

# REDES NEURONALES CONVOLUCIONALES PARA EL ANÁLISIS DE IMÁGENES AÉREAS TOMADAS A CAMPOS DE CULTIVO

Manuel de Jesús López Martínez<sup>1</sup>, M.I.A. Santiago Villagrana Barraza<sup>2</sup>,  
Dr. Carlos Alberto Olvera Olvera<sup>3</sup>, Dra. María Auxiliadora Araiza Esquivel<sup>4</sup> y Dr. Carlos Eric Galván Tejada<sup>5</sup>

**Resumen**—El análisis y la clasificación de imágenes dentro de la inteligencia artificial es uno de los temas que está teniendo una gran importancia dentro de la comunidad científica, ya que, gracias a los métodos y algoritmos utilizados es posible obtener datos importantes de una determinada selección de imágenes que nos pueden ayudar a generar conocimiento de cualquier tipo. Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés), es uno de los tipos de redes neuronales artificiales que nos permiten imitar la visión del ser humano, es decir, ver de manera artificial usando componentes de hardware y software. En este caso, las imágenes que se obtienen para el proyecto se adquieren a partir del uso vehículos aéreos no tripulados (UAV, por sus siglas en inglés), esto para analizar o procesar las imágenes adquiridas e identificar ciertas características que pueden presentarse en los campos de cultivo como; humedad, crecimiento regular o irregular de la plantación, plagas, maleza, alteraciones en el suelo o contaminación del mismo, etc. Para este proyecto se presenta el análisis de una red neuronal convolucional para identificar humedad dentro de un campo de cultivo usando *Keras*, que es una librería de Python que permite realizar análisis de imágenes.

**Palabras clave**—Redes Neuronales Convolucionales (CNN), Vehículos Aéreos no Tripulados (UAV), reconocimiento de patrones, clasificación de imágenes.

## Introducción

Como innovación tecnológica dentro de la agricultura, se han estado buscando nuevas tecnologías que puedan ayudar a mitigar los gastos innecesarios en la aplicación de monitoreo en los campos de cultivo, por lo que el uso de herramientas tecnológicas beneficia de gran forma al desarrollo agrícola [1].

Actualmente los Drones o Vehículos Aéreos no Tripulados (UAV), son dispositivos que tienen múltiples propósitos dentro de la vida cotidiana, ya que pueden ser usados para mapear grandes áreas en las que los aviones convencionales no pueden acceder, como por ejemplo en los cráteres de los volcanes, también se usan en la agricultura, meteorología, minería y muy actualmente hasta como servicio postal [2]. Las series de imágenes obtenidas de los Drones ayudan a extraer información de la superficie y así construir un mapa de un área para ser analizada o procesada y así obtener características que identifiquen el área muestreada.

En tiempos recientes es posible implementar múltiples algoritmos que nos permiten extraer información característica de una imagen, pero los métodos que principalmente se usan son los que tienen que ver con la inteligencia artificial o CNN [3]. Una de las herramientas que nos permite realizar el procesamiento o análisis de las imágenes es la librería *Keras* escrita en el lenguaje de programación *Python* [4]. Esta librería que permite implementar y crear prototipos de redes neuronales, además que es de licencia Open Source, en comparación con otras aplicaciones en las cuales se tiene que pagar por la licencia.

En esta investigación se propone un modelo de CNN para el análisis de las imágenes aéreas obtenidas de un Dron DJI phantom 4 en un campo de chile y del cual se tratará de encontrar la humedad que se encuentra en él.

1 Manuel de Jesús López Martínez, Estudiante de Maestría de la Universidad Autónoma de Zacatecas, Zacatecas, Zacatecas. [30110259@uaz.edu.mx](mailto:30110259@uaz.edu.mx).

2 M.I.A. Santiago Villagrana Barraza, Docente-Investigador de la Universidad Autónoma de Zacatecas, Zacatecas, Zacatecas [svillagrana@uaz.edu.mx](mailto:svillagrana@uaz.edu.mx).

