**Paso 1: La Imagen se Convierte en Números**

La imagen, que para nosotros es algo visual, se convierte en una **tabla de números** que la red pueda entender. Cada píxel de la imagen tiene un valor numérico que representa un color o intensidad. Si es una imagen a color, se divide en **3 capas**: rojo, verde y azul (RGB).

* **Ejemplo:** Una imagen de 64x64 píxeles se convierte en una tabla de **64x64x3** números (para cada color).

**Paso 2: Convolución – Buscando Patrones**

La red usa **filtros o kernels** (pequeñas matrices de números) para analizar bloques pequeños de la imagen, como porciones de **3x3 píxeles**. Estos filtros buscan **patrones simples**, como bordes o texturas.

* **Ejemplo:** Un filtro puede detectar **bordes verticales** mirando cómo cambian los valores en la imagen.

Cada vez que un filtro encuentra algo útil, genera un **nuevo conjunto de datos** llamado **mapa de características**.

**Paso 3: ReLU – Eliminando Números Irrelevantes**

Después de aplicar los filtros, algunos resultados pueden ser **números negativos** que no aportan mucho. Aquí entra la función **ReLU (Rectified Linear Unit)**, que convierte los **valores negativos en 0**, dejando solo los datos más importantes.

**Paso 4: Pooling – Reducción de Datos**

La red aún tiene mucha información, así que aplica **max pooling** para **reducir el tamaño** de los datos. Esto consiste en tomar solo el valor más alto de cada bloque pequeño (por ejemplo, de 2x2 píxeles).

* **Ejemplo:** Si encuentra varias formas de ruedas en diferentes partes, el pooling conserva la más clara y reduce la cantidad de datos a procesar.

**Paso 5: Capa Fully Connected – Interpretación Completa**

Después de varias capas de **convolución y pooling**, la red tiene suficiente información para **entender la imagen completa**. Ahora, convierte toda esa información en un **vector** (una lista de números) y la pasa a una **capa totalmente conectada (fully connected)**.

* **¿Qué hace esta capa?**: Junta todos los patrones encontrados (como ruedas, bordes, etc.) para hacer una **predicción** global de qué es la imagen.

**Paso 6: Softmax – Predicción Final**

En la última capa, la red usa **Softmax** para **convertir los resultados en probabilidades**. Esto le dice a la red qué tan segura está de cada posible resultado.

* **Ejemplo:**
  + Auto: 80%
  + Moto: 10%
  + Bicicleta: 10%

Aquí, la red cree que es **un auto** con 80% de seguridad.

**Paso 7: Entrenamiento – Ajuste de Filtros**

Durante el entrenamiento, la red compara su predicción con la **etiqueta correcta**. Si se equivoca, usa un proceso llamado **retropropagación** para **ajustar los filtros** y mejorar para la próxima vez.

* **Ejemplo:** Si predice "Moto" en lugar de "Auto", la red ajusta los filtros para que, en futuras imágenes, detecte mejor los **bordes y contornos** que corresponden a un auto.

**Paso 8: Repetición del Proceso hasta Aprender Bien**

Este proceso se repite **muchas veces** con diferentes imágenes. La red se entrena con miles de ejemplos, ajustando sus filtros cada vez que se equivoca. Con el tiempo, **aprende a reconocer objetos** con gran precisión.