

Universidad de La Habana

FACULTAD DE MATEMÁTICA Y COMPUTACIÓN



OCR PARA CIENCIAS MATEMÁTICAS

Paula Silva Lara C-312
José Miguel Zayas Pérez C-312
Ricardo Cápiro Colomar C-312
Alejandro Lamelas Delgado C-311

[Proyecto en GitHub](#)

Introducción

La revista Ciencias Matemáticas es una publicación académica de la Facultad de Matemática y Computación de la Universidad de La Habana, fundada en 1980, que abarca temas de matemáticas y computación. Dado que los números de la revista anteriores a su versión digital no están disponibles en formato accesible, es necesario digitalizarlos para facilitar su uso como recurso de estudio. El objetivo de este proyecto es precisamente llevar a cabo esa digitalización, permitiendo un acceso más amplio y eficiente a este valioso material académico.

Para llevar a cabo la digitalización, se utilizó un sistema de Reconocimiento Óptico de Caracteres (Optical Character Recognition - OCR), que permite leer las imágenes escaneadas y convertir el contenido en código LaTeX.

A continuación, se dará respuestas a las siguientes interrogantes, con el objetivo de una mejor comprensión del proyecto. ¿Qué herramientas se utilizaron? ¿Qué solución se propuso? ¿Qué aspectos faltan por solucionar?

TexTeller

TexTeller es una herramienta de software diseñada para la conversión de imágenes que contienen fórmulas matemáticas en código LaTeX. Este proceso se conoce como OCR (Reconocimiento Óptico de Caracteres) especializado en contenido matemático, que ayuda a automatizar la digitalización de ecuaciones complejas a formato digital. Este sistema está orientado a académicos, estudiantes, y profesionales que trabajan con documentación técnica y científica.

TexTeller utiliza técnicas avanzadas de aprendizaje profundo para interpretar ecuaciones y símbolos matemáticos desde imágenes y convertirlos en código LaTeX. Su propósito es simplificar la transcripción manual de fórmulas, un proceso que suele ser tedioso y propenso a errores. Al ser una herramienta basada en inteligencia artificial, TexTeller tiene una capacidad de generalización amplia, lo que significa que puede manejar una gran variedad de estilos de escritura y complejidades en las fórmulas.

TexTeller funciona en tres fases principales:

1. Carga de Imagen: El usuario proporciona una imagen que contiene las fórmulas matemáticas o símbolos que desea convertir.
2. Procesamiento y Reconocimiento: TexTeller utiliza su modelo OCR especializado en matemáticas para analizar la imagen. El software identifica caracteres individuales, relaciones espaciales y el contexto matemático.
3. Generación de Código LaTeX: Una vez reconocidos los símbolos, TexTeller genera un archivo o código LaTeX equivalente que puede ser utilizado en documentos técnicos, artículos científicos o libros académicos.

TexTeller además cuenta con un modelo para clasificar entre texto y fórmulas, el cual clasifica el contenido de las imágenes dividiendo los elementos con cajas (BBox – Bounding Boxes): identifica texto con la etiqueta "text", fórmulas incrustadas en el texto como "embedding", y fórmulas separadas del texto como "isolated".

Otras herramientas utilizadas

Además de TexTeller se utilizaron otras herramientas para lograr una mejor digitalización de las imágenes.

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto diseñada para aplicaciones de visión por computadora. OpenCV desempeña un papel clave en el preprocesamiento de imágenes. Antes de que el OCR matemático pueda convertir las fórmulas en LaTeX, OpenCV mejora la calidad de la imagen, aplicando técnicas de detección de bordes, umbralización, y eliminación de ruido. Estas operaciones optimizan las imágenes para mejorar la precisión del reconocimiento de caracteres y símbolos matemáticos, garantizando resultados más precisos y eficientes.

Tesseract es un motor de reconocimiento óptico de caracteres (OCR) de código abierto desarrollado por Google. Es una herramienta poderosa utilizada para convertir imágenes que contienen texto en texto editable. Tesseract puede manejar varios idiomas y soporta múltiples formatos de imagen, como PNG, JPEG, TIFF, y más. Aunque inicialmente fue diseñado para texto impreso, su funcionalidad ha mejorado con el tiempo, permitiendo su uso en la detección de caracteres en imágenes con diferentes niveles de complejidad, como manuscritos o documentos escaneados.

