



**Universidad Autónoma Metropolitana
Unidad Iztapalapa**

**Proyecto terminal I
Investigación teórica (2111107)**

**Métodos de inteligencia artificial para el reconocimiento
de defectos topológicos en simulaciones de medio
continuo de cristales líquidos.**

**Autor: Huerta Escobar J. Gerardo.
Asesor: Orlando Guzmán López.**

Huerta Escobar J. Gerardo.

Orlando Guzmán López.

**Ciudad de México
Marzo-2021**

Introducción

Los cristales líquidos son un estado de agregación de la materia en donde se exhiben algunos comportamientos propios de las fases sólida y líquida. Por este motivo estas fases son nombradas como mesofases o fases intermedias y a los sistemas que presentan estas mesofases también se les denomina mesógenos. La transición de fase exhibida por distintos cristales líquidos puede ser inducida con la variación de temperatura, concentración o exposición a campos eléctricos, entre otras. Dependiendo del cristal líquido, en su estructura puede haber movimiento con restricciones específicas, por ejemplo, que exista movimiento rotativo, pero no traslacional o movimiento sólo en uno de los ejes coordenados. En este proyecto se trabajó con un sistema que presenta una fase nemática, en esta fase existen orientaciones definidas, pero no hay un orden a nivel posicional. La fase nemática puede ser caracterizada por el parámetro de orden de nematización S , cuando $S=0$, el sistema presenta una estructura isotrópica, mientras que, si $S \neq 0$, el sistema presenta un orden orientacional y anisotropía.

En la fase nemática existen defectos topológicos los cuales son disclinaciones, entendidas como defectos o cambios abruptos de orientación local en el eje director. Lo que aquí se estudió fueron fronteras de dominio en donde el parámetro de orden pasa abruptamente de $S = -1$ a $S = +1$, o viceversa. En general se sabe que la identificación de los defectos topológicos en cristales líquidos no es una tarea trivial, por lo que el intento de emplear herramientas novedosas para su detección sigue siendo un esfuerzo activo.

La búsqueda de nuevas tecnologías por parte de distintas disciplinas científicas incluyendo a la física en estas, ha puesto en alta relevancia a las técnicas de Inteligencia Artificial debido a su alta efectividad en tareas como la automatización. Debido al crecimiento tan acelerado en las técnicas informáticas contemporáneas, la implementación de estos recursos está creciendo en popularidad, evidencia de ello es la creciente publicación de diversos artículos científicos en donde se adoptan técnicas enfocadas en el uso de la Inteligencia Artificial en distintos problemas dentro de la física. Este proyecto es motivado por la alta efectividad que estas técnicas presentan, en particular, al implementarlas en la identificación de defectos topológicos en cristales líquidos en simulaciones de sistemas moleculares haciendo uso de técnicas de visualización computacional inteligente. En una búsqueda preliminar no se encontraron propuestas de aplicación de técnicas de Inteligencia Artificial para la detección de defectos topológicos en cristales líquidos en simulaciones de medio continuo. Es esta razón la que motiva el presente trabajo, en donde el objetivo es explorar la efectividad de estas técnicas para la identificación en medios continuos.

En la primera parte de este proyecto*, los esfuerzos fueron enfocados en el aprendizaje de distintas técnicas de Inteligencia Artificial y su implementación en la detección de objetivos específicos en simulaciones de medio continuo. En general abordar el problema de la detección de defectos topológicos, ya sea en experimentos o en sistemas de modelado, presenta una diversidad de retos, uno de ellos es el proponer una o varias técnicas que sean capaces de hacer frente a dicha tarea. Por lo que en general en este proyecto los esfuerzos fueron invertidos en ver qué tan bien funcionan en la práctica las técnicas de Inteligencia Artificial encontradas. La manera de probar la funcionalidad de estas redes neuronales fue su implementación en los primeros resultados para la evolución temporal de una mezcla nanopartícula-mesógeno obtenida mediante solución numérica de ecuaciones dinámicas de medio continuo [1].

El objetivo del presente proyecto es la detección de defectos topológicos generados en la simulación de la evolución temporal de la nematización $S(x)$, cuando el cristal líquido se describe

únicamente por su parámetro de orden de nematización escalar. Las regiones específicas a identificar aparecen en la Figura 1. donde se pueden apreciar los dominios con distintos valores de nematización en azul y en rojo. La detección se realizó exclusivamente en los dominios de fronteras cerradas, estas fronteras son generadas por los defectos topológicos. Ver figura 1.

