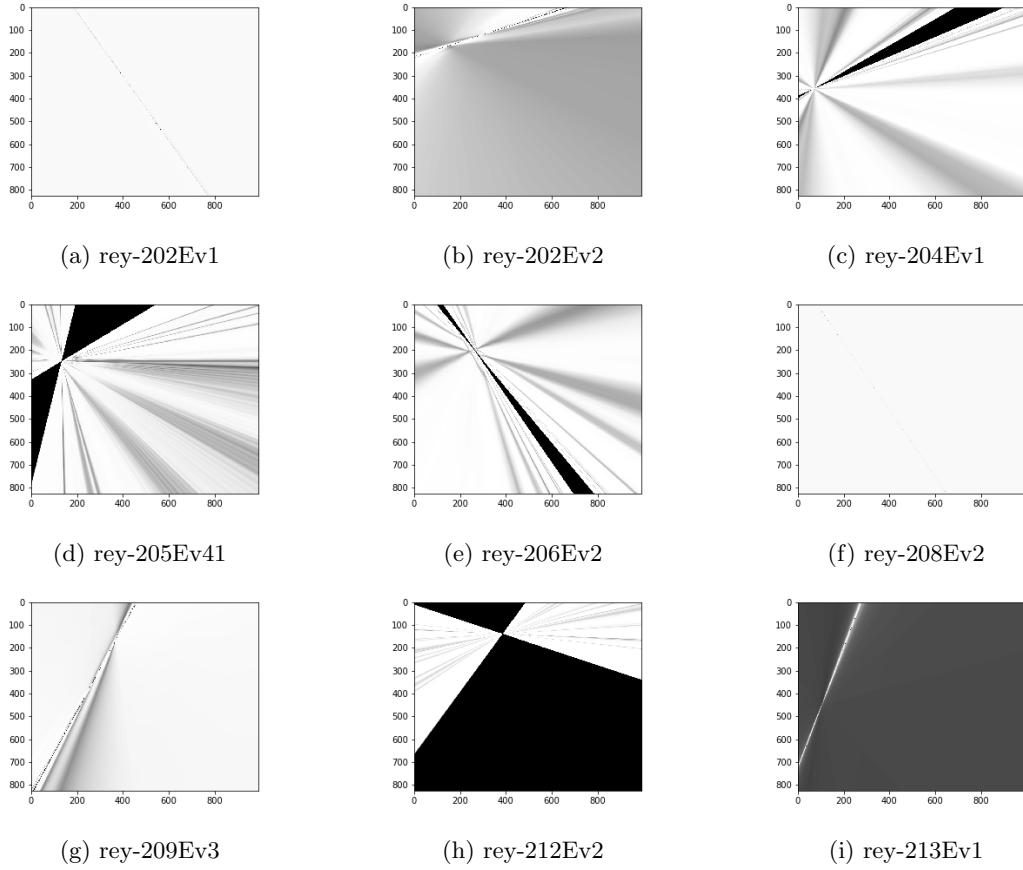


Figura 22: Transformaciones obtenidas con el descriptor SIFT

2. SURF

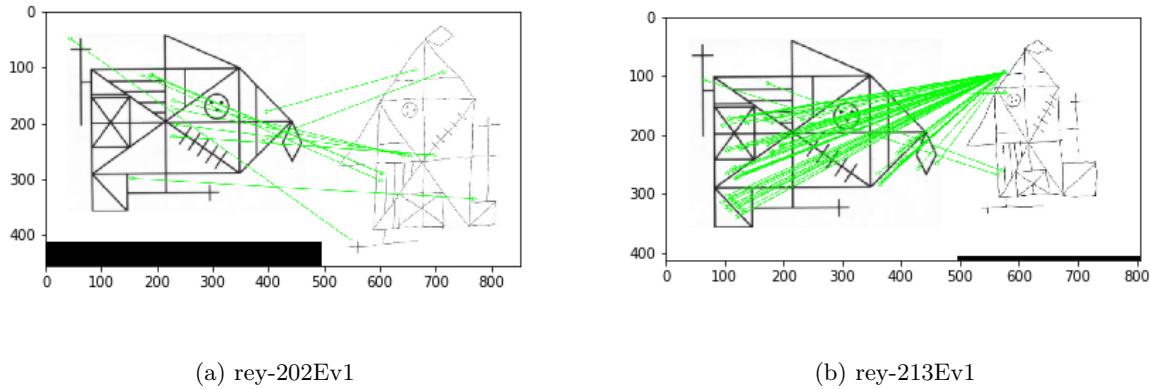


Figura 23: Correspondencias entre puntos de interés (SURF)

