

# Low-Cost Remote Reading of Analog Electric Meters Using Computer Vision and Neural Networks

David Juan Fuentes-Maza, Ingeniero de Telecomunicaciones<sup>1</sup> , Harold Sebastián Michael Castro-Apaza<sup>2</sup> ; Isahi Harvey Aranda-Utcañe<sup>3</sup> ; Enrique Taikin Linarez-Wong<sup>4</sup> 

<sup>1,2,3,4</sup> Universidad Tecnológica del Perú, Perú, c25670@utp.edu.pe, U19303879@utp.edu.pe, U21309656@utp.edu.pe, U21222557@utp.edu.pe

**Abstract**– This article presents the design and implementation of an automated remote reading system for analog electricity meters, based on computer vision and artificial neural networks. The ESP32-CAM module is used as the capture unit, evaluating three optical configurations of the OV2640 sensor (66°, 120°, and 120° adjustable) to identify the most suitable in terms of sharpness and coverage. The obtained images are processed using binarization, segmentation, and normalization techniques and input into a neural network trained with the MNIST dataset for automatic digit recognition. The results show that the 120° adjustable variant at a distance of 5 cm offers greater capture accuracy. The neural model achieved an accuracy of 96.18% when evaluated with external images, demonstrating its generalization capabilities in real-world conditions. This proposal represents a low-cost, replicable, and technically viable solution for environments where analog meters are still used, contributing to energy automation in contexts with limited infrastructure.

**Keywords:** Machine vision, ESP32-CAM, remote reading, neural networks, energy automation.

# Lectura remota de bajo costo de medidores analógicos mediante visión artificial y redes neuronales

David Juan Fuentes-Maza, Ingeniero de Telecomunicaciones<sup>1</sup> , Harold Sebastián Michael Castro-Apaza<sup>2</sup> ; Isahi Harvey Aranda-Utcañe<sup>3</sup> ; Enrique Taikin Linarez-Wong<sup>4</sup> 

<sup>1,2,3,4</sup> Universidad Tecnológica del Perú, Perú, c25670@utp.edu.pe, U19303879@utp.edu.pe, U21309656@utp.edu.pe, U21222557@utp.edu.pe

**Resumen—** Este artículo presenta el diseño e implementación de un sistema automatizado de lectura remota para medidores eléctricos analógicos, basado en visión artificial y redes neuronales artificiales. El módulo ESP32-CAM se utiliza como unidad de captura, evaluando tres configuraciones ópticas del sensor OV2640 (66°, 120° y 120° ajustable) para identificar la más adecuada en términos de nitidez y cobertura. Las imágenes obtenidas se procesan mediante técnicas de binarización, segmentación y normalización, y se introducen en una red neuronal entrenada con el conjunto de datos MNIST para el reconocimiento automático de dígitos. Los resultados muestran que la variante ajustable de 120° a una distancia de 5 cm ofrece una mayor precisión de captura. El modelo neuronal alcanzó una precisión del 96,18% al evaluarse con imágenes externas, demostrando su capacidad de generalización en condiciones reales. Esta propuesta representa una solución económica, replicable y técnicamente viable para entornos donde aún se utilizan medidores analógicos, contribuyendo a la automatización energética en contextos con infraestructura limitada.

**Palabras clave:** Visión artificial, ESP32-CAM, lectura remota, redes neuronales, automatización energética.

## I. INTRODUCCIÓN

En el Perú, en las últimas décadas, la industria energética enfrenta desafíos como la falta de personal capacitado y la alta inversión inicial para implementar nuevas tecnologías, debido a la necesidad de optimizar procesos y reducir costos [1]. Por lo tanto, se destaca la relevancia de la investigación e implementación de redes eléctricas inteligentes y de muy bajo costo en el país [2].

Es importante resaltar el papel que juegan las tecnologías emergentes de medición en el sector energético ejemplo la lectura remota de medidores (Automated Meter Reading, AMR) constituye un pilar fundamental las redes inteligentes (Smart Grids) [3], aquí, la empresas de energía coinciden en que éstas técnicas son esenciales para un suministro confiable, contribuyendo en la toma de decisiones para el uso de energía, reducir pérdidas y mejorar la seguridad [4].

En la mayoría de distritos del Perú, las lecturas manuales de medidores de energía eléctrica no son del todo exactas, debido a que los encargados de tomar estas lecturas pueden cometer errores por diversas razones, lo que afecta la precisión de la información sobre el consumo de energía [5]. A pesar de los esfuerzos de las empresas distribuidoras para reducir estos índices, persisten deficiencias en la regulación y en la

implementación de soluciones avanzadas, como la telemedición [6][7]. En el caso de Electronoroeste S.A., concesionaria peruana a cargo de toda la zona norte de Lima y varias regiones, por ejemplo, las pérdidas en transmisión durante 2020 alcanzaron el 3.26%, mientras que en distribución superaron el 10.99%, evidenciando la necesidad de estrategias más eficaces para mitigar estos valores [8]. En América Latina y el Caribe, naciones como Haití y Paraguay evidencian niveles críticos de pérdidas eléctricas, alcanzando 60.12% y 20.82%, respectivamente. De manera similar, en otras regiones del mundo como África y Asia la situación sigue siendo preocupante, con Argelia y Nepal registrando pérdidas del 17.12% y 32.21%. Estos valores ponen de manifiesto la urgencia de implementar tecnologías innovadoras y estrategias de optimización orientadas a disminuir las pérdidas energéticas y fortalecer la eficiencia de los sistemas eléctricos [8], [9].

