



DOI: [https://doi.org/10.37811/cl\\_rcm.v6i5.3420](https://doi.org/10.37811/cl_rcm.v6i5.3420)

Métricas de rendimiento para evaluar el aprendizaje automático  
en la clasificación de imágenes petroleras utilizando redes  
neuronales convolucionales.

**Adriana Zapeta Hernández**

[adri\\_zepeta@hotmail.com](mailto:adri_zepeta@hotmail.com)

<https://orcid.org/0000-0003-1682-9852>

**Gustavo Alejandro Galindo Rosales**

[gustavo.galindo@itspozarica.edu.mx](mailto:gustavo.galindo@itspozarica.edu.mx)

<https://orcid.org/0000-0002-4858-3781>

**Héctor Jesús Juan Santiago**

[hector.juan@itspozarica.edu.mx](mailto:hector.juan@itspozarica.edu.mx)

<https://orcid.org/0000-0002-0299-9861>

**Maribel Martínez Lee**

[maribel.martinez@itspozarica.edu.mx](mailto:maribel.martinez@itspozarica.edu.mx)

<https://orcid.org/0000-0002-3928-8967>

## RESUMEN

Las redes neuronales entran en un campo extenso de la inteligencia artificial, sus funciones son: el aprendizaje relacionado con imágenes, reconocimiento de voz, detección de texto, detección de objetos, detección de idioma, reconocimiento facial, etc. Una red neuronal aprende tareas analizando grandes cantidades de datos, el aprendizaje puede ser supervisado, reforzado o no supervisado. La red neuronal convolucional (CNN) utilizan una jerarquía de características combinando las de bajo nivel con una estructura de capas para formar características de alto nivel. (Rashka & Mirjalili, 2019) Este trabajo se centra en el desarrollo de un software inteligente basado en una CNN, que clasifica imágenes, patrones y texto de fotos tomadas a pozos petroleros, las cuales son evaluadas por medio de métricas de rendimiento, que ayudan a detectar posibles errores, falsas predicciones, suba justes y sobreajustes que pueden llegar a surgir en el aprendizaje automático. El objetivo de este trabajo es mostrar cómo se aplica cada una de las métricas de rendimiento para obtener una mejor precisión y exactitud para la clasificación de imágenes utilizando redes neuronales convolucionales.

**Palabras clave:** Machine learning; red neuronal convolucional; métricas rendimiento.

Correspondencia: [maribel.martinez@itspozarica.edu.mx](mailto:maribel.martinez@itspozarica.edu.mx)

Artículo recibido: 10 agosto 2022. Aceptado para publicación: 10 septiembre 2022.

Conflictos de Interés: Ninguna que declarar

Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo

Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) 

Como citar: Zapeta Hernández, A., Galindo Rosales, G. A., Juan Santiago, H. J., & Martínez Lee, M. (2022). Métricas de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales. *Ciencia Latina Revista Científica Multidisciplinar*, 6(5), 4624-4637.

[https://doi.org/10.37811/cl\\_rcm.v6i5.3420](https://doi.org/10.37811/cl_rcm.v6i5.3420)

## Performance metrics to evaluate machine learning in oil image classification using convolutional neural networks.

### ABSTRACT

Neural networks enter an extensive field of artificial intelligence, their functions are: image-related learning, speech recognition, text detection, object detection, language detection, facial recognition, etc. A neural network learns tasks by analyzing large amounts of data, learning can be supervised, reinforced or unsupervised. This work focuses on the development of intelligent software based on a convolutional neural network, whose objective is to classify images, patterns and text from photos taken of oil wells, which are evaluated through performance metrics, which help detect possible errors, false predictions, underfitting and overfitting that may arise in machine learning. The objective of this work is to show how each of the performance metrics is applied to obtain better precision and accuracy for image classification using convolutional neural networks.

**Keywords:** Machine learning, convolutional neural network, performance metrics.

## INTRODUCCIÓN

La inteligencia artificial ha influido en los avances tecnológicos y científicos, así como en la medicina, educación, ingeniería, robótica y agricultura. Empresas como Google, Amazon, Microsoft, Facebook han creado algoritmos de inteligencia artificial, como redes neuronales para la clasificación de imágenes, detección de objetos, reconocimiento de caracteres ópticos (OCR), etc.

