



**VALIDACIÓN AUTOMÁTICA DE ETIQUETAS DE RED  
MEDIANTE VISIÓN COMPUTACIONAL Y BOT EN CAMPO**

María Augusta Flores | Marcelo Andrade

[www.acclatam.lat/mmia](http://www.acclatam.lat/mmia)



**M & M**

INTELIGENCIA ARTIFICIAL  
INNOVACIONES



# Validación automática de etiquetas de red mediante visión computacional y Bot en campo

## Análisis Comparativo de Algoritmos.



Asignatura: Proyecto Integrador en Inteligencia Artificial –  
Maestría en Inteligencia Artificial

Docente: Ing. Gladys Villegas Rugel

Guayaquil – septiembre 2025

Quito, Ecuador. Telf. (593) 095 861 0558 / 098 270 4137  
E-mail: info@acclatam.lat  
www.acclatam.lat/mmia





# Tabla de Contenidos

<i>Validación automática de etiquetas de red mediante visión computacional y Bot en campo</i>	2
<i>Tabla de Contenidos</i>	1
<b>1. Random Forest</b>	3
1.1. Descripción Teórica	3
1.2. Ventajas y Desventajas	3
1.3. Complejidad Computacional	3
1.4. Casos de Uso Típicos	4
1.5. Aplicabilidad al Proyecto	4
<b>2. Support Vector Machines (SVM)</b>	4
2.1. Descripción Teórica	4
2.2. Ventajas y Desventajas	5
2.3. Complejidad Computacional	5
2.4. Casos de Uso Típicos	5
2.5. Aplicabilidad al Proyecto	5
<b>3. Redes Neuronales Artificiales (ANN)</b>	6
3.1. Descripción Teórica	6
3.2. Ventajas y Desventajas	6
3.3. Complejidad Computacional	6
3.4. Casos de Uso Típicos	7
3.5. Aplicabilidad al Proyecto	7
<b>4. Transformers</b>	7
4.1. Descripción Teórica	7
4.2. Ventajas y Desventajas	8
4.3. Complejidad Computacional	8





<b>4.4. Casos de Uso Típicos</b>	<b>8</b>
<b>4.5. Aplicabilidad al Proyecto</b>	<b>8</b>
<b>5. Convolutional Neural Networks (CNN)</b>	<b>9</b>
<b>5.1. Descripción Teórica</b>	<b>9</b>
<b>5.2. Ventajas y Desventajas</b>	<b>9</b>
<b>5.3. Complejidad Computacional</b>	<b>9</b>
<b>5.4. Casos de Uso Típicos</b>	<b>10</b>
<b>5.5. Aplicabilidad al Proyecto</b>	<b>10</b>
<b>6. Comparación Global de Algoritmos Evaluados</b>	<b>11</b>
<b>7. Justificación Final y Selección del Algoritmo</b>	<b>11</b>
<b>8. Referencias</b>	<b>12</b>





# 1. Random Forest

## 1.1. Descripción Teórica

Random Forest es un algoritmo de aprendizaje supervisado basado en ensamblado de árboles de decisión. Utiliza bagging (Bootstrap Aggregation), donde múltiples árboles de decisión se entran sobre subconjuntos aleatorios del conjunto de datos.

Cada árbol se construye seleccionando aleatoriamente muestras y características.

En clasificación, la predicción es por mayoría (voto); en regresión, se toma el promedio de las predicciones. Su base matemática se fundamenta en la reducción de varianza a través de la agregación de múltiples estimadores débiles.

**Fórmula de predicción en clasificación:**

$$\hat{y} = \text{mode}(T_1(x), T_2(x), \dots, T_n(x))$$

**Tipo de aprendizaje: Supervisado**

Parámetros principales: n\_estimators, max\_depth, max\_features, criterion

Referencias: Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

## 1.2. Ventajas y Desventajas

**Ventajas:**

- Robusto frente al overfitting.
- Funciona bien con datos de alta dimensionalidad.
- Puede manejar valores faltantes y datos no lineales.
- Estima importancia de variables.

**Desventajas:**

- Poco interpretable comparado con un árbol individual.
- Costoso en recursos de memoria y tiempo.
- Requiere ajuste de hiperparámetros para rendimiento óptimo.

