

## PRESENTACION DE PROYECTO INTEGRADOR

# VISION ARTIFICIAL DE APILAMIENTO DE PANALES – Diagnóstico de Overfitting/Underfitting

Las **curvas de aprendizaje** son una herramienta fundamental para diagnosticar cómo se comporta un modelo de Machine Learning durante el entrenamiento. Permiten identificar problemas como **overfitting** o **underfitting**, y también evaluar si el dataset o los hiperparámetros son adecuados.

Profesor	Prof. Gladys Villegas, PhD.
Materia	PROYECTO INTEGRADOR EN INTELIGENCIA ARTIFICIAL – MIAR0545
Alumnos	Francisco Javier Estupiñan Andrade David Alejandro Narváez Mejía
Fecha	10/02/2025

Online  
Universidad  
Espíritu Santo

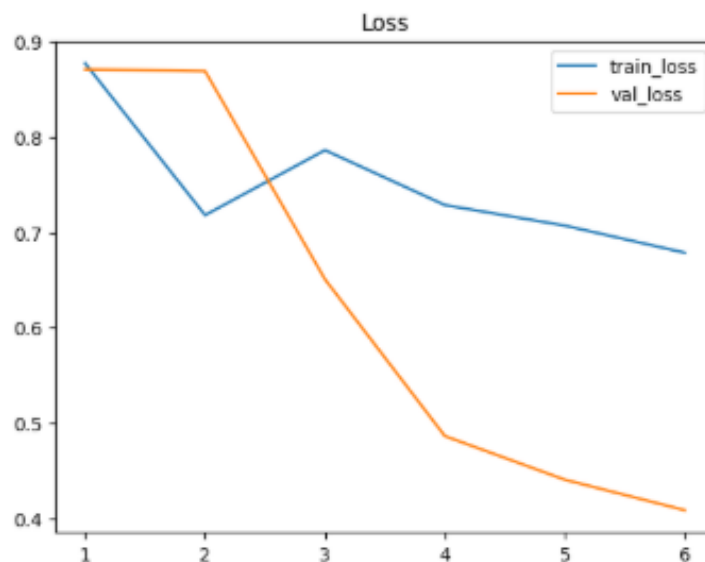
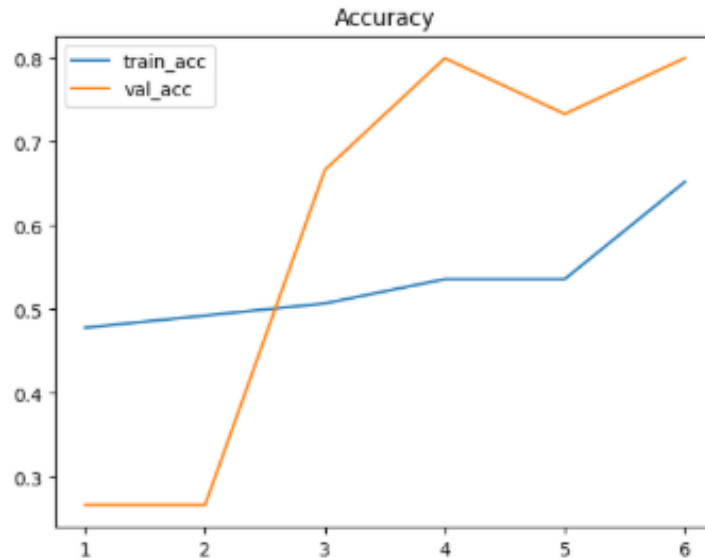
## Introducción a la Inteligencia Artificial