En México, Petróleos Mexicanos es una empresa de clase mundial, pero depende de empresas subsidiarias que realizan reportes diarios de la productividad de los pozos petroleros, estos reportes los realizan las cuadrillas que tienen a su cargo al visitar los pozos, la tardanza que se genera al procesar los reportes lleva a que estos se entreguen al día siguiente.

La presente investigación desarrolla un software inteligente, que clasifica imágenes, reconoce patrones y dígitos manuscritos, por medio de una red neuronal convolucional. Este trabajo se centra en el problema que tiene la red neuronal al clasificar un tipo de dato nuevo, por factores como el ruido de las imágenes y el sobre entrenamiento del algoritmo provoca que sólo las imágenes idénticas las considera como correctas.

Para resolver la problemática anterior se propone utilizar métricas de rendimiento que mejore las predicciones del algoritmo, estas se obtienen del Scikit-learn, que son librerías de código abierto para mejorar algoritmos y funciones en todas las etapas de procesamiento, entrenamiento, optimización y validación de modelos predictivos, en este caso la red neuronal convolucional. La importancia de este trabajo es ejemplificar la implementación en lenguaje Python de cada una de las métricas de rendimiento para mejorar la predicción de la red neuronal convolucional y su generalización para otros casos.

Este software se desarrolló en Laboratorio Mirai Innovation Research, Osaka Japón, con el apoyo del investigador Dr. Chistian Peñaloza.

## METODOLOGÍA

Para la programación de la red neuronal y las métricas de rendimiento se utilizó la metodología de programación extrema (XP), se considera una metodología ágil y se basa en procesos iterativos de desarrollo cortos, además hace comprobaciones del código, para poder determinar los fallos y eliminarlos rápidamente. (Bendini & Guerra, 2005)

El equipo que se utilizó para el diseño, codificación, pruebas y documentación fue una computadora Dell Alienware 15 R4 con un procesador Intel core i7-8750H CPU 2.20 GHz, 16 GB de RAM y sistema operativo Windows 10 Home Single.

La aplicación fue desarrollada en el entorno Visual Studio Code en Angular 7, NodeJS y MongoDB. La red neuronal fue codificada en lenguaje Python en el entorno Pycharm, utilizando las librerías de Tensorflow, Keras, OpenCV, Scipy y Numpy. Para la detección de texto en imágenes Google Api Cloud Vision.

El aprendizaje supervisado consiste en aprender un modelo a partir de los datos que se han etiquetado para el entrenamiento y así poder dar una predicción en la clasificación de los datos. Se llama aprendizaje supervisado porque se conocen los datos de

entrenamiento ya que anteriormente se han etiquetado. Dentro del aprendizaje supervisado tenemos:

1.-La clasificación de datos que han sido etiquetados y separados por clases y puestos en entrenamiento para llegar a una predicción, como se ejemplifica en la figura 1.1.

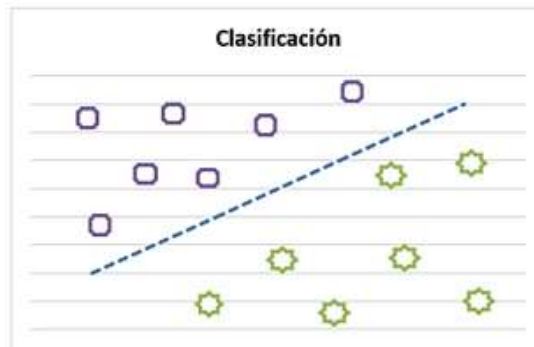


Figura 1.1: Aprendizaje Supervisado- Clasificación, (Zepeta et al, 2022).

2.- La Regresión donde existe un número de variables que predicen y una variable de respuesta, se determina la relación que pueda haber entre las variables y se llega a un resultado mediante la predicción, como se ejemplifica en la figura 1.2.

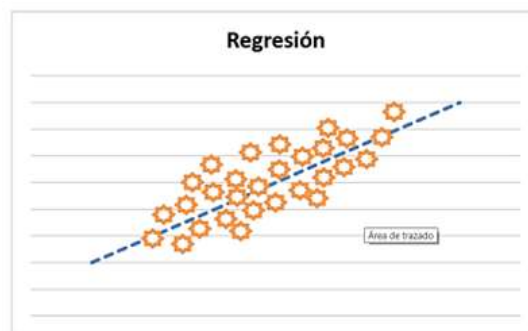


Figura 1.2: Aprendizaje Supervisado: Regresión lineal (Zepeta et al, 2022).