Solución propuesta

Para la realización de este proyecto, se procedió de la siguiente manera. Para un mejor entendimiento se muestra el flujo del proyecto con la siguiente imagen:

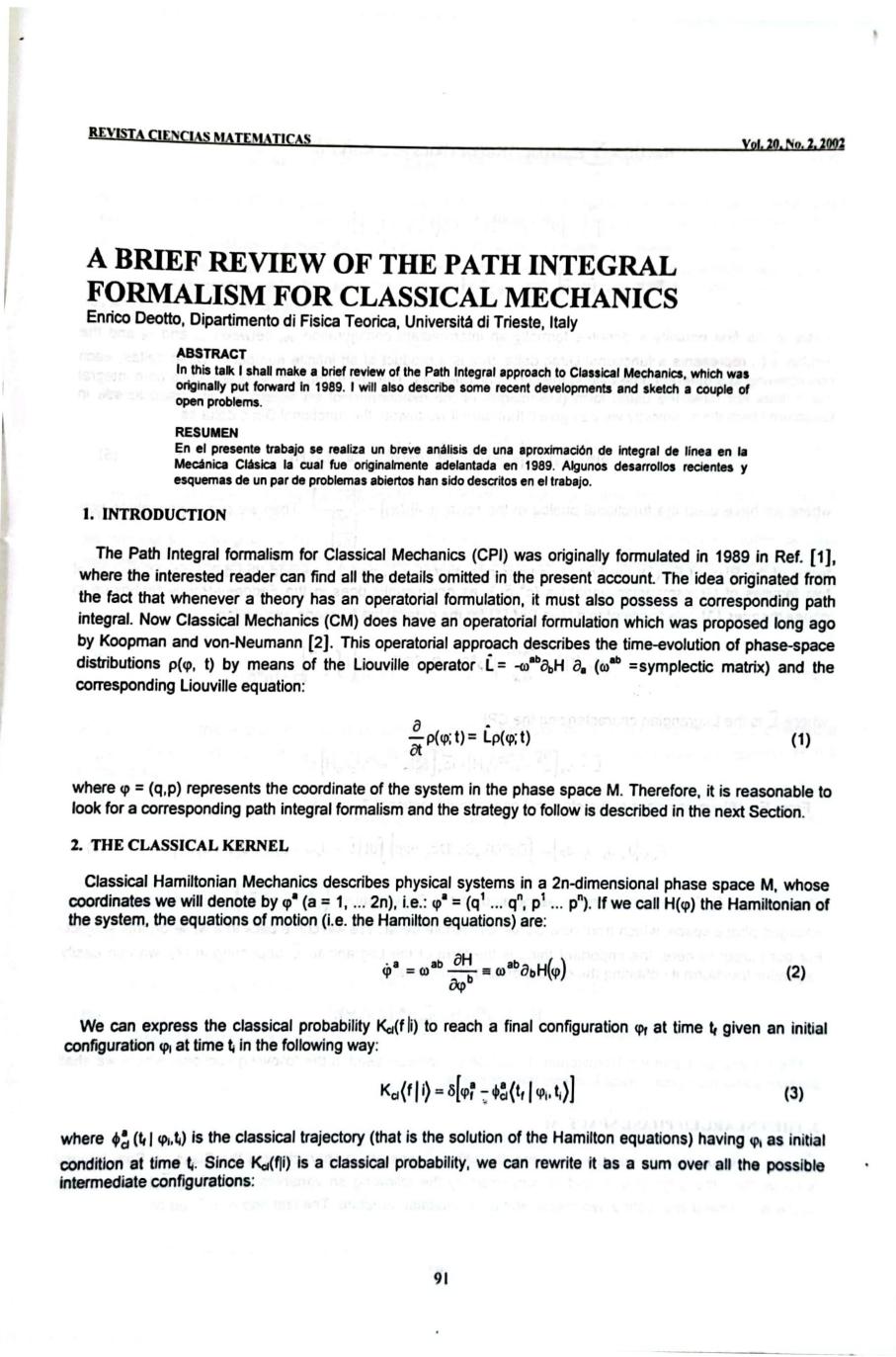


Figure 1: Imagen de la revista Ciencias Matemáticas

En primer lugar, se eliminó el ruido de la imagen. Al trabajar con una revista antigua, algunas páginas presentaban residuos de texto de otras páginas que aparecían como marcas de fondo. Estas letras, aunque casi invisibles, generaban ruido en el modelo, lo que podía incluirlos en el texto resultante. Para mitigar este problema, se utilizó la biblioteca de Python OpenCV, que permitió filtrar el contenido de las imágenes y eliminar los ruidos no deseados.