3 El Dr. Carlos Alberto Olvera Olvera, Docente-Investigador de la Universidad Autónoma de Zacatecas, Zacatecas, Zacatecas [colvera@uaz.edu.mx](mailto:colvera@uaz.edu.mx).

4 Dra. María Auxiliadora Araiza Esquivel, Docente-Investigador de la Universidad Autónoma de Zacatecas, Zacatecas, Zacatecas [araizama@uaz.edu.mx](mailto:araizama@uaz.edu.mx).

5 Dr. Carlos Eric Galván Tejada, Docente-Investigador de la Universidad Autónoma de Zacatecas, Zacatecas, Zacatecas, [ericgalvan@uaz.edu.mx](mailto:ericgalvan@uaz.edu.mx).

## Descripción del Método

### Obtención del conjunto de datos.

Las imágenes obtenidas a través de los Drones tienen una resolución de 4864x3648 (ver figura 1), cada imagen que se obtiene es parte de la secuencia que el Dron extrae de cada barrido al campo completo, es decir, estas imágenes aún no forman el ortomosaico completo de los campos de cultivo. Para realizar el ortomosaico de imágenes (ver figura 2), se utiliza la herramienta de software llamada OpenDroneMap, la cual es de software libre y permite unir toda la secuencia de imágenes obtenidas a través del Dron. Con las imágenes de los campos de cultivo se hace un recorte manual mediante una herramienta de edición de imágenes, los nuevos recortes quedan con una resolución de 243x183 (5% de la imagen original obtenida), los recortes obtenidos se clasifican en dos clases “humedad” y “no\_humedad”, tal y como se clasifica en el trabajo de Nassim *et al.* [5], mencionadas clases serán parte del conjunto de entrenamiento. Una vez obtenidos los conjuntos de datos se realiza el análisis correspondiente de estos mediante CNN. En la figura 3 se muestra la metodología para este proyecto.

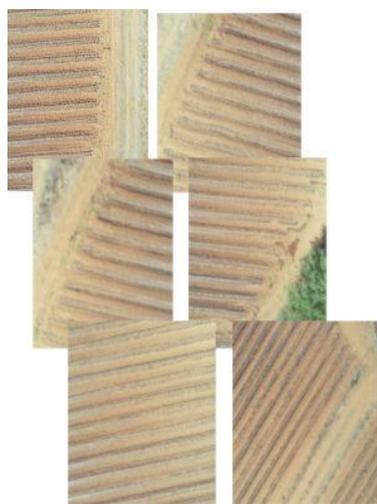


Figura 1. Secuencia de imágenes obtenidas por el Dron.

Figura 2. Ortomosaico procesado en OpenDroneMap.



Figura 3. Metodología aplicada para el proyecto.

### Análisis de las imágenes.

Las redes neuronales convolucionales se han convertido en una herramienta muy poderosa para realizar múltiples tareas como; el reconocimiento de patrones en imágenes, reconocimiento en tiempo real, procesamiento y caracterización de objetos entre otras. Es por ello que requieren de un gran nivel de exactitud al momento de hacer las predicciones [6]. Para obtener gran nivel de precisión también se necesita de hardware especializado para procesar grandes cantidades de imágenes, es por eso que la mayoría de las implementaciones de CNNs necesitan de tarjetas capaces de realizar tan forzada tarea.

Es importante tener un conjunto de datos lo suficientemente grande para poder obtener buenos resultados en el análisis de las imágenes con una red neuronal convolucional [7]. El conjunto de datos para la CNN diseñada contiene 300 imágenes, las cuales ya han sido recortadas previamente y pertenecen a las dos clases mencionadas anteriormente.