La detección fue llevada a cabo mediante una Red Neuronal Convolutiva basadas en Regiones, específicamente *Mask Region Based Convolutional Neural Networks (RCNN)*, una red neuronal estructurada con la arquitectura del aprendizaje profundo (**Deep Learning**), enfocada en el uso de la segmentación como método de identificación diferenciada.

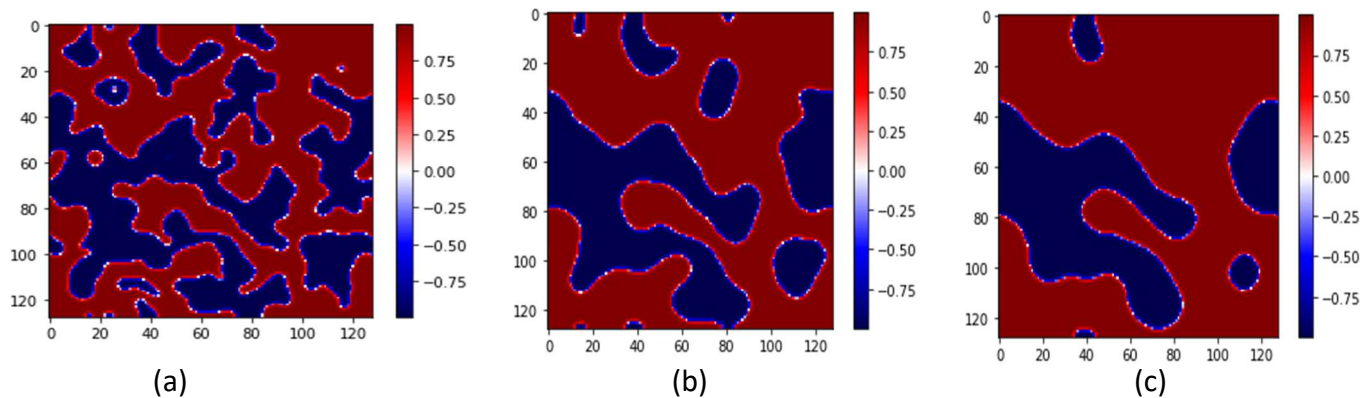


Figura 1. (a-c) Resultados obtenidos de la solución de ecuaciones dinámicas de medio continuo [1], para la evolución temporal de la nematización $S(x)$. En azul y rojo se observan dominios de nematización con valores negativos y positivos, respectivamente.

*En general el presente proyecto “Métodos de inteligencia artificial para el reconocimiento de defectos topológicos en simulaciones de medio continuo de cristales líquidos”, es desarrollado en una primera parte realizada en el presente Proyecto Terminal I y continuará su desarrollo en el Proyecto Terminal II.

Metodología

En el período de comparación de métodos propuestos para la detección de defectos topológicos en una simulación de medio continuo, se tomaron en cuenta distintas redes neuronales: YOLOv1, YOLOv2, YOLOv3, FeedForward (RNN) y Mask R-CNN. Debido a la alta eficiencia y adaptabilidad ante el cambio de la estructura del objeto de detección [16-18], se eligió a YOLOv3 y Mask R-CNN como las candidatas que más probablemente podrían lograr una detección efectiva en el conjunto de datos. Las dos técnicas elegidas, You Only Look Once *versión 3* (YOLOv3) y Mask R-CNN (R-CNN), están basadas en las librerías de OpenCV, TensorFlow y Keras, lo que nos permite aprovechar de manera eficiente a estas redes neuronales, ya que estas bibliotecas de componentes (*frameworks*) son de alta accesibilidad debido a que pertenecen al modelo de código abierto.

El empleo de redes neuronales obliga a seguir en orden estricto una serie de pasos, comenzando por la preparación del conjunto de datos, esta es la selección y anotación de los datos a analizar, distinguiendo aquellos que cumplen el criterio de selección de aquellos que no los cumplen, de este proceso se genera el conjunto de datos de entrenamiento que posteriormente se empleará en el entrenamiento de la red neuronal, también se generará el conjunto de datos de validación los cuales nos ayudarán a saber qué tan bien se realiza la detección. El siguiente paso es el entrenamiento de la red neuronal, en donde principalmente se eligen los parámetros específicos de la red para posteriormente ejecutar el entrenamiento, este entrenamiento consta de una cantidad específica de ciclos de entrenamiento (épocas), en donde se irán modificando los parámetros de la red que determinan la efectividad de la detección, en cada época se genera un modelo entrenado, típicamente se usa el último por ser el más entrenado. El producto final de la etapa de entrenamiento es una red neuronal entrenada y lista para la validación. El último paso es la validación, momento en el que se emplea el modelo obtenido para la detección de los objetivos en el conjunto de datos de validación.

