UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Análisis y desarrollo de una técnica de alineamiento para imágenes de documentos

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

AUTOR/A: Jiménez Mondéjar, Raquel

Tutor/a: Sánchez Peiró, Joan Andreu

CURSO ACADÉMICO: 2021/2022

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Departamento de Sistemas Informáticos y Computación
Universitat Politécnica de Valencia

Análisis y desarrollo de una técnica de alineamiento para imágenes de documentos

TRABAJO FIN DE MÁSTER

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Autor: Raquel Jiménez Mondéjar
Tutor: Joan Andreu Sánchez Peiró

Curso 2021-2022

Agradecimientos

Gracias a Joan Andreu por volver a ser mi tutor en este proyecto, y por orientarme y ayudarme para llevarlo a cabo.

Y, gracias a mi familia y amigos, en especial a mis padres y a Antonio, por apoyarme y confiar en mí en todo momento.

Resumen

Este proyecto consiste en el análisis y desarrollo de una técnica de alineamiento para imágenes de documentos con el objetivo de poder buscar información en las imágenes. En concreto, este proyecto se basa en la implementación de una aplicación gráfica para el alineamiento de imágenes de documentos que tienen una maquetación (layout) similar. Las imágenes de documentos con una maquetación similar pueden presentar variaciones debidas al proceso de captura de la imagen (ligeras rotaciones, curvatura, traslación, etc.). Alinear distintas imágenes en este caso requiere una comparación no lineal entre cada par de imágenes.

La técnica estudiada en este proyecto permite realizar eficientemente este alineamiento y cuantificarlo. Esto nos permite plantear un problema de búsqueda entre una imagen de referencia y un conjunto de imágenes de test. Para llevar a cabo la comparación entre ambas imágenes se utiliza el algoritmo de evaluación BIDM (Binary Image Distorision Model) para realizar un alineamiento no lineal y dotar de flexibilidad a la herramienta desarrollada.

Asimismo, para ayudarnos a evaluar los resultados obtenidos se han utilizado otras técnicas, como la técnica de los k vecinos más cercanos, para poder comparar y evaluar la técnica de alineamiento elegida.

Palabras clave: BIDM, Alineamiento de imágenes de documentos, búsqueda en documentos

Abstract

This project consists of the analysis and development of an alignment technique for document images with the aim of being able to search for information in the images. In particular, this project is based on the analysis and development of a GUI application for the alignment of document images that have a similar layout. The images of documents with a similar layout may present variatios due to the image capture process(slight rotations, curvature, translation, etc.). In this case, the alignment between two images requieres a non-linear comparison between a pair of images.

The technique studied in this project allows to perform this alignment efficiently and to quantify that alignment. This allows us to consider a search problem between a reference image and a set of test templates. To carry out the comparison between both images, the evaluation algorithm BIDM (Binary Image Distorision Model) is used to perform a non-linear alignment and to provide flexibility to the developed tool.

Furthermore, to help us evaluate the results obtained, other techniques such as the k-nearest neighbors algorithm have been used to be able to compare and evaluate the chosen alignment technique.

Key words: BIDM, document images alignment, document search

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

La detección y el procesamiento automático de información en imágenes o documentos es un problema recurrente en las últimas décadas. Actualmente, se poseen grandes colecciones de datos, los cuales principalmente han sido escaneados y se encuentran en formato de imágenes. Debido a esto, es difícil poder procesar dichas imágenes y obtener toda la información de las mismas para utilizarla en sus respectivos ámbitos.

En concreto, el reconocimiento de la información presente en tablas es una cuestión de gran interés, dado que las tablas contienen generalmente toda la información importante resumida. Dicha información se encuentra presente de forma estructurada, teniendo usualmente un número de columnas y filas variables según los datos que se quieren representar, además del contenido asociado al interior de cada casilla. Debido a esta variabilidad, la interpretación de las mismas se debe ajustar al contexto del documento y puede resultar de gran utilidad la existencia de herramientas que ayuden a procesar y a buscar cierta información específica de forma automática en base a información geométrica.

Cuando se desea buscar cierta información en una colección de datos, se deben establecer ciertos criterios de búsqueda. Si la base de datos presenta una larga colección de elementos y además, los criterios de búsqueda son muy específicos, el proceso de obtención y procesamiento de la información es un trabajo arduo.

Debido a la dificultad de dicha tarea, en este trabajo se ha analizado y desarrollado una aplicación para ayudar con la búsqueda y la obtención de información específica presente en documentos tabulares que se desea recuperar. Dicha búsqueda se basa en información geométrica. Para ello, se ha realizado un alineamiento entre distintas imágenes de documentos. Pero para dotar de flexibilidad a la técnica utilizada, el alineamiento se realiza utilizando el algoritmo de evaluación BIDM, detallado en el apartado 3.2, el cual realiza una comparación no lineal entre ambas imágenes.

Cabe destacar que las imágenes deben presentar una maquetación o plantilla similar, pudiendo presentar ciertas variaciones tales como rotación o traslación, debido al proceso de captura de las mismas. Pero esta restricción es importante para obtener un resultado que se asemeje con el resultado esperado de la búsqueda. Esto es debido a que la técnica de alineamiento utilizada permite cuantificar la similitud entre ambas, pero para ello, deben presentar ciertas características comunes.

El algoritmo de alineamiento es capaz de localizar la información que se desea recuperar, dado que aquella información que presente una mayor similitud con la asociada a los criterios de búsqueda, se corresponde con el área de cada imagen con el resultado esperado. De esta forma, la aplicación desarrollada devuelve los fragmentos de las imágenes del conjunto de datos de forma ordenada de mayor a menor semejanza con el resultado que se desea obtener.

1.1 Motivación

El reconocimiento de la información presente en imágenes de documentos, como se ha dicho anteriormente, es un problema que se encuentra presente a día de hoy y debido al alto contenido de colecciones de datos, generalmente imágenes, el desarrollo de herramientas capaces de ayudar con dicha tarea son de gran utilidad.

El reconocimiento de texto manuscrito de imágenes de documentos es una tarea que ha sido resuelta en los últimos años con buenos resultados, sobretodo con la utilización de redes neuronales. En este trabajo, no se pretende realizar un reconocimiento del contenido de la información presente en las imágenes de documentos de la colección utilizada, pero si se pretende poder ayudar a optimizar el mismo. Es decir, la motivación de este trabajo es poder ayudar a los modelos de reconocimiento de texto ya existentes, proporcionándoles la entrada óptima del fragmento de la imagen que se desea reconocer, evitando que se deba reconocer todo el documento al completo.

Por ejemplo, dado el caso de que se quiere recuperar la información presente en la primera casilla de una tabla específica, si se quiere reconocer dicho fragmento de la imagen, se debe realizar un recorte y extraer el contenido de la misma. Con los sistemas de reconocimiento actuales, se debe realizar un preprocesamiento automático para separar dicho fragmento y tras reconocer su contenido con el modelo establecido, se puede obtener el contenido de la casilla. Este proceso se realiza actualmente localizando líneas utilizando información fundamentalmente local. Con nuestra herramienta, lo que se pretende es poder obtener dicho recorte de forma automática de forma óptima para poder facilitar el reconocimiento de la misma, dado que de esta forma ya no se debe hacer un preprocesamiento menos local ni se debe reconocer toda la imagen completa y extraer la información de dicha casilla.

Por lo tanto, se puede concluir que la motivación de este trabajo es ayudar a optimizar los procesos de reconocimiento existentes en el estado del arte para poder centrar el esfuerzo en reconocer únicamente aquella información que es relevante para el estudio que se esté realizando.

1.2 Objetivos

El objetivo principal es analizar y desarrollar una herramienta capaz de alinear distintas imágenes de documentos para poder realizar búsquedas en un conjunto de imágenes determinado. Dicho objetivo se puede dividir en los siguientes objetivos a alcanzar:

- Recopilar un corpus con imágenes de documentos que presenten una maquetación similar.
- Analizar el algoritmo de alineamiento de imágenes 2D y su comportamiento con las imágenes del conjunto de datos.
- Desarrollar e implementar una herramienta que realice el alineamiento y la búsqueda de información en todo el conjunto de datos.
- Devolver de forma ordenada los resultados obtenidos de mayor a menor similitud con los criterios de búsqueda.
- Desarrollar una aplicación gráfica para poder visualizar los resultados obtenidos.
- Realizar distintos experimentos para evaluar los distintos resultados según los criterios establecidos.

1.3 Impacto deseado

El impacto deseado para la herramienta de este trabajo es que se pueda utilizar la técnica desarrollada para preprocesar y ayudar a optimizar las imágenes de entrada de los diferentes modelos de reconocimiento ya existentes actualmente en todos los dominios. En concreto, se pretende ayudar a buscar en grandes colecciones de datos cierta información requerida dado un contexto y ayudar a optimizar el reconocimiento de la misma.

En la actualidad existen modelos que obtienen buenos resultados en el ámbito del reconocimiento de imágenes. Pero para ello, se deben reconocer todas las imágenes en su totalidad y después se debe filtrar la información que se desea obtener.

En la herramienta desarrollada se pretende reconocer las imágenes del conjunto de datos que poseen la información que se desea encontrar y realizar un recorte del fragmento de forma automática. De esta forma, se pretende optimizar la entrada de los modelos de reconocimiento dado que ya no se tiene que realizar el recorte del fragmento de forma manual o con otras técnicas de filtrado suponiendo un gran esfuerzo para grandes colecciones de datos.

1.4 Metodología

La metodología utilizada para el desarrollo de este trabajo se ha basado en una metodología ágil. Esto es debido a que se han ido incorporando mejoras en el trascurso del desarrollo de la aplicación, sin necesidad de rediseñar su funcionamiento.

En primer lugar, se ha procedido a analizar la técnica de alineamiento como eje principal de la herramienta y su comportamiento con el corpus de imágenes establecido. Además se ha realizado un repaso por el estado del arte y las técnicas de preprocesamiento de las imágenes de documentos.

Para poder visualizar los resultados obtenidos gráficamente se ha procedido a analizar y, posteriormente diseñar, según las funcionalidades requeridas, la aplicación gráfica desarrollada. Cabe destacar que el diseño se ha visto influenciado por el desarrollo de la misma debido a que se han realizado pequeños cambios ágiles en el transcurso de su desarrollo.

Por último, se han realizado diversos experimentos para poder evaluar el comportamiento de la herramienta. Gracias a la aplicación gráfica desarrollada se han visualizado los resultados obtenidos para poder ser evaluados.

1.5 Estructura de la memoria

La memoria de este trabajo consta de nueve capítulos. El primer capítulo se corresponde con una introducción a la herramienta desarrollada y el dominio del proyecto.

En el segundo capítulo, se ha realizado un breve repaso sobre las diferente técnicas de reconocimiento de la información en imágenes de documentos. En concreto, las técnicas existentes sobre el preprocesamiento de imágenes escaneadas y los diferentes modelos existente en la literatura para el reconocimiento de información, especialmente en tablas.

En el tercer capítulo se trata la técnica de alineamiento utilizada y las diferentes técnicas en el estado del arte y sus posibles mejoras.

El cuarto y quinto capítulos se corresponden con el análisis y el diseño de la aplicación desarrollada. Para ello, se ha analizado el problema que se desea tratar y se ha diseñado la aplicación que contenga todas las funcionalidades necesarias para resolverlo.

El sexto capítulo detalla el desarrollo de la aplicación desarrollada, explicando con mayor detalle los pasos utilizados en la implementación y en el diseño establecidos de la herramienta.

En el séptimo capítulo se encuentran los distintos experimentos realizados para poder evaluar los resultados obtenidos y poder determinar si la aplicación desarrollada cumple los objetivos establecidos.

El octavo capítulo se corresponde con las conclusiones obtenidas tras el desarrollo de la técnica del proyecto para el reconocimiento y la búsqueda de información en imágenes de documentos.

Y por último, en el noveno capítulo se detallan los trabajos futuros asociados a la herramienta y a las mejoras que pueden incorporarse para optimizar los resultados obtenidos en la misma.

CAPÍTULO 2
Estado del arte

El reconocimiento de imágenes de documentos es una tarea que ha intentado ser resuelta con diversos enfoques en los últimos años, ya sea utilizando algoritmos tradicionales de aprendizaje automático o con las ultimas implementaciones de redes neuronales. Un caso de la búsqueda y extracción de información en imágenes de documentos ha sido llevada a cabo con diferentes modelos de aprendizaje automático por José Ramón Prieto et al. en al artículo [38], obteniendo buenos resultados.

Uno de los mayores problemas de este dominio es la segmentación de las imágenes. En la mayoría de los casos, las imágenes de documentos del dominio de este proyecto, como las imágenes del corpus FUNSD[16], suelen haber sido obtenidas mediante escáner y pueden presentar ruido y variaciones entre ellas. Para poder segmentarlas correctamente, existen técnicas utilizando redes neuronales y ciertas librerías como OpenCV que ayudan a resolver dicho problema [6] [47] .

Por otra parte, el reconocimiento de las tablas presentes en las imágenes de documentos es un problema importante que se debe resolver, debido a que la información presente en las mismas suele corresponderse con información de gran interés. A continuación, se va a realizar un repaso de los modelos implementados para resolver el reconocimiento de tablas presentes en imágenes de documentos siguiendo ambos enfoques hasta la actualidad y sus resultados obtenidos, los cuales se encuentran en el artículo *Current Status and Performance Analysis of Table Recognition in Document Images with Deep Neural Network* [12].

En los sucesivos apartados se van a detallar las diferentes implementaciones según los modelos utilizados para resolver el reconocimiento de tablas a través de la utilización de modelos basados en algoritmos tradicionales de aprendizaje automático y, basados en redes neuronales, respectivamente.

2.1 Reconocimiento de tablas con algoritmos de aprendizaje automático

A finales de los años 90, se publicaron los primeros trabajos relacionados con el reconocimiento de información estructurada en tablas. En concreto, el reconocimiento de tablas en documentos HTML. En ellos, se puede observar cómo se desea reconocer la estructura y realizar una segmentación de los elementos de la misma [7] [25] [31] [52].

Aunque uno de los estudios más antiguos en este área fue realizado por Kieninger et al. en 1998, en el cual utilizando su propio sistema denominado T-Recs extraía la estructura asociada a las tablas de los documentos HTML [22], y, utilizando segmentación en

bloques [23]. Años después aplicó su mismo sistema utilizando documentos asociados a cartas de empresa [24].

En 2002, otro de los pioneros en estudios sobre detección de tablas, Cesarini et al., propuso un sistema denominado Tabfinder. Este sistema convierte un documento en un árbol representando el contenido del mismo a través de una estructura jerárquica y ayudando a detectar si existe una tabla cuando aparecen lineas verticales y horizontales [5].

Wang et al. [54] no sólo se centró en la detección de la tabla, si no que también, se centró en la segmentación de la misma utilizando un algoritmo diseñado para poder entender el contenido utilizando métodos de optimización iterativos.

Los modelos ocultos de Markov (*Hidden-Markov-Models*) fueron utilizados para detectar las áreas tabuladas por Silva et al. [46]. Además, las máquinas de vectores soporte (*Support Vector Machine*) también fueron utilizadas al extraer ciertas características para detectar tablas por Paquet et al. [18]. Fan et al. implementó un sistema para detectar tablas utilizando diversos clasificadores entrenados sobre lingüística y sobre información de la maquetación de una tabla [10].

Todos las implementaciones anteriores comparten en común que se corresponden con sistemas que no son genéricos para cualquier documento de entrada y se deben reajustar para el dominio de la tarea. Esto implica que se debe realizar un procesamiento antes y después para los documentos o imágenes de entrada y salida, respectivamente con su correspondiente esfuerzo.

Este problema desaparece en el uso de sistemas basados en redes neuronales, en concreto, con el uso de redes convolucionales, en las cuales la extracción de características del dominio para hacer el preprocesamiento o la clasificación en los métodos anteriores, son extraídas automáticamente y es capaz de generalizar. En el siguiente apartado, se detalla con mayor profundidad los mejores modelos en el estado del arte utilizando este tipo de sistemas.

2.2 Reconocimiento de tablas con redes neuronales

El reconocimiento de tablas en una imagen de un documento se puede dividir en tres problemas a resolver: detección de la tabla, segmentación de la estructura de la tabla y reconocimiento de la información de la misma.

- Detección de la tabla: Consiste en detectar los límites y la maquetación de la tabla utilizando rectángulos delimitadores.
- Segmentación estructural de la tabla: Consiste en definir la estructura de la tabla analizando la información de las filas y columnas.
- Reconocimiento de la tabla: Consiste en definir la estructura de la tabla delimitada y en reconocer la información de cada celda.

Dependiendo de la tarea que se desee resolver hay diversas técnicas que han obtenido mejor precisión en los resultados obtenidos. En la gráfica presente en la Figura 2.1, la cual ha sido obtenida del artículo [12] previamente citado, se pueden observar los modelos que mejores resultados han obtenido para cada tarea.

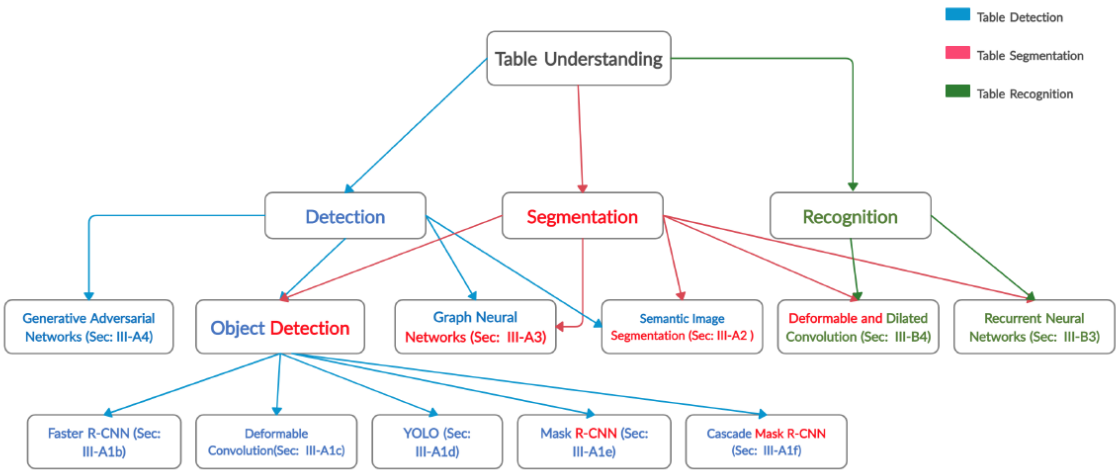


Figura 2.1: Gráfica redes neurales utilizadas en el reconocimiento de tablas

Los métodos utilizados para la detección de tablas se encuentran marcados en azul, los modelos utilizados para realizar la segmentación de la estructura se encuentran en rojo y los arquitecturas utilizadas para realizar el reconocimiento total de la tabla se encuentran en verde. Cabe destacar que varios métodos son compartidos para la resolución de varios problemas, por tanto, en la gráfica se puede observar métodos marcados con varios colores.

La detección de la tabla, como se puede observar en la Figura 2.1, se ha resuelto con métodos basados en redes generativas antagónicas (GAN), métodos de obtención de objetos, redes neuronales gráficas y segmentación semántica de la imagen. A continuación se van a detallar varios modelos relacionados con este problema. La evaluación y comparación de los mismos según el conjunto de datos utilizado puede observarse en la sección 5 del artículo [12].