## 1.3. Complejidad Computacional

- Entrenamiento:  $O(n_{\text{estimators}} \times m \log m)$ , donde  $m$  es el número de muestras.
- Predicción:  $O(n_{\text{estimators}})$ .
- Espacial: Alta, ya que almacena múltiples árboles.
- Escalabilidad: Buena en sistemas paralelos, limitada en hardware modesto.



## 1.4. Casos de Uso Típicos

- Clasificación médica (diagnóstico basado en síntomas).
- Detección de fraude en banca.
- Predicción de abandono de clientes.
- Segmentación de mercado.

Industria: Salud, finanzas, telecomunicaciones.

## 1.5. Aplicabilidad al Proyecto

Aunque Random Forest no opera directamente sobre imágenes, puede utilizarse tras un proceso de extracción de características (por ejemplo, utilizando embeddings generados por CNN).

Puede servir como clasificador de segunda etapa, comparando resultados de OCR vs etiqueta esperada. Es útil para escenarios con múltiples características simbólicas o numéricas.

## 2. Support Vector Machines (SVM)

### 2.1. Descripción Teórica

Support Vector Machines (SVM) es un algoritmo supervisado utilizado para clasificación y regresión.

Su objetivo es encontrar el hiperplano óptimo que separa las clases con el máximo margen. Utiliza puntos de datos denominados vectores de soporte para definir este hiperplano.

SVM puede extenderse a problemas no-lineales mediante el uso de funciones kernel, como RBF, polinomial o sigmoide.

Dado un conjunto de entrenamiento  $\{(x_i, y_i)\}$ , donde  $x_i \in \mathbb{R}^n$  y  $y_i \in \{-1, 1\}$ , el problema se formula como:

Minimizar:  $\frac{1}{2} ||w||^2$

Sujeto a:  $y_i(w \cdot x_i + b) \geq 1$

- **Tipo de aprendizaje: Supervisado**

- Fundamentos: Programación convexa, separación de clases

- Parámetros principales: kernel, C (penalización), gamma (en kernels RBF)





Referencias: Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273–297.

## 2.2. Ventajas y Desventajas

### Ventajas:

- Eficiente en espacios de alta dimensión.
- Robusto frente al overfitting si los parámetros están bien ajustados.
- Versátil mediante uso de kernels.
- Garantiza una solución óptima global (convexidad).

### Desventajas:

- Sensible a la elección de hiperparámetros.
- No escalable para datasets muy grandes.
- Difícil de interpretar.
- Entrenamiento lento con muchos datos o dimensiones.

## 2.3. Complejidad Computacional

- Entrenamiento:  $O(n^2)$  a  $O(n^3)$ , siendo  $n$  el número de muestras.
- Predicción:  $O(s)$  donde  $s$  es el número de vectores de soporte.
- Espacial: Alta si hay muchos vectores de soporte.
- Escalabilidad: Limitada en grandes volúmenes de datos.

## 2.4. Casos de Uso Típicos

- Clasificación de imágenes y rostros.
- Bioinformática (clasificación de genes, proteínas).
- Reconocimiento de escritura manuscrita.
- Filtrado de spam y minería de texto.

Industria: Salud, seguridad, legal, educación.

## 2.5. Aplicabilidad al Proyecto

SVM es útil cuando se trabaja con vectores extraídos de imágenes, como características obtenidas de OCR o descriptores locales.

Sin embargo, su escalabilidad es un problema si se quiere procesar en tiempo real muchas imágenes. Podría utilizarse como comparador en una arquitectura híbrida, pero no como algoritmo principal del pipeline de visión.





## 3. Redes Neuronales Artificiales (ANN)

### 3.1. Descripción Teórica

Las redes neuronales artificiales (ANN) están inspiradas en el funcionamiento del cerebro humano y consisten en nodos (neuronas) organizados en capas (entrada, ocultas, salida).

Cada conexión entre neuronas tiene un peso que se ajusta durante el entrenamiento mediante el algoritmo de retropropagación (backpropagation) y descenso del gradiente.

Dado un vector de entrada  $x$ , una red neuronal calcula la salida como:

$$\hat{y} = f(Wx + b),$$

donde  $W$  son los pesos,  $b$  el sesgo, y  $f$  la función de activación (ReLU, Sigmoid, etc.).