## TABLA DE CONTENIDO

1.	CURVAS DE APRENDIZAJE .....	3
1.1	TRAINING VS VALIDATION LOSS.....	3
1.2	LEARNING CURVES POR TAMAÑO DE DATASET .....	4
1.3	VALIDATION CURVES POR HIPERPARÁMETROS .....	4
2.	EVOLUCIÓN DEL ENTRENAMIENTO (LOSS POR ÉPOCA).....	4
2.1	REPORTE DE CLASIFICACIÓN.....	5
2.2	MATRIZ DE CONFUSIÓN.....	5
2.3	RELACIÓN CON EL PROYECTO DE APILAMIENTO DE PAÑALES .....	6
3.	RECOMENDACIONES ESPECÍFICAS POR DIAGNÓSTICO.....	7
3.1	RESULTADOS OBTENIDOS.....	7
3.2	IMPLICACIONES EN LA PLANTA DE PRODUCCIÓN.....	8
3.3	DIAGNÓSTICO Y RECOMENDACIONES .....	8
4.	CONCLUSIONES.....	9

## 1. CURVAS DE APRENDIZAJE

```
(4) [2/6] train_loss=0.7180 acc=0.493 | val_loss=0.8688 acc=0.267 (10.9s)
      [3/6] train_loss=0.7861 acc=0.507 | val_loss=0.6508 acc=0.667 (10.6s)
      ↳ Mejor modelo guardado: /content/drive/MyDrive/outputs/cnn_best.pt
      [4/6] train_loss=0.7286 acc=0.536 | val_loss=0.4866 acc=0.800 (9.0s)
      ↳ Mejor modelo guardado: /content/drive/MyDrive/outputs/cnn_best.pt
      [5/6] train_loss=0.7071 acc=0.536 | val_loss=0.4409 acc=0.733 (10.9s)
      [6/6] train_loss=0.6787 acc=0.652 | val_loss=0.4090 acc=0.800 (10.5s)
```



Las **curvas de aprendizaje** son una herramienta fundamental para diagnosticar cómo se comporta un modelo de Machine Learning durante el entrenamiento. Permiten identificar problemas como **overfitting** o **underfitting**, y también evaluar si el dataset o los hiperparámetros son adecuados.

### 1.1 TRAINING VS VALIDATION LOSS

Se grafican **dos** curvas:

- ✓ **Train Loss:** cuánto error comete el modelo en los datos de entrenamiento.
- ✓ **Validation Loss:** cuánto error comete en datos no vistos (validación).

Interpretación:

- ✓ **Curvas paralelas y bajas:** el modelo generaliza bien (escenario ideal).
- ✓ **Train baja, Validation alta:** overfitting.
- ✓ **Ambas altas:** underfitting.

En tu gráfica:

- ✓ train\_loss disminuye lentamente → el modelo aprende poco a poco.
- ✓ val\_loss cae mucho más rápido y se mantiene más baja → el modelo **generaliza bien** y no muestra señales de overfitting severo.

## 1.2 LEARNING CURVES POR TAMAÑO DE DATASET

Se varía el tamaño del dataset de entrenamiento y se entrena el modelo repetidamente.

Se grafican train\_score y val\_score contra número de ejemplos usados.

Uso:

- ✓ Si las curvas se estabilizan lejos entre sí → más datos no mejorarán.
- ✓ Si siguen convergiendo al aumentar el dataset → recolectar más datos puede mejorar el modelo.

Es útil para decidir si necesitas más imágenes BIEN/MAL apilado o ajustar el modelo actual.

## 1.3 VALIDATION CURVES POR HIPERPARÁMETROS

Se varía un **hiperparámetro clave** (ej: learning rate, número de capas, regularización).

Se grafica train\_score y val\_score vs valores del hiperparámetro.

