

SISTEMA INFORMÁTICO PARA LA CLASIFICACIÓN AUTOMÁTICA DE IMÁGENES DE GRANOS DE POLEN COMPUTER SYSTEM FOR THE AUTOMATIC CLASSIFICATION OF POLLEN GRAIN IMAGES

Autores: Daniel Orlando Jimenez Choy

<https://orcid.org/0000-0002-3015-7311>

Rodolfo L. Reyes Baños

<https://orcid.org/0000-0002-8114-6874>

Ioan A. Rodríguez Santana

<https://orcid.org/0000-0001-5780-7188>

Institución: Universidad de Ciego de Ávila Máximo Gómez Báez, Cuba

Correo electrónico: dojimenez@gmail.com

rodolfoforluis1964@gmail.com

ioan@unica.cu

RESUMEN

La presente investigación ofrece un sistema que facilita el proceso de clasificación de imágenes de granos de polen de La Empresa Apícola Cubana, cuyos resultados demostraron que existe lentitud en el proceso de análisis polínicos de las mieles, ya que las especies que pertenecen a una misma familia comparten características de identificación. Por esta razón se hizo necesario desarrollar un sistema informático que permitiera a los técnicos melisopolinólogos clasificar automáticamente una imagen de grano de polen. La implementación del sistema se realizó utilizando Keras para la creación de redes neuronales convolucionales y Tensor Flow para el trabajo con imágenes, ambas librerías de Python lo que posibilita su empleo en cualquier plataforma. Para guiar el proceso de desarrollo se utilizó la metodología *Rational Unified Process* (RUP). El sistema propuesto posibilita el identificación y clasificación rápida de imágenes de granos de polen. Almacena un conjunto de datos que permite al sistema identificar las especies de plantas.

Palabras clave: Clasificación automática, Imágenes microscópicas, Palinología, Polen.

ABSTRACT

The present investigation offers a system that facilitates the process of classifying images of pollen grains of the Cuban Beekeeping Company, whose results showed that there is slowness in the process of pollen analysis of honey, since the species that belong to the same family share characteristics of identification. For this reason, it became necessary to develop a computer system that would allow melissopalynologists to automatically classify a pollen grain image. The implementation of the system was carried out using Keras for the creation of convolutional neural networks and Tensor Flow for working with images, both Python libraries, which makes it possible to use them on any platform. To guide the development process, the Rational Unified Process (RUP) methodology was used. The proposed system enables the rapid identification and classification of images of pollen grains. It stores a data set that allows the system to identify plant species.

Keywords: Automatic classification, Microscopic images, Palynology, Pollen.

INTRODUCCIÓN

La Empresa Apícola Cubana es una organización que se dedica al acopio, beneficio y comercialización de los productos de la colmena con calidad y eficiencia. Como parte de sus políticas se realizan análisis polínicos de las mieles con el fin de asegurar la calidad de los productos al mismo tiempo que se revisa la dispersión de las plantas. Estos análisis tienden a tomar un tiempo y esfuerzo considerable por lo que retrasa los resultados.

La palinología, por medio de estudios del contenido de polen en la miel (melisopalinología) y el polen almacenado en la colmena, permite reconocer con exactitud, las plantas que visitan las abejas (Louveaux *et al.*, 1978) y (Moore y Webb, 1978), determinando por consiguiente el origen botánico de la miel y polen que almacenan estos insectos.

La identificación no se puede hacer más que por comparación de la morfología y las dimensiones de los granos de polen observados con aquellos granos conocidos que constituyen las referencias. Éstas pueden ser microfotografías, sobre papel o numerados; son un banco de datos que se pueden consultar por comparación. Las fotografías no son útiles a no ser las de calidad y tomadas en diferentes planos, sólo así se tiene una visión espacial del polen. Es igualmente necesario conocer la escala de la fotografía.

