Classification

Data preparation:

Techniques applied include:

- Rescaling: Images are normalized by dividing by 255.
- Rotation: Random rotations up to 30 degrees.
- Shear: Random shearing transformation with a magnitude of 0.2.
- Zoom: Random zooming with a range of 0.2.
- Horizontal Flip: Randomly flipping images horizontally.
- Width and Height Shift: Randomly shifting the width and height by 20%.
- Fill Mode: Filling in newly created pixels using the nearest available pixel values.

VAL:

- **ImageDataGenerator** is created for the validation data, performing scaling only.
- Rescaling: Images are normalized by dividing by 255.

Training and Val Generator:

- 'train_generator' is created using the augmented data to flow from a directory.
- It's set to target images to a size of 224x224 pixels.
- Batch size is set to 32 for training.
- Class mode is categorical for multiclass classification.
- Shuffle is enabled to data during training but in val Don't Shuffle.

```
train_data_augmented = ImageDataGenerator(
        rescale=1./255,
        rotation_range = 30,
        shear_range = 0.2,
        zoom_range = 0.2,
        horizontal_flip = True,
        width_shift_range=0.2,
        height_shift_range=0.2,
        fill_mode='nearest'
)
train_generator = train_data_augmented.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=True
)
```

```
Found 1162 images belonging to 5 classes.
```

```
val_data_generator = ImageDataGenerator(
        rescale=1./255
)

val_generator = val_data_generator.flow_from_directory(
    val_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)
```

```
Found 308 images belonging to 5 classes.
```

Model:

1) ResNet50:

- Sequential Model: A linear stack of layers.
- Layers Added:
    - The ResNet50 base model is added as the first layer.
    - Flatten Layer: Converts the multi-dimensional output of the ResNet50 base into a flat vector for further processing.
    - Dense Layers: Fully connected layers added on top of the flattened output.
        - Dense layer with 256 units and ReLU activation.
        - Final Dense layer with 5 units and softmax activation for multiclass classification (assuming 5 classes).

```
model = Sequential()
model.add(base_model)
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(5, activation='softmax'))

# Compile the model
model.compile(optimizer=Adam(lr=0.001), loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // 32,
    epochs=10,
    validation_data=val_generator,
    validation_steps=val_generator.samples // 32
)
```

```
Epoch 1/10
36/36 [==============================] - 25s 485ms/step - loss: 6.4855 - accuracy: 0.3283 - val_loss: 1.3992 - val_accuracy: 0.5694
Epoch 2/10
36/36 [==============================] - 16s 438ms/step - loss: 1.3876 - accuracy: 0.4947 - val_loss: 1.1367 - val_accuracy: 0.5243
Epoch 3/10
36/36 [==============================] - 16s 435ms/step - loss: 1.4263 - accuracy: 0.5133 - val_loss: 1.7231 - val_accuracy: 0.4757
Epoch 4/10
36/36 [==============================] - 16s 437ms/step - loss: 1.2562 - accuracy: 0.5168 - val_loss: 0.8181 - val_accuracy: 0.7292
Epoch 5/10
36/36 [==============================] - 16s 432ms/step - loss: 1.2392 - accuracy: 0.5265 - val_loss: 0.7315 - val_accuracy: 0.7500
Epoch 6/10
36/36 [==============================] - 16s 437ms/step - loss: 1.0992 - accuracy: 0.5805 - val_loss: 0.9095 - val_accuracy: 0.6042
Epoch 7/10
36/36 [==============================] - 16s 440ms/step - loss: 1.0903 - accuracy: 0.5752 - val_loss: 0.8223 - val_accuracy: 0.6493
Epoch 8/10
36/36 [==============================] - 16s 434ms/step - loss: 1.0868 - accuracy: 0.5779 - val_loss: 0.7592 - val_accuracy: 0.7743
Epoch 9/10
36/36 [==============================] - 15s 426ms/step - loss: 1.0905 - accuracy: 0.5823 - val_loss: 0.8699 - val_accuracy: 0.7188
Epoch 10/10
36/36 [==============================] - 16s 441ms/step - loss: 0.9874 - accuracy: 0.6195 - val_loss: 0.7813 - val_accuracy: 0.7396
```