- **Tipo de aprendizaje: Supervisado**
- Fundamentos: Álgebra lineal, derivadas parciales, optimización
- Parámetros: número de capas, número de neuronas, tasa de aprendizaje, función de activación

Referencia: Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

### 3.2. Ventajas y Desventajas

#### Ventajas:

- Capacidad de aproximar funciones complejas.
- Adaptables a múltiples dominios (imágenes, texto, audio).
- Buen desempeño con grandes volúmenes de datos.
- Soportan entrenamiento en GPU.

#### Desventajas:

- Necesitan gran cantidad de datos para evitar overfitting.
- Costoso en cómputo.
- Difícil de interpretar ('caja negra').
- Sensibles a hiperparámetros y normalización de datos.

### 3.3. Complejidad Computacional

- Entrenamiento:  $O(n \times m \times d)$ , donde  $n$  = muestras,  $m$  = neuronas,  $d$  = dimensiones
- Predicción:  $O(m \times d)$



- Espacial: Alta, según número de parámetros
- Escalabilidad: Muy buena si se usa GPU/TPU

### 3.4. Casos de Uso Típicos

- Reconocimiento de patrones en imágenes
- Predicción de demanda
- Clasificación de texto y sentimiento
- Modelado de series de tiempo

Industrias: comercio, salud, finanzas, transporte

### 3.5. Aplicabilidad al Proyecto

Las ANN pueden utilizarse para tareas simples de clasificación tras un OCR, pero su efectividad se incrementa al usar variantes profundas como CNN para imágenes. Su uso directo es limitado si se requiere localizar texto dentro de una imagen. Sin embargo, en arquitectura combinada (OCR + ANN) pueden ofrecer buenos resultados.

## 4. Transformers

### 4.1. Descripción Teórica

Los Transformers son una arquitectura de deep learning introducida por Vaswani et al. (2017), diseñada originalmente para tareas de procesamiento de lenguaje natural (NLP), pero que ha sido adaptada con gran éxito a visión computacional (Vision Transformers - ViT).

Su componente clave es el mecanismo de atención, que permite al modelo ponderar la importancia relativa de cada parte de la entrada.

Fórmulas base del mecanismo de atención:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) \cdot V$$

donde Q (query), K (key) y V (value) son vectores aprendidos, y  $d_k$  es la dimensión del key.

- Tipo de aprendizaje: Supervisado
- Fundamentos: Atención escalada, normalización, codificación posicional
- Parámetros: número de capas, cabezas de atención, dimensión oculta, dropout

Referencia: Vaswani, A. et al. (2017). Attention Is All You Need. NeurIPS.

Quito, Ecuador. Telf. (593) 095 861 0558 / 098 270 4137

E-mail: [info@acclatam.lat](mailto:info@acclatam.lat)  
[www.acclatam.lat/mmia](http://www.acclatam.lat/mmia)





## 4.2. Ventajas y Desventajas

### Ventajas:

- Capturan dependencias globales mejor que CNN y RNN.
- Paralelizables y eficientes en entrenamiento.
- Flexibles para texto, imagen, audio y más.
- Soportan modelos preentrenados como BERT, ViT, etc.

### Desventajas:

- Requieren grandes volúmenes de datos.
- Costosos en entrenamiento (muchos parámetros).
- Poca interpretabilidad.
- Necesitan hardware especializado (GPU/TPU).

## 4.3. Complejidad Computacional

- Entrenamiento:  $O(n^2d)$ , donde  $n$  = número de tokens/patches,  $d$  = dimensión de embeddings
- Predicción: similar a entrenamiento
- Espacial: Muy alta
- Escalabilidad: Alta en entornos distribuidos con GPU

## 4.4. Casos de Uso Tipicos

- Traducción automática (NLP)
- Clasificación y segmentación de imágenes (ViT, DETR)
- Generación de texto e imágenes (GPT, DALL-E)
- Análisis multicanal (texto + imagen + audio)

Industrias: tecnología, educación, salud, medios

## 4.5. Aplicabilidad al Proyecto

Transformers para visión (ViT) pueden ser muy útiles para detectar y clasificar etiquetas en entornos ruidosos, especialmente si se requiere contextualizar elementos (por ejemplo, texto + posición).

Sin embargo, su uso requiere gran poder de cómputo y un dataset extenso. Serían recomendables si se cuenta con etiquetas complejas y se desea precisión de nivel industrial.