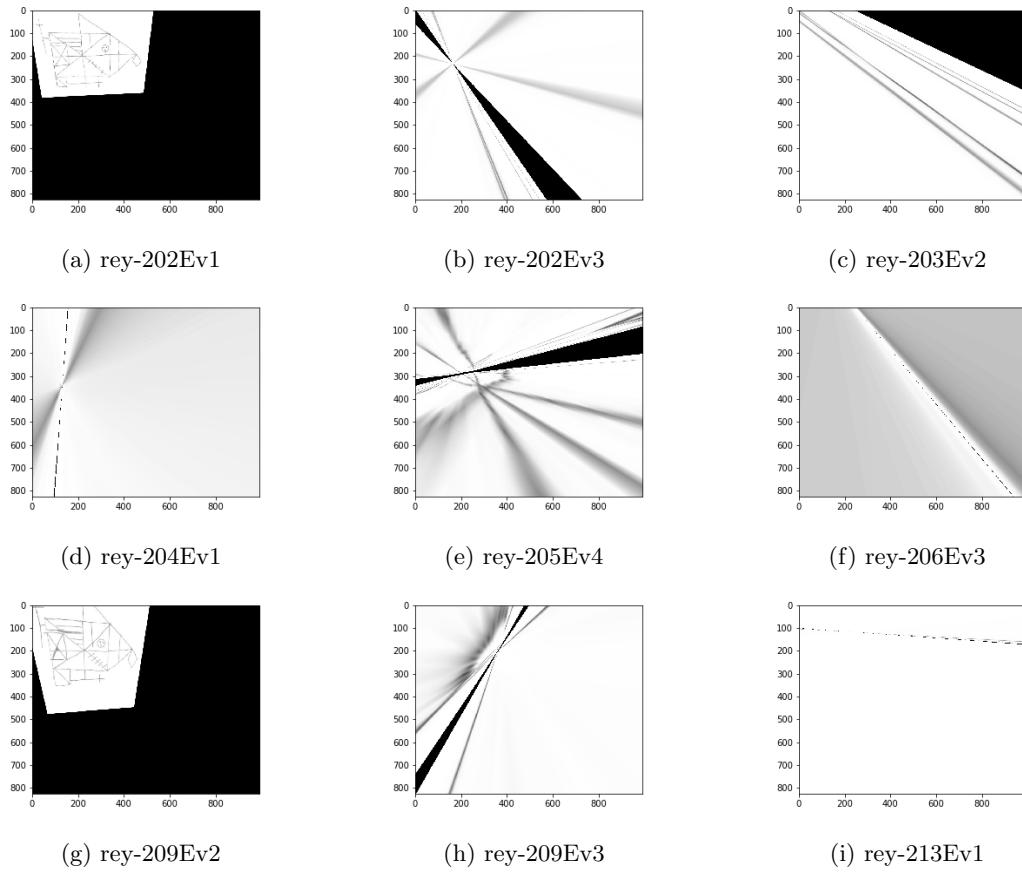
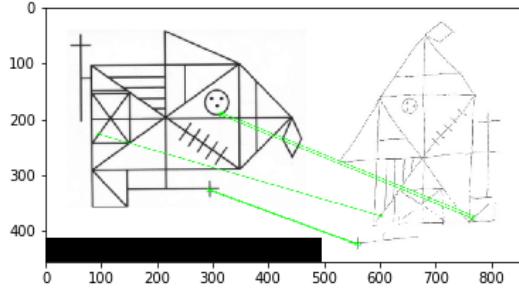
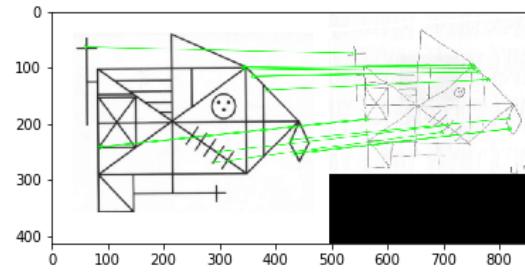


Figura 24: Transformaciones obtenidas con el descriptor SURF

3. BRIEF

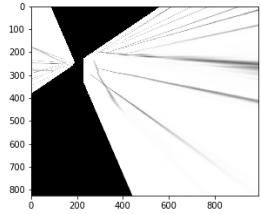


(a) rey-202Ev1

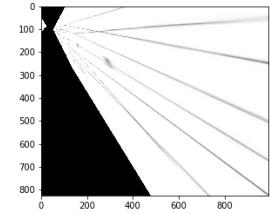


(b) rey-208Ev2

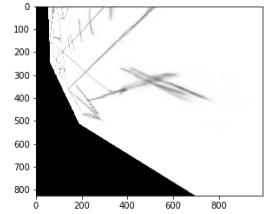
Figura 25: Correspondencias entre puntos de interés (BRIEF)



