Perfect 👌 Here's a **set of exercises on Transfer Learning** (progressively going from easy → medium → advanced). I'll design them in a way that you can practice concepts while applying them to real datasets like **Cats vs Dogs, CIFAR-10, or your own dataset**.

📝 Exercises on Transfer Learning

Section 1: Basics of Transfer Learning

1. **Warm-up**: Load a pretrained model (e.g., VGG16 or ResNet50) from `keras.applications`. Print its summary.
   - Q: How many layers are trainable by default?
   - Q: Which layer has the maximum number of parameters?
2. **Feature Extraction**: Use VGG16 as a fixed feature extractor (set `trainable=False`) and train only a new dense classifier on top for Cats vs Dogs.
   - Q: Compare accuracy with and without data augmentation.
3. **Freezing & Unfreezing**: Freeze all convolutional layers, then unfreeze the last block (like `block5_conv`). Train again.
   - Q: How did validation accuracy change compared to full freeze?

Section 2: Practical Training

4. Train ResNet50 with ImageNet weights on CIFAR-10. Replace its top with your own dense layers.
   - Q: What is the effect of using `GlobalAveragePooling2D` vs `Flatten()`?
5. Add **EarlyStopping and ReduceLROnPlateau** callbacks while fine-tuning InceptionV3.
   - Q: How do callbacks help prevent overfitting?
6. Compare training time and performance of MobileNetV2 vs VGG16 on the same dataset.
   - Q: Which one is faster? Which one generalizes better?

Section 3: Visualization & Evaluation

7. Plot training & validation curves (accuracy, loss) for two transfer learning runs:
   - (a) All layers frozen

   - (b) Last two blocks unfrozen
   - Q: Which strategy gives better generalization? Why?

8. Save and reload a fine-tuned model. Test it on 5 new images downloaded from the internet.

   - Q: How well does the model generalize to unseen real-world images?

```
Start coding or generate with AI.
```

Perfect 👌 Let's build this as a **Transfer Learning workbook**: exercises + solutions (with code + explanation). I'll give **solutions for each task** step by step (Keras/TensorFlow).

✅ Transfer Learning Exercises with Solutions

Section 1: Basics of Transfer Learning

Exercise 1 – Load Pretrained Model

Task: Load VGG16 pretrained on ImageNet. Print summary.

```
from tensorflow.keras.applications import VGG16

# Load pretrained VGG16 without top
model = VGG16(weights='imagenet', include_top=True)
model.summary()
```

Answer:

- By default, all layers are trainable (`trainable=True`).
- The layer with max params is usually `fc1` or `fc2` dense layers.

Exercise 2 – Feature Extraction (Freeze All)

Task: Use VGG16 as fixed feature extractor.

```
from tensorflow.keras import layers, models

base_model = VGG16(weights="imagenet", include_top=False, input_shape=(150,150,3))
base_model.trainable = False  # Freeze

# Add new classifier
```

```
x = layers.Flatten()(base_model.output)
x = layers.Dense(256, activation="relu")(x)
x = layers.Dense(1, activation="sigmoid")(x)
model = models.Model(base_model.input, x)

model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
```

Answer:

- With augmentation → higher validation accuracy.
- Without augmentation → risk of overfitting.

Exercise 3 – Freeze vs Unfreeze

Task: Unfreeze last block of VGG16.

```
# Unfreeze last block
for layer in base_model.layers:
    if "block5" in layer.name:
        layer.trainable = True
    else:
        layer.trainable = False

model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
```

Answer:

- Validation accuracy usually improves (because last conv layers adapt to dataset).