Li et al. [28] utilizan redes GAN junto con diferentes redes de detección de objetos para demostrar que con la aplicación de redes GAN se obtienen mejores resultados. Esto es debido a que esta arquitectura ayudan a extraer características similares, pero son vulnerables a imágenes de documentos que presenten tablas con diferentes maquetaciones.

Utilizando métodos de detección de objetos cabe destacar el uso de redes *Faster R-CNN*, YOLO, *Deformable Convolutions*, *Mask R-CNN* y *Cascade Mask R-CNN*. La arquitectura de *Faster Region-CNN*, en la cual las tablas son tratadas como objetos dentro de la imagen, han sido utilizadas por *Gelani et al.* [11], *DeepDeSRT* [42], *TableBank* [27] y *Sun et al.* [48]. *DeCNT* [43] también ha utilizado la arquitectura Faster R-CNN con convoluciones deformables.

La arquitectura de YOLO (*You Only Look Once*) ha sido utilizada para la detección de tablas por *Huang et al.* [15] y *Garcia et al.* [4]. El modelo *Cascade Mask R-CNN* ha sido implementado para esta tarea por *CascadeTabNet* [37] y *CDec-Net* [1]. Además, *GTE* [56] ha propuesto un algoritmo de detección de objetos genérico.

Por otra parte, otras arquitecturas basadas en redes neuronales gráficas han sido desarrolladas por *Martin et al.* [14] y por *Riba et al.* [41]. E implementaciones basadas en segmentación de imagen y redes totalmente convolucionales han sido desarrolladas por *Kavasidis et al.* [19] y *TableNet* [34], respectivamente.

El problema de la segmentación estructural de la tabla se ha intentado resolver utilizando métodos basados en detección de objetos, redes neuronales gráficas, redes convo-

lucionales y redes recurrentes. *CascadeTabNet* [37], *Hashmi et al.* [13] y *Raja et al.* [40] han utilizado arquitecturas basadas en Region-CNN utilizando una máscara (Mask R-CNN). *Siddiqui et al.* [44] ha empleado *Faster R-CNN* con convoluciones deformables. Además, el modelo genérico de *GTE* [56] también se ha empleado para intentar resolver esta tarea.

Si se desea resolver el problema con redes neuronales gráficas, *Qasim et al.* [39], *Xue et al.* [55] presentan sus modelos en sus respectivos artículos. Igualmente *Khan et al.* [21] utiliza la arquitectura de redes recurrentes para implementar su solución, y, *Siddiqui et al.* [45], *Tensmeyer et al.* [49] y *Zou et al.* [58] han utilizado redes totalmente convolucionales para defender su modelo de reconocimiento de la estructura.

Por último, para resolver la tarea de reconocimiento de la tabla, en la literatura, se ha encontrado las implementaciones de *Zhong et al.* [57] y *Deng et al.* [8]. Ambas implementaciones se basan en la arquitectura de redes *Encoder Decoder* con y sin mecanismos de atención respectivamente.

Los resultados de la evaluación de todos estos modelos, tal y como se ha comentado anteriormente se puede consultar en el artículo [12]. Dicho artículo se ha utilizado de referencia para realizar un repaso por el estado del arte del dominio del reconocimiento de imágenes de documentos, y más en detalle, de las tablas presentes en los mismos.

En los experimentos, se puede observar que dependiendo del conjunto de datos utilizado varía el método que mejores resultados obtiene. Gracias a la celebración de ICDAR (*International Conference on Document Analysis and Recognition*) [9], y a su competición asociada, se ha conseguido establecer un conjunto de datos común para poder evaluar y comparar las distintas soluciones propuestas.

Cada año, se puede observar que, dicho conjunto de datos se actualiza y los resultados de los modelos presentados en conferencias anteriores pueden variar. Esto es debido a la variabilidad de las muestras utilizadas y demuestra la dificultad de esta tarea para encontrar una solución que obtenga buenos resultados para cualquier conjunto de datos.

Cabe destacar que las modelos presentados en este apartado obtienen generalmente tasas de acierto relativamente altas con precisiones y valores de F-1 superiores al 95 % para los problemas de detección de tabla y segmentación de la estructura de la misma. Sin embargo, la tarea de reconocimiento total de la tabla presenta un margen de mejora, debido a la complejidad de la misma.

CAPÍTULO 3
Análisis de la técnica de alineamiento

El algoritmo BIDM [3] es el modelo principal utilizado para realizar la búsqueda y reconocimiento entre las imágenes de nuestro conjunto de datos. En este capítulo, se pretenden analizar las diferentes técnicas de alineamiento no lineal existentes en la literatura para imágenes de dos dimensiones y, explicar de forma detenida la técnica de alineamiento utilizada en nuestra aplicación. Además, para intentar optimizar el resultado obtenido para un alineamiento dado, se ha utilizado el algoritmo húngaro [2], el cual se detallará a continuación en su apartado correspondiente.

3.1 Técnicas de alineamiento para imágenes 2D

Keysers et al. [20] describen en su artículo los diferentes modelos que existen en la literatura capaces de realizar un alineamiento no lineal entre imagenes en 2 dimensiones. Dichas implementaciones se corresponden con el modelo 2DW (*Two-dimensional model*), el modelo P2DHMM (*Pseudo 2-Dimensional Hidden Markov Model*), el modelo P2DHMDM (*Pseudo 2-Dimensional Hidden Markov Model*) y el modelo IDM (*Image Distortion Model*). En la Figura 3.1, obtenida del artículo previamente citado, se puede observar los algoritmos y el procedimiento realizado para aplicar cada método gráficamente.

Los alineamientos pueden clasificarse según la dependencia del alineamiento de un píxel con respecto al alineamiento de sus vecinos. Es decir, se pueden clasificar según la influencia que tienen los alineamientos de los píxeles vecinos para realizar el alineamiento de un píxel de test dado.

Un alineamiento entre dos píxeles es de orden cero si para realizar el alineamiento de un píxel, con otro píxel de una imagen de referencia, no se tienen en cuenta los alineamientos de sus píxeles de test vecinos. Un alineamiento es de orden uno si el alineamiento de un píxel se realiza teniendo el cuenta el alineamiento de sus píxeles adyacentes en un eje, y por último, un alineamiento es de orden dos si se tienen en cuenta los alineamientos de los píxeles vecinos en ambos ejes para el alineamiento de un píxel dado.

El modelo 2DM es un modelo de orden dos, cuyos píxeles vecinos en ambos ejes no pueden corresponderse con píxeles que se desvíen más de un píxel con respecto a su posición relativa en la imagen original. El modelo P2DHMM es un modelo de orden uno con respecto a los píxeles vecinos en el eje horizontal. En este modelo, el desplazamiento horizontal es el mismo para todos los píxeles de una columna y no existe desplazamiento vertical. Por otra parte, el modelo P2DHDHMDM, que también es de orden uno, se corresponde con una implementación relajada del modelo anterior P2DHMM. Esto es de-

bido a que se permite como máximo un desplazamiento vertical de un píxel. Finalmente, el modelo IDM es de orden cero y tiene en cuenta otros parámetros como la ventana de contexto local y el rango de distorsión establecidos.

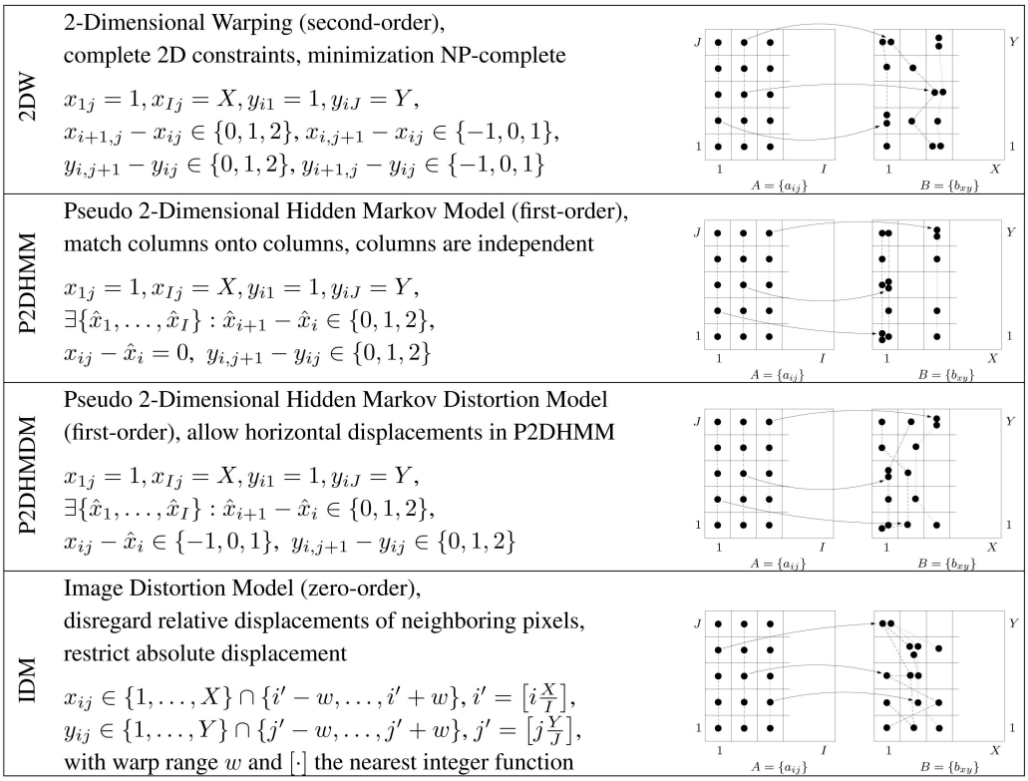


Figura 3.1: Modelos de alineamiento no lineal en 2 dimensiones

En la experimentación realizada en [20], los mejores resultados fueron obtenidos por el modelo P2DHDHMDM, aunque el modelo IDM presentaba una tasa de error semejante. Debido al alto coste computacional que supone la dependencia de los píxeles adyacentes para realizar el alineamiento de todos los píxeles de una imagen, y los resultados obtenidos, se puede concluir que el modelo IDM en relación coste computacional y tasa de error, es el mejor modelo.

Debido a esto, se ha utilizado una implementación basada en el modelo IDM junto con algunas variaciones en su implementación, la cual se detalla a continuación en el siguiente apartado.

3.2 Algoritmo de alineamiento no lineal BIDM

El algoritmo BIDM es un algoritmo de alineamiento no lineal entre dos imágenes: una imagen de test y una imagen de referencia. Para ello, el algoritmo realiza el alineamiento según un rango de distorsión y un contexto local, que deben ser proporcionados. En la Figura 3.2, se puede observar el código del algoritmo, extraído del artículo [3]. Las entradas del algoritmo se corresponden con una imagen de test (cuyas dimensiones son IxJ), una imagen de referencia (sus dimensiones son XxY), un rango de alineamiento W y un contexto local C .

El algoritmo recorre todos los píxeles de la imagen de test y para cada píxel, intenta encontrar el píxel de la imagen de referencia dentro del rango de alineamiento que minimice la desigualdad entre ambos. Dicha desigualdad se calcula con la diferencia entre las derivadas verticales y horizontales de ambos píxeles. La diferencia de este algoritmo con respecto al modelo IDM, previamente citado, se corresponde con el cálculo de las derivadas y la diferencia entre las mismas, dado que para BIDM se utiliza un método simplificado detallado en [51].

Una vez todos los píxeles de la imagen de test han sido alineados con la imagen de referencia, se calcula el número de píxeles con un buen alineamiento y se devuelve el porcentaje de los mismos. El coste temporal del algoritmo es $O\,(IJw^2c^2)$.

Input: test image A $(I \times J)$, warp range w
reference image B $(X \times Y)$, context window size c
Output: BIDM(w, c) from A to B

$A^v = \text{vertical\_der}(A)$; $A^h = \text{horizontal\_der}(A)$
$B^v = \text{vertical\_der}(B)$; $B^h = \text{horizontal\_der}(B)$
for $i = 1$ to I **do** {
  for $j = 1$ to J **do** {
    $i' = \lfloor i\frac{X}{I} \rfloor$, $j' = \lfloor j\frac{Y}{J} \rfloor$, $z = \lfloor \frac{c}{2} \rfloor$
    $S_1 = \{1, \ldots, X\} \cap \{i' - w, \ldots, i' + w\}$
    $S_2 = \{1, \ldots, Y\} \cap \{j' - w, \ldots, j' + w\}$
    $s = \min_{\substack{x \in S_1 \\ y \in S_2}} \sum_{m=-z}^{z} \sum_{n=-z}^{z} (A^v_{i+n,j+m} - B^v_{x+n,y+m})^2 + (A^h_{i+n,j+m} - B^h_{x+n,y+m})^2$
    $map(i, j) = s$
  }
}
normalize_depth(map, 255)
binarize(map) *//Otsu's method*

$\text{fg} = \{(x, y) \mid A(x, y) < 255\}$ *//Foreground pixels*
$\text{cp} = \text{fg} \cap \{(x, y) \mid map(x, y) = 0\}$ *//Correct pixels*

return $\frac{|cp|}{|fg|}$ *//Correct pixels ratio*

Figura 3.2: Algoritmo BIDM

Dado que el algoritmo BIDM es la base de la herramienta desarrollada en este trabajo, es clave poder entender su funcionamiento. A continuación, se detalla de forma gráfica el comportamiento del mismo. En primer lugar, en las Figuras 3.4 y 3.5 se puede observar la diferencia entre un algoritmo lineal y no lineal.

En un alineamiento entre dos píxeles linealmente, ambas imágenes deben tener las mismas dimensiones de altura y ancho, y por tanto, el mismo número de píxeles para poder alinear cada píxel con su píxel asociado. Un algoritmo no lineal es más flexible y permite comparar imágenes con distintas dimensiones y alinear cada píxel con su píxel proporcional según el alto y ancho de ambas imágenes.

Además, el algoritmo utilizado permite que varios píxeles de la imagen de test se alineen con un mismo píxel en la imagen de referencia. La relación de los posibles alineamiento se corresponde con una función no inyectiva y no sobreyectiva (véase Figura 3.3). Esto es debido a que dos o mas píxeles de la imagen de test (X) pueden asociarse con un mismo píxel de la imagen de referencia. Dicha función de alineamiento asociada no es inyectiva debido a que todos los píxeles de la imagen de test deberían asociarse con un píxel distinto, sin repeticiones, de la imagen Y de referencia.

Por otra parte, la función asociada al alineamiento no es sobreyectiva porque hay píxeles de la imagen de referencia (Y) que no se alinean con ningún píxel de la imagen de test (X), y para poder establecer que la relación se corresponde con una función sobreyectiva, todos los elementos del codominio deben estar alineados. Por tanto, la función que se corresponde con el alineamiento realizado entre los píxeles de las imágenes de nuestro problema se corresponde con una función no inyectiva y no sobreyectiva.

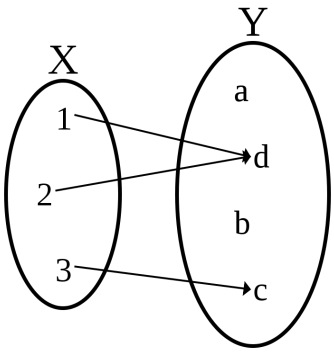


Figura 3.3: Función de alineamiento no inyectiva y no sobreyectiva

En la Figura 3.4, se puede observar un píxel de la imagen de test que se alinea con su píxel proporcional lineal en la imagen de referencia. Sin embargo, en la Figura 3.5, se puede observar el funcionamiento del algoritmo no lineal, en el cual un píxel de la imagen de test puede alinearse con un píxel de la imagen de referencia que minimice su diferencia y que se encuentre a w píxeles de su píxel proporcional lineal. Es decir, el píxel de la imagen de test se puede alinear con cualquier píxel de la ventana de 2w+1 x 2w+1, cuyo centro se corresponde con su píxel lineal asociado.

Por ejemplo, el píxel de test, se puede alinear con el píxel marcado en verde en la imagen de referencia. Cabe destacar que el algoritmo recorre todos los píxeles de la ventana determinada por el rango de distorsión de izquierda a derecha y de arriba a abajo, y alinea el píxel de test con el píxel cuya diferencia sea mínima. Esto es debido a que a menor diferencia, ambos píxeles son más parecidos y si la diferencia es cero, ambos píxeles son iguales.

Figura 3.4: Alineamiento lineal

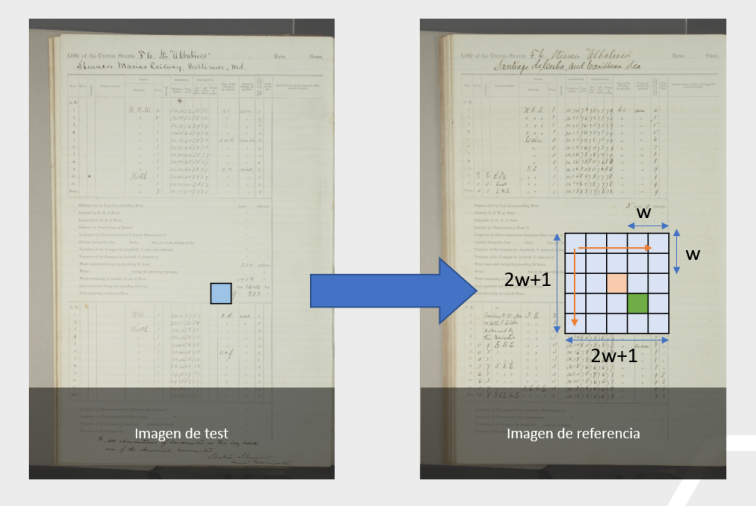


Figura 3.5: Alineamiento no lineal utilizando rango de distorsión (W)

El algoritmo BIDM no sólo tiene en cuenta el rango de distorsión determinado por W, sino que también hace el alineamiento teniendo en cuenta el contexto local de cada píxel, es decir, teniendo en cuenta a sus píxeles vecinos. En la Figura 3.6 se puede observar gráficamente cómo para un píxel de la imagen de test, se debe tener en cuenta a los píxeles vecinos marcados en rojo determinados por el tamaño de la ventana del contexto determinada por *C*.

Dicho píxel y su contexto, deben ser alineados con un píxel de la ventana del rango de distorsión determinada por W, de igual forma que en la figura anterior. Pero, en este caso, para realizar la diferencia entre ambos píxeles también se debe tener en cuenta el contexto local de los píxeles de referencia. Es decir, un píxel de la imagen de test se alinea con el píxel de la imagen de referencia cuya diferencia entre ambos píxeles y sus respectivos contextos locales (píxeles vecinos) sea mínima.

