



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**Proyecto fin de Carrera Grado**

**GITI**

**Segmentación de células en imágenes 3D con técnicas de machine learning**

**Realizado por  
(ponente): Adrián López Carrillo**

**Dirigido por  
Junta de Centro**

**Departamento  
Matemática Aplicada I**

**Sevilla, 02 de Agosto de 2020 (v.0.0.3)**

---

## Resumen

---

En la actualidad existen microscopios que emplean técnicas ópticas para reconstruir imágenes 3D con una alta precisión. Gracias a esto los investigadores tienen a su alcance imágenes microscópicas de alta calidad y pueden sacar conclusiones de ellas. Estas imágenes suelen requerir un procesamiento digital con el objetivo de discernir los elementos importantes del resto.

En este proyecto se abordará el problema de la segmentación de células en imágenes 3D. Para ello se usarán imágenes cedidas por el Departamento de Biología Celular de la Facultad de Biología de la Universidad de Sevilla. En cada imagen hay decenas de células, todas en contacto con otras células sin espacio entre ellas.

El procesamiento digital de estas imágenes actualmente se hace de forma manual teniendo una duración de una a dos semanas, por lo que la automatización de este proceso conllevará un gran ahorro de tiempo.

Respecto a las técnicas usadas, este proyecto se centrará en el uso de redes neuronales para la segmentación de células. Se validará la efectividad del uso de redes neuronales y se estudiará qué tipo de red neuronal y arquitectura será mejor para esta tarea, teniendo en cuenta la exactitud de la segmentación y el coste computacional.

En el análisis de antecedentes se comprobará que la CNN (Convolutional Neural Network o Red Neuronal Convolucional) será con la que se obtienen mejores resultados en el reconocimiento de patrones en imágenes. También se verá que al diseñar una CNN con la arquitectura U-Net se obtienen buenos resultados.

Se probarán varios modelos de CNNs para la segmentación de células, esperándose el mejor resultado de la arquitectura U-Net.

Adicionalmente, se estudiará el uso de un preprocesado y postprocesado para aumentar la eficacia de la segmentación, así como un posterior mapeado a color de las células para ayudar a su visualización.

Esta herramienta estará disponible para el usuario final por medio de una imagen docker, para facilitar su distribución y que perdure el paso del tiempo.

---

## Índice general

---

<b>Índice general</b>	<b>2</b>
<b>Índice de cuadros</b>	<b>3</b>
<b>Índice de figuras</b>	<b>4</b>
<b>1 Partes y plantillas</b>	<b>1</b>
1.1 Partes . . . . .	1
1.2 Impresión . . . . .	2
1.3 Plantillas . . . . .	2
<b>2 Ejemplos</b>	<b>3</b>
2.1 Manejo de la bibliografía . . . . .	3
2.2 Código . . . . .	3
2.3 Imágenes . . . . .	3
2.4 Cuadros (mal llamados Tablas) . . . . .	4
<b>3 Definición de objetivos</b>	<b>5</b>
<b>4 Análisis de antecedentes y aportación realizada</b>	<b>6</b>
4.1 Red Neuronal . . . . .	6
4.1.1 Perceptrón . . . . .	6
4.1.2 Partes de una Red Neuronal . . . . .	7
4.1.3 Algoritmos de aprendizaje . . . . .	7
4.2 Red Neuronal Convolucional . . . . .	7
4.3 Reto ImageNet . . . . .	7
4.4 Arquitectura U-Net . . . . .	7
<b>Referencias</b>	<b>8</b>

---

# Índice de cuadros

---

2.1	Cuadro de prueba . . . . .	4
-----	----------------------------	---

---

# Índice de figuras

---

2.1	Ada Lovelace . . . . .	4
4.1	Unidad de computación genérica . . . . .	6
4.2	Representación de todos los valores del vector de entrada . . . . .	7

---

## Partes y plantillas

---

### 1.1– Partes

Todas las memorias de Trabajo fin de Grado deberán constar de las siguientes partes

