Ex.No.5

TRANSFER LEARNING WITH CNN

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 transfer learning model with convolutional neural network 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.models import Sequential
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
```

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape(-1, 28, 28, 1) / 255.0
x_test = x_test.reshape(-1, 28, 28, 1) / 255.0

x_train = tf.image.resize(x_train, (32, 32))
x_test = tf.image.resize(x_test, (32, 32))
x_train = tf.image.grayscale_to_rgb(x_train)
x_test = tf.image.grayscale_to_rgb(x_test)

y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)

base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(32, 32, 3))
base_model.trainable = False

model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
loss='categorical_crossentropy',
metrics=['accuracy'])
```

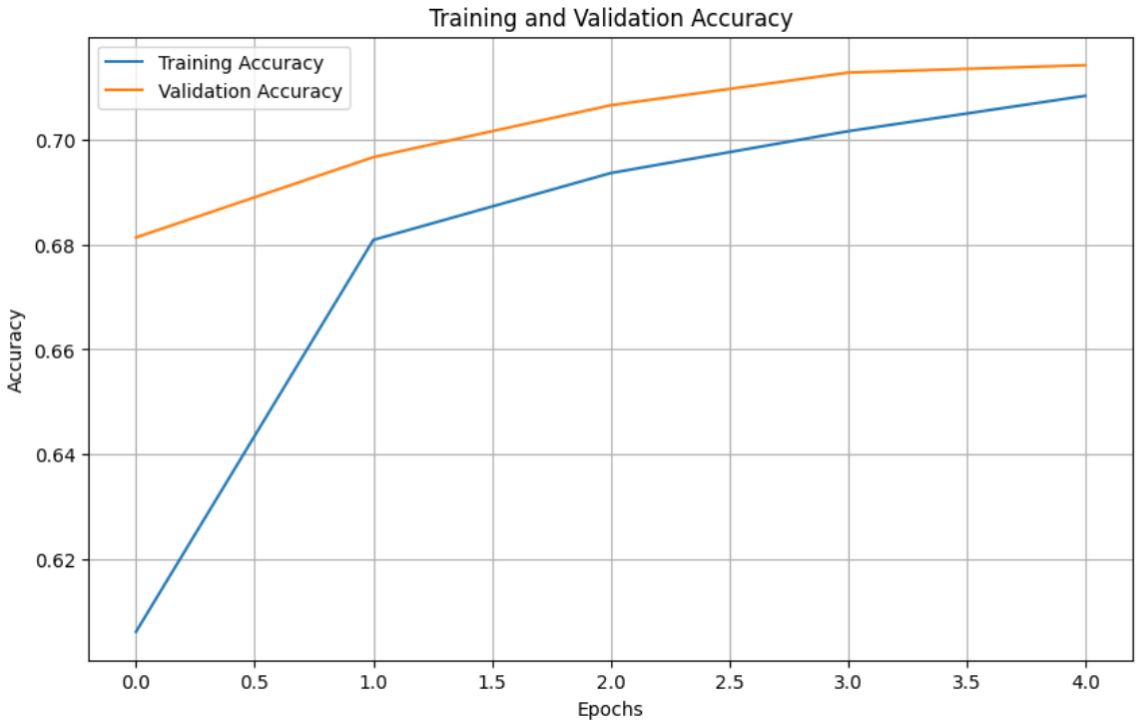
```
history = model.fit(x_train, y_train, epochs=5, validation_split=0.2, batch_size=64, verbose=2)


test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_acc:.4f}')


plt.figure(figsize=(10, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT:

```
Epoch 1/5
750/750 - 60s - 79ms/step - accuracy: 0.6061 - loss: 1.2020 - val_accuracy: 0.6814 - val_loss: 0.9596
Epoch 2/5
750/750 - 81s - 108ms/step - accuracy: 0.6809 - loss: 0.9455 - val_accuracy: 0.6967 - val_loss: 0.9044
Epoch 3/5
750/750 - 82s - 109ms/step - accuracy: 0.6937 - loss: 0.9021 - val_accuracy: 0.7067 - val_loss: 0.8749
Epoch 4/5
750/750 - 62s - 82ms/step - accuracy: 0.7017 - loss: 0.8736 - val_accuracy: 0.7129 - val_loss: 0.8513
Epoch 5/5
750/750 - 52s - 69ms/step - accuracy: 0.7085 - loss: 0.8508 - val_accuracy: 0.7143 - val_loss: 0.8374
313/313 ━━━━━━━━━━━━━━━━━━━━ 12s 38ms/step - accuracy: 0.6979 - loss: 0.8841
Test accuracy: 0.7065
```



RESULT:

Thus, a convolutional neural network with transfer learning was successfully implemented.