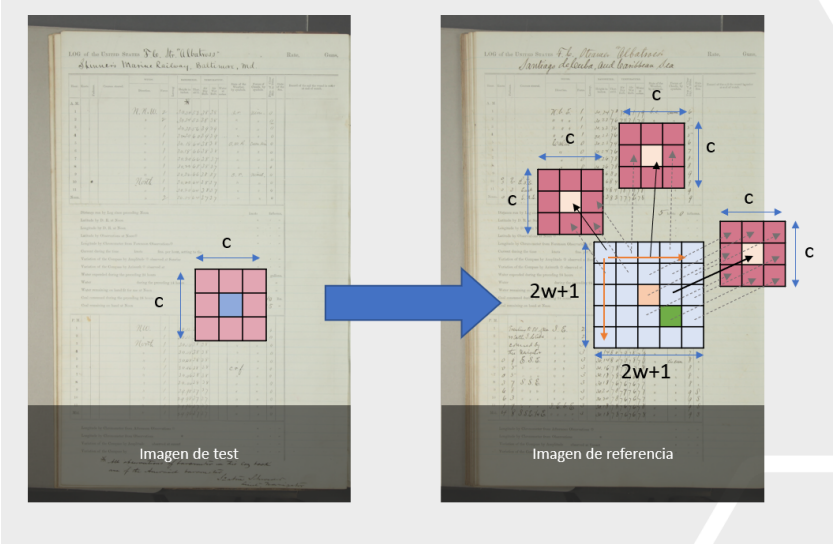


Figura 3.6: Algoritmo BIDM- Alineamiento no lineal utilizando rango de distorsión (W) y contexto (C)

El algoritmo alinea todos los píxeles de una imagen de test con respecto a la imagen de referencia siguiendo el procedimiento de la Figura 3.6 para cada píxel, tal y como se puede observar en la Figura 3.7. Es decir, el algoritmo recorre todos los píxeles de la imagen de test de izquierda a derecha, de arriba a abajo, y para cada píxel extrae la matriz de diferencias entre dicho píxel y los píxeles de la imagen de referencia dentro de la ventana del rango de alineamiento teniendo en cuenta el contexto local de todos los píxeles. Tal y como se puede observar cada píxel de la imagen de test tiene asociada una matriz de diferencias para determinar su alineamiento.

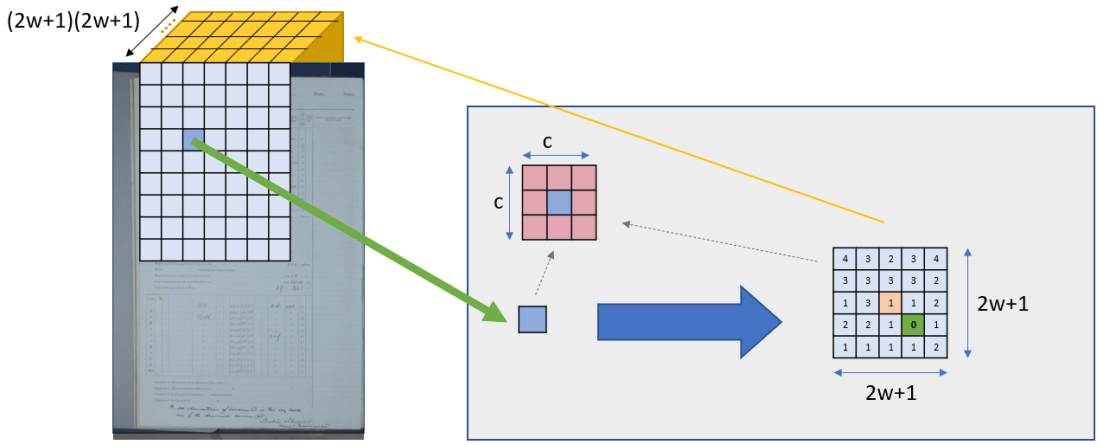


Figura 3.7: Algoritmo BIDM - Alineamiento píxeles

Cabe recordar que el algoritmo no limita que varios píxeles de la imagen de test se alineen con un mismo píxel de la imagen de referencia. En la Figura 3.8 se puede observar dicho caso, en el cual dos píxeles de la imagen de test se alinean con el mismo píxel. Si se quisiera establecer dicha restricción, el coste del algoritmo aumentaría considerablemente, siendo exponencial. Por tanto, para ello, se ha considerado la incorporación de un algoritmo de optimización, el algoritmo húngaro, para poder establecer dicha limitación

minimizando las diferencias de los alineamientos. En el siguiente apartado, se detalla su incorporación.

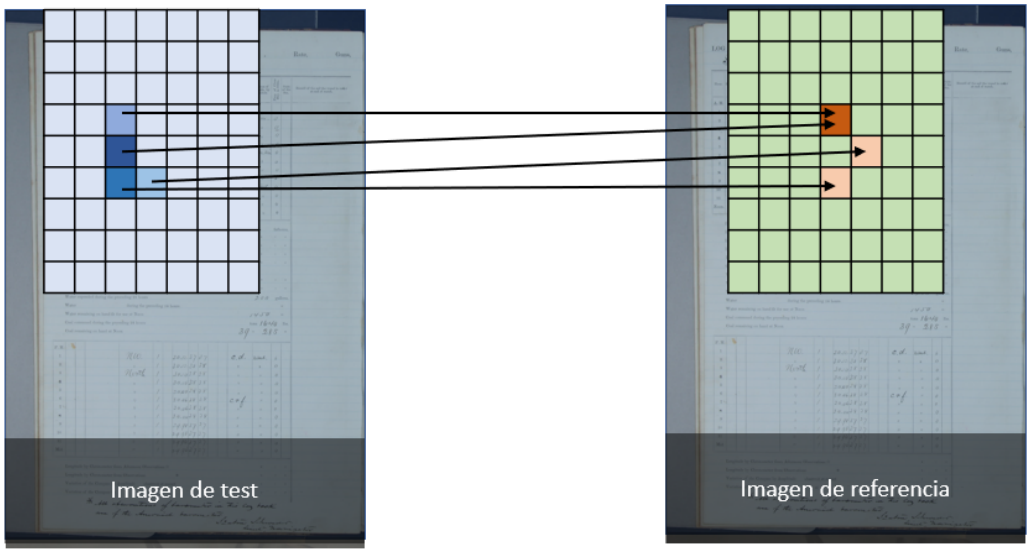


Figura 3.8: Algoritmo BIDM - Alineamiento

3.2.1. Incorporación del algoritmo húngaro al alineamiento

El algoritmo húngaro [2] es un algoritmo que se utiliza para minimizar el coste de asignación. El ejemplo básico que se suele utilizar es la asignación de N tareas a N trabajadores. Es decir, se establece una matriz de costes que representa lo que le cuesta a un trabajador realizar cada tarea y tras los pasos del algoritmo, se obtiene el mínimo y la asignación de cada trabajador a su tarea. Por ejemplo, en la tabla siguiente, se puede observar un ejemplo del problema de asignación de tareas a trabajadores con N=5, cuyo coste mínimo es 21.

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5
Trabajador 1	3	8	2	10	**3**
Trabajador 2	8	7	**2**	9	7
Trabajador 3	6	**4**	2	7	5
Trabajador 4	8	4	2	**3**	5
Trabajador 5	**9**	10	6	9	10

Tabla 3.1: Ejemplo algoritmo húngaro

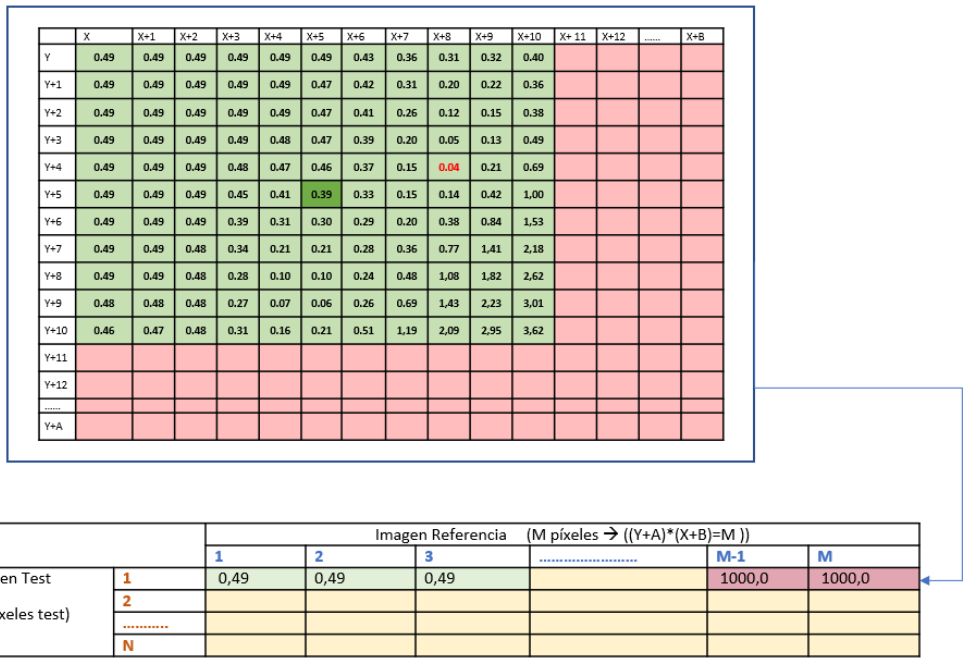
En nuestro caso, siguiendo la misma idea del problema de asignación, lo que se quiere es asignar 1 píxel de la imagen de test con 1 píxel de la imagen de referencia. Por tanto, se puede aplicar el problema de asignación para N píxeles de test, que se deben asociar con M píxeles de referencia.

La aplicación original del algoritmo suele realizarse en matrices cuadradas, es decir sería para N píxeles se quieren asignar a N píxeles de la imagen de referencia. Pero, esto es debido a que en los problemas de ejemplo de este dominio se quiere que todas las tareas se realicen, o que todos los píxeles de ambas imágenes estén asociados.

En nuestro caso, esto no tiene por qué ser así, pueden existir píxeles de la imagen de referencia sin alinear. Lo que sí se debe cumplir es que todos los píxeles de la imagen de test estén alineados. Por tanto, si se traslada este problema a una matriz de costes igual que el problema de asignación de tareas, se puede resolver con una matriz que sea de NxM, como la matriz de la Figura 3.9.Dicha matriz se encuentra rellenada con los valores de las diferencias entre cada par de píxeles. Estas diferencias se corresponden con los valores obtenidos en el alineamiento BIDM entre las diferencias de las derivadas de cada píxel con sus respectivos píxeles de la imagen de referencia dentro de la ventana de su rango de alineamiento.

Si dos píxeles se pueden alinear, en la casilla correspondiente en la matriz se rellena su valor asociado en la matriz del alineamiento en BIDM. Si por el contrario, dos píxeles no pueden llegar a ser alineados, dado que el píxel de referencia no se encuentra dentro de la ventana del rango de alineamiento del píxel de test, su valor dentro de la matriz se establece a 1000.0. De esta forma, se limita a que todos los píxeles de la imagen de test se intenten alinear con un píxel dentro de su rango en la imagen de referencia, intentando minimizar el coste general asociado a las diferencias, y por tanto, optimizando el alineamiento.

El algoritmo obtiene el mínimo coste de alineación de cada píxel de N con un píxel de M. Es decir, realiza N asignaciones, todos los píxeles de la imagen de test se alinean con un píxel de la imagen de referencia. Y si se quedan píxeles de la imagen de referencia sin alinear, se descartan del problema.



	X	X+1	X+2	X+3	X+4	X+5	X+6	X+7	X+8	X+9	X+10	X+ 11	X+12	.....	X+B
Y	0.49	0.49	0.49	0.49	0.49	0.49	0.43	0.36	0.31	0.32	0.40				
Y+1	0.49	0.49	0.49	0.49	0.49	0.47	0.42	0.31	0.20	0.22	0.36				
Y+2	0.49	0.49	0.49	0.49	0.49	0.47	0.41	0.26	0.12	0.15	0.38				
Y+3	0.49	0.49	0.49	0.49	0.48	0.47	0.39	0.20	0.05	0.13	0.49				
Y+4	0.49	0.49	0.49	0.48	0.47	0.46	0.37	0.15	0.04	0.21	0.69				
Y+5	0.49	0.49	0.49	0.45	0.41	0.39	0.33	0.15	0.14	0.42	1,00				
Y+6	0.49	0.49	0.49	0.39	0.31	0.30	0.29	0.20	0.38	0.84	1,53				
Y+7	0.49	0.49	0.48	0.34	0.21	0.21	0.28	0.36	0.77	1,41	2,18				
Y+8	0.49	0.49	0.48	0.28	0.10	0.10	0.24	0.48	1,08	1,82	2,62				
Y+9	0.48	0.48	0.48	0.27	0.07	0.06	0.26	0.69	1,43	2,23	3,01				
Y+10	0.46	0.47	0.48	0.31	0.16	0.21	0.51	1,19	2,09	2,95	3,62				
Y+11															
Y+12															
.....															
Y+A															

		Imagen Referencia (M píxeles → ((Y+A)*(X+B)=M))					
		1	2	3	.....	M-1	M
Imagen Test	1	0,49	0,49	0,49		1000,0	1000,0
(N píxeles test)	2						
	.....						
	N						

Figura 3.9: Algoritmo húngaro - Tabla con matriz de costes BIDM

En la Figura 3.10, se pueden observar los resultados obtenidos realizando el alineamiento entre dos imágenes utilizando el algoritmo BIDM, y utilizando el algoritmo BIDM y el algoritmo húngaro, conjuntamente. Las imágenes de referencia muestran los píxeles marcados en rojo que han sido alineados por los píxeles del área determinada en la imagen de test.

Si se comparan ambas imágenes de referencia, se puede observar que en el alineamiento con BIDM y el algoritmo húngaro, el área alineada se corresponde con un número de píxeles mayor y la información que se debería alinear (valor '36') se encuentra seleccionada por completo. Sin embargo, en la implementación con sólo el algoritmo BIDM, los píxeles alineados son inferiores y la información que se debería alinear no ha sido alineada en su totalidad, pero si ha seleccionado la mayoría de los píxeles de la información a alinear. En los experimentos realizados en el capítulo 7 se evalúa la incorporación del algoritmo húngaro al algoritmo BIDM en los resultados obtenidos en la aplicación analizada y desarrollada en los siguientes capítulos.

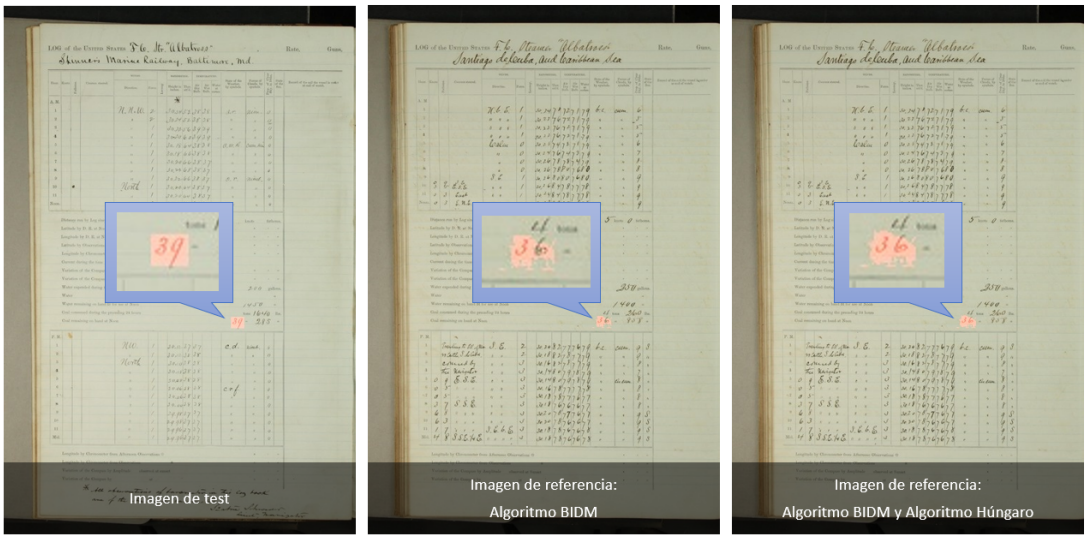


Figura 3.10: Ejemplo alineamiento BIDM con y sin la aplicación del algoritmo húngaro

CAPÍTULO 4
Análisis de la aplicación

En este capítulo, se pretende analizar la aplicación a desarrollar para poder observar el alineamiento y la búsqueda de información entre distintas imágenes de documentos. En primer lugar, se va a analizar el corpus de datos de las imágenes de documentos que se van a utilizar y en segundo lugar, en los siguientes apartados, se va a realizar un análisis de la herramienta a desarrollar.

4.1 Corpus de datos de imágenes de documentos

El corpus de imágenes de documentos se encuentra formado por páginas del cuaderno de bitácoras del barco *Albatross*. Dichas imágenes han sido obtenidas del conjunto de datos HisClima [38]. En ellas se pueden distinguir diferentes libros con su correspondiente maquetación. Cabe destacar que todas las maquetaciones son similares aunque no iguales.

En concreto, el conjunto de imágenes de Albatross está formado por seis volúmenes diferentes en los cuales se registra la información, normalmente cada hora, referente a las condiciones de navegación, a la presión atmosférica, la dirección del viento, etc. Todas las páginas están divididas en dos partes. La parte de arriba se corresponde con la información asociada al periodo AM de cada día, la parte inferior se corresponde con el periodo PM, y en medio, según el cuaderno, se puede anotar información adicional según corresponda.

Según el cuaderno de bitácoras seleccionado, además de diferencias en la maquetación, se pueden observar diferencias en los estilos de escritura y la información puede encontrarse fuera de las celdas y con diferentes marcas de escritura. Esta variabilidad dificulta la búsqueda y el alineamiento de las imágenes.

4.2 Análisis de requisitos de la aplicación

Para realizar el análisis de requisitos de la herramienta, se han debido analizar los requisitos funcionales y no funcionales de la misma. Dichos requisitos se encuentran a continuación.

4.2.1. Requisitos funcionales

Los requisitos funcionales se pueden observar en el diagrama de casos de uso de la Figura 4.1. En dicho diagrama, se puede observar al usuario de la aplicación, cómo único actor de la herramienta. Dicho usuario debe ser una usuario experto, que conozca el funcionamiento de la herramienta previamente. Los diferentes casos de uso de acciones que puede realizar dicho usuario se encuentran detallados en las tablas consecutivas al diagrama.

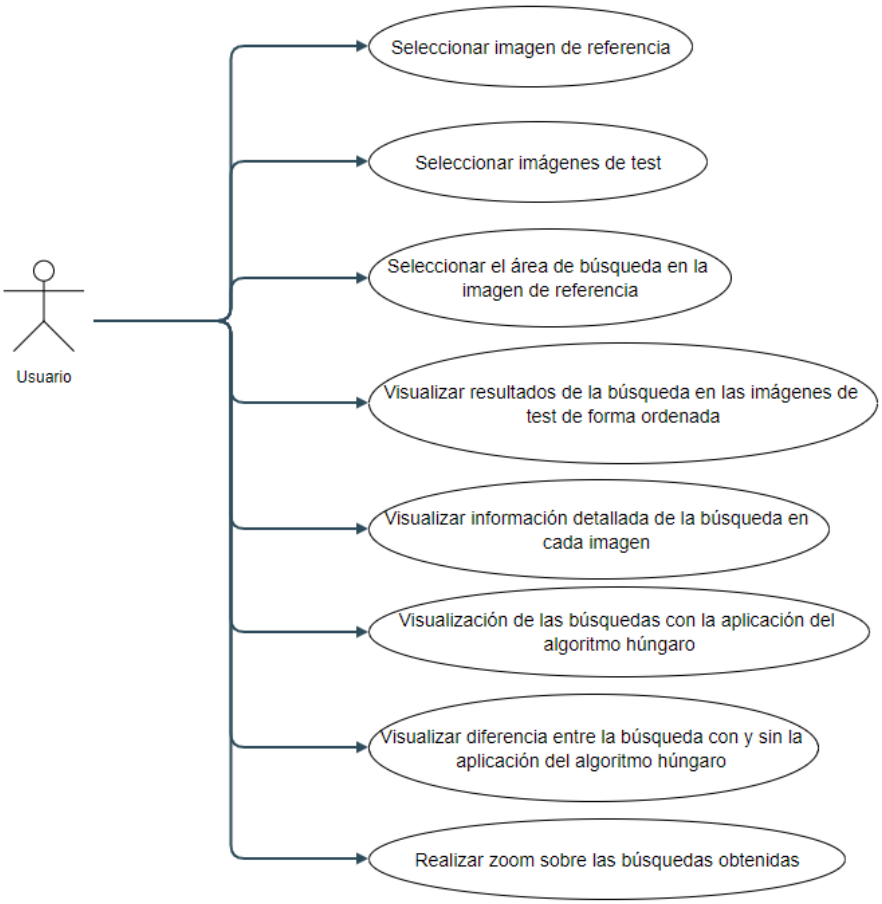


Figura 4.1: Diagrama de casos de uso

Caso de Uso	Seleccionar imagen de referencia
Actores	Usuario experto
Resumen	Permite al usuario elegir la imagen de referencia para determinar la información a buscar en el resto de imágenes de test.
Precondiciones	
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona la imagen de referencia.	
	2. El sistema visualiza la imagen de referencia elegida.

