

Detección de pozos en placas de ensayos mediante redes neuronales y transferencia de conocimiento

Emmanuel Rosa Delgado, Michael Rivera¹ Lazú, José O. Sotero Esteva*

Departamento de Matemáticas, Universidad de Puerto Rico en Humacao, Humacao, Puerto Rico

{emmanuel.rosa2, michael.rivera, jose.sotero}@upr.edu

*ORCID 0000-0001-9508-3766

Resumen—El uso de aparatos móviles para el análisis colorimétrico de ensayos químicos en contextos fuera de laboratorios es un área de continua gestión. En este contexto las plataformas en las que se ejecutan estos ensayos suelen estar hechos de materiales novedos, de bajo costo y ambientalmente amigables. Esto introduce dificultades adicionales ya que estos materiales pueden mostrar deformaciones y ser usados bajo iluminaciones de tipo variado. Este trabajo aborda la detección de pozos en placas de ensayos de 96 pozos hechas de materiales de que sujetos deformaciones. El método utilizado aprovecha la detección parcial usando transformadas de Hough y la aplicación de redes neuronales. El uso de transferencia de conocimiento hace posible que el conjunto de entrenamiento necesario sea pequeño y la intervención humana en su etiquetado sea mínima. Se muestra además un método para la evaluación sistemática de la efectividad de la técnica aplicada.

Index Terms—Análisis colorimétrico, transformada de Hough, redes neuronales

I. INTRODUCCIÓN

La literatura menciona con frecuencia creciente el uso de teléfonos móviles junto con plataformas innovadoras para realizar análisis fuera del entorno de laboratorio [1]–[3]. Ejemplos de ello son las aplicaciones móviles que facilitan los análisis preliminares de pruebas químicas en placas de 96 pozos, eliminando la necesidad de utilizar equipos costosos y voluminosos fuera del laboratorio. Un caso documentado es la aplicación *Spotxel® Microplate Reader* [4]. Esta y otras aplicaciones similares presentan en la pantalla del aparato una plantilla visual de la placa que el usuario debe alinear con la placa que se va a analizar. Este mecanismo asume que la placa a analizar es rígida, estándar y requiere destreza por parte del usuario.

En este estudio, implementamos técnicas que pueden permitir que aplicaciones de este tipo operen sin estos requisitos. Estas se basan en el uso de métodos de procesamiento de imágenes que han demostrado ser efectivos para este tipo de problema, con el objetivo de generar conjuntos de entrenamiento para una red neuronal con mínima intervención humana. Durante el entrenamiento se aprovecha, a su vez, la técnica de transferencia de conocimiento desde una red que ha sido entrenada para otros fines, lo que permite que su entrenamiento converja con un esfuerzo computacional relativamente bajo.

Este trabajo ha sido financiado por el programa PENN-UPR Partnerships for Education and Research in Materials bajo el auspicio de la Fundación Nacional de Ciencias de los EE.UU. (NSF-DMR-2122102).

I-A. Trasfondo

Dado que estas aplicaciones están diseñadas para operar mayormente fuera de un entorno de laboratorio formal, es preferible que las plataformas en las que se realizan estos ensayos estén fabricadas con materiales innovadores, de bajo costo y respetuosos con el medio ambiente, como las placas de ensayo de 96 pozos hechas de acetato de celulosa [5]. Estos materiales presentan nuevos retos para efectuar colorimetría efectiva, entre otras razones, por ser materiales inherentemente flexibles y deformables.

En la última década y media hemos sido testigos de una explosión de aplicaciones de inteligencia artificial, muchas de ellas basadas en redes neuronales. Cuando se aplican redes neuronales para analizar problemas científicos, los investigadores se encuentran inicialmente con dos desafíos fundamentales: diseñar una red que se ajuste a la aplicación y generar un conjunto de entrenamiento de alta calidad y amplio, con miles o incluso decenas de miles de ejemplos etiquetados manualmente. Además, se requiere equipo informático sofisticado y costoso para llevar a cabo el entrenamiento. Sin embargo, recientemente se han presentado ejemplos del uso de la transferencia de conocimiento, la adaptación de las redes neuronales y la transferencia de la información almacenada de una red a otra, que mitigan estos obstáculos.

