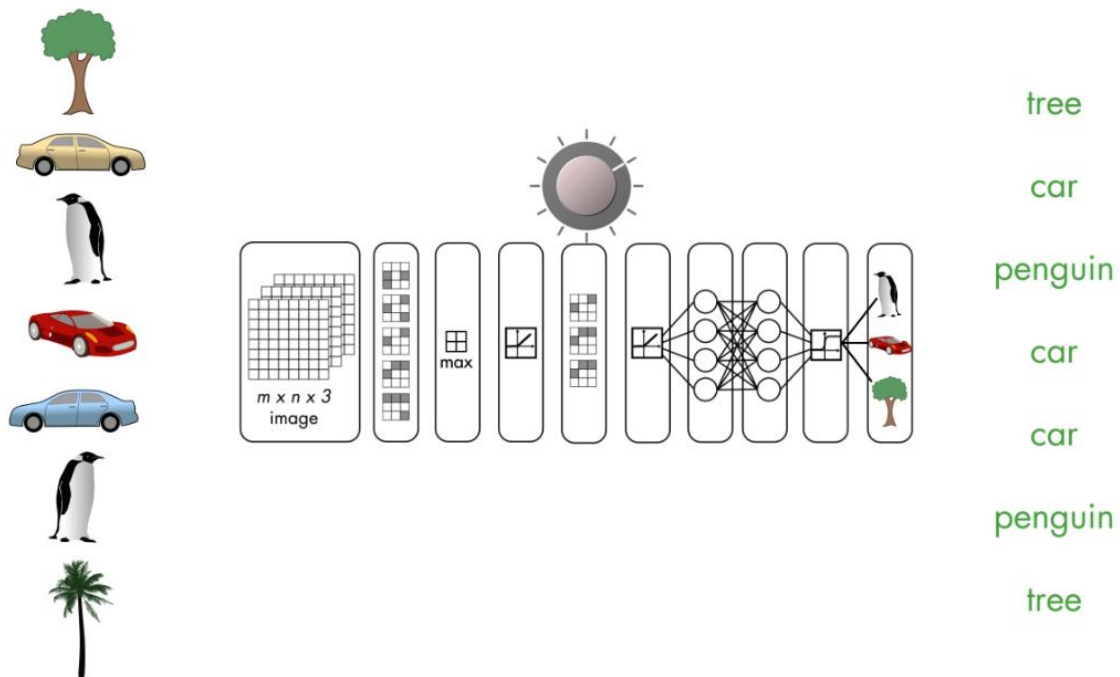


PREPARACIÓN DE LOS DATOS DE ENTRENAMIENTO

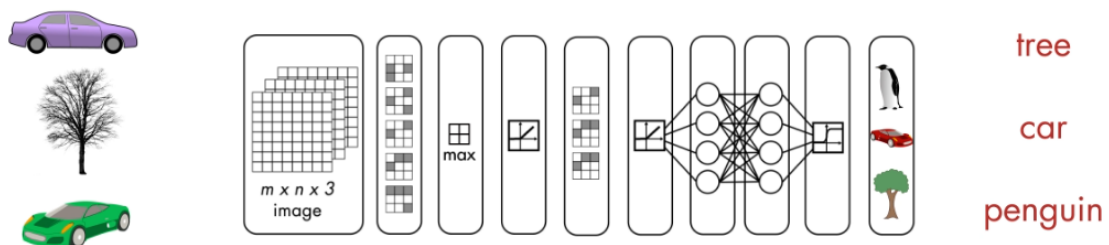
Dividir Datos para Entrenamiento y para Prueba

