Pattern Recognition

Assignment 1

Report

Submitted by:

Judy Nabil 7575

Amr Abdelaziz 7447

Ahmed Hany 7387

Data Loading:

Data Splitting:

5 folds

Data Preprocessing:

- **Resize**

```
Shape of data: (100, 128, 128, 3) Labels shape: (100,)
```

- **Normalized**

```
img_array = np.array(img_resized) / 255.0  # normalize
```

- **Augmentation**

```
datagen = ImageDataGenerator(
    rotation_range=30,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True
)
```

Creating the classifier:

```
def build_model(learning_rate=0.001, dropout_rate=0.5):
    model = models.Sequential([
        layers.Conv2D(32, (3,3), activation='relu', input_shape=(IMG_SIZE[0], IMG_SIZE[1], 3)),
        layers.MaxPooling2D((2,2)),
        layers.Conv2D(64, (3,3), activation='relu'),
        layers.MaxPooling2D((2,2)),
        layers.Conv2D(128, (3,3), activation='relu'),
        layers.MaxPooling2D((2,2)),
        layers.Flatten(),
        layers.Dense(128, activation='relu'),
        layers.Dropout(dropout_rate),
        layers.Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer=optimizers.Adam(learning_rate=learning_rate),
                  loss='binary_crossentropy',
                  metrics=['accuracy'])
    return model
```

Trying combination: learning_rate=0.005, dropout_rate=0.5, batch_size=8

Fold 1 Validation Loss: 0.6931, Accuracy: 0.5000

Fold 2 Validation Loss: 0.3159, Accuracy: 0.7500

Fold 3 Validation Loss: 0.2104, Accuracy: 0.9000

Fold 4 Validation Loss: 0.3553, Accuracy: 0.8000

Fold 5 Validation Loss: 0.6968, Accuracy: 0.5000

Average validation accuracy for this combination: 0.6900

Trying combination: learning_rate=0.005, dropout_rate=0.5, batch_size=16

Fold 1 Validation Loss: 0.2108, Accuracy: 0.9500

Fold 2 Validation Loss: 0.5451, Accuracy: 0.8000

Fold 3 Validation Loss: 0.6728, Accuracy: 0.6000

Fold 4 Validation Loss: 0.2086, Accuracy: 0.9000

Fold 5 Validation Loss: 0.3448, Accuracy: 0.9000

Average validation accuracy for this combination: 0.8300

Trying combination: learning_rate=0.005, dropout_rate=0.5, batch_size=32

Fold 1 Validation Loss: 0.6200, Accuracy: 0.7000

Fold 2 Validation Loss: 0.5189, Accuracy: 0.7500

Fold 3 Validation Loss: 0.5793, Accuracy: 0.6000

Fold 4 Validation Loss: 0.5987, Accuracy: 0.7000

Fold 5 Validation Loss: 0.6930, Accuracy: 0.5500

Average validation accuracy for this combination: 0.6600

Trying combination: learning_rate=0.005, dropout_rate=0.5, batch_size=64

Fold 1 Validation Loss: 0.5899, Accuracy: 0.5000

Fold 2 Validation Loss: 0.7063, Accuracy: 0.5000

Fold 3 Validation Loss: 0.6709, Accuracy: 0.5500

Fold 4 Validation Loss: 0.8764, Accuracy: 0.5000

Fold 5 Validation Loss: 0.3458, Accuracy: 0.9000

Average validation accuracy for this combination: 0.5900

Trying combination: learning_rate=0.005, dropout_rate=0.4, batch_size=8

Fold 1 Validation Loss: 0.1800, Accuracy: 0.9500

Fold 2 Validation Loss: 0.3333, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0688, Accuracy: 1.0000

Fold 4 Validation Loss: 0.3568, Accuracy: 0.9000

Fold 5 Validation Loss: 0.2946, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.005, dropout_rate=0.4, batch_size=16

Fold 1 Validation Loss: 0.2671, Accuracy: 0.9500

Fold 2 Validation Loss: 0.7526, Accuracy: 0.6500

Fold 3 Validation Loss: 0.2008, Accuracy: 0.9500

Fold 4 Validation Loss: 0.0741, Accuracy: 1.0000

Fold 5 Validation Loss: 0.3699, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9000