Preparación del conjunto de datos

Se implementó el uso de un conjunto de datos, compuesto por 1000 imágenes secuenciadas, que representan la evolución simultánea del parámetro de orden S de un cristal líquido en el que se encuentran suspendidas nanopartículas con concentración C . Las nanopartículas se difunden lentamente en comparación con el tiempo de relajación del cristal líquido. Estas imágenes provienen de las simulaciones generadas en [1]. Se seleccionaron las imágenes, generando un conjunto de datos de entrenamiento de 800 imágenes en donde se etiquetó cada elemento de detección con la letra S , el conjunto de validación fue compuesto de las restantes 200 imágenes, cumpliendo así una relación entre conjuntos de datos de 80%-20%, tal como se recomienda en la bibliografía estudiada [11]-[15].

El etiquetado del conjunto de datos de entrenamiento para R-CNN fue realizado mediante el uso del software *VGG Image Annotator*, proyecto de código abierto. Generalmente las herramientas de etiquetado se eligen en función de la técnica de visión artificial empleada: es de especial interés el tipo de formato generado la salida del etiquetado, la velocidad de anotación y el sistema operativo del que se dispone. Para el etiquetado del conjunto de datos de YOLO se eligió *LabelImg*; esta herramienta fue especialmente creada para la detección de objetos. Así, se obtuvieron los tipos de archivos necesarios para cada red neuronal en formatos *JavaScript Object Notation* (.JSON) para RCNN y texto (.txt) para YOLO.

Implementación de Mask RCNN

Primero se revisarán algunos conceptos básicos sobre las redes neuronales. Las redes neuronales son un modelo computacional disruptivo, inspirado en el funcionamiento de las redes neuronales biológicas. La estructura de las redes neuronales artificiales (ver Figura 2) típicamente está formada por tres bloques de componentes. La primera parte es la capa de entrada, en donde se lee la configuración general de los datos de entrada. El segundo bloque son las capas ocultas en donde se lleva a cabo la mayoría del procesamiento de información por medio de las conexiones entre las unidades fundamentales de procesamiento o neuronas. La neurona es una función matemática que utiliza los valores de entrada para realizar una suma ponderada de estos valores, por lo que cada conexión que llega a la neurona tendrá asociado un valor que servirá para definir con qué intensidad cada variable de entrada afecta a la neurona, estos valores son denominados como pesos de la red, los pesos son los parámetros más relevantes de las redes neuronales. El último bloque es la capa de salida, en esta capa se obtiene de forma observable la información generada como respuesta en el resto de la red, esta información puede ser presentada, por ejemplo, como vectores o números. Entrenar una red neuronal consiste en ajustar los pesos de entrada para cada neurona, logrando así que la información generada en la capa de salida responda eficientemente a la tarea que se le plantea resolver, en este caso es la detección de objetivos específicos. Normalmente la cantidad de capas, tipo de neurona y las conexiones entre capa y capa son lo que determina la efectividad de la red neuronal. El aprendizaje profundo (*Deep Learning*), típicamente aprovecha todas estas modificaciones que se le pueden hacer a las redes neuronales para así obtener sistemas más eficientes. El uso de estas redes neuronales tan complejas ha logrado que el aprendizaje profundo realice tareas de detección con una alta efectividad (Figura 3). Una de las técnicas avanzadas que ha sido obtenida gracias al aprendizaje profundo es la segmentación de imágenes, que consiste en identificar regiones en una imagen y etiquetarlas en diferentes clases.

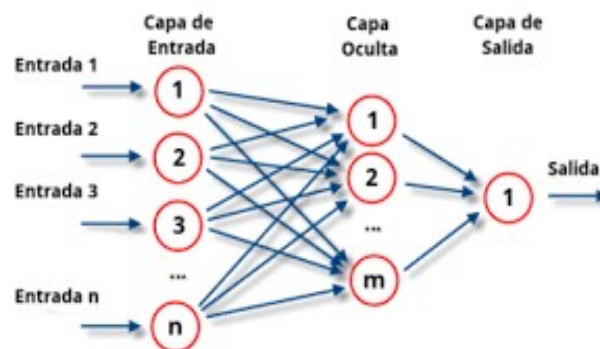


Figura 2. Estructura general de una red neuronal.