Como una solución alternativa surgen las tecnologías de telemedición que eliminan errores manuales y reduce costos operativos mediante monitoreo en tiempo real, crucial para previsión de demanda y detección de fraudes [10]. Si bien ofrece ventajas como precisión en facturación y optimización de redes, enfrenta desafíos como alta inversión inicial, requerimientos de infraestructura robusta y preocupaciones de seguridad [11]. Es aquí el uso del ESP32-CAM emergen como soluciones costo-efectivas para implementación, combinando capacidades inalámbricas y procesamiento de imágenes [12] de muy bajo costo combinando el uso de la la inteligencia artificial (IA) que permite simular procesos cognitivos humanos para automatizar tareas y reducir errores [6][13]. Entre sus técnicas destacan las redes neuronales artificiales (RNA), utilizadas en reconocimiento de imágenes y análisis de datos complejos [6]. Actualmente el desarrollo de sistemas automatizados integra IA y visión por computadora, utilizando Redes Neuronales Artificiales (RNAs) para reconocimiento automático de dígitos en medidores analógicos/digitales [14]. Países pioneros como EE. UU., Japón y Alemania lideran implementaciones a gran escala, con aplicaciones prácticas en facturación prepago, gestión de demanda y detección de fallas [15]. Diversas investigaciones respaldan el uso de tecnologías inteligentes y técnicas de visión artificial para automatizar la lectura de medidores. Por ejemplo, una solución de bajo costo y eficiencia energética fue implementada combinando cámaras, microcontroladores, redes NB-IoT y aprendizaje federado, alcanzando una precisión del 97.01% en el reconocimiento de dígitos, sin sobrecargar los dispositivos de borde [16]. Otro estudio propuso un sistema

automatizado basado en visión por computadora y redes neuronales artificiales, diseñado para capturar e interpretar imágenes de medidores eléctricos, abordando desafíos como la variabilidad en el ángulo de captura y la baja resolución de las imágenes, aspectos también abordados en la presente investigación [17].

Adicionalmente, se han desarrollado enfoques basados en procesamiento de imágenes que emplean técnicas como binarización, segmentación y comparación de plantillas, los cuales permiten identificar automáticamente los dígitos de los medidores, reduciendo errores humanos y mejorando la eficiencia en la facturación y el monitoreo del consumo eléctrico [18]. Asimismo, se ha implementado un sistema multi-red neuronal para la lectura de medidores de gas, que utiliza algoritmos de segmentación y reconocimiento de texto sobre imágenes reales demostrando alta precisión incluso en condiciones visuales adversas, reforzando la viabilidad de las soluciones basadas en redes neuronales en contextos no controlados [19]. Estas estadísticas y estudios respaldan integrar un sistema basado en visión por computadora para la lectura remota de medidores. Este sistema emplea cámaras y procesamiento de imágenes para extraer los valores de consumo de los medidores y transmitirlos de forma inalámbrica a una. La tecnología presentada en este trabajo permite automatizar la recolección de datos de medidores eléctricos, mejorando la eficiencia y precisión del proceso [20]. Asimismo, para garantizar el cumplimiento de este objetivo evaluar la calidad y adecuación de las imágenes capturadas por el ESP32-CAM, comparando distintas configuraciones ópticas para su uso en sistemas de visión artificial y analizar el rendimiento del modelo de red neuronal en la identificación automática de dígitos, considerando precisión de lectura y eficiencia computacional.

## II. METODOLOGÍA

La investigación es de tipo aplicada y de diseño experimental orientada a la implementación de un sistema de visión artificial para la lectura de medidores analógicos. El sistema de captura utiliza el módulo ESP32-CAM por su conectividad Wi-Fi y compatibilidad con el sensor OV2640.

La variable independiente es la configuración de la cámara, evaluando tres opciones: lente de 66°, lente de 120° y lente ajustable de 120°. Por su parte, la variable dependiente es la calidad de la imagen resultante y la precisión del modelo neuronal.

Se utilizaron como instrumentos un medidor analógico de electricidad, empleado comúnmente en la lectura de consumo en zonas sin telemedición en Lima, y una estación fotográfica fija impresa en 3D para asegurar la estabilidad y alineación de la cámara. Además, se implementó un sistema de iluminación artificial para optimizar la visibilidad durante el día, ajustando manualmente el nivel de luz según las condiciones.

Para el procesamiento y análisis de datos se utilizaron Jupyter

Notebook (Anaconda) y Visual Studio Code, facilitando la organización y ejecución del código. Se emplearon TensorFlow y Keras para el entrenamiento de la red neuronal, así como OpenCV para el procesamiento de imágenes y NumPy para el manejo de matrices, permitiendo la predicción automática de los dígitos.

Las imágenes capturadas por la ESP32-CAM se enviaron a un servidor local, donde fueron procesadas mediante binarización, segmentación y normalización a 28×28 píxeles. Se utilizó el dataset externo MNIST, compuesto por 70 000 imágenes de dígitos (60 000 de entrenamiento y 10 000 de prueba), en escala de grises y tamaño 28×28 píxeles, obtenido de bases de datos públicas. Además, se empleó un conjunto de imágenes reales del medidor para entrenar y validar la red neuronal.

El análisis se centró en la evaluación de dos aspectos clave: la calidad de las imágenes capturadas y el desempeño del modelo de red neuronal (accuracy, loss y tasa de error). Se compararon los resultados obtenidos por cada configuración óptica del ESP32-CAM, determinando cuál ofrecía mejores condiciones para la identificación de dígitos.

## III. DISEÑO

El microcontrolador ESP32-CAM, incorpora una cámara OV2640 para la captura de imágenes. El tamaño del lente del microcontrolador varía de acuerdo con el ángulo de visión del lente que se ha de utilizar (Fig.1), por lo que es necesario un soporte adecuado para su fijación y estabilidad.

Se evaluaron distintas configuraciones del ESP32-CAM con cámara OV2640, variando el ángulo de visión del lente. A partir de observaciones en laboratorio, se seleccionó la configuración que ofreció mayor nitidez y resolución de imagen. Esta elección permitió definir la cámara más adecuada para el procesamiento y entrenamiento de la red neuronal.

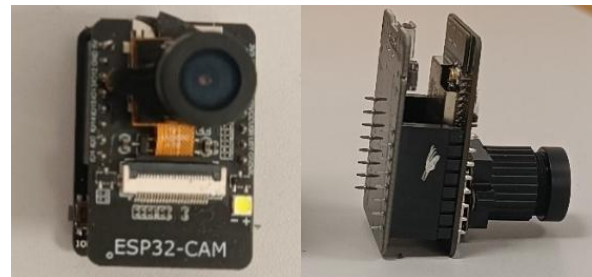


Fig. 1. ESP32-CAM acoplado con ángulo de visión de 120°

En la figura 2 se presenta un diseño 3D que corresponde a la base estructural donde se empuja el ESP32-CAM, concebido para garantizar la estabilidad y precisión en la captura de imágenes. La estructura se adapta de forma óptima al dispositivo, brindando soporte seguro a la cámara OV2640. Su diseño alineado permite una integración estable con la estación fotográfica, garantizando un posicionamiento fijo que asegura

Technical drawings of a mechanical part, showing a cross-section A-A and a side view B-B. The part is a rectangular block with a central slot and a smaller rectangular feature on the top surface. Dimensions are provided in millimeters.

