



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



## TFG del Grado en Ingeniería Informática

**Transformaciones invariantes a  
la iluminación para mejora de  
la segmentación en imágenes.  
Aplicación a piezas metálicas.**



Presentado por Jonás Martínez Sanllorente  
en Universidad de Burgos — 30 de junio  
de 2024

Tutor: Dr. Carlos López Nozal  
y Dr. Pedro Latorre Carmona







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Carlos López Nozal y D. Pedro Latorre Carmona, profesores del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Exponen:

Que el alumno D. Jonás Martínez Sanllorente, con DNI 71363040C, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado “Aplicación de transformaciones invariantes a la iluminación para mejora de la segmentación en imágenes. Aplicación a piezas metálicas”.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 30 de junio de 2024

Vº. Bº. del Tutor:

D. Carlos López Nozal

Vº. Bº. del co-tutor:

D. Pedro Latorre Carmona





## Resumen

El proceso de análisis visual piezas metálicas en imágenes se ve influido por diversos factores. Algunos de los más importantes están directamente relacionados con las condiciones de iluminación de la escena. En concreto, la forma como se adquieren las imágenes de las propias piezas metálicas. Dado que los materiales metálicos son altamente reflectantes, se producen tanto sombras como reflexiones de tipo especular [1] que degradan sustancialmente la propia calidad visual de la imagen.

En el caso se quisieran aplicar métodos de segmentación de imágenes sobre ellas, se producirían errores en la generación de las diferentes regiones a identificar en las mismas. Es, por tanto, necesario aplicar algún tipo de pre-procesamiento que mitigue dichas fuentes de error, en la medida de lo posible.

En este trabajo se propone analizar el rendimiento de una serie de técnicas de transformación sobre las imágenes en color adquiridas denominadas *transformaciones invariantes de la iluminación*, aplicadas en forma de técnica(s) de pre-procesamiento. Por otro lado, se creará una aplicación / aplicación web para que el usuario pueda aplicar dichas transformaciones, así como métodos de segmentación (agrupamiento o *clustering*) de una forma intuitiva y rápida.

## Descriptores

Piezas metálicas, reflexión metálica, identificación de piezas, métodos invariantes, segmentación de imágenes, aplicación.

## Abstract

Reflections on metal parts and the shadows that can be caused are a very common problem when it comes to correctly identifying the limits of a part.

In most cases, this means that after applying segmentation methods, erroneous identification of the parts is produced due to the fact that obtaining images in ideal conditions where neither reflections nor shadows are found is very complicated and trying to replicate a scenario of this kind can be very costly.

This is why by applying pre-processing methods to the images taken of the pieces, it is possible to obtain a result that has these reflections, shadows and other image defects mitigated so that when the image is subsequently segmented, the result will be of a higher quality.

This work proposes the demonstration of a better result when segmenting after having applied invariant methods to the initial image.

In addition, an application so that the user, can apply the different algorithms in an intuitive, fast and comfortable way.

## Keywords

Metallic parts, metallic reflection, part identification, invariant methods, image segmentation, application

---

# Índice general

---

<b>Índice general</b>	<b>iii</b>
<b>Índice de figuras</b>	<b>v</b>
<b>Índice de tablas</b>	<b>vii</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Estructura de la memoria . . . . .	2
1.2. Materiales adjuntos . . . . .	3
<b>2. Objetivos del proyecto</b>	<b>5</b>
2.1. Objetivos generales . . . . .	5
2.2. Objetivos técnicos . . . . .	5
2.3. Objetivos personales . . . . .	6
<b>3. Conceptos teóricos</b>	<b>7</b>
3.1. Preprocesado de imágenes . . . . .	7
3.2. Métodos de transformación invariante . . . . .	8
3.3. Métodos de agrupamiento de imágenes . . . . .	17
3.4. Clasificación de resultados . . . . .	24
<b>4. Técnicas y herramientas</b>	<b>27</b>
4.1. Control de versiones . . . . .	27
4.2. Hosting del repositorio . . . . .	27
4.3. Metodologías ágiles . . . . .	28
4.4. Gestión del proyecto . . . . .	28
4.5. Comunicación . . . . .	29

4.6. Entorno de desarrollo integrado (IDE) . . . . .	30
4.7. Documentación . . . . .	31
4.8. Librerías . . . . .	31
<b>5. Aspectos relevantes del desarrollo del proyecto</b> . . . . .	<b>33</b>
5.1. Inicio del proyecto . . . . .	33
5.2. Formación . . . . .	34
5.3. Metodologías . . . . .	34
5.4. Cambio de lenguaje y entorno de programación . . . . .	35
5.5. Desarrollo de la app . . . . .	35
5.6. Resultados de la experimentación . . . . .	42
<b>6. Trabajos relacionados</b> . . . . .	<b>53</b>
6.1. Artículos científicos . . . . .	53
6.2. Fortalezas y debilidades del proyecto . . . . .	55
<b>7. Conclusiones y Líneas de trabajo futuras</b> . . . . .	<b>59</b>
7.1. Conclusiones . . . . .	59
7.2. Líneas de trabajo futuras . . . . .	60
<b>Bibliografía</b> . . . . .	<b>63</b>

---

# Índice de figuras

---

3.1. Resultado de aplicar Álvarez . . . . .	9
3.2. Resultado de aplicar Maddern . . . . .	12
3.3. Resultado de aplicar Krajník . . . . .	14
3.4. Resultado de aplicar Upcroft . . . . .	15
3.5. Resultado de aplicar PCA . . . . .	17
3.6. Resultado de crear la imagen ‘ground truth’ . . . . .	24
5.1. Apartado de bienvenida de InvIMP. . . . .	37
5.2. Apartado de exploración de algoritmos de InvIMP. . . . .	38
5.3. Apartado de histórico de exploraciones de InvIMP. . . . .	41
5.4. Apartado de ayuda de InvIMP. . . . .	42
5.5. Tasa de acierto de los de la gemación aplicando K-Means sobre la imagen original frente a los resultados de utilizar previamente Álvarez. . . . .	44
5.6. Tasa de acierto de los de la gemación aplicando K-Means sobre la imagen original frente a los resultados de utilizar los distintos métodos invariantes. . . . .	45
5.7. Imagen piezas metálicas en condiciones idóneas. . . . .	46
5.8. Imagen ground truth de piezas metálicas en condiciones idóneas. . . . .	46
5.9. Imagen piezas metálicas en condiciones idóneas tras aplicar el algoritmo invariante de Alvarez. . . . .	47
5.10. Comparación de resultados de piezas metálicas en condiciones idóneas. . . . .	47
5.11. Imagen piezas metálicas en condiciones adversas. . . . .	48
5.12. Imagen ground truth de piezas metálicas en condiciones adversas. . . . .	48
5.13. Imagen piezas metálicas en condiciones adversas tras aplicar el algoritmo invariante de Alvarez. . . . .	49

5.14. Comparación de resultados de piezas metálicas en condiciones adversas. . . . .	49
5.15. Imagen piezas metálicas en condiciones de poca luz. . . . .	50
5.16. Imagen ground truth de piezas metálicas en condiciones de poca luz. . . . .	50
5.17. Imagen piezas metálicas en condiciones de poca luz tras aplicar el algoritmo invariante de Alvarez. . . . .	51
5.18. Comparación de resultados de piezas metálicas en condiciones de poca luz. . . . .	51

---

## **Índice de tablas**

---

5.1. Media de porcentajes de acierto. . . . .	43
5.2. Mediana de porcentajes de acierto. . . . .	43



---

# 1. Introducción

---

En el ámbito de la fabricación, las piezas metálicas son fundamentales para una amplia variedad de aplicaciones industriales. La precisión en su fabricación es crucial para garantizar la funcionalidad y durabilidad de los productos finales.

El control de calidad en el proceso de producción es de gran importancia, ya que asegura que los productos finales cumplan con los estándares establecidos y satisfagan las expectativas del cliente. A través de técnicas de inspección y pruebas rigurosas, se pueden identificar y corregir defectos en etapas tempranas del proceso de producción. Esto no solo reduce el riesgo de fallos en el producto terminado, sino que también optimiza el uso de materiales y recursos, incrementando la eficiencia general.[12]

Comúnmente, esta inspección técnica y pruebas suelen ser llevadas a cabo por un operario, ya que los sistemas basados en la toma de imágenes para detectar fallos en el producto dependen en gran medida de la calidad de la fotografía tomada. En la mayoría de las ocasiones, a menos que el fondo sea muy fácilmente distinguible de la pieza y las condiciones de iluminación sean ideales, los métodos de agrupamiento difícilmente pueden identificar la pieza correctamente. Por esta razón, los métodos de transformación invariante desempeñan un papel esencial, ya que permiten compensar y corregir las distorsiones causadas por variaciones en las condiciones de iluminación, como reflejos y sombras los cuales en piezas metálicas se dan con mucha facilidad. [14][12]

Los métodos invariantes son responsables de generar una nueva imagen a partir de la original en la cual las reflexiones de luz, las sombras o errores inducidos por las condiciones lumínicas en el momento de la captura de imágenes estén mitigados. Al aplicar sobre esta nueva imagen algún método

de agrupamiento de imágenes, se obtendrá un resultado mejor, o dicho en otras palabras, más cercano a la realidad.

Un método de agrupamiento de imágenes sirve para segmentarlas en diferentes partes, identificando y clasificando áreas similares dentro de la imagen. Este proceso es crucial para separar la pieza metálica del fondo y otros elementos no deseados, facilitando así una inspección técnica más precisa.<sup>[3]</sup>

La implementación de estos métodos de manera previa a la segmentación de imágenes no solo mejora la precisión, sino que también reduce la variabilidad que se da por diferencias en la iluminación del entorno. Esto es particularmente importante en entornos de producción industrial, donde las condiciones de iluminación pueden no ser siempre ideales y variar significativamente, afectando la visibilidad y el contraste en las imágenes capturadas.<sup>[4]</sup>

En resumen, el uso de métodos de transformación invariante en la segmentación de imágenes de piezas metálicas tiene un papel muy importante a la hora de contrarrestar los efectos adversos de las condiciones de iluminación. Estas técnicas no solo mejoran considerablemente la precisión de los resultados y la consistencia en la correcta identificación de las piezas metálicas en imágenes, sino que permiten automatizar el control de calidad dentro de la propia la industria, optimizando así los procesos de producción.  
<sup>[12]</sup>

## 1.1. Estructura de la memoria

La memoria sigue la siguiente estructura:

- **Introducción:** breve descripción del problema a resolver y la solución propuesta. Estructura de la memoria y listado de materiales adjuntos.
- **Objetivos del proyecto:** exposición de los objetivos que persigue el proyecto.
- **Conceptos teóricos:** breve explicación de los conceptos teóricos clave para la comprensión de la solución propuesta.
- **Técnicas y herramientas:** listado de técnicas metodológicas y herramientas utilizadas para gestión y desarrollo del proyecto.
- **Aspectos relevantes del desarrollo:** exposición de aspectos destacables que tuvieron lugar durante la realización del proyecto.
- **Trabajos relacionados:** estado del arte en el campo de la monitorización de la actividad de vuelo de colmenas y proyectos relacionados.

- **Conclusiones y líneas de trabajo futuras:** conclusiones obtenidas tras la realización del proyecto y posibilidades de mejora o expansión de la solución aportada.

Junto a la memoria se proporcionan los siguientes anexos:

- **Plan del proyecto software:** planificación temporal y estudio de viabilidad del proyecto.
- **Especificación de requisitos del software:** se describe la fase de análisis; los objetivos generales, el catálogo de requisitos del sistema y la especificación de requisitos funcionales y no funcionales.
- **Especificación de diseño:** se describe la fase de diseño; el ámbito del software, el diseño de datos, el diseño procedural y el diseño arquitectónico.
- **Manual del programador:** recoge los aspectos más relevantes relacionados con el código fuente (estructura, compilación, instalación, ejecución, pruebas, etc.).
- **Manual de usuario:** guía de usuario para el correcto manejo de la aplicación.

## 1.2. Materiales adjuntos

Los materiales que se adjuntan con la memoria son:

- Aplicación de escritorio para probar tanto con imágenes propias como con algunas de ejemplo.
- Proyecto de MATLAB para poder probar tanto con imágenes propias como con algunas de ejemplo además de tener acceso al código para añadir o modificar los algoritmos que se utilizan.

Además, los siguientes recursos están accesibles a través de internet:

- Repositorio del proyecto. [GitHub](#)



---

## **2. Objetivos del proyecto**

---

### **2.1. Objetivos generales**

El proyecto tiene los siguientes objetivos generales:

- Demostrar que al aplicar métodos de transformación invariante, se consiguen mejores resultados a la hora de identificar piezas metálicas mediante métodos de agrupamiento de imágenes.
- Desarrollar una aplicación de escritorio que permita al usuario identificar la pieza de una imagen que introduzca ofreciendo diferentes algoritmos tanto de transformación invariante (propuestos por Álvarez [2], Maddern [7], Krajník [6], Upcroft [13] y PCA [5]) como de agrupamiento de imágenes (K-Means [10], Fuzzy C-Means [9], GMM [11] y agrupamiento con información espacial [15]).
- Desarrollar una aplicación de escritorio que permita mostrar la mejoría que suponen los métodos de transformación invariante mostrando una comparación de los resultados tanto visual como numérica.
- Guardar las imágenes resultantes del método de transformación invariante y las correspondientes imágenes tras el agrupamiento, tanto de la original como de la invariante.

### **2.2. Objetivos técnicos**

El proyecto tiene los siguientes objetivos técnicos:

- Comparar los resultados de distintos métodos de transformación invariante de imágenes y argumentar la mejora que se obtiene en la identificación de piezas metálicas.

- Utilizar la plataforma GitHub como sistema de control de versiones.
- Utilizar Zube como herramienta de gestión de proyectos.
- Realizar tests para los distintos scripts utilizados.
- Utilizar un sistema de documentación.

### **2.3. Objetivos personales**

El proyecto tiene los siguientes objetivos personales:

- Realizar una aportación a la modernización de la industria.
- Profundizar en el uso de MATLAB, el cual tiene una gran potencia computacional, destacando en cálculos matriciales.
- Explorar métodos de visión artificial.
- La creación de sistemas que permitan la automatización y mejoren el control de calidad en diversos entornos.

---

## 3. Conceptos teóricos

---

Primero, se preprocesa la imagen seleccionada. Luego, se aplica un método de transformación invariante sobre la imagen original. Después, se segmentan tanto la imagen original como la imagen invariante. Finalmente, se clasifican los resultados para su posterior visualización.

A continuación, se explican los conceptos teóricos de cada fase.

### 3.1. Preprocesado de imágenes

Antes de aplicar los algoritmos de transformación invariante hay que realizar una serie de transformaciones a la imagen para poder trabajar correctamente con ella.

#### Eliminación del canal alpha

El canal alpha, que representa la transparencia de los píxeles, no es relevante ya que solo interesa la información de color. Al eliminar este canal, se reduce la complejidad de los datos, en este caso las imágenes, permitiendo centrarse únicamente en los tres canales RGB (rojo, verde y azul).

Para eliminar el canal alpha de una imagen con componentes de color  $R$ ,  $G$ ,  $B$  y  $\alpha$ , tan solo hay que guardar en una nueva imagen lo siguiente:

$$\text{imagen\_sin\_alpha}(i, j) = \begin{bmatrix} R(i, j) \\ G(i, j) \\ B(i, j) \end{bmatrix}$$

donde:

- $R(i, j)$  es el valor del canal rojo en el píxel  $(i, j)$
- $G(i, j)$  es el valor del canal verde en el píxel  $(i, j)$
- $B(i, j)$  es el valor del canal azul en el píxel  $(i, j)$

Se elimina el canal alpha  $\alpha(i, j)$ , que representa la transparencia, ya que no es necesario en este caso.

### 3.2. Métodos de transformación invariante

Los métodos de transformación invariante son distintos algoritmos utilizados para mitigar o incluso eliminar las variaciones en la iluminación en una imagen. Aunque comúnmente suelen ser usados para obtener resultados similares en un mismo escenario pero variando la cantidad de luz ambiental o la cantidad de focos de luz, también son muy útiles para mitigar los efectos de las sombras y reflejos que comúnmente se ocasionan al fotografiar piezas metálicas.

A continuación se presentan cinco métodos distintos de transformación invariante para poder comparar los resultados.

#### Álvarez

La transformación invariante de Álvarez y López [2] es un método para obtener una imagen que sea robusta a variaciones de iluminación, como sombras y cambios en la intensidad de luz. Esto se logra mediante la conversión de los valores de los píxeles al espacio logarítmico-cromático y la proyección sobre una dirección específica.

El funcionamiento básico de la transformación invariante de Álvarez y López es el siguiente:

##### 1. Conversión al espacio logarítmico-cromático

Cada píxel de la imagen  $RGB$  se convierte al espacio logarítmico-cromático utilizando la siguiente fórmula:

$$r = \log \frac{R}{G}, \quad b = \log \frac{B}{G}$$

donde:

- $R$ ,  $G$ , y  $B$  son los valores de los canales de color rojo, verde y azul, respectivamente.

## 2. Proyección en la dirección Invariante

Se determina el ángulo invariante  $\theta$  minimizando la entropía de la imagen proyectada. El valor de  $\theta$  se obtiene resolviendo el siguiente problema de optimización:

$$E_\alpha = - \sum_{i=1}^L H_\alpha(i) \log H_\alpha(i)$$

donde:

- $H_\alpha$  es el histograma de proyección en una dirección de pendiente  $\alpha$ .

Una vez obtenido el ángulo invariante, se proyectan los valores cromáticos en la dirección  $\theta$  para obtener la imagen invariante:

$$I = r \cos \theta + b \sin \theta$$

En la figura 3.1 se puede ver el resultado de aplicar este método invariante.



Figura 3.1: Resultado de aplicar Álvarez

## Maddern

La transformación invariante de Maddern [7] es un método que genera imágenes invariantes a cambios en la iluminación, reduciendo la influencia de sombras y variaciones en la intensidad de la luz. Se basa en un modelo de espectro del sensor de la cámara y la luz incidente.

El funcionamiento básico de la transformación invariante de Maddern es el siguiente:

### 1. Relación entre canales de color

Dada una cámara con sensibilidad espectral  $F(\lambda)$ , el modelo de respuesta de un canal específico  $R$  a la luz de una longitud de onda particular  $\lambda$  se define como:

$$R^{x,E} = a^x \cdot n^x \cdot I^x \int S^x(\lambda) E(\lambda) F(\lambda) d\lambda$$

donde:

- $a^x$  y  $n^x$ : vectores que representan la dirección de la luz y la normal de la superficie.
- $I^x$ : intensidad de la luz en el punto  $x$ .
- $S^x(\lambda)$ : reflectancia espectral de la superficie.
- $E(\lambda)$ : espectro de la fuente de luz.

### 2. Espacio logarítmico-cromático

Se transforman las relaciones de los canales de color al espacio logarítmico-cromático tomando el logaritmo de la respuesta espectral:

$$\log(R^{x,E}) = \log(G^x I^x) + \log S^x(\lambda_i) + \log E^x(\lambda_i)$$

donde:

- $G^x = a^x \cdot n^x$  representa la geometría entre la luz y la escena.

### 3. Proyección invariante

Se proyectan las respuestas espetrales en una dirección específica que elimina el componente de iluminación. La proyección final en la dirección  $\alpha$  se realiza utilizando:

$$I = \log(R_2) - \alpha \log(R_1) - (1 - \alpha) \log(R_3)$$

donde:

- $R_1, R_2, R_3$ : respuestas espetrales de tres canales de color (e.g.  $R, G, B$ ).
- $\alpha$ : parámetro determinado por las sensibilidades espetrales de los canales.

$$I = \log(R_2) - \alpha \log(R_1) - (1 - \alpha) \log(R_3)$$

### 4. Cálculo del parámetro $\alpha$

El parámetro  $\alpha$  se calcula para garantizar que la proyección sea independiente de la temperatura de color  $T$  de la luz:

$$\frac{1}{\lambda_2} = \frac{\alpha}{\lambda_1} + \frac{1 - \alpha}{\lambda_3}$$

### 5. Transformación invariante de Maddern

Finalmente, se aplica la siguiente transformación para obtener una imagen invariante:

$$I_{Maddern} = 0,5 + \log(G) - \alpha \log(B) - (1 - \alpha) \log(R)$$

En la figura 3.2 se puede ver el resultado de aplicar este método invariante.

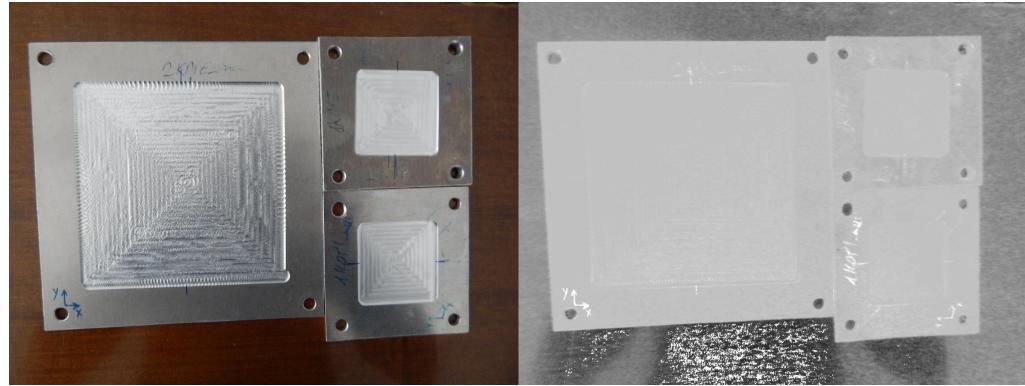


Figura 3.2: Resultado de aplicar Maddern

## Krajiník

La transformación invariante de Krajiník [6] es un método para obtener imágenes invariantes a cambios de iluminación. Se basa en el uso de imágenes intrínsecas, que separan las propiedades intrínsecas de la superficie (reflectancia) de las propiedades extrínsecas (iluminación). Este enfoque permite reducir la influencia de sombras y variaciones de iluminación en la imagen.

El funcionamiento básico de la transformación invariante de Krajiník es el siguiente:

### 1. Modelo Espectral *RGB*

La respuesta espectral de un canal específico  $\rho_k$  (con  $k = \{r, g, b\}$ ) se define como:

$$\rho_k = \sigma \int E(\lambda) S(\lambda) Q_k(\lambda) d\lambda$$

donde:

- $\sigma$ : factor constante que denota el sombreado de Lambert.
- $E(\lambda)$ : iluminación incidente en el elemento de superficie percibida.
- $S(\lambda)$ : reflectancia espectral de la superficie.
- $Q(\lambda)$ : sensibilidad espectral del canal  $k$ .
- $\lambda$ : longitud de onda.

## 2. Transformación Logarítmica

Se transforma el modelo a un espacio logarítmico para hacer que sea invariante a la iluminación. Esto se realiza calculando el logaritmo de la respuesta espectral relativa a un canal de referencia (e.g., azul):

$$\chi_{k,g} = \log \frac{\rho_{r,g}}{\rho_b}$$

## 3. Proyección a un Espacio Invariante

Se proyecta el espacio logarítmico-cromático en una dirección  $\theta$  para obtener la imagen invariante.

$$I_\theta = \chi_r \cos \theta + \chi_g \sin \theta$$

## 4. Minimización de la Entropía

Se determina el ángulo  $\theta$  que minimiza la entropía de la imagen proyectada, para que sea lo más invariante posible a la iluminación.

$$E_\theta = - \sum_i H_\theta(i) \log H_\theta(i)$$

donde:

- $H_\theta(i)$ : histograma proyectado en una dirección  $\theta$ .

En la figura 3.3 se puede ver el resultado de aplicar este método invariante

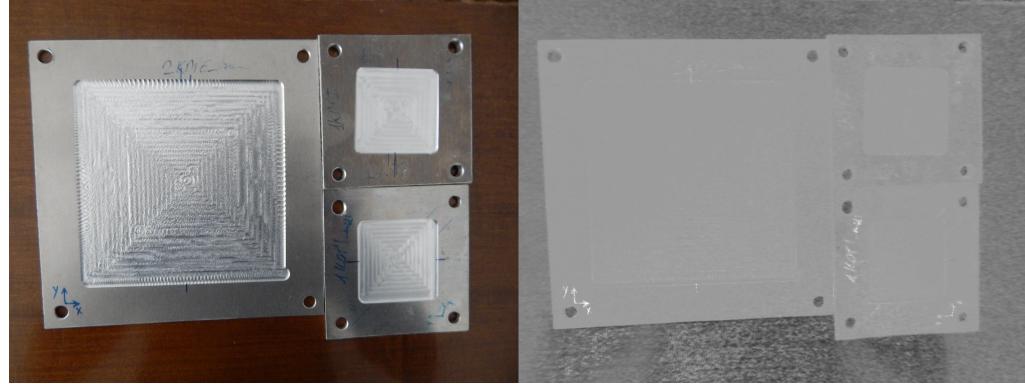


Figura 3.3: Resultado de aplicar Krajník

## Upcroft

La transformación invariante de Upcroft [13] es un método que genera imágenes robustas a variaciones en la iluminación, incluyendo sombras y cambios de intensidad. El método se basa en la relación logarítmica entre los canales de color y se utiliza para mejorar la clasificación de escenas urbanas en entornos cambiantes.

El funcionamiento básico de la transformación invariante de Upcroft es el siguiente:

### 1. Modelo espectral $RGB$

Aunque no aparece explícitamente en el artículo, se menciona que el método de transformación invariante se basa en la relación logarítmica entre los canales  $RGB$  y hace referencia a trabajos anteriores como los de Ratnasingam y Collins. La respuesta espectral de un canal específico  $\rho_k$  (con  $k = \{r, g, b\}$ ) se define como:

$$\rho_k = \sigma \int E(\lambda) S(\lambda) Q_k(\lambda) d\lambda$$

donde:

- $\sigma$ : factor constante que denota el sombreado de Lambert.
- $E(\lambda)$ : iluminación incidente en el elemento de superficie percibida.
- $S(\lambda)$ : reflectancia espectral de la superficie.
- $Q(\lambda)$ : sensibilidad espectral del canal  $k$ .
- $\lambda$ : longitud de onda.

## 2. Transformación logarítmica

La transformación invariante utiliza la relación logarítmica entre los canales *RGB* para minimizar la variabilidad causada por la geometría, la intensidad y el espectro de la fuente de luz.

$$F = \log(G) - \alpha \log(B) - (1 - \alpha) \log(R)$$

## 3. Cálculo del parámetro $\alpha$

El parámetro  $\alpha$  se calcula para minimizar la variabilidad de la fuente de luz. Esto se logra utilizando las longitudes de onda pico de los sensores:

$$\frac{1}{\lambda_G} = \frac{\alpha}{\lambda_B} + \frac{1 - \alpha}{\lambda_R}$$

En la figura 3.4 se puede ver el resultado de aplicar este método invariante.

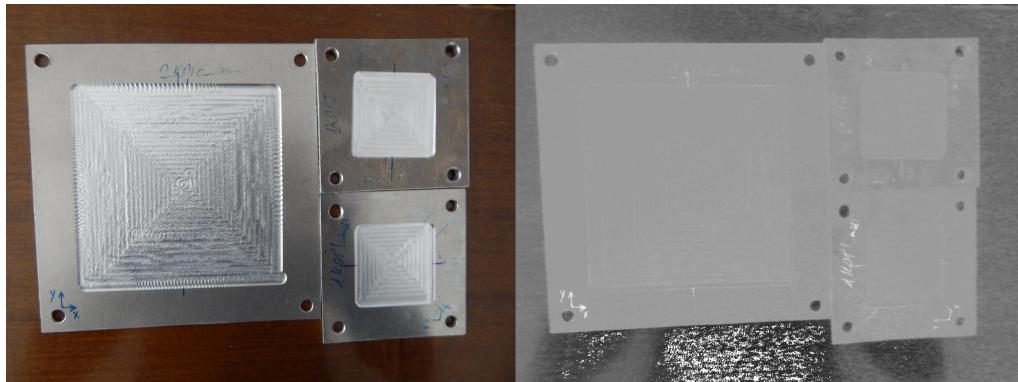


Figura 3.4: Resultado de aplicar Upcroft

## PCA

El método basado en PCA (Análisis de Componentes Principales) [5] propuesto por Kim genera un espacio invariante a la iluminación que permite reducir el efecto de sombras y variaciones en la intensidad de la luz. El algoritmo busca la proyección óptima en el espacio logarítmico-cromático para minimizar la variabilidad causada por la iluminación.

El funcionamiento básico de la transformación invariante PCA es el siguiente:

### 1. Modelo espectral *RGB*

La respuesta espectral de un canal específico  $\rho_k$  (con  $k = \{r, g, b\}$ ) se define como:

$$\rho_k = \sigma \int E(\lambda) S(\lambda) Q_k(\lambda) d\lambda$$

donde:

- $\sigma$ : factor constante que denota el sombreado de Lambert.
- $E(\lambda)$ : iluminación incidente en el elemento de superficie percibida.
- $S(\lambda)$ : reflectancia espectral de la superficie.
- $Q(\lambda)$ : sensibilidad espectral del canal  $k$ .
- $\lambda$ : longitud de onda.

### 2. Transformación logarítmica

El algoritmo convierte el modelo a un espacio logarítmico para minimizar el efecto de la iluminación. Se calculan las siguientes relaciones logarítmicas:

$$r = \log \frac{R}{(RGB)^{1/3}}, \quad b = \log \frac{B}{(RGB)^{1/3}}$$

donde:

- $(RGB)^{1/3}$ : media geométrica de los tres canales.

### 3. Proyección basada en PCA

Se proyectan los valores en un espacio logarítmico-cromático, donde se encuentra la dirección óptima utilizando PCA:

$$I_\theta = r \cos \theta + b \sin \theta$$

Esto se logra aplicando el Análisis de Componentes Principales a los valores calculados.

En la figura 3.5 se puede ver el resultado de aplicar este método invariante.

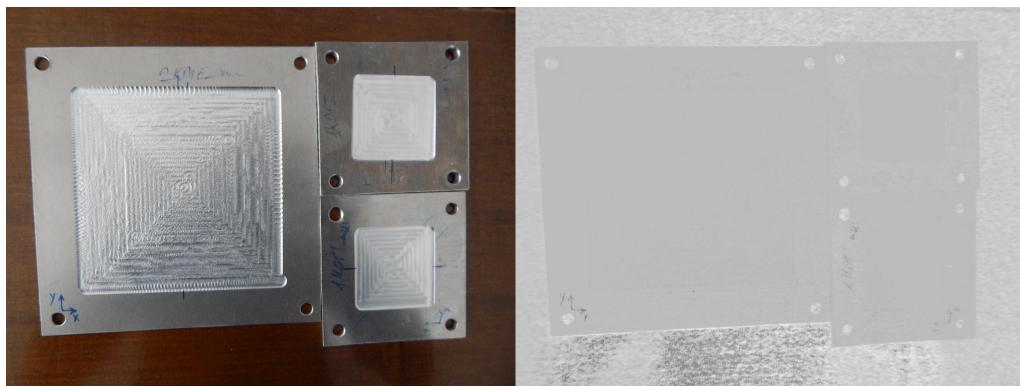


Figura 3.5: Resultado de aplicar PCA

### 3.3. Métodos de agrupamiento de imágenes

Los métodos de agrupamiento son técnicas de aprendizaje no supervisado que se utilizan para dividir un conjunto de datos en grupos o clústeres, de manera que los datos en el mismo clúster tengan una mayor similitud entre sí que con los datos de otros clústeres. Estos métodos se utilizarán para separar la pieza metálica del fondo y determinar si una transformación invariante previa supone algún cambio significativo.

#### K-Means

K-Means [10] es un algoritmo de clustering (agrupamiento) utilizado en análisis de datos y aprendizaje automático para dividir un conjunto de datos en un número específico de grupos (o clusters). El objetivo del algoritmo es asignar cada valor de los datos a un grupo de tal manera que la suma de las distancias cuadradas entre los valores de los datos y el centroide de su grupo (el punto medio de todos los valores en ese grupo) sea minimizada. Cada uno de estos grupos es definido por su centroide, que inicialmente es seleccionado al azar y se ajusta iterativamente basándose en los datos asignados al grupo.

Es especialmente útil en aplicaciones de procesamiento de imágenes para la segmentación de imágenes, donde se pueden identificar diferentes regiones basadas en las características de los píxeles. Este método es popular debido a su simplicidad y eficiencia.

Destaca en la segmentación de imágenes, permitiendo dividir la imagen en partes que representan áreas de interés diferentes, facilitando el análisis o procesamiento posterior de esas áreas específicas.

El funcionamiento básico de K-Means es el siguiente:

### 1. Inicialización

Se seleccionan aleatoriamente  $k$  centroides.

### 2. Asignación

Cada pixel en el conjunto de datos  $x_i$  es asignado al centroide más cercano utilizando la siguiente fórmula

$$C_i = \arg \min_{j \in \{1, \dots, k\}} \|x_i - \mu_j\|^2$$

donde:

- $C_i$  es el cluster asignado al punto  $x_i$ ,
- $x_i$  es el píxel en la imagen,
- $\mu_j$  es el centroide del cluster  $j$ ,
- $k$  es el número de clusters.

### 3. Actualización de Centroides

Los centroides se recalculan como el promedio de los valores de todos los píxeles asignados a su cluster.

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

donde:

- $\mu_j$  es el nuevo centroide del cluster  $j$ ,
- $C_j$  es el conjunto de píxeles asignados al cluster  $j$ .

### 4. Repetición

Los pasos 2 y 3 se repiten hasta que los centroides no cambian significativamente entre iteraciones, lo que indica que se ha alcanzado la convergencia. Siendo el umbral de convergencia por defecto  $\epsilon = 1e - 4$ .

## Fuzzy C-Means

Fuzzy C-Means (FCM) [9] es un algoritmo de clustering basado en la lógica difusa que permite a un punto de datos pertenecer a más de un grupo (o cluster) con diferentes grados de pertenencia. A diferencia de K-Means, donde cada punto pertenece exclusivamente a un solo cluster, en FCM cada punto tiene un valor de pertenencia para cada cluster, lo que indica el grado con el que un punto pertenece a un cluster específico.

FCM es especialmente útil en aplicaciones donde los límites entre las clases no están claramente definidos, lo que ocurre comúnmente en problemas de segmentación de imágenes. De esta forma, al segmentar una imagen que tenga regiones superpuestas con diferentes grados de pertenencia dará un resultado mejor, lo que es útil para segmentar áreas con transiciones suaves.

El funcionamiento básico de Fuzzy C-Means es el siguiente:

### 1. Inicialización

Se establece el número de clusters  $c$ , el parámetro de difusividad  $m$  (usualmente 2), y se inicializa la matriz de pertenencias aleatoriamente, denotando la probabilidad de que cada punto pertenezca a cada cluster.

### 2. Cálculo de centroides

Los centroides se calculan utilizando los valores de pertenencia. El centroide de cada cluster se calcula como:

$$v_j = \frac{\sum_{i=1}^n (u_{ij})^m \cdot x_i}{\sum_{i=1}^n (u_{ij})^m}$$

donde:

- $u_j$  es el centroide del cluster  $j$ ,
- $u_{ij}$  es el valor de pertenencia del pixel  $x_i$  al cluster  $j$ ,
- $m$  es el parámetro de difusividad,
- $n$  es el número total de píxeles.

### 3. Actualización de valores de pertenencia

Se actualizan los valores de pertenencia para cada píxel en cada cluster utilizando la siguiente fórmula:

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}}$$

donde:

- $u_{ij}$  es el valor de pertenencia del píxel  $x_i$  al cluster  $j$ ,
- $v_j$  es el centroide del cluster  $j$ ,
- $m$  es el parámetro de difusividad,
- $c$  es el número total de clusters.

#### 4. Repetición

Se repiten los pasos 2 y 3 hasta que la diferencia entre las actualizaciones de la matriz de pertenencias en iteraciones sucesivas sea menor que un umbral determinado.

Hay que tener en cuenta que el parámetro  $m$  controla la difusividad de los clusters. Si  $m=1$ , FCM se comporta como K-Means, mientras que valores mayores de  $m$  generan clusters más difusos.

### Gaussian Mixtures Models

Gaussian Mixture Models (GMM) [11] es un algoritmo de clustering basado en la idea de que los datos pueden ser modelados como una combinación de múltiples distribuciones gaussianas. Cada cluster se representa como una distribución gaussiana con su media y varianza propias. GMM es una extensión más flexible de K-Means, ya que permite que los clusters tengan diferentes formas y tamaños.

GMM es útil en aplicaciones donde los datos tienen una estructura más compleja que no puede ser modelada adecuadamente por algoritmos como K-Means. Algunas razones para utilizarlo incluyen:

1. Modelado de Datos Complejos: Es capaz de modelar clusters con formas y tamaños arbitrarios.
2. Identificación de Patrones Multimodales: Puede identificar clusters que tienen múltiples modos.
3. Análisis de Datos Probabilístico: Proporciona una probabilidad para cada punto de pertenecer a cada cluster.

El funcionamiento básico de Gaussian Mixtures Models es el siguiente:

### 1. Inicialización

Se establece el número de componentes gaussianas  $k$ , y se inicializan aleatoriamente los parámetros de las distribuciones: media ( $\mu_k$ ), covarianza ( $\Sigma_k$ ), y pesos ( $\pi_k$ ).

### 2. Cálculo de probabilidades posteriores ( $\gamma_{ik}$ )

Se calcula la probabilidad de que cada píxel  $x_i$  pertenezca al cluster  $k$  utilizando la regla de Bayes. Esto se conoce como la responsabilidad del cluster  $k$  para el píxel  $x_i$ :

$$\gamma_{ik} = \frac{\pi_k \cdot \mathcal{N}(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^k \pi_j \cdot \mathcal{N}(x_i | \mu_j, \Sigma_j)}$$

donde:

- $\mathcal{N}(x_i | \mu_k, \Sigma_k)$  es la función de densidad de probabilidad de una distribución gaussiana con media  $\mu_k$  y covarianza  $\Sigma_k$ .

### 3. Reestimación de parámetros

- Medias ( $\mu_k$ ):

$$\mu_k = \frac{\sum_{i=1}^n \gamma_{ik} \cdot x_i}{\sum_{i=1}^n \gamma_{ik}}$$

- Covarianzas ( $\Sigma_k$ ):

$$\Sigma_k = \frac{\sum_{i=1}^n \gamma_{ik} \cdot (x_i - \mu_k)(x_i - \mu_k)^\top}{\sum_{i=1}^n \gamma_{ik}}$$

- Pesos ( $\pi_k$ ):

$$\pi_k = \frac{\sum_{i=1}^n \gamma_{ik}}{n}$$

### 4. Repetición

Se repiten los pasos 2 y 3 hasta que los cambios en los parámetros sean menores a un umbral dado, o se alcance un número máximo de iteraciones.

## Información espacial

El modelo Hidden Markov Random Field (HMRF) [15] es un modelo probabilístico que combina la teoría de campos aleatorios de Markov con modelos ocultos. Se utiliza ampliamente para problemas de visión computacional, como la segmentación de imágenes. El algoritmo HMRF-EM combina el modelo HMRF con el algoritmo Expectation-Maximization (EM) para la segmentación de imágenes.

El modelo HMRF-EM (Hidden Markov Random Field - Expectation Maximization) es útil en segmentación de imágenes porque:

1. Regulariza la segmentación, reduciendo el ruido.
2. Considera relaciones entre píxeles vecinos para segmentar áreas similares.
3. El algoritmo EM ajusta parámetros para mejorar la precisión.
4. La versión Edge-Prior-Preserving mantiene bordes importantes en la imagen.

El funcionamiento básico del modelo Hidden Markov Random Field es el siguiente:

### 1. Inicialización

Se establece un conjunto inicial de parámetros  $\Theta_0$

$$\Theta_0 = [\mu_{initial}, \sigma_{initial}];$$

### 2. Cálculo de la Distribución de Verosimilitud

Se calcula la distribución de verosimilitud  $P^{(t)}(y_i|x_i, \theta_{x_i})$  en la iteración  $t$ .

$$P^{(t)}(y_i|x_i, \theta_{x_i}) = G(y_i; \theta_{x_i}) = \frac{1}{\sqrt{2\pi\sigma_{x_i}^2}} \exp\left(-\frac{(y_i - \mu_{x_i})^2}{2\sigma_{x_i}^2}\right)$$

### 3. Estimación MAP

Se estima la configuración de etiquetas mediante el criterio de Máxima A Posteriori (MAP):

$$x^{(t)} = \arg \max_{x \in \chi} \left\{ P(y|x, \Theta^{(t)})P(x) \right\} = \arg \min_{x \in \chi} \left\{ U(y|x, \Theta^{(t)}) + U(x) \right\}$$

donde:

- $U(y|x, \Theta)$  es la energía de verosimilitud:

$$U(y|x, \Theta) = \sum_i \left[ \frac{(y_i - \mu_{x_i})^2}{2\sigma_{x_i}^2} + \ln \sigma_{x_i} \right]$$

- $U(x)$  es la energía prior:

$$U(x) = \sum_{c \in C} V_c(x)$$

- $V_c(x)$  es el potencial de clique, que en este caso se define para un par de píxeles vecinos:

$$V_c(x_i, x_j) = \frac{1}{2}(1 - I_{x_i, x_j})$$

donde:

$$I_{x_i, x_j} = \begin{cases} 0 & \text{si } x_i \neq x_j \\ 1 & \text{si } x_i = x_j \end{cases}$$

#### 4. Actualización de Parámetros (M-step)

Se actualizan los parámetros  $\Theta^{(t+1)}$  usando las distribuciones posteriores.

- Media ( $\mu_l^{(t+1)}$ ):

$$\mu_l^{(t+1)} = \frac{\sum_i P^{(t)}(l|y_i)y_i}{\sum_i P^{(t)}(l|y_i)}$$

- Varianza ( $\sigma_l^2$ ):

$$(\sigma_l^{(t+1)})^2 = \frac{\sum_i P^{(t)}(l|y_i)(y_i - \mu_l^{(t+1)})^2}{\sum_i P^{(t)}(l|y_i)}$$

- Posteriore (  $P^{(t)}(l|y_i)$ ):

$$P^{(t)}(l|y_i) = \frac{G(y_i; \theta_l)P^{(t)}(l|x_{N_i}^{(t)})}{P^{(t)}(y_i)}$$

## 5. Repetición

Se repiten los pasos 2 a 4 hasta que la función de energía  $U(y|x, \Theta) + U(x)$  converja o se alcance el número máximo de iteraciones.

## 3.4. Clasificación de resultados

La clasificación de los resultados obtenidos es un paso fundamental, ya que permite evaluar de forma objetiva si la aplicación de métodos de transformación invariante mejora los resultados. Se compararán las imágenes, tanto la original como la invariante, con una imagen ‘ground truth’ para observar numéricamente el acierto de cada una de ellas.

### Imágenes ground truth

Las imágenes ‘ground truth’ son segmentaciones realizadas manualmente mediante programas de edición de imágenes. Al haber sido segmentadas por un ser humano, representan una segmentación casi perfecta, que sirve como ideal al que se aspira. Se dice ‘casi perfecta’ porque puede haber un mínimo error imperceptible a simple vista, pero que puede afectar matemáticamente, impidiendo alcanzar un 100 % de precisión.

En la figura 3.6 se puede ver el la imagen original y su correspondiente imagen ‘ground truth’.

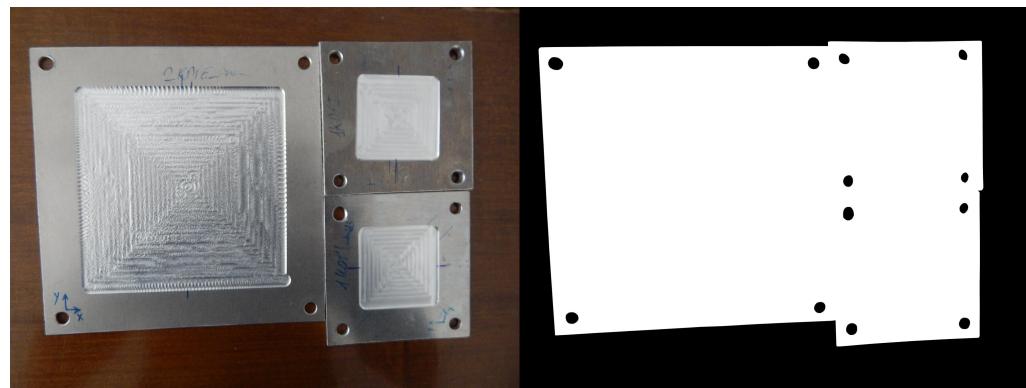


Figura 3.6: Resultado de crear la imagen ‘ground truth’

## Porcentaje de acierto

El porcentaje de acierto permite evaluar la precisión de la segmentación realizada al compararla con una imagen en la que se ha segmentado manualmente la pieza del fondo (imagen ground truth). La fórmula para calcularlo entre dos imágenes es:

$$\text{coincidencias} = \sum_{i,j} \delta(\text{img1\_bnw}(i,j), \text{img2\_bnw}(i,j))$$

donde:

- coincidencias es el número total de píxeles que coinciden entre las dos imágenes en blanco y negro.
- $\sum_{i,j}$  indica que se realiza una suma sobre todos los píxeles de la imagen.
- $\text{img1\_bnw}(i,j)$  es el valor de cada posición  $(i,j)$  de la primera imagen
- $\text{img2\_bnw}(i,j)$  es el valor de cada posición  $(i,j)$  de la segunda imagen
- $\delta$  es la función delta de Kronecker que se define como:

$$\delta(a,b) = \begin{cases} 1 & \text{si } a = b \\ 0 & \text{si } a \neq b \end{cases}$$

Finalmente la función con la que se obtiene el porcentaje de acierto entre la dos imágenes es la siguiente.

$$\text{porcentaje\_coincidencia} = \left( \frac{\text{coincidencias}}{\text{numero\_pixeles(img1)}} \right) \times 100$$

En la futura sección 5.6, se presentan ejemplos del resultado de aplicar este método para obtener un porcentaje de coincidencia en casos reales. En dicha sección se mostrarán análisis de imágenes con resultados numéricamente positivos, así como casos en los que este enfoque no es del todo acertado, debido a la suposición de que la pieza se encuentra en el centro de la imagen, lo cual no siempre es cierto. Finalmente, se considerarán casos más complejos en los que el resultado es visualmente mejor que numéricamente, dado que se logra minimizar las imperfecciones causadas por las reflexiones en las piezas metálicas pero alcanza porcentajes de coincidencia menores.



---

## **4. Técnicas y herramientas**

---

### **4.1. Control de versiones**

#### **Git**

Git es un sistema de control de versiones distribuido que permite trabajar de manera autónoma con copias completas del repositorio. Es ideal para utilizar múltiples ramas de desarrollo, facilitando la experimentación y el desarrollo paralelo sin interferencias. Destaca por su rapidez y eficiencia, almacenando información como cambios incrementales en lugar de archivos completos, lo que optimiza el uso del espacio y mejora el rendimiento en proyectos de cualquier tamaño.

### **4.2. Hosting del repositorio**

#### **GitHub**

- Herramientas consideradas: GitHub y GitLab.
- Motivo de la elección de GitHub:

GitHub es una plataforma de hosting de repositorios Git que se ha convertido en uno de los servicios más populares para la gestión colaborativa de proyectos de software. Cuenta con una interfaz de usuario intuitiva y su integración con una gran cantidad de herramientas de desarrollo, lo que lo hace accesible tanto para equipos pequeños como grandes empresas.

Es por esta mayor capacidad de integración por la cual se ha optado por usar GitHub en lugar de GitLab, además de que durante la carrera se ha utilizado siempre GitHub.

### **4.3. Metodologías ágiles**

#### **Scrum**

Scrum es una metodología ágil de gestión y planificación de proyectos de software que se centra en el trabajo en equipo, la responsabilidad y la iteración progresiva de objetivos a corto plazo, conocidos como sprints. Esto ha sido muy útil, ya que se han definido los sprints como el tiempo entre reuniones, en las cuales se revisaba el trabajo realizado y se plantearan las líneas a seguir en el siguiente periodo (sprint).

#### **Kanban**

Kanban se ha implementado dentro de los sprints de Scrum en el proyecto, combinando la estructura iterativa de Scrum con la flexibilidad y visualización del flujo de trabajo de Kanban. Esta integración ha permitido gestionar y ajustar continuamente las tareas dentro de cada sprint, optimizando la entrega y la eficiencia sin adherirse estrictamente a una sola metodología.

### **4.4. Gestión del proyecto**

#### **Zube**

- Herramientas consideradas: Zube, Trello, Jira.
- Motivo de la elección de Zube:

Zube es una plataforma de gestión de proyectos ágil que se integra directamente con GitHub, permitiendo una sincronización automática entre tareas e issues de GitHub. Su interfaz soporta configuraciones de tableros Kanban y Scrum, facilitando la adaptación a diferentes flujos de trabajo. Zube ofrece visibilidad del progreso del proyecto y permite gestionar y priorizar tareas eficientemente, haciendo que la planificación y el seguimiento sean más accesibles y dinámicos para los equipos de desarrollo.

Esta herramienta fue elegida frente a Trello o Jira por recomendación de los tutores. Además de su excepcional integración con GitHub, ofrece una interfaz más sencilla y directa para gestionar proyectos ágiles eficazmente.

## GitHub Desktop

- Herramientas consideradas: GitHub Desktop, GitKraken.
- Motivo de la elección de GitHub Desktop:

GitHub Desktop es una aplicación que simplifica la manera en que los desarrolladores interactúan con Git y GitHub, proporcionando una interfaz gráfica de usuario en lugar de la línea de comandos.

GitHub Desktop ofrece una mejor integración con GitHub, lo que facilita trabajar de forma más fluida y coherente dentro del ecosistema de GitHub, a diferencia de GitKraken, cuya interfaz y funcionalidades son más genéricas y menos especializadas para usuarios de GitHub.

## 4.5. Comunicación

### Microsoft Outlook

- Herramientas consideradas: Microsoft Outlook, Gmail.
- Motivo de la elección de Microsoft Outlook:

Microsoft Outlook, parte de la suite de Microsoft Office, es una aplicación de gestión de información personal y cliente de correo electrónico. Es altamente valorado por su integración con el calendario, lo cual facilita a los usuarios la programación y seguimiento de citas y reuniones de manera eficiente.

Se optó por utilizar Microsoft Outlook debido a su facilidad de uso y porque las reuniones eran programadas por D. Pedro Latorre Carmona. Además, la Universidad de Burgos proporciona una cuenta de Outlook a todos los miembros de la universidad, lo que facilitó la decisión de adoptar esta herramienta para la gestión de correos electrónicos y calendarios en el proyecto.

## **Microsoft Teams**

- Herramientas consideradas: Microsoft Teams, Zoom, Google Meet.
- Motivo de la elección de Microsoft Teams:

Microsoft Teams es una plataforma de colaboración integrada desarrollada por Microsoft. Teams permite a los usuarios crear y unirse a espacios de trabajo compartidos, donde pueden organizar reuniones, gestionar proyectos y co-crear documentos en tiempo real de manera fácil y eficiente.

La Universidad de Burgos proporciona cuentas de Microsoft Teams a todos los miembros de la universidad, lo que influyó en la decisión de utilizar esta plataforma.

## **4.6. Entorno de desarrollo integrado (IDE)**

### **MATLAB**

- Herramientas consideradas: Python, MATLAB.
- Motivo de la elección de MATLAB:

MATLAB es un entorno de programación y lenguaje de scripting desarrollado por MathWorks, utilizado principalmente para cálculos numéricos, análisis de datos, visualización y desarrollo de algoritmos. Es especialmente eficiente en el manejo de matrices, lo que lo hace ideal para el procesamiento de imágenes, donde cada imagen se trata como una matriz de píxeles.

Se eligió MATLAB sobre Python para este proyecto debido a su superior velocidad y eficiencia en el cálculo de operaciones matriciales. Inicialmente, se utilizó Python, pero los tiempos de ejecución eran muy elevados.

### **Overleaf**

- Herramientas consideradas: Overleaf, TeXstudio.
- Motivo de la elección de Overleaf:

Overleaf es una plataforma de edición y colaboración en línea para documentos LaTeX. Permite la colaboración en tiempo real y ofrece plantillas predefinidas para diversos documentos académicos. Por este motivo, al permitir la edición y revisión simultánea por varias personas, se eligió Overleaf en lugar de TeXstudio.

## 4.7. Documentación

### LaTeX

- Herramientas consideradas: LaTeX, OpenOffice.
- Motivo de la elección de LaTeX:

LaTeX es un sistema de preparación de documentos conocido por su alta calidad tipográfica, ideal para textos científicos y técnicos. Destaca en la gestión de referencias, citas y bibliografías, y ofrece un control detallado del formato.

Se eligió LaTeX en lugar de OpenOffice debido a estas ventajas y con el propósito de aprender a utilizar una herramienta nueva, ya que anteriormente no se había utilizado.

### Markdown

Markdown es un lenguaje de marcado ligero ideal para formatear texto en la web. Su sintaxis simple permite crear documentos con encabezados, listas, enlaces e imágenes de manera rápida y eficiente.

Por esta razón, se ha utilizado tanto en los README como en los distintos commits y comentarios de issues en GitHub.

## 4.8. Librerías

Se ha requerido de las siguientes librerías:

### Statistics and Machine Learning Toolbox

Statistics and Machine Learning Toolbox de MATLAB ofrece herramientas avanzadas para clustering con K-Means y Gaussian Mixture Models (GMM).

Se eligió esta toolbox porque, además de permitir al usuario ver el resultado del algoritmo invariante, ofrece la posibilidad de analizar de manera más clara la mejora en la posterior segmentación, en este caso con K-Means y GMM.

## **Fuzzy Logic Toolbox**

Fuzzy Logic Toolbox de MATLAB ofrece herramientas para implementar Fuzzy C-Means.

Al igual que en el caso de Statistics and Machine Learning Toolbox, Fuzzy Logic Toolbox ha sido necesario para ofrecer la posibilidad de segmentar mediante Fuzzy C-Means.

## **HMRF-EM-image**

HMRF-EM-image es una librería en MATLAB para segmentar imágenes usando Campos Aleatorios de Markov Ocultos (HMRF) y el algoritmo Expectation-Maximization (EM).

El uso de esta librería supone una mejora frente a los métodos de agrupamiento anteriores, ya que contempla la información espacial. Esto significa que no solo se analiza el valor del color del píxel, sino también los píxeles que lo rodean, lo que da lugar a segmentaciones de mejor calidad.

## **MATLAB Compiler**

MATLAB Compiler permite convertir aplicaciones MATLAB en ejecutables independientes y aplicaciones web, facilitando su distribución sin necesidad de licencias adicionales de MATLAB.

---

## **5. Aspectos relevantes del desarrollo del proyecto**

---

En esta sección se presentan los aspectos más destacados del desarrollo del proyecto. Esto abarca desde las decisiones tomadas hasta los desafíos que han surgido a lo largo del proceso.

### **5.1. Inicio del proyecto**

Decidí realizar este proyecto porque me pareció muy interesante su objetivo de demostrar que las transformaciones invariantes pueden mejorar la identificación de piezas metálicas.

El análisis de imágenes es un tema que me interesado durante toda la carrera especialmente en hardware de aplicación específica donde analizábamos imágenes y su contenido especialmente una práctica que consistía en analizar el contenido de una imagen separando diferentes objetos y finalmente sumando el valor de las monedas actuales que se encontraban en ella.

También tenía experiencia previa con algoritmos de agrupamiento de imágenes, gracias a mi participación en asignaturas como sistemas inteligentes. Esta familiaridad fue uno de los factores clave para elegir este proyecto.

Después una serie de reuniones con Pedro Latorre, este se ofreció a ser el tutor de mi proyecto. Me aclaró que este sería más teórico que otros y que requeriría una investigación profunda antes de comenzar.

## 5.2. Formación

El proyecto requería ciertos conocimientos técnicos que inicialmente no poseía, ya que no se tratan habitualmente en la carrera. Estos conocimientos eran principalmente teóricos y estaban relacionados con el comportamiento de la luz en superficies metálicas, así como con el entendimiento de lo que es una transformación invariante, su utilidad y cuáles métodos podrían ser beneficiosos para demostrar una mejoría en los resultados.

Durante el desarrollo del proyecto, también necesité familiarizarme con MATLAB App Designer para crear una aplicación que permitiera aplicar los diferentes algoritmos invariantes y de agrupamiento a imágenes propias del usuario o suministradas por la aplicación.

## 5.3. Metodologías

Desde la primera reunión, establecimos una serie de pautas para la correcta ejecución del proyecto, las cuales se reforzaron con la llegada de Carlos López. Empleamos la metodología Scrum, aunque adaptada a la escala de nuestro proyecto educativo, dado que no se trataba de un equipo grande de programadores. Por ejemplo, en lugar de tener reuniones diarias, inicialmente programamos las reuniones cada semana. Sin embargo, debido a distintos problemas de horarios, el intervalo entre reuniones se ajustó eventualmente a cada dos semanas. En estas reuniones, revisábamos los avances logrados y planificábamos las tareas a realizar para el próximo periodo. No fue hasta que Carlos se unió al equipo que comenzamos a denominar estos periodos como "sprints" y registrarlos en Zube para monitorear el progreso de las tareas durante cada sprint.

La estrategia seguida fue de prueba y error, experimentando con numerosos algoritmos y descartando aquellos que no ofrecían mejoras significativas en la identificación precisa de las piezas metálicas dentro de imágenes. La calidad de la imagen, la cantidad de luz y las imperfecciones resultaron ser factores críticos, lo que llevó al descarte de muchos algoritmos de transformación invariante. Esta misma dinámica se aplicó a los algoritmos de agrupamiento con información espacial: se probaron muchos, pero se descartaron los que no ofrecían ventajas claras sobre técnicas de agrupamiento más simples como K-Means, Fuzzy C-Means y GMM.

## 5.4. Cambio de lenguaje y entorno de programación

Al inicio del proyecto, comencé utilizando Python en un archivo de Jupyter Notebook, pero debido a los largos tiempos de ejecución del algoritmo invariante y del método de agrupamiento, decidimos migrar a MATLAB. Esta plataforma es reconocida por su eficiencia en el cálculo con matrices, y en esencia, eso es lo que son las imágenes.

Por esta razón, durante ese sprint, me dediqué a trasladar el código de Python a MATLAB, tarea que no resultó particularmente ardua. Para los métodos invariantes, el proceso consistió principalmente en seguir las especificaciones de las funciones detalladas en el artículo científico correspondiente. Además, los métodos de agrupamiento que estaba utilizando ya estaban implementados en las toolboxes de MATLAB, lo que facilitó aún más la migración.

## 5.5. Desarrollo de la app

La idea de desarrollar una aplicación surgió después de obtener resultados satisfactorios en una etapa temprana del proyecto, lo que motivó la creación de una herramienta que cualquier persona pudiera utilizar sin necesidad de conocimientos técnicos sobre métodos de transformación invariante, segmentación de imágenes o programación.

Inicialmente, se contempló el desarrollo de dos aplicaciones distintas. La primera permitía a los usuarios subir cualquier tipo de imagen, seleccionar los algoritmos que desearan utilizar de una lista de algoritmos invariantes y otra de métodos de agrupamiento, visualizar los resultados y finalmente guardarlos. La segunda aplicación era similar, pero limitada a imágenes ya proporcionadas por la propia aplicación, permitiendo a los usuarios comparar los resultados obtenidos con los ideales y evaluar así la eficacia de los métodos aplicados.

Más adelante, decidimos fusionar ambas aplicaciones en una sola, al ver que era viable continuar su desarrollo. Esta aplicación final combinó las características de las dos versiones anteriores y mejoró la estructura general. Este cambio fue posible gracias a que, tras practicar con la programación de aplicaciones en MATLAB, ya tenía más fluidez con el entorno de desarrollo.

Debido a los prolongados tiempos de ejecución de algunos de los métodos empleados por la aplicación, a causa de su complejidad computacional, se

decidió implementar una memoria caché para almacenar los resultados de ejecuciones anteriores. De esta manera, si se procesaba la misma imagen con los mismos algoritmos, no sería necesario ejecutar sobre esta los algoritmos, sino que los datos se cargarían directamente desde la memoria, ahorrando así recursos y tiempo. Esto puede ahorrar el calculo de una o tres imágenes ya que:

- Si únicamente coincidía el algoritmo de agrupamiento y la cantidad de centros, solo se ahorraría el calculo de la imagen original segmentada.
- Si únicamente coincidía el algoritmo invariante, solo se ahorraría el calculo de la imagen invariante.
- Si coincidía el algoritmo invariante, el algoritmo de agrupamiento y la cantidad de centros, se ahorrarían tanto el calculo de la imagen invariante, como el de las imágenes segmentadas original e invariante.

Además, se añadió una funcionalidad que permitiría mostrar una tabla con todas las imágenes guardadas en la memoria caché. Esta tabla incluiría datos relevantes sobre cada imagen, proporcionando al usuario la opción de acceder a cualquier imagen deseada o eliminar las que considerara innecesarias. Esta característica mejoró significativamente la eficiencia y la experiencia del usuario al interactuar con la aplicación.

## **Estructura de la aplicación**

La aplicación está diseñada de manera sencilla para evitar sobrecargar al usuario con información innecesaria además de ser responsive para de esta manera adaptarse de la mejor manera a las necesidades del usuario. Se divide en cuatro partes principales:

### **Bienvenida**

El apartado de bienvenida es la ventana con la que primero se encuentra el usuario, esta ofrece información básica sobre el objetivo de la aplicación así como los autores, la posibilidad de elegir entre usar la aplicación en inglés o en español y una ilustración que resume el funcionamiento básico de la aplicación de algoritmos invariantes sobre imágenes de piezas metálicas. Esto se puede apreciar en la figura 5.1

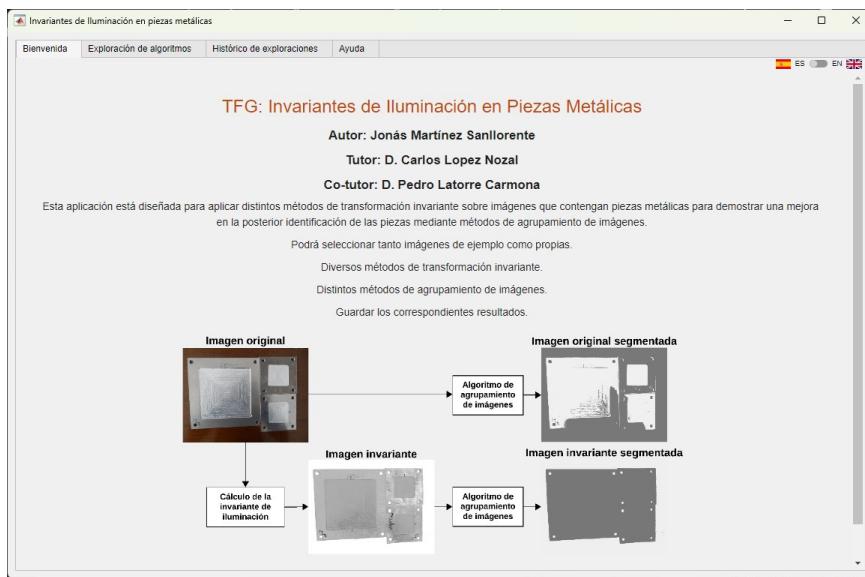


Figura 5.1: Apartado de bienvenida de InvIMP.

En esta figura se puede apreciar el aspecto de la ventana de bienvenida de InvIMP.

### Exploración de algoritmos

En este apartado, es donde el usuario puede cargar, ejecutar y visualizar los resultados de la ejecución.

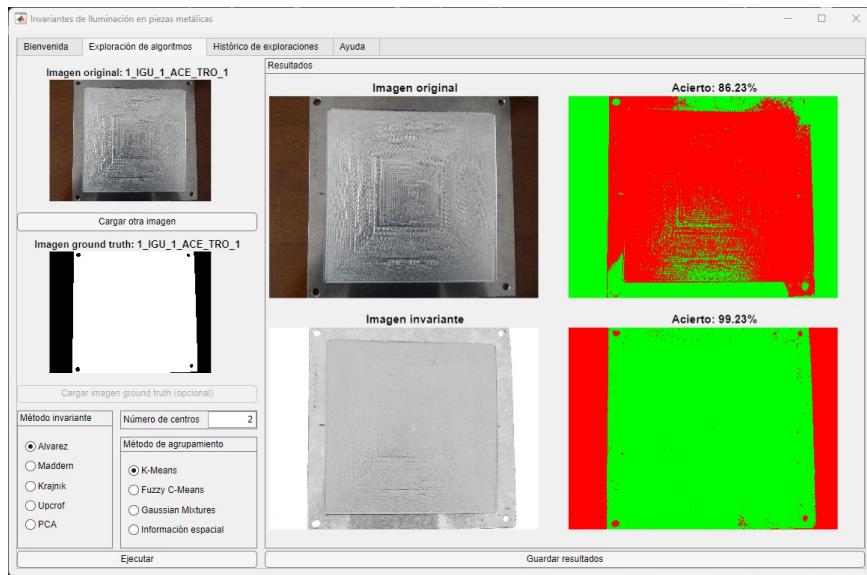


Figura 5.2: Apartado de exploración de algoritmos de InvIMP.

En esta figura se puede apreciar el aspecto de la ventana de exploración de algoritmos de InvIPM.

### Selección de imagen

Mediante el botón de ‘Cargar imagen de pieza metálica’ podremos seleccionar una imagen proporcionada por el sistema o una propia sobre la cual aplicar mas adelante.

### Selección de imagen ground truth

En caso de no ser una imagen proporcionada por el sistema se habilitará el botón de ‘Cargar imagen ground truth (opcional)’ para de esta manera poder proporcionar una propia. Esta imagen como indica el texto no es necesaria para la ejecución exceptuando el obtener el porcentaje de acierto.

### Selección de algoritmo invariante

Se proporcionan los distintos métodos invariantes: Álvarez [2], Maddern [7], Krajník [6], Upcroft [13] y PCA [5].

Estos están ordenados de manera que el primero es el que mejores resultados ofrece según las pruebas realizadas. Por otro lado, PCA, aunque muestra una clara mejoría en muchas situaciones, es un enfoque más clásico que numéricamente no alcanza resultados tan buenos como los demás.

### Selección de algoritmo de agrupamiento

Se proporcionan los distintos métodos de agrupamiento: K-Means [10], Fuzzy C-Means [9], Gaussian Mixtures [11], Información espacial, el cual es HMRF\_EM [15].

Estos están ordenados de manera que el primero es el menos costoso computacionalmente, por lo que ofrecerá mejores tiempos de ejecución. El último, en cambio, tiene tiempos de ejecución más largos, ya que es un algoritmo que no solo considera el color de un píxel en particular, sino también el de los píxeles circundantes.

### Selección de número de centros

Mediante este cuadro de texto el usuario puede indicar el numero de centros a utilizar en los algoritmos de agrupamiento. Por defecto este numero será 2.

El número de centros ha de cumplir las siguientes condiciones: ser igual o mayor que 2 y ser igual o menor que 10.

Esto se debe a que, como mínimo, se necesitan dos centros para poder separar entre pieza y fondo. El máximo se establece porque, a mayor cantidad de centros, aumenta la complejidad computacional, lo cual alargaría considerablemente los tiempos de ejecución.

### Ejecución

Al pulsar el botón de ‘Ejecutar’ se dará inicio a la ejecución sobre la imagen seleccionada de los distintos algoritmos elegidos. Durante la ejecución, se proporcionará retroalimentación sobre el porcentaje de progreso y la etapa que se está calculando en ese momento.

### Representación de resultados

Tras finalizar la ejecución se mostrarán en la parte derecha cuatro imágenes, estas serán:

- La imagen original (arriba a la izquierda).
- La imagen original segmentada (arriba a la derecha).
- La imagen invariante (abajo a la izquierda).
- La imagen invariante segmentada (abajo a la derecha).

En el caso de que se haya utilizado una imagen proporcionada por la aplicación o que el usuario haya proporcionado una imagen de referencia

(ground truth), el título tanto de la imagen original segmentada como de la imagen invariante segmentada mostrará el porcentaje de acierto. De esta manera, el usuario podrá comparar los resultados no solo visualmente, sino también numéricamente.

### **Histórico de exploraciones**

En este apartado, se nos presenta una tabla en la cual aparecen a forma de lista todas aquellas ejecuciones que hemos realizado tanto en esta sesión como en sesiones previas. El contenido de las columnas de esta tabla es el siguiente:

- **Nombre del fichero:** este es el nombre del archivo que el usuario seleccionó.
- **Fecha:** esta es la fecha de la ejecución. Esta está en formato DD-MMM-YYY HH-MM-SS.
- **Algoritmo invariante:** muestra el nombre del algoritmo invariante aplicado sobre la imagen. En el caso de no haberse aplicado ningún algoritmo invariante muestra ‘No aplicado’.
- **Algoritmo de agrupamiento:** muestra el nombre del algoritmo de agrupamiento aplicado sobre la imagen. En el caso de no haberse aplicado ningún algoritmo de agrupamiento muestra ‘No aplicado’.
- **Número de centros:** muestra la cantidad de centros utilizados en el algoritmo de agrupamiento aplicado sobre la imagen. En el caso de no haberse aplicado ningún algoritmo de agrupamiento muestra ‘-’.
- **Imagen ground truth:** indica mediante ‘Sí’ o ‘No’ el si la ejecución de dicha imagen tenía o bien imagen asociada o el usuario había proporcionado una propia.
- **Medida de calidad:** indica la tasa de acierto que se ha logrado al comparar la imagen correspondiente a dicha fila con si imagen ground truth. En caso de no tener imagen ground truth esta celda estará en blanco.
- **Archivo:** muestra un checkbox el cual al pulsar sobre el abre en una nueva ventana la imagen correspondiente a dicha fila. Esta nueva ventana tiene una serie de opciones aparte de la visualización permitiendo que el usuario la guarde o imprima entre otras opciones.

El usuario también puede si lo desea ordenar la tabla por la columna que quiera.

Nombre del fichero	Fecha	Algoritmo invariante	Algoritmo de agrupamiento	Número de centros	Imagen ground truth	Medida de calidad	Archivo
1_GU_1_ACE_TRO_1	23-Jun-2024 19:48:42	Alvarez	No aplicado	-	No		<input type="checkbox"/>
1_GU_1_ACE_TRO_1	23-Jun-2024 19:48:42	No aplicado	K-Means	2	Sí	86.2285%	<input type="checkbox"/>
1_GU_1_ACE_TRO_1	23-Jun-2024 19:48:42	Alvarez	K-Means	2	Sí	99.231%	<input type="checkbox"/>

Figura 5.3: Apartado de histórico de exploraciones de InvIMP.

En esta figura se puede apreciar el aspecto de la ventana del histórico de exploraciones de InvIPM.

A parte de la tabla, en la parte inferior hay dos botones, uno para ver las imágenes y otro para borrar el contenido de la memoria cache.

### Ver imágenes

Al pulsar en este botón se abrirá una ventana nueva donde podremos ver y seleccionar todas aquellas imágenes que están almacenadas en la memoria caché.

### Borrar datos

Al pulsar en este botón aparecerá una ventana emergente indicando si realmente queremos borrar de forma irreversible el contenido de la memoria caché. En caso de pulsar en el botón de ‘Borrar’ se borrará su contenido además de actualizarse la tabla ya que dichas imágenes no se encontraran disponibles. Tras esto nos indicará que la memoria cache se ha borrado exitosamente.

## Ayuda

En este apartado, tal y como se muestra en la figura 5.4, se puede obtener información sobre el funcionamiento, los algoritmos invariantes, los algoritmos de agrupamiento, la caché y la tasa de acierto. Al estar la propia documentación implementada dentro de la aplicación, resulta muy cómodo buscar información sobre cualquier tema de interés.

Además, el idioma de la documentación corresponde con el que se ha seleccionado en la bienvenida 5.5.

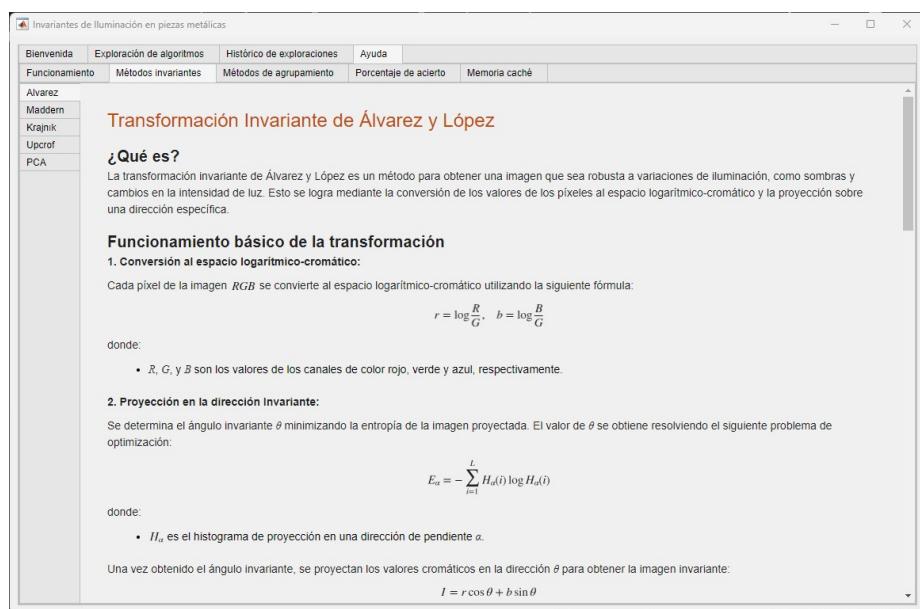


Figura 5.4: Apartado de ayuda de InvIMP.

En esta figura se puede apreciar el aspecto de la ventana de ayuda de InvIMP.

## 5.6. Resultados de la experimentación

Para analizar los resultados, he ejecutado cada método de transformación invariante sobre las 24 imágenes de prueba, generando para cada una de ellas la imagen original segmentada, la imagen invariante y la imagen invariante segmentada. Esto se ha realizado para cada uno de los cinco métodos de transformación invariante y para cada uno de los cuatro métodos de agrupamiento, tanto para dos como para tres centros. Evitando imágenes repetidas, se han obtenido un total de 1.272 imágenes.

Este análisis ha sido posible gracias a las 24 imágenes de prueba que contiene la aplicación, junto con sus correspondientes imágenes ground truth, las cuales han sido editadas manualmente para separar correctamente las piezas del fondo.

Dichos resultados se pueden encontrar el siguiente directorio de el repositorio de GitHub del proyecto [8].

## Resultados de calidad de segmentación

Como se puede apreciar en los resultados recopilados en los dos archivos Excel que se encuentran en el repositorio [8] en la mayoría de casos el aplicar métodos de transformación invariante sobre las imágenes de las piezas metálicas mejora considerablemente los resultados de la posterior segmentación. Y no solo eso sino que en los resultados mas complejos donde a primera vista pueda parecer que el resultado es numéricamente peor se puede apreciar una disminución de los errores de identificación de las piezas debido a reflexiones sobre el metal, esto se expondrá mas adelante en 5.6.

Después de toda la experimentación hay un claro algoritmo que es superior a la hora de diferenciar las piezas de el fondo, este es el propuesto por Álvarez [2] el cual da los siguientes resultados tanto de media 5.1 como de mediana 5.2 utilizando los distintos algoritmos de agrupamiento.

Agrupamientos \ Método	Método	RGB	Álvarez	Maddern	Krajnik	PCA
<b>K-Means</b>		74,91 %	<b>89,49 %</b>	65,51 %	55,94 %	50,41 %
<b>Fuzzy</b>		74,88 %	<b>89,41 %</b>	65,50 %	59,84 %	62,16 %
<b>C-Means</b>						
<b>GMM</b>		83,14 %	<b>90,79 %</b>	65,16 %	78,63 %	69,35 %
<b>HMRF_EM</b>		78,46 %	<b>91,29 %</b>	65,49 %	74,76 %	69,80 %

Tabla 5.1: Media de porcentajes de acierto.

Agrupamientos \ Método	Método	RGB	Álvarez	Maddern	Krajnik	PCA
<b>K-Means</b>		76,26 %	<b>98,10 %</b>	52,68 %	47,85 %	37,51 %
<b>Fuzzy</b>		76,21 %	<b>98,09 %</b>	52,68 %	69,51 %	<b>80,83 %</b>
<b>C-Means</b>						
<b>GMM</b>		84,86 %	<b>99,15 %</b>	52,68 %	<b>86,31 %</b>	83,79 %
<b>HMRF_EM</b>		79,82 %	<b>98,56 %</b>	52,68 %	<b>87,26 %</b>	<b>83,99 %</b>

Tabla 5.2: Mediana de porcentajes de acierto.

Como se puede apreciar no se han añadido a las tablas los resultados de Upcroft, esto de debe a que los resultados entre Upcroft y Maddern son mínimos ya que comparten la estructura del algoritmo casi al 100% y únicamente sus resultados se alejan en el caso de imágenes muy complejas.

Como se puede apreciar numéricamente Álvarez supone una mejora notable frente a los demás algoritmos obteniendo resultados de una gran calidad. Si únicamente comparamos estos resultados con los originales nos da el siguiente gráfico de la figura 5.5. Más tarde se analizara el gráfico con todos los algoritmos invariantes, pero al contener demasiada malformación se analizara el de Álvarez por separado.

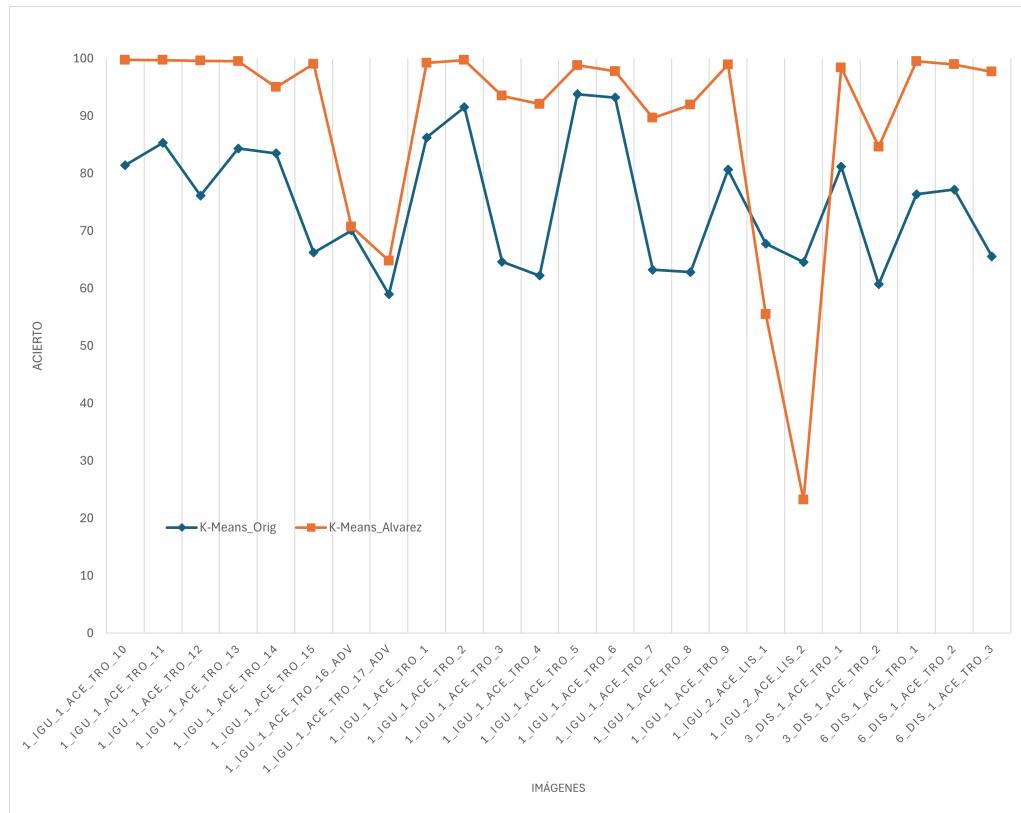


Figura 5.5: Tasa de acierto de los de la gemación aplicando K-Means sobre la imagen original frente a los resultados de utilizar previamente Álvarez.

En la única situación en la que da resultados numéricamente menores a la segmentación sin aplicar ningún algoritmo invariante en el caso de las dos imágenes complejas de las cuales una aparecerá mas adelante en la figura 5.12.

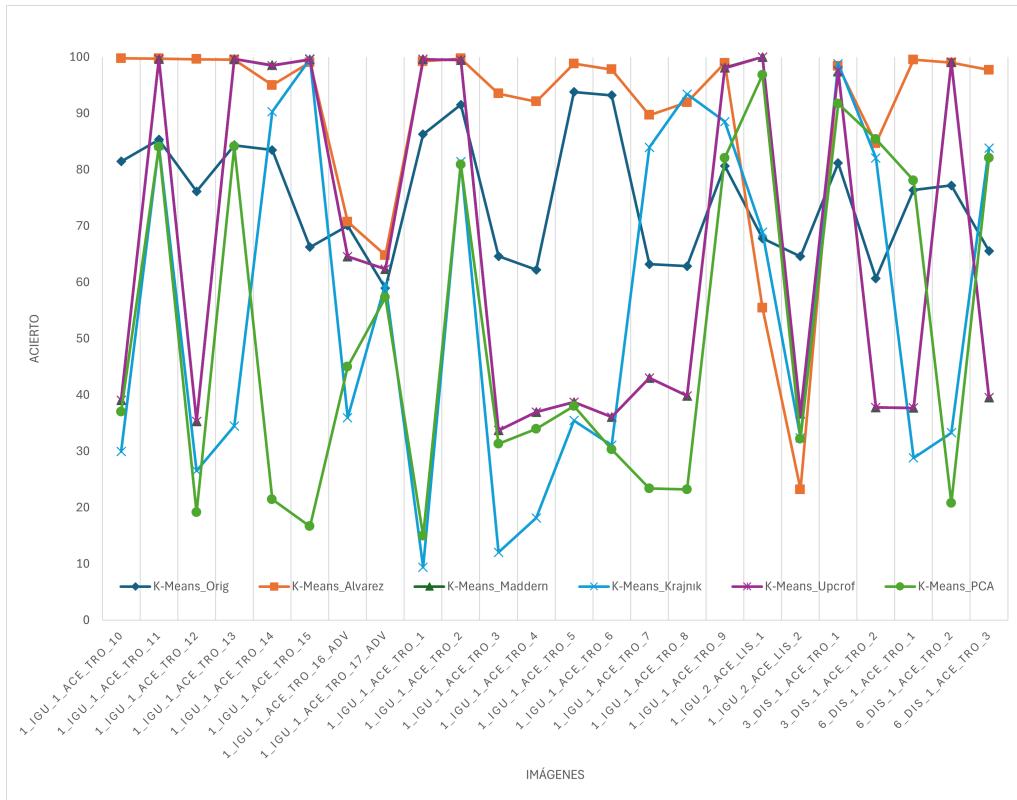


Figura 5.6: Tasa de acierto de los de la gemación aplicando K-Means sobre la imagen original frente a los resultados de utilizar los distintos métodos invariantes.

Con los resultados de todos los demás tipos de transformaciones invariantes el gráfico quedaría como el de la figura 5.6.

## Análisis de casos

A continuación se mostrarán dos resultados de imágenes diferentes, estos suponen una caso con condiciones ideales mientras que el otro es un caso más complejo y en cierta manera más cercano a lo que puede llegar a ser un escenario más realista en la industria.

### Resultados mejores numéricamente

En este caso se va a analizar la mejora numérica que se puede llegar a conseguir al aplicar sobre una imagen obtenida en unas condiciones óptimas,

un único foco de luz, un ángulo de 90°, sin objetos metálicos o de colores parecidos en el fondo etc.

La figura que se analizará es la siguiente la cual forma parte de las imágenes que ofrece la propia aplicación para probar los distintos algoritmos, esta imagen es la 3\_DIS\_1\_ACE\_TRO\_1:

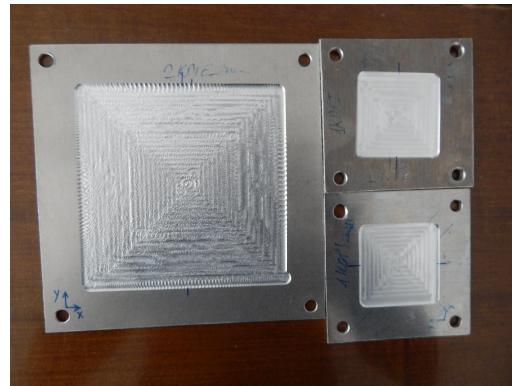


Figura 5.7: Imagen piezas metálicas en condiciones idóneas.

La imagen ground truth que se ha creado para comparar los resultados de una segmentación perfecta es la de la figura 5.8.

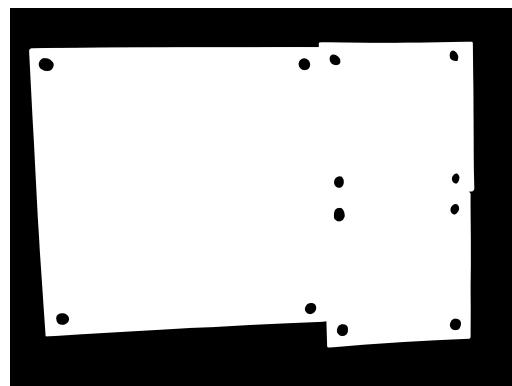


Figura 5.8: Imagen ground truth de piezas metálicas en condiciones idóneas.

Tras aplicar el método de transformación invariante de Álvarez [2], la imagen que obtenemos es la siguiente:

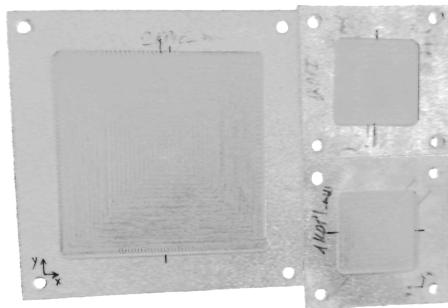


Figura 5.9: Imagen piezas metálicas en condiciones idóneas tras aplicar el algoritmo invariante de Alvarez.

A continuación podemos ver una comparación de los resultados en la figura 5.10, a la izquierda esta la imagen original segmentada mediante K-Means mientras que a la derecha esta la imagen segmentada de la previa transformación invariante.

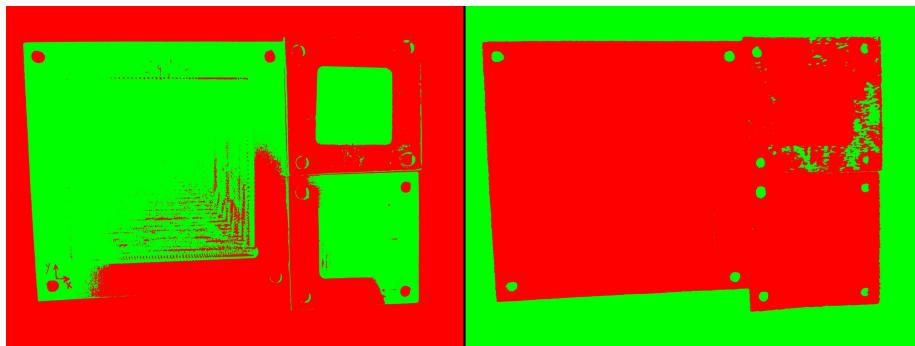


Figura 5.10: Comparación de resultados de piezas metálicas en condiciones idóneas.

Como se puede apreciar numéricamente la segmentación sobre la imagen invariante tiene una mayor precisión (98,40 % frente a 81,17 %) además visualmente la mejora es apreciable ya que la forma de la pieza se distingue perfectamente mientras que en la segmentación sobre la original no se ha identificado correctamente la pieza además de contener distintos defectos dentro de la zona delimitada como pieza.

## Resultados mejores visualmente

### Imágenes complejas

En este caso la imagen de la pieza a analizar es una imagen mas compleja, con distintos focos de luz, en un entorno que contiene mas objetos de un material metálico, en definitiva una situación mas factible en el mundo laboral.

La figura que se analizará es la siguiente la cual forma parte de las imágenes que ofrece la propia aplicación para probar los distintos algoritmos, esta imagen es 1\_IGU\_2\_ACE\_LIS\_2:



Figura 5.11: Imagen piezas metálicas en condiciones adversas.

La imagen ground truth que se ha creado para comparar los resultados de una segmentación perfecta es la de la figura 5.12.



Figura 5.12: Imagen ground truth de piezas metálicas en condiciones adversas.

Tras aplicar el método de transformación invariante de Álvarez [2], la imagen que obtenemos es la siguiente:

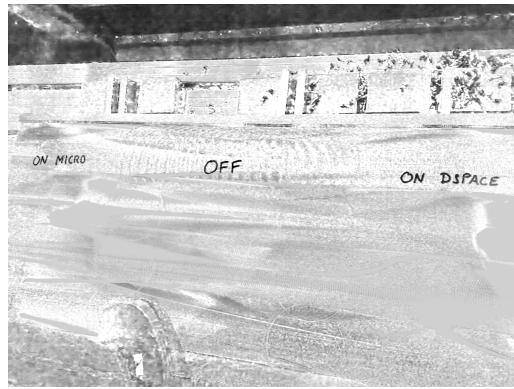


Figura 5.13: Imagen piezas metálicas en condiciones adversas tras aplicar el algoritmo invariante de Alvarez.

A continuación podemos ver una comparación de los resultados en la figura 5.14, a la izquierda esta la imagen original segmentada mediante K-Means mientras que a la derecha esta la imagen segmentada de la previa transformación invariante.

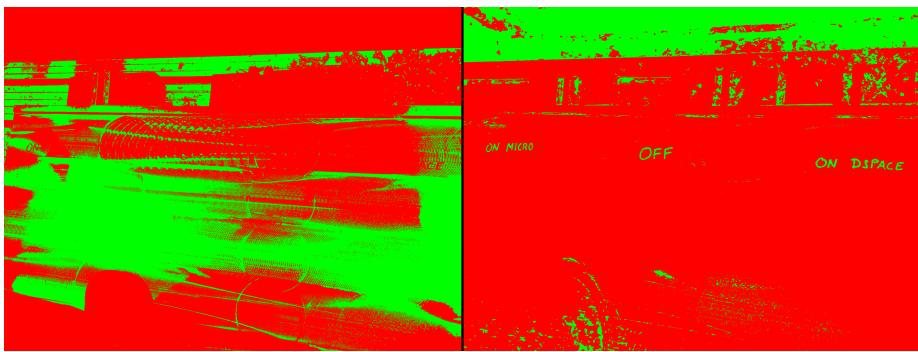


Figura 5.14: Comparación de resultados de piezas metálicas en condiciones adversas.

Esta vez, numéricamente la segmentación es mejor sin haber aplicado el algoritmo invariante (64,73 % frente a 23,20 %) pero si nos fijamos, visualmente el resultado de la transformación invariante ha conseguido eliminar una gran cantidad de brillos e imperfecciones los cuales la pieza original presentaba. Hay que mencionar además que el resultado puede parecer bajo debido a la forma de calcular el porcentaje de acierto, puesto que asigna

como fondo al color de la segmentación mas común en los bordes de la imagen ya que la pieza suele estar en el centro. Esta forma de calcularlo claramente se ve que perjudica en gran medida a esta ejecución en concreto puesto que al haber mas parte de pieza en la parte inferior de la imagen incluyendo los bordes esta calculando el acierto de una forma incorrecta.

### **Imágenes con poca luz**

En este caso la imagen de la pieza a analizar es una imagen simple en un entorno con muy poca luz.

La figura que se analizará es la siguiente la cual forma parte de las imágenes que ofrece la propia aplicación para probar los distintos algoritmos, esta imagen es 1\_IGU\_1\_ACE\_TRO\_16\_ADV:

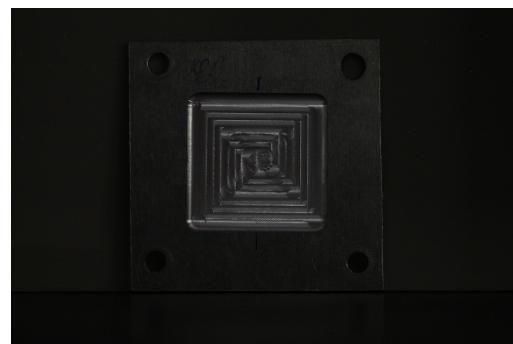


Figura 5.15: Imagen piezas metálicas en condiciones de poca luz.

La imagen ground truth que se ha creado para comparar los resultados de una segmentación perfecta es la de la figura 5.16.

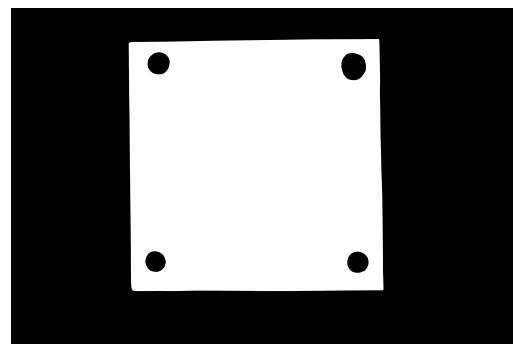


Figura 5.16: Imagen ground truth de piezas metálicas en condiciones de poca luz.

Tras aplicar el método de transformación invariante de Álvarez [2], la imagen que obtenemos es la siguiente:

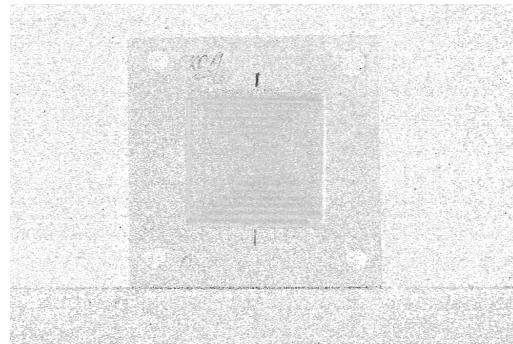


Figura 5.17: Imagen piezas metálicas en condiciones de poca luz tras aplicar el algoritmo invariante de Alvarez.

A continuación podemos ver una comparación de los resultados en la figura 5.18, a la izquierda esta la imagen original segmentada mediante HMRF\_EM mientras que a la derecha esta la imagen segmentada de la previa transformación invariante.

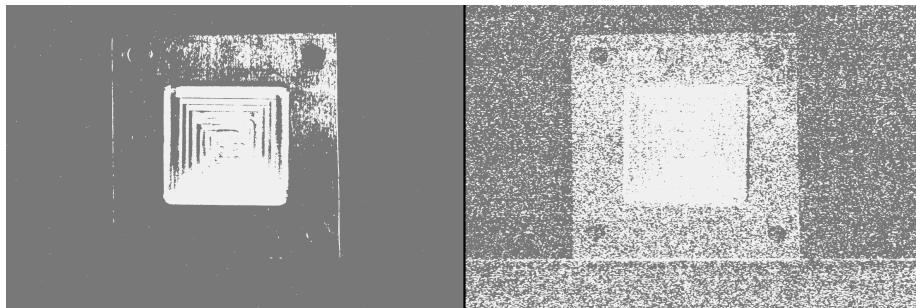


Figura 5.18: Comparación de resultados de piezas metálicas en condiciones de poca luz.

Esta vez, numéricamente la segmentación es mejor sin haber aplicado el algoritmo invariante (76,96 % frente a 74,14 %) pero si nos fijamos, visualmente el resultado de la transformación invariante ha conseguido identificar mas parte de la pieza, partes que no se alcanzaban a identificar sin haber realizado la transformación invariante. Se puede observar además que visualmente es muy fácil identificar la pieza en la imagen resultante de la segmentación tras el algoritmo de transformación invariante ya que al contrario del otro resultado esta no pierde información relevante.



---

## **6. Trabajos relacionados**

---

La utilización de métodos de transformación invariante no es un concepto novedoso, dado que modelos como el Análisis de Componentes Principales (PCA) se desarrollaron a principios del siglo XX. Sin embargo, no fue hasta hace un par de décadas que se han producido avances significativos en este campo, impulsados por el auge de las técnicas de aprendizaje profundo y el aumento en la capacidad de procesamiento computacional. Estos desarrollos han permitido mejorar considerablemente la precisión y eficiencia de los métodos de transformación invariante en diversas aplicaciones científicas y tecnológicas.

### **6.1. Artículos científicos**

#### **Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces**

Se trata del primer artículo publicado sobre el uso del Análisis de Componentes Principales (PCA) en imágenes, fechado en 1991. En él los autores, Kirby y Sirovich, demostraron cómo PCA podía aplicarse para la representación eficiente de rostros humanos. En este trabajo, propusieron un método para reducir la dimensionalidad de imágenes faciales, permitiendo su compresión y reconstrucción con alta fidelidad.

#### **PCA based Computation of Illumination-Invariant Space for Road Detection**

Este artículo describe un método que emplea el Análisis de Componentes Principales (PCA) para transformar imágenes de carreteras en un espacio

invariante a la iluminación, facilitando así la detección precisa de las vías. Utilizando PCA, los autores logran minimizar los efectos de variaciones de iluminación como sombras y cambios en la intensidad de la luz, mejorando la robustez de los algoritmos de detección de carreteras en condiciones adversas, y ofreciendo una solución eficaz para vehículos autónomos y sistemas de asistencia al conductor.

### **Road Detection Based on Illuminant Invariance**

Este artículo es publicado en 2011 por José M. Álvarez y Antonio M. López, presenta un método innovador para detectar la superficie de la carretera utilizando una cámara a bordo. Este enfoque es robusto frente a variaciones de iluminación y sombras, lo cual es un desafío común en la detección de carreteras. Los autores combinan un espacio de características invariante a sombras con un clasificador basado en modelos, mejorando la adaptabilidad del algoritmo a las condiciones de iluminación actuales y la presencia de otros vehículos. Los experimentos muestran que el método propuesto es eficaz y supera a los algoritmos basados en HSI en condiciones difíciles

### **Illumination Invariant Imaging: Applications in Robust Vision-based Localisation, Mapping and Classification for Autonomous Vehicles**

Este artículo es publicado en 2014 por Will Maddern y colaboradores, propone una técnica que usa un espacio de color invariante a la iluminación para mejorar la localización, el mapeo y la clasificación de escenas en vehículos autónomos. La técnica se basa en propiedades espectrales de la cámara y demuestra mayor consistencia en imágenes diurnas comparado con imágenes RGB. Aplicaciones incluyen localización métrica 6-DoF, mapeo estereoscópico a largo plazo y clasificación de escenas urbanas.

### **Visual Road Following Using Intrinsic Images**

Este artículo es publicado en 2015 por Tomáš Krajník y colaboradores, presenta un método que utiliza imágenes invariantes a la iluminación para el seguimiento de caminos por robots móviles en entornos exteriores. Este enfoque combina procesamiento de imágenes para eliminar sombras con un algoritmo de seguimiento de caminos, permitiendo la navegación autónoma en diversas condiciones de iluminación. Los experimentos demostraron que

el robot pudo identificar y seguir caminos en parques urbanos, incluso con sombras presentes

### **Lighting Invariant Urban Street Classification**

Este artículo es publicado en 2014 por Ben Upcroft y colaboradores, propone el uso híbrido de imágenes RGB e invariantes a la iluminación para clasificar escenas urbanas a pesar de las variaciones en las condiciones de luz. La técnica mejora la robustez de la clasificación mediante la aplicación de transformaciones invariantes a la iluminación en las imágenes antes de calcular las características de los superpíxeles. La evaluación se realizó utilizando los conjuntos de datos KITTI y un conjunto de datos propio, mostrando mejoras significativas en la clasificación bajo condiciones de iluminación adversas.

### **The Retinex theory of color vision**

La teoría Retinex, propuesta por Edwin Land en 1964, explica cómo el sistema visual humano percibe colores consistentemente bajo diferentes condiciones de iluminación. El término Retinex combina "retina" y "cortex", indicando que la percepción del color es resultado de procesos en el ojo y el cerebro. Land demostró que el color percibido depende del contexto y la iluminación de la escena, no solo de la luz reflejada por un objeto. Esta teoría aunque finalmente fue descartada por no alcanzar una tasa de acierto aceptable, se ha considerado el mencionarla ya que durante gran parte del proyecto se ha estado experimentado con ella.

## **6.2. Fortalezas y debilidades del proyecto**

Las principales fortalezas del proyecto son las siguientes:

- Mejora significativa en la identificación de piezas metálicas: el uso de algoritmos de transformación invariante ha demostrado una mejora considerable en la correcta identificación de piezas metálicas en comparación con la aplicación de métodos de agrupamiento sobre las imágenes originales. Esta capacidad es crucial para su aplicación en entornos industriales donde la precisión es esencial.
- Facilidad de instalación: la instalación de la aplicación resultante del proyecto es extremadamente sencilla. Los usuarios solo necesitan

descargar la última versión compatible con su sistema operativo desde GitHub, descomprimir el archivo .zip y ejecutar el programa siguiendo las instrucciones del archivo readme.txt. Esta simplicidad reduce las barreras de entrada para nuevos usuarios.

- Independencia de conexión a internet: la aplicación no requiere conexión a internet, lo que garantiza su funcionamiento en entornos laborales donde la conectividad puede ser limitada o inexistente. Esta característica asegura que la herramienta sea robusta y confiable en diversas situaciones.
- Flexibilidad en métodos de análisis: la aplicación permite seleccionar entre múltiples métodos de transformación invariante y algoritmos de agrupamiento, ofreciendo a los usuarios una mayor flexibilidad para adaptar el análisis a sus necesidades específicas.
- Visualización y comparación de resultados: los resultados de la ejecución se muestran en la misma ventana de la aplicación, ordenados de manera que resulta cómodo comparar los resultados obtenidos. Esta funcionalidad facilita la interpretación y el análisis de datos por parte del usuario.
- Memoria caché para resultados anteriores: la aplicación cuenta con una memoria caché que se utiliza para recuperar resultados de ejecuciones anteriores, lo que disminuye en gran medida los tiempos de ejecución. Además, el usuario puede obtener información detallada de cada ejecución anterior y acceder a la imagen correspondiente si así lo desea.
- Documentación integrada: la aplicación incluye documentación interna que permite a los usuarios resolver rápidamente cualquier duda que puedan tener. Esta documentación facilita el uso de la herramienta y mejora la experiencia del usuario.

Las principales debilidades del proyecto son las siguientes:

- Disponibilidad limitada a sistemas operativos: actualmente, la aplicación solo está disponible para sistemas operativos Windows y Linux. La falta de soporte para Mac limita su accesibilidad a usuarios de dicho sistema operativo.

- Desempeño en entornos muy complejos: en entornos altamente complejos, aunque los distintos métodos de transformación invariante logran minimizar considerablemente las reflexiones, si en la imagen hay múltiples objetos metálicos, la tasa de acierto disminuye considerablemente. Esta limitación afecta la eficacia de la herramienta en escenarios particularmente desafiantes.
- Dependencia de conexión eléctrica: exceptuando los dispositivos portátiles, la aplicación requiere conexión eléctrica para su funcionamiento. Esta necesidad puede ser una limitación en entornos donde la disponibilidad de energía es un problema.



---

## **7. Conclusiones y Líneas de trabajo futuras**

---

En esta sección se presentan las conclusiones obtenidas del proyecto, así como las posibles líneas de trabajo para futuras investigaciones que podrían dar continuidad al proyecto.

### **7.1. Conclusiones**

Tras la finalización del proyecto, se puede concluir que el objetivo principal de demostrar la mejora en la identificación de piezas en imágenes mediante el uso de algoritmos de transformación invariante ha sido alcanzado exitosamente. Este proyecto ha requerido la aplicación exhaustiva de una gran parte de los conocimientos adquiridos a lo largo de la carrera, desde fundamentos teóricos hasta habilidades prácticas en programación y análisis de datos.

La búsqueda y análisis del estado del arte han sido aspectos cruciales en el desarrollo del proyecto. Este proceso no solo ha permitido identificar los algoritmos más prometedores, sino que también ha fomentado el desarrollo de un pensamiento crítico esencial para la investigación científica. La capacidad de evaluar, seleccionar y descartar algoritmos basándose en sus resultados y eficiencia ha sido un aprendizaje significativo. Se han probado diversas técnicas, evaluando su rendimiento en diferentes escenarios y ajustando los parámetros necesarios para optimizar los resultados.

Además, la gestión del tiempo ha sido un aspecto en el que se ha observado una notable mejora. Al enfrentarse a la planificación y ejecución de tareas complejas, se ha demostrado la importancia de un conocimiento profundo del

tema para estimar adecuadamente los tiempos requeridos. Esta experiencia ha permitido desarrollar habilidades de gestión del tiempo más precisas y efectivas, cruciales para la realización de proyectos futuros.

El desarrollo de una aplicación que permite a los usuarios aplicar estos algoritmos de manera sencilla representa un logro significativo y un valor añadido al proyecto. Esta aplicación no solo facilita la accesibilidad y usabilidad de los algoritmos propuestos, sino que también proporciona una herramienta práctica que puede ser utilizada en diversos contextos industriales y de investigación. La interfaz amigable y la funcionalidad robusta de la aplicación aseguran que incluso usuarios con conocimientos técnicos limitados puedan beneficiarse de los avances logrados.

En resumen, el proyecto ha cumplido con sus objetivos iniciales y ha proporcionado importantes aprendizajes tanto a nivel técnico como de gestión. La combinación de investigación teórica, desarrollo práctico y aplicación real ha resultado en una experiencia integral que ha contribuido significativamente a la formación profesional y académica. Los conocimientos y habilidades adquiridos a lo largo de este proceso serán de gran utilidad en futuros desafíos profesionales.

## **7.2. Líneas de trabajo futuras**

Tras un análisis detallado de las posibles líneas a seguir, se han identificado varias mejoras que podrían ser implementadas en el futuro. Estas mejoras no se han podido realizar dentro del marco de tiempo del proyecto debido a la complejidad y el esfuerzo adicional que requieren. A continuación, se detallan estas líneas de trabajo futuras:

### **Mejora de algoritmos**

En esta sección se describen los posibles avances en la investigación, implementación y optimización de los algoritmos actuales:

- Mejora del segmentado: Ampliar las capacidades del algoritmo de segmentación para permitir una diferenciación más precisa, no solo entre la pieza y el fondo, sino también identificando las zonas fresadas de la pieza. Esta mejora permitiría una mayor precisión en la identificación de detalles específicos de las piezas, aumentando la exactitud del reconocimiento.

- Identificación en imágenes complejas: optimizar los algoritmos para mejorar la identificación de piezas específicas en imágenes de gran complejidad. Esto incluye la capacidad de reconocer piezas en entornos con múltiples objetos y fondos variados, facilitando su utilización en situaciones reales.

## Mejoras de experiencia de usuario

Esta sección aborda las posibles mejoras relacionadas con la funcionalidad y la interfaz de usuario de la aplicación desarrollada:

- Soporte para dispositivos con sistema operativo Mac: ampliar la compatibilidad de la aplicación para incluir dispositivos que operen bajo el sistema operativo Mac. Esto permitirá a un mayor número de usuarios acceder y utilizar la herramienta, aumentando su alcance y utilidad.
- Desarrollo de una web app: crear una aplicación web que permita a los usuarios acceder de manera más cómoda a un espacio de trabajo personal. Esta plataforma en línea ofrecería una mayor flexibilidad y accesibilidad, permitiendo a los usuarios trabajar desde cualquier lugar y dispositivo con acceso a internet.
- Implementación de concurrencia en la aplicación: mejorar la eficiencia de la aplicación mediante la implementación de concurrencia. Esto permitiría la ejecución simultánea de múltiples tareas, mejorando el rendimiento y reduciendo los tiempos de espera para los usuarios además de permitir cancelar la ejecución en cualquier momento.



---

## Bibliografía

---

- [1] Reflexión de la luz - Wikipedia, la enciclopedia libre — es.wikipedia.org. [https://es.wikipedia.org/wiki/Reflexi%C3%B3n\\_de\\_la\\_luz](https://es.wikipedia.org/wiki/Reflexi%C3%B3n_de_la_luz). [Accessed 29-04-2024].
- [2] José M Álvarez Alvarez and Antonio M. López. Road detection based on illuminant invariance. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):184–193, 2011.
- [3] João B. Bento, Ana Lopez, Inês Pires, Luísa Quintino, and Telmo G. Santos. Non-destructive testing for wire + arc additive manufacturing of aluminium parts. *Additive Manufacturing*, 29:100782, 2019.
- [4] KP Karunakaran, Alain Bernard, S Suryakumar, Lucas Dembinski, and Georges Taillandier. Rapid manufacturing of metallic objects. *Rapid Prototyping Journal*, 18(4):264–280, 2012.
- [5] Taeyoung Kim, Yu-Wing Tai, and Sung-Eui Yoon. Pca based computation of illumination-invariant space for road detection. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 632–640, 2017.
- [6] Tomáš Krajiník, Jan Blažíček, and João M. Santos. Visual road following using intrinsic images. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2015.
- [7] Will Maddern, Alex Stewart, Colin McManus, Ben Upcroft, Winston Churchill, and Paul Newman. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In *Proceedings of the Visual Place Recognition*

- in *Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, volume 2, page 5, 2014.
- [8] Jonás Martínez. <https://github.com/jms1008/TFG-Invariantes-de-iluminacion-en-piezas-metalicas/tree/main/experimentacion>. [Accessed 26-06-2024].
  - [9] MathWorks Inc. *Fuzzy Logic Toolbox: Design and Simulate Fuzzy Logic Systems*, 2023. Version 10.3 (R2023b).
  - [10] MathWorks Inc. *MATLAB: The Language of Technical Computing*, 2023. Version 10.3 (R2023b).
  - [11] MathWorks Inc. *Statistics and Machine Learning Toolbox: Use fitgmdist for Fitting Gaussian Mixture Models*, 2023. Version 10.3 (R2023b).
  - [12] Saber Nemati, Hamed Ghadimi, Xin Li, Leslie G. Butler, Hao Wen, and Shengmin Guo. Automated defect analysis of additively fabricated metallic parts using deep convolutional neural networks. *Journal of Manufacturing and Materials Processing*, 6(6), 2022.
  - [13] Ben Upcroft, Colin McManus, Winston Churchill, Will Maddern, and Paul Newman. Lighting invariant urban street classification. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1712–1718, 2014.
  - [14] Ana Vafadar, Ferdinando Guzzomi, Alexander Rassau, and Kevin Hayward. Advances in metal additive manufacturing: A review of common processes, industrial applications, and current challenges. *Applied Sciences*, 11(3):1213, 2021.
  - [15] Quan Wang. Hmrf-em-image: implementation of the hidden markov random field model and its expectation-maximization algorithm. *arXiv preprint arXiv:1207.3510*, 2012.