IA - Clase 2B

Aprendizaje de Máquina

(ML – Machine Learning)

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Imagen digital = matriz de valores numéricos que representan la intensidad de luz y color.
- ML tratamos a la imagen como un tensor, para poder aplicar técnicas matemáticas y de optimización.
- Una imagen en blanco y negro puede representarse como:

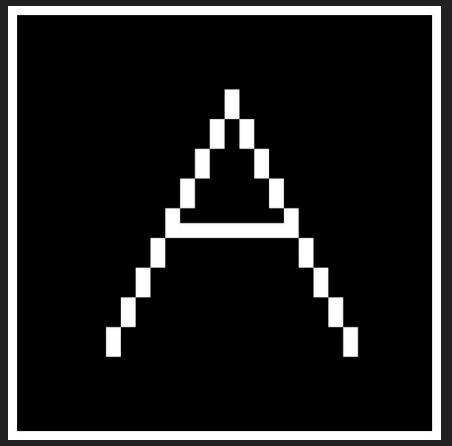
$$\mathbf{X} \in \mathbb{R}^{H \times W}$$

  - H: altura en pixeles
  - W: ancho en pixeles
  - Cada elemento X(i,j) ∈ [0,255]
  - [0,255] indica la intensidad (0 = negro, 255 = blanco)
    - Ejemplo: una imagen de 28×28 (MNIST) se representa como una matriz 28×28

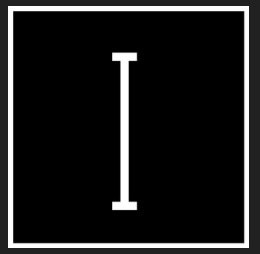
Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Letra A representada en una matriz de 28×28 píxeles
- Representación simbólica (matriz)

$$A = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,28} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,28} \\ \vdots & \vdots & \ddots & \vdots \\ x_{28,1} & x_{28,2} & \cdots & x_{28,28} \end{bmatrix}$$



- Con $x_{i,j} \in [0,255]$:
  - 255 = píxel blanco (trazo de la letra)
  - 0 = píxel negro (fondo)
  - Submatriz de ejemplo (sección de la barra horizontal central)



$$\begin{bmatrix} 0 & 255 & 0 & \cdots & 255 & 0 \\ 0 & 255 & 255 & \cdots & 255 & 0 \\ 255 & 0 & 0 & \cdots & 0 & 255 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 255 & 0 & 0 \\ 0 & 0 & 255 & 0 & 0 \\ 0 & 0 & 255 & 0 & 0 \\ 0 & 0 & 255 & 0 & 0 \\ 0 & 0 & 255 & 0 & 0 \end{bmatrix}$$

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Una imagen en color contiene 3 canales y cada pixel es un vector de 3 dimensiones:
  - Rojo (R), Verde (G) y Azul (B). $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ $p_{ij} = (R_{ij}, G_{ij}, B_{ij})$
  - Imagen de 224×224 usada en ImageNet → tensor 224×224×3
- ¿Qué significa 224×224×3?
  - 224 alto (filas),
  - 224 ancho (columnas),
  - 3 canales (R, G, B).
- Ejemplo:
  - Cargo en raw memory (memoria cruda) una imagen en color RGB desde disco (ej. gato.jpg)
  - Para eso uso librerías (Pillow o OpenCV)
  - Obtengo matriz tridimensional (un tensor) con dimensiones H×W×C
    - H = Height (alto, cantidad de filas, o número de píxeles verticales)
    - W = Width (ancho, cantidad de columnas, o número de píxeles horizontales)
    - C = Channels (canales de color). Para RGB, C=3.