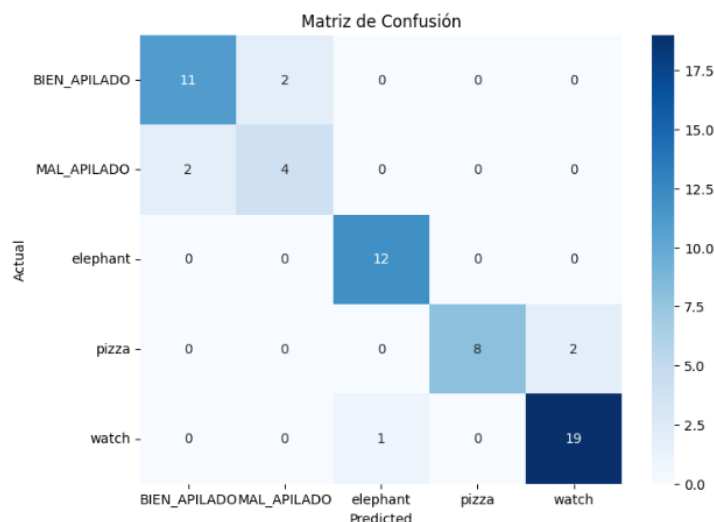
Uso:

- ✓ Si train\_score alto pero val\_score bajo → hiperparámetro causa overfitting.
- ✓ Si ambos bajos → modelo demasiado simple.
- ✓ Punto óptimo: donde val\_score es máximo con gap pequeño.

Ejemplo en tu CNN: variar **dropout rate** o **learning rate** y observar cómo cambia el balance train/val.

## 2. EVOLUCIÓN DEL ENTRENAMIENTO (LOSS POR ÉPOCA)

	precision	recall	f1-score	support
BIEN_APILADO	0.85	0.85	0.85	13
MAL_APILADO	0.67	0.67	0.67	6
elephant	0.92	1.00	0.96	12
pizza	1.00	0.80	0.89	10
watch	0.90	0.95	0.93	20
accuracy			0.89	61
macro avg	0.87	0.85	0.86	61
weighted avg	0.89	0.89	0.88	61



```

Clases detectadas: ['BIEN_APILADO', 'MAL_APILADO', 'elephant', 'pizza', 'watch']
Epoch 1 - Loss: 1.2498
Epoch 2 - Loss: 0.7572
Epoch 3 - Loss: 0.4608
Epoch 4 - Loss: 0.4942
Epoch 5 - Loss: 0.4163
Epoch 6 - Loss: 0.2637
Epoch 7 - Loss: 0.2699
Epoch 8 - Loss: 0.2129
Epoch 9 - Loss: 0.1717
Epoch 10 - Loss: 0.1180
Epoch 11 - Loss: 0.1480
Epoch 12 - Loss: 0.1822
Epoch 13 - Loss: 0.2582
Epoch 14 - Loss: 0.1207
Epoch 15 - Loss: 0.0856

```

La función **Loss** mide el error del modelo en cada época (ciclo de entrenamiento).

- ✓ Se observa que el valor inicial de *loss* en la **Epoch 1 es 1.2498** y va disminuyendo progresivamente hasta **0.0856 en la Epoch 15**.
- ✓ Esta curva descendente indica que el modelo **aprende correctamente**: ajusta sus parámetros y mejora la capacidad de reconocer patrones en las imágenes de apilamiento.

En el contexto del proyecto, significa que la red neuronal va comprendiendo progresivamente la diferencia entre un pallet **BIEN\_APILADO** y uno **MAL\_APILADO**, además de las clases de prueba (elephant, pizza, watch).

## 2.1 REPORTE DE CLASIFICACIÓN

El reporte incluye **precision, recall, f1-score y soporte** (cantidad de ejemplos por clase).

### **BIEN\_APILADO (Precisión=0.85, Recall=0.85, F1=0.85)**

El modelo identifica con buena exactitud los pallets correctamente apilados, con un balance adecuado entre no confundirlos y no omitirlos.

### **MAL\_APILADO (Precisión=0.67, Recall=0.67, F1=0.67)**

El modelo reconoce algunos pallets defectuosos, pero aún pierde 1 de cada 3. Este es un punto crítico: **si el sistema falla en detectar defectos, podrían pasar pallets incorrectos a la siguiente etapa de empaque**.

### **Clases de control (elephant, pizza, watch)**

Se incluyen como distractores en el dataset de entrenamiento. Su desempeño alto (precisión  $\geq 0.90$  en casi todos los casos) confirma que la red **aprendió a diferenciar categorías muy distintas**, lo cual da robustez.