El melisopalinólogo debe confeccionar él mismo su colección polínica de referencia a partir de pólenes recolectados de las plantas y montados en láminas y laminillas. Ciertamente que es un trabajo largo y se necesita poseer una colección de pólenes que permitan una identificación precisa. Luego de todos estos elementos es evidente que, para hacer análisis polínico de mieles, es indispensable tener buenos conocimientos de botánica y de taxonomía vegetal. La identificación precisa de especies florales es naturalmente indispensable.

Es necesario destacar que se reconocen 354 especies que son de alguna manera importantes para la actividad apícola (Santos et al. 2000) de unas 6,700 especies de plantas vasculares presentes en la isla (Vales et al., 1998).

Por esta razón, se hizo necesario desarrollar un sistema informático que permitiera a los técnicos melisopalinólogos clasificar automáticamente una imagen de grano de polen.

La visión por computadora se ocupa de la parte de la percepción visual: es la ciencia de percibir y comprender el mundo a través de imágenes y videos mediante la construcción de un modelo físico del mundo para que un sistema de IA pueda tomar las acciones apropiadas. (Elgendy, 2020)

La clasificación de imágenes es una tarea que asigna imágenes a uno de los grupos predefinidos basados en el contenido de las imágenes. Tiene una amplia gama de aplicaciones en varios campos como la medicina, biología, visión por computador, e ingeniería. Sin embargo, es una tarea desafiante debido a las altas variaciones de fondo, iluminación, punto de vista, escala, deformación y oclusión a través de imágenes. (Bi et al., 2021)

Habitualmente se distingue entre clasificación supervisada y no supervisada. En la clasificación supervisada se conocen las clases y se busca extraer características que identifiquen a cada clase. Esta clasificación es la que se pretende aplicar en la investigación y presenta cuatro fases:

- I. Establecer el número de clases y caracterizarlas en función de su morfología. Para ello deben analizarse las características de una serie de casos cuya clase se conoce a priori y que, en teledetección, se denominan áreas de entrenamiento.
- II. Calibrar un modelo de clasificación a partir de estas áreas de entrenamiento.

- III. Validar el modelo de clasificación utilizando áreas de validación. Se trata de áreas obtenidas del mismo modo que las áreas de entrenamiento, pero no utilizadas en la fase de calibración.
- IV. Utilizar el modelo de clasificación generado para obtener una matriz de pesos que posteriormente se utilizará para la clasificación de las imágenes.

Un sistema informático para la clasificación automática de granos de polen implica que el análisis de las muestras de granos de polen se realiza de forma automática, mediante sistemas informáticos; sin que se produzca intervención humana en la realización de estas acciones. Para lograr este resultado utilizamos redes neuronales convolucionales.

Las redes neuronales son modelos matemáticos que se incluyen en la familia de algoritmos conocida como *deep learning* (aprendizaje profundo en castellano), donde una red neuronal es un modelo computacional, paralelo, compuesto de unidades procesadoras adaptativas (llamadas neuronas) con una alta interconexión entre ellas (García Sánchez *et al.*, 2019).

Las redes neuronales convolucionales o CNNs para abreviar, son un tipo especializado de modelo de red neuronal diseñado para trabajar con datos de imágenes bidimensionales, aunque se pueden usar con datos unidimensionales y tridimensionales (Brownlee 2019).

Las CNNs están formadas por diferentes capas, donde las primeras capas se encargan de extraer características como los bordes o esquinas de una imagen. Una vez detectados estos bordes son utilizados para detectar formas en las capas posteriores. Cuando se detectan las formas se procede a detectar las características de alto nivel como puede ser una rueda en una imagen donde aparece un coche. Las últimas capas están totalmente conectadas y son las encargadas de dar una predicción a partir de estas últimas características extraídas. Las CNNs son particularmente adecuadas para aprender características discriminatorias a partir de datos de entrada de imágenes. (Acero-Cuellar *et al.*, 2022)

El éxito de las convoluciones proviene de su capacidad para aprender patrones espaciales, lo que las ha convertido en el método predeterminado para cualquier dato que se parezca a una imagen. Cuando aplica una convolución a una imagen, puede aprender a detectar patrones simples como líneas horizontales o verticales, cambios

de color o patrones de cuadrícula. Cuando apila circunvoluciones en capas, comienzan a reconocer patrones más complejos, basándose en las circunvoluciones más simples anteriores. (Raff 2022)

Se pretende en esta investigación presentar un sistema capaz de identificar cada una imagen de un grano de polen, la especie a la cual pertenece.