En un trabajo anterior presentamos un método de detección de pozos en placas de 96 pozos con deformaciones basado en el método de Hough [6]. En una de sus formulaciones, el método de Hough sirve para detectar círculos en imágenes que han sido preprocesadas para detectar bordes de figuras. Efectos naturales en la imagen imposibilitan la detección precisa de bordes. El método de Hough provee para ajustar la tolerancia a estos defectos. Pero un sólo conjunto de parámetros no produce las clasificaciones esperadas en todas las imágenes. En el caso de la detección de pozos de placas de ensayo la ausencia de círculos, repeticiones, o detección de círculos espurios son problemas comunes. En nuestro trabajo previo comenzamos la detección de los pozos con parámetros elegidos de manera deliberada para subestimar significativamente la detección de círculos. Luego se itera el procedimiento incrementando la sensibilidad del método hasta que se obtiene al menos una fila de 12 pozos y una columna de 8 pozos detectados. Los pozos restantes se obtienen usando interpolación.

El mérito de el presente estudio radica en la demostración

de un método que genera un conjunto de entrenamiento para una red neuronal que identifica pozos en placas de ensayo con mínima intervención humana, basándose en técnicas estándar de procesamiento de imágenes. Además, se utiliza la transferencia de conocimiento para alcanzar altos niveles de precisión en la red neuronal con un conjunto de entrenamiento pequeño, y se emplea un método de medición de la precisión de los resultados altamente automatizado y confiable. Esta estrategia puede ser aplicada a otros problemas similares con ajustes mínimos.

II. MÉTODOS, SOFTWARE Y EQUIPO

II-A. Insumo

El insumo del sistema consiste de imágenes en formato RGB de placas de ensayos de 96 pozos estándar de colores blanco o negro bajo las fuentes de iluminación solar indirecta, flash, y lámparas LED y fluorescente. Imágenes adicionales iluminadas con luz ultravioleta son tomadas con placas blancas solamente. Dado que en fases posteriores de este proyecto será el desarrollo de aplicaciones para teléfonos móviles las fotos fueron tomadas con ese tipo de aparatos. Cada imagen termina representada como un arreglo de tres dimensiones $I(i, j, c)$ en el cual cada entrada está asociada a un pixel donde i, j son números enteros que representan la fila y la columna del pixel y $c = 0, 1, 2$ corresponden a las intensidades de rojo, verde y azul R, G, B enteros entre 0 y 255.

II-B. Datos para Entrenamiento

El conjunto de datos para el entrenamiento consiste de pares de imágenes I_n con su respectivo etiquetado. El etiquetado que describe las aperturas de los pozos de la placa consiste de una imagen en tonos de grises representada como una matriz de dos dimensiones $P_n(i, j)$ que almacena p_n valores distintos, uno por cada pozo, según se ilustra en la Figura 1.

II-B1. Imágenes para el conjunto de datos: El objetivo es que el conjunto de datos para el entrenamiento tenga imágenes de placas recortadas y deformadas como muestra la foto izquierda de la Figura 1. Inicialmente se selecciona una imagen representativa de cada una de las variantes descritas en la sección II-A para un total de nueve fotos. Antes de ser incluidas en el conjunto de entrenamiento las imágenes crudas son pasadas por un filtro gaussiano multidimensional [7] para la ecualización de brillantez que reduce el ruido y facilita la detección de los bordes de la placa que son localizados con un filtro tipo Sobel [8]. Con estos bordes detectados la imagen original sin filtros es recortada para remover trasfondo innecesario quedando sólo las placas. en este punto se ejecuta una transformación de perspectiva para corregir la deformación trapezoidal.

Este conjunto inicial de imágenes es aumentado mediante un proceso de deformación controlada en distintos grados. Para lograr este efecto se definió un mapa en el que la i -ésima fila es proyectada a una parábola usando la transformación

$$(i, j) \rightarrow (i, -\frac{k}{h^2}((x - h))^2 + k) \quad (1)$$

donde (h, k) son las coordenadas del vértice de la parábola, $h = M/2$, y $k = 0, 20, 40, 60, \dots, 200$ representa la cantidad de pixeles por encima de la horizontal del vértice. Se seleccionó la forma de una parábola por su similitud al efecto producido por apretar por los bordes una lámina flexible. Pero el método es adaptable a cualquier transformación definible con funciones de uso común.

En este punto el conjunto inicial de nueve imágenes se ha expandido a 99 imágenes.