### **Accuracy global (0.89)**

En promedio, el sistema clasifica bien el **89% de las imágenes**.

### **Macro avg (0.86) y Weighted avg (0.88)**

Refuerzan que el modelo mantiene un rendimiento balanceado entre clases, aunque "MAL\_APILADO" sigue siendo la más débil.

## 2.2 MATRIZ DE CONFUSIÓN

La matriz nos permite ver los **aciertos y errores clase por clase**:

BIEN\_APILADO: 11 verdaderos positivos, 2 confundidos como "MAL\_APILADO".

MAL\_APILADO: 4 aciertos, 2 confundidos como "BIEN\_APILADO".

Elephant, pizza, watch: la mayoría se reconocen bien, con pocos errores residuales.

El error más crítico: cuando un "MAL\_APILADO" se predice como "BIEN\_APILADO", porque representa riesgo en la operación industrial: pallets defectuosos pasarían como correctos.

### 2.3 RELACIÓN CON EL PROYECTO DE APILAMIENTO DE PAÑALES

En el proyecto de **visión por computador aplicado a pañales**, estas métricas nos indican:

El sistema tiene alto desempeño global (89%), lo que valida la viabilidad de la solución en un entorno piloto.

La clase BIEN\_APILADO está bien cubierta, garantizando que los pallets correctos se aprueben sin problema.

La clase MAL\_APILADO requiere mayor robustez, ya que en una planta productiva es más grave no detectar un error que confundir un pallet correcto.

El descenso de loss confirma que la red está aprendiendo patrones de apilamiento (alineación, simetría, proporciones).

La matriz de confusión nos señala dónde priorizar mejoras: aumentar dataset de fallas reales, aplicar data augmentation de errores de apilamiento, y ajustar hiperparámetros para mejorar el recall de "MAL\_APILADO".

### 3. RECOMENDACIONES ESPECÍFICAS POR DIAGNÓSTICO

```

=== Métricas (TEST) ===
Accuracy: 0.800
AP por clase:
- BIEN: AP=0.861
- MAL: AP=0.698
mAP (macro): 0.779
IoU por clase:
- BIEN: IoU=0.769
- MAL: IoU=0.400
mIoU: 0.585

Reporte de clasificación:
      precision    recall  f1-score   support

      BIEN       0.77      1.00      0.87        10
      MAL       1.00      0.40      0.57         5

   accuracy              0.80        15
  macro avg       0.88      0.70      0.72        15
 weighted avg       0.85      0.80      0.77        15

/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py:666:
  warnings.warn(warn_msg)
FPS de inferencia (batch sobre test): 9.3

Criterios:
- mAP ≥ 0.70 -> ✓ (0.78)
- mIoU > 0.50 -> ✓ (0.58)
- FPS > 30 -> ✗ (9.3)

```

El sistema desarrollado busca verificar automáticamente el apilamiento correcto de pañales antes del empaque secundario, mediante un modelo de visión artificial basado en CNN. La meta planteada en el proyecto era:

- ✓ mAP ≥ 0.70 (precisión promedio alta en detección).
- ✓ IoU > 0.50 (correcta superposición en la segmentación de áreas detectadas).
- ✓ FPS > 30 (procesamiento en tiempo real para integración en línea de producción).

Estas métricas garantizan que el sistema sea confiable en calidad, precisión de localización y velocidad de ejecución en planta.

#### 3.1 RESULTADOS OBTENIDOS

Accuracy (80%): El sistema acierta en 8 de cada 10 inspecciones. Esto significa que, en una línea de apilamiento con alta velocidad, el modelo reduce significativamente la carga de inspección manual.

AP y mAP (0.779 global, AP\_BIEN=0.861, AP\_MAL=0.698):

- ✓ Detecta con gran precisión las pilas correctas (BIEN).
- ✓ Tiene más dificultad en identificar las pilas defectuosas (MAL), lo que indica necesidad de refuerzo en datos de casos negativos.