MATERIALES Y MÉTODOS

Se selecciono un conjunto de 3000 imágenes de polen pertenecientes a tres especies diferentes como conjunto de entrenamiento para la red neuronal. Dado que la generación de grandes conjuntos de entrenamiento es un tema clave para los métodos basados en el aprendizaje profundo, se aplicó el aumento de datos en el conjunto de entrenamiento rotando aleatoriamente las imágenes hasta 15 grados, cambiando aleatoriamente las dimensiones de las imágenes hasta 10 % del ancho y alto de cada imagen, transformación de volteo horizontal, y cambio de contraste de hasta un 10 % de intensidad. (Ibrahim *et al.*, 2019)

Se realizó una transferencia de aprendizaje para aprovechar una red neuronal preentrenada y economizar recursos de computo. En específico se utilizó el método EfficientNetB0 de la familia EfficientNets propuestos por Tan y Le (Tan y Le, 2020), que logra una precisión y eficiencia mucho mejores que otras CNNs (ver Anexo 1).

Para el entrenamiento de la red neuronal convolucional se utilizará el método de *dropout*. *Dropout* es un método muy utilizado para abordar el problema de sobreentrenamiento de una red neuronal. La idea es soltar unidades aleatoriamente de la red neuronal durante el entrenamiento. (Belbahri *et al.*, 2019)

Dado que en este documento se utilizó un conjunto de datos multiclase, se eligió *Sparse Categorical Cross-entropy loss function* (función categórica de pérdida de entropía cruzada) para la validación del modelo (Mardanirad *et al.*, 2021), la pérdida de entropía cruzada, o pérdida logarítmica, mide el rendimiento de un modelo de clasificación cuyo resultado es un valor de probabilidad entre 0 y 1. La pérdida de entropía cruzada aumenta a medida que la probabilidad predicha diverge de la etiqueta real.

RESULTADOS Y DISCUSIÓN

La implementación del sistema se realizó utilizando Keras para la creación de redes neuronales convolucionales y Tensor Flow para el trabajo con imágenes, ambas librerías de Python lo que posibilita su empleo en cualquier plataforma.

El uso de Tensor Flow nos permite, mientras se entrena un modelo, realizar un seguimiento de métricas como exactitud de clasificación, precisión, recuperación, etc. Además, se pueden monitorear estas métricas no solo en los datos de entrenamiento, sino también en un conjunto de validación.

Para realizar la clasificación hay dos procesos que deben llevarse a cabo: el entrenamiento y la validación. El entrenamiento es un proceso de aprendizaje de la red neural el cual intenta reconocer patrones de datos. Los resultados del entrenamiento son los pesos y el modelo entrenados (Wadi 2021). La validación mide que tan bien el modelo puede tomar lo que ha aprendido y generalizarse a nuevos datos.

Para la validación del modelo se eligieron dos de las métricas más importantes para una red neuronal convolucional: precisión y pérdida. Precisión es la métrica principal para el rendimiento ampliamente aprovechada para evaluar redes (Wong 2018). Calcula la frecuencia con la que las predicciones son iguales a las etiquetas y nos da una medida de que tan eficaz es la red neural a la hora de clasificar. Pérdida funciona penalizando las clasificaciones falsas. Pérdida más cercana a 0 indica una mayor precisión. En general, la minimización de la pérdida proporciona una mayor precisión para el clasificador.

Para la corroboración del modelo se realizaron 20 iteraciones sobre un conjunto de 3000 imágenes pertenecientes a 3 categorías.

