Ex No: 5 TRANSFER LEARNING WITH CNN AND VISUALIZATION

AIM:

To build a convolutional neural network with transfer learning and perform visualization

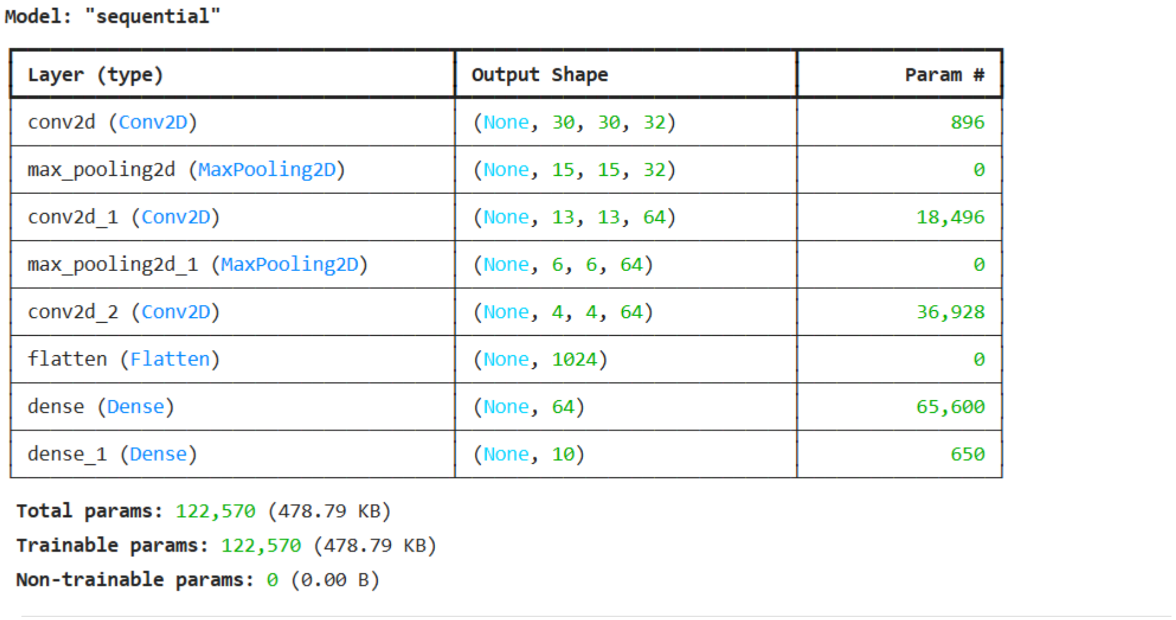
PROCEDURE:

1. Download and load the dataset.
2. Perform analysis and preprocessing of the dataset.
3. Build a simple neural network model using Keras/TensorFlow.
4. Compile and fit the model.
5. Perform prediction with the test dataset.
6. Calculate performance metrics.

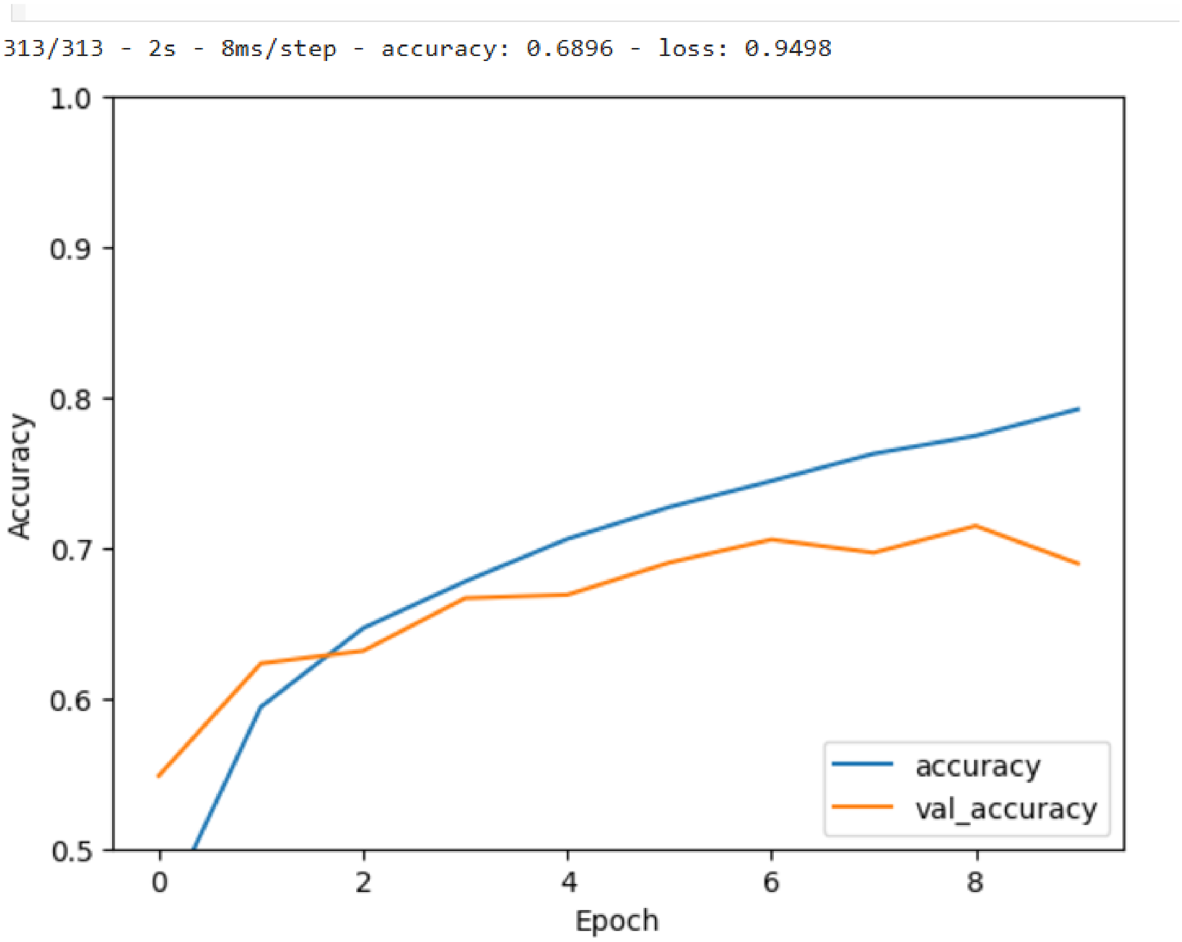
PROGRAM:

```
import tensorflow as tf


from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()


# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0
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.summary()
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10))
model.summary()
model.compile(optimizer='adam',
        loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics=['accuracy'])

history = model.fit(train_images, train_labels, epochs=10,
            validation_data=(test_images, test_labels))

plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0.5, 1])
plt.legend(loc='lower right')

test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)

print(test_acc)
```

OUTPUT:



```
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)
```



```
313/313 - 2s - 8ms/step - accuracy: 0.6896 - loss: 0.9498
```

RESULT:

Thus a convolutional neural network with transfer learning and perform visualization is built.