**View A (1:1):** A cross-section of the part, showing a central slot with a width of 10.00 mm and a depth of 10.00 mm. The top surface has a rectangular feature with a width of 10.00 mm and a height of 10.00 mm. The bottom surface has a rectangular feature with a width of 10.00 mm and a height of 10.00 mm. The overall width of the part is 41.00 mm. The overall height of the part is 10.00 mm. The overall length of the part is 10.00 mm.

**View B (2:1):** A side view of the part, showing a central slot with a width of 10.00 mm and a depth of 10.00 mm. The top surface has a rectangular feature with a width of 10.00 mm and a height of 10.00 mm. The bottom surface has a rectangular feature with a width of 10.00 mm and a height of 10.00 mm. The overall width of the part is 41.00 mm. The overall height of the part is 10.00 mm. The overall length of the part is 10.00 mm.

El medidor eléctrico analógico es de la marca EDELNOR S.A., este se caracteriza por contar con una pantalla de dígitos mecánicos que muestra el consumo acumulado de electricidad en kilovatios hora (kWh). Muy usado en zonas rurales y de telemedición nula. A este medidor se acopló y adaptó espárragos de 3/16 pulgadas en sus laterales (Fig.3), que actúan como puntos de anclaje para fijar de manera estable la base estructural impresa en 3D y ajustar la distancia de enfoque de la Cámara a la numeración, sostiene el microcontrolador y el módulo.



### A. Prueba de calidad de imagen con el ángulo de visión del módulo OV2640

Se utilizó una cámara OV2640 con ángulo de visión de  $66^\circ$  a una distancia de 7 a 9 cm, pero la calidad de imagen no fue suficiente para una lectura precisa. Se observó que reducir la distancia a entre 5 y 7 cm mejora la nitidez, ajustándose mejor a las dimensiones del medidor y permitiendo una captura más clara y consistente de los dígitos. Es aquí donde la flexibilidad del soporte fotográfico es de suma utilidad para este y el resto de los escenarios.



El ESP32-CAM, al ser utilizado con su cámara predeterminada, presenta ciertas limitaciones en términos de calidad de imagen.



Las imágenes obtenidas a través de esta cámara tienen una resolución deficiente, no son ideales cuando se requiere una mayor claridad o un campo de visión más amplio.



### A.2) Ángulo de visión de $120^\circ$

En comparación, al utilizar el ESP32-CAM con una cámara OV2640 de 120° bajo condiciones similares, la mejora en la calidad de las imágenes es considerable.



Fig.8.- OV2640 120°





Fig.9. Foto obtenida con OV2640 120° a una distancia 6 cm

### A.3) Ángulo de visión de 120° ajustable

Finalmente, el módulo OV2640 con lente de 120° ajustable ofreció mejores resultados a una distancia de 5 cm, permitiendo un enfoque más claro sobre los dígitos. Esta proximidad optimiza la calidad de imagen sin distorsiones, facilitando una lectura precisa mediante visión por computadora, además de ocupar menos espacio debido a estar más cerca al contador. Da visos a ser replicable en escenarios reales a un futuro kit que no ocupará mucho espacio por la distancia tan corta de tomado de imágenes.



Fig.10. Ajuste de OV2640 con un ángulo de visión de 120° ajustable

La cámara OV2640 de 120° ajustable permite modificar manualmente la distancia focal, lo que facilita capturar un área amplia con mejor nivel de detalle. Esta capacidad mejora la visibilidad de los dígitos y aporta flexibilidad al sistema, permitiendo adaptarse a distintas condiciones de monitoreo.



Fig.11. OV2640 120° ajustable



Fig.12. Foto obtenida con OV2640 120° ajustable a una distancia 5 cm

### B. Interacción de elementos del sistema

Se establece una conexión Wi-Fi, el ESP32-CAM genera un servidor HTTP local, al cual se puede acceder remotamente para capturar una fotografía del medidor en tiempo real. La

imagen obtenida es posteriormente descargada por un sistema externo y ejecutadas en Jupyter. Una vez descargada, la imagen es procesada mediante una red neuronal artificial previamente entrenada, con el objetivo de identificar y predecir los dígitos visibles en el medidor.

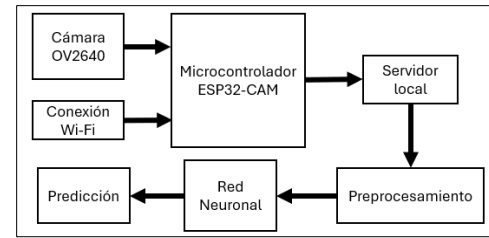


Fig.13. Diagrama de bloques de operación del sistema

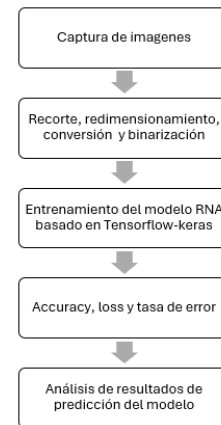


Fig.14. Etapas del modelo RNA

## IV. RESULTADOS

Las imágenes fueron cargadas para su preprocesamiento. Primero, se convirtieron a escala de grises para simplificar el análisis y reducir el ruido visual. Luego, se aplicó un umbral binario para separar las cifras del fondo, facilitando la detección de las ruedas numéricas (dígitos del medidor).



Fig.15. Binarización mediante umbral y OTSU

Tras la binarización, se extrajeron los contornos más relevantes de la imagen, permitiendo identificar y recortar las seis cifras visibles del medidor: cinco correspondientes a los kilovatios hora enteros, y una cifra decimal adicional. Para mejorar la limpieza visual de cada cifra y evitar fragmentación en los bordes, se aplicó un cierre morfológico sobre cada región detectada.