El aprendizaje reforzado consta de predicciones y respuestas continuas, donde se debe encontrar la relación entre estos para predecir el resultado. En este tipo de aprendizaje se define la recompensa para acciones directas mediante el agente. En la figura 1.3, se observa la relación entre el agente con su entorno/ambiente por medio de una acción y el resultado será el aprendizaje, que es una recompensa a esa acción (negativa o positiva).

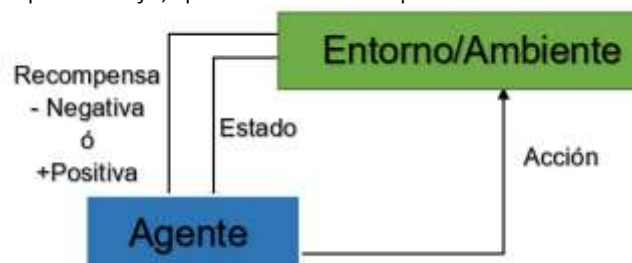


Figura 1.3: Aprendizaje Reforzado (Zepeta et al, 2022).

El aprendizaje no supervisado, se trabaja con datos sin etiquetar, esto quiere decir, que no se conoce la estructura de los datos, para ello existen técnicas que ayudan a descubrir las estructuras ocultas de los datos y una de ellas es el agrupamiento o clasificación sin supervisión, la cual es una técnica que clasifica y organiza los datos en subgrupos, estos subgrupos reciben el nombre de clúster. El clúster que van surgiendo van juntando los objetos que tienen una semejanza en distintos grupos, como se ejemplifica en la figura

1.4. Dentro del aprendizaje no supervisado esta la reducción de dimensionalidad, donde se puede eliminar el ruido en las imágenes, degradar el rendimiento predictivo de ciertos algoritmos y comprimir datos en un subespacio dimensional más pequeño, manteniendo la información más importante de los datos. (Rashka & Mirjalili, 2019)

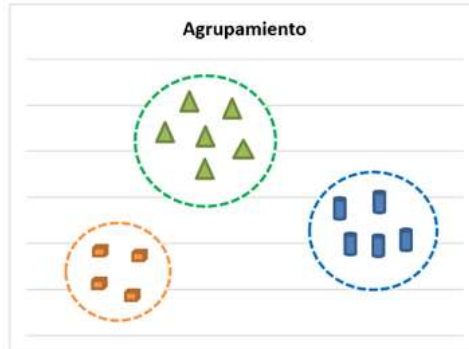


Figura 1.4 Aprendizaje sin supervisar- Agrupamiento, (Zepeta et al 2022).

Por lo anterior, para el aprendizaje automático que se uso fue de clasificación, también es de suma importancia la data (imágenes) que se ocupara para entrenamiento de la red, esta data debe ser suficiente y variada, por lo que se crean dos clases, una positiva (pozos) y una clase negativa (bimbas), como mínimo 1,000 imágenes por clase, uno de los problemas fue conseguir tantas imágenes, por lo que se procedió a aumentar la data que se tenía con `keras.preprocessing.image import ImageDataGenerator` que es una técnica de aumento de imágenes, después de tener más imágenes con diferentes ángulos, se creó el modelo de la red neuronal convolucional. En la figura 1.5, se observa que la figura de izquierda es la original (bimba) y la de la derecha ha sido modificada con el aumento de datos.

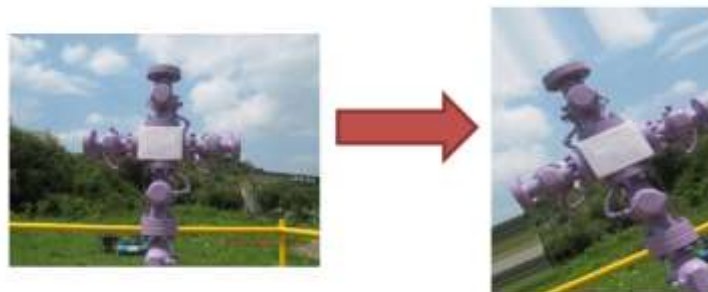


Figura 1.5: Aumento de datos, (Zepeta et al 2022).