Entrenamiento frente a pruebas

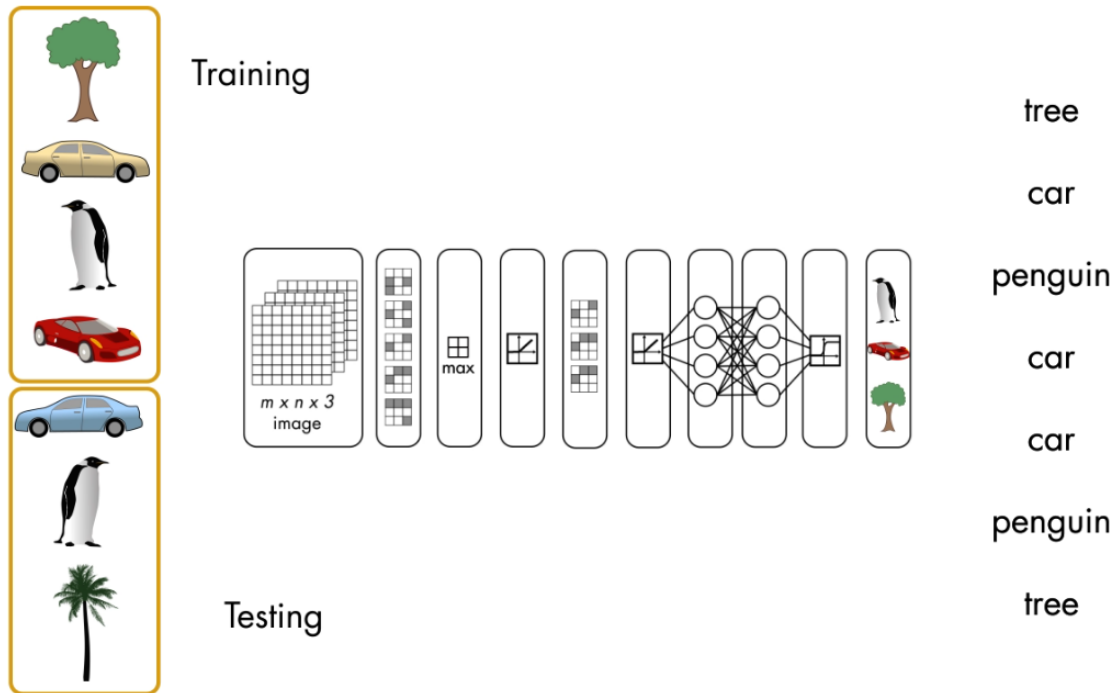
Durante el entrenamiento, los pesos de la red se ajustan para que la red aprenda a asociar las entradas proporcionadas con las salidas ofrecidas. Pero, incluso si la red clasifica correctamente todos los datos de entrenamiento, ¿ha aprendido algo realmente?



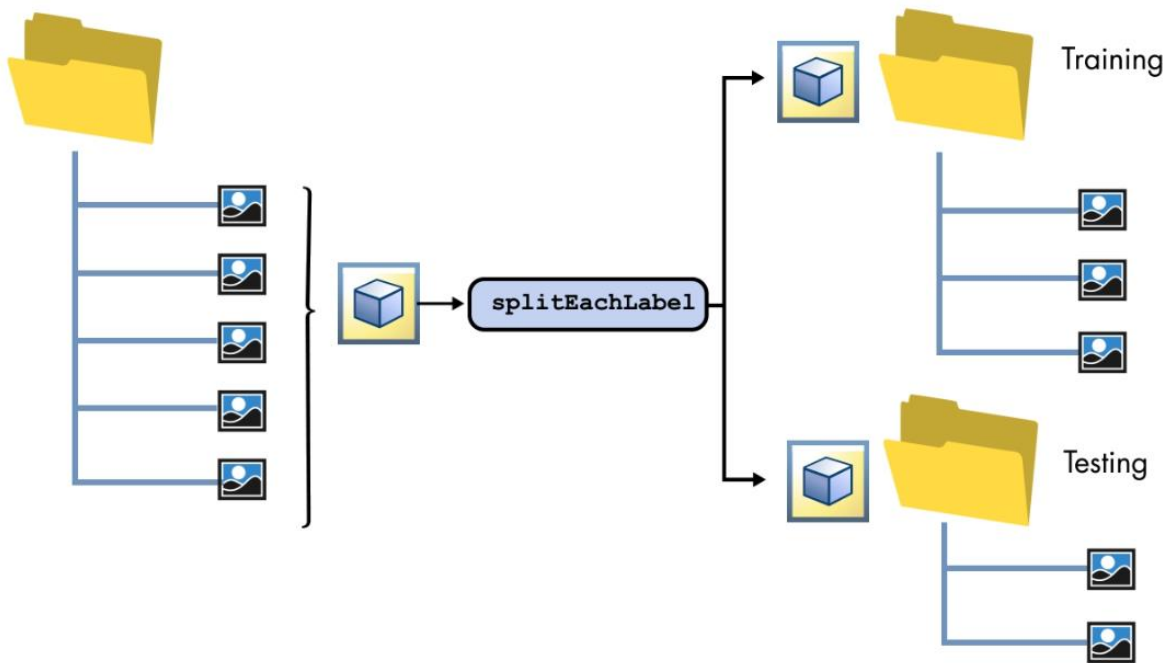
El uso real de una red es aplicarla a nuevas imágenes, imágenes que no haya visto antes.



Es una buena práctica reservar algunos datos de entrenamiento para hacer pruebas. Este conjunto reservado para pruebas no se utiliza para entrenar la red, sino solo con el fin de evaluar su rendimiento.



En esta actividad, usará la función `splitEachLabel` para dividir las imágenes de la carpeta `Flowers` en dos almacenes de datos independientes: uno para entrenamiento y otro para pruebas.