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- En memoria cruda tenemos 224 filas × 224 columnas × 3 canales.
  - 224×224×3=150528 valores
  - Cada valor en la matriz es un entero sin signo de 8 bits (uint8).
  - Rango: [0,255]
  - Con 8 bits se pueden representar 2^8=256 posibles valores → 0, 1, 2, ..., 255.
  - Cada canal (R, G, B) de cada píxel se guarda como un número entero en ese rango.
    - Supongamos que el píxel en la fila 0, columna 0 tiene estos valores:
    - R=123,G=104,B=84
    - En memoria ese píxel se guarda como el vector: [123, 104, 84]
      - 123 → intensidad de rojo
      - 104 → intensidad de verde
      - 84 → intensidad de azul

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Imagen 224×224×3 en uint8:
  - Son 150528 valores.
  - Cada valor ocupa 1 byte
  - Total ≈ 150 KB
  - En float32 (4 bytes por valor) → ≈ 602 KB.

- Imaginar una caja tridimensional de números enteros, donde cada número indica cuánta intensidad de rojo, verde o azul tiene ese píxel, y cada valor va de 0 (nada de color) a 255 (máxima intensidad).

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Ejemplo: Gato con pelaje atigrado.
- Porción del tensor de 4×4 de la esquina superior izquierda de una imagen RGB, representado como un tensor tridimensional H×W×C.
- Canales Rojo, Verde, Azul

$$T \in \mathbb{R}^{4\times4\times3}$$

$$T = \begin{bmatrix} \begin{bmatrix}123,104,84\end{bmatrix} & \begin{bmatrix}126,108,88\end{bmatrix} & \begin{bmatrix}130,112,93\end{bmatrix} & \begin{bmatrix}118,100,80\end{bmatrix} \\ \begin{bmatrix}120,101,82\end{bmatrix} & \begin{bmatrix}124,105,86\end{bmatrix} & \begin{bmatrix}129,110,91\end{bmatrix} & \begin{bmatrix}115,98,78\end{bmatrix} \\ \begin{bmatrix}110,96,79\end{bmatrix} & \begin{bmatrix}118,102,83\end{bmatrix} & \begin{bmatrix}125,108,88\end{bmatrix} & \begin{bmatrix}112,97,77\end{bmatrix} \\ \begin{bmatrix}105,92,76\end{bmatrix} & \begin{bmatrix}112,98,80\end{bmatrix} & \begin{bmatrix}119,104,85\end{bmatrix} & \begin{bmatrix}110,95,74\end{bmatrix} \end{bmatrix}$$

$$T[:,:,0] = \begin{bmatrix}123 & 126 & 130 & 118 \\ 120 & 124 & 129 & 115 \\ 110 & 118 & 125 & 112 \\ 105 & 112 & 119 & 110\end{bmatrix}$$

$$T[:,:,1] = \begin{bmatrix}104 & 108 & 112 & 100 \\ 101 & 105 & 110 & 98 \\ 96 & 102 & 108 & 97 \\ 92 & 98 & 104 & 95\end{bmatrix}$$

$$T[:,:,2] = \begin{bmatrix}84 & 88 & 93 & 80 \\ 82 & 86 & 91 & 78 \\ 79 & 83 & 88 & 77 \\ 76 & 80 & 85 & 74\end{bmatrix}$$

- Cada entrada es un vector [R,G,B]
- T[0,0,:]=[123,104,84] → píxel fila 0, colun
- T[2,1,:]=[118,102,83] → píxel fila 2, colun

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Ejemplo: Gato con pelaje atigrado.
- Porción del tensor de 4×4 de la esquina superior izquierda de una imagen RGB, representado como un tensor tridimensional H×W×C.
- Canales Rojo, Verde, Azul