2) CNN

- Convolutional Neural Network (CNN): Utilizes convolutional layers for feature extraction, followed by max pooling for spatial reduction, and fully connected layers for classification.
- Pooling Layers: Employed to downsample the spatial dimensions, capturing the most important features.
- Flatten Layer: Reshapes the output for passing it to fully connected layers.
- Activation Functions: ReLU activation in convolutional layers to introduce non-linearity.
- Softmax Activation: Used in the output layer for multi-class classification, providing probabilities for each class.
- RMSprop Optimizer: Adaptive learning rate optimization technique.
- Learning Rate Reduction: Adjusts learning rate dynamically based on validation accuracy plateaus, potentially aiding convergence.

```python
# Define the model
CNN_Model = Sequential()

CNN_Model.add(Conv2D(16, (3, 3), strides=1, activation='relu', input_shape=(224, 224, 3)))
CNN_Model.add(MaxPooling2D())

CNN_Model.add(Conv2D(32, (3, 3), strides=1, activation='relu'))
CNN_Model.add(MaxPooling2D())

CNN_Model.add(Conv2D(16, (3, 3), strides=1, activation='relu'))
CNN_Model.add(MaxPooling2D())

CNN_Model.add(Flatten())

CNN_Model.add(Dense(256, activation='relu'))  # Adjusted units to match Flatten layer output
CNN_Model.add(Dense(5, activation='softmax'))  # Assuming 5 output classes, adjust if needed
```

```
Epoch 1/30
37/37 [==============================] - 18s 423ms/step - loss: 1.5432 - accuracy: 0.5422 - val_loss: 0.8165 - val_accuracy: 0.7208 - lr: 0.0010
Epoch 2/30
37/37 [==============================] - 16s 424ms/step - loss: 0.9747 - accuracy: 0.6351 - val_loss: 2.0923 - val_accuracy: 0.1234 - lr: 0.0010
Epoch 3/30
37/37 [==============================] - 16s 431ms/step - loss: 0.9088 - accuracy: 0.6661 - val_loss: 0.6570 - val_accuracy: 0.7468 - lr: 0.0010
Epoch 4/30
37/37 [==============================] - 16s 423ms/step - loss: 0.7759 - accuracy: 0.7031 - val_loss: 0.5768 - val_accuracy: 0.7890 - lr: 0.0010
Epoch 5/30
37/37 [==============================] - 16s 430ms/step - loss: 0.7765 - accuracy: 0.7091 - val_loss: 0.5922 - val_accuracy: 0.7630 - lr: 0.0010
Epoch 6/30
37/37 [==============================] - 16s 422ms/step - loss: 0.7439 - accuracy: 0.7289 - val_loss: 0.5180 - val_accuracy: 0.8084 - lr: 0.0010
Epoch 7/30
37/37 [==============================] - 16s 420ms/step - loss: 0.6823 - accuracy: 0.7392 - val_loss: 0.5750 - val_accuracy: 0.7825 - lr: 0.0010
Epoch 8/30
37/37 [==============================] - 16s 419ms/step - loss: 0.6793 - accuracy: 0.7298 - val_loss: 0.7117 - val_accuracy: 0.7013 - lr: 0.0010
Epoch 9/30
37/37 [==============================] - 16s 422ms/step - loss: 0.6476 - accuracy: 0.7496 - val_loss: 0.4806 - val_accuracy: 0.8182 - lr: 0.0010
Epoch 10/30
37/37 [==============================] - 16s 423ms/step - loss: 0.6601 - accuracy: 0.7539 - val_loss: 0.5960 - val_accuracy: 0.7500 - lr: 0.0010
Epoch 11/30
37/37 [==============================] - 16s 427ms/step - loss: 0.6429 - accuracy: 0.7487 - val_loss: 0.5313 - val_accuracy: 0.7922 - lr: 0.0010
Epoch 12/30
37/37 [==============================] - ETA: 0s - loss: 0.6937 - accuracy: 0.7418
Epoch 12: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
...
Epoch 30/30
37/37 [==============================] - ETA: 0s - loss: 0.4167 - accuracy: 0.8322
Epoch 30: ReduceLROnPlateau reducing learning rate to 1e-05.
37/37 [==============================] - 16s 432ms/step - loss: 0.4167 - accuracy: 0.8322 - val_loss: 0.4415 - val_accuracy: 0.8052 - lr: 1.5625e-05
```