Trying combination: learning_rate=0.005, dropout_rate=0.4, batch_size=32

Fold 1 Validation Loss: 0.4646, Accuracy: 0.7500

Fold 2 Validation Loss: 0.6034, Accuracy: 0.8000

Fold 3 Validation Loss: 0.6442, Accuracy: 0.5500

Fold 4 Validation Loss: 0.4638, Accuracy: 0.8000

Fold 5 Validation Loss: 0.2816, Accuracy: 0.9000

Average validation accuracy for this combination: 0.7600

Trying combination: learning_rate=0.005, dropout_rate=0.4, batch_size=64

Fold 1 Validation Loss: 0.6299, Accuracy: 0.6000

Fold 2 Validation Loss: 0.5594, Accuracy: 0.6500

Fold 3 Validation Loss: 0.6849, Accuracy: 0.5000

Fold 4 Validation Loss: 0.6016, Accuracy: 0.6500

Fold 5 Validation Loss: 0.6020, Accuracy: 0.9000

Average validation accuracy for this combination: 0.6600

Trying combination: learning_rate=0.005, dropout_rate=0.3, batch_size=8

Fold 1 Validation Loss: 0.3776, Accuracy: 0.9000

Fold 2 Validation Loss: 0.2507, Accuracy: 0.8500

Fold 3 Validation Loss: 0.0791, Accuracy: 1.0000

Fold 4 Validation Loss: 0.6933, Accuracy: 0.5000

Fold 5 Validation Loss: 0.6931, Accuracy: 0.5000

Average validation accuracy for this combination: 0.7500

Trying combination: learning_rate=0.005, dropout_rate=0.3, batch_size=16

Fold 1 Validation Loss: 0.1543, Accuracy: 0.9500

Fold 2 Validation Loss: 0.6935, Accuracy: 0.5000

Fold 3 Validation Loss: 0.0846, Accuracy: 1.0000

Fold 4 Validation Loss: 0.2592, Accuracy: 0.9000

Fold 5 Validation Loss: 0.3863, Accuracy: 0.8500

Average validation accuracy for this combination: 0.8400

Trying combination: learning_rate=0.005, dropout_rate=0.3, batch_size=32

Fold 1 Validation Loss: 0.5210, Accuracy: 0.6500

Fold 2 Validation Loss: 0.5182, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1684, Accuracy: 0.9500

Fold 4 Validation Loss: 0.5488, Accuracy: 0.6500

Fold 5 Validation Loss: 0.4237, Accuracy: 0.8500

Average validation accuracy for this combination: 0.7900

Trying combination: learning_rate=0.005, dropout_rate=0.3, batch_size=64

Fold 1 Validation Loss: 0.5574, Accuracy: 0.7000

Fold 2 Validation Loss: 0.6908, Accuracy: 0.5000

Fold 3 Validation Loss: 0.6335, Accuracy: 0.7500

Fold 4 Validation Loss: 0.7319, Accuracy: 0.5000

Fold 5 Validation Loss: 0.9043, Accuracy: 0.5000

Average validation accuracy for this combination: 0.5900

Trying combination: learning_rate=0.005, dropout_rate=0.2, batch_size=8

Fold 1 Validation Loss: 0.2461, Accuracy: 0.9000

Fold 2 Validation Loss: 0.7790, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0997, Accuracy: 0.9500

Fold 4 Validation Loss: 0.8305, Accuracy: 0.7000

Fold 5 Validation Loss: 0.3752, Accuracy: 0.9000

Average validation accuracy for this combination: 0.8500

Trying combination: learning_rate=0.005, dropout_rate=0.2, batch_size=16

Fold 1 Validation Loss: 0.1273, Accuracy: 1.0000

Fold 2 Validation Loss: 0.6932, Accuracy: 0.5000

Fold 3 Validation Loss: 0.0838, Accuracy: 0.9500

Fold 4 Validation Loss: 0.7613, Accuracy: 0.7000

Fold 5 Validation Loss: 0.0718, Accuracy: 1.0000

Average validation accuracy for this combination: 0.8300

Trying combination: learning_rate=0.005, dropout_rate=0.2, batch_size=32

Fold 1 Validation Loss: 0.6625, Accuracy: 0.7500

Fold 2 Validation Loss: 0.6901, Accuracy: 0.5000

Fold 3 Validation Loss: 0.6331, Accuracy: 0.5500

Fold 4 Validation Loss: 0.4101, Accuracy: 0.7500

Fold 5 Validation Loss: 1.1056, Accuracy: 0.5500

Average validation accuracy for this combination: 0.6200

Trying combination: learning_rate=0.005, dropout_rate=0.2, batch_size=64

Fold 1 Validation Loss: 0.6930, Accuracy: 0.5000

Fold 2 Validation Loss: 0.6820, Accuracy: 0.5000

Fold 3 Validation Loss: 0.7471, Accuracy: 0.5000

Fold 4 Validation Loss: 0.7186, Accuracy: 0.5000

Fold 5 Validation Loss: 0.6366, Accuracy: 0.5000

Average validation accuracy for this combination: 0.5000

Trying combination: learning_rate=0.001, dropout_rate=0.5, batch_size=8

Fold 1 Validation Loss: 0.2492, Accuracy: 0.9500

Fold 2 Validation Loss: 0.6476, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0222, Accuracy: 1.0000