$$T \in \mathbb{R}^{4\times4\times3}$$

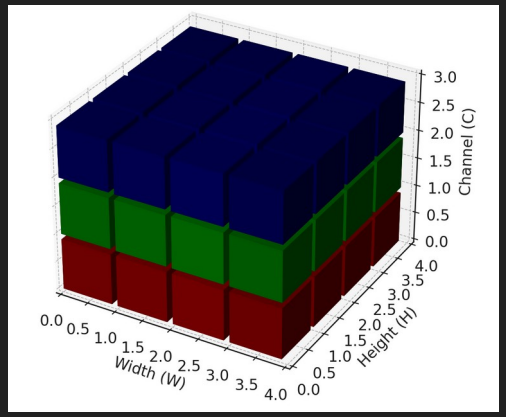
$$T = \begin{bmatrix} \begin{bmatrix}123,104,84\end{bmatrix} & \begin{bmatrix}126,108,88\end{bmatrix} & \begin{bmatrix}130,112,93\end{bmatrix} & \begin{bmatrix}118,100,80\end{bmatrix} \\ \begin{bmatrix}120,101,82\end{bmatrix} & \begin{bmatrix}124,105,86\end{bmatrix} & \begin{bmatrix}129,110,91\end{bmatrix} & \begin{bmatrix}115,98,78\end{bmatrix} \\ \begin{bmatrix}110,96,79\end{bmatrix} & \begin{bmatrix}118,102,83\end{bmatrix} & \begin{bmatrix}125,108,88\end{bmatrix} & \begin{bmatrix}112,97,77\end{bmatrix} \\ \begin{bmatrix}105,92,76\end{bmatrix} & \begin{bmatrix}112,98,80\end{bmatrix} & \begin{bmatrix}119,104,85\end{bmatrix} & \begin{bmatrix}110,95,74\end{bmatrix} \end{bmatrix}$$

$$T[:,:,0] = \begin{bmatrix}123 & 126 & 130 & 118 \\ 120 & 124 & 129 & 115 \\ 110 & 118 & 125 & 112 \\ 105 & 112 & 119 & 110\end{bmatrix}$$

$$T[:,:,1] = \begin{bmatrix}104 & 108 & 112 & 100 \\ 101 & 105 & 110 & 98 \\ 96 & 102 & 108 & 97 \\ 92 & 98 & 104 & 95\end{bmatrix}$$

$$T[:,:,2] = \begin{bmatrix}84 & 88 & 93 & 80 \\ 82 & 86 & 91 & 78 \\ 79 & 83 & 88 & 77 \\ 76 & 80 & 85 & 74\end{bmatrix}$$

- Cada entrada es un vector [R,G,B]
- T[0,0,:]=[123,104,84] → píxel fila 0, columna 0.
- T[2,1,:]=[118,102,83] → píxel fila 2, columna 1.