Fig.16. Región de interés

Cada cifra fue individualmente redimensionada a 28x28 píxeles, centrada sobre un lienzo del mismo tamaño y normalizada para que su intensidad de píxeles fuese adecuada al modelo de red neuronal. Estas imágenes procesadas fueron almacenadas en un conjunto y quedaron listas para ser ingresadas al modelo RNA.



Fig.18. Manchas detectadas y redimensionadas

Para la identificación automática de los dígitos en el medidor de luz, el uso del conjunto de datos MNIST, un estándar en visión por computadora que ofrece 70 000 imágenes en escala de grises de 28x28 píxeles, cada una asociada a una etiqueta entera del 0 al 9.

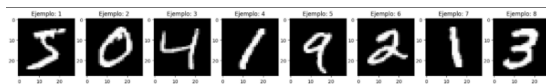


Fig.19. Muestra de números de 28x28 píxeles

Se emplea la función “mnist.load\_data()” de TensorFlow, se cargó directamente los tensores en 4 grupos: imagenes\_train y etiquetas\_train para entrenamiento, y imagenes\_ent y etiquetas\_ent para prueba. A fin de validar la robustez del modelo, se dividió el subconjunto de prueba en dos mitades con “train\_test\_split” con un “test\_size=0.5”, obteniendo 5 000 ejemplos para validación y 5 000 para testeo final.

```

1 #Extracción de datos en 2 grupos (ENTRENAMIENTO Y TEST)
2 (image_train, label_train), (image_testi, label_testi) = mnist.load_data()
3
4 from sklearn.model_selection import train_test_split
5
6 # DIVISIONE DATOS DE TESTEADO EN TESTEADO Y VALIDACION
7 image_test, image_val, label_test, label_val = train_test_split(image_testi, label_testi, test_size=0.5, random_state=42)
8
9
10 Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11 11490434/11490434 0% 0s/step

```

Fig.20. Muestra la distribución de los datos

Se normalizan los valores de píxel a un rango [0, 1] mediante “astype” y convertimos cada imagen 2D en un vector 1D de 784 elementos con “reshape”, de modo que la capa de entrada acepte un único tensor de 784 elementos. Las etiquetas se mantienen como enteros usar la función de pérdida “SparseCategoricalCrossentropy”, optimizada para etiquetas discretas y que internamente aplica “softmax” a las salidas sin necesidad de una capa adicional de normalización.

```

[9] 1 # NORMALIZACION DE DATOS (VALORES ENTRE 0 Y 1) primer intento
2 image_train_prep1 = image_train.reshape((60000, 28*28))
3 image_train_prep2 = image_train_prep1.astype('float32') / 255.0
4
5 image_test_prep1 = image_test.reshape((5000, 28*28))
6 image_test_prep2 = image_test_prep1.astype('float32') / 255.0
7
8 image_val_prep1 = image_val.reshape((5000, 28*28))
9 image_val_prep2 = image_val_prep1.astype('float32') / 255.0

```

Fig.21.-Normalización de los datos

### C) Arquitectura de la RNA

Se define en “model\_builder(hp)” dos hiperparámetros con un numero de TLUs limitado entre 32-512 neuronas con un paso de 32 neuronas y una tasa de aprendizaje de: 0.1, 0.01 y 0.001. El tuner evaluó decenas de configuraciones a lo largo de hasta 10 épocas por prueba, usando “tuner.search” con los grupos de datos de entrenamiento, 5 épocas y la validación del grupo de datos de validación. Una vez identificados los mejores hiperparámetros, reconstruimos el modelo con “tuner.hypermodel.build(best\_hps)” y procedimos a un ajuste final con 10 épocas completas.

```

1 tuner = kt.Hyperband(
2     model_builder,
3     objective = 'val_accuracy',
4     max_epochs = 10,
5     factor = 3,
6     directory = 'test_dir',
7     project_name = 'hp_tuning',
8 )

```

Fig.22. Parámetros del modelo de Keras\_Turner en 3 cuerpos: Model\_buidler, Hyperparametros y Turner.search

Se crea una instancia del modelo secuencial mediante network = models.Sequential con tres capas densas (Dense) al modelo. La primera capa densa contiene 448 neuronas y utiliza la función de activación ReLU (Rectified Linear Unit) con una entrada establecida mediante input\_shape=(28\*28,) para elementos aplanados en un vector de 784 elementos. La segunda capa también es una capa densa, con 224 neuronas empleando la función de activación ReLU, con el objetivo de seguir capturando patrones no lineales dentro de los datos.

La tercera y última capa contiene 10 neuronas función de activación softmax,

```

1 network = models.Sequential()
2
3 network.add(layers.Dense(448, activation='relu', input_shape=(28*28,)))
4 network.add(layers.Dense(224, activation='relu'))
5 network.add(layers.Dense(10, activation='softmax'))

```

Fig.23. Construcción de capas y especificación de cantidad de neuronas

El modelo se compila usando el método compile(). En este paso se especifican tres elementos clave: el optimizador, la función de pérdida (loss) y las métricas de evaluación. Para el optimizador se utiliza el algoritmo Adam (Adaptive Moment Estimation), donde se establece una tasa de aprendizaje fija de 0.001. La función de pérdida utilizada es CategoricalCrossentropy, que es la más adecuada para problemas de clasificación multiclase con etiquetas categóricas codificadas en one-hot encoding. Las métricas de precisión (accuracy), se incluye explícitamente la métrica de precisión.

```

1 from tensorflow.keras import optimizers
2
3 opt_func = optimizers.Adam(learning_rate= 0.001)
4
5
6 network.compile(
7     optimizer=opt_func,
8     loss=keras.losses.CategoricalCrossentropy(from_logits=True),
9     metrics=['accuracy', 'Precision']
10 )

```

Fig.24. Definición de arquitectura del compilador y métricas

Una vez compilado, el modelo se entrena utilizando el método `fit()`. En este proceso, se proporciona como entrada los datos de entrenamiento previamente procesados. Las imágenes (`image_train_prep2`) han sido normalizadas y aplanadas. Las etiquetas (`label_train_prep`) han sido codificadas en formato categórico. El entrenamiento se realiza durante 40 épocas (`epochs=40`) y un `batch_size=540` indica que los datos se procesarán en bloques de 540 imágenes por vez, lo que permite aprovechar mejor los recursos computacionales y estabilizar el descenso del gradiente. También se utiliza el `EarlyStopping()` para detener las iteraciones y llegar a un aprendizaje de la red equilibrada con correcta validación del error.