## RESULTADOS Y DISCUSIÓN

Se realizó un etiquetado de todas las imágenes de árboles de producción y sistemas de bombeo mecánico, fueron 1098 fotos etiquetadas de cada clase para poder entrenar nuestra red. El procedimiento de etiquetado se puede observar en la imagen 1.6.

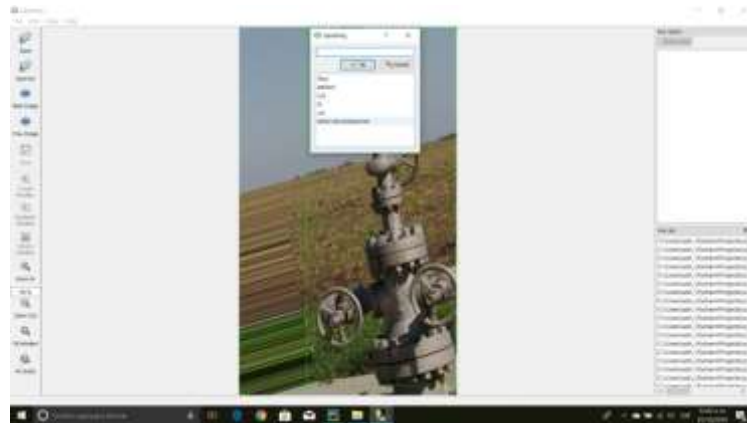


Figura 1.6: Etiquetado de imágenes, (Zepeta et al 2022).

Una vez contando con las imágenes, empieza el entrenamiento de la red neuronal. En la figura 1.7, se observa el entrenamiento de la red neuronal convolucional.

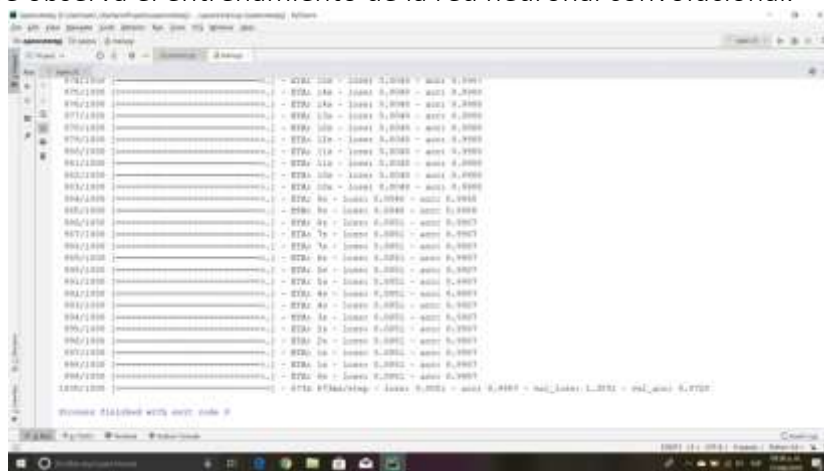


Figura 1.7: Entrenamiento de la CNN, (Zepeta et al 2022)

El siguiente paso es la predicción o verificación después de que entreno nuestra red neuronal para comprobar que lo que aprendió ha sido correcto para la detección y clasificación de imágenes de las dos clases dadas. En la imagen 1.8 y 1.9 se observa como la predicción para el sistema de bombeo mecánico artificial ha sido acertado.



Figura 1.8: Validación del entrenamiento para el sistema de bombeo mecánico artificial, (Zepeta et al 2022).

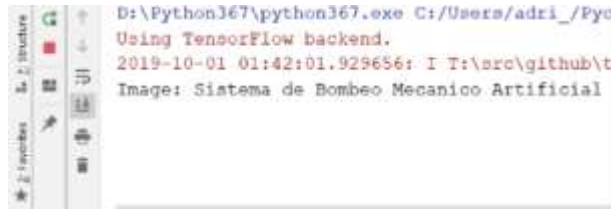


Figura 1.9: Resultado de reconocimiento de la imagen para el sistema de bombeo mecánico artificial, (Zepeta et al 2022).