Tabla 4.1: Seleccionar imagen de referencia

Caso de Uso	Seleccionar imágenes de test
Actores	Usuario experto
Resumen	Permite al usuario elegir las imagen de test para buscar la información requerida.
Precondiciones	
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona las imágenes de test.	
	2. El sistema lee las imagen de test establecidas.

Tabla 4.2: Seleccionar imágenes de test

Caso de Uso	Seleccionar el área de búsqueda en la imagen de referencia
Actores	Usuario experto
Resumen	Permite al usuario determinar el área de búsqueda en la imagen de referencia.
Precondiciones	La imagen de referencia y las imágenes de test deben estar seleccionadas
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario selecciona el área de búsqueda sobre la imagen de referencia con el ratón.	
	2. El sistema muestra el área de búsqueda seleccionada.

Tabla 4.3: Seleccionar área de búsqueda

Caso de Uso	Visualizar resultados de la búsqueda en las imágenes de test
Actores	Usuario experto
Resumen	Permite al usuario ver la información buscada en las imágenes de test según el área seleccionada en la imagen de referencia.
Precondiciones	La imagen de referencia debe tener seleccionada el área de búsqueda
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario pulsa el botón para realizar la búsqueda.	
	2. El sistema realiza la búsqueda en todas las imágenes de test.
	3. El sistema muestra los resultados obtenidos de forma ordenada según similitud (de mayor a menor) con respecto al área de búsqueda de la información seleccionada.

Tabla 4.4: Visualizar resultados de la búsqueda en las imágenes de test

Caso de Uso	Visualizar información detallada de la búsqueda en cada imagen
Actores	Usuario experto
Resumen	Permite al usuario ver la información con mayor detalle de una imagen y el área resultado de la búsqueda de información en ella.
Precondiciones	La búsqueda ha debido ser realizada
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario elige la imagen que desea ver en detalle.	
	2. El sistema muestra el área obtenida del alineamiento y toda la información asociada a la imagen en una nueva ventana.

Tabla 4.5: Visualizar información detallada de la búsqueda en cada imagen

Caso de Uso	Visualización de las búsquedas con la aplicación del algoritmo húngaro
Actores	Usuario experto
Resumen	Permite al usuario ver los resultados de la información obtenida aplicando el algoritmo húngaro.
Precondiciones	La búsqueda ha debido ser realizada
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario elige la imagen que desea ver en detalle con la incorporación del algoritmo húngaro a la búsqueda.	
	2. El sistema muestra el área obtenida del alineamiento utilizando el algoritmo húngaro y toda la información asociada a la imagen en una nueva ventana.

Tabla 4.6: Visualización de las búsquedas con la aplicación del algoritmo húngaro

Caso de Uso	Visualizar diferencia entre la búsqueda con y sin la aplicación del algoritmo húngaro
Actores	Usuario experto
Resumen	Permite al usuario ver la diferencia entre los resultados obtenidos utilizando el algoritmo húngaro y sin utilizarlo.
Precondiciones	La búsqueda ha debido ser realizada y se ha debido seleccionar la visualización de una imagen en detalle.
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario pide visualizar la diferencia entre los resultados obtenidos con y sin utilizar el algoritmo húngaro a través de un evento (por ejemplo, un botón) .	
	2. El sistema muestra una imagen que representa las diferencias entre ambos resultados en una nueva ventana.

Tabla 4.7: Visualizar diferencia entre la búsqueda con y sin la aplicación del algoritmo húngaro

Caso de Uso	Realizar zoom sobre las búsquedas obtenidas
Actores	Usuario experto
Resumen	Permite al usuario ver con detalle los resultados obtenidos en las diferentes imágenes.
Precondiciones	La búsqueda ha debido ser realizada y se ha debido seleccionar la visualización de una imagen en detalle.
Postcondiciones	–
Incluye	–
Extiende	–
Hereda de	–
Flujo de eventos	
Actor	Sistema
1. El usuario acerca y aleja la imagen con el ratón .	
	2. El sistema redimensiona la imagen.

Tabla 4.8: Realizar zoom sobre las búsquedas obtenidas

4.2.2. Requisitos no funcionales

El principal requisito no funcional de la herramienta es que la aplicación debe ejecutarse en un entorno de desarrollo Linux. Esto es debido a que la codificación de las imágenes utilizadas para la ejecución del algoritmo BIDM, dado que en el código del mismo proporcionado por los autores [3] se producen problemas si las imágenes se codifican con carácteres especiales de otro entorno.

Otro requisito es que la aplicación debe ser intuitiva y sencilla para el uso. Además, la búsqueda en las imágenes de test y su posterior visualización, no debe tardar más de 15-20 minutos, según el área seleccionada. Esto es debido a que se desea una aplicación que permita visualizar las búsquedas en el menor tiempo posible.

4.3 Análisis del marco legal y ético

La herramienta desarrollada no está enfocada para ser utilizada por cualquier usuario, sino que debe ser utilizada por un usuario experto. Dicho usuario debe tener permisos en el dominio del problema en el cual nuestra aplicación va a ser utilizada. Es decir, el usuario debe poder visualizar e interpretar los resultados obtenidos para el conjunto de datos utilizado para realizar las búsquedas.

Este requisito es debido a que la aplicación almacena cierta información temporal para aligerar el proceso de búsqueda y sólo un usuario experto debe hacer uso de ella para asegurar un correcto uso.

4.4 Análisis de soluciones posibles y solución propuesta

Actualmente, los lenguajes más utilizados y con mayor popularidad son Java y Python. Este último destaca por ser un lenguaje que hace hincapié en la legibilidad y sencillez del código a alto nivel y por tanto, se está usando para múltiples aplicaciones en todos los ámbitos.

Debido a esto y a que presenta múltiples librerías para desarrollar una aplicación gráfica, se ha utilizado dicho lenguaje para el desarrollo de nuestra herramienta. En concreto, se va a utilizar la librería de Python llamada Tkinter para el desarrollo de la interfaz gráfica.

Además, este lenguaje es compatible con OpenCV para el procesamiento de las imágenes se va a utilizar y otras librerías como NumPy, que permiten optimizar el código.

Por último, este proyecto se centra en la evaluación de la técnica de alineamiento utilizando el algoritmo BIDM, debido a que se tiene experiencia previa personal con este algoritmo en el dominio de imágenes médicas (véase [17]) y se pretende evaluar y ver el comportamiento de dicha técnica en imágenes de documentos.

CAPÍTULO 5
Diseño de la aplicación

En este capítulo, se va a proceder a detallar el diseño detallado de la herramienta desarrollada. En concreto, se va a detallar el diseño de la interfaz realizada para que el usuario pueda realizar todas las funcionalidades descritas en el capítulo anterior. Además, se van a detallar las tecnologías utilizadas para la implementación de la solución desarrollada en el capítulo siguiente.

5.1 Diseño detallado

El diseño de la interfaz puede observarse en la Figura 5.1. El diseño de las diferentes ventanas que se pueden observar en la figura con su correspondiente flujo, se ha realizado siguiendo el flujo de la aplicación determinado en la Figura 5.2. En ambas figuras se quiere diseñar e implementar las funcionalidades requeridas por el usuario.

Para ello, en primer lugar, en la ventana asociada al paso 1, se debe establecer la imagen de referencia para seleccionar la información a buscar en las distintas imágenes de test asociadas a la misma. En concreto, se va a establecer una imagen de referencia por cada libro presente en el corpus de imágenes, y sus imágenes de test asociadas serán las imágenes restantes de dicho volumen. En la sección 6.3, se explica con mayor detalle el desarrollo de la relación entre las imágenes de referencia y de test y su correspondiente búsqueda.

Cuando el usuario ha seleccionado una imagen de referencia, éste debe seleccionar un área de búsqueda dibujando sobre ella el área a buscar. Una vez dicha área haya sido delimitada, se debe pulsar el botón asociado a la búsqueda en el conjunto de imágenes de test establecido. En la ventana del paso 2 aparecerán todas las imágenes del conjunto de test ordenadas con el área de la información requerida delimitada y ordenadas de mayor a menor, según similitud con la información a buscar en la imagen de referencia.

Si el usuario desea ver con mayor detalle el alineamiento y la búsqueda de información en una imagen dada, a través de la tabla resumen de todas las imágenes, se puede observar en una nueva ventana la información asociada a una imagen. En concreto, se puede visualizar los píxeles alineados en la imagen de test y en la imagen de referencia y el recorte de la información devuelta como el resultado de la búsqueda. Aunque dicho resultado, se puede observar con mayor detalle en una nueva ventana, realizando zoom, si se pulsa con el ratón sobre la imagen del recorte.

Además, en la misma ventana, se puede modificar el modo de visualización para mostrar el resultado obtenido utilizando el algoritmo húngaro. Para ver la diferencia entre ambos resultados, a través del botón *Diferencia*, en una nueva ventana se puede observar con mayor detalle los píxeles que varían en ambos recortes de la información obtenida.

En esta ventana, también se puede utilizar zoom para poder observar las diferencias con detalle.

La información asociada al área de los píxeles seleccionados en cada búsqueda cuyo interior contiene la información requerida se puede consultar pulsando sobre el botón de información y en una nueva ventana aparecerá la información asociada a la búsqueda utilizando el algoritmo BIDM sin el algoritmo húngaro, y con el algoritmo húngaro.

Si el usuario desea ver la información detallada del alineamiento en otra imagen, a través de la tabla se pueden seleccionar todas las imágenes de test que se desee. Y por consiguiente, si el usuario desea realizar otra búsqueda en la misma imagen de referencia o desea utilizar otra imagen como referencia, en el paso 1 se puede cambiar el área de búsqueda o la imagen, respectivamente, y repetir el flujo de la aplicación.

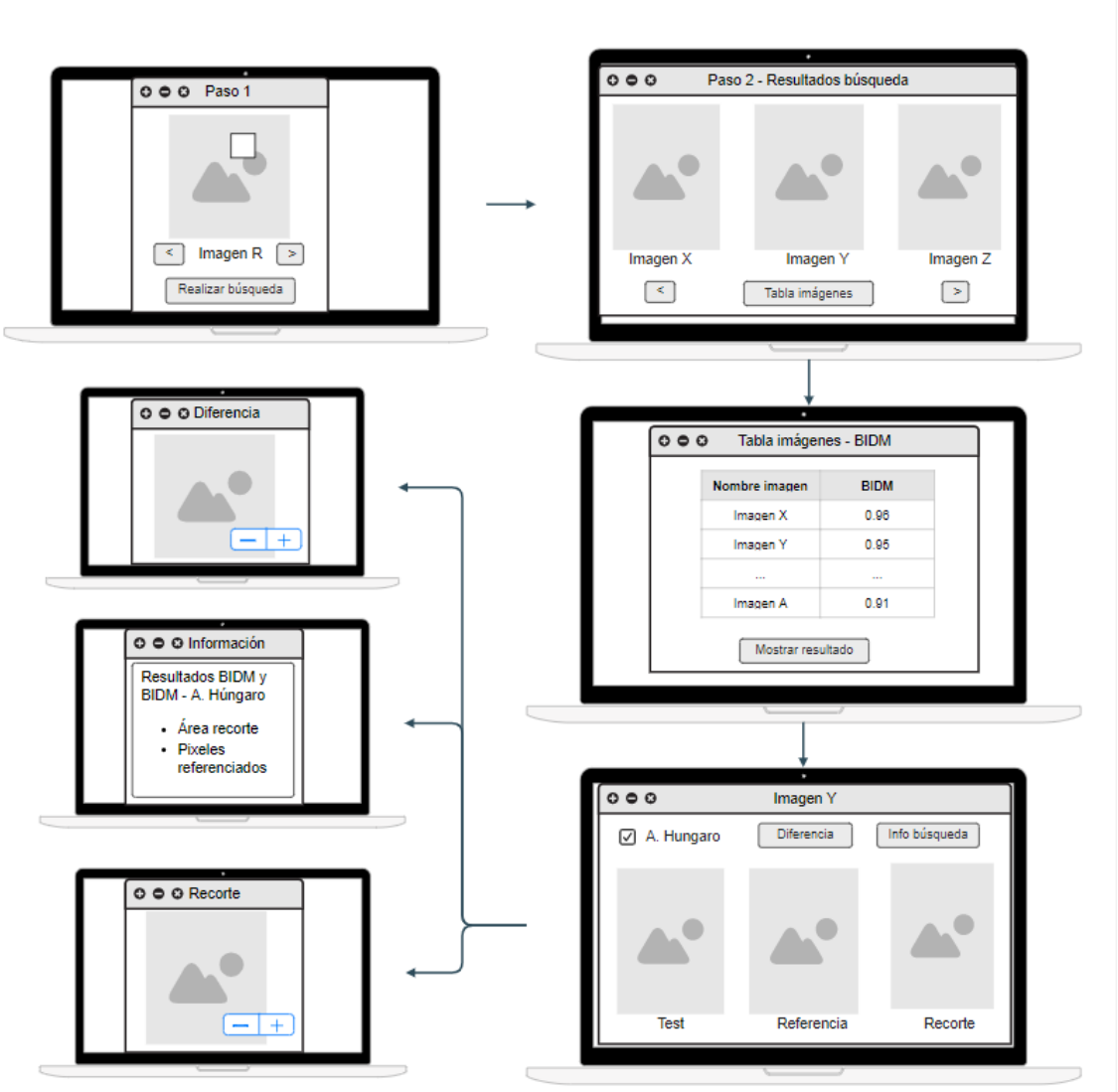


Figura 5.1: Diseño de la interfaz de la aplicación

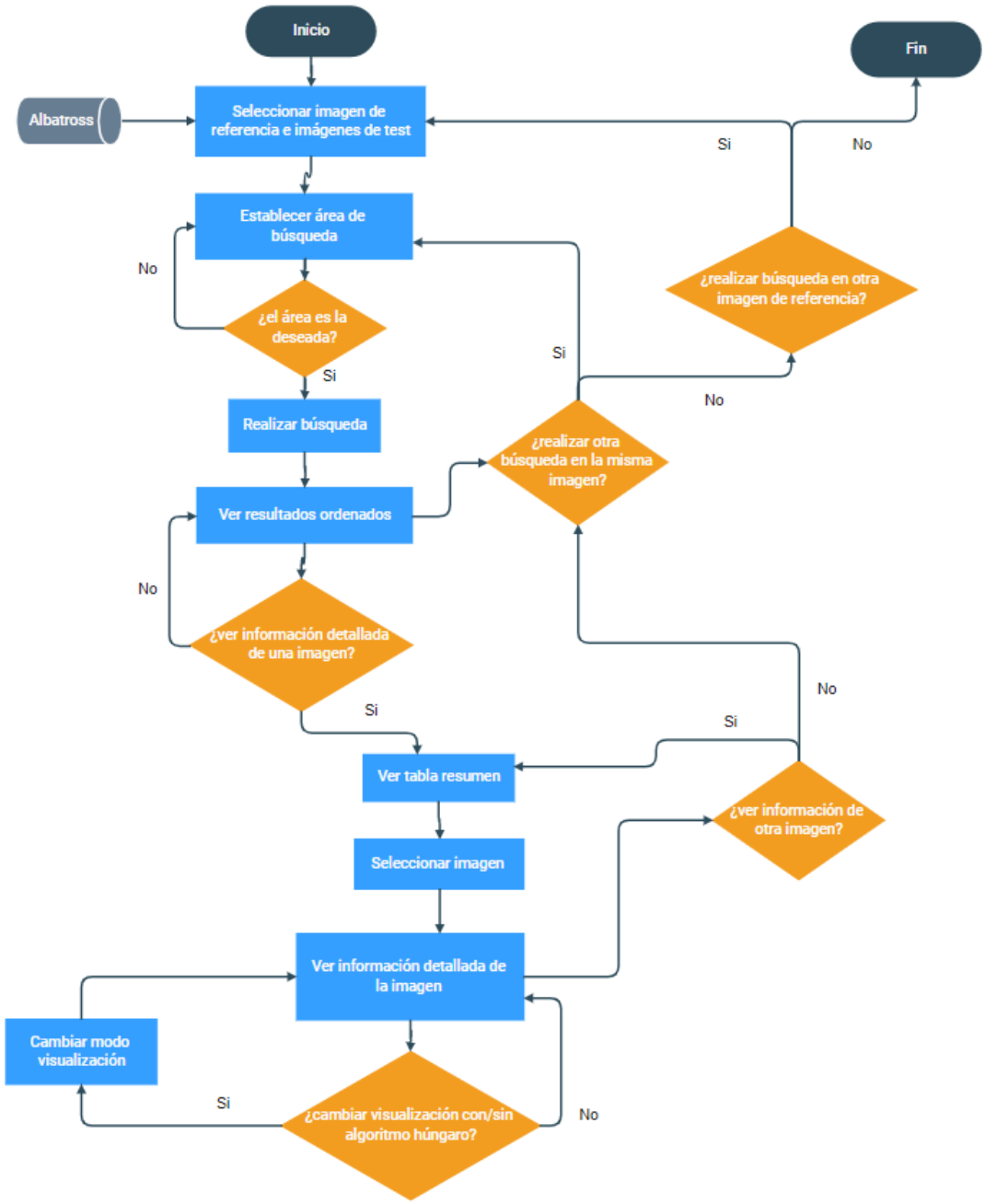


Figura 5.2: Diagrama de flujo de la aplicación

5.2 Tecnología utilizada

El alineamiento de las imágenes de documentos del corpus se va a realizar utilizando el algoritmo BIDM, explicado en el apartado 3.2. Además, se va a proceder a comparar los resultados obtenidos con la incorporación del algoritmo húngaro para optimizar el alineamiento general de todos los píxeles, limitando que cada píxel de test se alinee con un píxel de la imagen de referencia.

A modo de recordatorio, el algoritmo BIDM no es simétrico y alinea todos los píxeles de una imagen de test con los píxeles de una imagen de referencia. El alineamiento entre ambas imágenes y el procedimiento utilizado dado que la información a buscar se

encuentra seleccionada por el usuario en la imagen de referencia, se encuentra detallado en el siguiente capítulo.

Por otra parte, para limpiar las imágenes de documentos del corpus y eliminar el ruido para realizar un mejor alineamiento, se ha utilizado la herramienta *imgtxtenh* [53]. Dicha herramienta toma como entrada una imagen y genera como salida otra imagen del mismo tamaño con la imagen resultado limpia (véase 6.1).

Además, para el procesamiento de las imágenes para convertirlas al formato compatible con el código del algoritmo BIDM y otros procesamientos como el recorte de las áreas de búsqueda, se ha utilizado la librería ImageMagick [50]. Cabe destacar que para la implementación del algoritmo BIDM, se ha utilizado el código en lenguaje C realizado por los autores del mismo [3].