II-B2. Etiquetado de las imágenes: Las imágenes pre-procesadas son sometidas al proceso de detección de círculos usando la transformada de Hough variando el parámetro umbral de acumulación incrementando paulatinamente la sensibilidad del método hasta que se consigue una detectar una porción de los 96 pozos. Se hace notar que en la mayoría de los casos esto no identifica la totalidad de los pozos. En este punto se producen las imágenes objetivo, el conjunto P_n descrito arriba, en escala de grises con negro representando trasfondo sin pozos y áreas con distintos grados de gris para cada pozo detectado como se muestra en la Figura 1.

II-C. Red Neuronal

La red neuronal se construye basada en la redes neuronales de tipos Faster-RCNN [9] y Mark-RCNN [10] provistas por las funciones FastRCNNPredictor y MaskRCNNPredictor del paquete TorchVision [11]. Esta red viene pre-entrenada con el conjunto de entrenamiento COCO-v1 [12] para la detección de diversos objetos. El uso de las redes con pesos pre-establecidos consigue una transferencia de conocimiento a nuestra red. Esta red es sometida a entrenamiento adicional con el conjunto de entrenamiento descrito en la sección anterior. Los nuevos pesos de la red son almacenados para uso posterior.

El conjunto de datos para entrenamiento es dividido en un subconjunto de entrenamiento de 50 imágenes etiquetadas y el de prueba de 49 imágenes etiquetadas. El método de optimización utilizado fue el descenso de gradiente estocástico con momentum con tasa de aprendizaje inicial de 0.005 decreciendo por 10 % cada tres pasos.

Mediciones de la pérdida en los conjuntos de entrenamiento y de prueba acopiados durante el entrenamiento muestran que pasadas las dos épocas las pérdidas de validación se hacen mayores que las de entrenamiento dando muestras de sobre-



Figura 1. Ejemplo de un par de imágenes insumo-objetivo en el conjunto de entrenamiento de la red neuronal. (izquierda) imagen de la placa deformada $k=60$ pixeles en el centro. (derecha) imagen-objetivo mostrando el subconjunto de los pozos detectados por el método de Hough.

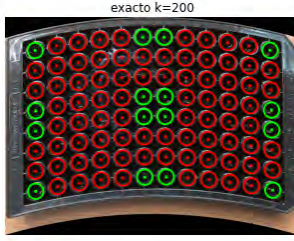


Figura 2. Placa de ensayos con deformación máxima marcada con centros considerados exactos. Los centros de pozos de color verde son los determinados visualmente en la placa sin deformar. Los rojos son resultado de la interpolación.

entrenamiento de la red, razón por la cual se seleccionó detener el entrenamiento cumplidas las dos épocas.

II-D. Localizaciones “Exactas” de pozos para la Evaluación

Una corroboración visual de los resultados podría considerarse como una validación de la precisión del modelo. Sin embargo, la función de pérdida usada para el entrenamiento del modelo mide la diferencia entre la predicción del modelo en el conjunto de prueba y el etiquetado que se hizo automáticamente con una técnica de procesamiento de imágenes.

Para corroborar cuantitativamente que, en efecto, el método de etiquetado automatizado produce resultados robustos que producen un modelo con resultados precisos definimos un método que mide la distancia entre la localización de los centros de pozos predichos por el modelo y la de los centros exactos determinados como sigue. Esto permite también comparar la precisión de este modelo con la de otros métodos de detección de pozos.

La localización exacta de los centros de los pozos se hace mediante una detección visual de los pozos localizados en las esquinas y centros de las placas sin deformar luego de haber sido procesadas según descrito en la Sección II-A (Figura 2). Con estas coordenadas se calculan las coordenadas de los centros de los demás pozos con una interpolación simple.

Luego las coordenadas de los centros de los pozos en las imágenes deformadas de manera controlada se calculan aplicando la transformación usada para la deformación (Ecuación 1) a las coordenadas de los centros detectadas en las placas sin deformar.