Actividad 1

Puede utilizar la función `splitEachLabel` para dividir las imágenes de un almacén de datos en dos almacenes de datos independientes.

```
[ds1,ds2] = splitEachLabel(imds,p)
```

La proporción `p` (un valor de 0 a 1) indica la proporción de imágenes con cada etiqueta de `imds` que debería estar contenida en `ds1`. El resto de archivos se asignan a `ds2`.

Tarea: Cree un almacén de datos para todas las imágenes de las subcarpetas de la carpeta `Flowers`

```
ds_flores = imageDatastore("Flowers", ...  
    "IncludeSubfolders",true, ...  
    "LabelSource","foldernames")
```

Divida el almacén de datos `ds_flores` en dos almacenes de datos, `flor_entrena` y `flor_prueba`, de manera que el 60% de los archivos de cada categoría esté en `flor_entrena`.

Actividad 2

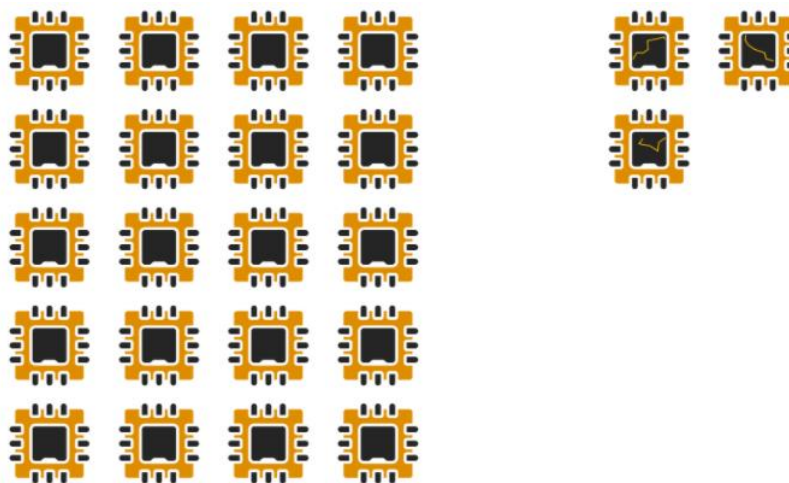
De manera predeterminada, `splitEachLabel` mantiene los archivos ordenados. Puede mezclar de forma aleatoria los archivos, para lo cual deberá añadir el indicador `"randomized"`.

```
[ds1,ds2] = splitEachLabel(imds,p,"randomized")
```

Tarea: Divida el almacén de datos `ds_flores` en dos almacenes de datos, `flor_entrena` y `flor_prueba`, tal que, de manera aleatoria, el 80% de los archivos de cada categoría esté en `flor_entrena`.

Datos de entrenamiento no compensados

En algunas aplicaciones es habitual tener muchas más imágenes de una clase que de otras. Por ejemplo, al intentar detectar defectos, por lo general es fácil obtener muchas imágenes sin defectos, pero más difícil obtenerlas con defectos.



En este caso, dividir los datos proporcionalmente por clase dará como resultado que la red recibirá entrenamiento fundamentalmente con imágenes sin defectos. Esto puede sesgar el entrenamiento

y dar como resultado una red que “cumple con los porcentajes” en lugar de aprender realmente a identificar características que indiquen la presencia de defectos.

Para evitarlo, puede resultar útil dividir los datos de manera que las imágenes de entrenamiento tengan el mismo número de cada clase.

Actividad 3

Cuando `p` es un valor de 0 a 1, se interpreta como una proporción. Las imágenes entonces se dividen de manera que cada etiqueta se divida proporcionalmente. También puede especificar un número exacto de archivos que tomar de cada etiqueta para asignarlos a `ds1`.

```
[ds1,ds2] = splitEachLabel(imds,n)
```

Así se asegura que todas las etiquetas de `ds1` tienen `n` imágenes, incluso aunque no todas las categorías contengan el mismo número de imágenes.

Tarea: Divida el almacén de datos `ds_flores` en dos almacenes de datos, `flor_entrena` y `flor_prueba`, de manera que 50 archivos de cada categoría estén en `flor_entrena`.

Tarea adicional

¡Listo! Dividir las imágenes en dos conjuntos facilita la utilización de un conjunto para entrenar su red y el otro para probar el rendimiento del resultado.

También puede utilizar un conjunto de “validación” para supervisar el rendimiento de la red durante el entrenamiento. En ese caso, puede dividir los datos en tres conjuntos: uno para el entrenamiento, otro para la validación durante el entrenamiento y otro más para una prueba independiente del resultado final. Pruebe a utilizar `splitEachLabel` para dividir las imágenes de `Flowers` en varios conjuntos. Asigne varios valores a `p` o `n` como entradas y pida el número adecuado de almacenes de datos como salidas.

Archivos requeridos:

```
\Flowers\
```