Finalmente, se incluye también un conjunto de datos de validación mediante el argumento `validation_data=(image_val_prep2, label_val_prep)`. Este conjunto se utiliza para monitorear el desempeño del modelo en datos no vistos durante el entrenamiento, lo que permite detectar posibles problemas de sobreajuste (*overfitting*).

El objeto devuelto por `fit()` se almacena en la variable `history`, que contiene información detallada sobre la evolución de las métricas y la pérdida durante el entrenamiento y validación.

```
1 from tensorflow.keras import optimizers
2
3 opt_func = optimizers.Adam(learning_rate= 0.001)

1 network.compile(
2     optimizer=opt_func,
3     loss=keras.losses.CategoricalCrossentropy(from_logits=True),
4     metrics=['accuracy', 'Precision']
5 )

1 from tensorflow.keras.callbacks import EarlyStopping
```

Fig.25. Definición las variables de entrada y validación de la RNA

Mediante la importación de *pandas* y *matplotlib.pyplot* visualizar las gráficas de comparación en un plot de (10,6) y llamando a la variable `history` que contine al entrenamiento `fit()`. Las gráficas entre `loss / val_loss` y `accuracy / val_accuracy` muestran la variación de sus determinados grados de aprendizaje a lo largo de cada iteración. La validación de perdida decrece y estabiliza en 0.1 (Fig.26), y la validación de predicción aumenta y se estabiliza en un 0.98 (Fig.27).

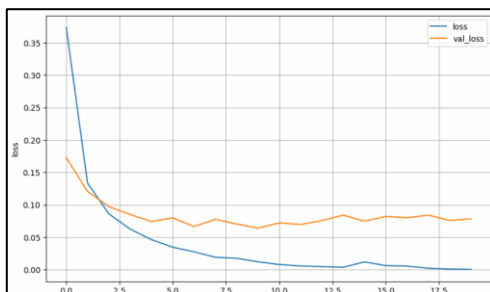


Fig.26. Pérdida durante el entrenamiento

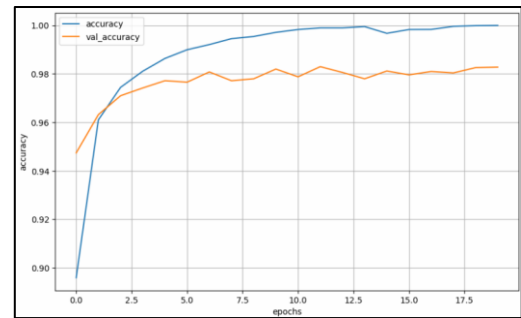


Fig.27. Predicción durante el entrenamiento

Entrenado y validado el modelo de red neuronal, es fundamental evaluar su desempeño en condiciones más cercanas al entorno real de aplicación, utilizando imágenes externas o nuevas que no hayan sido vistas durante el entrenamiento ni la validación.

Se implementa un sistema automatizado de prueba sobre un conjunto de imágenes almacenadas en una estructura de carpetas. Cada carpeta representa una clase numérica específica (del 0 al 9), y en su interior contiene imágenes segmentadas que corresponden a esa clase. Estas imágenes fueron extraídas previamente del medidor. El primer paso consiste en importar las bibliotecas necesarias: `Os`, `cv2`, `numpy` y `tensorflow.keras.models.load_model`.

El modelo se carga desde el archivo `keras`, que contiene los pesos y la arquitectura entrenados en fases anteriores y se define la ruta principal `base_folder = "REC_NUMS"`, la cual contiene las carpetas clasificadas por dígito.

```
import os
import cv2
import numpy as np
from tensorflow.keras.models import load_model

# CONFIGURACIÓN
modelo_path = "modelo_aranda_2.keras"
base_folder = "REC_NUMS" # Ruta donde están las carpetas por clase
img_size = (28, 28) # Tamaño esperado por tu modelo

# CARGAR MODELO
model = load_model(modelo_path)
```

Fig.28. Configuración y carga del modelo Keras

A continuación, se inicializan las listas `y_true` y `y_pred` para almacenar respectivamente las etiquetas reales y las predicciones realizadas por el modelo. Esto permite calcular posteriormente métricas de desempeño como la precisión.

Se aplica el modelo mediante `model.predict(img)`, obteniendo una probabilidad para cada una de las 10 posibles clases.

```
#Lista decomparacion
y_true = []
y_pred = []

# RECORRER CARPETAS
total = 0
correctas = 0

for etiqueta_carpeta in sorted(os.listdir(base_folder)):
    carpeta_clase = os.path.join(base_folder, etiqueta_carpeta)
    if not os.path.isdir(carpeta_clase):
        continue
```

Fig.29. Definición de variables para almacenar etiquetas y predicciones

Luego se usa `np.argmax()` para obtener el índice (es decir, la clase numérica) con mayor probabilidad, que se toma como la predicción final. Se muestra en consola el resultado individual de cada imagen, incluyendo su nombre de archivo, la etiqueta real y la predicción generada por el modelo, lo que permite tener una trazabilidad detallada.

Una vez procesadas todas las imágenes, se calcula la precisión global del modelo como la razón entre el número de predicciones correctas y el total de imágenes evaluadas, expresada como porcentaje.

```
# Predicción
pred = np.argmax(model.predict(img), axis=-1)[0]
verdadera = int(etiqueta_carpeta)

y_true.append(verdadera)
y_pred.append(pred)

print(f"archivo → Etiqueta real: {verdadera}, Predicción: {pred}")

total += 1
if pred == verdadera:
    correctas += 1
```

Fig.30. Normalización de las imágenes después de la predicción

En esta ejecución particular, el modelo evaluó un total de 275 imágenes y acertó en 264 de ellas, lo que resulta en una precisión del 96.18%. Esta métrica es muy alentadora considerando que las imágenes externas probablemente poseen ruido, distorsión, variaciones de iluminación o escritura más irregular que las imágenes limpias y centradas del conjunto MNIST original.

```
1/1 0s 41ms/step
captura_20250706_162143_959602_mancha_1.png → Etiqueta real: 9, Predicción: 5

Total imágenes evaluadas: 275
Predicciones correctas: 264
Precisión: 96.18%
```

Fig.31. Porcentaje de predicción con fotos procesadas del medidor

#### D. Sistema de energizado y funcionamiento en campo

El sistema está diseñado para operar de forma autónoma sin conexión a la red eléctrica, utilizando una batería de litio tipo Li-SOCl<sub>2</sub>, caracterizada por su baja autodescarga y larga vida útil. Estas baterías se emplean en telemetría y medición inteligente, donde el consumo es mínimo y solo se generan picos breves durante la transmisión de datos.