II-E. Implementación

La programación fue escrita en lenguaje *Python* versión 3.12.3. Para la lectura y escritura de imágenes, conversión de colores, y filtrado se utilizó *Skimage* (v 0.22.0) [8], *Numpy* (v 1.25.2) [13] para cálculos numéricos con arreglos, *Scipy* (v 1.13.0) [7] para el filtro gaussiano, *OpenCV* (v 4.9.0) [14] para las transformaciones de perspectivas y *Matplotlib* (v 3.8.4) [15] para la producción de gráficas. Las redes neuronales fueron implantadas usando *PyTorch* (v 2.1.2.post100) [16] y *TorchVision* (v 0.16.1+b88453f). Para el entrenamiento de la red se utilizó un servidor equipado con CPUs con 64 hilos de procesamiento y tres GPUs marca NVIDIA modelo

Tesla V100 con 16GB de memoria y 640 procesadores de Tensores cada uno. Las otras partes del proceso se ejecutaron en computadoras personales con CondaForge [17].

III. RESULTADOS

III-A. Medición de Error

Para cuantificar la precisión del presente método de detección de pozos definimos los conjuntos de medidas de errores como

$$E(k) = \{d(c_{i,j}^k, w_{i,j}^k) \mid i = 1, \dots, 8, j = 1, \dots, 12\}$$

donde k es el parámetro de deformación, $c_{i,j}^k$ es el centro exacto del pozo en la fila i , columna j en la placa deformada k píxeles, $d(c_{i,j}^k, w_{i,j}^k)$ es la distancia entre $c_{i,j}^k$ y el centro detectado $w_{i,j}^k$. Las coordenadas exactas de los centros de los pozos $c_{i,j}^0$ de las placas sin deformar ($k = 0$) fueron obtenidas identificando visualmente los centros de 16 pozos localizados en los bordes y el centro de la placa y generando el resto por interpolación. Para computar los centros $c_{i,j}^k$ para placas deformadas ($k > 0$) se aplicó la fórmula (1) a los centros $c_{i,j}^0$ de la placa sin deformar.

Placas de ambos colores negras y blancas mostraron niveles de error muy similares. La gráfica de violín (figura 3) muestra una variabilidad relativamente pequeña para ambos colores con muy pocos casos por encima de los 10 píxeles. No hay crecimiento del error al aumentar el grado de deformación. El error promedio para las placas negras fue de 4.48 píxeles mientras que para las blancas fue de 4.97 píxeles. Para todas las placas fue de 4.75 píxeles. Considerando que las imágenes recortadas tienen dimensiones que superan los 2000 por 3000 píxeles, estos errores son menores que un 0.14 % de la diagonal de la placa. La tabla I muestra los errores relativos pormenorizados por color y fuente de luz. La comparación con los resultados del trabajo previo que corrobora una disminución del error de los centros de los pozos encontrados por la red neuronal comparado con la combinación Hough-interpolación.

Al examinar los errores promedio por pozo (figura 4) no se observa un patrón con respecto a la localización de los promedios de error menores o mayores. En particular, no hay diferencias apreciables entre la precisión en los bordes de las placas en comparación con los centros.

La figura 5 muestra el resultado al someter una foto de una placa representativa de un contexto común en el que un usuario tomaría la foto. La placa está colocada sobre

Tabla I
PORMENORIZACIÓN DE ERRORES
COMO PORCIENTO DE LA DIAGONAL
DE LA FOTO RECORTADA

Fuente de iluminación	Hugh+interpolación		Red Neuronal	
	negra	blanca	negra	blanca
Sol indirecto	0.46 %	0.47 %	0.11 %	0.16 %
Fluorescente	0.36 %	0.54 %	0.12 %	0.12 %
Lámpara LED	0.68 %	0.24 %	0.12 %	0.09 %
Flash	0.48 %	0.38 %	0.15 %	0.14 %
Ultravioleta	(n/a)	0.33 %	(n/a)	0.18 %

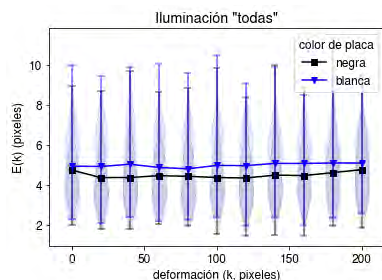


Figura 3. Errores promedio según el nivel de deformación de localización (k) y las distribuciones de errores alrededor de estos.

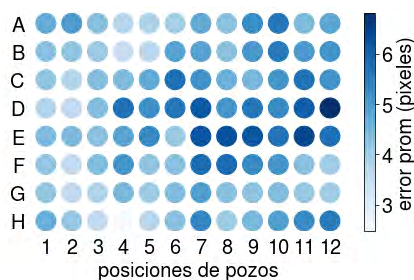


Figura 4. Error promedio por cada posición de pozo en la placa.

una mesa de madera y la sombra de quien toma la foto se proyecta parcialmente sobre ella. La foto no fue sujeta a ningún tipo de pre-procesamiento. Se observa que la red detectó razonablemente todos los pozos.

IV. DISCUSIÓN

La captación precisa para el análisis colorimétrico de placas de ensayo de 96 pozos empieza por la localización precisa de los pozos. Los resultados obtenidos aquí mejoran los obtenidos anteriormente por la combinación de la transformada de Hough e interpolación que a su vez mejoraba significativamente el resultado del uso de la transformada de Hough únicamente. De hecho, los errores medidos aquí son tan pequeños que podrían ser atribuibles tanto a la red neuronal como a error humano en la localización visual de los centros "exactos".

Además, el método utilizado aquí muestra el potencial de ser aplicable a otros tipos de medios para el análisis colorimétrico como por ejemplo en aparatos fluidicos. Por último, se ha demostrado que la combinación de la generación automática de conjuntos de entrenamiento combinado con la transferencia de conocimiento de redes neuronales previamente entrenadas para tareas afines elimina la necesidad de construir manualmente conjuntos de entrenamiento extensos.

RECONOCIMIENTOS

Los autores reconocen la ayuda los grupos de la Dra. Vibha Bansal y el Dr. Ezio Fasoli de los Departamentos de Química de la Universidad de Puerto Rico en Cayey y Humacao respectivamente.

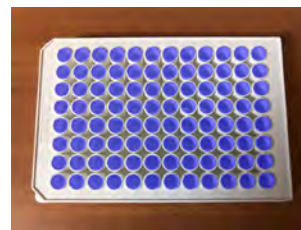


Figura 5. Pozos encontrados por la red neuronal (áreas azules) en una foto de una placa sin pre-procesar.

REFERENCIAS

- [1] G. M. Fernandes, W. R. Silva, D. N. Barreto, R. S. Lamarca, P. C. F. L. Gomes, J. F. da S. Petruc, and A. D. Batista, "Novel approaches for colorimetric measurements in analytical chemistry—a review," *Analytica Chimica Acta*, vol. 1135, pp. 187–203, 2020.
- [2] L. F. Capitán-Vallvey, N. Lopez-Ruiz, A. Martinez-Olmos, M. M. Erenas, and A. J. Palma, "Recent developments in computer vision-based analytical chemistry: A tutorial review," *Analytica Chimica Acta*, vol. 899, pp. 23–56, 2015.
- [3] S. Banik, S. K. Melanthota, Arbaaz, J. M. Vaz, V. M. Kadambalithaya, I. Hussain, S. Dutta, and N. Mazumder, "Recent trends in smartphone-based detection for biomedical applications: a review," *Analytical and Bioanalytical Chemistry*, vol. 413, pp. 2389–2406, 2021.
- [4] N. Qin, Z. Liu, L. Zhao, M. Bao, X. Mei, and D. Li, "Promising instrument-free detections of various analytes using smartphones with spotxel® reader," *Analytical Sciences*, vol. 39, no. 2, pp. 139–148, 2023.
- [5] G. B. Gomez-Dopazo, R. J. A. Nieves, R. L. A. Rivera, S. M. C. Morera, D. R. Nazario, I. Ramos, I. J. Dmochowski, D. Lee, and V. Bansal, "Cellulose acetate microwell plates for high-throughput colorimetric assays," *RSC advances*, vol. 14, no. 22, pp. 15319–15327, 2024.
- [6] E. Rosa Delgado and J. O. Sotero Esteva, "Detection of wells in images of deformed 96-wells plates," *MRS Advances*, pp. 1–6, 2024.
- [7] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.
- [8] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Goullart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.
- [9] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
- [10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- [11] T. Maintainers et al., "Torchvision: Pytorch's computer vision library, november 2016," 2023.
- [12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.
- [13] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al., "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- [14] G. Bradski, "The opencv library. dr dobb's j softw tools," *The OpenCV Library. Dr Dobb's J Softw Tools*, 2000.
- [15] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 03, pp. 90–95, 2007.
- [16] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
- [17] C.-F. Community, "The conda-forge project: Community-based software distribution built on the conda package format and ecosystem," *Zenodo*, 2015.