



**Universidad Autónoma Metropolitana Unidad
Iztapalapa**

**Proyecto terminal II
Investigación teórica (2111109)**

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

J. Gerardo Huerta Escobar.

Asesor: Dr. Orlando Guzmán López.



J. Gerardo Huerta Escobar.



Dr. Orlando Guzmán López.

14 de junio de 2021

Introducción

El presente proyecto es la continuación del proyecto terminal I presentado el trimestre 20-O titulado: Métodos de inteligencia artificial para el reconocimiento de defectos topológicos en simulaciones de medio continuo de cristales líquidos. Parte I. En esa primera parte se propuso un modelo basado en redes neuronales artificiales para el reconocimiento de defectos topológicos extendidos, en dos dimensiones, en una simulación de medio continuo de cristales líquidos generada en [1]. En esta parte II se amplió el modelo neuronal utilizado en la parte I, lo que permitió realizar una detección de defectos topológicos puntuales, también en una simulación de medio continuo de cristales líquidos pero esta vez tomando en cuenta su carga topológica $+1/2$ o $-1/2$. En cada defecto, hay una carga topológica asociada, definida como el número de veces que el campo vectorial se enrolla alrededor de un bucle que rodea el defecto en sentido contrario a las agujas del reloj. La detección de los defectos topológicos se hizo en la simulación de la evolución temporal de la orientación del director $\theta(\mathbf{x})$, para el parámetro de movilidad relativo entre las nanopartículas y el cristal líquido, $R = 0,1$, en una mezcla de nanopartícula-mesógeno obtenida mediante solución numérica de ecuaciones dinámicas de medio continuo [1]. Posteriormente, por medio de los datos obtenidos del modelo, se analizó la concentración de los defectos topológicos en función del tiempo.

En los cristales líquidos existen defectos topológicos los cuales son disclinaciones, estas son entendidas como cambios abruptos en la orientación local del eje director. Lo que aquí se estudió fueron disclinaciones puntuales, es decir, defectos topológicos aislados en donde la orientación del director $\theta(\mathbf{X})$ cambia abruptamente. En el campo $\theta(\mathbf{X})$, se pueden diferenciar los cambios abruptos de orientación del director, lo que en este se campo posibilita la identificación de los defectos topológicos puntuales y su carga asociada. En el conjunto de datos analizado, se puede identificar un defecto topológico por medio de ese cambio abrupto en el valor de la orientación del director $\theta(\mathbf{x})$. Cuando el director tiene un valor creciente en la escala de colores (ver figura 1), se le puede asociar con una carga topológica de $+1/2$, de otro modo, el director tendrá un valor decreciente en la escala de colores, por lo que se le podrá asociar con una carga topológica de $-1/2$.

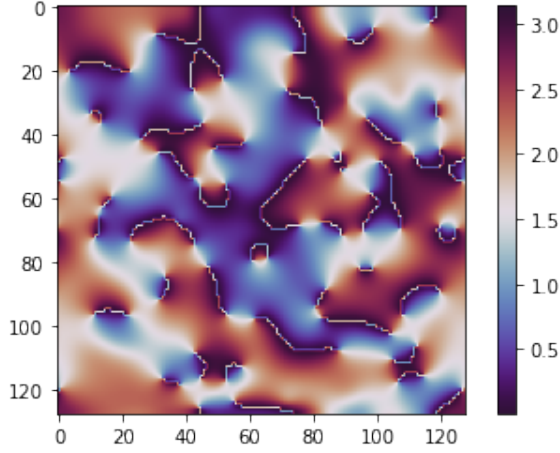


Figure 1: Imagen número 69 del conjunto de datos generado en [1].

El modelo empleado está basado en el mismo tipo de red neuronal convolucional, *Mask Region Based Convolutional Neural Networks* (**RCNN**), utilizada en la parte I del presente proyecto. En esta parte II se amplió el número de clases a identificar. Las clases se entienden como una característica específica e identificable de un objetivo de detección: para este proyecto, el objetivo de detección es un defecto topológico y las dos clases son las cargas topológicas $+1/2$ y $-1/2$.