Para el diseño de la CNN se tomaron en cuenta múltiples capas de convolución (*convolution layers*), de normalización (*normalization layers*), capas de agrupación (*pooling layers*) y capas completamente conectadas (*fully connected layers*), y de las cuales, las capas de convolución son de los principales bloques que caracterizan a este tipo de redes neuronales artificiales y tienen como parámetros un conjunto de filtros de aprendizaje formados por lo alto, ancho y en algunas ocasiones la profundidad de las imágenes[8]. Las capas de normalización dentro de las CNNs tienen una contribución mínima ya que su propuesta se basa en los esquemas de inhibición que a veces son observados en un cerebro biológico. Contrario a las capas *pooling* o de agrupación, las cuales son insertadas entre las capas de convolución, esto para reducir progresivamente el tamaño de la imagen, la cantidad de parámetros y cálculo de la red neuronal, esto para evitar el sobreajuste (*overfitting*). El tamaño de filtro que se utiliza comúnmente es de 2x2.

Las capas completamente conectadas contienen las neuronas con conexiones completas con todas las activaciones de la capa anterior, por lo que sus activaciones se pueden calcular con una multiplicación de matrices seguida por un desplazamiento de sesgo.

Cada capa de convolución contiene una función de activación que se llama ReLU (Rectified Linear Unit), la cual asigna un 0 si recibe una entrada negativa, pero para un valor  $x$  positivo regresa un valor mayor que 0, se representa por la función:

$$f(x) = \max(0, x)$$

Otra función que se utiliza es la “Softmax”, la cual se implementa en la capa de salida y ésta representa una función exponencial normalizada y se utiliza para representar distribuciones categóricas y comprimir vectores arbitrarios con valores reales de rango [0,1], se representa por la función:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K.$$

En la figura 4 se muestra el modelo de la CNN utilizada para el proyecto la cual contiene 2 capas de convolución con función de activación “ReLU”, en la segunda capa contiene una función de pérdida del 25% (*Dropout*), 2 capas ocultas con función de activación “ReLU” y con una función de pérdida del 50% (*Dropout*) en ambas, y 1 capa de salida la cual contiene las dos clases a analizar con función de activación “Softmax”.

### Experimentación

Para la compilación del modelo de la CNN, se usó una función de pérdida de tipo *categorical\_crossentropy*, las funciones de pérdida definen la señal de retroalimentación usada para el aprendizaje durante el entrenamiento del modelo [9]. Las del tipo *entropía cruzada*, como la que se usó para este modelo, permiten simular eventos raros donde se necesitan probabilidades muy pequeñas y requieren que se estimen con precisión [10]. Como optimizador (*optimizer*) se tomó en cuenta “Adadelta”, el cual adapta las tasas de aprendizaje en función de la ventana móvil de las actualizaciones, es decir, continúa aprendiendo aun cuando se hayan hecho muchas actualizaciones [11].

Para el entrenamiento se tomaron en cuenta varios parámetros aparte del conjunto de entrenamiento y de validación como; el tamaño del lote o *batch\_size* el cual define la cantidad de iteraciones que tendrá una época por cada conjunto de dato de entrenamiento y las épocas o *epochs* que son la cantidad de iteraciones en un recorrido hacia adelante y hacia atrás en la red neuronal, a diferencia del *batch\_size*, las épocas recorren en un solo paso todo el conjunto de entrenamiento [12]. Es importante definir de manera correcta los valores de los parámetros anteriores ya que junto con el conjunto de entrenamiento y el conjunto de validación son de gran importancia para poder tener diseñado un buen modelo de red neuronal y así obtener excelentes resultados.

Para el entrenamiento del modelo, se utilizó un tamaño de lote de 32 y épocas 20, los dos valores fueron utilizados para adaptarlos tanto a la CNN como al rendimiento de la computadora en la que fueron procesados los datos.