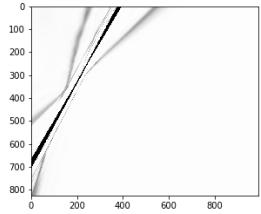
(a) rey-202Ev1



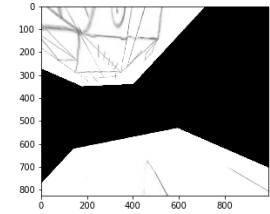
(b) rey-202Ev5



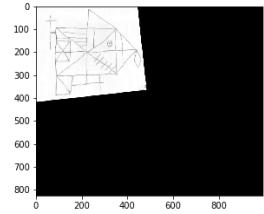
(c) rey-204Ev1



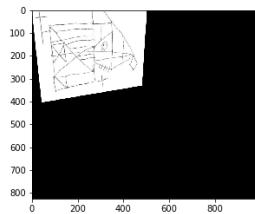
(d) rey-205Ev41



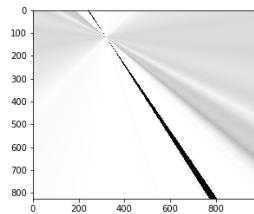
(e) rey-206Ev2



(f) rey-208Ev2



(g) rey-209Ev3



(h) rey-212Ev2

Figura 26: Transformaciones obtenidas con el descriptor BRIEF

4. ORB

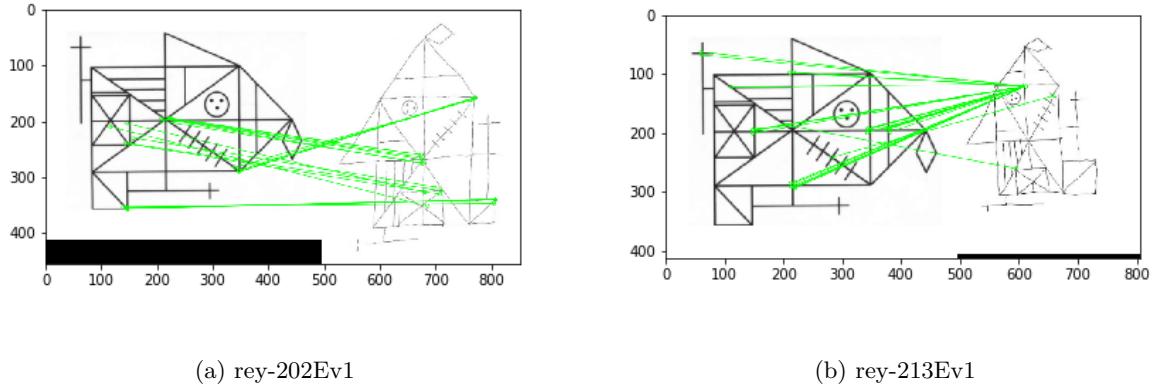


Figura 27: Correspondencias entre puntos de interés (ORB)

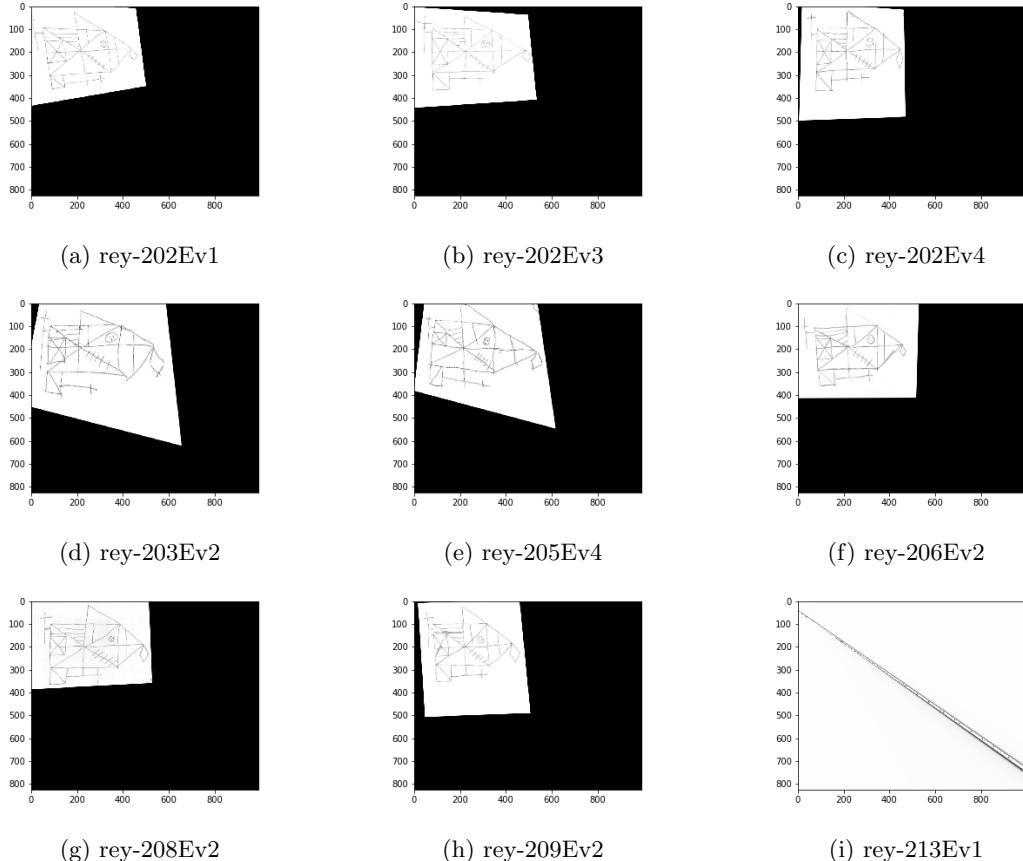


Figura 28: Transformaciones obtenidas con el descriptor ORB

5. BRISK

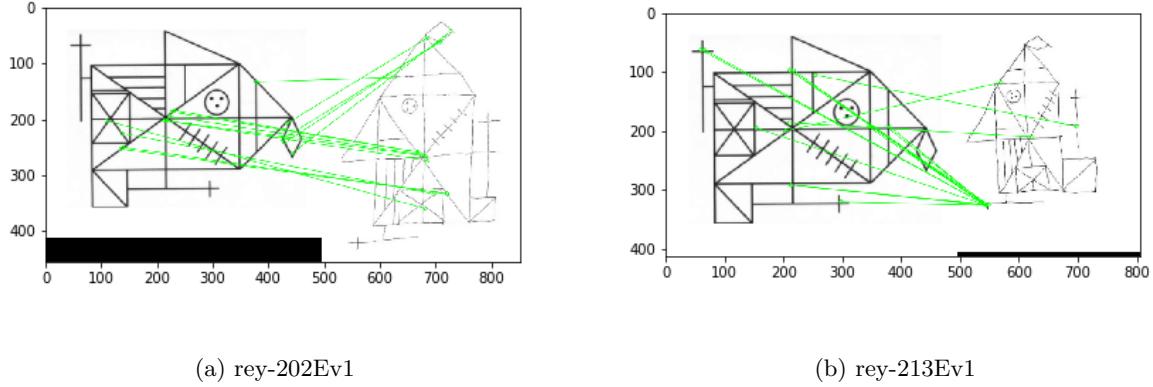


Figura 29: Correspondencias entre puntos de interés (BRISK)