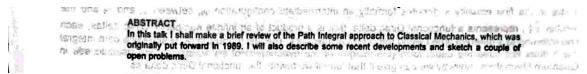


Figure 2: Antes de eliminar el ruido

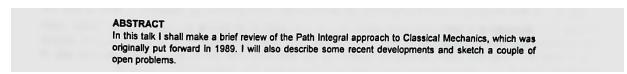


Figure 3: Después de eliminar el ruido

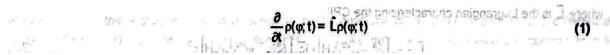


Figure 4: Antes de eliminar el ruido

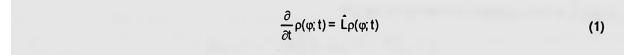


Figure 5: Después de eliminar el ruido

Luego, utilizando el modelo de clasificación de texto y fórmulas de TexTeller, se establecieron las cajas delimitadoras (BBox) para el texto y las fórmulas matemáticas. Se generaron dos listas de cajas delimitadoras: ocr_bboxes (lista de BBox de texto) y latex_bboxes (lista de BBox de fórmulas). Estas listas se ordenaron y se resolvieron los conflictos de superposición entre ellas para evitar solapamientos y confusiones en la identificación de elementos.

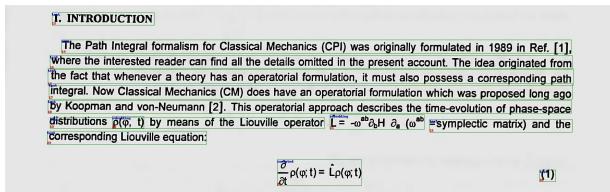


Figure 6: Antes de eliminar los conflictos de las cajas delimitadoras

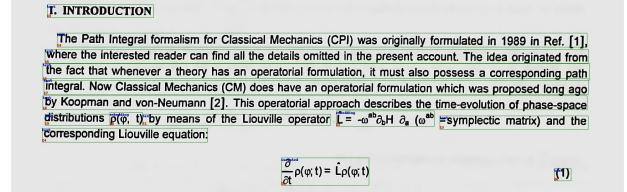


Figure 7: Después de eliminar los conflictos de las cajas delimitadoras

TexTeller identifica las cajas delimitadoras de texto por cada renglón del párrafo que está reconociendo, pero presenta inconvenientes cuando el contenido de las imágenes está levemente inclinado o distorsionado. Esto puede provocar que las cajas BBox no identifiquen correctamente el contenido, resultando en cortes de letras o símbolos. Para resolver este problema, se realizaron modificaciones que permiten tratar párrafos completos como una sola caja, en lugar de dividirlos en varias líneas. Además, se introdujo un margen de error en el tamaño de las cajas para que el contenido distorsionado o inclinado sea reconocido sin problemas.

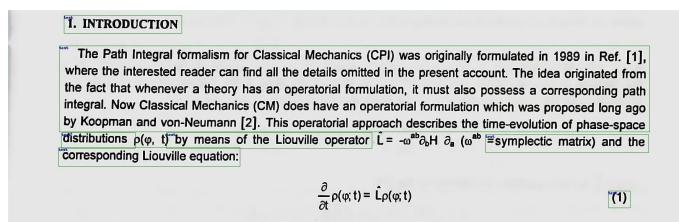


Figure 8: Unión de las cajas delimitadoras de texto contiguas

Seguidamente, se extrajo y se reconoció el contenido en cada BBox. En el caso del texto, fue necesario utilizar un OCR más potente, como Tesseract, ya que el texto reconocido con TexTeller se devolvía con todas las palabras unidas, sin espacios entre ellas, y no se lograba identificar con exactitud algunos caracteres. En el caso de las fórmulas, se utilizó el modelo de TexTeller.

Por último, se formatearon las fórmulas en su correspondiente código LaTeX, y se devolvió a la aplicación visual el contenido completo escaneado de la imagen.

REVISTA CIENCIAS MATEMATICAS

02.

1

A BRIEF REVIEW OF THE PATH INTEGRAL FORMALISM FOR CLASSICAL MECHANICS Enrico Deotto, Dipartimento di Fisica Teorica, Universita di Trieste, Italy

ABSTRACT

In this talk I shall make a brief review of the Path Integral approach to Classical Mechanics, which was originally put forward in 1989. I will also describe some recent developments and sketch a couple of

im)* ac Open problems.