- Portada (según formato oficial). No debe incluir número de página. Debe incluir:
  - Sello de la universidad de Sevilla a dos tintas
  - ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
  - Trabajo fin de Grado
  - *Denominación del Grado*
  - Realizado por: *Nombre y apellidos del estudiante*
  - Dirigido por: *Nombre y apellidos del tutor o tutores*
  - Departamento *Nombre del departamento en el que se lee el TfG*
  - Sevilla, *Mes y año de la convocatoria de entrega*
- Preliminares: las páginas no se numeran o se numeran con números romanos.
  - Resumen en castellano (máximo una hoja)
  - Abstract (resumen en inglés, obligatorio para el caso de las memorias escritas en inglés, opcional para las escritas en castellano)
  - Agradecimientos (opcional)
  - Índice general (contenido de la memoria, con mención de las partes en que está dividida)
  - Índice de figuras (opcional)
  - Índice de cuadros (mal llamados en general, tablas) (opcional)
  - Índice de código o algoritmos (opcional).
- Cuerpo de la memoria, dividida en capítulos. El contenido de la memoria ha de incluir los elementos característicos de un proyecto de ingeniería o de un estudio o trabajo en el ámbito de una investigación, los cuales, en el sentido más amplio, son:
  - Definición de objetivos.
  - Análisis de antecedentes y aportación realizada.
  - Análisis temporal y de costes de desarrollo.

- Análisis de requisitos, diseño e implementación.
- Manual de usuario, en su caso.
- Pruebas.
- Comparación con otras alternativas.
- Conclusiones y desarrollos futuros

Estos puntos podrán ser ajustados y modificados en función de la naturaleza del proyecto realizado.

- Bibliografía: se deben documentar las fuentes bibliográficas utilizadas en el formato APA 2009.
- Índice alfabético o glosario (Lista ordenada de los conceptos, los nombres propios, etc.; que aparecen en la memoria, con las indicaciones necesarias para su localización)(opcional)
- Apéndices: Si la memoria contiene algún artículo de investigación o similar, este se incluirá en los Apéndices. Es necesario, en ese caso, incluir en la página anterior una hoja con la citación bibliográfica. En esta citación, el título del artículo se debe enlazar con la página web de la revista en la que aparecen el resumen o abstract y el acceso al texto completo.

La numeración de las páginas de la bibliografía y del glosario debe continuar la del cuerpo de la memoria. Los apéndices pueden llevar su propia numeración independiente o usar la general del cuerpo de la memoria.

## 1.2– Impresión

La entrega del memoria y en su caso, el depósito en biblioteca, se hacen en formato electrónico. Debido a ello, la memoria se presentará *a una cara*. Si se requiriera por algún motivo la impresión de la misma, se recomienda vivamente preparar la memoria adecuadamente para su impresión. Algunas sugerencias:

- Dejar una página en blanco cuando sea necesario para que los capítulos comiencen siempre en página impar (derecha)
- Ajustar los márgenes para que el exterior sea ligeramente más grande que el interior.
- Ajustar las cabeceras y pies de página (en su caso). Por ejemplo, si el número de página ocurriese en un lateral de la cabecera o pie, este debe ser siempre el exterior (derecho para las páginas impares, izquierdo para las pares)

## 1.3– Plantillas

Se encuentra disponible una plantilla LaTeX en la sección de documentos de la plataforma de la aplicación de TfG de la ETSII <https://tfc.eii.us.es/TfG/>. Esa plantilla ha sido utilizada para preparar este documento. Se espera en breve disponer de plantillas de ejemplo para las aplicaciones OpenOffice Writer y Office Word.

Nota: Las plantillas se proporcionan como ejemplo, las condiciones obligatorias son las que constan en este procedimiento.

---

### Ejemplos

---

#### 2.1– Manejo de la bibliografía

En esta sección mostramos brevemente ejemplos de referencia a la bibliografía citando un libro (de Sousa, 2004), un artículo (Bezos, 2007) y una página web (Autores, 2014).

Se recuerda que son campos obligatorios en todos los ítems de la bibliografía: autor(es), título del libro o artículo y año de publicación. En el caso de páginas web, es obligatoria la fecha de la última consulta. En general, la bibliografía debe ayudar al lector a encontrar fácilmente los ítems citados.