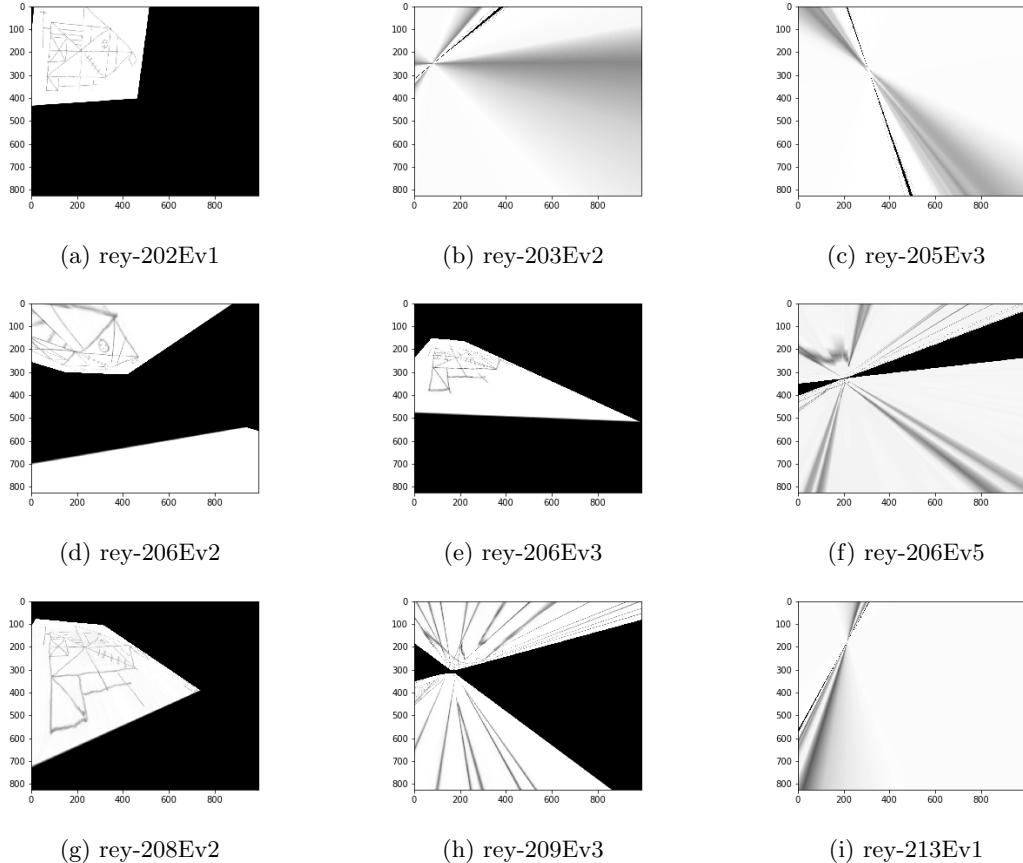


Figura 30: Transformaciones obtenidas con el descriptor BRISK

6. AKAZE

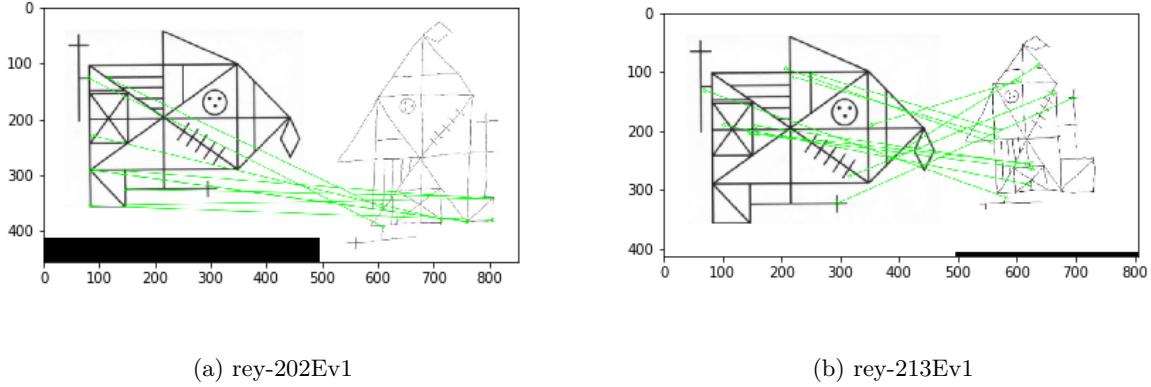


Figura 31: Correspondencias entre puntos de interés (AKAZE)