RESUMEN

En el presente trabajo se realiza un breve análisis de una aproximación de integral de línea en la Mecánica Clásica la cual fue originalmente adelantada en 1989. Algunos desarrollos recientes y esquemas de un par de problemas abiertos han sido descritos en el trabajo.

1. INTRODUCTION

The Path Integral formalism for Classical Mechanics (CPI) was originally formulated in 1989 in Ref. [1], where the interested reader can find all the details omitted in the present account. The idea originated from the fact that whenever a theory has an operatorial formulation, it must also possess a corresponding path integral. Now Classical Mechanics (CM) does have an operatorial formulation which was proposed long ago by Koopman and von-Neumann [2]. This operatorial approach describes the time-evolution of phase-space distributions $\rho(\varphi, t)$ by means of the Liouville operator $\hat{L} = -\omega^{ab}\partial_b H\partial_a$ (ω^{ab} =symplectic matrix) and the corresponding Liouville equation:

$$\frac{\partial}{\partial t}\rho(\varphi; t) = \hat{L}\rho(\varphi; t)$$

where $\varphi = (q, p)$ represents the coordinate of the system in the phase space M. Therefore, it is reasonable to look for a corresponding path integral formalism and the strategy to follow is described in in the — next Section.

2. THE CLASSICAL KERNEL

Classical Hamiltonian Mechanics describes physical systems in a $2n$ -dimensional phase space M, whose coordinates we will denote by : φ^a (a= 1, ... 2n), ie: $\varphi^a = (q^1 \cdots q^n, p^1 \cdots p^n)$. If we call $H(\varphi)$ the Hamiltonian of "Th ae te"

the system, the equations of motion (i. e. the Hamilton equations) are:

$$\dot{\varphi}^a = \omega^{ab}\frac{\partial H}{\partial \varphi^b} \equiv \omega^{ab}\partial_b H(\varphi) \quad (2)$$

We can express the classical probability $K_{cl}(f|i)$ to reach — final configuration — φ_1 , at time — t_1 , given an initial configuration φ_i at time — t_i in the following way:

$$K_{cl}(f|i) = \delta[\varphi_f - \phi_{cl}^f(t_1|\varphi_i, t_1)] \quad (3)$$

where $\phi_{cl}^f(t_1|\varphi_i, t_1)$ is the classical trajectory (that is the solution of the Hamilton equations) having φ_i , as initial condition at time t_i . Since K_{cl} is a classical probability, we can rewrite it as a sum over all the possible

ne gS ns ee oat intermediate configurations:

Figure 9: Código que devuelve el OCR compilado en LaTeX

Tipos de errores identificados

Leyenda

- Reconocimiento incorrecto por parte del modelo.
- Reconocimiento de corte de letras o fórmulas en las cajas delimitadoras.
- Reconocimiento de un texto como fórmula o viceversa.

Deficiencias por resolver

Hasta el momento, el OCR funciona adecuadamente para imágenes que contienen texto y fórmulas, pero presenta deficiencias cuando la imagen incluye figuras. Además, es crucial que las imágenes no tengan una inclinación o distorsión significativa, ya que, a pesar del margen de error aplicado, cuando las imágenes están muy modificadas, el OCR no puede procesarlas correctamente, lo que genera interferencias en el texto escaneado.

se puede interpretar y reescribir usando las convenciones notacionales de la teoría de conjuntos como
 $A = A_{11} \cup \dots \cup A_{1n} \cup \dots \cup A_{1m} \cup \dots \cup A_m$

Esta interpretación ha sido adoptada implícitamente en muchos lenguajes, escribiéndose en la forma siguiente:

$$A \rightarrow A_{11} \cup \dots \cup A_{1n} \mid \dots \mid A_{1m} \cup \dots \cup A_m$$

Ocurre que estas dos operaciones no siempre bastan para confeccionar una especificación clara tales como la agrupación de operaciones y las operaciones de clausura, complemento y diferencia (complemento relativo). Con estas adiciones se puede eliminar la "superposición" de producciones y no-terminales, mantener locales las definiciones superfluyas y conservar la unidad conceptual tan útil en muchos casos.