El funcionamiento se basa en la técnica de power-gating, que corta totalmente la alimentación del microcontrolador

cuando no se usa, manteniendo activo solo un reloj en tiempo real (RTC DS3231) con consumo del orden de 110  $\mu$ A. Este despierta al sistema una vez al mes para realizar la captura de imagen y el registro de la lectura, logrando así un consumo mensual muy bajo y una autonomía de varios años con una sola celda.

En este escenario, el consumo mensual total estimado se obtiene sumando el gasto en reposo del RTC y el breve consumo del módulo ESP32-CAM durante la toma de medición:

$$Q_{sleep} = 0.11mA \times 720 h = 79 mAh/mes$$

$$Q_{total} = Q_{sleep} \times 2.5mAh = 81.5 mAh/mes$$

Utilizando una batería litio 3.6V 19000mAh Li-SOCl<sub>2</sub> tamaño D R20 PKCELL ER34615 con un costo de 24 dólares [22]., la autonomía teórica del sistema alcanza:

$$Autonomía = \frac{19000 mAh}{81.5 mAh/mes} = 233 meses = 19 años (teórico)$$

Sin embargo, en un escenario real, la autonomía de 19 años es un límite teórico derivado de supuestos ideales (consumo en reposo casi nulo, un único despertar breve al mes y capacidad nominal íntegra). En campo intervienen factores que elevan el consumo y reducen la capacidad útil: autodescarga y vida de calendario de la Li-SOCl<sub>2</sub>, fugas de reguladores/conmutación, deriva del RTC, temperatura, reintentos de Wi-Fi y variaciones de tiempo activo.

Bajo estas nuevas consideraciones, usamos supuestos de fuga de 200  $\mu$ A para el estado de reposo con el RTC. La activación mensual que involucra captura y transmisión con una duración de 60 segundos consumiría 300 mA [21]. Bajo un mes estándar de 720 horas tenemos.

$$Q_{act} = 300 mA \times 0.0167h = 5 mAh/mes$$

De esta consideración de fuga se tendría:

$$Q_{sleep} = 0.2mA \times 720 h = 144 mAh/mes$$

$$Q_{total} = Q_{sleep} + Q_{act} = 149 mAh/mes$$

Finalmente tenemos la autonomía con batería Li-SOCl<sub>2</sub>:

$$Autonomía = \frac{19000 mAh}{149 mAh/mes} = 127 meses = 10.5 años$$

Aunque el cálculo con supuestos máximos entrega 10.5 años, en campo la vida de calendario de la celda y las pérdidas adicionales (temperatura, autodescarga, reintentos de enlace, reguladores) reducen la autonomía. Aplicando un margen conservador (derating 40%), la autonomía esperable queda entre 5 años con lecturas mensuales. Este cambio de batería se programaría en los mantenimientos preventivos o correctivos por la empresa eléctrica.



TABLA I. COMPARATIVA DE CONSUMO Y AUTONOMÍA

| Parámetro                               | Escenario Ideal (Teórico) | Escenario Realista (Campo) |
|-----------------------------------------|---------------------------|----------------------------|
| Corriente en reposo (Sleep)             | 110 $\mu$ A               | 200 $\mu$ A                |
| Corriente activa (Transmisión)          | 300 mA                    | 300 mA                     |
| Consumo total mensual                   | 81.5 mAh/mes              | 149 mAh/mes                |
| Factor de reducción ( <i>Derating</i> ) | 0%                        | 40%                        |
| Autonomía estimada                      | 19.3 años                 | 5 años                     |

## V. DISCUSIÓN

Los resultados obtenidos demuestran que el Sistema Automatizado de Lectura Remota (SALR) alcanzó una precisión del 96.18 % en la identificación de cifras en medidores analógicos, empleando imágenes reales capturadas con el módulo ESP32-CAM y procesadas mediante redes neuronales artificiales. A diferencia de trabajos que utilizan conjuntos de datos sintéticos o controlados, el presente estudio se basó en fotografías obtenidas en condiciones reales, lo que refuerza su validez externa y aplicabilidad práctica.

En comparación con el estudio de Nikić et al. [16], que reportó una precisión del 97.01 % utilizando NB-IoT y aprendizaje federado, el SALR se diferencia al enfocarse en medidores analógicos convencionales, que aún constituyen la mayoría en países latinoamericanos. Este enfoque lo convierte en una solución más económica y fácilmente desplegable, adecuada para zonas donde la infraestructura de medición inteligente aún no está disponible.

Por otro lado, en contraste con Yin et al. [17], quienes propusieron un sistema basado en cámaras optimizado para corrección angular y resolución, este trabajo incorporó un análisis comparativo de tres configuraciones ópticas (66°, 120° y 120° ajustable). Se determinó que una distancia de captura reducida de 5 cm y el uso de lente ajustable mejoran la nitidez y estabilidad de la lectura. Este ajuste estructural constituye una ventaja técnica en contextos con espacio reducido o iluminación variable.

Asimismo, frente al trabajo de Elrefaei et al. [18], que recurrió a técnicas de binarización y segmentación, el SALR integra estos procesos dentro del propio modelo neuronal, simplificando el flujo de procesamiento. En comparación con Vanetti et al. [19], cuyo sistema multicapa demandaba alta capacidad computacional, el SALR adopta una arquitectura ligera optimizada para microcontroladores, manteniendo un equilibrio entre precisión, consumo y costo.