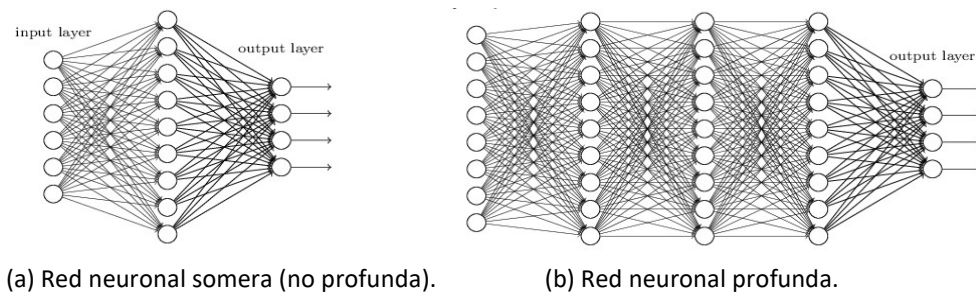


Figura 3. El aumento de capas ocultas, neuronas y las conexiones entre cada capa y neurona-neurona, generan sistemas más robustos, propios del aprendizaje profundo (b).

Mask RCNN es una Red Neuronal enfocada en la detección de imágenes mediante técnicas de segmentación basadas en la estructura propia del Aprendizaje Profundo (Deep Learning). Por ser una técnica basada en segmentación y sustentada por técnicas de Deep Learning, se tenía una fuerte sospecha de que esta herramienta podría ser útil a nuestro propósito. Específicamente, la propuesta de R-CNN como una técnica capaz de resolver el problema de detección, fue realizada después del análisis de su funcionamiento: está enfocado en aumentar el tamaño y la profundidad de la red neuronal, aumentar también su capacidad para aprender patrones y características más complejas contenidas en imágenes y luego reconocer estos patrones aprendidos en una nueva imagen. La RCNN, así como la mayoría de las redes neuronales convolucionales hacen uso de esta estrategia mediante el aumento del número de capas, el aumento de complejidad de estas capas y su entrelazamiento. En general estas características son las que la hacen una muy poderosa técnica para la identificación que se realizó. Mask RCNN trabaja en dos etapas: la primera escanea la imagen objetivo y genera propuestas, regiones de la imagen en donde probablemente haya un objeto a detectar. La segunda etapa clasifica las propuestas y genera cuadros delimitadores y etiquetas.

Para la programación de la red neuronal se contó con múltiple soporte teórico [3]-[10], además se implementó el uso de *coco-dataset* para los pesos de la red [2]. El ambiente de desarrollo fue soportado totalmente por Google Colab, de donde se adoptó el entorno de ejecución de GPU, para poder utilizar la GPU prestada por el equipo de Google en nuestra fase de entrenamiento y carga de paquetería, así como la fase de pruebas.

Implementación de YOLO

Como método alternativo a R-CNN, se implementaron esfuerzos en construir una red neuronal basada en YOLO, que es también un modelo de Aprendizaje Profundo. Primeramente, se implementaron redes totalmente funcionales de YOLOv1 y YOLOv2 que funcionaron insatisfactoriamente ya que las detecciones generadas no tenían una certeza de detección mayor al 40%, lo que hacía inviable este método. De ahí que se buscara otra opción, búsqueda que terminó con YOLOv3. Esta es una red neuronal convolucional y está caracterizada por tratar el problema de detección de imágenes con una velocidad de detección superior a la de las redes neuronales más simples, ya que el proceso de detección lo realiza sólo una vez en cada imagen. Debido a que procesa la imagen entera en vez de hacerlo por partes, esto le permite obtener una visión global de la información, resultando en una disminución de errores de falsas predicciones. YOLOv3 tiene en total 103 capas de arquitectura totalmente convolucional, esto explica la reducción de velocidad en

comparación con la segunda versión, que solo tiene 30 capas. Las redes neuronales convolucionales consisten principalmente en varias capas de filtros secuenciados ya que después de cada capa se añade una función para realizar un mapeo general.