Tabla 1: Pérdida y precisión por iteraciones

Iteración	Pérdida	Precisión
1	0.60535413	0.833333313
2	0.400913268	0.898989916
3	0.327958256	0.914141417
4	0.289420724	0.919191897
5	0.265031397	0.919191897
6	0.247766033	0.924242437
7	0.236093238	0.934343457
8	0.226920679	0.929292917

9	0.219852895	0.934343457
10	0.215432122	0.929292917
11	0.211437494	0.924242437
12	0.208388895	0.919191897
13	0.206878111	0.919191897
14	0.205288455	0.914141417
15	0.204366311	0.914141417
16	0.204637155	0.909090936
17	0.203017816	0.909090936
18	0.201891541	0.914141417
19	0.200215295	0.919191897
20	0.200523898	0.914141417

Los resultados de la Tabla 1 demuestran que a medida que transcurren las iteraciones se mejora la precisión del modelo y disminuye la pérdida, lo cual revela que el entrenamiento del modelo es correcto (Figura 3).

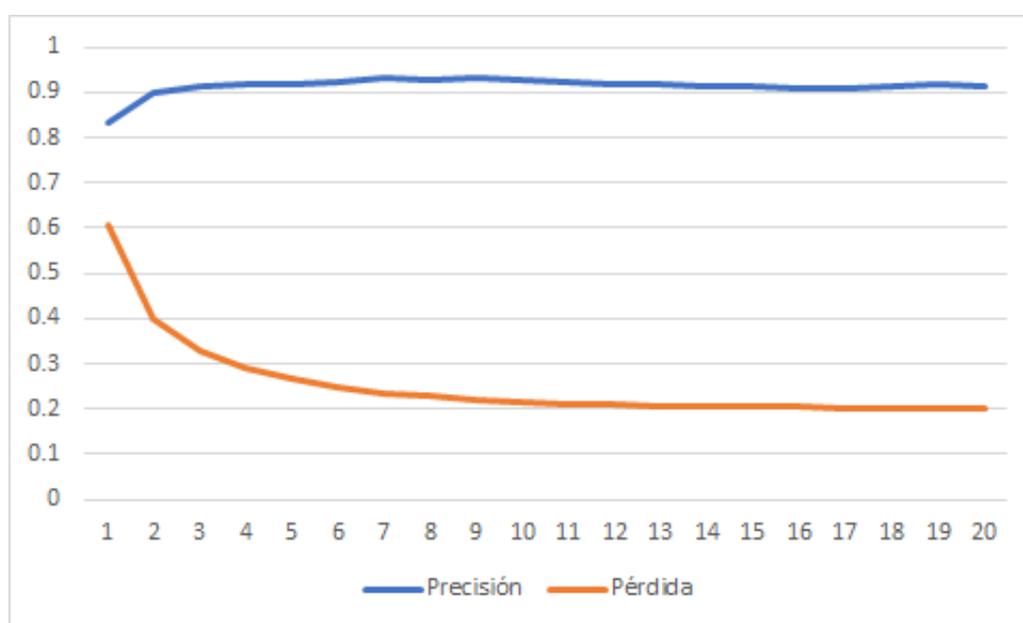


Figura 3: Gráfico de progresión del entrenamiento

Como podemos observar, aplicando el método de pérdida de entropía cruzada se logra obtener 91 % de exactitud en la vigésima iteración, los valores de pérdida de entropía cruzada disminuyeron progresivamente a medida que aumentaba el número de épocas. Esto confirma que la red neuronal sigue aprendiendo de los datos de forma continua durante todo el proceso de entrenamiento.

Aunque los resultados obtenidos por la red neuronal no son los mejores entre las investigaciones relacionadas, es la más alta de un 94.77 % (Sevillano y Aznarte, 2018), se obtiene un resultado lo suficientemente preciso para que pueda ser utilizada en el sistema desarrollado utilizando menos recursos, ya que al utilizar una red neuronal preentrenada se reduce el tiempo necesario para entrenar la red neuronal de días a minutos y la cantidad de imágenes necesarias de millones a miles.