La agrupación permite factorizar sin necesidad de adicionar a la gramática un no-terminal o una producción. Esto se puede ver en la construcción `case`, que sin la agrupación se escribiría:

```
<case> → case <lista de etiqs> of <lista case> <parte implic> end
<parte implic> → else <instrucción> | ε
```

y utilizando la agrupación escribiríamos:

```
<case> → case <lista de etiqs> of <lista case> (else <instrucción> | ε) end
```

Aquí se puede notar la interpretación de la agrupación como la introducción del andamiento de estructuras, algo equivalente a lo hecho por la notación EBNF —equivalente a las gramáticas libres de contexto—, o las expresiones regulares —una notación equivalente a las gramáticas regulares. Otra interpretación es el considerarla como una estructura anónima, parecida a los tipos anónimos de algunos lenguajes de programación; esto puede extenderse a otras estructuras lingüísticas. Las expresiones regulares deben su potencia expresiva, en buena medida, a la operación de clausura. Las expresiones compacta la generación de cadenas de longitud variable, formadas por los elementos de un conjunto. Usándola se puede prescindir de la recursividad izquierda o derecha en una gramática, y en vez de escribir:

$$A \rightarrow a A \mid ε$$

escribiríamos:

$$A \rightarrow a^*$$

Además, al incorporar la operación de clausura a las gramáticas libres del contexto se obtiene uniformidad notacional entre la especificación del análisis lexicográfico y el sintáctico, y puede prescindirse de una especificación separada para ambas fases.

La diferencia puede ser muy útil cuando se desea definir un lenguaje mediante la exclusión de casos a otro lenguaje. Un ejemplo es el caso de una herramienta que genere código hacia otros lenguajes de identificadores de todos los lenguajes para los cuales es probable se exporte, y para no quitarle la flexibilidad de restricciones. Entonces el nombre de una entidad podría definirse usando la diferencia como una restricción al lenguaje de los alias.

El complemento de un conjunto C se define a través de la diferencia, por:

$$C^c = U - C$$

donde U es el conjunto universo con respecto al cual se complementa C. Al usar esta operación en la definición de un lenguaje usando una gramática libre del contexto, debe definirse el conjunto de identificadores de todos los lenguajes para los cuales es probable se exporte, y para no quitarle la flexibilidad de restricciones. Un ejemplo del uso de esta opción es la clara definición de los comentarios delimitados por llaves con la producción:

151

Figure 11: Código en la LaTeX compilado

Figure 10: Imagen con texto inclinado

Al editar la imagen un poco para solucionar el problema de la inclinación manualmente, se muestran los siguientes resultados:

se puede interpretar y reescribir usando las convenciones notacionales de la teoría de conjuntos como

$$A = A_{11} \cup \dots \cup A_{1n} \cup \dots \cup A_{im} \cup \dots \cup A_{in}$$

Esta interpretación ha sido adoptada implícitamente en muchos lenguajes, escribiéndose en la forma siguiente:

$$A \rightarrow A_{11} \cup \dots \cup A_{1n} \cup \dots \cup A_{im} \cup \dots \cup A_{in}$$

Ocurre que estas dos operaciones no siempre bastan para confeccionar una especificación clara y compacta. Al usar las austeras gramáticas libres del contexto es muy deseable contar con otros recursos, tales como la agrupación de operaciones y las operaciones de clausura, complemento y diferencia (complemento relativo). Con estas adiciones se puede eliminar la "superpoblación" de producciones y no terminales, mantener locales las definiciones superfljas y conservar la unidad conceptual tan útil en muchos casos.

La agrupación permite factorizar sin necesidad de adicionar a la gramática un no-terminal o una producción. Esto se puede ver en la construcción `case`, que sin la agrupación se escribiría:

$$\begin{aligned} <\text{case}> &\rightarrow \text{case } <\text{lista de etiqs.}> \text{ of } <\text{lista case}> <\text{parte implic.}> \text{ end} \\ &\quad <\text{parte implic.}> \rightarrow \text{else } <\text{instrucción}> \mid \epsilon \end{aligned}$$

y utilizando la agrupación escribiríamos:

$$<\text{case}> \rightarrow \text{case } <\text{lista de etiqs.}> \text{ of } <\text{lista case}> (\text{else } <\text{instrucción}> \mid \epsilon) \text{ end}$$