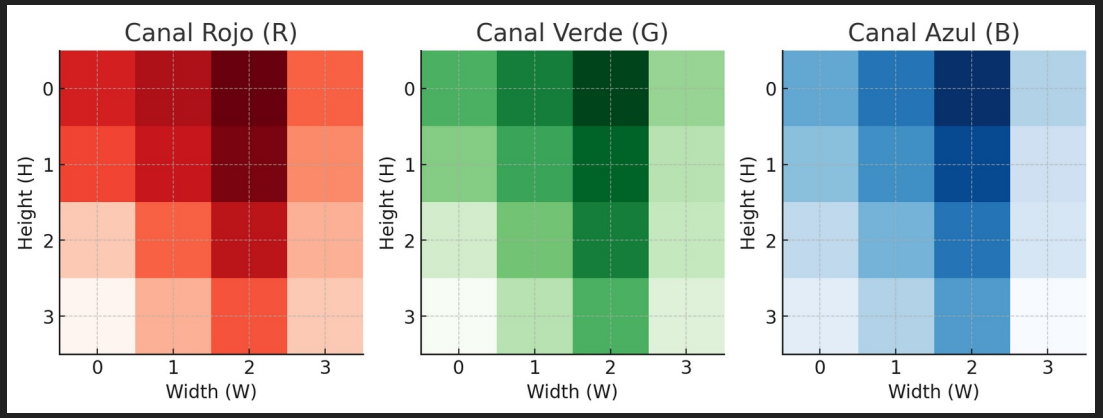


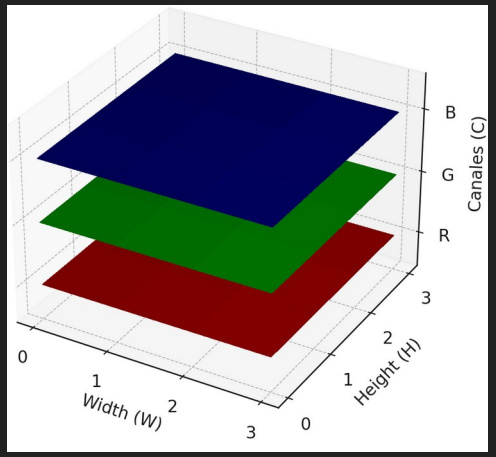
Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- ¿Qué representa el H W C?
  - H = Height (alto)
    - Cantidad de filas de píxeles en la imagen.
    - Dimensión vertical
    - H=224, la imagen tiene 224 píxeles de alto.
  - W = Width (ancho)
    - Cantidad de columnas de píxeles en la imagen.
    - Dimensión horizontal.
    - W=224, la imagen tiene 224 píxeles de ancho.
  - C = Channels (canales de color)
    - Profundidad de cada píxel (los valores que lo describen).
    - Para RGB, C=3 → Rojo, Verde, Azul.
    - Para escala de grises, C=1.
    - H y W indican la resolución espacial de la imagen: cuántos píxeles de alto (H) y cuántos de ancho (W). Es un plano 2D (alto × ancho).
    - Imagen tridimensional (H×W×C): 3 “capas” apiladas, para cada canal.
    - Capa 0 = Rojo (R) Capa 1 = Verde (G) Capa 2 = Azul (B).
    - Tercera dimensión no es “profundidad física”, número de canales de color que describen cada píxel.

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

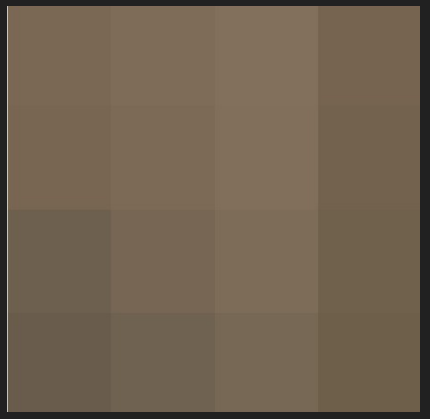
- ¿Podemos tener más canales?
  - SI.
  - Imagen con transparencia (RGBA):
    - H×W×4
    - (rojo, verde, azul y alfa = opacidad).
  - La tercera dimensión del tensor es el canal de color.
  - No representa un volumen físico como en un cubo 3D.
  - Son “capas de información” que se combinan para formar la imagen.
    - “Láminas” 2D apiladas (3 “láminas”, 4 “láminas”, etc)



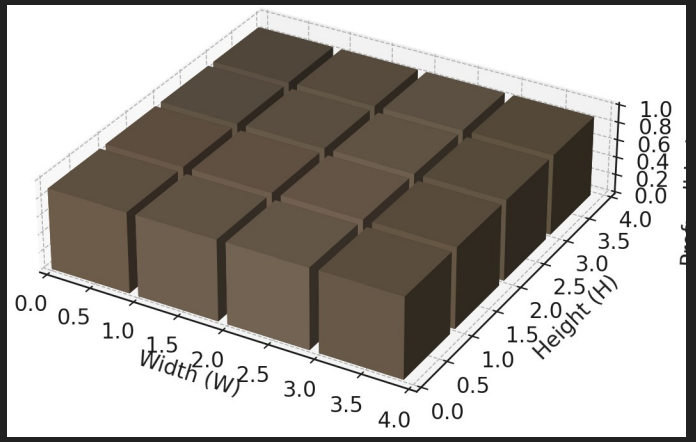


Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Fusión Plana
  - Reconstrucción RGB 2D (4×4) a partir de las tres capas R, G y B.



- Fusion 3D. Cada cubo corresponde a un píxel.
  - Color de cada cubo = resultado de combinar los tres canales (R,G,B).