3) Xception_model

- Transfer Learning with Xception: Leveraging a powerful pre-trained Xception model for feature extraction.
- Global Average Pooling: Condenses spatial dimensions into a vector to be fed into fully connected layers.
- Dropout Regularization: Helps prevent overfitting by randomly deactivating a percentage of neurons during training.
- Flatten Layer: Transforms the tensor shape from 3D to 1D for the subsequent dense layers.
- Additional Dense Layers: Allows for learning more complex patterns by adding fully connected layers with varying units and activation functions.
- Softmax Activation: Used in the output layer for multi-class classification to obtain class probabilities.

```
Xception_model = Sequential([
    pretrained_base_model,
    GlobalAveragePooling2D(),
    Dropout(0.5),
    Flatten(),                      # Adding Flatten layer to transition from 3D to 1D tensor
    Dense(1024, activation='relu'),  # Additional Dense layer
    Dropout(0.5),
    Dense(256, activation='relu'),   # Another Dense layer
    Dense(5, activation='softmax')   # Output layer for 5 classes
])
```

```
Epoch 1/30
37/37 [==============================] - 41s 593ms/step - loss: 0.7877 - accuracy: 0.6885 - val_loss: 0.3478 - val_accuracy: 0.9026
Epoch 2/30
37/37 [==============================] - 22s 588ms/step - loss: 0.2329 - accuracy: 0.9277 - val_loss: 0.1817 - val_accuracy: 0.9610
Epoch 3/30
37/37 [==============================] - 21s 560ms/step - loss: 0.1840 - accuracy: 0.9441 - val_loss: 0.2418 - val_accuracy: 0.9545
Epoch 4/30
37/37 [==============================] - 21s 550ms/step - loss: 0.1237 - accuracy: 0.9656 - val_loss: 0.3058 - val_accuracy: 0.9545
Epoch 5/30
37/37 [==============================] - 21s 559ms/step - loss: 0.1210 - accuracy: 0.9656 - val_loss: 0.7074 - val_accuracy: 0.9156
Epoch 6/30
37/37 [==============================] - 21s 566ms/step - loss: 0.0673 - accuracy: 0.9802 - val_loss: 0.4150 - val_accuracy: 0.9643
Epoch 7/30
37/37 [==============================] - 21s 564ms/step - loss: 0.0942 - accuracy: 0.9768 - val_loss: 0.3473 - val_accuracy: 0.9610
Epoch 8/30
37/37 [==============================] - 21s 558ms/step - loss: 0.0537 - accuracy: 0.9811 - val_loss: 0.3777 - val_accuracy: 0.9513
Epoch 9/30
37/37 [==============================] - 21s 559ms/step - loss: 0.0800 - accuracy: 0.9776 - val_loss: 0.4306 - val_accuracy: 0.9578
Epoch 10/30
37/37 [==============================] - 21s 563ms/step - loss: 0.0512 - accuracy: 0.9854 - val_loss: 0.3734 - val_accuracy: 0.9675
Epoch 11/30
37/37 [==============================] - 21s 563ms/step - loss: 0.0531 - accuracy: 0.9888 - val_loss: 0.5257 - val_accuracy: 0.9610
Epoch 12/30
37/37 [==============================] - 21s 561ms/step - loss: 0.0714 - accuracy: 0.9836 - val_loss: 1.4782 - val_accuracy: 0.8961
Epoch 13/30
...
Epoch 29/30
37/37 [==============================] - 21s 559ms/step - loss: 0.0185 - accuracy: 0.9940 - val_loss: 0.7525 - val_accuracy: 0.9481
Epoch 30/30
37/37 [==============================] - 21s 561ms/step - loss: 0.0229 - accuracy: 0.9957 - val_loss: 0.4329 - val_accuracy: 0.9675
```

visualization: