



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 — 7 de julio de 2024
Tutor: Dr. Carlos López Nozal y Dr. Pedro
Latorre Carmona

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	2
A.3. Estudio de viabilidad	10
Apéndice B Especificación de Requisitos	15
B.1. Introducción	15
B.2. Objetivos generales	16
B.3. Catálogo de requisitos	16
B.4. Especificación de requisitos	19
Apéndice C Especificación de diseño	47
C.1. Introducción	47
C.2. Diseño de datos	47
C.3. Diseño procedimental	49
C.4. Diseño arquitectónico	52
Apéndice D Documentación técnica de programación	55
D.1. Introducción	55
D.2. Estructura de directorios	55
D.3. Manual del programador	56

D.4. Compilación, instalación y ejecución del proyecto	61
D.5. Pruebas del sistema	64
Apéndice E Documentación de usuario	69
E.1. Introducción	69
E.2. Requisitos de usuarios	69
E.3. Instalación	70
E.4. Manual del usuario	71
Apéndice F Anexo de sostenibilización curricular	79
F.1. Introducción	79
F.2. Desarrollo del proyecto	79
F.3. Competencias de sostenibilidad adquiridas	80
F.4. Reflexión personal	81
Bibliografía	83

Índice de figuras

A.1. Burndown del sprint 4.	5
A.2. Burndown del sprint 5.	6
A.3. Burndown del sprint 6.	7
A.4. Burndown del sprint 7.	8
A.5. Burndown del sprint 8.	9
B.1. Diagrama de casos de uso de la aplicación InvIPM.	20
C.1. Diagrama de secuencia correspondiente al caso de uso 01 gestión de la transformación de invariante de iluminación.	49
C.2. Diagrama de flujo detallado del procesamiento de datos en la aplicación.	51
C.3. Diagrama de componentes mostrando la interacción entre los distintos módulos de la aplicación.	53
D.1. Instalación Statistics and Machine Learning Toolbox	58
D.2. Descarga del repositorio mediante el entorno web.	59
D.3. Ventana principal de MATLAB App Designer.	60
D.4. Icono de compilación y ejecución de MATLAB App Designer.	61
D.5. Ventana de la aplicación compilada y ejecutada.	62
D.6. Creación de MATLAB Standalone App.	64
D.7. Resultados de la ejecución de los tests.	67
E.1. Instalación de la release desde GitHub.	70
E.2. Apartado de bienvenida de InvIMP.	72
E.3. Apartado de exploración de algoritmos de InvIMP.	73
E.4. Apartado de histórico de exploraciones de InvIMP.	77
E.5. Apartado de ayuda de InvIMP.	78

Índice de tablas

A.1. Desglose de tiempo dedicado al proyecto.	10
A.2. Costes de hardware.	11
A.3. Costes de software.	11
A.4. Costes de personal.	11
A.5. Costes totales.	12
A.6. Licencias	13
B.1. CU-1 Gestión de transformación invariante.	22
B.2. CU-2 Seleccionar imagen.	23
B.3. CU-3 Imagen del sistema.	24
B.4. CU-4 Carga automática de la imagen ground truth.	25
B.5. CU-5 Imagen propia del usuario.	26
B.6. CU-6 Seleccionar una imagen ground truth propia.	27
B.7. CU-7 Modificar imagen ground truth seleccionada.	28
B.8. CU-8 Modificar imagen seleccionada.	29
B.9. CU-9 Seleccionar algoritmo invariante.	30
B.10.CU-10 Seleccionar algoritmo de agrupamiento.	31
B.11.CU-11 Seleccionar número de centros.	32
B.12.CU-12 Almacenar resultados en caché.	33
B.13.CU-13 Recuperar resultados almacenados en caché.	34
B.14.CU-14 Guardar imágenes resultantes.	35
B.15.CU-15 Gestión de caché.	36
B.16.CU-16 Visualizar información de antiguas ejecuciones en formato tabla.	37
B.17.CU-17 Poder reordenar las columnas.	38
B.18.CU-18 Visualizar la imagen correspondiente a la fila seleccionada.	39
B.19.CU-19 Abrir el directorio de la caché.	40
B.20.CU-20 Borrar contenido de la caché.	41

B.21.CU-21 Configuración.	42
B.22.CU-22 Seleccionar el idioma.	43
B.23.CU-23 Ayuda.	44
B.24.CU-24 Documentación dentro de la aplicación.	45
B.25.CU-25 Tooltips.	46

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La fase de planificación es esencial en cualquier proyecto. Durante esta etapa, se calculan el trabajo, el tiempo y el dinero necesarios para completar el proyecto. Para ello, se analiza detalladamente cada una de las partes que lo componen, lo que permite conocer los recursos necesarios para su realización. En este anexo se describe todo este proceso.

La fase de planificación se puede dividir en dos partes:

- **La planificación temporal:** Se crea un calendario o cronograma. Aquí se calcula el tiempo necesario para completar cada una de las partes del proyecto. Se debe establecer una fecha de inicio fija y una fecha de finalización estimada, considerando la importancia de cada tarea y los requisitos necesarios para comenzar a trabajar en ellas.
- **El estudio de viabilidad:** Se enfoca en la factibilidad del proyecto y se divide en dos apartados.
 - **La viabilidad económica:** En esta parte se estiman los costos y los beneficios que podría traer la realización del proyecto.
 - **La viabilidad legal:** Aquí es donde se tiene en cuenta el contexto del proyecto y toca aquella ley que le afecta. Se deben analizar todas aquellas que afecten al proyecto.

A.2. Planificación temporal

En líneas generales, se ha seguido el marco de trabajo Scrum, aunque de manera adaptada a la situación de ser el único desarrollador, manteniendo ciertos principios clave.

- Se aplicó una estrategia de desarrollo incremental mediante iteraciones (sprints) y revisiones.
- La duración media de los sprints ha sido de dos semanas.
- Aunque no se han tenido reuniones diarias, se han realizado revisiones periódicas al final de cada sprint para evaluar el progreso y planificar el siguiente.
- Esto ha permitido mantener la esencia de la mejora continua y la entrega incremental de valor.

Cabe mencionar que no fue hasta la reunión de control del 26 de marzo cuando se empezó a reflejar el progreso en GitHub, añadiendo todas las issues de reuniones anteriores y etiquetándolas con S1, S2 y S3 en el nombre para indicar la semana correspondiente. Por ello, en GitHub todas las issues aparecen como añadidas entre el 26 y el 28 de marzo. Zube se comenzó a utilizar para organizar mejor los sprints a partir del sprint 4, debido a la recomendación de Carlos para tener un mejor control de estos.

Sprint 0 (04/03/2024 - 11/03/2024)

La reunión que dio inicio a este sprint marcó el comienzo del proyecto de fin de carrera. Inicialmente, Pedro Latorre había explicado en qué consistía la idea inicial del proyecto en una reunión previa. En esta ocasión, se profundizó más en la idea y se indicaron unas pautas a seguir para investigar el estado del arte en esta área y plantear unos objetivos a cumplir durante este primer sprint.

Los objetivos de la reunión del primer sprint fueron: crear un repositorio en GitHub para el proyecto, probar dos métodos de procesamiento de imágenes, específicamente el Análisis de Componentes Principales (PCA) y la Teoría Retinex, y guardar la imagen resultante permitiendo acceder al valor de cada píxel. También se acordó investigar y aplicar métodos de agrupamiento de datos, como K-Means, en la imagen original y en la imagen procesada para comparar los resultados.

Además, se decidió eliminar el cuarto canal en las imágenes que lo tuvieran y evaluar críticamente los resultados obtenidos. Se enfatizó que no se esperaran resultados inmediatos, sino que se debía evaluar y descartar posibles caminos de manera progresiva.

En este momento del proyecto, simplemente se crearon una serie de tareas a completar antes de la próxima semana, por lo que no se tiene un registro temporal en Zube de cuándo se completó cada tarea. Más tarde, estas tareas se pasaron a GitHub el 26/03/2024 por recomendación del tutor, ya que así se podía reflejar el trabajo realizado hasta el momento. Estas issues tienen en el nombre [S1].

Sprint 1 (11/03/2024 - 25/03/2024)

Este sprint tuvo como objetivos: mostrar la representación del histograma de los valores de los píxeles de cada canal en la imagen resultante, determinar los valores mínimos y máximos de los píxeles, y visualizar la matriz resultante utilizando a función de MATLAB 'imshow'. Se discutió que, aunque 'imshow(M, [])' normaliza la imagen para visualización, 'imwrite' guarda la imagen sin normalizar. Se acordó guardar la imagen utilizando el método 'saveas' para conservar el rango dinámico y almacenar la matriz en un archivo '.mat' con el resultado de la transformación. También se destacó la importancia de guardar una imagen que se pueda visualizar fácilmente y familiarizarse con los resultados de cada método aplicado.

Se decidió añadir los métodos de agrupamiento Fuzzy C-Means y GMM, y probar otros métodos de agrupamiento que tengan en cuenta el contexto y no solo el valor del píxel. Además, se acordó buscar información sobre Geoffrey Macthlan para profundizar en el tema.

Al igual que en el sprint 0, simplemente se crearon una serie de tareas a completar antes de la próxima semana, por lo que no se tiene un registro temporal en Zube y en GitHub. Las tareas fueron añadidas el 26/03/2024. Estas issues tienen en el nombre [S2].

Sprint 2 (25/03/2024 - 04/04/2024)

Este sprint tuvo como objetivo: trasladar los algoritmos de agrupamiento de Python a MATLAB, ya que los tiempos de ejecución en Python, el lenguaje utilizado hasta ese momento, eran muy elevados. Se decidió crear una comparación en la misma ventana, dependiendo del número de centros, para así poder observar las diferencias de manera más cómoda y rápida. Además, se acordó agregar 4-5 métodos más de transformación invariante.

También se decidió generar la imagen ground truth y evaluar el resultado mediante una máscara binaria, calculando la tasa de acierto con la fórmula $(n^{\circ} \text{ de píxeles que coinciden} / n^{\circ} \text{ total de píxeles}) * 100$, y mostrar en la ventana el valor de esta tasa.

Al igual que en el sprint 0 y sprint 1, simplemente se crearon una serie de tareas a completar antes de la próxima semana, por lo que no se tiene un registro temporal en Zube. En GitHub, las tareas fueron añadidas el 26/03/2024. Estas issues tienen en el nombre [S3].

Sprint 3 (04/04/2024 - 18/04/2024)

En la reunión de preparación de este sprint, Carlos López se unió como cotutor al proyecto a solicitud de Pedro Latorre. En el futuro, Carlos López pasaría a ser el tutor del proyecto de fin de grado.

Este sprint tuvo como objetivo familiarizarse con Overleaf y comenzar a rellenar las secciones de introducción, objetivos del proyecto y conceptos teóricos de la memoria del TFG. Además, se acordó crear una cuenta en Zube para organizar las tareas en milestones y reorganizar los directorios del proyecto. La nueva estructura incluirá un directorio 'doc' con la plantilla de documentación de TFG disponible en UBUVirtual y un directorio para el programa MATLAB, que contendrá todos los métodos utilizados, separados en distintos scripts para mejorar la modularidad, reutilización y mantenibilidad del código.

Otro objetivo fue la integración de MATLAB con una aplicación web, iniciando un pequeño proyecto en GitHub para gestionar esta integración. También se planteó el desarrollo de una pequeña aplicación de escritorio de MATLAB, que servirá para probar distintos algoritmos sobre diferentes imágenes. De esta manera, además de demostrar numéricamente que hay una mejora notable tras la aplicación de métodos invariantes sobre piezas metálicas, permitirá probar distintos métodos sobre imágenes propias de forma sencilla y visual.

En el sprint anterior, no se logró encontrar un método de agrupamiento con contexto espacial que supusiera una mejora notable en los resultados, por lo que esta issue no se cerrará y formará parte también de este cuarto sprint.

Al ser este el primer sprint en el que se incorporó Zube para gestionar las issues abiertas, aún no se dominaba completamente la herramienta. Aunque ayudó con la gestión, no se creó el sprint en Zube, por lo que no se dispone de un gráfico burndown de este sprint.

Sprint 4 (18/04/2024 - 02/05/2024)

Es en este sprint cuando se comienza a reflejar formalmente los distintos sprints en Zube. Gracias a esto, a partir de este momento, resultó más cómodo trabajar con las issues de cada sprint.

Este sprint tuvo como objetivos: mejorar las funcionalidades de la aplicación y realizar cambios menores en el texto mostrado al usuario dentro de la aplicación para evitar confusiones y corregir incoherencias. Se acordó cambiar el orden en el que se muestran los resultados de la aplicación y se discutió la importancia de eliminar las imágenes mostradas de ejecuciones anteriores.

También se añadió un selector para guardar los archivos y su correspondiente mensaje de confirmación indicando que los archivos se han guardado correctamente. Además, se buscó una solución para el problema relacionado con el sistema de directorios donde se guardan los archivos.

Finalmente, se planteó la creación de otra aplicación para poder probar con imágenes propias. Al tener una imagen ground truth, se podría obtener un porcentaje de acierto de la ejecución, lo que ayudaría a evaluar mejor los resultados obtenidos.

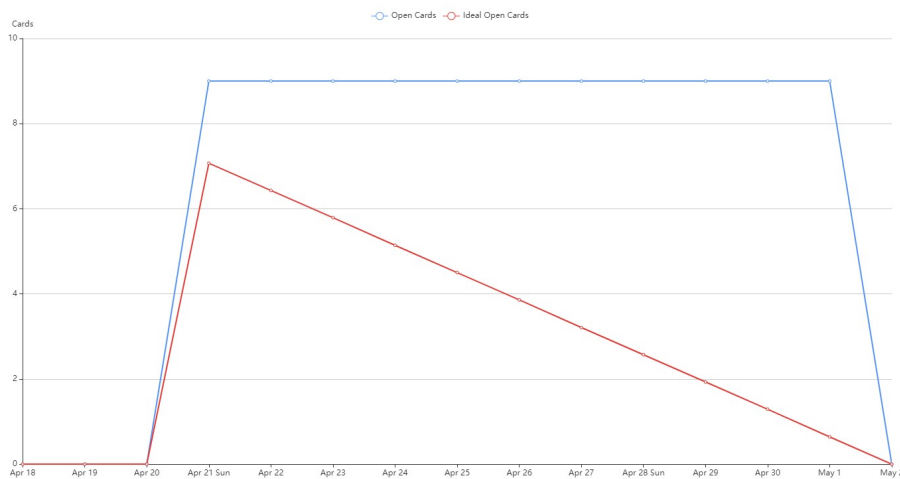


Figura A.1: Burndown del sprint 4.

Como se puede ver en el gráfico burndown [A.1](#), las issues de este sprint no fueron marcadas como completadas hasta el último día. En ese momento, no se era consciente de que debían cerrarse en lugar de dejarlas en revisión para cerrarlas después de la reunión.

Sprint 5 (02/05/2024 - 23/05/2024)

Este sprint tuvo como objetivo mejorar la experiencia del usuario y optimizar la funcionalidad de la aplicación. Se implementó una waitbar que periódicamente informaba sobre el estado de la ejecución, indicando que podría tomar mucho tiempo, además de añadir la funcionalidad de borrar resultados anteriores. También se realizaron pequeños cambios en la interfaz, como eliminar el ‘-’ de los porcentajes y añadir un botón para guardar imágenes.

Se realizaron distintas modificaciones en la estructura de la interfaz, añadiendo menús de ayuda, una lista de nombres de imágenes en caché, un menú de bienvenida y ayuda contextual. Se actualizó el nombre de los archivos de resultados, añadiendo la fecha de ejecución al inicio. Además, se corrigieron errores en la selección de imágenes desde rutas diferentes a /data y se mejoró la gestión de errores.

Se llegó a la conclusión de agregar una caché de imágenes para almacenar temporalmente los resultados, cargando ejecuciones anteriores con los mismos parámetros para reducir el coste computacional. Al cerrar la aplicación, la caché se borraría. Finalmente, se planteó la corrección de algunos errores detectados en los algoritmos, así como una mejora de la documentación y el README del proyecto.

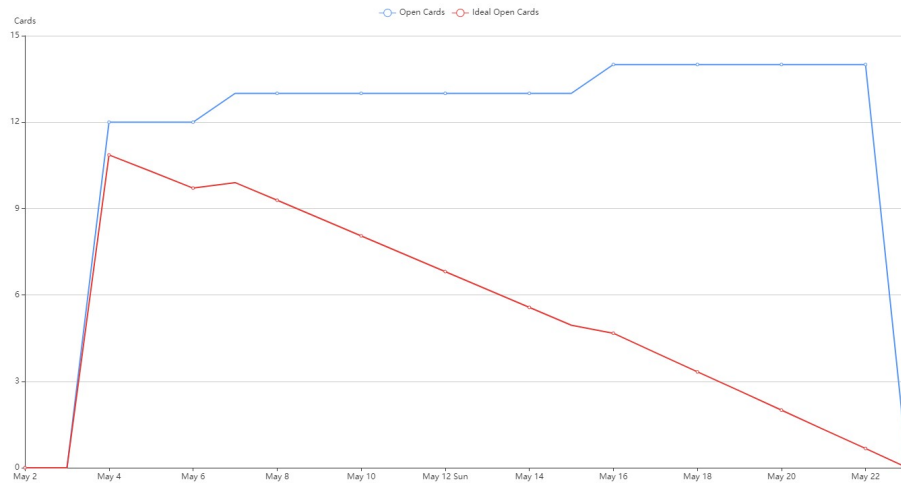


Figura A.2: Burndown del sprint 5.

Como se puede ver en el gráfico burndown [A.2](#), las issues de este sprint no fueron marcadas como completadas hasta el último día. En ese momento,

no se era consciente de que debían cerrarse en lugar de dejarlas en revisión para cerrarlas después de la reunión.

Sprint 6 (23/05/2024 - 30/05/2024)

Este sprint tuvo como objetivo mejorar la funcionalidad y usabilidad de la aplicación. Se planean diversas mejoras en la documentación en HTML y la creación de un README en la carpeta de imágenes. Además, se modificará la visualización de los nombres de las imágenes dentro de la aplicación y se añadirán tooltips para una mejor comprensión del usuario.

Se establecerán condiciones numéricas en los campos de número de centros y se añadirá un límite máximo de iteraciones en los métodos de agrupamiento. También se trabajará en la caché para hacerla persistente, permitiendo borrar el historial y mejorando su escalabilidad.

En cuanto a la evaluación de la calidad, se incorporará una medida de comparación utilizando la imagen ground truth y se mejorará la nomenclatura de los algoritmos invariantes, reemplazando los nombres vacíos con “no aplicado”. Se abordarán problemas técnicos como la minimización de la app al cargar una imagen, la eliminación de la imagen ground truth al introducir una nueva que no tenga una asociada, y se revisarán soluciones para evitar que la aplicación se detenga por errores. Además, se tratará de introducir concurrencia para manejar tareas como cancelar la ejecución.

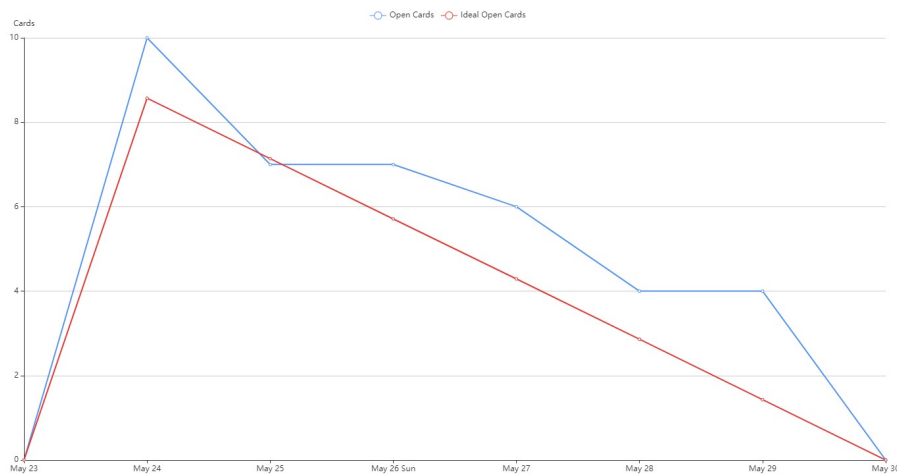


Figura A.3: Burndown del sprint 6.

Como se puede ver en el gráfico burndown [A.3](#), las issues de este sprint se fueron cerrando correctamente a medida que se completaban.

Sprint 7 (31/05/2024 - 10/06/2024)

Este sprint tuvo como objetivo preparar la primera release oficial de la aplicación. Se realizarán mejoras en la gestión de errores para asegurar una experiencia de usuario más robusta. Además, se tratará de aplicar concurrencia para permitir la cancelación de la ejecución, una tarea que no se logró completar en el sprint anterior.

Se deshabilitarán los botones durante la ejecución de procesos para evitar acciones no deseadas y se revisará el código, actualizando la documentación interna para mantener la coherencia. También se implementará soporte bilingüe, añadiendo opciones en español e inglés para hacer la aplicación accesible a un público más amplio.

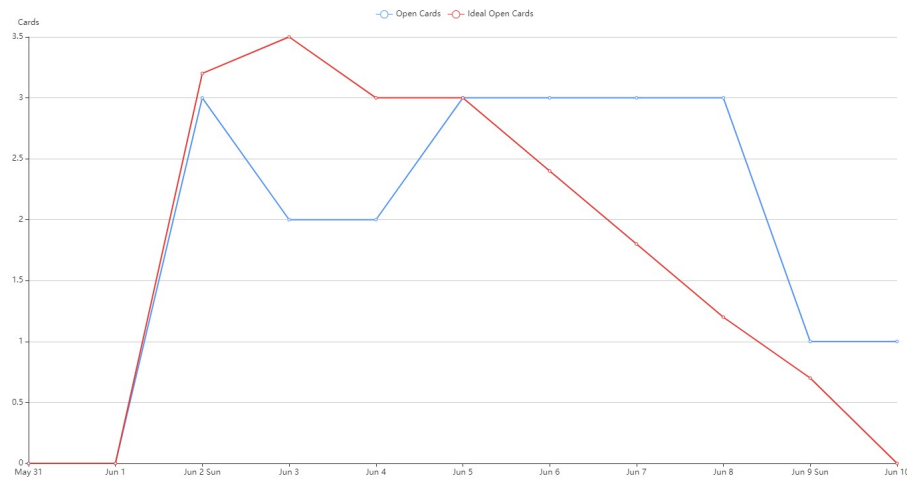


Figura A.4: Burndown del sprint 7.

Como se puede ver en el gráfico burndown [A.4](#), las issues de este sprint se fueron cerrando correctamente a medida que se completaban, a excepción de una que, por falta de tiempo, quedó pendiente para completar en el siguiente sprint.

Sprint 8 (11/06/2024 - 27/06/2024)

En este sprint, se implementaron mejoras y se preparó la aplicación para su versión oficial. Se mejoraron los tooltips, especialmente el de centros, y se añadió la opción de imprimir y visualizar imágenes en la caché, además de modificar el botón 'Abrir directorio' por 'Ver imágenes'. Se habilitó la ordenación por filas en la caché y se actualizó la imagen de bienvenida.

Para garantizar compatibilidad, la aplicación se adaptó a Linux, y se realizaron pruebas unitarias. Se creó un README detallado y se reorganizaron los directorios, eliminando los obsoletos. También se subió la versión anterior con mejoras en el manejo de errores y adaptación al inglés.

En la documentación, se añadieron citas y se revisaron correcciones en la memoria, y se redactaron el plan de proyecto, especificaciones de requisitos y diseño en los anexos.

Finalmente, se realizó la versión con todos los cambios del sprint, consolidando las mejoras para la versión final.

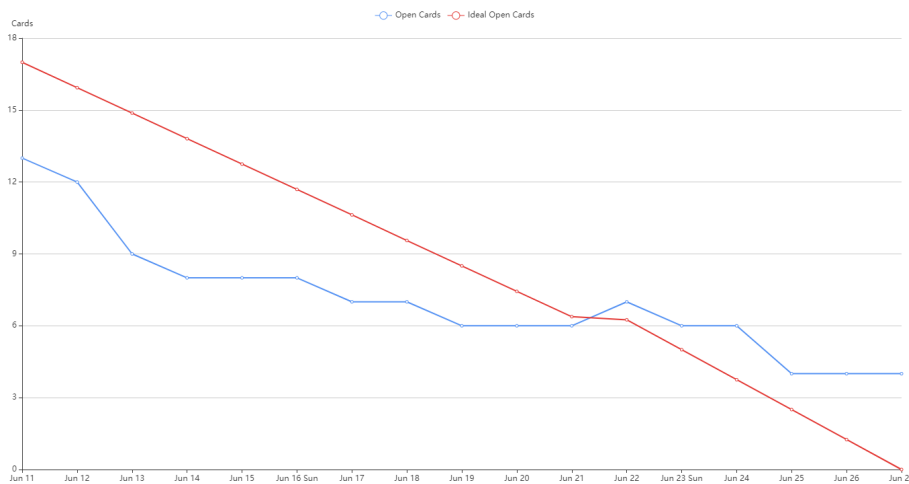


Figura A.5: Burndown del sprint 8.

Como se puede apreciar en la figura A.5, no se lograron finalizar todas las tareas planificadas para este supuesto sprint final, debido a que la documentación requirió una considerable cantidad de tiempo. Por este motivo, se acordó reunirse a mitad de semana con todas estas tareas finalizadas y con al menos una versión inicial de la documentación completada.

Sprint 9 (28/06/2024 - 01/07/2024)

Este sprint se dedicó completamente a finalizar las tareas pendientes del periodo anterior, incluyendo la elaboración de una primera versión de la documentación, tanto de la memoria como de los anexos. Además, se identificó y resolvió un error en la aplicación.

Resumen del tiempo empleado

A continuación se muestra un resumen del tiempo dedicado a la realización del proyecto.

Categoría	Tiempo (horas)
Sprint 0	20
Sprint 1	32
Sprint 2	20
Sprint 3	26
Sprint 4	49
Sprint 5	54
Sprint 6	28
Sprint 7	42
Sprint 8	71
Sprint 9	22
Total	364 horas

Tabla A.1: Desglose de tiempo dedicado al proyecto.

A.3. Estudio de viabilidad

A continuación, se realizará tanto el estudio de viabilidad económica como el de viabilidad legal.

Viabilidad económica

En este apartado se analizarán los costes y beneficios estimados de haber desarrollado el proyecto en un entorno empresarial. Cabe mencionar que dichos datos son aproximados.

Costes

A continuación, se realizará tanto el análisis de costes materiales y beneficios.

Costes materiales:

En este apartado se desglosarán todos aquellos costes materiales, tanto de hardware como de software, necesarios para el correcto desarrollo del proyecto. En ambos casos, se considera que se amortizarán a 5 años y se han utilizado durante los cuatro meses de desarrollo del proyecto.

El desglose de todos los dispositivos hardware necesarios para la realización del proyecto se encuentra a continuación en la figura A.2.

Concepto	Coste	Coste amortizado
Ordenador portátil	900,00€	60,00€
Total	900,00€	60,00€

Tabla A.2: Costes de hardware.

En cuanto a las licencias software no gratuitas se tiene el siguiente desglose en la figura A.3.

Concepto	Coste	Coste amortizado
Windows 10 Pro	259,00€	17,27€
MATLAB Academic (Anual)	262,00€	87,33€
Total	521,00€	104,60€

Tabla A.3: Costes de software.

Hay que destacar que la licencia de MATLAB utilizada ha sido la Academic que la propia Universidad de Burgos ofrece a los alumnos, la cual se paga anualmente.

Costes de personal:

El proyecto lo ha llevado a cabo un ingeniero informático recién graduado a tiempo completo durante cuatro meses. Se considera que el salario bruto anual es de 24.000,00€ [6], el IRPF sobre el salario bruto es del 13,55 % y la Seguridad Social es del 6,35 % respectivamente como se puede ver en la figura A.4. Se considera que la cantidad de horas de trabajo anuales en España a tiempo completo es de 1820 [1].

Concepto	Coste
Salario anual en bruto	24.000,00€
IRPF	3.251,04€
Seguridad Social	1.524,00€
Salario anual neto	19.224,96€
Salario neto por hora	10,56€
Salario 364 horas	3.844,99€

Tabla A.4: Costes de personal.

Costes totales:

El sumatorio de todos los distintos tipos de costes se encuentra en la figura A.5.

Concepto	Coste
Hardware	60,00€
Software	104,60€
Personal	3.844,99€
Total	4.009,59€

Tabla A.5: Costes totales.

Beneficios

La aplicación desarrollada se distribuirá de forma libre y sin restricciones, permitiendo su uso sin limitaciones adicionales. La única condición necesaria para su funcionamiento será disponer de MATLAB. En una fase posterior, se evaluará la posibilidad de incorporar características adicionales que mejoren la experiencia del usuario, manteniendo siempre el enfoque en la accesibilidad y la libertad de uso.

Viabilidad legal

Apartado dedicado al marco legal que envuelve el proceso de desarrollo del proyecto, su distribución, modificación y utilización. No es necesario atender al uso de licencias de terceros, pues todo el proyecto ha sido desarrollado con la licencia MATLAB Academic proporcionada por la Universidad de Burgos para desarrolladores individuales o estudiantes y bajo una finalidad no comercial. En concreto, la licencia utilizada ha sido:

- **MATLAB Academic:** Al tratarse de un desarrollador individual, el uso de la licencia MATLAB Academic está justificado. Esta licencia es distribuida de forma gratuita por la Universidad de Burgos a sus alumnos.

Debido al uso de distintos papers y toolboxes de MATLAB la aplicación esta limitada por las licencias de estos. Las licencias son las siguientes:

Elemento	Licencia
Statistics and Machine Learning Toolbox	MathWorks Software License Agreement (MWL)
Fuzzy Logic Toolbox	MathWorks Software License Agreement (MWL)
MATLAB Compiler	MathWorks Software License Agreement (MWL)
HMRP-EM-image	BSD 3-Clause License
Road Detection Based on Illuminant Invariance	IEEE Terms and Conditions
Illumination Invariant Imaging: Applications in Robust Vision-based Localisation, Mapping and Classification for Autonomous Vehicles	IEEE Terms and Conditions
Visual Road Following Using Intrinsic Images	IEEE Terms and Conditions
Lighting Invariant Urban Street Classification	IEEE Terms and Conditions
PCA based Computation of Illumination-Invariant Space for Road Detection	IEEE Terms and Conditions

Tabla A.6: Licencias

El proyecto actual ha sido asignado bajo la licencia BSD 3-Clause. Esto implica que cualquier usuario es libre de compartir, redistribuir y adaptar la aplicación, siempre y cuando se cumplan las siguientes condiciones:

- **Atribución:** Se debe dar crédito apropiado al autor original, manteniendo el aviso de copyright, la lista de condiciones y la exención de responsabilidad en todas las distribuciones.
- **Redistribución:** Las redistribuciones en formato de código fuente deben incluir el aviso de copyright, esta lista de condiciones y la exención de responsabilidad.
- **Restricción de Uso de Nombre:** Ni el nombre de los autores ni los nombres de sus colaboradores pueden ser utilizados para apoyar o promocionar productos derivados sin un permiso específico por escrito.

Además, el uso de MATLAB Runtime y ciertas herramientas de MATLAB en este proyecto está sujeto al MathWorks Software License Agreement (MWL). Los usuarios deben tener una licencia válida de MathWorks para ejecutar este software. La redistribución de MATLAB Runtime o cualquier toolbox de MathWorks no está permitida. Los usuarios deben descargar e instalar MATLAB Runtime desde el sitio oficial de MathWorks de acuerdo con los términos de su licencia según se indica en https://www.mathworks.com/company/aboutus/policies_statements/.

Esta combinación de licencias asegura que la aplicación mantenga su accesibilidad y flexibilidad, fomentando la colaboración y el desarrollo comunitario mientras protege los derechos del autor original y cumple con las restricciones de las herramientas utilizadas.

Apéndice B

Especificación de Requisitos

B.1. Introducción

Este anexo recoge la especificación de requisitos que define el comportamiento del sistema desarrollado. Se han seguido las recomendaciones del estándar IEEE 830-1998 [3], que establece que una buena especificación de requisitos software debe ser:

- **Completa:** Todos los requerimientos deben estar incluidos y todas las referencias definidas.
- **Consistente:** Debe ser coherente tanto internamente como con otros documentos de especificación.
- **Inequívoca:** La redacción debe ser clara para evitar malentendidos.
- **Correcta:** El software debe cumplir con los requisitos especificados.
- **Trazable:** Debe ser posible rastrear la historia, ubicación o aplicación de un ítem a través de su identificación documentada.
- **Priorizable:** Los requisitos deben poder organizarse jerárquicamente según su relevancia para el negocio, clasificándolos en esenciales, condicionales y opcionales.
- **Modificable:** Los requerimientos deben ser fácilmente modificables.
- **Verificable:** Debe existir un método finito y sin costo para probar los requisitos.

B.2. Objetivos generales

El proyecto tiene los siguientes objetivos generales.

- Demostrar empíricamente que al aplicar métodos de transformación invariante de la iluminación, se consiguen mejores resultados a la hora de identificar piezas metálicas mediante métodos de agrupamiento de imágenes.
- Desarrollar una aplicación de escritorio que tome como entrada una imagen y permita al usuario evaluar los resultados de la segmentación para la identificación de piezas metálicas. La aplicación debe ofrecer diferentes algoritmos tanto de transformación invariante (propuestos por Álvarez [4], Maddern [9], Krajník [8], Upcroft [14] y PCA [7]) como de agrupamiento de imágenes (K-Means [11], Fuzzy C-Means [10], GMM [12] y agrupamiento con información espacial [15]).
- Desarrollar una aplicación de escritorio que permita evaluar la mejora que supone la aplicación de los métodos de transformación invariante en procesos de segmentación de imágenes, mostrando una comparación de los resultados tanto visual como numérica.
- Guardar las imágenes resultantes del método de transformación invariante y las dos imágenes resultados de la segmentación, una aplicando la transformación de invariantes de iluminación y otra aplicando la segmentación directamente a la imagen original.
- Comparar los resultados de distintos métodos de transformación invariante de imágenes y argumentar la mejora que se obtiene en la identificación de piezas metálicas.

B.3. Catálogo de requisitos

En este apartado se enumeran los requisitos específicos que se derivan de los objetivos generales del proyecto.

Requisitos funcionales

A continuación, se enumerarán los requisitos funcionales.

- **RF-1 Gestión de transformación invariante:** La aplicación debe ser capaz de gestionar la transformación invariante sobre imágenes.

- **RF-1.1 Seleccionar imagen:** El usuario debe poder seleccionar una imagen sobre la cual aplicar las diferentes transformaciones.
 - **RF-1.1.1 Imagen del sistema:** El usuario debe poder seleccionar una imagen de prueba contenida en el programa.
 - ◊ **RF-1.1.1.1 Carga automática de la imagen ground truth:** Al haber seleccionado una imagen contenida en el programa, esta tendrá asociada una imagen ground truth.
 - **RF-1.1.2 Imagen propia del usuario:** El usuario debe poder seleccionar una imagen propia. Al no tener imagen ground truth asociada esta no se cargará automáticamente.
 - ◊ **RF-1.1.2.1 Seleccionar una imagen ground truth propia:** Debido a que la imagen que el usuario ha seleccionado es una imagen propia, el sistema permitirá aportar su correspondiente imagen ground truth.
 - ◊ **RF-1.1.2.2 Modificar imagen ground truth seleccionada:** El usuario debe poder modificar la imagen ground truth proporcionada en el caso de que se haya seleccionado una imagen propia.
 - **RF-1.1.3 Modificar imagen seleccionada:** El usuario debe poder seleccionar una imagen diferente a la seleccionada anteriormente sobre la cual aplicar las diferentes transformaciones.
- **RF-1.2 Seleccionar algoritmo invariante:** El usuario debe poder seleccionar el algoritmo de transformación invariante que desee de la lista de algoritmos que la aplicación ofrece.
- **RF-1.3 Seleccionar algoritmo de agrupamiento:** El usuario debe poder seleccionar el algoritmo de agrupamiento invariante que desee de la lista de algoritmos que la aplicación ofrece.
- **RF-1.4 Seleccionar número de centros:** El usuario debe poder seleccionar el numero de centros que desee como parámetro del algoritmo de agrupamiento seleccionado dentro de unos limites preestablecidos.
- **RF-1.5 Almacenar resultados en caché:** La aplicación debe guardar en caché los resultados de la ejecución.
- **RF-1.6 Recuperar resultados almacenados en caché:** La aplicación recuperar los resultados de antiguas ejecuciones con los mismos parámetros para ahorrar costes computacionales.

- **RF-1.7 Guardar imágenes resultantes:** El usuario debe poder guardar los resultados de la ejecución en el directorio que desee.
- **RF-2 Gestión de caché:** La aplicación debe ser capaz de gestionar correctamente una memoria caché donde almacenar y mostrar los resultados de las ejecuciones del usuario.
 - **RF-2.1 Visualizar información de antiguas ejecuciones en formato tabla:** La aplicación debe mostrar en una tabla una lista de los resultados de las distintas ejecuciones almacenadas en caché.
 - **RF-2.1.1 Poder reordenar las columnas:** El usuario debe poder reordenar la tabla de la caché en función de los datos contenidos en las diferentes columnas.
 - **RF-2.1.2 Visualizar la imagen correspondiente a la fila seleccionada:** El usuario debe poder acceder en específico a la imagen a la que hace alusión cada fila de la tabla caché.
 - **RF-2.2 Abrir el directorio de la caché:** El usuario debe poder acceder al directorio de la caché.
 - **RF-2.3 Borrar contenido de la caché:** El usuario debe poder borrar el contenido de la caché.
- **RF-3 Configuración:** El usuario debe poder configurar los parámetros disponibles de la aplicación.
 - **RF-3.1 Seleccionar el idioma:** El usuario debe poder seleccionar el idioma de la aplicación dependiendo si desea trabajar con esta en español o en inglés.
- **RF-4 Ayuda:** El usuario debe poder obtener ayuda sobre los distintos elementos de la aplicación.
 - **RF-4.1 Documentación dentro de la aplicación:** La aplicación debe contener distintos apartados dedicados a documentar el funcionamiento y métodos aplicados en la aplicación.
 - **RF-4.2 Tooltips:** La aplicación debe proporcionar información sobre la funcionalidad de las distintas acciones dentro de esta.

Requisitos no funcionales

A continuación, se enumerarán los requisitos no funcionales.

- **RNF-1 Usabilidad:** La aplicación debe tener una interfaz de usuario intuitiva y fácil de usar, con una curva de aprendizaje mínima.
- **RNF-2 Mensajes de error explicativos:** La aplicación debe proporcionar mensajes de error claros y explicativos que ayuden al usuario a entender y resolver el problema.
- **RNF-3 Rendimiento:** La aplicación debe tener unos tiempos de carga aceptables.
- **RNF-4 Disponibilidad:** La aplicación debe estar preparada para funcionar en cualquier localización y sin acceso a internet.
- **RNF-5 Soporte:** La aplicación debe dar soporte a versiones de iguales o superiores a MATLAB 2023b.
- **RNF-6 Uso de recursos:** La aplicación debe ser eficiente en el uso de recursos del sistema, reutilizando los resultados de ejecuciones anteriores para reducir costes computacionales.
- **RNF-7 Modularidad:** La aplicación debe estar diseñada de manera modular, permitiendo la fácil adición o modificación de funcionalidades sin afectar otras partes del sistema.
- **RNF-8 Internacionalización:** La aplicación debe estar preparada para soportar varios idiomas.
- **RNF-9 Compatibilidad:** La aplicación debe poderse ejecutar en distintos sistemas operativos. Por lo menos en Linux y en Windows.

B.4. Especificación de requisitos

En esta sección se mostrará el diagrama de casos de uso resultante de los requisitos funcionales y se desarrollará cada uno de ellos.

Diagrama de casos de uso

A continuación, se muestra en la figura [B.1](#) el diagrama de casos de uso.

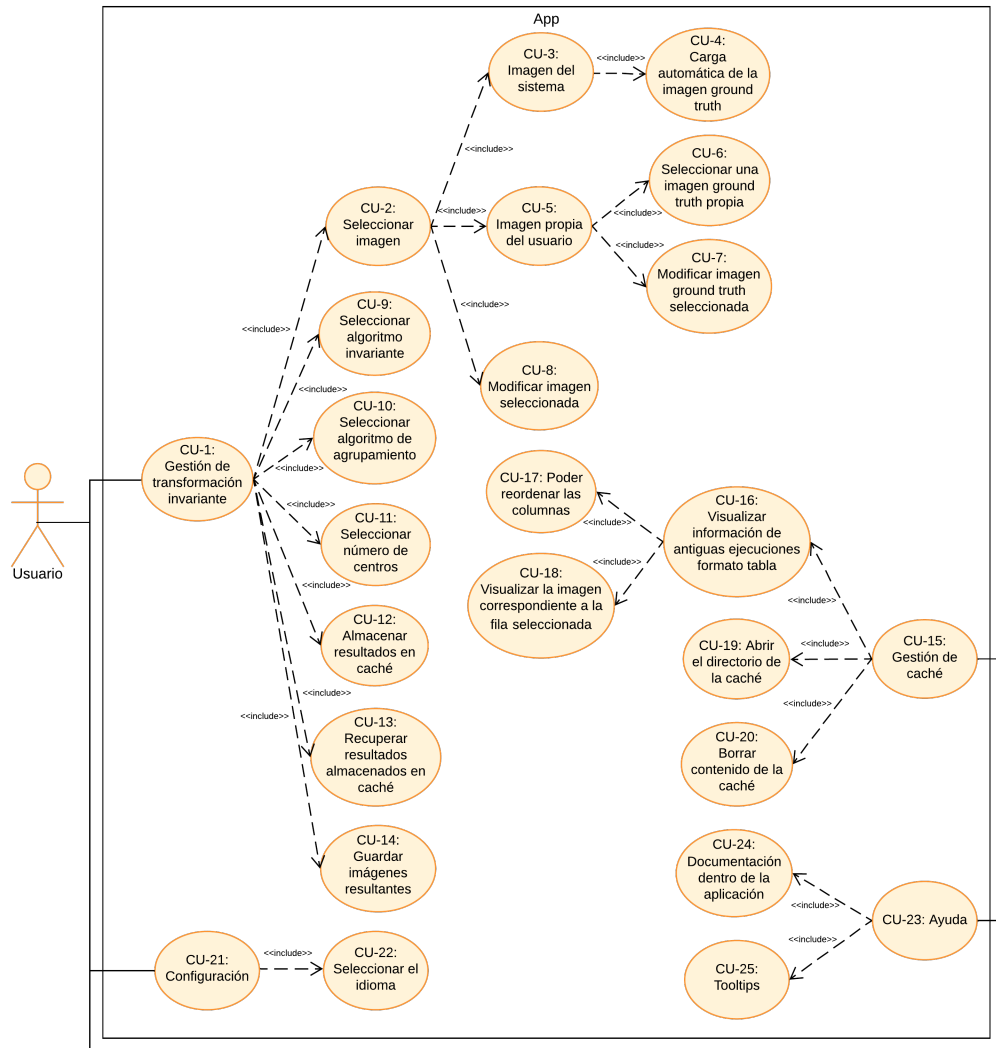


Figura B.1: Diagrama de casos de uso de la aplicación InvIPM.

Actores

El sistema está diseñado para la interacción de un único actor, identificado como el usuario. Este usuario idealmente será personal especializado en el análisis de la calidad de piezas metálicas a través de imágenes, generalmente asociado a una empresa. No obstante, la aplicación ha sido desarrollada con la capacidad de instruir a usuarios que no posean los conocimientos necesarios, a través de una documentación integrada y accesible dentro de la propia aplicación.

Casos de uso

A continuación, se definirán los casos de uso.

CU-1	Gestión de transformación invariante
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1, RF-1.1, RF-1.1.1, RF-1.1.1.1, RF-1.1.2, RF-1.1.2.1, RF-1.1.2.2, RF-1.1.3, RF-1.2, RF-1.3, RF-1.4, RF-1.5, RF-1.6, RF-1.7
Descripción	Permite al usuario aplicar sobre una imagen los distintos métodos de transformación invariante y guardar los resultados.
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa en la pestaña de exploración de algoritmos 2. El usuario selecciona una imagen de pieza metálica. 3. El usuario selecciona un algoritmo de transformación invariante. 4. El usuario selecciona un algoritmo de agrupamiento. 5. El usuario selecciona un numero de centros (por defecto 2). 6. El usuario hace click en el botón de ejecutar. 7. El programa ejecuta sobre la imagen seleccionada el algoritmo invariante seleccionado y mas tarde el algoritmo de agrupamiento sobre la original y sobre la invariante. 8. En el caso de que tenga imagen ground truth calcula el porcentaje de acierto. 9. El programa guarda en la cache los resultados. 10. El programa muestra cuatro imágenes tras finalizarse la ejecución: <ul style="list-style-type: none"> ■ Imagen original. ■ Imagen original segmentada. ■ Imagen invariante. ■ Imagen invariante segmentada. 11. El usuario hace click en el botón de guardar resultados y selecciona un directorio donde guardar las imágenes.
Postcondición	Se muestran los resultados de la transformación invariante y las segmentaciones indicando en el caso de que haya imagen ground truth la tasa de acierto.
Excepciones	Error (mensaje)
Importancia	Alta

Tabla B.1: CU-1 Gestión de transformación invariante.

CU-2	Seleccionar imagen
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.1, RF-1.1.1, RF-1.1.1.1, RF-1.1.2, RF-1.1.2.1, RF-1.1.2.2, RF-1.1.3
Descripción	Permite al usuario seleccionar una imagen sobre la que aplicar los distintos métodos de transformación invariante.
Precondición	La aplicación se encuentra disponible
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa en la pestaña de exploración de algoritmos 2. El usuario hace click en el botón de cargar imagen de pieza metálica. 3. Se muestra el directorio de imágenes del programa. 4. El usuario selecciona una imagen del programa o propia. 5. El usuario hace click en el botón de aceptar. <ul style="list-style-type: none"> ■ En el caso de seleccionar una imagen del programa, la aplicación busca su correspondiente imagen ground truth. ■ En el caso de seleccionar una imagen propia, se notifica al usuario de que al no ser una imagen del programa esta no tiene imagen ground truth asociada. 6. Si no hay ningún error: Se guardan en memoria la imagen y se muestra en sus correspondiente plot y si es una imagen del programa también la imagen ground truth.
Postcondición	<ol style="list-style-type: none"> 1. La imagen seleccionada se ha cargado correctamente. 2. Activa la interacción de los selectores de método invariante, método de agrupamiento, numero de centro y el botón de ejecutar.
Excepciones	<p>Se ha cancelado la selección de imágenes. (mensaje).</p> <p>Image selection has been canceled. (mensaje).</p> <p>Imagen ground truth no encontrada. (mensaje).</p> <p>Ground truth image not found (mensaje).</p>
Importancia	Alta

Tabla B.2: CU-2 Seleccionar imagen.

CU-3	Imagen del sistema
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.1.1, RF-1.1.1.1
Descripción	El usuario debe poder seleccionar una imagen de prueba contenida en el programa
Precondición	La aplicación se encuentra disponible. El usuario ha hecho click en el botón de cargar imagen de pieza metálica.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona una imagen de las proporcionadas por el sistema. 2. La aplicación carga la imagen seleccionada. 3. La aplicación busca la imagen ground truth correspondiente a la imagen seleccionada. 4. Si no hay ningún error se muestra la imagen y su correspondiente imagen ground truth.
Postcondición	La imagen seleccionada se ha cargado correctamente.
Excepciones	Se ha cancelado la selección de imágenes. (mensaje). Image selection has been canceled. (mensaje). Imagen ground truth no encontrada. (mensaje). Ground truth image not found (mensaje).
Importancia	Alta

Tabla B.3: CU-3 Imagen del sistema.

CU-4	Carga automática de la imagen ground truth
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.1.1.1
Descripción	Al haber seleccionado una imagen de la aplicación, se cargara su correspondiente imagen ground truth.
Precondición	La aplicación se encuentra disponible. El usuario ha hecho click en el botón de cargar imagen de pieza metálica. El usuario ha seleccionado una imagen del sistema.
Acciones	<ol style="list-style-type: none"> 1. La aplicación busca la correspondiente imagen ground truth de la imagen seleccionada en el directorio de imágenes ground truth. 2. Si no hay ningún error se muestra la imagen ground truth en el plot correspondiente.
Postcondición	Imagen ground truth se ha cargado correctamente.
Excepciones	Imagen ground truth no encontrada. (mensaje). Ground truth image not found (mensaje).
Importancia	Media.

Tabla B.4: CU-4 Carga automática de la imagen ground truth.

CU-5	Imagen propia del usuario
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.1.2, RF-1.1.2.1, RF-1.1.2.2
Descripción	El usuario debe poder seleccionar una imagen propia.
Precondición	La aplicación se encuentra disponible. El usuario ha hecho click en el botón de cargar imagen de pieza metálica.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona una imagen propia. 2. La aplicación carga la imagen seleccionada. 3. La aplicación busca informa al usuario que al ser una imagen propia, esta no tiene imagen ground truth asociada. 4. Si no hay ningún error se muestra la imagen. 5. La aplicación activa la interacción del botón de cargar una imagen ground truth propia.
Postcondición	La imagen seleccionada se ha cargado correctamente.
Excepciones	Se ha cancelado la selección de imágenes. (mensaje). Image selection has been canceled. (mensaje). Imagen ground truth no encontrada. (mensaje). Ground truth image not found (mensaje).
Importancia	Alta

Tabla B.5: CU-5 Imagen propia del usuario.

CU-6	Seleccionar una imagen ground truth propia
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.1.2.1
Descripción	La descripción del CU
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen propia.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el botón de cargar imagen ground truth. 2. La aplicación abre un directorio para que el usuario seleccione el archivo que desee. 3. El usuario selecciona la imagen que quiere utilizar como ground truth. 4. Si no hay ningún error se muestra la imagen ground truth.
Postcondición	Imagen ground truth se ha cargado correctamente.
Excepciones	Error al leer la imagen ground truth: (mensaje). Error reading ground truth image: (mensaje).
Importancia	Media

Tabla B.6: CU-6 Seleccionar una imagen ground truth propia.

CU-7	Modificar imagen ground truth seleccionada
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.1.2.2
Descripción	El usuario debe poder modificar la imagen ground truth proporcionada en el caso de que se haya seleccionado una imagen propia.
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen propia. El usuario ha cargado una imagen ground truth propia.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el botón de cargar imagen ground truth. 2. La aplicación abre un directorio para que el usuario seleccione el archivo que desee. 3. El usuario selecciona la imagen que quiere utilizar como ground truth. 4. La aplicación elimina de memoria la imagen ground truth anterior y guarda en su lugar la nueva. 5. Si no hay ningún error se muestra la imagen ground truth sustituyendo a la anterior.
Postcondición	Imagen ground truth nueva se ha cargado correctamente.
Excepciones	Error al leer la imagen ground truth: (mensaje). Error reading ground truth image: (mensaje).
Importancia	Media

Tabla B.7: CU-7 Modificar imagen ground truth seleccionada.

CU-8	Modificar imagen seleccionada
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.1.3
Descripción	El usuario debe poder seleccionar una imagen diferente a la seleccionada anteriormente
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen anteriormente.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el botón de cargar imagen de pieza metálica. 2. Se muestra el directorio de imágenes del programa. 3. El usuario selecciona una imagen del programa o propia. 4. El usuario hace click en el botón de aceptar. <ul style="list-style-type: none"> ■ En el caso de seleccionar una imagen del programa, la aplicación busca su correspondiente imagen ground truth. ■ En el caso de seleccionar una imagen propia, se notifica al usuario de que al no ser una imagen del programa esta no tiene imagen ground truth asociada. 5. Si no hay ningún error se dejan de mostrar las imágenes tanto de entrada como de salida de anteriores ejecuciones, se boquea la interacción del botón de guardar resultados y: <ul style="list-style-type: none"> ■ En el caso de seleccionar una imagen del programa, se guardan en memoria tanto la imagen como la imagen ground truth y ambas se muestran en sus respectivos plots. ■ En el caso de seleccionar una imagen propia, se guardan en memoria la imagen y se muestra en sus correspondiente plot.
Postcondición	La imagen seleccionada se ha cargado correctamente.
Excepciones	Se ha cancelado la selección de imágenes. (mensaje). Image selection has been canceled. (mensaje). Imagen ground truth no encontrada. (mensaje). Ground truth image not found (mensaje).
Importancia	Alta

Tabla B.8: CU-8 Modificar imagen seleccionada.

CU-9	Seleccionar algoritmo invariante
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.2
Descripción	El usuario debe poder seleccionar el algoritmo de transformación invariante que desee de la lista de algoritmos que la aplicación ofrece.
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en una de las distintas opciones del contenedor de algoritmos invariantes. 2. Si no hay ningún error, se guarda la preferencia del usuario.
Postcondición	Se ha seleccionado correctamente el algoritmo invariante deseado.
Excepciones	Error (mensaje).
Importancia	Alta

Tabla B.9: CU-9 Seleccionar algoritmo invariante.

CU-10	Seleccionar algoritmo de agrupamiento
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.3
Descripción	El usuario debe poder seleccionar el algoritmo de agrupamiento invariante que desee de la lista de algoritmos que la aplicación ofrece.
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en una de las distintas opciones del contenedor de algoritmos de agrupamiento. 2. Si no hay ningún error, se guarda la preferencia del usuario.
Postcondición	Se ha seleccionado correctamente el algoritmo de agrupamiento deseado.
Excepciones	Error (mensaje).
Importancia	Alta

Tabla B.10: CU-10 Seleccionar algoritmo de agrupamiento.

CU-11	Seleccionar número de centros
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.4
Descripción	El usuario debe poder seleccionar el numero de centros que desee como parámetro del algoritmo de agrupamiento seleccionado dentro de unos limites preestablecidos.
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el selector del numero de centros. 2. El usuario escribe el numero de centros deseado. 3. Si no hay ningún error, se guarda la preferencia del usuario.
Postcondición	Se ha seleccionado correctamente la cantidad de centros deseada.
Excepciones	Value must be numeric (mensaje). Value must be between 2 and Infinity (mensaje).
Importancia	Alta

Tabla B.11: CU-11 Seleccionar número de centros.

CU-12	Almacenar resultados en caché
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.5
Descripción	La aplicación debe guardar en caché los resultados de la ejecución.
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen. El usuario ha pulsado el botón de ejecutar.
Acciones	<ol style="list-style-type: none"> 1. El programa ejecuta el algoritmo de transformación invariante sobre la imagen seleccionada. 2. El programa aplica el algoritmo sobre la imagen original y sobre la imagen invariante. 3. El programa informa al usuario de que se va a proceder a guardar la ejecución en la memoria caché. 4. El programa guarda las tres imágenes en el directorio caché añadiendo el acierto en caso haber propiciando una imagen ground truth. 5. El programa actualiza la tabla de la memoria caché con la información de las tres imágenes.
Postcondición	Se han guardado correctamente las tres imágenes resultantes de la ejecución en el directorio caché. Se ha actualizado la tabla de la memoria caché.
Excepciones	Error (mensaje).
Importancia	Media

Tabla B.12: CU-12 Almacenar resultados en caché.

CU-13	Recuperar resultados almacenados en caché
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.6
Descripción	La aplicación recuperar los resultados de antiguas ejecuciones con los mismos parámetros para ahorrar costes computacionales.
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen. El usuario ha pulsado el botón de ejecutar.
Acciones	<ol style="list-style-type: none"> 1. La aplicación comprueba el la cache si hay alguna ejecución de la misma imagen con los mismos parámetros. 2. La aplicación recupera las imágenes que coinciden, estas puede ser: <ul style="list-style-type: none"> ■ La imagen invariante si coincide únicamente el método invariante sobre esa imagen. ■ La imagen original segmentada si coinciden únicamente el método de agrupamiento y la cantidad de centros sobre esa imagen. ■ La imagen invariante, la imagen original segmentada y la imagen invariante segmentada si coinciden todos los parámetros.
Postcondición	Se ha recuperado una o más imágenes si ha habido una ejecución sobre la misma imagen anterior sin la necesidad de ejecutar de nuevo.
Excepciones	Error (mensaje).
Importancia	Media

Tabla B.13: CU-13 Recuperar resultados almacenados en caché.

CU-14	Guardar imágenes resultantes
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-1.7
Descripción	El usuario debe poder guardar los resultados de la ejecución en el directorio que desee.
Precondición	La aplicación se encuentra disponible. El usuario ha cargado una imagen. El usuario ejecutado el algoritmo sobre la imagen. La aplicación ha mostrado los resultados de la ejecución.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el botón de guardar resultados. 2. El usuario selecciona el directorio donde quiere guardar las imágenes. 3. La aplicación guarda las imágenes con nombres descriptivos que indiquen con que parámetros se ha ejecutado.
Postcondición	Se han guardado correctamente las tres imágenes resultantes de la ejecución en el directorio seleccionado.
Excepciones	Error al guardar imágenes: (mensaje). Error saving images: (mensaje).
Importancia	Media

Tabla B.14: CU-14 Guardar imágenes resultantes.

CU-15	Gestión de caché
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-2, RF-2.1, RF-2.1.1, RF-2.1.2, RF-2.2, RF-2.3
Descripción	La aplicación debe ser capaz de gestionar correctamente una memoria caché donde almacenar y mostrar los resultados de las ejecuciones del usuario.
Precondición	La aplicación se encuentra disponible.
Acciones	<ol style="list-style-type: none"> 1. La aplicación al iniciarse crea en caso de no existir un directorio que se utilizara como memoria caché.
Postcondición	Se han creado/cargado correctamente el directorio que se usara como memoria caché.
Excepciones	Error al crear el directorio de la memoria caché (mensaje).
Importancia	Media

Tabla B.15: CU-15 Gestión de caché.

CU-16	Visualizar información de antiguas ejecuciones en formato tabla
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-2.1, RF-2.1.1, RF-2.1.2
Descripción	La aplicación debe mostrar en una tabla una lista de los resultados de las distintas ejecuciones almacenadas en caché.
Precondición	La aplicación se encuentra disponible. La aplicación cuenta con el directorio de la caché.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el menú de histórico de exploraciones. 2. La aplicación lee una por una cada imagen almacenada en el directorio de la caché. 3. La aplicación identifica los campos de las imágenes y en caso de tener la estructura adecuada añade una nueva fila a la tabla con la información de la imagen. <ul style="list-style-type: none"> ■ Esto se consigue mediante expresiones regulares. 4. La aplicación tras cada ejecución guarda la información de cada imagen resultante en una nueva fila en la tabla de la caché.
Postcondición	La tabla de la memoria caché se muestra correctamente y esta actualizada.
Excepciones	Error al mostrar caché (mensaje).
Importancia	Media

Tabla B.16: CU-16 Visualizar información de antiguas ejecuciones en formato tabla.

CU-17	Poder reordenar las columnas
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-2.1.1
Descripción	El usuario debe poder reordenar la tabla de la caché en función de los datos contenidos en las diferentes columnas.
Precondición	La aplicación se encuentra disponible. La aplicación cuenta con al menos una imagen en la caché.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el icono de ordenar por la fila seleccionada que se encuentra a la derecha. 2. La aplicación depende del numero de clicks de usuario reordena la tabla en función de esa columna en orden: <ul style="list-style-type: none"> ■ Ascendente. ■ Descendente. ■ Elimina la reordenación. 3. Si no hay ningún problema, la tabla se reordena según la selección del usuario.
Postcondición	La tabla de la memoria caché se muestra reordenada correctamente.
Excepciones	Error al ordenar por columna (mensaje). No hay imágenes guardadas en la caché. (mensaje).
Importancia	Baja

Tabla B.17: CU-17 Poder reordenar las columnas.

CU-18	Visualizar la imagen correspondiente a la fila seleccionada
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-2.1.2
Descripción	El usuario debe poder acceder en específico a la imagen a la que hace alusión cada fila de la tabla caché.
Precondición	La aplicación se encuentra disponible. La aplicación cuenta con al menos una imagen en la caché.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en cualquiera de los iconos de la última columna de la imagen. 2. La aplicación ha de abrir en una ventana a parte la imagen correspondiente a la fila mediante expresiones regulares. 3. La aplicación ha de facilitar una serie de opciones como guardar la imagen o imprimirla.
Postcondición	La imagen correspondiente a la fila seleccionada se muestra correctamente.
Excepciones	El archivo de imagen: (imagen)(extensión) (mensaje). The image file: (imagen)(extensión) (mensaje).
Importancia	Media

Tabla B.18: CU-18 Visualizar la imagen correspondiente a la fila seleccionada.

CU-19	Abrir el directorio de la caché
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-2.2
Descripción	El usuario debe poder acceder al directorio de la caché.
Precondición	La aplicación se encuentra disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el botón de ver imágenes. 2. La aplicación ha de abrir el directorio correspondiente a la memoria caché.
Postcondición	El usuario ha de visualizar en pantalla el directorio de la memoria caché.
Excepciones	Error al intentar abrir la carpeta de caché: (mensaje). Error while trying to open the cache folder: (mensaje).
Importancia	Baja

Tabla B.19: CU-19 Abrir el directorio de la caché.

CU-20	Borrar contenido de la caché
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-2.3
Descripción	El usuario debe poder borrar el contenido de la caché.
Precondición	La aplicación se encuentra disponible. La aplicación cuenta con al menos una imagen en la caché.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el botón de borrar datos. 2. La aplicación mediante un menú de confirmación advierte de que de continuar, se borrarán todos los datos de la memoria caché. 3. El usuario hace click en el botón de confirmar. 4. La aplicación borra el contenido del directorio de la memoria caché. 5. La aplicación actualiza el estado de la tabla que muestra las imágenes guardadas en la memoria cache, esta pasara a estar vacía. 6. La aplicacion informa al usuario de que se han borrado todos los datos de la memoria cache.
Postcondición	El contenido de la memoria caché se ha borrado correctamente.
Excepciones	Error al intentar borrar la caché: (mensaje). Error while trying to clear the cache: (mensaje).
Importancia	Baja

Tabla B.20: CU-20 Borrar contenido de la caché.

CU-21	Configuración
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-3, RF-3.1
Descripción	El usuario debe poder configurar los parámetros disponibles de la aplicación.
Precondición	La aplicación se encuentra disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario ha de hacer click en la opción u opciones que la aplicación ofrezca. 2. La aplicación ha de guardar la selección del usuario y realizar los ajustes pertinentes. 3. Si no hay errores, el usuario verá los cambios reflejados.
Postcondición	El usuario ha de poder aplicar la configuración deseada.
Excepciones	Error al cambiar el idioma: (mensaje). Error while changing the language: (mensaje).
Importancia	Media

Tabla B.21: CU-21 Configuración.

CU-22	Seleccionar el idioma
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-3.1
Descripción	El usuario debe poder seleccionar el idioma de la aplicación dependiendo si desea trabajar con esta en español o en inglés.
Precondición	La aplicación se encuentra disponible.
Acciones	<ol style="list-style-type: none"> 1. Por defecto la aplicación estará en español y con el selector a la izquierda (ES). 2. El usuario debe hacer click en el selector de idioma, este cambiara de posición y con ello el idioma de la aplicación. <ul style="list-style-type: none"> ■ Al hacer click este pasara a estar a la derecha (EN). ■ Al hacer click por segunda vez este pasara a estar a la izquierda (ES). 3. Si no hay errores, el usuario verá la aplicación en el idioma seleccionado.
Postcondición	El usuario ha de ver la aplicación al completo en el idioma seleccionado.
Excepciones	Error al cambiar el idioma: (mensaje). Error while changing the language: (mensaje).
Importancia	Media

Tabla B.22: CU-22 Seleccionar el idioma.

CU-23	Ayuda
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-4, RF-4.1, RF-4.2
Descripción	El usuario debe poder obtener ayuda sobre los distintos elementos de la aplicación.
Precondición	La aplicación se encuentra disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace click en el menú de ayuda. 2. El usuario selecciona dentro del menú de ayuda el área de interés. 3. La aplicación le muestra una explicación detallada del funcionamiento, algoritmo o proceso seleccionado. 4. El usuario hace click en el menú de exploración de algoritmos o en el de histórico de exploraciones. 5. El usuario mantiene el puntero dos segundo sobre cualquier botón de la aplicación sobre el cual quiera obtener mas información. 6. La aplicación muestra una pequeña ventana no información relevante.
Postcondición	El usuario ha de poder solucionar cualquier duda que le surja sobre la funcionalidad o el funcionamiento de la aplicación mediante la interfaz de esta.
Excepciones	Error al cargar liveness (mensaje). Error al cargar tooltips (mensaje).
Importancia	Alta

Tabla B.23: CU-23 Ayuda.

CU-24	Documentación dentro de la aplicación
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-4.1
Descripción	La aplicación debe contener distintos apartados dedicados a documentar el funcionamiento y métodos aplicados en la aplicación.
Precondición	La aplicación se encuentra disponible. El usuario hace click en el menú de ayuda.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona dentro del menú de ayuda el área de interés. 2. La aplicación le muestra una explicación detallada del funcionamiento, algoritmo o proceso seleccionado.
Postcondición	El usuario ha de poder entender mediante la documentación facilitada el funcionamiento de cada apartado y algoritmo dentro de la aplicación.
Excepciones	Error al cargar liverscripts (mensaje).
Importancia	Alta

Tabla B.24: CU-24 Documentación dentro de la aplicación.

CU-25	Tooltips
Versión	1.0
Autor	Jonás Martínez Sanllorente
Requisitos asociados	RF-4.2
Descripción	La aplicación debe proporcionar información sobre la funcionalidad de las distintas acciones dentro de esta.
Precondición	La aplicación se encuentra disponible. El usuario hace click en el menú de exploración de algoritmos o en el de histórico de exploraciones.
Acciones	<ol style="list-style-type: none"> 1. El usuario mantiene el puntero dos segundos sobre cualquier botón de la aplicación sobre el cual quiera obtener más información. 2. La aplicación muestra una pequeña ventana con información relevante.
Postcondición	La aplicación ha de facilitar información sobre la utilidad de los distintos botones y sobre las posibles restricciones de algunos campos.
Excepciones	Error al cargar tooltips (mensaje).
Importancia	Media

Tabla B.25: CU-25 Tooltips.

Apéndice C

Especificación de diseño

C.1. Introducción

En este apéndice se abordarán los aspectos relacionados con el diseño de tratamiento de datos, el diseño de procesos y el diseño arquitectónico, seguidos a lo largo del desarrollo de este proyecto

C.2. Diseño de datos

En esta sección, se describen las estructuras de datos utilizadas en la aplicación. Los datos de entrada son imágenes que se procesan mediante una serie de scripts. Las librerías de MATLAB gestionan operaciones específicas como la manipulación de imágenes y el análisis de datos. Las variables globales se utilizan para almacenar información clave, como parámetros de configuración y resultados intermedios.

Convección de nombres de imágenes

A continuación se detallan las distintas convenciones de nombres de las imágenes tanto de las proporcionadas por la aplicación como de las imágenes resultado.

Imágenes proporcionadas por la aplicación

Las imágenes proporcionadas por el sistema están nombrados siguiendo una convención específica que facilita la comprensión rápida del contenido y las características de cada imagen. Cada imagen contiene información

relevante como el número de piezas metálicas, si las piezas son iguales o distintas, el número de focos de luz utilizados, el material de las piezas (como aluminio, acero o hierro) y la técnica de formado aplicada (como troquelado o lisa). Estos elementos se codifican en el nombre del archivo utilizando abreviaturas, lo que permite identificar rápidamente las características de cada imagen sin necesidad de abrir el archivo.

La convención de nombres de archivos sigue un formato preestablecido: ‘<Número de Piezas>_<Forma de las Piezas>_<Número de Focos>_<Material>_<Técnica>.jpg’. Por ejemplo, el archivo ‘3_DIS_1_ALU_TRO.jpg’ indica que contiene una imagen con tres piezas metálicas distintas, iluminadas por un solo foco de luz, hechas de aluminio y formadas mediante troquelado. De manera similar, el archivo 2_IGU_2_ACE_LIS.jpg sugiere que la imagen contiene dos piezas metálicas iguales, iluminadas por dos focos de luz, hechas de acero y formadas con una técnica lisa. Esta metodología de nomenclatura asegura una identificación rápida y eficiente del contenido y las características principales de cada imagen, facilitando el análisis y la organización de los archivos.

Imágenes resultado

Las imágenes de solución tienen los nombres formateados de la siguiente manera:

- fecha_1__nombre__invariante.extensión
- fecha_2__nombre__Original_agrupamiento_numeroDeCentros_c_acierto%.extensión
- fecha_3__nombre__invariante_agrupamiento_numeroDeCentros_c_acierto%.extensión

Estos nombres reflejan la fecha de ejecución, el nombre de la imagen, el algoritmo utilizado, el número de centros y el porcentaje de acierto, lo que permite una organización clara y detallada de las imágenes.

Estos son los mismos nombres que acepta la memoria caché, por lo que se pueden copiar y pegar imágenes en la memoria caché para agilizar la ejecución. Al mantener la coherencia en el formato de los nombres de los archivos, se facilita su uso en procesos automatizados y se optimiza el rendimiento del sistema al reducir el tiempo de acceso a las imágenes necesarias durante la ejecución de las tareas.

C.3. Diseño procedimental

En este apartado se recogen los detalles más relevantes respecto a la ejecución de la aplicación desde que el usuario selecciona una imagen a analizar hasta que guarda los resultados.

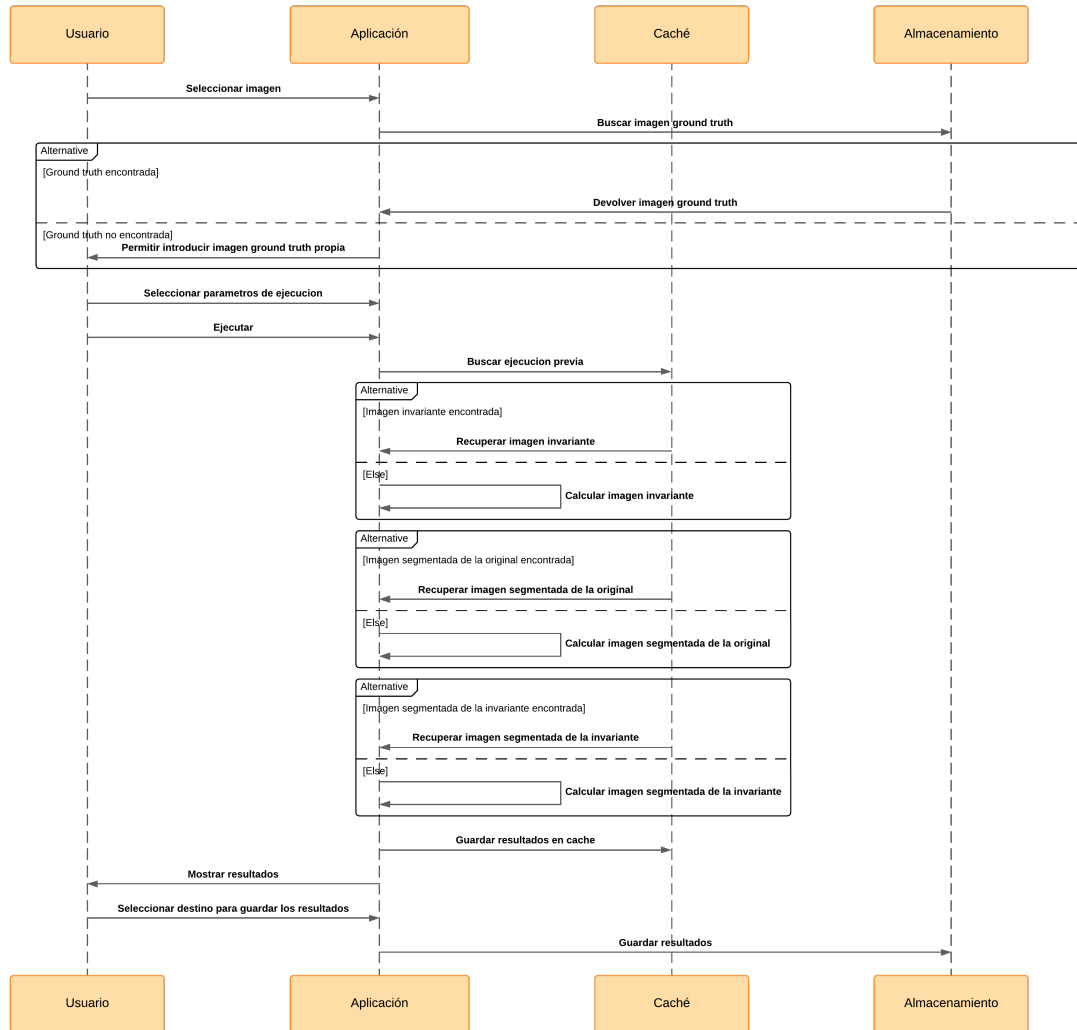


Figura C.1: Diagrama de secuencia correspondiente al caso de uso 01 gestión de la transformación de invariante de iluminación.

El proceso se inicia cuando el usuario selecciona una imagen que contenga piezas metálicas. La imagen se carga y se preprocesa eliminando el canal

alpha. A continuación se le aplica sobre la imagen el algoritmo de transformación invariante seleccionado. Después se aplica sobre la imagen original y sobre la imagen invariante el método de agrupamiento. Posteriormente, en el caso de que tenga imagen ground truth, se calcula el porcentaje de acierto de las segmentaciones. Finalmente, los resultados se presentan al usuario, quien tiene la opción de guardarlos.

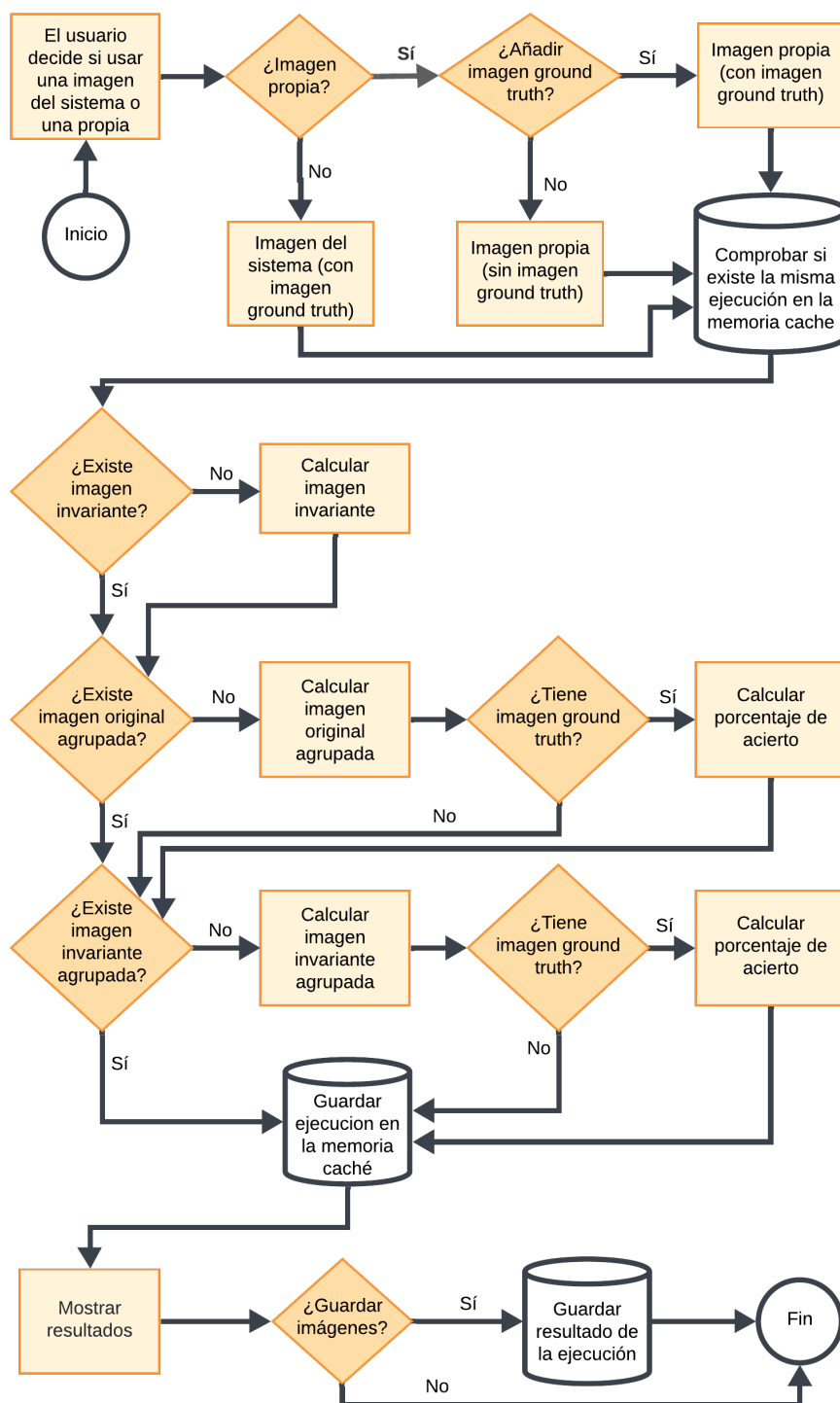


Figura C.2: Diagrama de flujo detallado del procesamiento de datos en la aplicación.

En la figura C.2 se muestra de un diagrama de flujo con los pasos a seguir desde que el usuario selecciona una imagen hasta que este la guarda.

C.4. Diseño arquitectónico

La arquitectura de la aplicación está basada en una serie de módulos interconectados que utilizan funciones específicas para implementar los casos de uso. A continuación se presentan los componentes principales:

- **Interfaz de usuario:** permite la carga de imágenes, la selección de transformación invariante, la selección de número de centros, la selección de algoritmo de agrupamiento y la visualización de resultados ente otras cosas.
- **Módulo de procesamiento:** son todas aquellas funciones que la aplicación ejecuta de forma opaca al usuario que hacen uso de los scripts para obtener los resultados.
- **Bibliotecas de MATLAB:** conjunto de funciones predefinidas utilizadas para operaciones específicas de manipulación de datos y análisis de imágenes.
- **Sistema de almacenamiento:** hay de tres tipos:
 - ‘/data’ que guarda las imágenes de prueba y su correspondiente imagen ground truth.
 - ‘/cache’ que se encarga de almacenar los resultados de todos las anteriores ejecuciones exceptuando el caso de que el usuario lo desee eliminar.
 - ‘/results’ que es la ruta donde se almacenan por defecto las imágenes que el usuario desea guardar.

En la figura C.3 se muestra la interacción que tienen los distintos componentes de la aplicación.

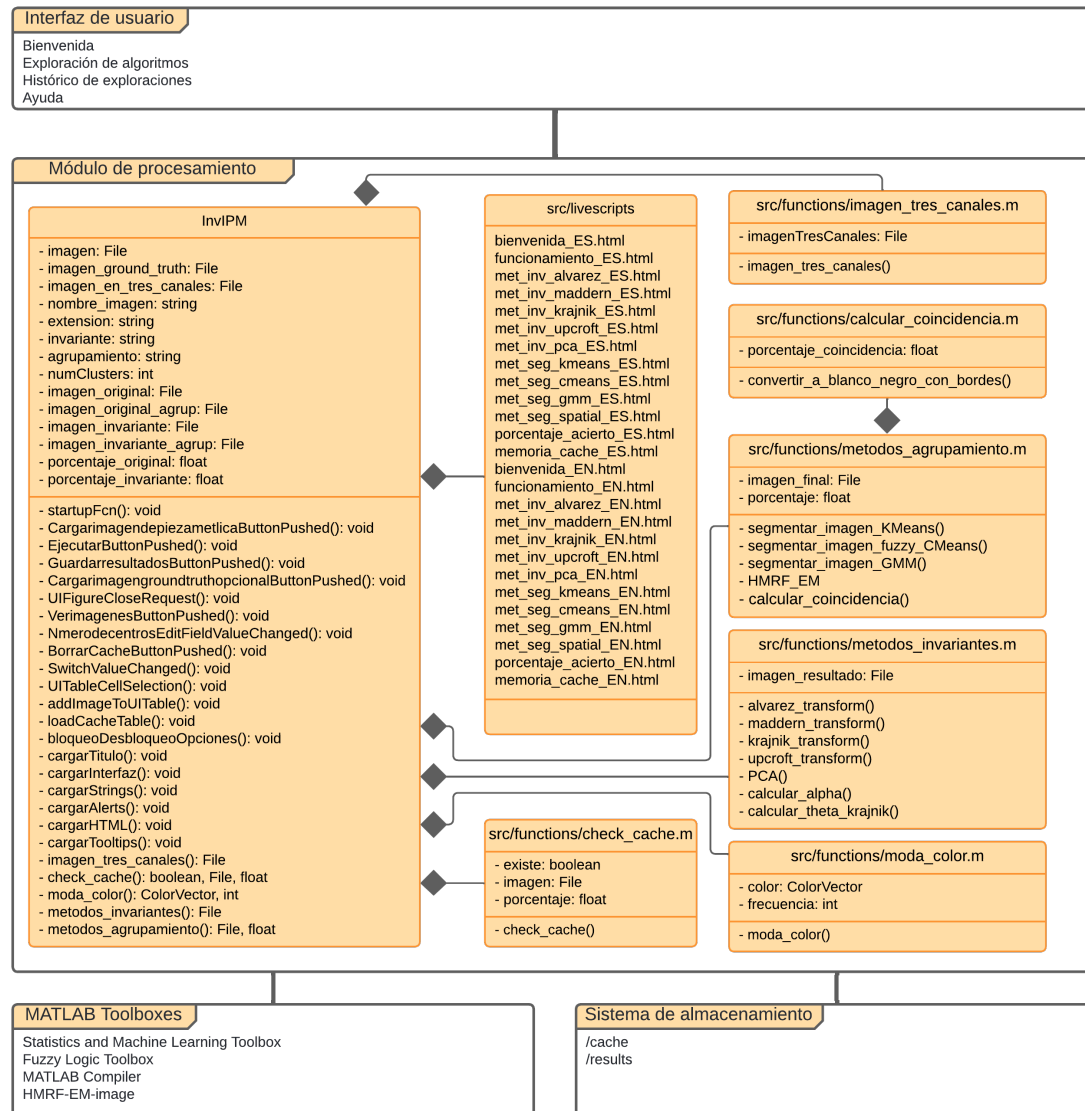


Figura C.3: Diagrama de componentes mostrando la interacción entre los distintos módulos de la aplicación.

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este anexo se detalla la documentación técnica de programación, incluyendo la instalación del entorno de desarrollo, la estructura de la aplicación, su compilación, la configuración de los diversos servicios de integración utilizados y las baterías de pruebas realizadas.

D.2. Estructura de directorios

El repositorio del proyecto tiene la siguiente estructura:

- **/:** Directorio raíz del repositorio. Es este se encuentra todos los directorios del proyecto además de una README.
- **/app:** Directorio de la aplicación. Este contiene tanto la propia aplicación como aquellos directorios necesarios para la correcta ejecución de la misma.
- **/app/src:** Directorio del código fuente de la aplicación.
- **/app/src/functions:** Directorio de los scripts utilizados por el sistema.

- **/app/src/livescripts:** Directorio de los livescripts y sus correspondientes HTML de la documentación de dentro de la aplicación tanto en español como en inglés.
- **/app/src/livescripts/img:** Directorio de las imágenes utilizadas en la documentación de los livescripts.
- **/app/src/test:** Directorio de los tests unitarios de los scripts de ‘/functions’. Cuentan con un README para poderlos ejecutar correctamente.
- **/app/data:** Directorio de las imágenes proporcionadas y sus respectivas imágenes ground truth, además contiene un README para explicar la nomenclatura utilizada en las imágenes.
- **/app/data/Imágenes:** Directorio de las imágenes proporcionadas por la aplicación.
- **/app/data/Imágenes_ground_truth:** Directorio de las imágenes ground truth correspondientes a las imágenes proporcionadas por la aplicación.
- **/app/cache:** Directorio de la memoria caché de la aplicación. En el se guardarán las imágenes generadas durante la ejecución.
- **/app/results:** Directorio donde se almacenarán las imágenes por defecto si el usuario decide guardar los resultados de la ejecución.
- **/docs:** Directorio de la documentación del proyecto.
- **/docs/img:** Directorio de las imágenes utilizadas en la documentación.
- **/docs/tex:** Directorio de las fuentes en formato \LaTeX para memoria y anexos.

D.3. Manual del programador

El siguiente manual tiene como objetivo explicar cómo un programador puede descargar el proyecto y desarrollar su propia versión.

Entorno de desarrollo

Para poder trabajar con el proyecto es necesario tener instalado esta serie de programas y dependencias.

- MATLAB 2023b.
- MATLAB Add-On Toolboxes.
- GitHub Desktop.

A continuación, se detallan las instrucciones para instalar y configurar correctamente cada uno de ellos.

MATLAB 2023b

El principal e indispensable programa para la ejecución de este proyecto es MATLAB, más concretamente la versión 2023b. Para descargarlo tenemos que ir a la página oficial de MATLAB [13]. Una vez iniciada la aplicación, se pedirá introducir una cuenta de MATLAB. La Universidad de Burgos ofrece a los alumnos acceso a la versión académica de MATLAB con su correo institucional, versión con la cual se ha realizado este proyecto.

MATLAB Add-On Toolboxes

Una vez instalado MATLAB, será necesario instalar una serie de Add-On Toolboxes para la correcta ejecución del proyecto. Estas son las siguientes:

- Statistics and Machine Learning Toolbox.
- Fuzzy Logic Toolbox.
- HMRF-EM-image.
- MATLAB Compiler.

Esto se puede hacer desde el menú superior APPS' y luego pulsando en Get More Apps'. A continuación, se mostrará una nueva ventana donde, mediante un buscador, se podrá acceder a las Add-On Toolboxes necesarias. A continuación, se muestra un ejemplo de la instalación de 'Statistics and Machine Learning Toolbox' en la figura D.1.

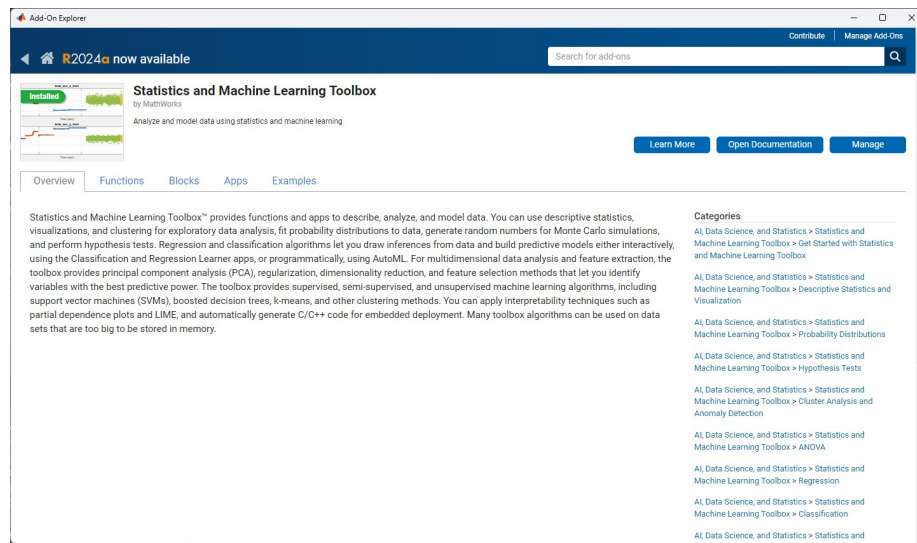


Figura D.1: Instalación Statistics and Machine Learning Toolbox

En la figura [D.1](#) se muestra la ventana que se obtiene tras instalar Statistics and Machine Learning Toolbox.

GitHub Desktop

La instalación de GitHub Desktop no es estrictamente necesaria pero sí recomendable ya que permite tener un control de versiones lo cual es de gran interés a la hora de hacer cambios en la aplicación.

Para instalar GitHub Desktop, se debe ir a la página oficial [\[5\]](#) y desde allí descargar los archivos necesarios. Tras la instalación, se pedirá ingresar el usuario de GitHub

Descarga del repositorio

Para la descarga del repositorio hay dos principales opciones:

Obtención manual del código fuente

En el caso de que no estimemos necesario un control de versiones, se puede descargar directamente el archivo .zip del proyecto desde el repositorio de GitHub (<https://github.com/jms1008/TFG-Invariantes-de-iluminacion-en-piezas-metalicas>). Esto se realiza pulsando en el botón 'Code' y más tarde en el de 'Download ZIP' como se puede ver en la figura [D.2](#):

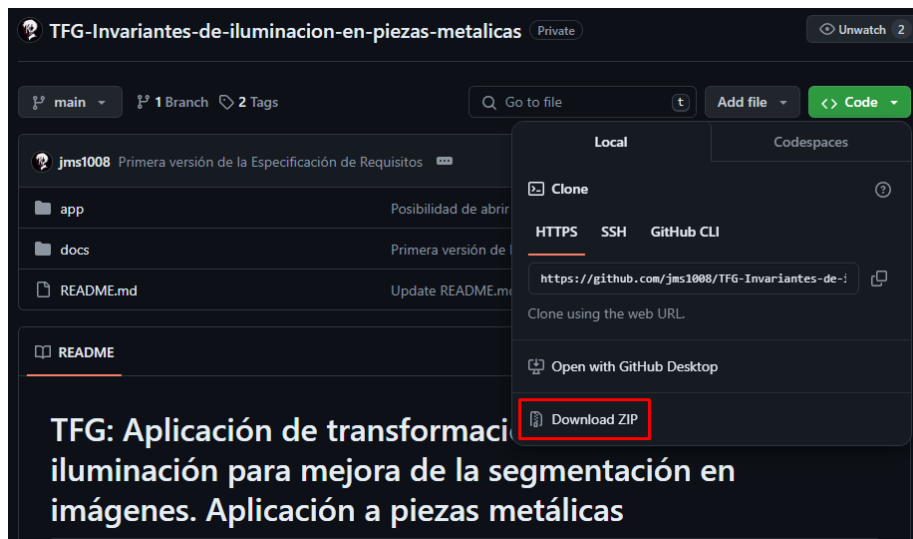


Figura D.2: Descarga del repositorio mediante el entorno web.

En la figura D.2 se muestra donde está ubicado el botón para poderse descargar el repositorio desde GitHub.

Tras esto solo quedaría descomprimir el proyecto en el directorio deseado.

Obtención del código fuente mediante GitHub Desktop

En el caso de querer tener un control de versiones, se puede utilizar la aplicación GitHub Desktop mencionada anteriormente en D.3.

Para ello, se debe entrar a la aplicación, pulsar en 'Clonar repositorio' e introducir el enlace del proyecto que se desea clonar. En este caso, el enlace del repositorio de este proyecto es el siguiente: <https://github.com/jms1008/TFG-Invariantes-de-iluminacion-en-piezas-metalicas>. A continuación, se debe indicar dónde se desea almacenar el repositorio.

Importación del proyecto

Para la correcta ejecución del proyecto, se debe abrir MATLAB 2023b y navegar dentro de MATLAB al directorio donde se encuentre el repositorio. Una vez allí, se debe escribir en la consola 'appdesigner' para que se abra la ventana correspondiente, donde se podrá visualizar, modificar y ejecutar, entre muchas otras funcionalidades, la aplicación.

MATLAB App Designer

La en la primera pantalla que aparece nada más abrir MATLAB App Designer nos permitirá abrir un proyecto. Aquí hemos de seleccionar el archivo ‘InvIPM.mlapp’ que se encuentra en el directorio ‘/app’.

Una vez abierto el proyecto veremos una vista de diseño de la aplicación donde se puede modificar la posición y algunos valores dentro del apartado visual como se muestra en la figura D.3 y otro botón que permite acceder a la vista de código.

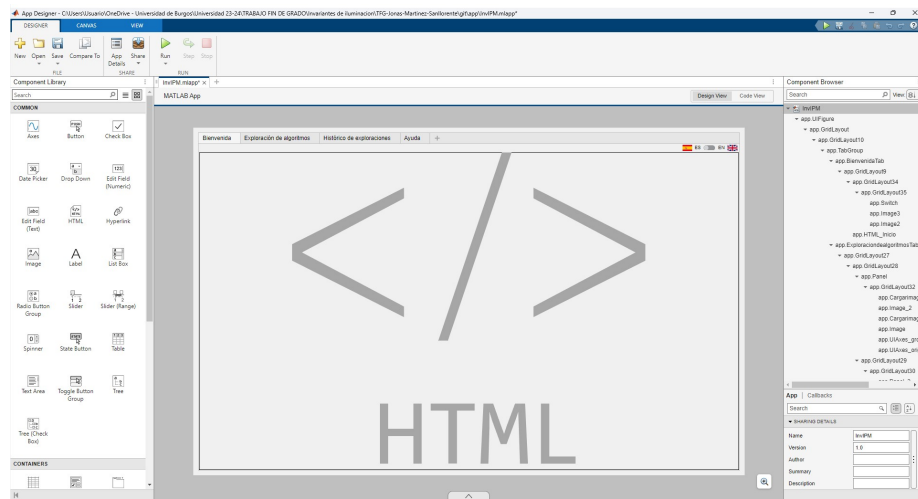


Figura D.3: Ventana principal de MATLAB App Designer.

La figura D.3 muestra la interfaz que muestra MATLAB tras abrir el proyecto descargado previamente.

Es en la vista de código donde se podrá modificar más en profundidad los objetos visuales y programar las funcionalidades de las distintas interacciones de la aplicación.

Aunque la forma de programar en MATLAB App Designer es muy similar a la programación en MATLAB, hay una serie de cambios y funcionalidades que son muy relevantes. En el siguiente enlace se encuentra una guía de las funcionalidades y la forma de programar en MATLAB App Designer, <https://es.mathworks.com/help/matlab/app-designer.html>

D.4. Compilación, instalación y ejecución del proyecto

A continuación, se explicará cómo:

- Compilar y ejecutar el proyecto dentro de MATLAB.
- Crear una nueva release mediante una MATLAB Standalone Application.
- Directorios creados por el compilador de MATLAB.
- Ejecución de la Standalone App.

Compilación en App Designer

MATLAB permite ejecutar directamente desde la interfaz compilar y ejecutar la aplicación. Para ello simplemente tendremos que pulsar el icono de ‘run’ que se encuentra en la barra superior, dicho icono es el que se muestra en la figura D.4.



Figura D.4: Icono de compilación y ejecución de MATLAB App Designer.

La figura D.4 muestra la ubicación en la interfaz del botón que se utiliza para compilar y ejecutar la app.

Tras esto, se abrirá la aplicación en una ventana emergente donde se podrán probar de forma muy sencilla las distintas funcionalidades que sean añadidas. Dicha ventana tendrá un aspecto similar al mostrado en la figura D.5



Figura D.5: Ventana de la aplicación compilada y ejecutada.

La figura [D.5](#) muestra ventana que se crea automáticamente tras la compilación y ejecución de la aplicación.

Crear una nueva release

Tras realizar las pertinentes modificaciones o añadir nuevas funcionalidades a la aplicación, el siguiente paso será compartir la aplicación. MATLAB para ello tiene tres opciones:

- **MATLAB App:** Esta opción permite crear un archivo de instalación para compartir la aplicación con otros usuarios de MATLAB. Si el objetivo es que estos puedan modificar o ver el funcionamiento de la aplicación, esta es la mejor opción.
- **Web App:** Esta opción permite crear y desplegar aplicaciones web utilizando MATLAB Compiler, facilitando su distribución y uso en diferentes entornos. Esto ha sido desestimado ya que esta funcionalidad la ofrece pero es de pago.
- **Standalone Desktop App:** Esta opción permite crear aplicaciones de escritorio independientes usando MATLAB Compiler, que requieren

tener MATLAB instalado para su ejecución. Esta es la opción que se ha utilizado y que se desarrollará con más detalle a continuación.

MATLAB Standalone Application

Para crear una Standalone Application dentro de MATLAB App Designer, se debe pulsar el icono de compartir y, a continuación, seleccionar la opción de ‘Standalone Desktop App’. Tras esto, aparecerá una ventana para añadir la información relevante de la aplicación que se va a crear. Aquí se deben introducir los siguientes datos, como se muestra en la figura D.6:

- Nombre de la aplicación.
- Versión de la aplicación.
- Autor (nombre, correo, compañía).
- Descripción (opcional).
- Archivos y directorios necesarios para la correcta ejecución del programa.
- Archivos visibles para el usuario.

La figura D.6 muestra la interfaz que MATLAB ofrece para la creación de una Standalone App.

Ejecución de la Standalone App

Al abrir el archivo ‘InvIPM.exe’, se accederá a la aplicación, la cual ejecutará todos los scripts y el código de manera transparente para el usuario, mostrando únicamente los directorios deseados. En este caso, estos serían:

- El directorio ‘/data’ ya que contienen las imágenes de prueba con sus respectivas imágenes ground truth.
- El directorio ‘/cache’ ya que sera donde se almacenen todos los resultados de antiguas ejecuciones de forma temporal.
- El directorio ‘/results’ ya que sera la ruta por defecto que se le ofrezca al usuario cuando este desee guardar los resultados de una ejecución.
- El README con las pertinentes explicaciones.
- El splash de la aplicación.

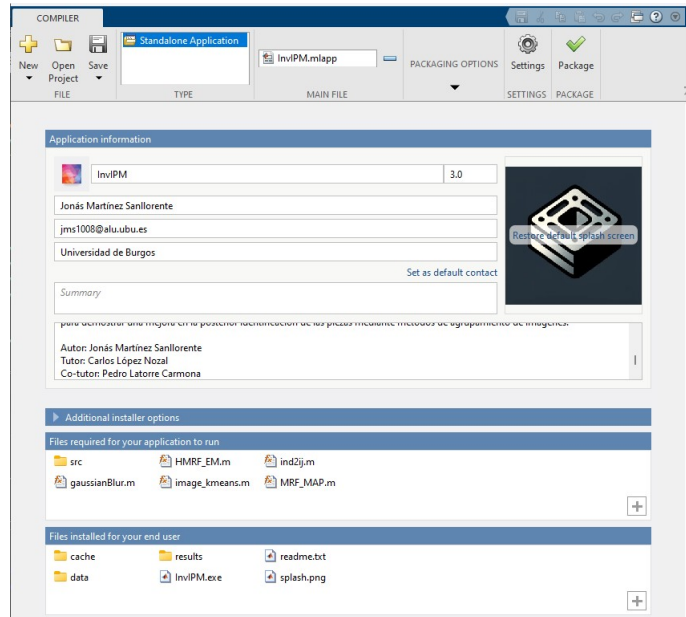


Figura D.6: Creación de MATLAB Standalone App.

D.5. Pruebas del sistema

A continuación, se describen las distintas pruebas del sistema.

Casos de prueba

Se han creado múltiples tests unitarios para comprobar el correcto funcionamiento de manera aislada de los scripts que utiliza la aplicación.

Los distintos tests creados son los siguientes:

- `alvarez_transformTest.m`
 - Test básico.
 - Test de valores extremos.
 - Test con una imagen uniforme.
 - Test con una imagen en blanco y negro.
- `maddern_transformTest.m`
 - Test básico.
 - Test de valores extremos.

- Test con una imagen uniforme.
- `krajnik_transformTest.m`
 - Test básico.
 - Test de valores extremos.
 - Test con una imagen uniforme.
- `upcroft_transformTest.m`
 - Test básico.
 - Test de valores extremos.
 - Test con una imagen uniforme.
- `calcular_alphaTest.m`
 - Test valores básicos.
 - Test con λ a 0.
 - Test con las tres λ s iguales.
- `calcular_theta_krajnikTest.m`
 - Test valores básicos.
 - Test con una imagen uniforme.
 - Test con una imagen en blanco y negro.
- `calcular_coincidenciaTest.m`
 - Test con imágenes iguales.
 - Test con imágenes diferentes.
- `check_cacheTest.m`
 - Test existe caché (la crea, añade una imagen y luego borra la caché).
- `convertir_a_blanco_negro_con_bordesTest.m`
 - Test con un caso perfecto.
 - Test con un caso no perfecto.
- `imagen_tres_canalesTest.m`

- Test con una imagen de cuatro canales.
- Test con una imagen de tres canales.
- `metodos_agrupamientoTest.m`
 - Test de K-Means sin imagen ground truth.
 - Test de K-Means con imagen ground truth.
- `metodos_invariantesTest.m`
 - Test de el invariante de Álvarez.
 - Test de el invariante de Maddern.
- `moda_colorTest.m`
 - Test básico, un color aparece repetido.
 - Test de colores únicos.
- `segmentar_imagen_fuzzy_CMeansTest.m`
 - Test básico.
- `segmentar_imagen_GMMTest.m`
 - Test básico.
- `segmentar_imagen_KMeansTest.m`
 - Test básico.

Como se puede apreciar, ni PCA ni HMRF_EM tienen test unitarios propios ya que dichas librerías ya han sido testeadas por sus desarrolladores.

Ejecución de los casos de prueba

Para ejecutar los casos de prueba, como viene indicado en el README de la carpeta test, hay de copiar los scripts del directorio ‘/test’ al directorio ‘/functions’.

Una vez realizado esto tenemos dos opciones.

1. **Ejecutar uno a uno:** si se desea ejecutar un test específico por separado, se debe abrir la consola de comandos y utilizar el siguiente comando, reemplazando ‘nombreDelTest’ por el nombre del test que deseas ejecutar ‘`disp(runtests(‘nombreDelTest’))`’.

2. **Ejecutar todos a la vez:** si se desea ejecutar todos los tests a la vez, simplemente se debe ejecutar el archivo 'run_all_tests.m'. Este archivo se encargará de correr todos los tests de forma automática.

Los resultados de ejecutar todos los tests son los que se pueden observar en la figura D.7.

TestName	Passed	Failed
{'alvarez_transformTest'}	4	0
{'calcular_alphaTest'}	3	0
{'calcular_theta_krajnikTest'}	3	0
{'calcular_coincidenciaTest'}	2	0
{'check_cacheTest'}	1	0
{'convertir_a_blanco_negro_con_bordesTest'}	2	0
{'imagen_tres canalesTest'}	2	0
{'moda_colorTest'}	2	0
{'metodos_invariantesTest'}	2	0
{'metodos_agrupamientoTest'}	2	0
{'segmentar_imagen_KMeansTest'}	1	0
{'segmentar_imagen_fuzzy_CMeansTest'}	1	0
{'segmentar_imagen_GMMTest'}	1	0

Figura D.7: Resultados de la ejecución de los tests.

La figura D.7 muestra un resumen de los resultados de la ejecución de los tests sobre los scripts que utiliza la aplicación.

Apéndice *E*

Documentación de usuario

E.1. Introducción

Este apéndice está destinado a abordar los temas relacionados con la parte del usuario. Aquí se tratarán los requisitos, la instalación de la aplicación y el manual de uso para el usuario.

E.2. Requisitos de usuarios

Los requisitos necesarios para poder utilizar la aplicación son:

- Disponer de un ordenador, ya que se trata de una aplicación de escritorio.
- MATLAB Runtime 2023b, disponible en <https://es.mathworks.com/products/compiler/matlab-runtime.html>.

MATLAB Runtime 2023b

El único programa necesario para la ejecución de este proyecto es MATLAB Runtime, específicamente la versión correspondiente a MATLAB 2023b. Para descargarlo, debes ir a la página oficial de MathWorks [2]. Una vez descargado e instalado, no se requiere una cuenta de MATLAB para su uso. MATLAB Runtime permite ejecutar aplicaciones compiladas con MATLAB Compiler sin necesidad de tener una licencia de MATLAB completa.

E.3. Instalación

No hay un proceso de instalación adicional aparte de las dependencias mencionadas anteriormente, ya que la aplicación es auto contenida. Esto significa que todos los archivos y bibliotecas necesarios ya están incluidos en la carpeta de la versión. Gracias a esto, la instalación y el uso se simplifican considerablemente, permitiendo que incluso los usuarios sin experiencia en MATLAB tengan una experiencia agradable.

El primer paso será ir al repositorio de GitHub y descargarse la ultima release, esto se puede hacer tal y como se muestra en la figura E.1 en el menú de la derecha:



Figura E.1: Instalación de la release desde GitHub.

La figura E.1 muestra una la release v0.3.1 de la aplicación InvIPM en GitHub incluyendo la versión, una breve descripción de los cambios y funcionalidades nuevas respecto a la anterior y un zip con los binarios para ejecutar en Linux y en Windows. En este apartado es muy importante descargarse la versión correspondiente al sistema operativo utilizado.

Una vez descargado el archivo zip tendremos que descomprimirlo. Tras ello, nos quedaran una serie de archivos y carpetas en el directorio deseado. Estas son:

- /InvIPM.exe: este es el ejecutable de la aplicación.
- /readme.txt: es un documento de ayuda el cual nos indica los requisitos necesarios además de como ejecutar la aplicación.
- /data: en este directorio se almacenan las imágenes de ejemplo que trae la propia aplicación, estas podrán ser borradas por el usuario si así lo desea.
- /cache: en este directorio se almacenaran las imágenes guardadas en caché.
- /results: este es el directorio que se le ofrece al usuario por defecto cuando quiere guardar los resultados de una ejecución.
- /splash.png: es la imagen que aparece nada mas abrir la aplicación.

E.4. Manual del usuario

Sección dedicada a la redacción de un manual sencillo para que el usuario comprenda cómo utilizar la aplicación.

