**1. La imagen se convierte en números**

Cuando subimos una imagen, la computadora no la ve como nosotros. Lo que hace es convertirla en **una tabla de números**. Por ejemplo, cada color de un píxel se convierte en un número. Si la imagen tiene 64x64 píxeles y es a color (RGB), la computadora crea una **tabla con 3 capas** (una para cada color: rojo, verde y azul).

**Ejemplo:** Imagina que tomas una foto de un auto. Para la computadora, es solo una gran lista de números que representan cada color y parte del auto.

**2. La red busca patrones pequeños (Convolución)**

La computadora usa "lupas" (llamadas **filtros**) que revisan la imagen por partes pequeñas. Busca cosas como **bordes** o **texturas**. Esto es como mirar un rompecabezas por pedazos, tratando de encontrar piezas que encajen.

**Ejemplo:** La red puede encontrar las **ruedas** del auto en la imagen o las líneas del chasis.

**3. Se eliminan los datos irrelevantes (ReLU)**

A veces, la red encuentra valores que no sirven (como números negativos). Estos valores se **eliminan** para que la red no se confunda. Esto es como borrar los garabatos que no aportan al dibujo final.

**4. Se hace más pequeña la tabla de datos (Pooling)**

Ahora la red tiene mucha información. Para manejarla mejor, toma solo los datos más importantes. Usa un truco llamado **max pooling**, que guarda solo los números más altos de cada parte de la tabla.

**Ejemplo:** Si encontró varias formas parecidas a ruedas, solo se guarda la más clara y grande.

**5. La red interpreta la imagen completa (Capa Fully Connected)**

Después de encontrar las partes importantes (como las ruedas y bordes), la red junta toda esa información y trata de **adivinar** qué es la imagen completa. Es como juntar las piezas de un rompecabezas para ver si es un auto.

**6. La red da su predicción (Softmax)**

Al final, la red usa **Softmax** para decir qué cree que hay en la imagen y con cuánta seguridad.

**Ejemplo:** "Creo que es un auto con un 80% de certeza. También podría ser una moto (10%) o una bicicleta (10%)."

**7. Google entrena la red con muchas imágenes**

Antes de que la red pueda funcionar bien, necesita **aprender**. Google le pasa **miles de imágenes etiquetadas** (como fotos de autos, bicicletas y motos) para que la red mejore con el tiempo. Si se equivoca, ajusta sus cálculos hasta acertar más.

**8. Los desarrolladores usan la API**

Una vez que la red ha aprendido, los desarrolladores pueden **subir imágenes a la API de Vision de Google**. La API analiza las imágenes y devuelve su predicción.

**Ejemplo práctico:** Un desarrollador sube la foto de un perro, y la API responde: “Perro – 95%”, indicando que está casi segura de que es un perro.