## 5. Convolutional Neural Networks (CNN)

### 5.1. Descripción Teórica

Las CNN son una clase especializada de redes neuronales para procesamiento de imágenes.

Utilizan capas convolucionales que aplican filtros (kernels) para extraer características espaciales locales, seguidas de capas de pooling que reducen dimensionalidad y ayudan a generalizar.

Se usan funciones de activación como ReLU y capas fully connected al final para clasificar.

Operación convolucional básica:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n K(m,n) \cdot I(i+m, j+n)$$

donde  $K$  es el kernel y  $I$  la imagen.

- Tipo de aprendizaje: Supervisado
- Fundamentos: Álgebra de matrices, convolución, redes profundas
- Parámetros: número y tamaño de filtros, stride, padding, número de capas

Referencia: LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE.

### 5.2. Ventajas y Desventajas

#### Ventajas:

- Excelentes para reconocimiento de patrones espaciales.
- Requieren menos preprocesamiento que otros algoritmos.
- Soportan arquitecturas profundas (ResNet, EfficientNet).
- Generalizan bien si se usan con regularización.

#### Desventajas:

- Dependientes de gran volumen de datos etiquetados.
- A veces pierden contexto global (limitación de campo de visión).
- Costosas en tiempo de entrenamiento sin GPU.
- Poca interpretabilidad en capas intermedias.

### 5.3. Complejidad Computacional

- Entrenamiento:  $O(n \times k^2 \times m^2 \times f)$ , donde  $k$  = tamaño de kernel,  $m$  = tamaño imagen,  $f$  = filtros



- Predicción: O similar
- Espacial: Alta, especialmente en modelos con muchas capas
- Escalabilidad: Alta con GPU o TPU

## 5.4. Casos de Uso Típicos

- Clasificación de imágenes
- Detección de objetos (YOLO, SSD)
- OCR y lectura automática de texto (CRNN)
- Análisis médico por imágenes (tumores, rayos X)

Industrias: salud, transporte, manufactura, retail

## 5.5. Aplicabilidad al Proyecto

Las CNN son el enfoque más directo y efectivo para este proyecto, pues permiten detectar y validar automáticamente etiquetas en fotos capturadas en campo.

Son compatibles con técnicas OCR modernas y pueden integrarse con bots que usen la salida de la CNN para validar en tiempo real. Su uso es altamente recomendable como núcleo del sistema.





## 6. Comparación Global de Algoritmos Evaluados

Algoritmo	Precisión Esperada	Velocidad de Entrenamiento	Interpretabilidad	Escalabilidad	Adecuación al Proyecto
Random Forest	Alta	Media	Baja	Alta	Moderada
SVM	Alta	Baja	Baja	Limitada	Baja
ANN	Media-Alta	Media	Muy baja	Alta	Moderada
Transformers	Muy Alta	Baja	Muy baja	Alta	Alta si hay recursos
CNN	Muy Alta	Media	Baja	Alta	Excelente

## 7. Justificación Final y Selección del Algoritmo

En base a un análisis exhaustivo, y considerando los criterios establecidos en la rúbrica de evaluación —incluyendo profundidad técnica, precisión en la complejidad computacional, relevancia de casos de uso y aplicabilidad al proyecto— se concluye que el algoritmo más adecuado para nuestro caso es **Convolutional Neural Networks (CNN)**.

CNN es una técnica probada, robusta y optimizada para tareas de visión por computadora, lo cual se alinea directamente con la necesidad de validar etiquetas de red a través de imágenes capturadas por técnicos en campo.

Las CNN pueden extraer características espaciales críticas incluso en condiciones complejas (iluminación, rotación, ruido), y permiten su integración natural con sistemas OCR modernos. Esta capacidad de combinación es esencial para el flujo de trabajo de nuestro sistema, que incluye detección visual + reconocimiento textual + validación automatizada.

Adicionalmente, CNN ofrece alta escalabilidad con soporte GPU, precisión comprobada en benchmarks industriales (YOLO, EfficientNet, etc.) y una comunidad técnica amplia para soporte y mejora continua.

Por tanto, con base en la justificación técnica, la alineación al problema real y la factibilidad de implementación con recursos disponibles, se selecciona CNN como núcleo del modelo de IA propuesto para este proyecto.





## 8. Referencias

- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273–297.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