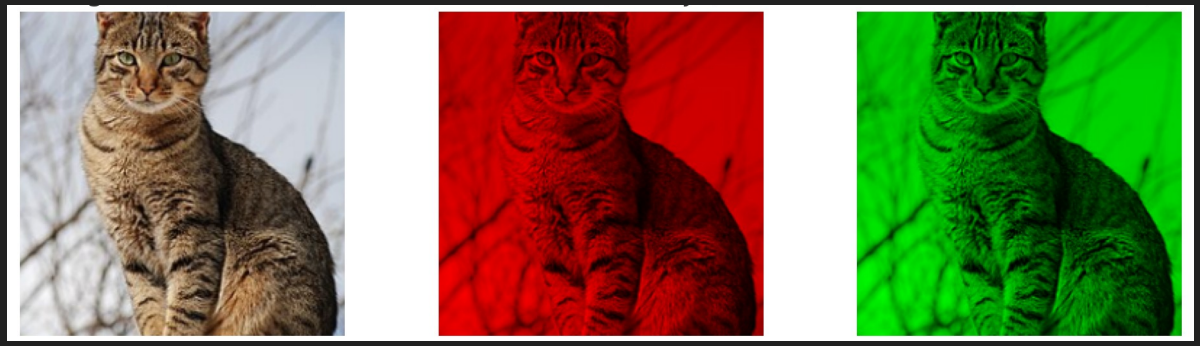
Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Preparación para preprocesamiento en modelos pre-entrenados en ImageNet (ResNet, VGG, EfficientNet).
- Entrenamiento desde cero: para definir pipeline.
- Transfer learning con modelos de PyTorch/TensorFlow: porque el backbone se entrenó así.
  - resize → crop → normalizar → tensor CHW
  - Resize con lado corto=256, manteniendo aspecto.
  - CenterCrop a 224×224.
    - Todos los modelos entrenados en ImageNet esperan entradas de 224×224 píxeles.
    - Se redimensiona y recorta en el centro para tener siempre la misma resolución sin deformar demasiado la imagen.
  - Vemos canales R, G, B mostrados como láminas en color (dejando un canal y poniendo los otros en 0). Y luego en escala de grises.
  - Tensor normalizado (CHW 3×224×224, float32) usando ImageNet mean/std.
    - Las librerías de deep learning como PyTorch esperan las imágenes en formato [Canales, Alto, Ancho] (CHW).
    - Para PIL o OpenCV suelen estar en HWC

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

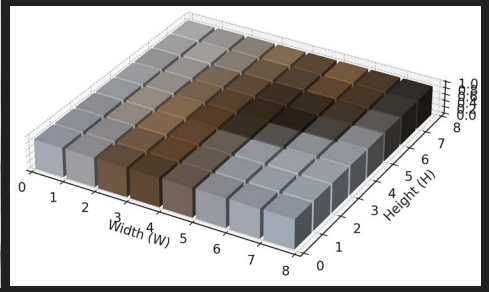
- Preparación para preprocesamiento en modelos pre-entrenados en ImageNet (ResNet, VGG, EfficientNet)
  - resize → crop → normalizar → tensor CHW
  - Normalización con mean/std de ImageNet
  - Durante el entrenamiento original, todas las imágenes se normalizaron con:
  - $\mu=[0.485,0.456,0.406], \sigma=[0.229,0.224,0.225]$
  - Aplicar la misma normalización.
  - La distribución de tus píxeles debe coincidir con la del entrenamiento → caso contrario el modelo produce resultados peores o directamente incoherentes.
  - Guardar en .npy para:
    - inspeccionar, documentar o reutilizar exactamente qué datos se le pasan al modelo.
    - se ve la entrada después de cada etapa.

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes



Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- CenterCrop 224×224
- Canales con cmap="gray".
  - En escala de grises se entiende mejor la intensidad
  - Canal R (224×224)
  - Canal G (224×224)
  - Canal B (224×224)
- RGB de 8×8 del recorte 224



Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- El paso de normalización no cambia los colores de la imagen.
- Cambia cómo se representan los valores dentro del tensor que va al modelo.
  - En crudo, cada canal está en uint8 $\rightarrow$ [0,255]
  - Se escala a [0,1] dividiendo por 255.
  - Luego se normaliza por canal restando la media y dividiendo por la desviación típica (valores fijos, calculados de todas las imágenes de ImageNet).
  - μ=[0.485,0.456,0.406] (medias de R,G,B en ImageNet)
  - σ=[0.229,0.224,0.225] (desvíos estándar)
  - Centra y ajusta la escala de los datos para que los modelos pre-entrenados trabajen mejor (estandarizar en estadística).
  - Son los promedios y desvíos estándar de los 1.2 millones de imágenes de entrenamiento.