La precisión de la red neuronal convolucional es de un 72% de reconocimiento de bimbos y de un 75% en el reconocimiento de imágenes de pozos petroleros, por lo que se implemento métricas de rendimiento para mejorar los porcentajes de reconocimiento. Las métricas de rendimiento se utilizan para mejorar los modelos de clasificación ya que puede haber falsas predicciones, subajustes y sobreajustes. Entre las métricas de rendimiento esta la exactitud, precisión, métrica de exhaustividad (Recall), F1\_Score, tasa de verdaderos positivos y tasa de falsos positivos. Todas estas métricas se obtienen de Scikit-learn la cual es una herramienta con algoritmos de aprendizaje de Machine Learning que ayudan en el momento de evaluar el modelo de red neuronal. Para entender cómo funcionan las métricas de evaluación se utiliza la matriz de confusión. Una matriz de confusión se representa como una matriz de 2 x 2 que da las predicciones de verdadero positivo, verdadero negativo, falso positivo y falso negativo de un clasificador de imágenes, esta matriz nos da información de los posibles errores que se pueden cometer en la clasificación de imágenes, por lo tanto, una matriz de confusión ayuda a evaluar la exactitud o precisión del clasificador, en la figura 1.10, se muestra los cuadrantes que se utilizan en la matriz de confusión para realizar la predicción.



Figura 1.10: Matriz de confusión, (Zepeta et al 2022).

La exactitud o accuracy en inglés es una métrica que ayuda a saber que tan exacto o cercano es el resultado al valor verdadero y brinda información sobre los posibles errores que se pueden encontrar en la clasificación, por lo tanto, se evalúan los modelos para obtener los resultados correctos en la clasificación de imágenes.

La exactitud es la suma de las predicciones verdaderas dividida por el número total de predicciones hechas, la exactitud se representa mediante la siguiente ecuación:

$$Exactitud = \frac{\text{Número de predicciones Verdaderas}}{\text{Número total de predicciones}}$$

$$ACC = \frac{VP + VN}{VP + VN + FP + FN}$$

Ecuación 1.1: Fórmula para medir la exactitud o accuracy, (Simón & Ruz, 2020, pág. 48).

Donde:

- VP = Verdaderos Positivos
- VN= Verdaderos Negativos
- FP = Falsos Positivos
- FN= Falsos Negativos



Para obtener el gráfico de la curva de aprendizaje del clasificador, se ejecuta el siguiente código, el cual arroja el gráfico que se muestra a continuación en la figura 1.1, donde se muestra dos gráficas, una azul, donde se muestra el porcentaje de exactitud del entrenamiento y la gráfica amarilla, muestra la exactitud de la predicción de la red neuronal.

```
# Accuracy Curve
plt.figure()
plt.plot( history.history["acc"], label = "train_acc" )
plt.plot( history.history["val_acc"], label = "val_acc" )
plt.title( "Exactitud en la Clasificación" )
plt.xlabel( "Numero total de Entrenamientos" )
plt.ylabel( "Accuracy" )
plt.legend( loc = "lower left" )
plt.show()
```

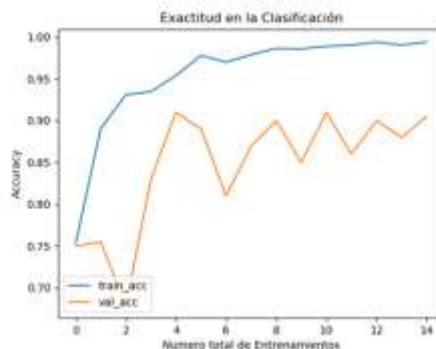


Figura 1.11: Accuracy – Curva de aprendizaje, (Zepeta et al 2022)

La precisión es una métrica más de sklearn y se basa en los positivos verdaderos y falsos positivos, se representa en la siguiente ecuación, donde se mide que tan precisas son las predicciones de la red neuronal por medio de porcentaje de predicciones acertadas.

$$PRE = \frac{VP}{VP + FP}$$

Ecuación 1.2: Fórmula para medir la precisión, (Simón & Ruz, 2020, pág. 47).

Donde:

VP = Verdaderos Positivos

FP = Falsos Positivos

Esta métrica es importante ya que la precisión asegura que el clasificador no etiquete una muestra de la clase negativa como muestra de clase positiva.