Bienvenida

Nada más entrar en la aplicación, se presentará una imagen de bienvenida que explica en términos generales el objetivo de la misma, además de mostrar el autor y los tutores del proyecto. El contenido de esta pantalla se muestra en la figura [E.2](#)

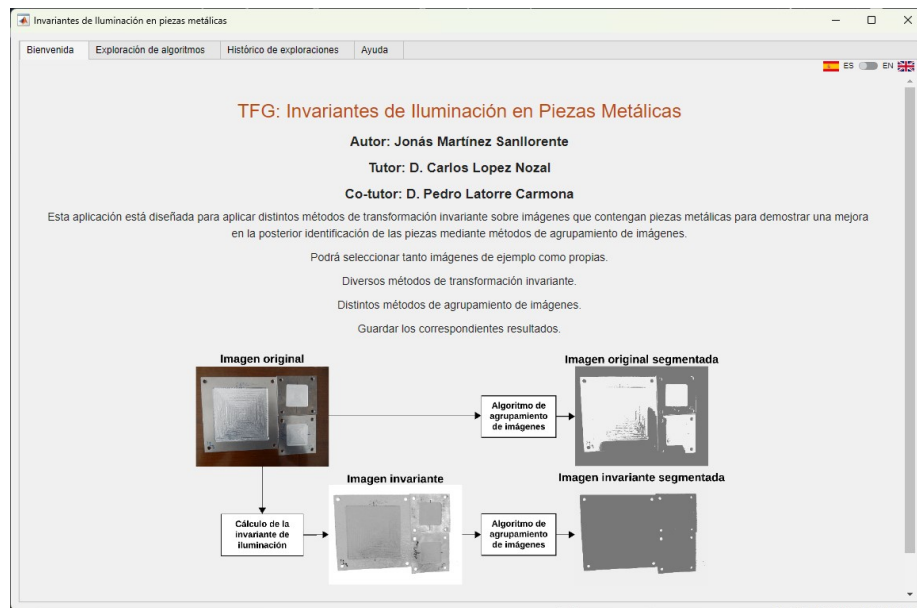


Figura E.2: Apartado de bienvenida de InvIMP.

En la figura E.2 se puede apreciar el aspecto de la ventana de bienvenida de InvIPM.

En la parte superior derecha de esta pantalla, se encuentra un selector de idioma entre español e inglés, con sus correspondientes banderas. Esto lo hace más autoexplicativo, permitiendo al usuario identificarlo sin necesidad de leer el texto. Esto es especialmente útil cuando la aplicación está en un idioma diferente al del usuario.

Exploración de algoritmos

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

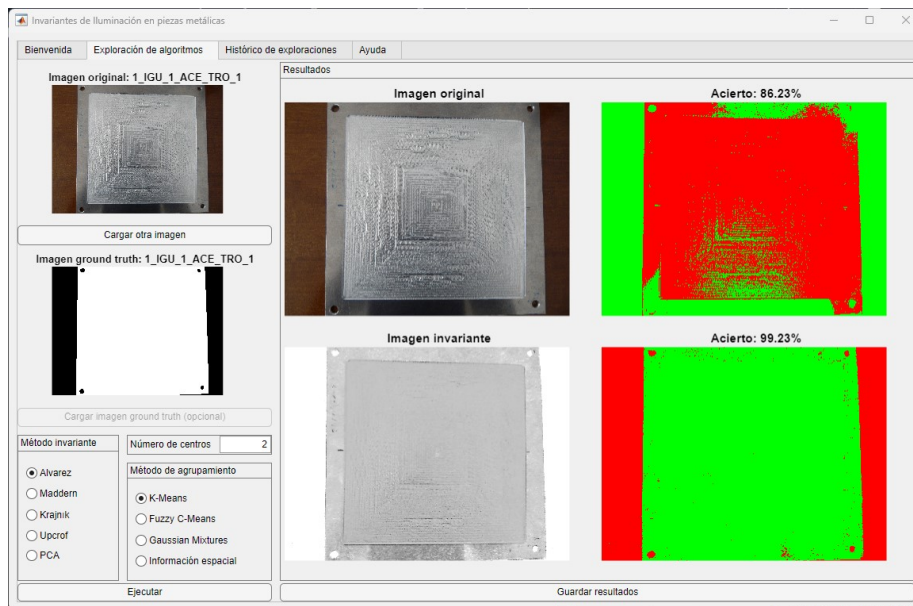


Figura E.3: Apartado de exploración de algoritmos de InvIMP.

En la figura E.3 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’, se podrá seleccionar una imagen proporcionada por el sistema o una propia sobre la cual se aplicarán los algoritmos más 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.
- Maddern.

- Krajník.
- Upcroft.
- PCA.

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 otros métodos, en concreto Álvarez.

Selección de algoritmo de agrupamiento

Se proporcionan los distintos métodos de agrupamiento:

- K-Means.
- Fuzzy C-Means.
- Gaussian Mixtures.
- Información espacial, el cual es HMRF_EM.

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 número de centros a utilizar en los algoritmos de agrupamiento. Por defecto este número será 2.

El número de centros ha de cumplir las siguientes condiciones:

- Ser un numero positivo.
- Ser igual o mayor que 2.
- 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 mostraran 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.

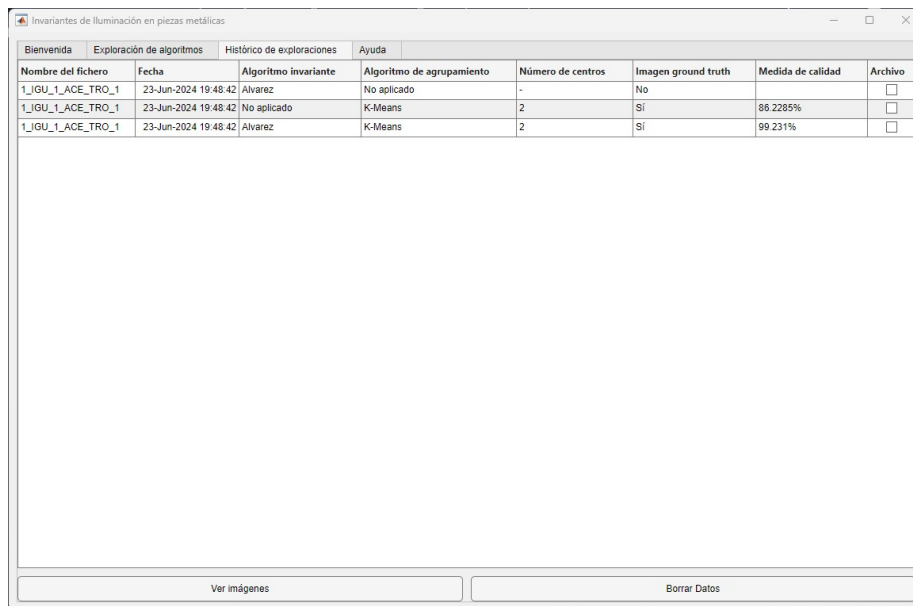
Histórico de exploraciones

En este apartado, se presenta una tabla en la cual aparecen, a modo de lista, todas las ejecuciones realizadas 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 ven-
tana 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_IGU_1_ACE_TRO_1	23-Jun-2024 19:48:42	Alvarez	No aplicado	-	No		<input type="checkbox"/>
1_IGU_1_ACE_TRO_1	23-Jun-2024 19:48:42	No aplicado	K-Means	2	Sí	86.2285%	<input type="checkbox"/>
1_IGU_1_ACE_TRO_1	23-Jun-2024 19:48:42	Alvarez	K-Means	2	Sí	99.231%	<input type="checkbox"/>

Figura E.4: Apartado de histórico de exploraciones de InvIMP.

En la figura E.4 se puede apreciar el aspecto de la ventana de 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 se podrá ver y seleccionar todas aquellas imágenes que están almacenadas en la memoria caché.

Borrar datos

Al pulsar este botón, aparecerá una ventana emergente indicando si realmente se desea borrar de forma irreversible el contenido de la memoria caché. En caso de pulsar el botón 'Borrar', se eliminará su contenido y se actualizará la tabla, ya que dichas imágenes no se encontrarán disponibles. Tras esto, se indicará que la memoria caché se ha borrado exitosamente.

Ayuda

En este apartado, tal y como se muestra en la figura E.5, 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.

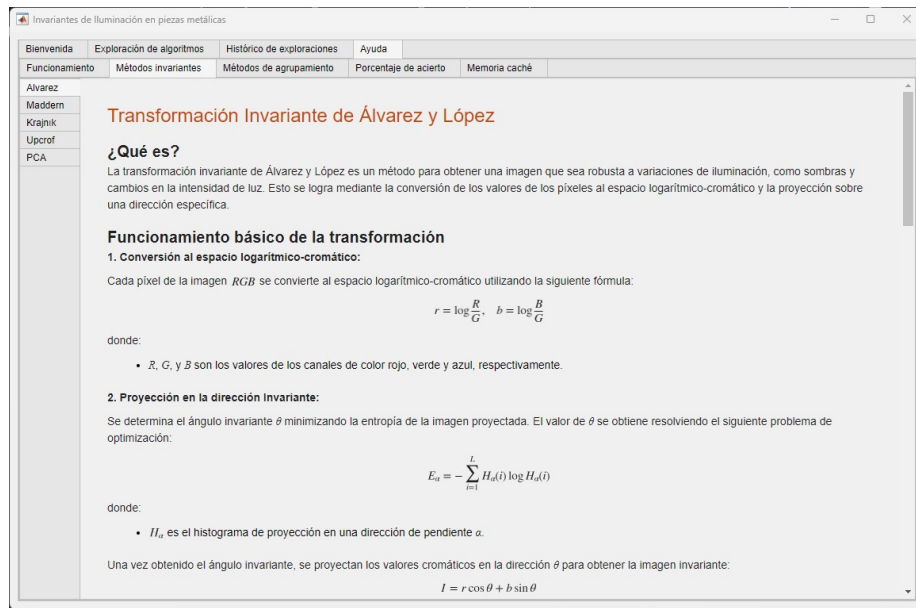


Figura E.5: Apartado de ayuda de InvIMP.

En la figura E.5 se puede apreciar el aspecto de la ventana de ayuda de InvIMP.

Anexo de sostenibilización curricular

F.1. Introducción

Este anexo tiene como objetivo reflexionar sobre los aspectos de sostenibilidad abordados en el proyecto. Durante el desarrollo de este, se han adquirido competencias en sostenibilidad que se aplican en el Trabajo de Fin de Grado. A continuación, se detallan estos aspectos y competencias adquiridas.

F.2. Desarrollo del proyecto

El proyecto consistió en investigar, probar y finalmente crear una aplicación que utiliza transformaciones invariantes para mejorar la segmentación de imágenes de piezas metálicas. La hipótesis inicial era que la incorporación de estas transformaciones aumentaría la precisión de los algoritmos de segmentación, una suposición que fue confirmada a través de experimentos rigurosos.

Primero, se realizó una extensa revisión bibliográfica sobre técnicas de transformación invariante y segmentación de imágenes. Se analizaron diversas metodologías y se seleccionaron las que mostraban mayor potencial para mejorar los resultados de segmentación. Esta fase inicial fue crucial para establecer una base sólida y asegurar que las técnicas seleccionadas fueran las más adecuadas para el objetivo del proyecto.

Posteriormente, se desarrollaron prototipos de la aplicación, implementando diferentes transformaciones invariantes y evaluando su impacto en la segmentación. Se utilizó un conjunto de imágenes variado para garantizar que los resultados fueran robustos y aplicables a la industria, teniendo en cuenta diferentes niveles de iluminación y complejidad de las imágenes. Los resultados mostraron un incremento significativo en la tasa de acierto, confirmando la validez de la hipótesis.

F.3. Competencias de sostenibilidad adquiridas

Durante el desarrollo del proyecto, se adquirieron diversas competencias de sostenibilidad, aplicadas de manera transversal a lo largo del trabajo:

Contextualización crítica del conocimiento

Se desarrolló la capacidad de contextualizar críticamente el conocimiento adquirido, relacionándolo con problemáticas sociales y ambientales. La mejora en la segmentación de piezas metálicas tiene aplicaciones directas en la industria, donde una segmentación precisa es crucial para reducir el desperdicio de materiales y mejorar la eficiencia de los procesos de producción, contribuyendo así a una menor huella ambiental.

Uso sostenible de recursos

El proyecto promovió el uso sostenible de recursos tecnológicos y materiales. Se optimizaron los algoritmos para que fueran eficientes en términos de tiempo de computación y consumo de energía, reduciendo así el impacto ambiental. Además, al mejorar la segmentación y la inspección de piezas metálicas, se minimiza el desperdicio de materiales y se incrementa la vida útil de los productos, promoviendo un ciclo de vida más sostenible para los recursos metálicos.

Aplicación de principios éticos

Se aplicaron principios éticos relacionados con la sostenibilidad en cada etapa del proyecto. Esto incluyó la elección de técnicas y métodos que minimizaran el impacto ambiental y social negativo. Además, se aseguró la transparencia y la ética en la presentación de resultados y la difusión de

conocimientos, promoviendo un uso responsable y sostenible de la tecnología desarrollada.

F.4. Reflexión personal

Este proyecto ha sido una oportunidad para integrar competencias de sostenibilidad en un contexto técnico, demostrando que la innovación tecnológica puede y debe alinearse con principios de desarrollo sostenible. La aplicación de transformaciones invariantes no solo mejora la precisión técnica, sino que también puede contribuir a avances significativos en la industria, reduciendo el desperdicio de materiales y mejorando la eficiencia energética.

La experiencia estos últimos años ha resaltado la importancia de considerar siempre el impacto ambiental y social de nuestras decisiones técnicas. En el futuro, como profesionales, es esencial que adoptemos un enfoque responsable y sostenible en nuestro trabajo, contribuyendo así a la creación de un futuro más equitativo y sostenible.

En conclusión, este proyecto no solo ha demostrado la validez de los métodos de transformación invariante aplicados a piezas metálicas, sino que también ha sido un ejercicio práctico en la aplicación de principios de sostenibilidad, demostrando que es posible desarrollar tecnología de manera ética y responsable. La integración de transformaciones invariantes en la segmentación de piezas metálicas representa un avance significativo tanto en la eficiencia de la producción industrial como en la minimización del impacto ambiental, subrayando la importancia de la sostenibilidad en la ingeniería y la tecnología.

Bibliografía

- [1] Jornada laboral — expansion.com. <https://www.expansion.com/diccionario-juridico/jornada-laboral.html#:~:text=C%20L%C3%ADmites%20de%20trabajo%20en%20jornada,torno%20de%20las%20horas>. [Accessed 04-07-2024].
- [2] Matlab runtime - matlab compiler — es.mathworks.com. <https://es.mathworks.com/products/compiler/matlab-runtime.html>. [Accessed 04-07-2024].
- [3] Ieee recommended practice for software requirements specifications. *IEEE Std 830-1998*, pages 1–40, 1998.
- [4] 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.
- [5] GitHub. Github desktop download. https://desktop.github.com/?ref_cta=download+desktop&ref_loc=installing+github+desktop&ref_page=docs. [Accessed 21-06-2024].
- [6] Glassdoor. Sueldos para el puesto de Ingeniero Informático en España. https://www.glassdoor.es/Sueldos/ingeniero-inform%C3%A1tico-sueldo-SRCH_K00,21.htm. [Accessed 16-06-2024].
- [7] 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.

- [8] Tomáš Krajník, Jan Blažíček, and João M. Santos. Visual road following using intrinsic images. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2015.
- [9] 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.
- [10] MathWorks Inc. *Fuzzy Logic Toolbox: Design and Simulate Fuzzy Logic Systems*, 2023. Version 10.3 (R2023b).
- [11] MathWorks Inc. *MATLAB: The Language of Technical Computing*, 2023. Version 10.3 (R2023b).
- [12] MathWorks Inc. *Statistics and Machine Learning Toolbox: Use fitgmdist for Fitting Gaussian Mixture Models*, 2023. Version 10.3 (R2023b).
- [13] MATLAB. Obtener productos de matlab y simulink. <https://es.mathworks.com/downloads/>. [Accessed 21-06-2024].
- [14] 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.
- [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.