Aquí se puede notar la interpretación de la agrupación como la introducción del anidamiento de estructuras, algo equivalente a lo hecho por la notación EBNF —equivalente a las gramáticas libres del contexto—, o las expresiones regulares —una notación equivalente a las gramáticas regulares. Otra lenguaje de programación: esto puede extenderse a otras estructuras lingüísticas. Las expresiones regulares deben su potencia expresiva, en buena medida, a la operación de clausura. Ésta define de forma compacta la generación de cadenas de longitud variable, formadas por los elementos de un conjunto. Usándola se puede prescindir de la recursividad izquierda o derecha en una gramática, y en vez de escribir:

$$A \rightarrow aA \mid \epsilon$$

escribiríamos:

$$A \rightarrow a^*$$

Además, al incorporar la operación de clausura a las gramáticas libres del contexto se obtiene uniformidad notacional entre la especificación del análisis lexicográfico y el sintáctico, y puede prescindirse de una especificación separada para ambas fases.

La diferencia puede ser muy útil cuando se desea definir un lenguaje mediante la exclusión de casos a otro lenguaje. Un ejemplo es el caso de una herramienta que genere código hacia varios lenguajes de programación, por tanto los nombres de las entidades deben cumplir con las restricciones para los identificadores de todos los lenguajes para los cuales es probable se exporte, y para no quitarle la flexibilidad de nombramiento de referencias a la especificación permite asociar un alias que no poseerá todas estas restricciones. Entonces el nombre de una entidad podría definirse usando la diferencia como una restricción al lenguaje de los alias.

El complemento de un conjunto C se define a través de la diferencia, por:

$$C^c = U - C$$

donde U es el conjunto universo con respecto al cual se complementa C. Al usar esta operación en la definición de un lenguaje usando una gramática libre del contexto, debe definirse el conjunto asumido como universo. En las implementaciones encontradas siempre se escoge el conjunto de símbolos terminales. Un ejemplo del uso de esta opción es la clara definición de los comentarios delimitados por llaves con la producción:

151

Figure 12: Imagen con problema de inclinación solucionado

$$A = A_{11} \cup \dots \cup A_{1n} \cup \dots \cup A_{im} \cup \dots \cup A_{in} \quad (1)$$

sigue':

$$A \rightarrow A_{11} \cup \dots \cup A_{1n} \mid \dots \mid A_{im} \cup \dots \cup A_{in} \quad (2)$$

casos.

EEN NESTS LE
iparte implic._b b — else jinstrucción_c — c

$$A \rightarrow aA \mid \epsilon \quad (3)$$

escribiríamos:

$$A \rightarrow a^* \quad (4)$$

especificación separada para ambas fases.

La diferencia puede ser muy útil cuando se desea definir un lenguaje mediante la exclusión de casos a otro lenguaje. Un ejemplo es el caso de una herramienta que genere código hacia varios lenguajes de programación, por tanto los nombres de las entidades deben cumplir con las restricciones para los identificadores de todos los lenguajes para los cuales es probable se exporte, y para no quitarle la flexibilidad de nombramiento de referencias a la especificación permite asociar un alias que no poseerá todas estas restricciones. Entonces el nombre de una entidad podría definirse usando la diferencia como una restricción

oO. 7 1 wo al lenguaje de los alias.

El complemento de un conjunto C se define a través de la diferencia, por:

$$C^c = U - C \quad (5)$$

donde U es el conjunto universo con respecto al cual se complementa C. Al usar esta operación en la definición de un lenguaje usando una gramática libre del contexto, debe definirse el conjunto asumido como universo. En las implementaciones encontradas siempre se escoge el conjunto de símbolos terminales. Un ejemplo del uso

Pa a ew eee ee Ee Pal em ree a Nee — (5 ee a en ee — a de esta opción es la clara definición de los comentarios delimitados por llaves con la producción:

151

Figure 13: Código en LaTeX compilado

A pesar de la mejora mostrada por el OCR con la última imagen, es notable la ausencia de algunos párrafos que aparecen en la imagen original y no en el código en LaTeX. Estos párrafos son los que mayor inclinación presentan. Incluso con la edición de la imagen, esta inclinación persiste y afecta su reconocimiento en las cajas delimitadoras.