Desde la perspectiva tecnológica y económica, el prototipo propuesto demuestra un costo total inferior a S/ 400 (USD 100), incluyendo cámara, soporte impreso y alimentación autónoma. Esto representa un ahorro de más del 80 % respecto a soluciones comerciales como Arduino Nicla Vision [22] (USD 115) o módulos FPGA especializados [21].

En términos energéticos, la incorporación de una batería de litio Li-SOCl<sub>2</sub> PKCELL ER34615 (3.6 V, 19000 mAh) [23] permite una operación prolongada gracias al modo de bajo consumo (deep sleep). Considerando una corriente en reposo de 200  $\mu$ A y un consumo de 300 mA mensuales durante los ciclos de captura, la autonomía teórica estimada se sitúa entre 5 y 10 años, dependiendo del entorno y la temperatura. Este resultado refuerza la viabilidad del sistema en zonas rurales o de difícil acceso, sin requerir conexión directa a la red eléctrica.

## VI. CONCLUSIONES

En base a los hallazgos, el desarrollo y diseño del SALR permitió validar la viabilidad de integrar visión por computadora, redes neuronales y el módulo ESP32-CAM para automatizar la lectura de medidores analógicos. Se determinó que la distancia óptima de captura es de cinco a siete centímetros, siendo la lente ajustable de 120° la que ofrece mayor nitidez y cobertura, asegurando imágenes adecuadas para la identificación de dígitos.

Respecto al desempeño del modelo neuronal, se alcanzó una precisión del 96.18 % con imágenes externas al MNIST, evidenciando su capacidad de generalización en escenarios reales y un adecuado equilibrio entre simplicidad y precisión.

A pesar de estos avances, se reconocen limitaciones inherentes al alcance experimental. Las pruebas se realizaron bajo condiciones controladas y con un único modelo de medidor, lo cual restringe la extrapolación de los resultados a otros formatos y contextos de instalación. Por ello, se prevé realizar una fase de validación extendida en campo, incluyendo distintos tipos de medidores y variaciones de iluminación, con el propósito de evaluar la robustez del sistema frente a interferencias visuales y condiciones ambientales adversas. Asimismo, se recomienda ampliar la muestra de estudio, incorporando un mayor número de medidores y escenarios de prueba. Esto permitirá obtener resultados más representativos y mejorar la confiabilidad del sistema propuesto.

Asimismo, se proyecta solicitar financiamiento externo de instituciones académicas o entidades vinculadas al sector energético con el fin de completar la versión final del prototipo y desplegarlo en zonas rurales durante un periodo de un año, lo que permitirá verificar de forma práctica la autonomía energética del sistema y el comportamiento real del consumo de batería bajo condiciones prolongadas de operación.

Las mejoras futuras contemplan la implementación de procesamiento local mediante modelos cuantizados en el ESP32-CAM, reduciendo la dependencia externa y optimizando memoria y energía. Asimismo, se propone integrar mecanismos de autoenfoco, autocalibración y modos de suspensión adaptativos o paneles solares para mejorar la autonomía. Finalmente, se considera la incorporación de protocolos de seguridad en la transmisión de datos para garantizar la integridad y confidencialidad de la información.

## AGRADECIMIENTOS

Se expresa especial agradecimiento a la Universidad Tecnológica del Perú (UTP) por los ambientes de laboratorio prestados para las pruebas de esta investigación.