Por último, como ya se ha mencionado anteriormente en el análisis de la aplicación, para el desarrollo de la aplicación gráfica de este trabajo para visualizar los resultados de búsqueda, se ha utilizado Python como lenguaje de programación y la librería Tkinter de Python para llevar a cabo su desarrollo gráfico.

CAPÍTULO 6
Desarrollo de la solución propuesta

A continuación, en este capítulo se va a detallar la implementación realizada de la herramienta de alineamiento y búsqueda en imágenes propuesta en este trabajo. En primer lugar, se va a detallar el preprocesamiento de las imágenes de documentos que se ha debido realizar para poder obtener mejores resultados en el alineamiento. Y por último, se va a detallar la implementación de la aplicación gráfica.

6.1 Preprocesamiento de las imágenes de documentos

Las imágenes de documentos del corpus de Albatross se corresponden con imágenes escaneadas de distintas páginas pertenecientes a distintos cuadernos de bitácoras asociados al mismo barco. El algoritmo BIDM requiere como entrada dos imágenes en escala de grises para realizar el alineamiento, en concreto, en formato pgm [36]. Por tanto, se deben convertir las imágenes del conjunto a datos a dicho formato.

Para eliminar posibles ruidos y limpiar las imágenes para mejorar el alineamiento entre ellas, se ha utilizado la herramienta Imgtxtenh [53]. Esta herramienta aplica una máscara de limpieza en todas las imágenes del corpus según unos parámetros establecidos, ajustados al corpus. En la Figura 6.1, se pueden observar los resultados obtenidos para una imagen del conjunto de datos.

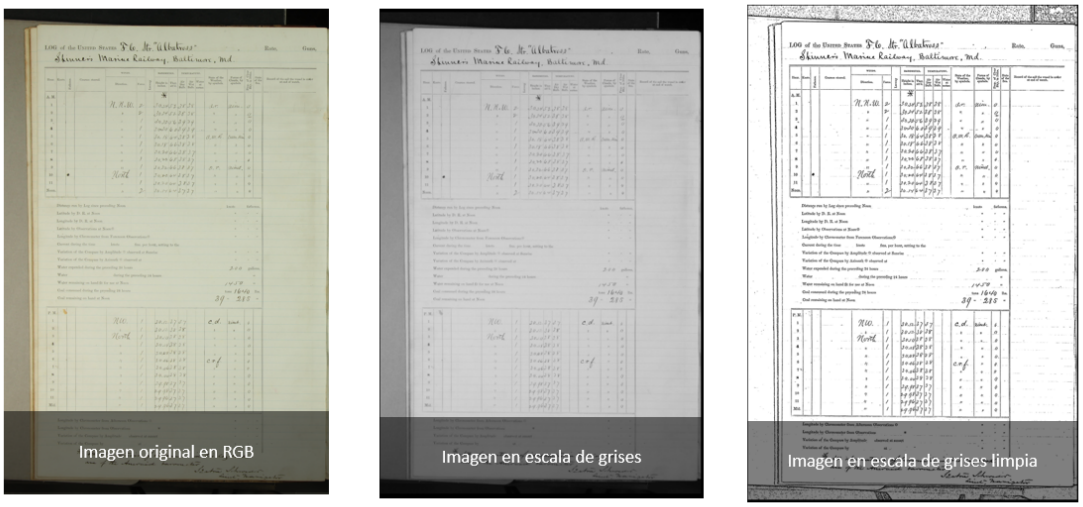


Figura 6.1: Preprocesamiento de las imágenes

Además, dado que las imágenes del corpus de Albatross presentan grandes dimensiones para la ejecución del algoritmo, se ha necesitado reducir sus dimensiones para minimizar el coste del algoritmo dado que es directamente proporcional al tamaño de las imágenes. Para ello, se ha utilizado la herramienta Imagemagick y se han redimensionado todas las imágenes al 20 % de su tamaño original, sin apreciar una pérdida de calidad significativa.

6.2 Alineamiento entre las imágenes de test y la imagen de referencia

Una vez las imágenes se han preprocesado y se encuentran limpias, se puede llevar a cabo, el alineamiento entre ellas y por consiguiente, la búsqueda de información que se pretende evaluar en este trabajo. Pero para ello, se deben especificar las imágenes de test y de referencia que se desean alinear.

Anteriormente, en el apartado del algoritmo BIDM 3.2, se ha detallado el funcionamiento del mismo. A modo de resumen, todos los píxeles de una imagen de test se alinean con un píxel de la imagen de referencia (varios píxeles de la imagen test pueden quedar relacionados con un mismo píxel de la imagen de referencia).

En la aplicación diseñada en el capítulo 5, se ha especificado que se debe elegir el área de búsqueda en la imagen de referencia. En la Figura 6.2, se puede observar el procedimiento a seguir para realizar el alineamiento de la imagen de test con la imagen de referencia, y la búsqueda del área establecida en la imagen de referencia en la imagen de test. El sentido de ambos procedimientos es el contrario.

Para realizar la búsqueda de un área determinada, para cada píxel que se encuentre dentro de dicha área, se debe obtener el píxel que ha sido alineado con un píxel de la imagen de test, si hubiese sido recuperado. En caso de que dicho píxel no haya sido alineado con ningún píxel de la imagen de test, se comprueba el alineamiento si lo existiera del siguiente. Una vez, se han recuperado todos los píxeles de la imagen de test que han sido alineados con un píxel dentro del área seleccionada, se devuelve dicho resultado. En concreto, el área resultante de la imagen de test se corresponde con el área de la búsqueda obtenida.

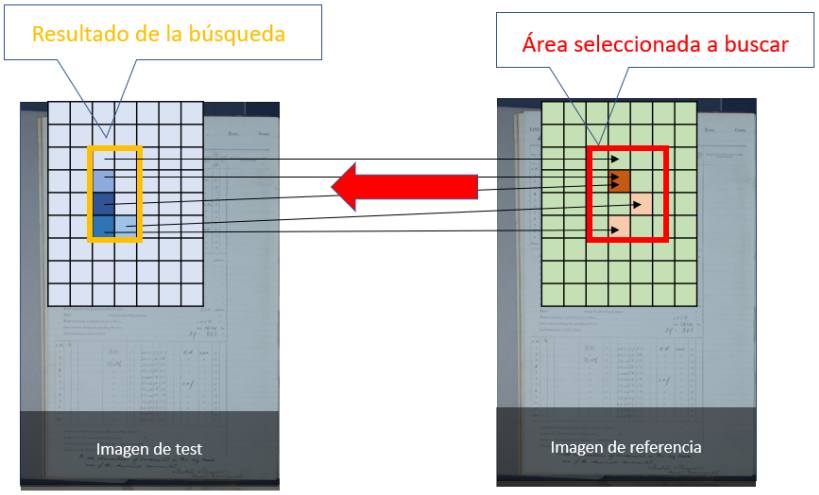


Figura 6.2: Alineamiento y búsqueda entre imagen de test y referencia

6.3 Búsqueda de la información seleccionada en la imagen de referencia

Para obtener el área de búsqueda asociada al alineamiento de los píxeles seleccionados en la imagen de referencia, en primer lugar, se deben determinar la imagen de referencia y las imágenes de test en las que se desea buscar. Dichas imágenes deben ser establecidas por el usuario en la configuración de la aplicación. Es decir, la aplicación puede comparar y buscar en todas las imágenes que el usuario decida y establezca como referencia y como test en su configuración.

En concreto, en este trabajo, dado que el corpus de datos está dividido en seis tipos diferentes de imágenes, cada tipo asociado a un volumen que contiene 12 imágenes, se va a tomar una imagen aleatoria como referencia y las once restantes de cada volumen, como imágenes de test de dicha imagen. Esta decisión se ha tomado para poder evaluar en el siguiente capítulo los resultados obtenidos, dado que no tiene sentido buscar información en dos imágenes con maquetaciones e información diferentes puesto que se entiende que ambas imágenes pueden contener distinta información.

Cabe destacar, que antes de establecer una imagen aleatoria de cada volumen como referencia, se ha explorado la posibilidad de crear una plantilla formada con la media de varias imágenes de cada volumen, respectivamente, y establecer dicha plantilla como la imagen de referencia asociada a cada cuaderno. Pero esta solución no fue factible dado que para ello, se debían alinear varias imágenes y los resultados obtenidos no fueron los esperados. Esto es debido a que las imágenes del conjunto de datos presentan varias variaciones de rotación y traslación en su obtención y por tanto, realizar un alineamiento perfecto entre ambas ha sido una tarea compleja que requería un excesivo trabajo para cumplir con el objetivo establecido de crear una plantilla.

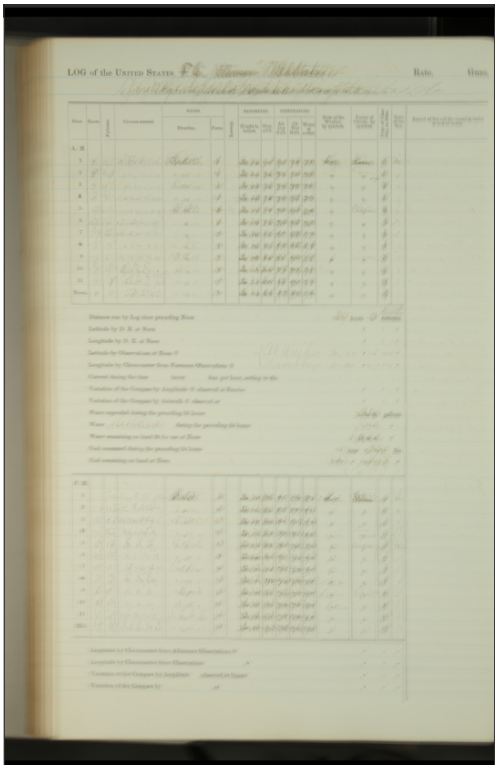


Figura 6.3: Plantilla volumen 009

En la Figura 6.3 se puede observar el resultado obtenido para un tipo de imágenes. Es probable que con la utilización de más imágenes para la creación de la plantilla, los resultados mejoren. Pero aún así, cabe destacar la complejidad de esta tarea debido a la variabilidad de obtención de las muestras.

6.4 Desarrollo de la aplicación gráfica

El desarrollo de la aplicación ha sido realizado siguiendo los prototipos diseñados en el capítulo anterior, en concreto, en los bocetos de la Figura 5.1. La implementación de esta herramienta, como ya se ha comentado anteriormente, se ha realizado en el lenguaje de Python y para realizar la interfaz gráfica, se ha utilizado su librería Tkinter. Para explicar el desarrollo de la misma, se va a seguir el mismo flujo presente en la figura del diseño, previamente citada.

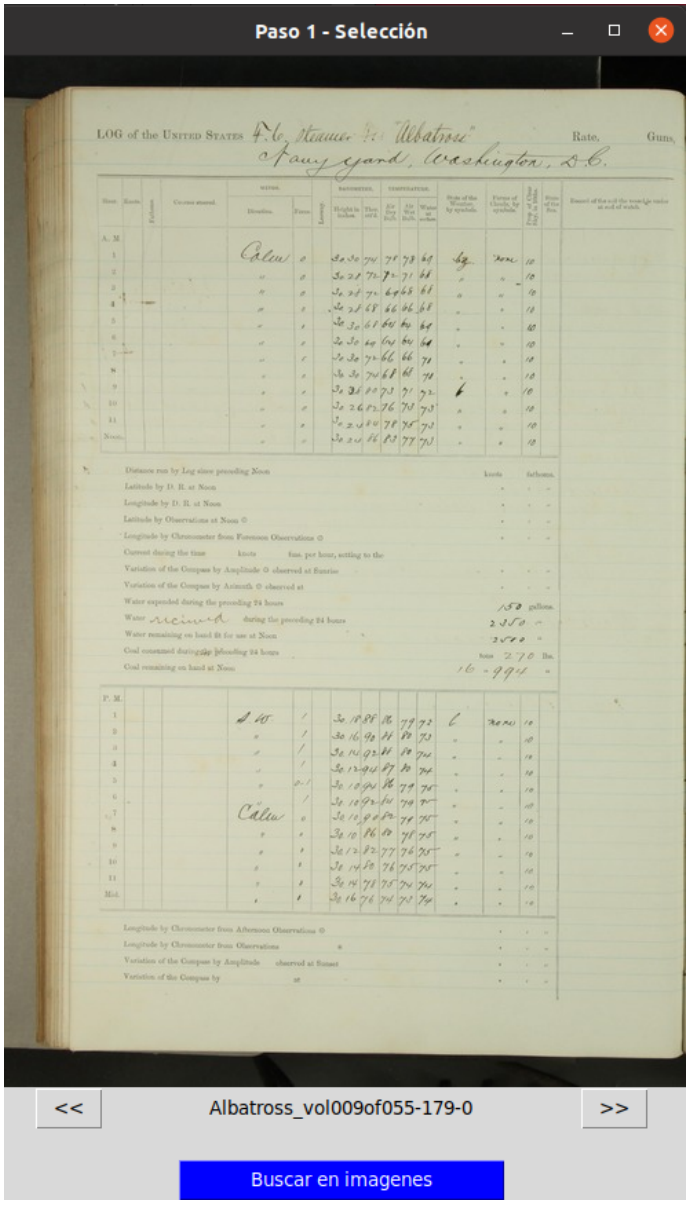


Figura 6.4: Paso 1 - Imagen de referencia

En primer lugar, en la ventana del paso 1, véase Figura 6.4, se pueden visualizar las diferentes imágenes de referencia que pueden ser seleccionadas por el usuario. Es decir, aparece una imagen de cada tipo de volumen presente en el corpus, que pueden ser visualizadas con el visor implementado. La imagen de referencia que se visualiza se corresponde con la imagen seleccionada. En el visor, utilizando los botones situados debajo de la imagen se pueden visualizar el resto de imágenes de referencia. Además, se muestra el identificador de la imagen seleccionada que está mostrando.

El usuario puede seleccionar el área de búsqueda de referencia en la imagen seleccionada utilizando el ratón y dibujando un rectángulo de selección. Dicho rectángulo se establece según los puntos seleccionados por el puntero por el usuario al clicar y desclicar sobre la imagen. Si el usuario, desea establecer otro área, puede volver a repetir el procedimiento anterior. Y una vez el área seleccionada sea la información a buscar que desea el usuario, a través del botón de búsqueda situado en la parte inferior de la ventana, se comienza el proceso de búsqueda.

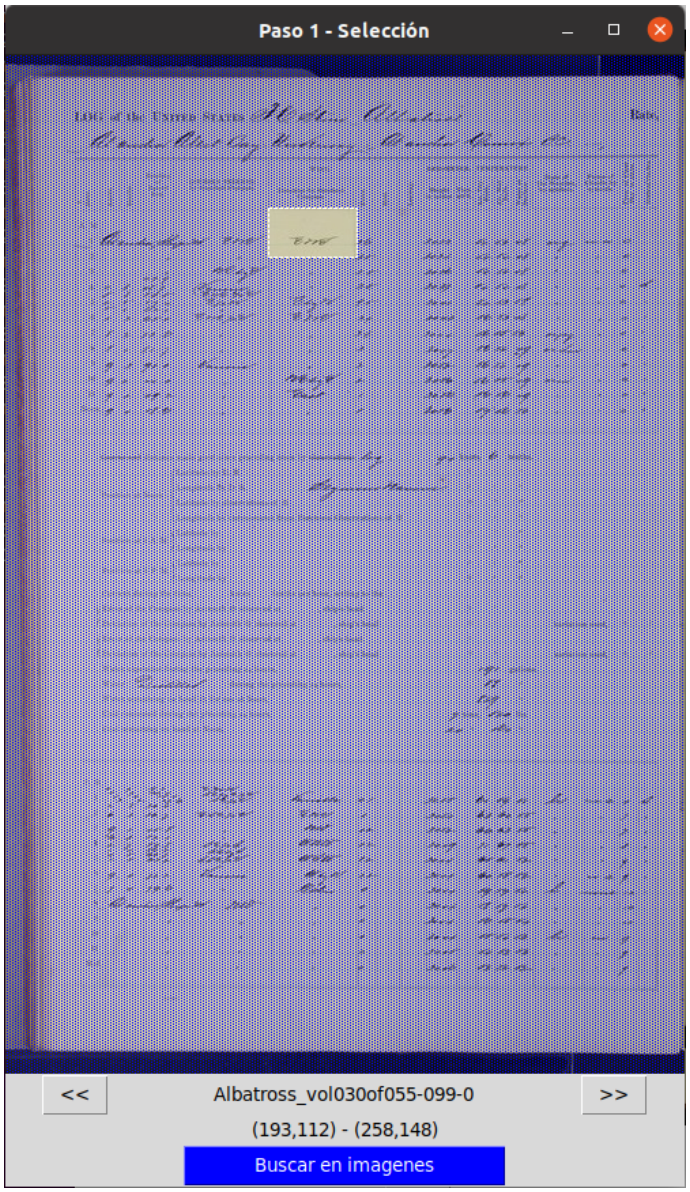


Figura 6.5: Paso 1 - Selección del área de búsqueda en imagen de referencia

En la Figura 6.5, se puede observar un área de búsqueda seleccionada. Se ha implementado que se destaque el área seleccionada de búsqueda y el resto de la imagen se muestre con un filtro para ayudar al usuario a visualizar el área que ha determinado.

Para realizar la búsqueda se alinean todas las imágenes de test del volumen seleccionado con la imagen de referencia asociada. Para ello, se debe ejecutar el código de los autores de la implementación del algoritmo con una pequeña modificación, que se ha realizado en este proyecto, para la obtención de la matriz de alineamiento y la relación del alineamiento de cada píxel. El alineamiento entre dos imágenes y la obtención de todas las matrices de alineamiento según el rango de distorsión y el contexto establecidos, con un coste temporal de $O\left(IJw^2c^2\right)$, tarda en torno a 15 minutos por alineamiento entre cada par de imágenes. [1] Este proceso se puede realizar antes de la ejecución de la aplicación y después acceder a la información del alineamiento en el uso de la herramienta.

Por tanto, para minimizar el tiempo de búsqueda de la aplicación, se ha realizado una búsqueda con unos parámetros por defecto establecidos y se ha guardado toda la información del alineamiento en distintos diccionarios. En concreto, con un rango de distorsión de 5 píxeles y una ventana de contexto local de 7 píxeles. Se han establecido estos parámetros dado que por la experiencia previa personal con el algoritmo BIDM, se conoce que dichos parámetros se ajustan al corpus de las imágenes utilizado realizando buenos alineamientos.

La información de los alineamientos guardada en los distintos diccionarios según las imágenes de test y referencia asociadas, se puede recuperar al comienzo del proceso de búsqueda de información y de esta forma, el tiempo de búsqueda de la aplicación sólo se corresponde con el procesamiento de dicha información almacenada.

En concreto, al principio del proceso de búsqueda, se recupera la información del alineamiento y para todos los píxeles del área seleccionada en la imagen de referencia, se recuperan los píxeles de la imagen de test (véase Figura 6.2) que habían sido alineados con un píxel del área de búsqueda y se establece el área del resultado. El área del resultado de la búsqueda se delimita con el área rectangular que contenga a todos los píxeles de test que hayan sido alineados.

Además, para el área de búsqueda seleccionada se ha calculado el alineamiento del algoritmo BIDM con la incorporación del algoritmo húngaro. Para ello, toda la información de las matrices de alineamiento de todos los píxeles que pueden ser alineados con el área seleccionada, se ha recopilado en la matriz del algoritmo húngaro, tal y como se detalló en la Figura 3.9. La implementación del algoritmo húngaro se ha realizado utilizado el el método *linear sum assignment* [30] de la librería *scipy.optimize*, el cual implementa este algoritmo.