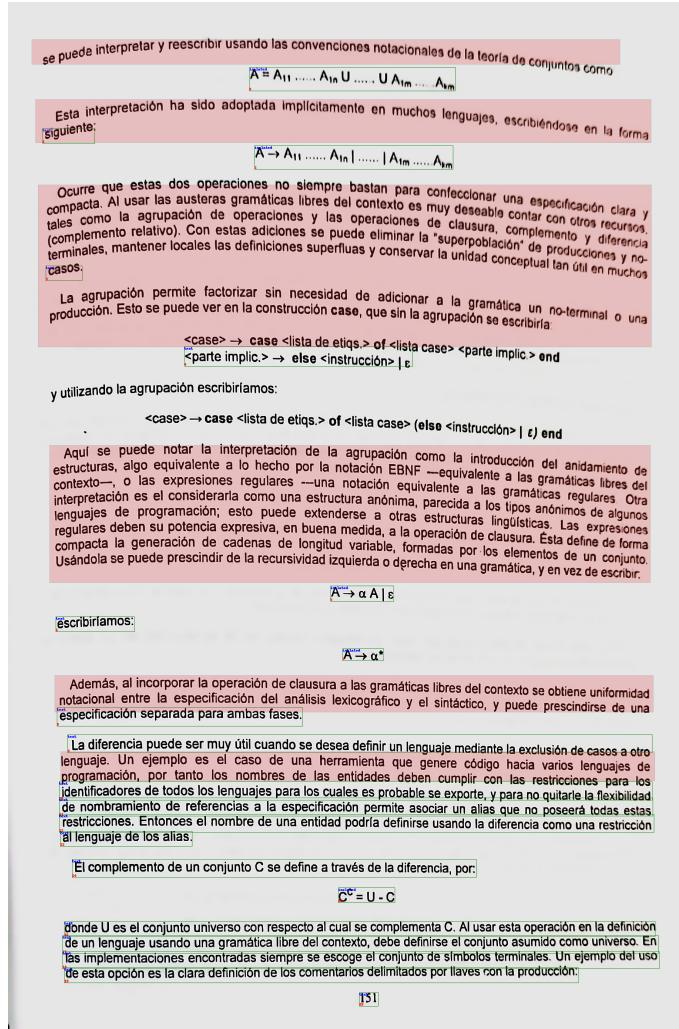


Figure 14: Párrafos que no son reconocidos por las cajas delimitadoras

Aunque el OCR actual ofrece resultados bastante precisos, continúa siendo necesaria la presencia de un ojo humano que verifique que el código LaTeX sea correcto y que solucione los problemas que puedan presentarse. Además, debido al tamaño del modelo de TexTeller y la complejidad de sus múltiples funciones, el OCR toma un tiempo considerable para devolver el código LaTeX escaneado; sin embargo, este proceso sigue siendo más rápido que realizar la transcripción manualmente.

A continuación otro ejemplo de imagen procesada por el OCR:

$$F_\varphi^m(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\varphi e^{-im\varphi} \int_{-\infty}^{\infty} dz e^{-ikz} F_\varphi(\varphi, z)$$

$$F_2^m(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\varphi e^{-im\varphi} \int_{-\infty}^{\infty} dz e^{-ikz} F_2(\varphi, z) \quad (9)$$

$$F_2^m(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} d\varphi e^{-im\varphi} \int_{-\infty}^{\infty} dz e^{-ikz} F_2(\varphi, z) \quad (10)$$

For example the component of A_φ [superiormente] for $p > a$:

$$A_\varphi = \frac{\mu_0 a}{4\pi} \sum_{m=-\infty}^{\infty} \int_0^\infty e^{im\varphi} e^{ikz} [l_{m-1}(|k|p) K_{m-1}(|k|a) + l_{m+1}(|k|p) K_{m+1}(|k|a))] F_\varphi^m(k) dk \quad (11)$$

Given the axial symmetry of the problem one can build up the solution starting with the expression of the magnetic field in z direction for a loop of current of radio located in the X-Y plane with its center at the origin.

Then using $\nabla \times \vec{A} = \vec{B}$ we have the Bessel - Fourier series expansion in terms of cylinder functions for the magnetic field;

$$B_z(p, \varphi, z) = \frac{\mu_0 a}{2\pi} \sum_{m=-\infty}^{\infty} \int_0^\infty e^{im\varphi} e^{ikz} [k] F_\varphi^m(k) l_m(|k|p) K_m(|k|a)] \quad (12)$$