Metodología

La motivación de utilizar la misma red neuronal que se utilizó en la parte I del proyecto, fue que ese modelo tuvo éxito en la detección propuesta y parte de la metodología inicial es explorar los límites del modelo frente a dificultades que pueden aparecer en distintos escenarios de detección. En esta parte II se continúa con el mismo enfoque: mejorar el modelo estudiado y extenderlo para realizar una identificación de características en sistemas más complejos. La propuesta de una red neuronal como modelo de detección, obliga a seguir una serie de pasos estrictos, debido a que una vez elegido el tipo de algoritmo que se va a emplear, se establece una metodología específica.

Generalmente se comienza con la preparación del conjunto de datos, esta parte consiste en realizar una selección y anotación en los datos a analizar. Es aquí cuando se distinguen aquellos datos que cumplen con el criterio de selección de los que no lo cumplen y en su caso, diferenciar las características de cada objetivo seleccionado. Es en este proceso en el que se genera el conjunto de datos que se utilizarán en el entrenamiento, es decir, el conjunto de entrenamiento. Al mismo tiempo, se tiene que seleccionar el conjunto de validación, estos datos nos servirán para evaluar el rendimiento del modelo después del entrenamiento.

Naturalmente, el siguiente paso es el entrenamiento del modelo, el entrenamiento está constituido de una cantidad específica de épocas o ciclos de entrenamiento. Después de cada ciclo de entrenamiento se obtiene un modelo entrenado: típicamente se elige el modelo asociado a la última etapa de entrenamiento para evaluarlo en la fase de validación. La validación es la última fase del entrenamiento, y es aquí en donde se evalúa el rendimiento del modelo sobre el conjunto de datos de validación anteriormente construido. La condición que debe cumplir este proceso de validación es que los datos que conforman el conjunto de validación, tienen que ser tales que la red neuronal no haya interactuado con ellos de alguna manera, es decir, que no hayan formado parte del conjunto de entrenamiento.

La concentración de las nanopartículas está dado por el campo $C(\mathbf{X})$. Inicialmente se intentó emplear un conjunto de datos constituido por los resultados de una simulación realizada en [1] para el campo $C(\mathbf{X})$ (ver figura 2 (a)), para el cual no se pudo realizar la identificación de las cargas topológicas debido al bajo contraste en los objetivos de detección (es decir, debido a que la concentración de nanopartículas en los defectos topológicos no es suficientemente alta comparada con aquella en las regiones donde no hay defectos topológicos).

El segundo conjunto de datos a analizar fue el campo de la evolución temporal de la nematización $S(\mathbf{X})$ (ver figura 2 (b)). Por medio del campo $\theta(\mathbf{X})$ se identificó la carga topológica y pudo realizarse el etiquetado. Una prueba del modelo obtenido para este conjunto de datos mostró que no se encontró un patrón diferenciable en el campo $S(\mathbf{X})$ que identificara las cargas topológicas de los defectos topológicos.

El último conjunto de datos que se consideró fue el campo $\theta(\mathbf{X})$, del cual se prepararon 100 imágenes. Entre éstas, el 80% se utilizó para el entrenamiento y el 20% restante se utilizó en la etapa de validación, tal como se recomienda en [2]-[5]. Para el conjunto de entrenamiento y validación se realizó el etiquetado mediante el uso del software VGG Image Annotator, versión 2.0.0, un software de código abierto. La implementación de esta herramienta fue debido a los requerimientos de los datos de entrada que la configuración del algoritmo necesita: un archivo con formato JavaScript Object Notation (.JSON).

Debido a las limitaciones computacionales locales (es decir, que todo el análisis se tenía que hacer con una laptop, no con estaciones de trabajo debido a que no se tenía acceso a ese equipo de cómputo en la Unidad Iztapalapa) se utilizó Google Colab como ambiente de desarrollo. Colab permite usar una de las máquinas virtuales de Google para llevar a cabo tareas de aprendizaje automático y crear modelos sin preocuparse por las limitaciones de potencia informática en un entorno local. Con la versión gratuita que ofrece Colab para el tipo de entorno de ejecución de GPU fue suficiente para la realización este proyecto.