#### 2.2– Código

En general se debe evitar incluir código o pseudocódigo en la memoria. Si fuese preciso, se destacará de forma que sea fácilmente identificable y se indexarán los trozos de código incluidos. Un ejemplo puede verse a continuación.

```
1 %COMANDO PARA INSERTAR UN CUADRO UTILIZANDO EL FORMATO:
2 %1---> especificar numero de columnas y su alineacion ejm:
3 % |r||c|c| r=right, c=center, l=left
4 %2---> especificar el caption o titulo de la figura
5 %3---> label para hacer referencia a la tabla insertada
6 %4---> contenido de tabla separando columnas con & y filas con \\
7 \newcommand{\cuadro}[4]{
8     \begin{table}[htb]
9         \centering
10        \begin{tabular}{#1}
11            \hline
12            #4
13            \hline
14        \end{tabular}
15        \caption{#2}
16        \label{#3}
17    \end{table}
18 }
```

Código 2.1: Código de ejemplo en LaTeX

#### 2.3– Imágenes

Este es un ejemplo de inclusión de figura en el texto (véase la figura 2.1).





Figura 2.1: Ada Lovelace

Figuras y cuadros se colocarán preferentemente tras el párrafo en el que son llamados por primera vez. Si no cupieran, se colocarán (en orden de preferencia):

- Al final de la página en que se llaman
- Al principio de la siguiente página
- Al final del capítulo

siempre respetando el orden de aparición en el texto.

## 2.4– Cuadros (mal llamados Tablas)

Este es un ejemplo de inclusión de cuadro en el texto. Véase el cuadro 2.1

elemento	elemento	elemento
elemento	elemento	elemento

Cuadro 2.1: Cuadro de prueba

---

### Definición de objetivos

---

Este proyecto tiene como objetivo segmentar correctamente las células mostradas en imágenes 3D mediante la aplicación de técnicas de machine learning.

Posteriormente se colorearán las células de distintos colores para facilitar su visualización.

Este proceso podrá ser reproducido por un usuario por medio de una imagen docker, con la cual el usuario podrá entrenar el modelo con un nuevo conjunto de imágenes o inferir la segmentación de una o más imágenes.

Se intentará que esta herramienta sea accesible al mayor n° de usuarios posibles, para ello se optimizará todo lo posible el proceso de entrenamiento e inferencia y se estudiarán varios modelos con distinto grado de complejidad. Con esto se reducirá el hardware necesario.

## Análisis de antecedentes y aportación realizada

### 4.1– Red Neuronal

#### 4.1.1. Perceptrón

El perceptrón es el elemento computacional básico de una red neuronal.

Fue propuesto inicialmente por Frank Rosenblatt en 1958 y perfeccionado por Minsky y Papert en la década de de 1960 (Rojas, 1996, p. 55-56). Su diseño está inspirado en las neuronas como bloque elemental en el funcionamiento del cerebro. En la figura 4.1 se puede ver una unidad de computación genérica con la que puede ser descrita un perceptrón.

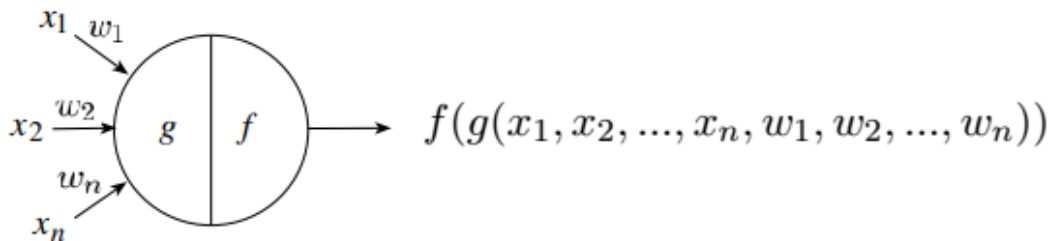


Figura 4.1: Unidad de computación genérica

Siendo:

- $(x_1, x_2, \dots, x_n)$  el vector de entrada.
- $(w_1, w_2, \dots, w_n)$  el vector de pesos.
- $g$  la función de integración, encargada de reducir el vector de entrada a un único valor.
- $f$  la función de activación, encargada de producir la salida de este elemento.