For next, on using the polynomial expansion of the Bessel functions we obtain

$$B_z(p, \varphi, z) = B_0(z)p^0 + B_2(z)p^2 + B_4(z)p^4 + B_6(z)p^6 + \dots \quad (13)$$

calling on superposition we should then integrate over the sample surface for J_s and over the volume for J_V . The magnetic field in Z direction due to these current densities is given by the following integrals and

$$B_{zx}(p, z) = \int_{-1}^1 B_z(z') dz' \quad \text{and} \quad B_{zy}(p, z) = \int_0^\pi \int_{-1}^1 B_z(p', z') dz' \quad (14)$$

on replacing eqs. (13) in (14), we have series expansion of the trapped magnetic field of the form (13) and making $p = \omega$, we get the field along the z -axis.

$$B_{zx}(z) = \frac{\mu_0}{2} \left[\int_0^\infty \left[\frac{(z+1)}{\sqrt{a^2 + (z+1)^2}} - \frac{z}{\sqrt{a^2 + z^2}} \right] + \int_0^\infty \left[(z+1) \cdot \text{Log} \left[\frac{a + \sqrt{a^2 + (z+1)^2}}{(z+1)} \right] - z \cdot \text{Log} \left[\frac{a + \sqrt{a^2 + z^2}}{z} \right] \right] \right] \quad (15)$$

LEVITATION FORCE

The vertical component of the levitation force between the magnet and the superconductor is given by

$$F = m \frac{\partial}{\partial z} B_{zx}(z) \quad (16)$$

At this point will also incorporate the contribution of J_V to which the previous procedure applies. On using the Taylor series expansion of at the dipole position $z = d$ and substituting the expression subsequently in (16) we obtain the final expression of the levitation force. If one keeps just the lower order terms the following expression arises

$$F = m \cdot [j_s A_{1s}(d) + j_v A_{1v}(d) + j_n A_{1n}(d)] \quad (17)$$

Figure 15: Imagen de la revista Ciencias Matemáticas

Figure 16: Código en LaTeX compilado

Ventajas

A pesar de no lograr un reconocimiento perfecto de las imágenes, el OCR propuesto representa un avance significativo, al mejorar la eficiencia en la conversión de revistas matemáticas antiguas a formato digital. El tiempo requerido por un ser humano para corregir los errores generados por el código en LaTeX es notablemente inferior al que se necesitaría para escribir todo ese código manualmente.

Aplicación visual

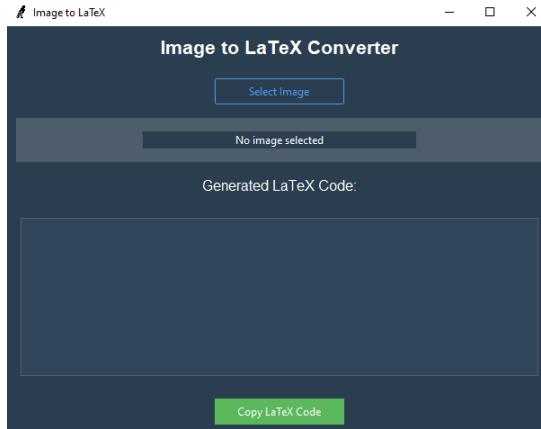


Figure 17: Inicio de la aplicación visual

Es una aplicación sencilla en la cual se puede seleccionar la imagen que se desea escanear y cuando el código es devuelto, se puede copiar dicho código LaTeX para ejecutarlo en un compilador de ese lenguaje.

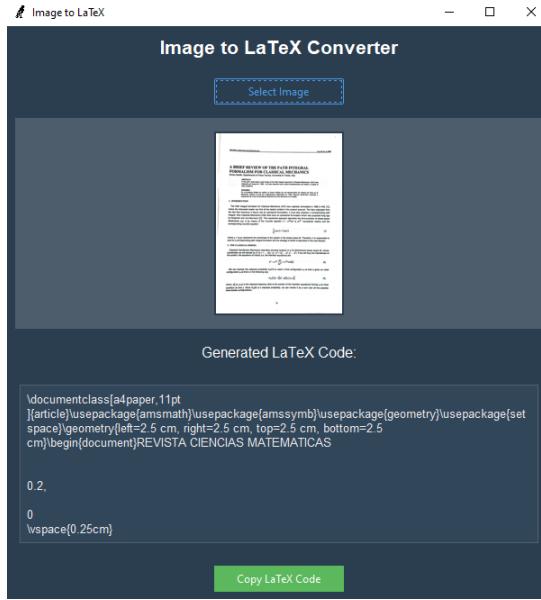


Figure 18: Vista con el texto escaneado devuelto

Referencias

[Repositorio en GitHub sobre TexTeller](#)