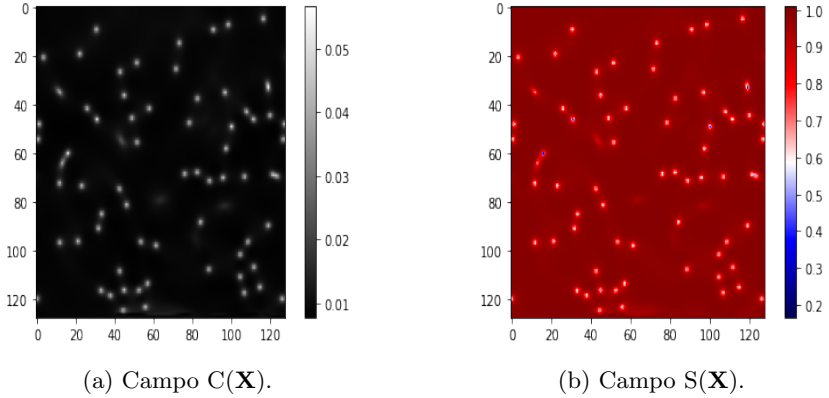


Figure 2: Para cada uno de estos dos conjuntos de datos se intentó utilizar el modelo neuronal. Las limitaciones en (a) fueron debidas al contraste y a la escala de grises. La red neuronal propuesta no fue pensada originalmente para realizar detecciones en imágenes en escala de grises; las imágenes sobre las que se realiza la detección deben de tener un modo de color RGB, esto y el bajo contraste para realizar el etiquetado es una de las limitantes encontradas en el método utilizado. En (b), el modelo aprendió a identificar los defectos topológicos, pero no logró encontrar un patrón que los diferenciara por medio de su carga topológica.

Implementación de la red neuronal.

La red neuronal utilizada en este proyecto es una red neuronal convolucional basada en regiones (R-CNN). pertenece a una familia de modelos de aprendizaje automático enfocados en la visión por computadora y están diseñados específicamente para realizar detección de objetos. Mask R-CNN está basada en el paradigma de aprendizaje supervisado, éste es un tipo de aprendizaje que se basa en descubrir la relación existente entre una variable de entrada y una variable de salida. El aprendizaje surge de enseñarle a estos algoritmos lo que se espera de un determinado valor de entrada. Tras un entrenamiento con un gran número de ejemplos, el algoritmo será capaz de dar un resultado correcto incluso para valores que no haya visto antes.

La red está 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 presente 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 hace 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.

La preparación del conjunto de datos de entrenamiento estuvo constituida aproximadamente por el 50% del conjunto de datos original, debido al ruido que hay en las imágenes iniciales (1-30), en donde no es posible identificar efecivamente los defectos topológicos y sus cargas asociadas. El 20% de las imágenes restantes se utilizaron para conformar el conjunto de validación. Ver figura 3.

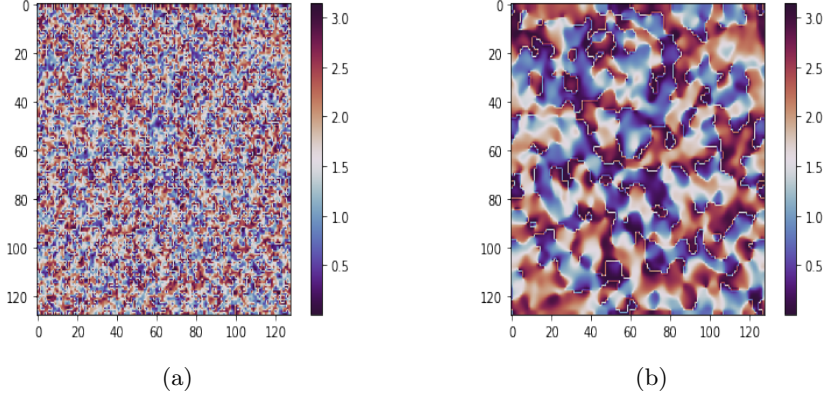


Figure 3: Imagen 1 (a) y 10 (b). Hasta la imagen 30 los defectos topológicos y sus cargas eran difíciles de etiquetar. Se sabe que al momento de entrenar una red neuronal es necesario preparar los datos de entrenamiento de una manera cuidadosa debido a que al administrar información errónea al modelo, este tomará en consideración esta información en su etapa de entrenamiento y posteriormente en la etapa de validación, lo que resultará en una eficiencia menor en la detección. Los datos para el conjunto de entrenamiento se etiquetaron de la imagen 31 a la 80. La validación se realizó de la imagen 81 a la 100.