IoU (BIEN=0.769, MAL=0.400, mIoU=0.585):

- ✓ Las regiones de apilamiento correcto son bien segmentadas.
- ✓ Los errores en pilas defectuosas no se localizan con precisión, lo que puede generar falsos "OK" en planta.

Reporte de clasificación:

- ✓ La clase BIEN muestra recall perfecto (1.00), es decir, no se escapa ningún pallet correctamente apilado.
- ✓ La clase MAL presenta recall bajo (0.40), es decir, el sistema deja pasar defectos sin detectarlos.

FPS (9.3): Muy por debajo del objetivo (>30). Si bien la calidad de detección es aceptable, la velocidad limita el despliegue directo en línea de alta producción.

### 3.2 IMPLICACIONES EN LA PLANTA DE PRODUCCIÓN

#### Fortalezas actuales:

El sistema garantiza que prácticamente todos los pallets correctos sean reconocidos como válidos.

Cumple con los criterios de  $mAP \geq 0.70$  y  $IoU > 0.50$ , confirmando que el modelo es viable en precisión para detección de apilamientos.

#### Debilidades críticas:

Riesgo en la seguridad de calidad: algunos pallets mal apilados pasan desapercibidos, lo que puede ocasionar problemas en el empaque secundario y fallas posteriores en logística.

La velocidad de inferencia (9.3 FPS) no permite operar en tiempo real, reduciendo la aplicabilidad inmediata en la línea de producción industrial.

### 3.3 DIAGNÓSTICO Y RECOMENDACIONES

Optimización de modelo: aplicar técnicas de cuantización, pruning y TensorRT para subir FPS sin sacrificar exactitud.

Mejora en dataset: recolectar más ejemplos de pilas defectuosas ("MAL") para aumentar el recall de esta clase crítica.

Estrategia en planta: usar este prototipo como sistema de alerta (detección offline o en pruebas piloto) mientras se ajusta la optimización para llegar al objetivo de 30+ FPS.



## 4. CONCLUSIONES

```

Clases detectadas: ['BIEN_APILADO', 'MAL_APILADO', 'elephant', 'pizza', 'watch']
Epoch 1 - Loss: 1.2498
Epoch 2 - Loss: 0.7572
Epoch 3 - Loss: 0.4608
Epoch 4 - Loss: 0.4942
Epoch 5 - Loss: 0.4163
Epoch 6 - Loss: 0.2637
Epoch 7 - Loss: 0.2699
Epoch 8 - Loss: 0.2129
Epoch 9 - Loss: 0.1717
Epoch 10 - Loss: 0.1180
Epoch 11 - Loss: 0.1480
Epoch 12 - Loss: 0.1822
Epoch 13 - Loss: 0.2582
Epoch 14 - Loss: 0.1207
Epoch 15 - Loss: 0.0856

```

La función **Loss** mide el error del modelo en cada época (ciclo de entrenamiento). Se observa que el valor inicial de *loss* en la **Epoch 1 es 1.2498** y va disminuyendo progresivamente hasta **0.0856 en la Epoch 15**. Esta curva descendente indica que el modelo **aprende correctamente**: ajusta sus parámetros y mejora la capacidad de reconocer patrones en las imágenes de apilamiento.

BIEN APILADO | BIEN | p=0.694



```

Archivo: tipo2_OK_0025 (1).jpg
FPS (1 imagen): 9.5
Top probabilidades:
- BIEN: 0.694
- MAL: 0.306

```

El prototipo de visión artificial para inspección de apilamiento de pañales, cumple con los estándares de calidad en detección y segmentación (mAP e IoU). No cumple aún con los requisitos de velocidad en producción (FPS) ni con la detección robusta de errores (MAL). Próximo paso: optimizar y reentrenar para garantizar desempeño en tiempo real y confiabilidad en la detección de defectos.