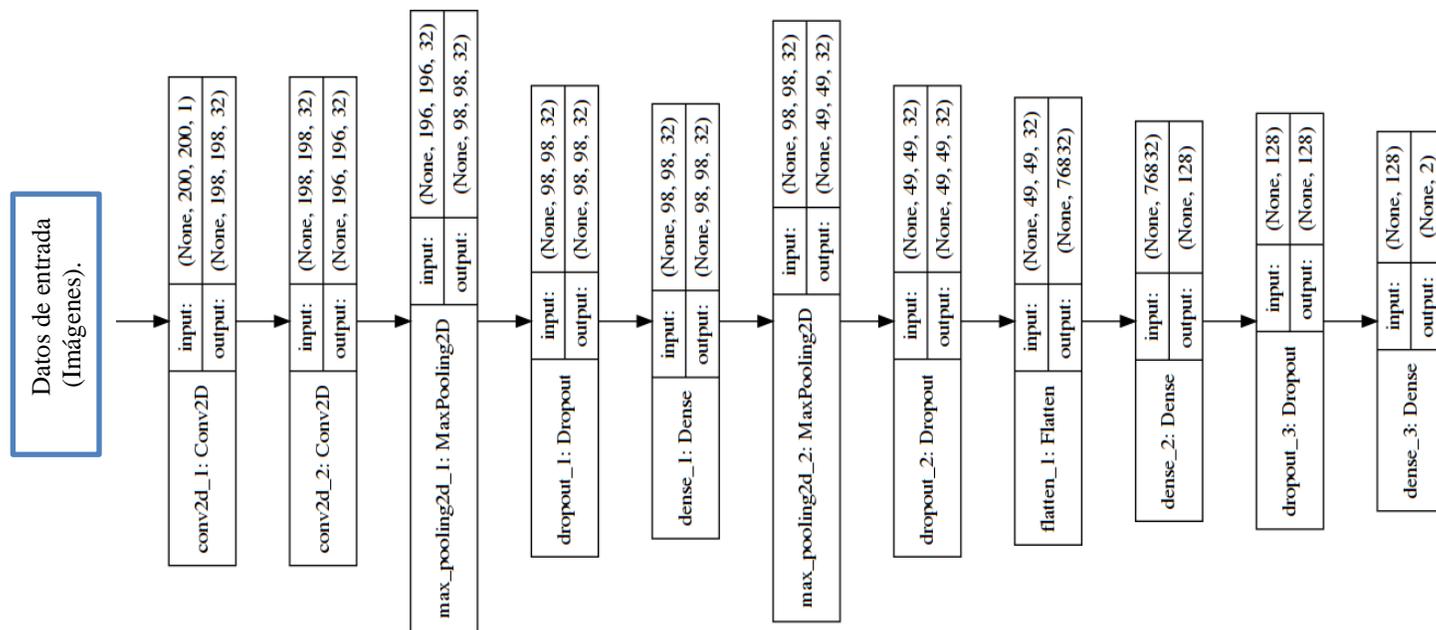


Figura 4. Modelo de la CNN utilizada.

Resumen de resultados

Después de aplicar la metodología anterior, para verificar los resultados se consideraron los valores de *accuracy* o exactitud y la función de pérdida o *loss*. Para probar la CNN y hacer una comparación de la exactitud o *accuracy* que se puede obtener con diferente cantidad de datos, se definieron dos conjuntos de datos uno de 200 imágenes y otro de 300, al realizar el análisis correspondiente se obtuvieron los resultados mostrados en la tabla 1:

Cantidad de datos.	Accuracy	Loss	Score Accuracy	Score Loss
200	0.49	0.69	0.55	0.64
300	0.66	0.64	0.70	0.62

Tabla 1. Resultados obtenidos del entrenamiento, la pérdida y *accuracy* total con diferente cantidad de datos de entrenamiento.

Con lo anterior se puede deducir que con un conjunto de datos de 200 imágenes se obtiene una probabilidad del 55% de que las imágenes sean clasificadas de manera correcta, mientras que con un conjunto de datos de 300 imágenes se tiene la probabilidad de que el 70% de las imágenes sean clasificadas de manera correcta, por lo que es importante considerar que, como se mencionó anteriormente, el conjunto de datos sea lo suficientemente grande para obtener una mejor probabilidad de clasificación. Las gráficas que representan los datos anteriores se muestran en las figuras 5 y 6:

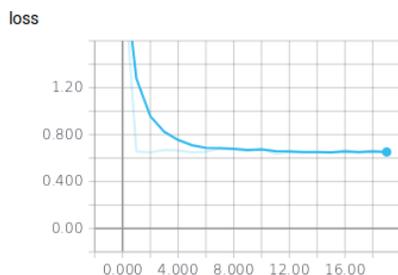


Figura 5. Función de pérdida obtenida durante el entrenamiento

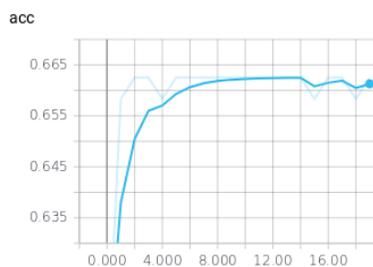


Figura 6. *Accuracy* obtenida durante el entrenamiento

En caso particular de este proyecto, es importante obtener buenos resultados para garantizar que los cultivos tengan un buen desarrollo y el producto obtenido en este caso el chile sea de calidad. Es por ello que en base a los resultados que se obtuvieron es necesario reestructurar la red neuronal convolucional y adaptarla para que pueda ser procesada en un hardware mucho más especializado, ya que en la mayoría de los trabajos relacionados se utilizan tarjetas NVIDIA o tarjetas gráficas capaces de procesar imágenes en tiempos mucho más cortos, y es importante mencionar que en este proyecto se utilizó una tarjeta Intel® HD Graphics 620 cuyas características de rendimiento no se comparan a las recientes tarjetas sacadas al mercado, como por ejemplo la NVIDIA GeForce GTX 1080.

### Conclusiones

En éste estudio se analizó una CNN para identificar la humedad en los campos de chile ya que en algunas ocasiones estas pueden representar fugas en el sistema de riego, lo que compromete la integridad del cultivo, además del desperdicio de agua lo que puede generar gastos extras para los agricultores. En base a los resultados obtenidos se puede deducir que la CNN es funcional para el proyecto, el tiempo que tardaba en hacer el entrenamiento por época era aproximado a 30 segundos por lo que en promedio para hacer el análisis de todo el conjunto de datos fue de 10 minutos. El *accuracy* obtenido (0.70), así como la función de pérdida o *loss* (0.62), representan valores aceptables para poder considerar que la CNN clasifica las imágenes en “humedad” y “no\_humedad” en un 70% de efectividad.

### Referencias

1. Barreiro P, Valero C. (2016). Drones Aplicados a la Agricultura de Precisión. Dossier, p. 36.
2. Bryson M, Reid A, Ramos F, Sukkarieh S. (2010). Airborne vision-based mapping and classification of large farmland environments. *Journal of Field Robotics* 27(5):632–655.
3. Tao Qu, Quanyuan Zhang, Shilei Sun. (2016). Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks. Springer, NY.
4. Keras: The Python Deep Learning Library. Available Online: <https://keras.io/> (Acceded on sep. 2018).
5. Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., & Zuair, M. (2017). Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sensing*, 9(4), 312.
6. Wang, J., Lin, J., & Wang, Z. (2018). Efficient Hardware Architectures for Deep Convolutional Neural Network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(6), 1941–1953.
7. Zhao, Y., Ma, J., Li, X., & Zhang, J. (2018). Saliency Detection and Deep Learning-Based Wildfire Identification in UAV Imagery. *Sensors*, 18(3), 712.
8. Božić-Štulić, D., Kružić, S., Gotovac, S., & Papić, V. (2018). Complete Model for Automatic Object Detection and Localisation on Aerial Images using Convolutional Neural Networks. *Journal of Communications Software and Systems*, 14(1).
9. F. Chollet, J. J. Allaire, *Deep learning with R*. 1st ed. Manning, 2018. 360 p.
10. Kroese D.P., Rubinstein R.Y., Cohen I., Porotsky S., Taimre T. (2013) Cross-Entropy Method. In: Gass S.I., Fu M.C. (eds) *Encyclopedia of Operations Research and Management Science*. Springer, Boston, MA.
11. Zeiler, M.D. (2012). Adadelta: An Adaptive Learning Rate Method. Available Online: <https://arxiv.org/abs/1212.5701> (Acceded on sep. 2018).
12. A. Gulli, S. Pal. *Deep Learning with Keras*. Packt Publishing, 2017. 318 p.