Recall es otra métrica de rendimiento que va de la mano con la precisión, puesto que las dos están relacionadas con las tasas de falsos y verdaderos, es equivalente a la tasa de verdaderos positivos. Recall es igual a todos los verdaderos positivos divididos por la suma de falsos negativos y verdaderos positivos, se representa de la siguiente manera.

$$REC = \frac{VP}{FN + VP}$$

Ecuación 1.3: Fórmula para medir el Recall, (Simón & Ruz, 2020, pág. 47).

F1\_Score es una media de precisión y Recall, f1\_score asigna el valor 1 al mejor valor y 0 al peor, se representa de la siguiente manera:

$$F1 = 2 * \frac{PRE * REC}{PRE + REC}$$

Ecuación 1.4: Fórmula para medir el Recall, (Borja, Monleón, & Rodellar, 2020, pág. 176)

Implementación de las métricas de scikit-learn precisión, Recall y F1\_Score:

```

from sklearn.metrics import precision_score
from sklearn.metrics import recall_score, f1_score

print ('Precision: %.3f' % precision_score(y_true = testY, y_pred=testY))
print ('Recall: %.3f' % recall_score(y_true = testY, y_pred=testY))
print ('F1: %.3f' % f1_score(y_true = testY, y_pred=testY))

```

También se puede sintetizar el código importando las métricas de clasificación de scikit-learn para que despliegue un reporte con las principales métricas.

```

from sklearn.metrics import classification_report

print("[INFO] Evaluacion: Reporte de clasificación...")
predictions = model.predict(testX, batch_size=BS)
print(classification_report(testY.argmax(axis=1),
    predictions.argmax(axis=1), target_names=le.classes_))

```

En la siguiente figura se observa el resultado de la evaluación utilizando las métricas de accuracy, precisión, Recall y F1\_Score.

```

[INFO] evaluating network...

```

	precision	recall	f1-score	support
bimbas	0.88	0.86	0.87	531
pozos	0.85	0.87	0.86	484
accuracy			0.87	1015
macro avg	0.86	0.87	0.86	1015
weighted avg	0.87	0.87	0.87	1015

Figura 1.12: Resultado de la evaluación usando métricas de clasificación.

De la figura 1.12, se interpretan los resultados obtenidos aplicando las métricas:

- Para reconocer bimbas, se tiene una precisión y Recall altos, por lo que la red neuronal detecta bien las imágenes de bimbas y es confiable.
- Para reconocer pozos, tiene una precisión aceptable y un Recall alto, por lo que la red neuronal detecta bien los pozos y es apenas confiable.
- El F1\_Score es en promedio del 87%, lo que se considera aceptable.

En la figura 1.13, se ejemplifica un resultado donde la red neuronal predijo correctamente la clase negativa, donde el modelo dedujo que una imagen de la clase sistema de bombeo mecánico artificial no era un árbol de producción, por lo que tenemos un verdadero negativo.



Figura 1.13: Ejemplo verdadero negativo, (Zepeta et al, 2022).

Resultado donde el modelo predijo correctamente la clase positiva, el modelo dedujo que una imagen de la clase árbol de producción es de esa misma clase y fue correcto, así como se observa la figura 1.14.



Figura 1.14: Ejemplo verdadero positivo, (Zepeta et al, 2022).

Resultado donde el modelo predijo de manera incorrecta la clase negativa. En la figura 1.15, el modelo infirió que una imagen de la clase sistema de bombeo mecánico artificial no era sistema de bombeo (la clase negativa), pero esa imagen si es un sistema de bombeo mecánico artificial.



Figura 1.15: Ejemplo falso negativo, (Zepeta et al, 2022).

En la figura 1.16, le red neuronal predijo de manera incorrecta la clase positiva, el modelo dedujo que una imagen de la clase árbol de producción (la clase positiva) era de la clase sistema de bombeo mecánico artificial (clase negativa), pero la imagen no era de la clase de sistema bombeo mecánico artificial.



Figura 1.16: Ejemplo falso positivo, (Zepeta et al, 2022).