Para la detección realizada en este trabajo, a diferencia de la versión del modelo utilizado en la parte I del proyecto, se habilitó en la red el modo de detección de dos clases, las cuales fueron las cargas $+1/2$ y $-1/2$. El modelo fue evaluado usando varios números de etapas de entrenamiento, siendo la etapa 10 la que se eligió para la detección final debido a la eficiencia registrada y al tiempo computacional requerido.

Resultados

El tiempo promedio de entrenamiento para las 10 épocas fue aproximadamente de 158 segundos. Ver Figura 4.

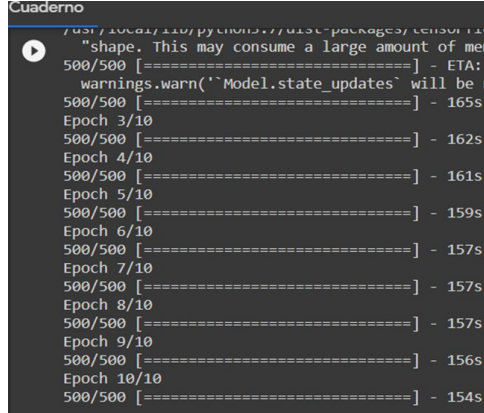
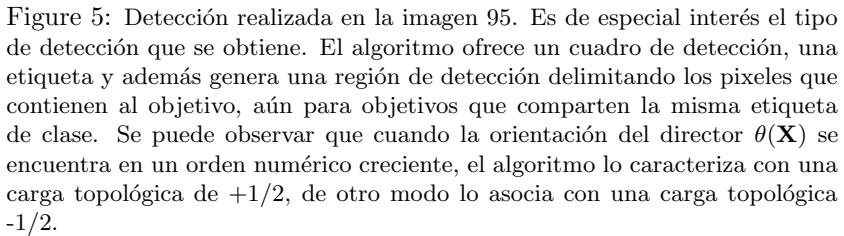


Figure 4: Tiempo requerido para cada época de entrenamiento realizado en Colab, desde la primera hasta la décima.

La detección realizada está representada por la Figura 5, que corresponde a la detección en la imagen 95. La detección también se realizó para el resto de las imágenes del conjunto de validación, obteniendo el mismo tipo de resultado. El éxito de detección promediado para las 20 imágenes de validación fue mayor al 92%. El valor mínimo de la probabilidad de detección impuesto fue de 0.6.

Una vez comprobado que la detección era correcta, para obtener una visión más completa del comportamiento de la concentración de los defectos topológicos en el tiempo, la detección también se realizó sobre el conjunto de entrenamiento, exceptuando las imágenes 1 a 30.



10

Posteriormente, se realizó un análisis de los datos generados en las detecciones, de donde se pudo extraer el número de defectos topológicos en el tiempo $N(t)$. Ese número fue dividido por el área total de detección ($A = 217^2$) para obtener la densidad de defectos y se graficó a esa densidad con respecto al número de pasos de la simulación, ver la figura 7. Se pueden explicar las 3 regiones de la curva de la manera siguiente: al principio ($t=30$ a $t=50$), el algoritmo no detectaba demasiados defectos debido al ruido en las imágenes, ese ruido iba disminuyendo a medida que pasaba el tiempo por lo que el algoritmo iba detectando más objetivos. Después de esa etapa, se ve que el numero de defectos es constante en el tiempo ($t=50$ a $t=80$), eso debido a la aniquilación de los defectos pero también a que la maquina empezó a contar nuevos defectos que se iban definiendo mejor. En la última etapa de la curva es notorio que la cantidad de defectos por imagen empieza a disminuir, es en esa región ($t=80$ a $t=100$), en donde los defectos ya estan muy bien definidos, por lo que el algoritmo fue capaz de realizar una detección con menos perturbaciones debidas al ruido.

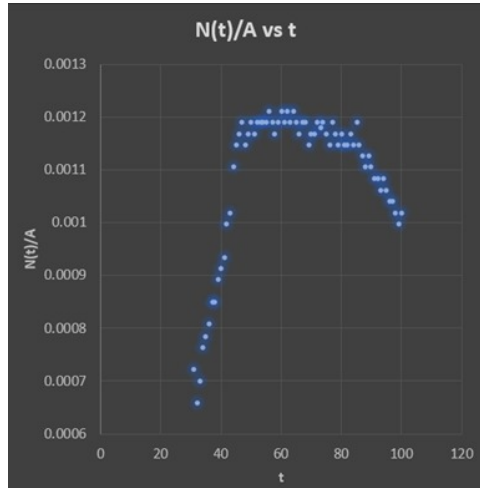


Figure 7: Concentración de los defectos por imagen respecto al tiempo.

Debido a lo anterior, se analizaron los datos en la última región de la curva. La pendiente resultante de los datos en escala logarítmica fue de -0.821. Ver figura 8.

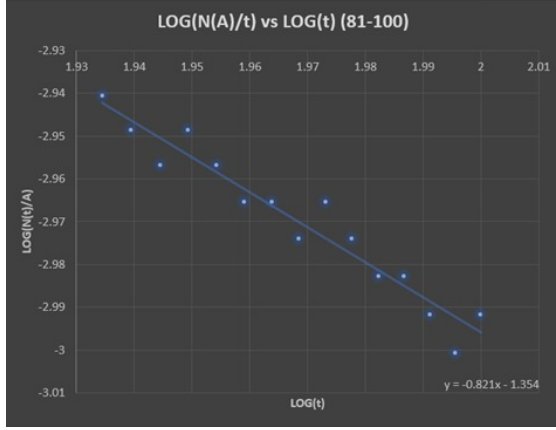


Figure 8: Los datos ajustados a un modelo lineal corresponden a la cantidad de defectos topológicos detectados en los pasos 81 a 100.

Una vez obtenida la densidad de los defectos se pueden sacar conclusiones sobre su dinámica. Para ello, se considera la distribución de defectos mediante la densidad por unidad de área $\rho(t) = N(t)/A$, y se analiza estadísticamente su escalamiento con el tiempo. La dinámica procede como una secuencia de pasos de aniquilación individuales de defectos puntuales. A partir de argumentos de escala, considerando la tasa de disipación de energía y la energía de defecto de una red de defectos en un nemático, se predijo una disminución de la densidad de defectos según $\rho(t) \propto t^{-1}$ en 2D [6]. Lo que, en una escala logarítmica, para los datos de $\rho(t)$, significaría una pendiente de -1 en la ecuación de su ajuste lineal. La pendiente obtenida para el ajuste de los datos analizados en este proyecto es cercana al valor teórico esperado reportado en [6], lo cual es esperable dado que la simulación no fue muy larga.

Conclusiones

Para un mejor estudio de la concentración de los defectos topológicos en el tiempo, es necesario analizar más cantidad de pasos de simulación: en tal caso, la detección se realizaría en regiones con cada vez menor ruido, por lo que el algoritmo identificará los objetivos con una mejor probabilidad de detección.

El valor obtenido del análisis de los datos para la pendiente es cercano al valor teórico esperado y experimentalmente confirmado, según [6]. Lo que indica que con el modelo neuronal empleado se pueden obtener datos cercanos a los reportados en la literatura de una manera eficiente.

Bibliografía

[1] Segura-Fernández FG, Serrato-García EF, Flores-Calderón JE and Guzmán O (2021) Dynamics of Nanoparticle Self-Assembly by Liquid Crystal Sorting in Two Dimensions. *Front. Phys.* 9:636288. doi: 10.3389/fphy.2021.636288

[2] 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).

[3] 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).

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

[5] 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>.

[6] Harth K and Stannarius R (2020) Topological Point Defects of Liquid Crystals in Quasi-Two-Dimensional Geometries. *Front. Phys.* 8:112. doi: 10.3389/fphy.2020.00112

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick: “Mask R-CNN”, 2017; [<http://arxiv.org/abs/1703.06870> arXiv:1703.06870].