Cuando se ha realizado la búsqueda y el alineamiento para todas las imágenes de test con respecto a la imagen de referencia utilizada, en el paso 2 se muestra una ventana con un visualizador de las áreas resultantes de las búsquedas obtenidas. Además, las búsquedas se encuentran ordenadas según la similitud entre las imágenes comparadas. Para realizar dicha ordenación se ha realizado el alineamiento entre dichas imágenes con BIDM y el valor resultante asociado, el cual se encuentra relacionado con la similitud entre ambas imágenes, se ha utilizado para determinar el orden.

El proceso de la búsqueda de información en las diversas imágenes de test tarda en torno a 15-25 minutos, según las dimensiones del área de búsqueda y del número de imágenes en las que se desea buscar. La complejidad temporal de la búsqueda es de $O\left(MX'Y'\right)$, siendo M el número de imágenes de test en las que se desea buscar, y X' e

[1] La máquina utilizada se corresponde con una máquina virtual de dos procesadores, 16 GB de memoria RAM y 60 GB de disco duro.

Y' las dimensiones del área de búsqueda en la imagen de referencia. Por tanto, debido a que el tamaño del área de búsqueda y el número de imágenes de test son directamente proporcionales con la complejidad de la búsqueda, si se aumentan las dimensiones de dicho área o el número de imágenes, el tiempo transcurrido para obtener los resultados aumenta proporcionalmente.

Como se puede observar en la Figura 6.6, en el visualizador implementado de tres imágenes de test se pueden observar las áreas obtenidas tras la búsqueda y marcados en rojo, los píxeles de test que se han alineado. Para ver con mayor detalle la información del alineamiento de una imagen dada, se debe seleccionar la imagen requerida a través de la tabla resumen.

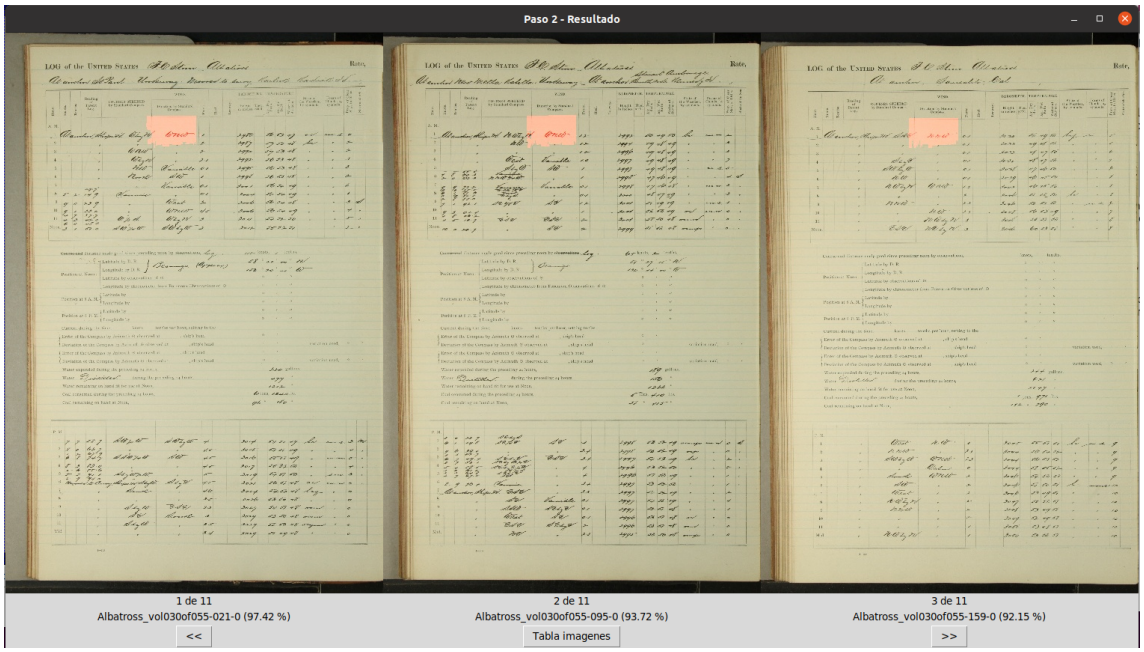


Figura 6.6: Paso 2 - Visualización de resultados obtenidos

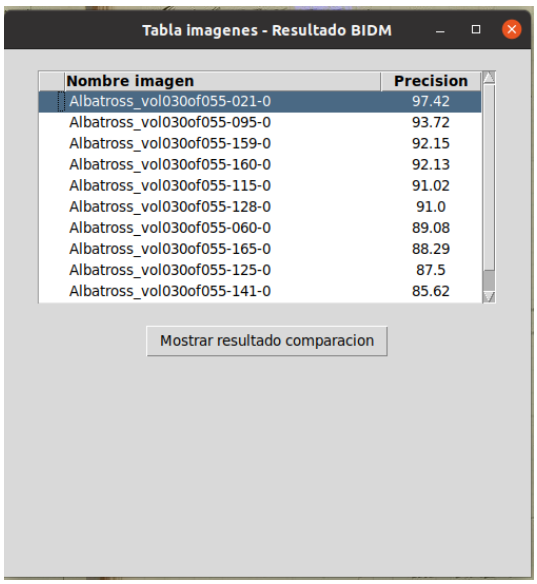


Figura 6.7: Tabla resumen de la ordenación de las imágenes

La ventana de la tabla resumen (véase Figura 6.7) contiene las imágenes de test ordenadas según similitud y si se desea ver con mayor detalle una imagen, basta con seleccionar la fila correspondiente y pulsar el botón asociado.

La información detallada de una imagen puede observarse en la ventana de la Figura 6.8. En dicha ventana aparecen tres imágenes en las cuales se pueden observar los píxeles de las imagen de test y de referencia que han sido alineados y el área de búsqueda resultante de dicho alineamiento.

En concreto, la imagen de la izquierda se corresponde con la imagen de test y presenta marcados en rojo los píxeles que han sido alineados con el área de búsqueda determinada por el usuario. En el medio, se muestra la imagen de referencia tomada como plantilla para seleccionar la búsqueda en el paso anterior y muestra marcados en azul los píxeles que han sido alineados con los píxeles de la imagen de test marcados en rojo.Y a la derecha, se puede visualizar la imagen de test con el área de búsqueda resultante marcada en el área rectangular seleccionada.

Si se desea observar el resultado obtenido con la incorporación del algoritmo húngaro, se puede modificar la visualización en la opción superior. Si se selecciona dicha visualización (véase Figura 6.9), la imagen de test de la izquierda muestra los píxeles alineados en rojo, la imagen de referencia, situada en el medio, muestra los píxeles referenciados en verde, y la imagen de test de la derecha actualiza su área de búsqueda resultante para dicha imagen según los píxeles alineados en la imagen de test para este caso.

Para poder ver el detalle el área de búsqueda resultante, la imagen de la derecha, la cual está asociada a la imagen de test y que muestra su área resultado seleccionada, tiene un evento asociado y tras clicar sobre dicha imagen, en una nueva ventana, se puede observar con mayor detalle el resultado obtenido. En dicha ventana se puede redimensionar y mover la imagen con el ratón para poder apreciar con detalle el resultado (véase Figura 6.12).

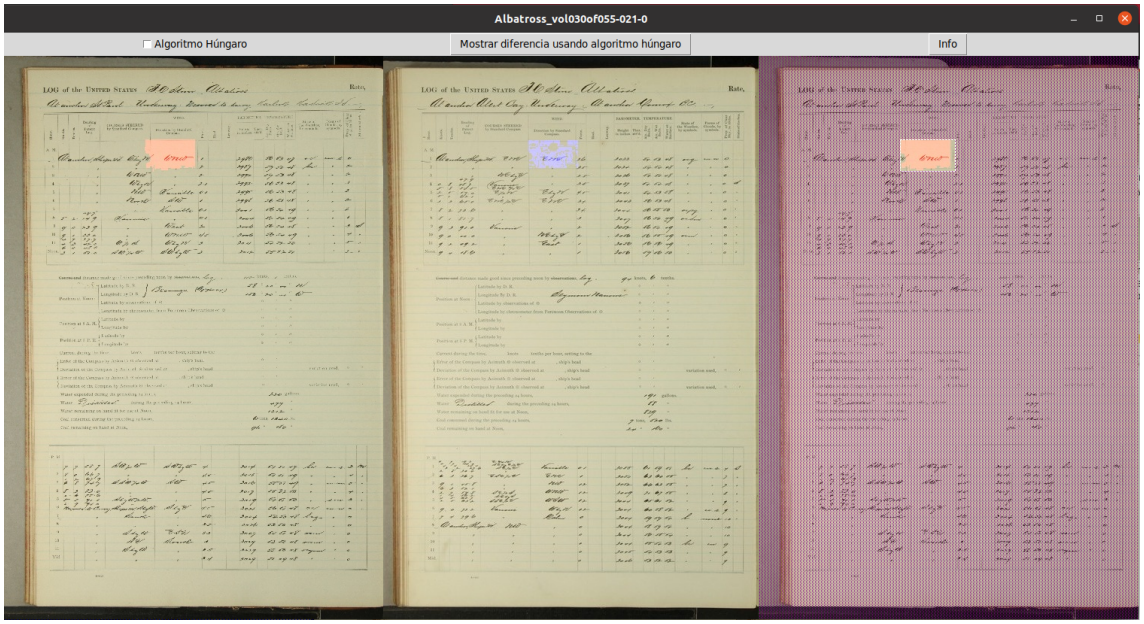


Figura 6.8: Detalle imagen con algoritmo BIDM

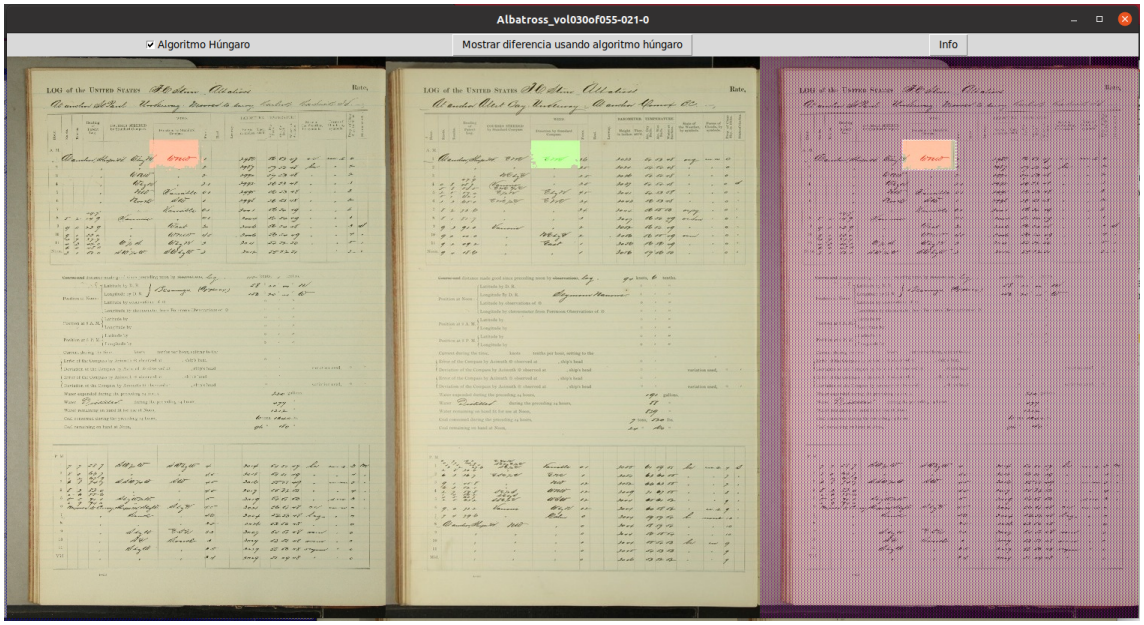


Figura 6.9: Detalle imagen con algoritmo BIDM y algoritmo húngaro

Para realizar la comparación entre ambos resultados obtenidos para una misma imagen, en una nueva ventana (véase Figura 6.11) pulsando el botón de Diferencia se puede observar los píxeles de la imagen de test alineados que han variado. En concreto, en dicha imagen se muestran en rojo los píxeles que se encuentran alineados en ambas implementaciones. En azul se encuentran seleccionados los píxeles que se han alineado con el algoritmo BIDM pero no se encuentran alineados con la incorporación del algoritmo de optimización. Y por ultimo, en verde se encuentran seleccionados los píxeles que se han alineado tras la incorporación del algoritmo húngaro y que se encontraban alineados en la anterior solución. Al igual que con la imagen del área resultante, esta imagen también se puede redimensionar y mover con el ratón para poder apreciar los diferentes píxeles y su color correspondiente.

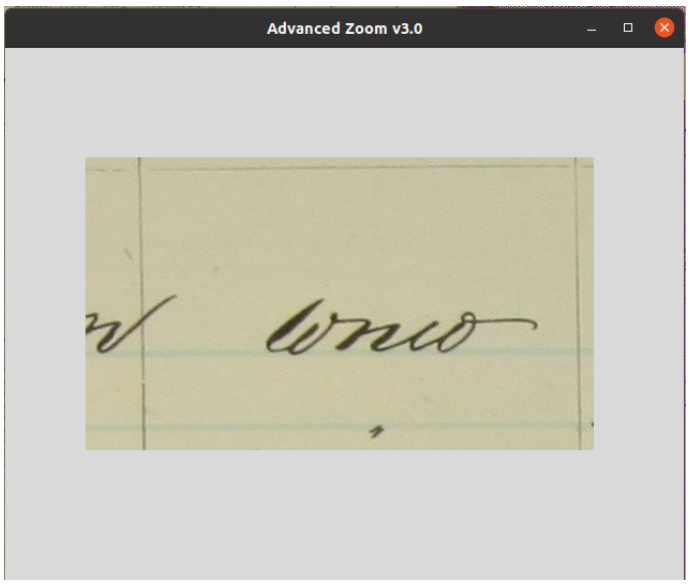


Figura 6.10: Área resultado de búsqueda

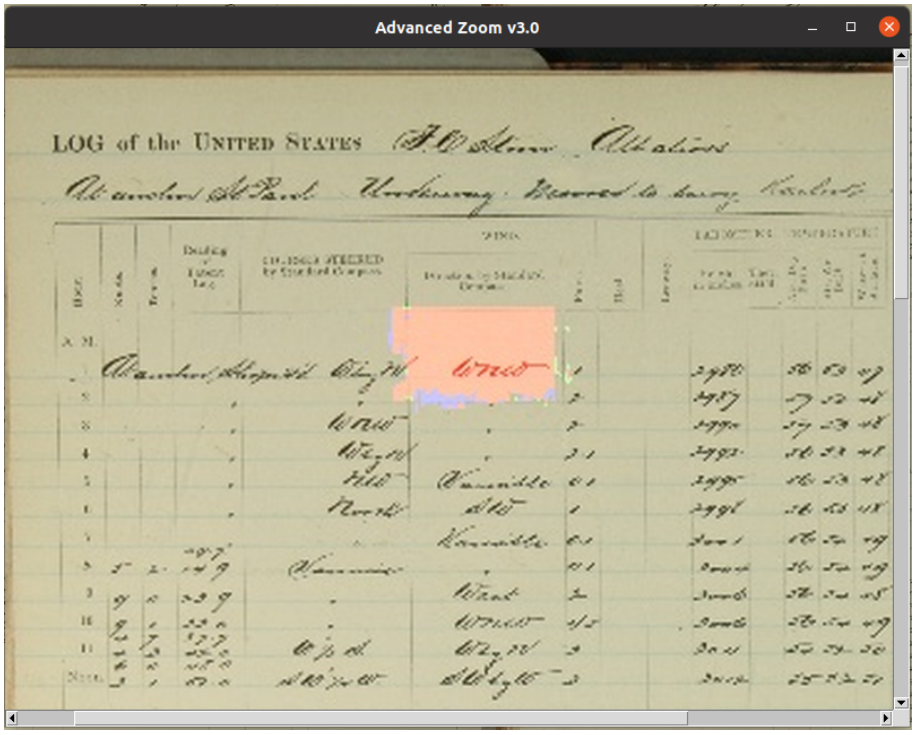


Figura 6.11: Imagen diferencia

Por último, en la ventana de información de la Figura 6.12 se encuentran descritas las dimensiones de ambas áreas resultantes y los píxeles de la imagen de referencia que han sido alineados.

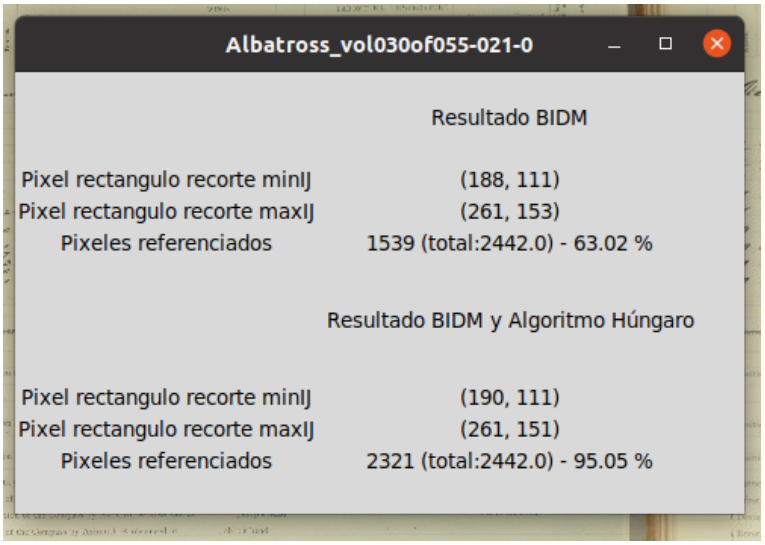


Figura 6.12: Información de la búsqueda

CAPÍTULO 7
Experimentos

En este capítulo, se va a proceder a evaluar la herramienta desarrollada en este trabajo. Para ello, en primer lugar se van a realizar diferentes pruebas de clasificación para comprobar si el algoritmo es capaz de distinguir entre las diferentes clases (plantillas) de documentos. Y en segundo lugar, se van a realizar diferentes experimentos de búsqueda de información para poder evaluar la tasa de acierto de las áreas de búsqueda obtenidas.

En lo siguientes experimentos, todas la imágenes utilizadas para la clasificación y búsqueda de información se han preprocesado siguiendo el procedimiento detallado en la sección 6.1.

7.1 Experimentos de clasificación

Los experimentos de clasificación se han realizado para evaluar si la técnica utilizada consigue distinguir los diferentes tipos de imágenes que existen en el corpus. El objetivo en una nueva versión de la herramienta es poder buscar y extraer la información de forma automática en las imágenes del mismo tipo que la imagen establecida como referencia. Por tanto, la herramienta debe ser capaz de filtrar las imágenes de la misma clase correctamente de forma automática utilizando la técnica de alineamiento como método de clasificación.

En los experimentos realizados se ha utilizado el algoritmo de evaluación BIDM como medida de disimilitud[1], y el algoritmo de los k vecinos más cercanos para poder comparar los resultados obtenidos. A continuación, en los siguientes apartados se detallan los resultados obtenidos para cada aproximación.

Los experimentos se han realizado utilizando validación cruzada de 12 particiones en ambas implementaciones. Para ello, se ha realizado la clasificación con las mismas muestras de entrenamiento o referencia, según el algoritmo, y con las mismas muestras de test para poder realizar una comparación entre los resultados obtenidos en cada partición.