## REFERENCIAS

- [1] SCADA Automatización, “La industria de la automatización industrial en Perú: Un sector en crecimiento y transformación,” *SCADA Automatización*, [Online]. Available: <https://scadaautomatizacion.com/la-industria-de-la-automatizacion-industrial-en-peru-un-sector-en-crecimiento-y-transformacion/>. Accessed: Jun. 10, 2025.
- [2] J. A. Muro Rosado, “Smart grid en el Perú: Retos y factores críticos de éxito,” *Rev. Peruana de Energía*, 2014. [Online]. Available: <https://santivanez.com.pe/wp-content/uploads/2024/07/Revista-Peruana-de-Energia-4.pdf>. Accessed: Jun. 10, 2025.
- [3] Mesh-Net, “Wireless automatic meter reading,” *Mesh-Net*, [Online]. Available: <https://www.mesh-net.co.uk/wireless-automatic-meter-reading/>. Accessed: Jun. 6, 2025.
- [4] Perú Energía, “Proyectan instalación de medidores inteligentes en Lima para el uso responsable de la electricidad,” *Perú Energía*, [Online]. Available: <https://peruenergia.com.pe/proyectan-instalacion-de-medidores-inteligentes-en-lima-para-el-uso-responsable-de-la-electricidad/>. Accessed: Jun. 6, 2025.
- [5] J. A. Gutiérrez, “Análisis de la factibilidad de implementación de medidores inteligentes con la tecnología de radio cognitiva para mejorar el registro de consumo de energía eléctrica en las comunidades campesinas del distrito de San Jerónimo – 2017,” *B.S. thesis*, Univ. Nac. de San Agustín de Arequipa, Peru, 2021. [Online]. Available: <https://repositorio.unsaac.edu.pe/handle/20.500.12918/3962>. Accessed: Jun. 10, 2025.
- [6] UPME, “Unidad de Planeamiento Minero Energético,” 2017. [Online]. Available: <https://www1.upme.gov.co/Paginas/Smart-Grids-Colombia-Visi%C3%B3n-2030.aspx>. Accessed: Oct. 25, 2025.
- [7] MINEM, “Estadísticas eléctricas mensuales: Comportamiento mensual del subsector eléctrico,” *Ministerio de Energía y Minas del Perú*, Lima, Peru, 2021. [Online]. Available: <https://www.minem.gob.pe/>. Accessed: Oct. 25, 2025.
- [8] BID, “Pérdidas eléctricas en América Latina y el Caribe: un problema crónico para la sostenibilidad del sector,” *Blogs del Banco Interamericano de Desarrollo (BID)*, May 10, 2018. [Online]. Available: <https://blogs.iadb.org/energia/es/perdidas-electricas-en-america-latina-y-el-caribe-un-problema-cronico-para-la-sostenibilidad-del-sector/>. Accessed: Oct. 25, 2025.
- [9] A. F. Jiménez-Jiménez, “La eficiencia energética en los sistemas eléctricos de distribución: un reto para la sostenibilidad,” *Redalyc*, vol. 79, no. 323, pp. 31–40, May 2017. [Online]. Available: <https://www.redalyc.org/pdf/4977/497779323003.pdf>. Accessed: Oct. 25, 2025.
- [10] U.S. Energy Information Administration (EIA), “Customer-level advanced metering infrastructure,” [Online]. Available: <https://www.eia.gov/tools/faqs/faq.php?id=108&t=3>. Accessed: May 28, 2025. ELEKS, “The essential guide to smart grids: Benefits and challenges,” [Online]. Available: <https://eleks.com/blog/guide-smart-grids-benefits-challenges/>. Accessed: May 28, 2025.
- [11] D. Scaini, “AI-based water meter reading: Get your old meter into the IoT!,” *Elektor Mag.*, [Online]. Available: <https://www.elektormagazine.com/articles/ai-based-water-meter-reading-part1>. Accessed: May 28, 2025.
- [12] SunFounder, “ESP32 CAM documentation,” [Online]. Available: [https://docs.sunfounder.com/projects/galaxy-rvr/es/latest/hardware/cpn\\_esp\\_32\\_cam.html](https://docs.sunfounder.com/projects/galaxy-rvr/es/latest/hardware/cpn_esp_32_cam.html). Accessed: May 18, 2025.
- [13] J. A. Sánchez Abello and K. P. Parra Tarazona, “Diseño de un sistema digital para la lectura de contadores de agua por medio de cámaras e IA,” *B.S. thesis*, Univ. Industrial de Santander, Colombia, 2023. [Online]. Available: <https://noesis.uis.edu.co/items/884b67ed-a567-4202-afb5-46c5e4835a3a>. Accessed: Jun. 10, 2025.
- [14] U.S. Energy Information Administration (EIA), “How many smart meters are installed in the United States, and who has them?,” [Online]. Available: <https://www.eia.gov/tools/faqs/faq.php?id=108&t=3>. Accessed: May 28, 2025.
- [15] V. Nikić, D. Bortnik, M. Lukić, D. Vukobratović, and I. Mezei, “Lightweight digit recognition in smart metering system using narrowband Internet of Things and federated learning,” *Future Internet*, vol. 16, no. 11, Art. no. 402, Oct. 2024, doi: 10.3390/fi16110402.
- [16] J. Yin, P. Xia, J. He, L. Gu, and K. Yang, “An automatic electricity meter reading system based on neural vision location,” *J. Phys. Conf. Ser.*, vol. 1267, no. 1, Art. no. 012076, Jul. 2019, doi: 10.1088/1742-6596/1267/1/012076.
- [17] L. A. Elrefaei, A. Bajaber, S. Natheir, N. AbuSanab, and M. Bazi, “Automatic electricity meter reading based on image processing,” in *Proc. IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. (AEECT)*, Nov. 2015, pp. 1–5, doi: 10.1109/AEECT.2015.7360571.
- [18] M. Vanetti, I. Gallo, and A. Nodari, “GAS meter reading from real world images using a multi-net system,” *Pattern Recognit. Lett.*, vol. 34, no. 5, pp. 519–526, Apr. 2013, doi: 10.1016/j.patrec.2012.11.014.
- [19] S. Zhao, B. Li, J. Yuan, and G. Cui, “Research on remote meter automatic reading based on computer vision,” in *Proc. IEEE/PES Transm. Distrib. Conf. Expo.: Asia and Pacific*, 2005, pp. 1–4, doi: 10.1109/TDC.2005.1546972.
- [20] P. A. Inquiltupa Cortez, “Implementación de un sistema de reconocimiento facial basado en inteligencia artificial para la detección de intrusos con cámaras de seguridad convencionales en condominios,” *B.S. thesis*, Univ. Nac. de San Antonio Abad del Cusco, Peru, 2024. [Online]. Available: <https://repositorio.unsaac.edu.pe/handle/20.500.12918/9548>. Accessed: Jun. 10, 2025.
- [21] Arrow Electronics, “ESP32 power consumption can be reduced with sleep modes,” *Arrow.com*, Jun. 2022. [Online]. Available: <https://www.arrow.com/es/resources/articles/2022/06/esp32-power-consumption-can-be-reduced-with-sleep-modes.html>. Accessed: Apr. 17, 2026.
- [22] Arduino and OpenMV, “Arduino Nicla Vision,” *OpenMV Official Store*, 2025. [Online]. Available: <https://openmv.io/products/arduino-nicla-vision>. Accessed: Oct. 27, 2025.
- [23] [3] PKCELL, “Batería de litio 3.6 V 19000 mAh Li-SOCL<sub>2</sub> tamaño D R20 PKCELL ER34615,” *Bracsan Importaciones*, Lima, Perú, 2025. [Online]. Available: <https://bracsan.pe/producto/bateria-de-litio-3-6v-19000mah-li-socl2-tamano-d-r20-pkcell-er34615-2/>. Accessed: Oct. 27, 2025.
- [24] ENOSA, “Memoria anual ENOSA 2020,” *Mesa Directiva del Grupo Distrital*, Piura, Peru, 2020. [Online]. Available: <https://www.enosa.com.pe/>. Accessed: Oct. 25, 2025.
- [25] J. M. Jomjol, “AI-on-the-edge-device: Documentation and source code,” *GitHub Pages*, 2025. [Online]. Available: <https://jomjol.github.io/AI-on-the-edge-device-docs/>. Accessed: Oct. 27, 2025.
- [26] Noctuai, “Automated Reading of Analog Electricity Meters,” *Case Study*, 2025. [Online]. Available: <https://noctuai.com/case-study/automated-analog-electricity-meter-reading/>. Accessed: Oct. 27, 2025.
- [27] Roboflow, “Automating Water Meter Reading using Computer Vision,” *Roboflow Blog*, 2025. [Online]. Available: <https://blog.roboflow.com/water-meter-reading/>. Accessed: Oct. 27, 2025.
- [28] Sensorfact, “How to make your existing analog meters smart,” *Sensorfact Blog*, 2025. [Online]. Available: <https://sensorfact.eu/blog/how-to-make-your-existing-analog-meters-smart/>. Accessed: Oct. 27, 2025.