```
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
```

```
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.l
oad_data()
train_images, test_images = train_images / 255.0, test_images / 255.0
```

```
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer','dog', 'frog',
'horse', 'ship', 'truck']
```

```
plt.figure(figsize=(10,10))
for i in range(10):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i])
    plt.xlabel(class_names[train_labels[i][0]])
plt.show()
```

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3
)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))
```

```
model.summary()
```

```
/opt/anaconda3/lib/python3.12/site-packages/keras/src/layers/convolutional/ba
se_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argumen
```

```
t to a layer. When using Sequential models, prefer using an `Input(shape)` ob
ject as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
Model: "sequential"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━┓
┃ Layer (type)                     ┃ Output Shape            ┃        Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━┩
│ conv2d (Conv2D)                  │ (None, 30, 30, 32)      │            896 │
├──────────────────────────────────┼─────────────────────────┼────────────────┤
│ max_pooling2d (MaxPooling2D)     │ (None, 15, 15, 32)      │              0 │
├──────────────────────────────────┼─────────────────────────┼────────────────┤
│ conv2d_1 (Conv2D)                │ (None, 13, 13, 64)      │         18,496 │
├──────────────────────────────────┼─────────────────────────┼────────────────┤
│ max_pooling2d_1 (MaxPooling2D)   │ (None, 6, 6, 64)        │              0 │
├──────────────────────────────────┼─────────────────────────┼────────────────┤
│ conv2d_2 (Conv2D)                │ (None, 4, 4, 64)        │         36,928 │
├──────────────────────────────────┼─────────────────────────┼────────────────┤
│ flatten (Flatten)                │ (None, 1024)            │              0 │
├──────────────────────────────────┼─────────────────────────┼────────────────┤
│ dense (Dense)                    │ (None, 64)              │         65,600 │
├──────────────────────────────────┼─────────────────────────┼────────────────┤
│ dense_1 (Dense)                  │ (None, 10)              │            650 │
└──────────────────────────────────┴─────────────────────────┴────────────────┘

 Total params: 122,570 (478.79 KB)

 Trainable params: 122,570 (478.79 KB)

 Non-trainable params: 0 (0.00 B)
```

```python
model.compile(optimizer='adam',loss=tf.keras.losses.SparseCategoricalCrossent
ropy(from_logits=True),metrics=['accuracy'])
epochs = 1
h = model.fit(train_images, train_labels, epochs=epochs, validation_data=(tes
t_images, test_labels))
```

```
1563/1563 ━━━━━━━━━━━━━━━━━━━━12s 8ms/step - accuracy: 0.3300 - loss: 1.8015
- val_accuracy: 0.5138 - val_loss: 1.3597
```