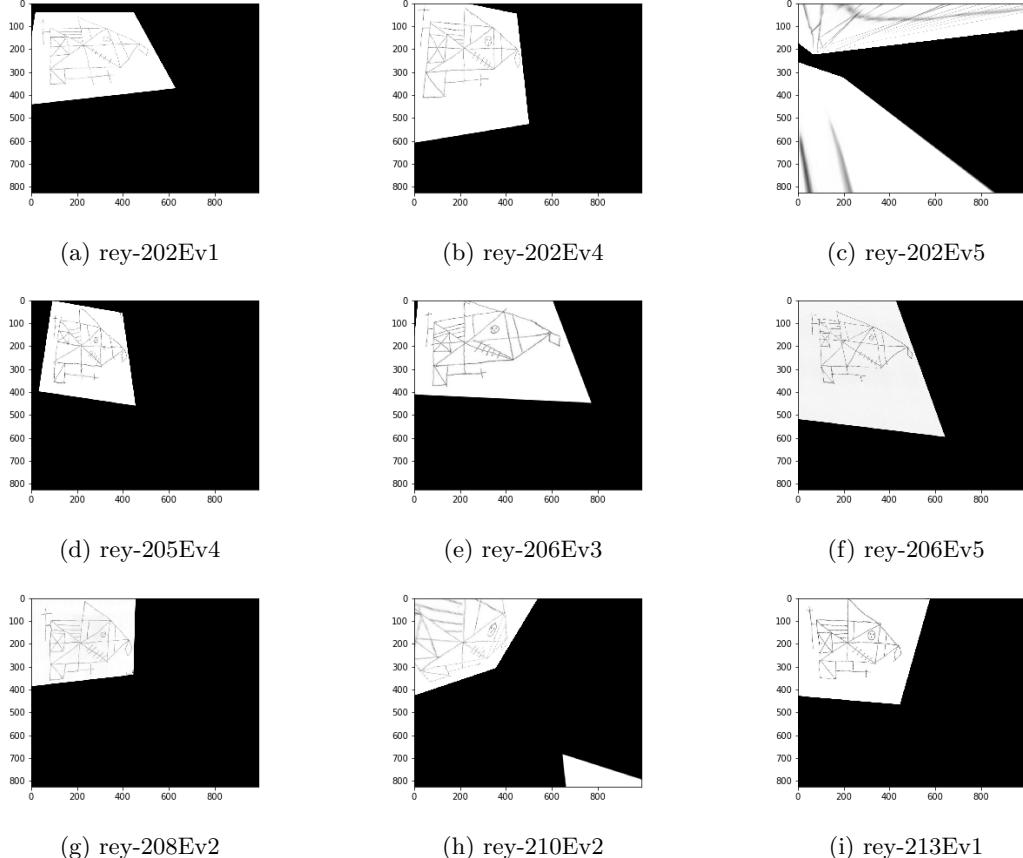


Figura 32: Transformaciones obtenidas con el descriptor AKAZE

2.3.2. Discusión y conclusiones

Teniendo en cuenta el tipo de imágenes y la evaluación realizada al primer conjunto de datos, cabía esperar un empeoramiento notable en los resultados obtenidos. La primera diferencia a tener en cuenta es la cantidad de correspondencias entre los puntos de interés encontradas, mucho menor para este segundo conjunto de datos, así lo enseñan las figuras 21, 23, 25, 27, 29, 31. Por otro lado, la calidad de estas correspondencias también decrece, entendiendo que una correspondencia es mejor cuando los puntos de interés en cuestión pertenecen efectivamente a la misma zona del dibujo. Estos detalles explican los malos resultados obtenidos con los descriptores, en las figuras 22, 24, 26, 28, 30, 32 se observan imágenes transformadas en las que apenas puede distinguirse el modelo original. También se debe considerar que no se han extraído suficientes correspondencias para calcular la matriz de homografía de todas las imágenes con los descriptores BRIEF(10/20), BRISK(19/20) y AKAZE(17/20).

En cuanto a calidad de las transformaciones, los métodos con resultados más prometedores son ORB y AKAZE, consiguiendo en algunos casos recuperar el modelo de la copia.

2.4. Dataset 3

El conjunto de imágenes a tratar está formado por copias dibujadas a mano de 9 modelos distintos, realizadas por individuos con diferentes niveles de deterioro cognitivo. Se dispone de 611 imágenes de individuos categorizados como "sanos", 461 imágenes de la categoría de "enfermos estables" y 320 imágenes de "enfermos evolución". Para abordar este problema se deben extraer ciertas características de las imágenes que sean representativas y entrenar un clasificador de *machine learning* para que sea capaz de predecir el grado de deterioro cognitivo del individuo según los dibujos.

Puesto que la implementación debe ser sencilla y tras comentarlo con algunos compañeros, se ha empleado la siguiente estructura:

- Clasificador: algoritmo *K-Nearest Neighbors* (librería Scikit-learn). Este método de clasificación estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento pertenezca a cierta clase a partir de la información proporcionada por el conjunto de prototipos. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras. La implementación de este algoritmo está basada en [9].
- Clases: "Sano", "Enfermo_DCL_estable" y "Enfermo_DCL_evolución"

Se han realizado dos implementaciones, la primera utilizando el número de contornos, área y perímetro de los mismos como características y la segunda detectando las regiones de interés y empleando el número de las mismas y el área y perímetro de la de mayor tamaño. Los resultados con ambas aproximaciones son muy similares y se resumen en las tablas 4 y 7 respectivamente. A mayores se ha probado calculando los momentos centrales, pero los resultados obtenidos son peores.

A la hora de configurar el clasificador debe especificarse el número de vecinos que utilizará el algoritmo para predecir los resultados. Se han realizado varias ejecuciones con diferente número de vecinos para determinar este parámetro K. Los distintos valores de exactitud media(*accuracy*) conseguidos para 10 ejecuciones y de 1 a 25 vecinos se representan en las gráficas 34 y 36.

2.4.1. Contornos

La extracción de los contornos utilizados para la definición de las características de las imágenes se realiza en base al código presentado en [10], utilizando el detector de bordes Canny. Como ejemplo de la eficacia de este método se muestran, para el modelo "casa", en la Figura 33 los contornos encontrados en distintas imágenes para cada nivel de deterioro cognitivo.

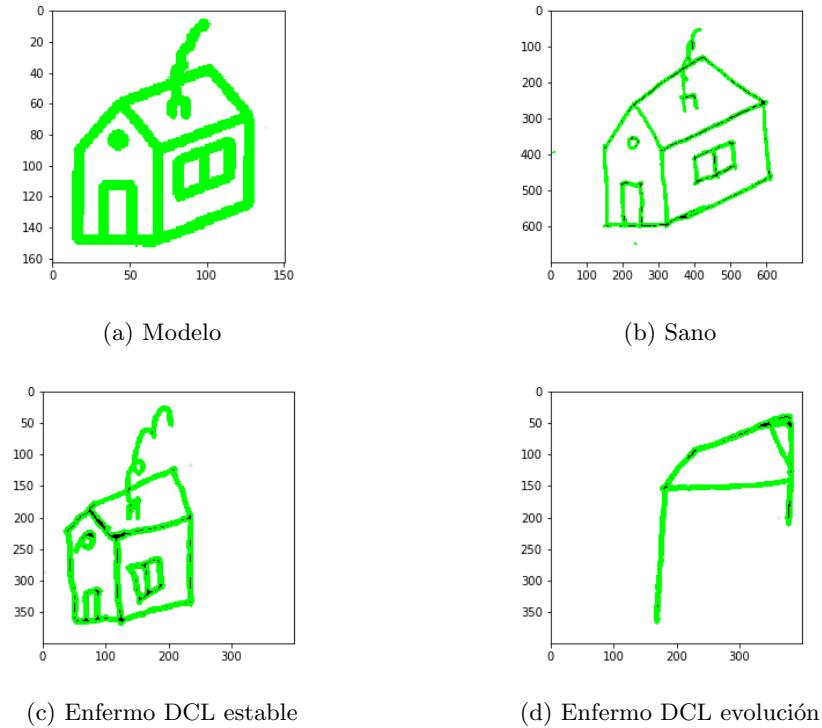


Figura 33: Contornos

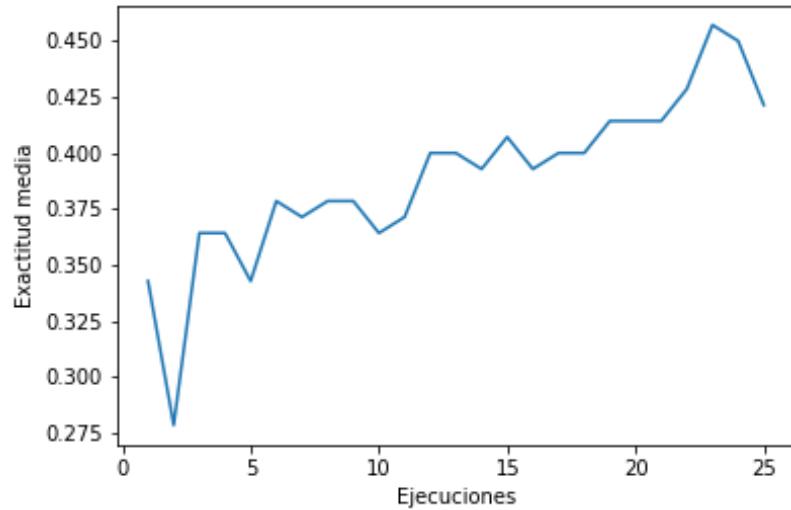


Figura 34: Elección del parámetro K

Clase	Precisión	Exhaustividad	Valor-F	Número de muestras
enfermos_DCL_estables	0.36	0.40	0.38	10
enfermos_DCL_evolucion	0.50	0.25	0.33	4
sanos	0.53	0.56	0.55	16
Exactitud			0.47	30
Media	0.46	0.40	0.42	30
Media ponderada	0.47	0.47	0.46	30

Tabla 4: Informe de clasificación para 3 clases (Características extraídas de los contornos)

Clase	Precisión	Exhaustividad	Valor-F	Número de muestras
enfermos	0.50	0.57	0.53	14
sanos	0.57	0.50	0.53	16
Exactitud			0.53	30
Media	0.54	0.54	0.53	30
Media ponderada	0.54	0.53	0.53	30

Tabla 5: Informe de clasificación para 2 clases (Características extraídas de los contornos)

Modelo	Número de clases	Exactitud
Bucle	2	0.51
	3	0.39
Casa	2	0.63
	3	0.43
Círculo	2	0.67
	3	0.54
Cruz	2	0.56
	3	0.53
Cuadrado	2	0.57
	3	0.51
Cubo	2	0.43
	3	0.36
Minimental	2	0.51
	3	0.41
Picos mesetas	2	0.52
	3	0.48
Triángulo	2	0.53
	3	0.46

Tabla 6: Exactitud obtenida para los distintos modelos con 2 o 3 clases

2.4.2. Regiones de interés

Este método está basado en el cálculo de las regiones de interés presentes en cada imagen, para ello se han realizado ciertas modificaciones sobre la implementación descrita en [11].

En este caso, en la figura 35, se muestra el modelo "bucle" con la *bounding box* que representa la región de interés de mayor área en rojo y el resto de regiones de interés encontradas en verde, para los distintos grados de deterioro cognitivo.

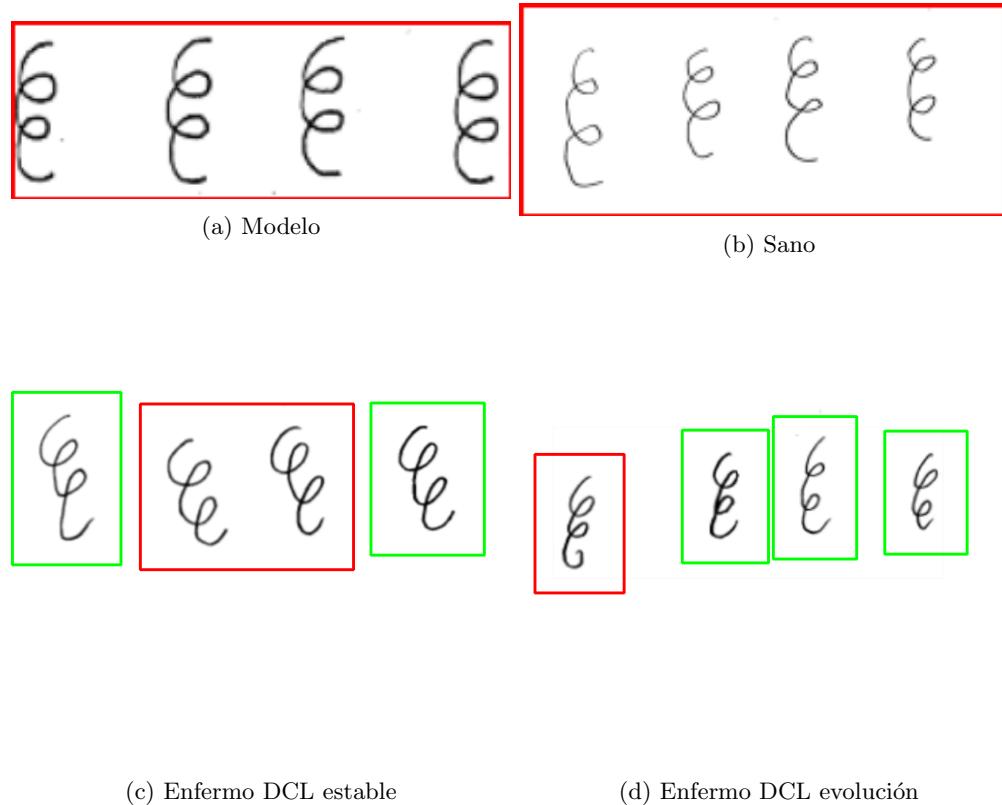


Figura 35: Regiones de interés

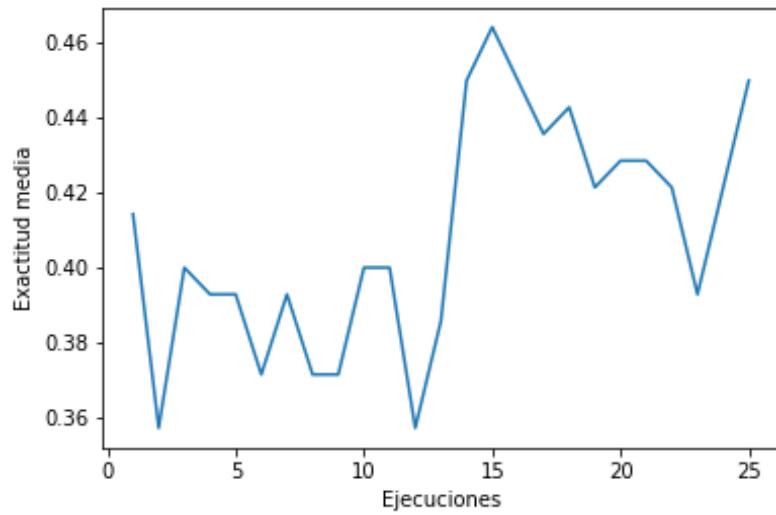


Figura 36: Elección del parámetro K

Clase	Precisión	Exhaustividad	Valor-F	Número de muestras
enfermos_DCL_estables	0.45	0.42	0.43	12
enfermos_DCL_evolucion	1.00	0.11	0.20	9
sanos	0.48	0.83	0.61	12
Exactitud			0.48	33
Media	0.64	0.45	0.41	33
Media ponderada	0.61	0.48	0.43	33

Tabla 7: Informe de clasificación para 3 clases (Características extraídas de las regiones de interés)

Clase	Precisión	Exhaustividad	Valor-F	Número de muestras
enfermos	0.63	0.89	0.74	19
sanos	0.67	0.29	0.40	14
Exactitud			0.64	33
Media	0.65	0.59	0.57	33
Media ponderada	0.65	0.64	0.60	33

Tabla 8: Informe de clasificación para 2 clases (Características extraídas de las regiones de interés)

Modelo	Número de clases	Exactitud
Bucle	2	0.54
	3	0.33
Casa	2	0.43
	3	0.3
Círculo	2	0.51
	3	0.27
Cruz	2	0.72
	3	0.34
Cuadrado	2	0.54
	3	0.39
Cubo	2	0.37
	3	0.53
Minimental	2	0.74
	3	0.41
Picos mesetas	2	0.64
	3	0.45
Triángulo	2	0.56
	3	0.41

Tabla 9: Exactitud obtenida para los distintos modelos con 2 o 3 clases

2.4.3. Discusión y conclusiones

Puesto que la tarea a resolver es clasificar las imágenes según el grado de deterioro cognitivo, en una primera aproximación se planteó no diferenciar entre los distintos modelos, es decir, evaluar la solución implementada sobre el conjunto completo de imágenes. Esta idea se desechó en base a que las características planteadas no son comunes entre los modelos y, por tanto, se obtendrían peores resultados en la clasificación, rozando la aleatoriedad. Finalmente, las imágenes utilizadas son las copias de los modelos "casa" (68 imágenes de la clase "sanos", 50 de "enfermos estables" y 31 de "enfermos evolución"), para la versión con contornos y "bucle" (68, 52 y 42 imágenes para cada una de las clases, respectivamente), para la aproximación con regiones de interés.

En primer lugar, cabe destacar que el algoritmo KNN no aumenta su eficacia al aumentar el número de vecinos utilizados para la clasificación. Como se observa en las gráficas 34 y 36 los resultados, en cuanto a exactitud, mejoran para ambos casos a partir de aproximadamente 12 vecinos. Estos valores son variables dependiendo de la implementación (datos de entrada, tipo de clasificación, modelado de características, etc.). La evaluación se ha realizado utilizando las siguientes métricas:

- Precisión: fracción de instancias relevantes entre las instancias recuperadas.
- Exhaustividad: fracción de la cantidad total de instancias relevantes que han sido recuperadas.
- Valor-F: se emplea para determinar un valor único ponderado de la precisión y la exhaustividad y se define como

$$F_1 = 2 \cdot \frac{\text{Precisión} \cdot \text{Exhaustividad}}{\text{Precisión} + \text{Exhaustividad}}$$

Se han llevado a cabo dos experimentos sobre ambos modelos, variando el número de etiquetas utilizadas en la clasificación. Por un lado, se divide el conjunto de datos en "enfermos estables", "enfermos evolución" y "sanos". A la vista de las tablas 4 y 7 se puede apreciar que los resultados se encuentran lejos de los deseables para un sistema real, obteniendo valores por debajo del 50 % para cada una de las métricas calculadas.

La división del conjunto de datos de entrenamiento y validación se lleva a cabo de forma automática (80-20 %), por lo que el número de muestras de cada una de las clases varía.

En el segundo experimento, se unen las imágenes de las clases "enfermos estables", "enfermos evolución" en una sola, etiquetada como "enfermos". Los resultados de esta evaluación, presentes en las tablas 5 y 8, son considerablemente mejores que los obtenidos anteriormente, alcanzando aproximadamente el 60% de exactitud en la clasificación. Esta mejora se debe a que las diferencias entre las características extraídas en ambas clases son mayores y el aumento de muestras en la clase "enfermos" que permite un aprendizaje más eficiente durante el entrenamiento. También se debe considerar que la reducción de etiquetas disminuye la complejidad de la solución propuesta, puesto que el modelo debe escoger solo entre dos opciones durante la clasificación.

En general, se han obtenido mejores resultados con el modelo basado en las regiones de interés, sin embargo las diferencias entre este y el modelo basado en los contornos son mínimas. Esto puede ser debido a que el cálculo de las *bounding boxes* que enmarcan las regiones de interés se hace a partir de los contornos extraídos de las imágenes y, por tanto, ambas implementaciones están íntimamente ligadas.

A modo de resumen, se puede concluir que las características escogidas se adaptan a la tarea propuesta, permitiendo al clasificador diferenciar entre los distintos tipos de deterioro cognitivo en prácticamente la mitad de los casos, pero la eficacia de este sistema es considerablemente baja. Con el fin de mejorarla, se debería aumentar el número de características representativas de cada imagen, de forma que el modelo pudiese discriminar mejor las diferentes clases y afinar la detección y el cálculo de las mismas, quedando esto ya fuera del alcance de este trabajo.

Referencias

- [1] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.
- [2] D. Viswanathan, “Features from accelerated segment test (fast),” 2011.
- [3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” vol. 6314, 09 2010, pp. 778–792.
- [4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2564–2571.
- [5] S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2548–2555.
- [6] P. Fernández Alcantarilla, “Fast explicit diffusion for accelerated features in nonlinear scale spaces,” 09 2013.
- [7] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, “Multi-probe lsh: Efficient indexing for high-dimensional similarity search .” 01 2007, pp. 950–961.
- [8] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, p. 509–517, Sep. 1975. [Online]. Available: <https://doi.org/10.1145/361002.361007>
- [9] A. Navlani, “Knn classification using scikit-learn,” 2018. [Online]. Available: <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
- [10] S. Sinha, “Find and draw contours using opencv python.” [Online]. Available: <https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/>
- [11] “Extract roi from image with pyhton and opencv.” [Online]. Available: <https://cvisiondemy.com/extract-roi-from-image-with-python-and-opencv/>