## CONCLUSIONES

Los avances en la inteligencia artificial en el campo del aprendizaje profundo utilizan redes neuronales convolucionales para tener resultados satisfactorios, al utilizar las métricas propuestas en la red neuronal convolucional para la clasificación de imágenes, se observa una mejora en el rendimiento del algoritmo, ya que evita que la red neuronal tenga falsas predicciones al identificar patrones y texto en las imágenes.

En el ramo petrolero no existe una base de datos para poder entrenar al algoritmo para hacer la clasificación de pozos, ductos y bimbos, por lo que la data es insuficiente para llegar a que se tengan buenos resultados, pero al aplicar las métricas de rendimiento, junto con la técnica de aumento de datos (imágenes con distintas posiciones), se logra obtener una mayor precisión, exactitud y la disminución de falsas predicciones de la red neuronal convolucional.

## LISTA DE REFERENCIA

- Bae, S. (1997). *Neuronal Networkd Classifiers for Recognizing Totally Unconstrained Handwritten Numerals* (Vol. 8). IEEE.
- Baidyk, T., & Kussul, E. (2009). *Redes neuronales, visión computacional y micromecánica*. Cd. México: Itaca, UNAM.
- Basogain, X. (2000). *Redes neuronales artificiales y sus aplicaciones*. Bilbao, España: UPV-EHU.
- Bendini, A., & Guerra, L. (2005). *Gestión de proyectos de software*. Valparaiso, Chile.: Universidad Técnica Federico Santa María.
- Berzal, F. (2018). *Redes neuronales & Deep learning: Edición en color*. Independently published.
- Borja, R., Monleón, A., & Rodellar, J. (2020). *Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning*. Barcelona, España.: Revista Ibérica de Sistemas y Tecnologías de Información.

- Caicedo, E. F., & López, J. A. (2009). *Una aproximación práctica a las redes neuronales*. Santiago de Calí, Colombia: Universidad del Valle.
- Cónstenla, G. R. (2010). *Óptico de dígitos con redes neuronales*. Belgrado, Argentina: Universidad de Belgrado.
- Cun, L., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). *Handwritten digit recognition with a back-propagation network*. Holmdel, New Jersey.: AT&T Bell Laboratories.
- Dahl, J. (2019). *Leading lean: Ensuring success and developing a framework for leadership*. USA: O'Reilly Media.
- Hilera, J. R., & Martínez, V. J. (1995). *Redes Neuronales. Fundamentos, Modelos y Aplicaciones*. RA-MA.
- Howse, J. (2013). *OpenCV computer with Python*. Toronto, Canada: Packt Publishing.
- Jones, H. (2019). *Las redes neuronales: una guía esencial para principiantes de las redes neuronales artificiales y su papel en el aprendizaje automático y la inteligencia artificial*. Independently published.
- Journals. (s.f.). Neuronal Computation. *IEEE Transactions on Neuronal Networks*.
- Marínez, J. M., Becerra, A., Zepeda, G., Esparza, S., & Ramírez, U. (2017). *Sistema de reconocimiento de dígitos manuscritos utilizando redes neuronales*. Celaya: Académica Española.
- Martín, B., Medrano, N., Pollán, T., & Sanz, A. (2002). *Redes neuronales y sistemas borros*. Zaragoza, España.: Universidad de Zaragoza.
- Pratt, H., Coenen, F., Harding, D. M., & Zheng, Y. (2016). *Convolutional Neural Networks for Diabetic Retinopathy*. Computer Science. Obtenido de <https://doi.org/10.1016/j.procs.2016.07.014>
- Rashid, T. (2016). *Make your own neuronal network*. Independent Publishing.
- Rashka, S., & Mirjalili, V. (2019). *Python machine learning. Aprendizaje automático y aprendizaje profundo con Python, Scikit-learn y TensorFlow*. Wisconsin-Madison: Marcombo.
- Rodríguez, A. (18 de Agosto de 2015). *Oficina de software libre de la Universidad de la Laguna*. Obtenido de Universidad de la Laguna Web site: <https://osl.ull.es/software-libre/opencv-libreroa-vision-computador/>

Simón, A., & Ruz, R. (2020). *Machine learning algorithm evaluation on advance driver assistance*. Madrid, España.: Universidad Complutense de Madrid/Universidad de Granada (SICUE).

Vieira, V., & Palacios, M. (2017). *Aprendizaje de máquina. Redes neuronales*. México: Académica Española.