El perceptrón simple es una unidad de computación con un umbral  $\theta$  el cual, al recibir un vector de entrada de tamaño  $n$  y dado un vector de pesos de tamaño  $n$ , devuelve 1 si  $\sum_{i=1}^n w_i x_i \geq \theta$  y 0 en otro caso. (Rojas, 1996, p. 60).

Para entender de forma intuitiva esta definición podemos tomar un ejemplo en el que  $n = 2$ ,  $w_1 = 0,5$ ,  $w_2 = 2$ ,  $\theta = 1$ . En la figura 4.2 se puede ver la línea  $0,5x_1 + 2x_2 = 1$  y los puntos

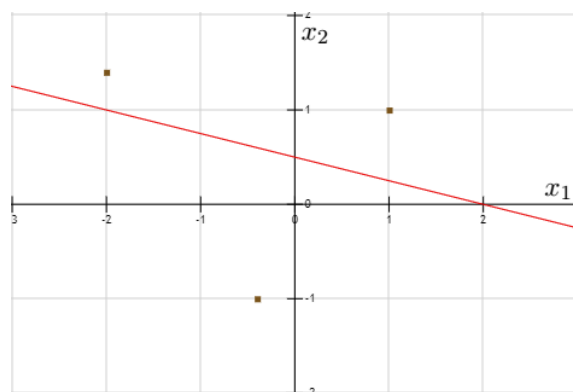


Figura 4.2: Representación de todos los valores del vector de entrada

$p_1 = (1; 1)$ ,  $p_2 = (-2; 1,4)$  y  $p_3 = (-0,4; -1)$ . Se puede ver cómo  $p_1$  y  $p_2$  están por encima de la línea, por lo que devuelven 1 y  $p_3$  por debajo, por lo que devuelve 0.

Se puede intuir cómo al modificar el valor de  $\theta$  la línea se mueve de forma vertical, así cómo  $x_1$  afecta a la pendiente y  $x_2$  afecta a ambos aspectos. Como se verá más adelante entrenar una red neuronal consiste en ir actualizando los pesos de los perceptrones, por lo que será importante representar un perceptrón de forma que se pueda modificar  $\theta$  al modificar un peso.

Para representar el perceptrón simple serán necesarios los siguientes cambios respecto a la unidad de computación genérica:

- Al vector de entradas se le añade un elemento de valor 1, siendo ahora de tamaño  $n + 1$ .
- Al vector de pesos se le añade un elemento de valor inicial  $-\theta$ , siendo ahora de tamaño  $n + 1$ . A este valor se le llamará *bias*.
- Se usará como función de integración el sumatorio.
- Se usará como función de activación la función escalón unitario.

$$1w_{n+1} \sum f(x_1w_1 + x_2w_2 + \dots + x_nw_n + 1(-\theta))f(g(x_1w_1, x_2w_2, \dots, x_nw_n))$$

#### 4.1.2. Partes de una Red Neuronal

#### 4.1.3. Algoritmos de aprendizaje

### 4.2– Red Neuronal Convolutiva

### 4.3– Reto ImageNet

### 4.4– Arquitectura U-Net

---

## Referencias

---

- Autores, V. (2014). *Escuela Técnica Superior de Ingeniería Informática*. ETSII. Descargado de <http://www.informatica.us.es> (fecha de consulta: 24 de Noviembre de 2014)
- Bezoz, J. (2007). The titlesec and titletoc packages. *TexEmplares*, 8, 283–298.
- de Sousa, J. M. (2004). *Ortografía y ortotipografía del español actual*. Trea.
- Rojas, R. (1996). *Neural networks: A systematic introduction* (Springer, Ed.). Springer-Verlag.  
Descargado de <https://page.mi.fu-berlin.de/rojas/neural/>