En todas las particiones, se ha establecido una imagen de referencia de cada clase y el resto de imágenes se han establecido como imágenes de test. Es decir, en cada partición, se han determinado 6 imágenes de referencia y 66 imágenes de test. Además, las imágenes de referencia se han elegido de forma aleatoria en cada partición y como con-

[1] El algoritmo BIDM devuelve la proporción de píxeles correctos en el alineamiento de las imágenes comparadas. A mayor número de píxeles correctamente alineados, se entiende que ambas imágenes presentan mayor similitud entre ellas y por tanto, el valor obtenido en cada alineamiento se ha utilizado para realizar la clasificación. Aún así, queda pendiente en este trabajo estudiar las propiedades de dicha medida, la cual ha sido considerada como una disimilitud.

dición adicional, todas las imágenes han sido consideradas una vez como imágenes de referencia.

7.1.1. Experimentos de clasificación con BIDM

Los experimentos de clasificación con el algoritmo BIDM se han realizado comparando todas las imágenes de test con las imágenes de referencia y asignando cada imagen de test a la clase o volúmen de referencia que obtenga un valor resultante de la semejanza del algoritmo BIDM superior con respecto al resto.

La clasificación realizada depende de las imágenes de referencia establecidas y por tanto, para poder evaluar el comportamiento del algoritmo BIDM como clasificador, se ha utilizado la técnica de validación cruzada de 12 particiones, explicada anteriormente en este apartado.

Los resultados obtenidos junto con la matriz de confusión asociada a las diferentes comparaciones se pueden observar en la Tabla 7.1. En dicha tabla se puede observar la clasificación de todas las imágenes para cada clase con su correspondiente porcentaje de error y el porcentaje de error total que se corresponde con un **43.68 %**.

Cabe recordar que todas las imágenes tienen una maquetación parecida aunque tienen características diferentes en cada volumen. Además, para las imágenes de una misma clase, aunque la plantilla sea la misma hay variabilidad entre ellas. En concreto, no todas las imágenes de una misma clase presentan completada toda la información de la plantilla ni se encuentran completos los mismos datos en las mismas. Debido a esto, si dos imágenes de la misma clase se comparan y una celda concreta está rellenada en una imagen y vacía en la otra, el algoritmo distingue dicha diferencia y penaliza la similitud entre ambas. Por tanto, para realizar la clasificación, la variabilidad de la información escrita en las imágenes es importante para entender los resultados.

En la Tabla 7.1 se puede observar que en los resultados de la clasificación varían según las clases. En concreto, para las imágenes del volumen 030, el algoritmo extrae las características de dicha clase y se obtienen buenos resultados con una tasa de error de 0.0 %. Pero la clase del volumen 042 ha obtenido una tasa de acierto significativamente baja. Esto es debido a que, observando el corpus, se ha detectado que el volumen 042 y el volumen 038 presentan la misma maquetación con una pequeña modificación de escritura en la cabecera de la misma.

Si se observa la matriz de confusión de la clase 042, se puede observar que la gran mayoría de imágenes del cuaderno 042 se han clasificado con el volumen 038. Por tanto, a pesar de haber obtenido una tasa de error alta, se puede determinar que el clasificador si que ha reconocido la maquetación general pero no ha clasificado correctamente la variación del título. Aún así, dado que se ha producido un error al clasificar una imagen en su clase, las asignaciones de las imágenes del volumen 042 con las imágenes del volumen 038 se han evaluado como errores.

Además, en la clasificación de las imágenes de los volúmenes 009 y 049, se puede observar que un alto número de imágenes de ambas clases se clasifican con el volumen 042. Por tanto, se puede afirmar que el algoritmo no consigue realizar una correcta extracción de las características del volumen 042.

Por otra parte, en los resultados obtenidos para la clasificación de los volúmenes 009, 049 y 055, en las clasificaciones erróneas se ha observado que las imágenes mal clasificadas presentaban una variabilidad de información completada parecida entre ambas, y por tanto, no se ha distinguido la maquetación, sino que la clasificación se ha realizado según la información presente en cada imagen.

En conclusión, se puede determinar que el algoritmo realiza una buena clasificación y obtención de características del volumen 030 y una clasificación aceptable para los volúmenes 009 y 055. Sin embargo, para el resto de volúmenes se puede observar una matriz más dispersa y que no realiza una correcta extracción de las características en todos los casos para dichas imágenes.

A pesar de ello, de forma general la tasa de error es menor al 50 % y se puede considerar que el algoritmo BIDM obtiene resultados aceptables como clasificador. En el siguiente apartado, se detallan los resultados obtenidos con el algoritmo del vecino más cercano para poder comparar ambos resultados.

	Clase obtenida con BIDM						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	77	1	0	49	3	3	41.66 %
vol030	0	132	0	0	0	0	0.00 %
vol038	4	0	55	44	16	13	58.33 %
vol042	15	4	60	46	7	0	65.15 %
vol049	23	4	15	34	56	0	57.57 %
vol055	22	6	17	6	1	80	39.39 %
							43.68 %

Tabla 7.1: Matriz de confusión en clasificación con BIDM

7.1.2. K-Vecinos más cercanos

El algoritmo de los K-vecinos [26] más cercanos se ha utilizado para poder comparar los resultados obtenidos anteriormente con BIDM. En concreto para poder replicar y comparar la clasificación del apartado anterior, las muestras tomadas como referencia, en este caso, se convierten en las muestras de entrenamiento y las muestras de test se mantienen. Además, el problema de clasificación se debe implementar con la obtención del vecino más cercano. Es decir, con K=1, puesto que solo se cuenta con una imagen de referencia por clase.

La implementación de este experimento se ha realizado utilizando las librería que implementa el algoritmo de los K vecinos más cercanos de scikit-learn [35]. La experimentación de este algoritmo se ha realizado con y sin utilizar la técnica de PCA para evaluar su comportamiento y poder comparar los resultados obtenidos con la técnica de alineamiento anterior.

En concreto, en el algoritmo del vecino más cercanos, se ha utilizado PCA para evaluar si dicha técnica es capaz de extraer las características y optimizar la clasificación. Los resultados obtenidos pueden observarse en las Tablas 7.2 y 7.3.

	Clase obtenida con el vecino más cercano						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	132	0	0	0	0	0	0.00 %
vol030	0	132	0	0	0	0	0.00 %
vol038	0	0	132	0	0	0	0.00 %
vol042	0	0	0	123	4	5	6.81 %
vol049	0	0	0	10	103	19	21.96 %
vol055	0	0	0	1	10	121	8.33 %
							6.18 %

Tabla 7.2: Resultados clasificación con 1-Vecino más cercano

	Clasificación con PCA						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	132	0	0	0	0	0	0.00 %
vol030	0	132	0	0	0	0	0.00 %
vol038	0	0	132	0	0	0	0.00 %
vol042	0	0	0	129	3	0	2.27 %
vol049	0	0	0	27	103	2	21.96 %
vol055	0	0	0	0	0	132	0.00 %
							4.03 %

Tabla 7.3: Resultados clasificación con 1-Vecino más cercano y PCA (5 dimensiones)

En las tablas de los resultados obtenidos se puede observar que se obtiene una mejor clasificación utilizando PCA. En concreto, la tasa de error disminuye de un 6.18 % a un 4.03 %. En ambas matrices de confusión se puede observar que las imágenes de los volúmenes 009, 030 y 038 se clasifican correctamente en todos los casos y las imágenes de la clase 049 presentan una distribución de clasificación similar en ambos casos. Por otra parte, las imágenes de las clases 042 y 055 mejoran su clasificación utilizando PCA.

Por tanto, se puede afirmar que la utilización de PCA ayuda a extraer las características de cada imagen y ayudan a realizar la clasificación. Para evaluar el número de dimensiones a las que se debía reducir la imagen para obtener buenos resultados se ha analizado el comportamiento del problema con diferentes dimensiones. En el anexo A se pueden observar los resultados obtenidos con las diferentes dimensiones utilizadas. Dichos resultados se pueden observar en la gráfica de la Figura 7.1.

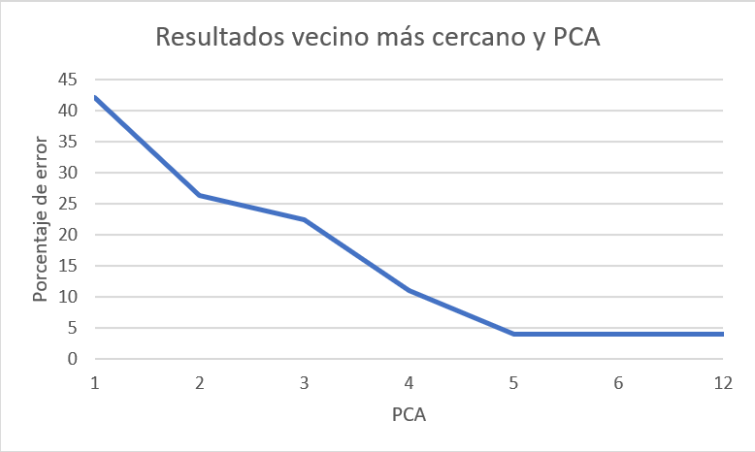


Figura 7.1: Gráfica resultados utilizando PCA

En dicha gráfica se puede observar que con la reducción de la imagen a 5 se alcanza el porcentaje de error mínimo de 4.03 %. Si se aumenta el número de dimensiones, el porcentaje de error no varía y las matrices de confusión obtenidas son iguales. Debido a esto, se ha establecido la reducción de dimensiones con PCA a 5 dimensiones para realizar las comparaciones entre ambos modelos.

En conclusión, se han obtenido mejores resultados utilizando el algoritmo del vecino más cercano y principalmente, con la utilización de PCA a 5 dimensiones. El porcentaje de error obtenido con el algoritmo BIDM es significativamente superior y si se comparan ambas matrices de confusión, se puede determinar que la clasificación con la técnica de alineamiento obtiene peores resultados para todas las clases, a excepción del volumen 030.

El algoritmo BIDM se puede considerar como una técnica de clasificación aceptable aunque al existir otras técnicas que obtienen un comportamiento mejor para este problema, se debe considerar cualquier mejora o preprocesamiento de las imágenes para optimizar los resultados con el alineamiento no lineal entre dos imágenes.

7.2 Experimentos de búsqueda de información

Para poder evaluar la búsqueda en imágenes de documentos y en concreto, las áreas obtenidas de búsqueda de información establecida en una imagen de referencia en la aplicación, se ha realizado el siguiente proceso.

Para cada volumen, que contiene 12 imágenes, se ha establecido una imagen como referencia y las once restantes se han determinado como test. Para evaluar las búsquedas, se han seleccionado 20 áreas de búsqueda en la imagen de referencia y se ha etiquetado como resultado válido o inválido las 11 áreas resultantes para cada búsqueda (véase 7.3). Es decir, para un volumen dado se han establecido 20 áreas de búsqueda y se han evaluado las 220 áreas obtenidas en las imágenes de test.

Cabe destacar que dichas áreas de búsqueda se han definido eligiendo diversas áreas de búsqueda en la imagen con dimensiones variables y que contengan diferente información. En concreto, se ha realizado la búsqueda de casillas concretas, de filas completas, de columnas, de una tabla completa, de información manuscrita en la cabecera, etc. Para cada imagen de referencia se han establecido 20 áreas de búsqueda y se han etiquetado los resultados obtenidos. La evaluación se ha realizado por inspección manual puesto

que en el momento de realizar este trabajo dicha evaluación comportaba otros problemas que se ha pretendido evitar, como la definición de una buena medida automática.

Para poder etiquetar un área de búsqueda resultado como válida, el contenido a buscar debe encontrarse en dicho recorte y poder leerse completamente. Por ejemplo, en la Figura 7.2, se puede observar un ejemplo de dicho procedimiento. Para un área de búsqueda dada, en dicho caso, una fila que debe contener los 5 valores asociados, en el área resultado deben aparecer dicha información. En la caso establecido como resultado correcto y válido, dicha información se encuentra legible, pero por ejemplo en el caso de resultado incompleto o no válido, el primer valor de la fila (30.24) y en concreto, la primera cifra (3) se encuentra cortada. Dicho resultado no es un resultado correcto dado que no se puede encuentra toda la información que se desea buscar de forma completa y en un proceso posterior, puede no reconocerse correctamente.

En la Figura 7.3, puede observarse un experimento de los veinte realizados sobre el volumen 009 y, siguiendo el procedimiento de evaluación explicado anteriormente, se puede observar que en 10 imágenes se encuentra la información de búsqueda totalmente completa y una imagen que se encuentra ligeramente cortada en su primer elemento. Por tanto, para dicho experimento, se han obtenido 10 aciertos, 1 fallo y una tasa de acierto del 90.9 %.

Los resultados obtenidos para todos los volúmenes y tras realizar todos los experimentos, se pueden observar en la Tabla 7.4. Se puede observar que la tasa de error media para todos los volúmenes es inferior al 10 %, en concreto, 8.03 %. Debido a esto, se puede determinar que la técnica de alineamiento utilizada en este trabajo obtiene buenos resultados.

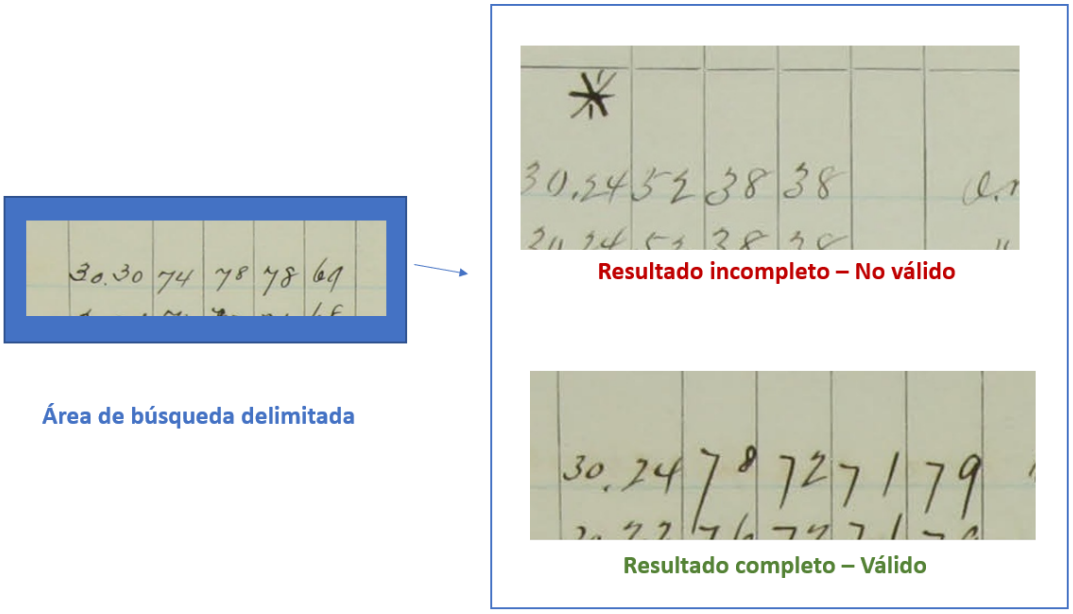


Figura 7.2: Evaluación búsqueda ejemplo

Figura 7.3: Experimento 1 - volumen 009

Resultados búsqueda	
Clase	Porcentaje de error
Clase vol009	6.36 %
Clase vol030	3.63 %
Clase vol038	7.72 %
Clase vol042	9.54 %
Clase vol049	10.90 %
Clase vol055	10.0 %
	8.03 %

Tabla 7.4: Resultados búsqueda de información

Cabe destacar que en dicha tabla se encuentran los porcentajes de acierto y de error utilizando BIDM y el algoritmo húngaro, dado que se han obtenido los mismos resultados. Esto es debido a que la incorporación del algoritmo húngaro, según el caso, modificaba el área de búsqueda pero los cambios entre ambas áreas no eran muy significativos y por tanto, los resultados no variaban.

En la Figura 7.4 pueden observarse dos ejemplos distintos de áreas de búsqueda obtenidas con el algoritmo BIDM y con el algoritmo húngaro incorporado. Para ambos casos, los resultados se evalúan igual pero si que se puede observar que con el algoritmo húngaro se afina la búsqueda a la información requerida. En el primer caso, se pretendía buscar la primera fila únicamente y por tanto, con el algoritmo húngaro, el área de búsqueda se centra en el área objetivo.

En el segundo caso, el área de búsqueda era la primera fila y los dos valores que se muestran. En ambas áreas resultado se muestra la información a buscar, pero en el recorte del algoritmo húngaro la información se encuentra centrada y puede visualizarse mejor.

Por tanto, en resumen, se puede determinar que la técnica de alineamiento utilizada se puede utilizar para realizar búsquedas en imágenes de documentos debido a su alta tasa de acierto. Y con respecto, a la incorporación del algoritmo húngaro, aunque los resultados obtenidos en cuanto a tasa de error y acierto sean iguales, las áreas resultantes obtenidas son ligeramente mejores.

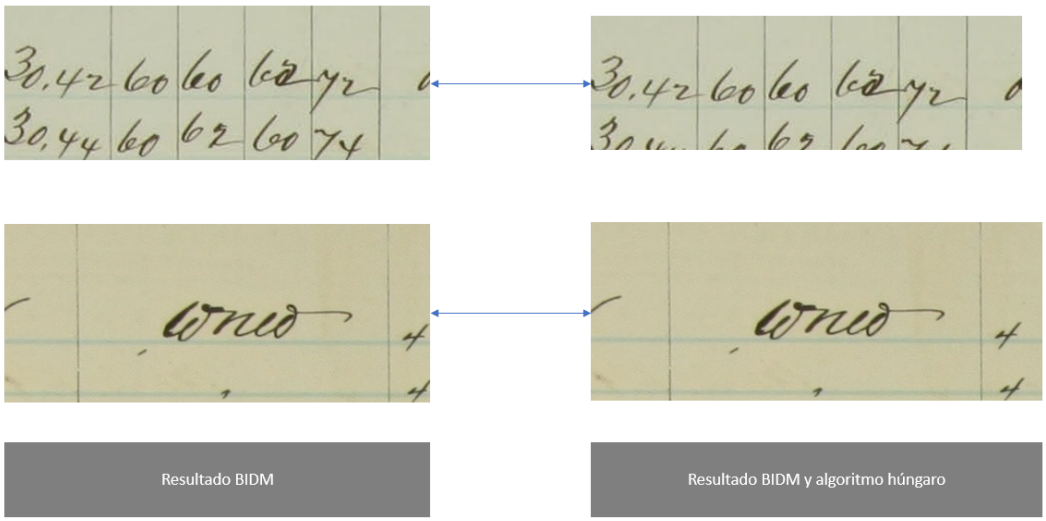


Figura 7.4: Resultados de búsqueda obtenidos

CAPÍTULO 8
Conclusiones

En este trabajo, se ha analizado y desarrollado una aplicación para la comparación y búsqueda en diferentes imágenes de documentos. En concreto, se ha analizado el comportamiento del algoritmo de alineamiento no lineal BIDM y se ha desarrollado una aplicación gráfica para poder observar los resultados de búsqueda obtenidos. Dado que el objetivo principal de este proyecto era desarrollar una herramienta capaz de alinear distintas imágenes para realizar búsquedas, se puede afirmar que dicho objetivo se ha alcanzado.

Para ello, en primer lugar, se ha recopilado el corpus de imágenes de documentos de Albatross, el cual contiene imágenes que presentasen una maquetación parecida para poder realizar un alineamiento.