Se utilizó Django como *framework* para el desarrollo de la aplicación informática permitiendo integrar la red neuronal desarrollada en Tensor Flow y Keras directamente con el *framework*. Django es una buena plataforma basada en Python usada normalmente para una mayor velocidad y flexibilidad en el diseño, así como para una amplia gama de aplicaciones (Adams 2021).

CONCLUSIONES

En este trabajo se han presentado un modelo basado en redes neuronales convolucionales de aprendizaje profundo y capaz de clasificar satisfactoriamente las imágenes de granos de polen del conjunto de validación. La solución ha superado el umbral del 90 % de clasificaciones correctas. Encontrar una solución automatizada para la clasificación del polen implica no solo un mejor rendimiento en la Empresa Apícola Cubana, sino también mejoras en la agricultura y la medicina.

Estos resultados nos impulsan a promover la creación de una base de datos de polen presente en la isla, ampliando así el número de imágenes disponibles para aumentar la capacidad de clasificación de la solución propuesta.

REFERENCIAS BIBLIOGRÁFICAS

ACERO-CUELLAR, T. ... [et al.] (2022). There's no difference: Convolutional Neural Networks for transient detection without template subtraction. Disponible en: <https://arxiv.org/pdf/2203.07390v1.pdf>. Visitado: 25 de marzo de 2022.

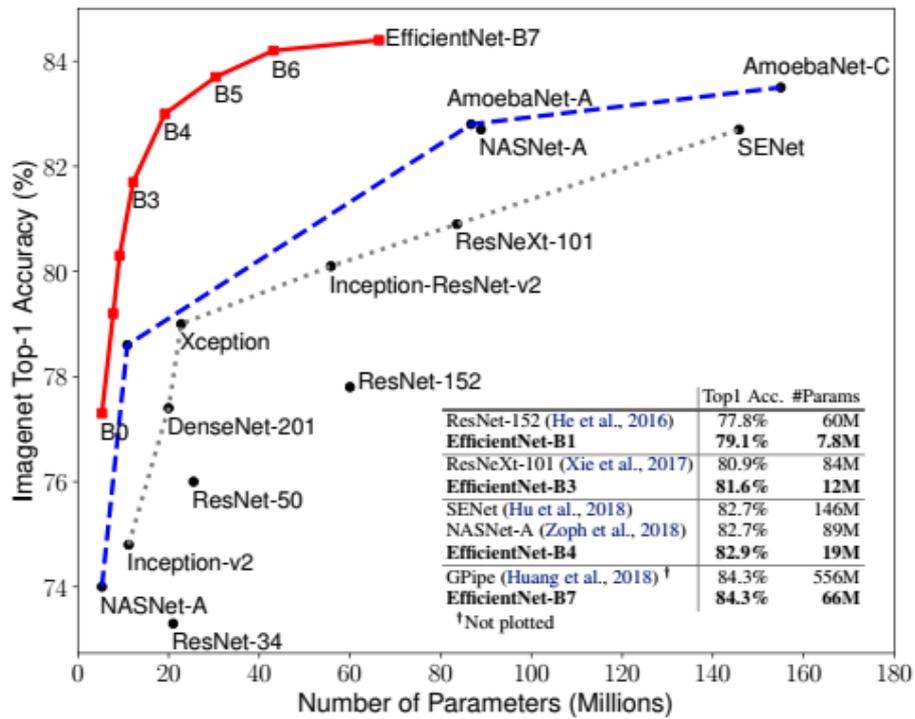
ADAMS, A. (2021). Python Programming, Deep Learning: 3 Books in 1: A Complete Guide for Beginners, Python Coding for Ai, Neural Networks, & Machine Learning, Data Science/Analysis with Practical Exercises for Learners. S. L.: Anthony Adams.

BELBAHRI, M. ... et al.] (2019). Foothill: A Quasiconvex Regularization for Edge Computing of Deep Neural Networks (arXiv:1901.06414). arXiv. Disponible en: <https://doi.org/10.48550/arXiv.1901.06414>. Visitado: 21 de septiembre de 2022.