Los problemas presentados por YOLO son la incompatibilidad del sistema operativo de mi PC (Windows) y del sistema para el que YOLO fue creada (Linux), esto hace especialmente difícil la implementación de librerías, que, finalmente terminaron frustrando el intento de uso de este método en la fase de entrenamiento. Se continúa trabajando en la implementación de YOLOv3 por varios frentes: cambio de sistema operativo y el uso de paqueterías más fácilmente adaptables. Se tiene que hacer hincapié en que este es un método que todavía consideramos posiblemente útil, por lo que se seguirán destinando esfuerzos para lograr resultados positivos.

Resultados

Mask RCNN

-El tiempo promedio de entrenamiento para 10 épocas (*epochs*) fue aproximadamente de 72 segundos. Ver Figura 4.

```
500/500 [=====] - ETA: 0s - batch: 249.5000 - size: 1
  warnings.warn("`Model.state_updates` will be removed in a future version. '
500/500 [=====] - 90s 156ms/step - batch: 249.5000 -
Epoch 2/10
500/500 [=====] - 72s 145ms/step - batch: 249.5000 -
Epoch 3/10
500/500 [=====] - 72s 144ms/step - batch: 249.5000 -
Epoch 4/10
500/500 [=====] - 72s 145ms/step - batch: 249.5000 -
Epoch 5/10
500/500 [=====] - 72s 144ms/step - batch: 249.5000 -
Epoch 6/10
500/500 [=====] - 72s 144ms/step - batch: 249.5000 -
Epoch 7/10
500/500 [=====] - 72s 144ms/step - batch: 249.5000 -
Epoch 8/10
500/500 [=====] - 72s 144ms/step - batch: 249.5000 -
Epoch 9/10
500/500 [=====] - 72s 144ms/step - batch: 249.5000 -
Epoch 10/10
500/500 [=====] - 72s 144ms/step - batch: 249.5000 -
```

Figura 4: En la segunda columna se presenta el tiempo requerido para efectuar cada época de entrenamiento desde la primera hasta la décima.

El hecho de que se obtengan estos valores en la duración de cada etapa de entrenamiento para esta red neuronal extiende la posibilidad de emplearla en sistemas más complejos en donde para responder a las exigencias de detección de estos sistemas la red tenga que pasar por una cantidad significativamente mayor de etapas de entrenamiento.

-El éxito de detección para las 200 imágenes de validación está en 99%. La probabilidad promedio del error de detección dada por el algoritmo es de 0.001.

La detección que se presenta en la Figura 5, corresponde a la detección de los objetivos en la imagen 258 del conjunto de datos general. El sistema nunca había interactuado con esta imagen, pero logró una detección significativa. Este mismo procedimiento se corrió para el resto de las imágenes del Conjunto de datos de validación, obteniendo el mismo tipo de resultado favorable en un margen de detección idéntico.

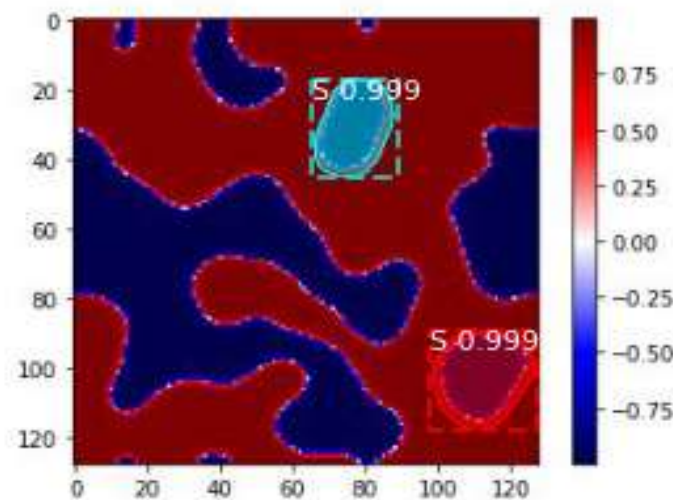


Figura 5: Imagen 258 del Conjunto de datos general.

Se pueden apreciar las etiquetas de detección **S** junto con su probabilidad de detección asociada.

Es de especial relevancia el sistema de detección, basado en segmentación. Porque además de ofrecer un recuadro de detección, segmenta el espacio en donde se encuentra el objetivo detectado. Esto nos abre la posibilidad de extender esta técnica a sistemas en donde la línea que demarca un objeto a detectar sea especialmente difusa respecto a la de otro objeto de detección, así como también la detección y clasificación de distintas categorías dentro del sistema.