$$x' = \frac{x - \mu}{\sigma}$$

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- Ejemplo. Estos valores son los que realmente entran al modelo de deep learning. Ajuste matemático para que las entradas tengan medias y escalas similares a las que vio el modelo durante su entrenamiento.
  - Píxel superior izquierdo del gato = [141, 140, 140]
  - Original: [141, 140, 140]
  - Escalado [0,1]: [0.553, 0.549, 0.549]
  - Normalizado:

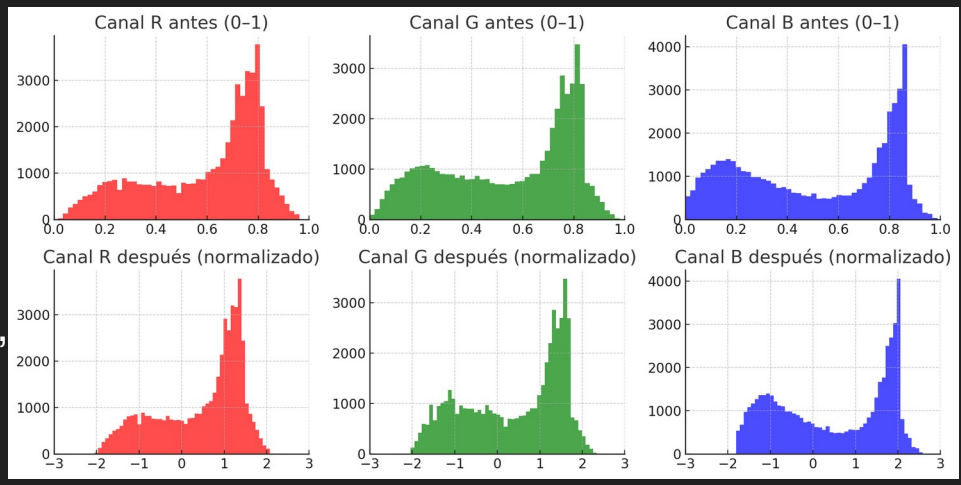
$$R' = \frac{0.553 - 0.485}{0.229} = 0.297, \quad G' = \frac{0.549 - 0.456}{0.224} = 0.415, \quad B' = \frac{0.549 - 0.406}{0.225} = 0.636$$

  - Si no se escalan, los gradientes y pesos en la red neuronal tendrían que adaptarse a números grandes (0–255).
  - Pasarlos a [0,1] hace que los valores estén en una escala más razonable y comparable con otros datasets.

Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- La normalización hace que cada canal tenga aproximadamente media 0 y varianza 1.
- Esto ayuda a:
  - Que los gradientes no exploten ni desaparezcan.
  - Que las neuronas de la red activen de manera más equilibrada.
  - Que el entrenamiento (o inferencia) sea más estable y rápido.
- Si uso una foto sin normalizar (valores 0–255):
  - La red puede saturar sus activaciones (ReLU, sigmoid, etc.).
  - Lo que antes era un contraste sutil puede parecer un "pico enorme" → predicción errónea.
  - Fila superior (antes, [0–1]):
    - Ej rojo anda entre 0.5–0.7

    en gran parte de la imagen.
  - Fila inferior (después, normalizado):
    - Llos valores ya no están en [0,1], sino centrados cerca de 0, con dispersión alrededor de ±1.
  - → Esto coincide con la distribución que vio la red en su entrenamiento.



Aprendizaje de Máquina (ML)
Representación Matemática de Imágenes

- ¿Por qué no usar la media de mi imagen?
  - Si uso la media y σ de la imagen individual:
    - Cada foto quedaría “centrada en sí misma”.
    - El modelo fue entrenado con otra estadística global.
    - Resultado → mala predicción (porque lo que “ve” la red no coincide con lo que vio al entrenarse).

$\hat{P}$