- BI, Y., XUE, B., y ZHANG, M. (2021). Genetic Programming for Image Classification: An Automated Approach to Feature Learning. S.L.: Springer Nature. ISBN 978-3-030-65927-1.
- BROWNLEE, J. (2019). Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python. Machine Learning Mastery. Disponible en: <https://machinelearningmastery.com/deep-learning-for-computer-vision/>. Visitado: 3 de mayo de 2022.
- ELGENDY, M. (2020). Deep Learning for Vision Systems. Shelter Island, NY: s.n. ISBN 978-1-61729-619-2.
- GARCÍA SÁNCHEZ, E., ALCALÁ NALVAIZ, J. T. y ORELLANA LOZANO, L. (2019). Introducción a las redes neuronales de convolución. Aplicación a la visión por ordenador—Repositorio Institucional de Documentos. Universidad de Zaragoza. Disponible en: <https://zaguan.unizar.es/record/87398?ln=es#> Visitado: 6 julio 2021.
- IBRAHIM, Y., NAGY, B. y BENEDEK, C. (2019). CNN-Based Watershed Marker Extraction for Brick Segmentation in Masonry Walls. En F. Karray, A. Campilho, y A. Yu (Eds.), Image Analysis and Recognition (pp. 332-344). Springer International Publishing. Disponible en: https://doi.org/10.1007/978-3-030-27202-9_30 Visitado: octubre de 2022.
- LOUVEAUX, J., MAURIZIO, A. y VORWOHL, G., (1978). Methods of Melissopalynology. *Bee World*, Vol. 59, No. 4, pp. 139-157. Disponible en: <https://doi.org/10.1080/0005772X.1978.11097714>. Visitado: 24 de junio de 2022.
- MARDANIRAD, S., WOOD, D. y ZAKERI, H., (2021). The application of deep learning algorithms to classify subsurface drilling lost circulation severity in large oil field datasets. *SN Applied Sciences*, Vol. 3. Disponible en: <https://doi.org/10.1007/s42452-021-04769-0>. Visitado: 24 de junio de 2022.
- MOORE, P.D. y WEBB, J.A., (1978). An Illustrated Guide to Pollen Analysis. London: s.n. ISBN 978-0-340-21449-7.
- RAFF, E. (2022). Inside Deep Learning: Math, Algorithms, Models. S.I.: Simon and Schuster.
- SANTOS, I.E.M. ...[et al.], (2000). Flora apícola de la provincia de Camagüey. *Revista del Jardín Botánico Nacional*, Vol. 21, No. 2, pp. 235-252.

- SEVILLANO, V. y AZNARTE, J.L., (2018). Improving classification of pollen grain images of the POLEN23E dataset through three different applications of deep learning convolutional neural networks. *PLOS ONE*, Vol. 13, No. 9, pp. e0201807. Disponible en: <https://doi.org/10.1371/journal.pone.0201807>. Visitado: 10 de octubre de 2022.
- TAN, M. y LE, Q.V., (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. En: arXiv:1905.11946 [cs, stat]type: article [en línea]. S.I.: arXiv. arXiv:1905.11946. Disponible en: <http://arxiv.org/abs/1905.11946>. Visitado: 9 junio 2022.
- VALES, M.A., ÁLVAREZ, A., MONTES, L. y ÁVILA, A., (1998). Estudio Nacional sobre la Diversidad Biológica en la República de Cuba [en línea]. Disponible en: <https://docplayer.es/storage/68/59769548/59769548.pdf>. Visitado: 10 de septiembre de 2022.
- WADI, H. (2021). Step By Step Neural Networks for Image Classification using Python GUI: A practical approach to understand the neural networks algorithm for image classification with project based example. S.I.: Turida Publisher.
- WONG, A. (2018). NetScore: Towards Universal Metrics for Large-scale Performance Analysis of Deep Neural Networks for Practical On-Device Edge Usage (arXiv:1806.05512). arXiv. Disponible en: <https://doi.org/10.48550/arXiv.1806.05512> Visitado: 9 junio 2022.

1. Anexo



Anexo 1: Comparativa de EfficientNets. Extraída de (Tan y Lee 2020)