YOLO

Debido a los problemas encontrados a la hora de efectuar la implementación de YOLO aún no se han obtenido resultados satisfactorios, sólo se demostró que las versiones 2 y 3 de YOLO no son capaces de resolver nuestro problema de detección.

Conclusiones

- El resultado obtenido en la búsqueda de una técnica de inteligencia artificial para la detección de defectos topológicos en cristales líquidos fue exitoso: se seleccionaron dos métodos de aprendizaje automático de los cuales uno dio resultados satisfactorios.
- Contamos con suficiente evidencia de que Mask RCNN presenta la suficiente robustez frente al problema de detección presentado. Esto abre la posibilidad de continuar con nuestro objetivo de detectar defectos topológicos en simulaciones de medio continuo de cristales líquidos.
- La implementación de YOLO en el análisis de nuestro sistema aún está en período de ajuste, específicamente la parte del entrenamiento. Aún se tiene que resolver la compatibilidad del sistema operativo con esta Red Neuronal.

Referencias

- [1] F. Gael Segura-Fernández, Erick F. Serrato-García, J. Emmanuel Flores-Calderón [y] Orlando Guzmán López. Dynamics of nanoparticle self-assembly on 2D by liquid crystal sorting. Departamento de Física, Universidad Autónoma Metropolitana. Preprint.
- [2] Tsung-Yi, Lin Michael, Maire Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick [y]Piotr Dollar. Microsoft COCO: Common Objects in Context. arXiv: 1405.0312v3 [cs.CV] 21 Feb 2015. (Link de descarga: <https://arxiv.org/pdf/1405.0312.pdf>)
- [3] Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1974). The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, Massachusetts.
- [4] Weiss, I., and Bandyopadhyay, A. (1988). Active vision. International Journal of Computer Vision, 1:333-356.
- [5] Walters, M., Wei, Q. & Chen, J.Z. Y. Machine learning topological defects of confined liquid crystals in two dimensions. Physical Review E 99, 062701 (2019).
- [6] Aplicaciones de la Inteligencia Artificial en la Actividad Empresarial la Ciencia y la industria: fundamentos Aplicaciones/ por Wendy B.Rauch Hindin.
- [7] Redes neuronales artificiales: Fundamentos, modelos y aplicaciones/por José Ramón Hilera Gonzalez [y] Víctor José Martínez Hernando.

[8] Inteligencia artificial: Un enfoque moderno/por Stuart J. Russell; Peter Norvig [y] Traducción de Raul Bautista Gutiérrez.

[9] <https://github.com/owainlewis/awesome-artificial-intelligence>

[10] <https://github.com/TarrySingh/Artificial-Intelligence-Deep-Learning-Machine-Learning-Tutorials>

[11] Walters, M., Wei, Q. & Chen, J.Z. Y. Machine learning topological defects of confined liquid crystals in two dimensions. *Physical Review E* 99, 062701 (2019).

[12] Doi H., Takahashi K., Tagashira K., Fukuda J. & Aoyagi A. Machine learning-aided analysis for complex local structure of liquid crystal polymers. *Scientific Reports*. 9,16370 (2019).

[13] Rodriguez-Nieva J. F. & Scheurer M. S. Identifying topological order through unsupervised machine learning. *arXiv:1805.05961v2 [cond-mat.stat-mech]* (2020).

[14] Ruoshi Liu, Michael M. Norton, Seth Fraden, Pengyu Hong. Detecting topological defects in 2D active nematics using convolutional neural networks. *Sometido a International Conference on Learning Representations*, (2019), <https://openreview.net/pdf?id=HklVTi09tm>.

[15] M. Durante, S. Tlili, T. Homan, B. Guirao, F. Graner & H. Delanoë-Ayari. Fast determination of coarse-grained cell anisotropy and size in epithelial tissue images using Fourier transform. *Physical Review E* 99, 062401 (2019).

[16] Edgar Roberto Silva Guzmán, A. (2020). Detección de objetos con redes neuronales profundas para un robot de servicio (Tesis de Maestría). Universidad Nacional Autónoma de México, Ciudad Universitaria, CD. MX.

[17] Redmon, J., Farhadi, A. (2018). YOLO9000: An Incremental Improvement. <https://arxiv.org/pdf/1804.02767.pdf>

[18] Alejandro Torres Alonso, A. (2020). Detección de frutas en árboles (Tesis de grado). Universidad Complutense de Madrid. Madrid, España.