En segundo lugar, se ha analizado el comportamiento del algoritmo BIDM con respecto al conjunto de imágenes utilizado. Y, para poder observar los resultados de clasificación y búsqueda establecidos se ha desarrollado una aplicación gráfica para poder buscar en las imágenes que el usuario decida y poder visualizar las áreas de búsqueda que contienen la información que se desea buscar.

Además, para poder visualizar los resultados de forma ordenada según los criterios de la búsqueda se ha utilizado el mismo algoritmo como clasificación, para poder visualizar las imágenes de mayor a menor similitud con la imagen de referencia.

Por último, en los experimentos realizados se ha evaluado la técnica de alineamiento utilizada en el corpus mediante experimentos de clasificación y de búsqueda de información, respectivamente. En los experimentos de clasificación también se ha utilizado el algoritmo de los k-vecinos más cercanos, en nuestro caso, el vecino más cercano para poder replicar los experimentos del caso anterior y poder comparar ambos resultados.

En los resultados obtenidos en los experimentos de clasificación, se ha obtenido una tasa de error aceptable y en algunas clases se puede comprender el mal funcionamiento del mismo. Por tanto, dado que se ha comprobado que existen técnicas que obtienen mejores resultados para el mismo conjunto de imágenes dado, en los trabajos futuros del siguiente apartado se debe plantear alguna mejora en este ámbito para mejorar el clasificador.

Por otra parte, los resultados obtenidos en los experimentos de búsqueda de información han obtenido una tasa de error significativamente baja y por tanto, se puede establecer la técnica utilizada como una herramienta para poder automatizar y extraer las áreas de información que se desean obtener.

Cabe recordar que el objetivo de búsqueda en imágenes se encuentra ligado al problema de la automatización de información para la entrada de sistemas de reconocimiento ya existentes. Por tanto, dado que la técnica de alineamiento utilizada obtiene buenos

resultados en la mayoría de los casos, se pueden utilizar las áreas de búsqueda obtenidas como imágenes de entrada en los diferentes modelos de reconocimiento ya existentes.

Además, en el desarrollo de la herramienta de este proyecto se ha evaluado la incorporación de un algoritmo de optimización, el algoritmo húngaro. Con dicho algoritmo se pretendía mejorar las áreas de búsqueda resultantes en las imágenes de test. Pero en los experimentos se han observado que los resultados obtenidos no variaban. Las áreas de búsqueda resultantes eran iguales o presentaban pequeñas variaciones, y por tanto, ambas imágenes se etiquetaban con el mismo valor de válido o inválido. Debido a que no se presentan variaciones significativas entre ambas implementaciones y que la incorporación del algoritmo húngaro, supone un coste computacional añadido, se puede determinar que el algoritmo BIDM obtiene buenos resultados sin necesidad de aplicar otro algoritmo de optimización.

Asimismo, se puede concluir, que la técnica de alineamiento utilizada para la comparación y la búsqueda en diferentes imágenes de documentos ha obtenido buenos resultados. Y, debido a esto, la herramienta desarrollada puede ser utilizada para ayudar a usuarios expertos a optimizar búsquedas en imágenes con una maquetación similar.

CAPÍTULO 9
Trabajos futuros

En trabajos futuros, se plantean desarrollar diversas mejoras de la aplicación desarrollada y del procesamiento y alineamiento de las imágenes para intentar obtener mejores resultados.

En primer lugar, una de las mejoras principales que se plantean es que la aplicación pueda ser usada por cualquier usuario y en cualquier dominio. Para el desarrollo de esta herramienta, la implementación se ha centrado en el corpus establecido y tal y como se ha comentado el usuario debe ser un usuario experto.

Por otra parte, dado que la clasificación obtenida con el algoritmo ha obtenido un resultado peor que con otras técnicas, como futura mejora se plantea analizar la incorporación de otro preprocesamiento para intentar mejorar los resultados. Y por consiguiente, en futuras versiones se plantea poder incorporar la funcionalidad de que la herramienta pueda filtrar las imágenes del corpus que se correspondan con el mismo tipo que el tipo de la imagen que se ha tomado como referencia y así, poder automatizar la búsqueda de la información en las imágenes de test.

Además, con respecto a la búsqueda de información en una imagen, en futuras versiones de la aplicación, se plantea mejorar dicha funcionalidad. En concreto, se plantea establecer varias áreas de búsqueda en una imagen, en vez de una única área. Es decir, para una imagen de referencia, se plantea establecer dos áreas de búsqueda o más. Y se plantea permitir al usuario que decida si quiere visualizar en el resto de imágenes la información que se encuentra entre ambas áreas, inclusives o no, según la preferencia del usuario. De esta forma, se pretende optimizar con varios puntos de referencia el proceso de búsqueda.

También, otra mejora sería la incorporación de un modelo de reconocimiento de la información a la aplicación para dadas las áreas de búsqueda obtenidas poder reconocer la información que contienen y realizar informes o estadísticas.

Otro aspecto a tener en cuenta es el coste temporal y computacional de los procesos de búsqueda, y por tanto, otra tarea a resolver se correspondería con la optimización de tiempos para reducir el coste lo máximo posible y dotar de rapidez a la herramienta.

En conclusión, las mejoras detallada anteriormente se corresponderían con mejoras en la aplicación desarrollada para dotarla de flexibilidad en cualquier dominio.

Bibliografía

[1] Agarwal, M., Mondal, A. y Jawahar, C. (2020) Cdec-net: Composite deformable cascade network for table detection in document images. arXiv:2008.10831, 2020.

[2] Algoritmo húngaro. (2021). Wikipedia, Disponible en `https://es.wikipedia.org/wiki/Algoritmo_húngaro`

[3] Álvaro, F., Sanchez, J.A. y Benedí, J.M. (2013) An image-based measure for evaluation of mathematical expression recognition In *Iberian Conference on Pattern Recognition and Image Analysis*.Springer, Berlin, Heidelberg, 2013. p.682-690.

[4] Casado-García, Á., Domínguez, C., Heras, J., Mata, E., y Pascual, V. (2020) The benefits of close-domain fine-tuning for table detection in document images. in Int. Workshop Document Anal. Sys. Springer, 2020, pp. 199–215.

[5] Cesarini, F., Marinai, S., Sarti, L. , y Soda, G. (2002) Trainable table location in document images. in Object Recognit. supported user interaction service robots, vol. 3, 2002, pp. 236–240.

[6] Chakraborty, V. y Chakraborty, A. (2022) Automatic Document Scanner using OpenCV. Disponible en `https://learnopencv.com/automatic-document-scanner-using-opencv/?ck_subscriber_id=825227789`

[7] Chen, H.-H., Tsai, S.-C., y Tsai, J.-H. (2000) Mining tables from large scale html texts. in COLING Volume 1: The 18th Int. Conf. Comput. Linguistics, 2000.

[8] Deng, Y., Rosenberg, D. y Mann, G. (2019) Challenges in end-to-end neural scientific table recognition. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 894–901.

[9] 16th International Conference on Document Analysis and Recognition ICDAR 2021. (2021). Disponible en `https://icdar2021.org/`

[10] Fan, M. y Kim, D. S. (2015) Detecting table region in pdf documents using distant supervision. arXiv:1506.08891, 2015.

[11] Gilani, A., Qasim, S. R., Malik, I., y Shafait, F. (2017) Table detection using deep learning. in 14th IAPR Int. Conf. document Anal. Recognit. (ICDAR), vol. 1, 2017, pp. 771–776.

[12] Hashmi, K. A., Liwicki, M., Stricker, D., Afzal, M. A., Afzal, M. A. y Afzal, M. Z. (2021) Current Status and Performance Analysis of Table Recognition in Document Images With Deep Neural Networks. In IEEE Access, vol. 9, pp. 87663-87685, 2021, doi: 10.1109/ACCESS.2021.3087865.

[13] Hashmi, K. A., Stricker, D., Liwicki, M., Afzal, M. N. y Afzal, M. Z. (2021) Guided table structure recognition through anchor optimization. arXiv:2104.10538, 2021.

[14] Holecek, M., Hoskovec, A., Baudiš, P. y Klinger, P. (2019) Table understanding in structured documents. in Int. Conf. Document Anal. Recognit. Workshops (ICDARW), vol. 5, 2019, pp. 158–164.

[15] Huang, Y., Yan, Q., Li, Y., Chen, Y., Wang, X., Gao, L. y Tang, Z. (2019) A yolobased table detection method. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 813–818.

[16] Jaume, G., Ekenel, H. K., Thiran, J. (2019) FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. Accepted to ICDAR-OST. Disponible en `https://guillaumejaume.github.io/FUNSD/`

[17] Jiménez, R. y Sánchez, J.A. (2020). Desarrollo de una aplicación de comparación de imágenes médicas. `http://hdl.handle.net/10251/149381`

[18] Kasar, T., Barlas, P., Adam, S., Chatelain, C. y Paquet, T. (2013) Learning to detect tables in scanned document images using line information. in 12th Int. Conf. Document Anal. Recognit., 2013, pp. 1185–1189.

[19] Kavasidis, I., Palazzo, S., Spampinato, C., Pino, C., Giordano, D., Giuffrida, D. y Messina, P. (2018) A saliency-based convolutional neural network for table and chart detection in digitized documents. arXiv:1804.06236, 2018.

[20] Keysers, D., Deselaers, T., Gollan, C. y Ney, H.(2007) Deformation Models for Image Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 29. 1422-35, 2007

[21] Khan, S. A., Khalid, S. M. D., Shahzad, M. A. y Shafait, F. (2019) Table structure extraction with bi-directional gated recurrent unit networks. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 1366–1371.

[22] Kieninger, T. y Dengel, A. (1998) A paper-to-html table converting system. in Proc. document Anal. Sys. (DAS), vol. 98, 1998, pp. 356–365.

[23] Kieninger, T. G. (1998) Table structure recognition based on robust block segmentation. in Document Recognit. V, vol. 3305, 1998, pp. 22–32.

[24] Kieninger, T. y Dengel, A. (2001) Applying the t-recs table recognition system to the business letter domain. in Proc. 6th Int. Conf. Document Anal. Recognit., 2001, pp. 518–522.

[25] Kim, Y.-S. y Lee, K.-H (2008) Extracting logical structures from html tables. Comput. Standars and Interfaces, vol. 30, no. 5, pp.296-308, 2008

[26] K vecinos más próximos. (2021). Wikipedia, La enciclopedia libre. Disponible desde `https://es.wikipedia.org/w/index.php?title=K_vecinos_m%C3%A1s_pr%C3%B3ximos&oldid=140386536`.

[27] Li, M., Cui, L., Huang, S., Wei, F., Zhou, M. y Li, Z. (2020) Tablebank: Table benchmark for image-based table detection and recognition. in Proc. The 12th Lang. Resour. Eval. Conf., 2020, pp. 1918–1925.

[28] Li, Y., Gao, L., Tang, Z., Yan, Q. y Huang, Y. (2019) A gan-based feature generator for table detection. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 763–768.

[29] Liang, L.R., y Loonewy, C.G. (2020) *Dive into Image Processing: Book 1* [Libro electrónico]. N/A Recuperado de https://books.google.es/books?id=WyLdDwAAQBAJ&pg=PP12&lpg=PP12&dq=pgm+image+p2+y+p5&source=bl&ots=Duk1cR8hb8&sig=ACfU3U0w5vYs1iJC5Czr_rOJruqyDr3oug&hl=es&sa=X&ved=2ahUKEwjt2a7slanqAhUrxIUKHUclDlcQ6AEwB3oECAoQAQ#v=onepage&q=pgm%20image%20p2%20y%20p5&f=false

[30] Linear sum assignment. Disponible en https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

[31] Masuda H., Tsukamoto S., Yasutomi S., y Nakagawa H. (2004) Recognition of html table structure. in The 1st Int. Joint Conf. Natural Lang. Process. (IJCNLP-04), 2004, pp. 183–188

[32] *Online Mockup, Wireframe y UI Prototyping Tool. Moqups.* Recuperado 15 de junio de 2020, de https://moqups.com

[33] Otsu, N.(1979) A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Sysems, Man and Cybernetics*, 9(1):62-66,1979

[34] Paliwal, S. S., Vishwanath, D., Rahul, R., Sharma, M. y Vig, L. (2019) Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 128–133.

[35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. y Duchesnay, E. Dimensionality Reduction with Neighborhood Components Analysis. (2011) Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. https://scikit-learn.org/stable/auto_examples/neighbors/plot_nca_dim_reduction.html

[36] Poskanzer, J. (1991). *PGM Format Specification* http://netpbm.sourceforge.net/doc/pgm.html

[37] Prasad, D., Gadpal, A., Kapadni, K., Visave, M. y Sultanpure, K. (2020) Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. in Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. Workshops, 2020, pp. 572–573.

[38] Prieto, J. R., Andrés, J., Granell, E., Sánchez, J.A. y Vidal, E. (2022) Information Extraction in Handwritten Historical Logbooks In Document Analysis Systems: 15th IAPR International Workshop, DAS 2022, La Rochelle, France, May 22–25, 2022

[39] Qasim, S. R., Mahmood, H. y Shafait, F. (2019) Rethinking table recognition using graph neural networks. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 142–147.

[40] Raja, S., Mondal, A. y Jawahar, C. (2020) Table structure recognition using top-down and bottom-up cues. in Eur. Conf. Comput. Vision. Springer, 2020, pp. 70–86.

[41] Riba, P., Dutta, A., Goldmann, L., Fornés, A., Ramos, O. y Lladós, J. (2019) Table detection in invoice documents by graph neural networks. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 122–127.

[42] Schreiber, S., Agne, S., Wolf, I., Dengel, A. y Ahmed, S. (2017) Deepdesrt: Deep learning for detection and structure recognition of tables in document images in 14th IAPR Int. Conf. document Anal. Recognit. (ICDAR), vol. 1, 2017, pp. 1162–1167.

[43] Siddiqui, S. A., Malik, M. I., Agne, S., Dengel, A., y Ahmed, S. (2018) Decnt: Deep deformable cnn for table detection. IEEE Access, vol. 6, pp. 74 151–74 161, 2018.

[44] Siddiqui, S. A., Fateh, I. A., Rizvi, S. T. R., Dengel, A., y Ahmed, S. (2019) Deeptabstr: Deep learning based table structure recognition. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 1403–1409.

[45] Siddiqui, S. A., Khan, P. I., Dengel, A. y Ahmed, S. (2019) Rethinking semantic segmentation for table structure recognition in documents. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 1397–1402.

[46] Silva, A. C. e (2009) Learning rich hidden markov models in document analysis: Table location. in 10th Int. Conf. Document Anal. Recognit., 2009, pp. 843–847.

[47] Singh, V. (2022) Document Scanner using Semantic Segmentation Architecture DeepLabV3. Disponible en `https://learnopencv.com/deep-learning-based-document-segmentation-using-semantic-segmentation-deeplabv3-on-custom-dataset/`

[48] Sun, N., Zhu, Y. y Hu, X. (2019) Faster r-cnn based table detection combining corner locating. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 1314–1319.

[49] Tensmeyer, C., Morariu, V. I., Price, B., Cohen, S. y Martinez, T. (2019) Deep splitting and merging for table structure decomposition. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 114–121

[50] The ImageMagick Development Team.(2020). ImageMagick, Disponible en `https://imagemagick.org`

[51] Toselli, A.H., Juan, A. y Vidal, E. (2004) Spontaneous Handwriting Recognition and Classification. In *Proceedings of ICPR*, pages 433-436, England, UK,2004

[52] Tyan, C.-Y., Huang, H. K., y Niki, T. (1999) Generator for document with html tagged table having data elements which preserve layout relationships of information in bitmap image of original document. apr 1999, uS Patent 5,893,127.

[53] Villegas, M. (2012) imgtxtenh - Tool for enhancing noisy scanned text images. Disponible en `https://github.com/mauvilsa/imgtxtenh`

[54] Wang, Y., Phillips, I. T., y Haralick, R. M. (2004) Table structure understanding and its performance evaluation," Pattern Recognit. vol. 37, no. 7, pp. 1479–1497, 2004.

[55] Xue, W., Li, Q. y Tao, D. (2019) Res2tim: reconstruct syntactic structures from table images. in Int. Conf. Document Anal. Recognit. (ICDAR), 2019, pp. 749–755.

[56] Zheng, X., Burdick, D., Popa, L., Zhong, X. y Wang, N. X. R. (2021) Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vision, 2021, pp. 697–706.

[57] Zhong, X., ShafieiBavani, E. y Yepes, A. J. (2019) Image-based table recognition: data, model, and evaluation. arXiv:1911.10683, 2019.

[58] Zou, Y. y Ma, J. (2020) A deep semantic segmentation model for imagebased table structure recognition. in 15th IEEE Int. Conf. Signal Process. (ICSP), vol. 1, 2020, pp. 274–280.

APÉNDICE A

Resultados de clasificación obtenidos utilizando el vecino más cercano con PCA

	Clasificación con PCA a 1 dimensión						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	32	63	31	0	0	6	75.75 %
vol030	21	86	25	0	0	0	34.84 %
vol038	24	2	106	0	0	0	19.69 %
vol042	0	0	0	100	32	0	24.24 %
vol049	0	0	0	76	53	3	59.84 %
vol055	39	11	0	0	0	82	37.87 %
							42.03 %

Tabla A.1: Resultados clasificación con 1-Vecino más cercano y PCA con 1 dimensión

	Clasificación con PCA de 2 dimensiones						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	46	82	4	0	0	0	65.15 %
vol030	9	117	6	0	0	0	11.36 %
vol038	1	0	131	0	0	0	0.75 %
vol042	0	0	0	100	32	0	24.24 %
vol049	0	0	0	72	59	1	55.30 %
vol055	0	1	0	0	0	131	0.75 %
							26.25 %

Tabla A.2: Resultados clasificación con 1-Vecino más cercano y PCA de 2 dimensiones

	Clasificación con PCA de 3 dimensiones						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	68	64	0	0	0	0	48.48 %
vol030	8	124	0	0	0	0	6.06 %
vol038	0	0	132	0	0	0	0.00 %
vol042	0	0	0	94	38	0	28.78 %
vol049	0	0	0	67	65	0	50.75 %
vol055	0	0	0	0	0	132	0.00 %
							22.34 %

Tabla A.3: Resultados clasificación con 1-Vecino más cercano y PCA de 3 dimensiones

	Clasificación con PCA de 4 dimensiones						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	132	0	0	0	0	0	0.00 %
vol030	0	132	0	0	0	0	0.00 %
vol038	0	0	132	0	0	0	0.00 %
vol042	0	0	0	95	37	0	28.03 %
vol049	0	0	0	45	87	0	34.09 %
vol055	0	0	0	0	0	132	0.00 %
							10.98 %

Tabla A.4: Resultados clasificación con 1-Vecino más cercano y PCA de 4 dimensiones

	Clasificación con PCA a 5 dimensión						
Clase real	vol009	vol030	vol038	vol042	vol049	vol055	Porcentaje de error
vol009	132	0	0	0	0	0	0.00 %
vol030	0	132	0	0	0	0	0.00 %
vol038	0	0	132	0	0	0	0.00 %
vol042	0	0	0	123	3	0	2.27 %
vol049	0	0	0	27	103	2	21.96 %
vol055	0	0	0	0	0	132	0.00 %
							4.03 %

Tabla A.5: Resultados clasificación con 1-Vecino más cercano y PCA con 5 dimensiones

Los resultados obtenidos utilizando PCA con 6 dimensiones y 12 dimensiones se corresponden con la misma matriz de confusión y con los mismos resultados que con PCA con 5 dimensiones (véase Tabla A.5).