Fold 4 Validation Loss: 0.1500, Accuracy: 1.0000

Fold 5 Validation Loss: 0.4157, Accuracy: 0.8000

Average validation accuracy for this combination: 0.9100

Trying combination: learning_rate=0.001, dropout_rate=0.5, batch_size=16

Fold 1 Validation Loss: 0.3636, Accuracy: 0.9000

Fold 2 Validation Loss: 1.0173, Accuracy: 0.7500

Fold 3 Validation Loss: 0.0085, Accuracy: 1.0000

Fold 4 Validation Loss: 0.1017, Accuracy: 1.0000

Fold 5 Validation Loss: 0.1893, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.001, dropout_rate=0.5, batch_size=32

Fold 1 Validation Loss: 0.1301, Accuracy: 0.9500

Fold 2 Validation Loss: 0.4089, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1598, Accuracy: 0.9500

Fold 4 Validation Loss: 0.4491, Accuracy: 0.7500

Fold 5 Validation Loss: 0.1927, Accuracy: 0.9500

Average validation accuracy for this combination: 0.8900

Trying combination: learning_rate=0.001, dropout_rate=0.5, batch_size=64

Fold 1 Validation Loss: 0.2420, Accuracy: 0.9500

Fold 2 Validation Loss: 0.6384, Accuracy: 0.7000

Fold 3 Validation Loss: 0.6950, Accuracy: 0.5000

Fold 4 Validation Loss: 0.4636, Accuracy: 0.7000

Fold 5 Validation Loss: 0.3712, Accuracy: 0.8000

Average validation accuracy for this combination: 0.7300

Trying combination: learning_rate=0.001, dropout_rate=0.4, batch_size=8

Fold 1 Validation Loss: 0.7040, Accuracy: 0.9500

Fold 2 Validation Loss: 0.2945, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1801, Accuracy: 0.9000

Fold 4 Validation Loss: 0.7737, Accuracy: 0.7000

Fold 5 Validation Loss: 0.0048, Accuracy: 1.0000

Average validation accuracy for this combination: 0.8800

Trying combination: learning_rate=0.001, dropout_rate=0.4, batch_size=16

Fold 1 Validation Loss: 0.1935, Accuracy: 0.9500

Fold 2 Validation Loss: 0.4789, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0115, Accuracy: 1.0000

Fold 4 Validation Loss: 0.0624, Accuracy: 1.0000

Fold 5 Validation Loss: 0.0916, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9400

Trying combination: learning_rate=0.001, dropout_rate=0.4, batch_size=32

Fold 1 Validation Loss: 0.1260, Accuracy: 0.9000

Fold 2 Validation Loss: 0.4247, Accuracy: 0.8000

Fold 3 Validation Loss: 0.1363, Accuracy: 1.0000

Fold 4 Validation Loss: 0.1541, Accuracy: 0.9500

Fold 5 Validation Loss: 0.0802, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.001, dropout_rate=0.4, batch_size=64

Fold 1 Validation Loss: 0.6694, Accuracy: 0.5000

Fold 2 Validation Loss: 0.6065, Accuracy: 0.7500

Fold 3 Validation Loss: 0.4624, Accuracy: 0.7000

Fold 4 Validation Loss: 0.5400, Accuracy: 0.7000

Fold 5 Validation Loss: 0.6354, Accuracy: 0.5500

Average validation accuracy for this combination: 0.6400

Trying combination: learning_rate=0.001, dropout_rate=0.3, batch_size=8

Fold 1 Validation Loss: 0.1506, Accuracy: 0.9000

Fold 2 Validation Loss: 0.4071, Accuracy: 0.8500

Fold 3 Validation Loss: 0.0126, Accuracy: 1.0000

Fold 4 Validation Loss: 0.2622, Accuracy: 0.8000

Fold 5 Validation Loss: 0.2326, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9000

Trying combination: learning_rate=0.001, dropout_rate=0.3, batch_size=16

Fold 1 Validation Loss: 0.0940, Accuracy: 0.9500

Fold 2 Validation Loss: 0.4467, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0573, Accuracy: 1.0000

Fold 4 Validation Loss: 0.6169, Accuracy: 0.7000

Fold 5 Validation Loss: 0.2067, Accuracy: 0.9500

Average validation accuracy for this combination: 0.8800

Trying combination: learning_rate=0.001, dropout_rate=0.3, batch_size=32

Fold 1 Validation Loss: 0.1947, Accuracy: 0.9000

Fold 2 Validation Loss: 0.3531, Accuracy: 0.7500

Fold 3 Validation Loss: 0.0657, Accuracy: 0.9500

Fold 4 Validation Loss: 0.7700, Accuracy: 0.7500

Fold 5 Validation Loss: 0.4285, Accuracy: 0.7500

Average validation accuracy for this combination: 0.8200

Trying combination: learning_rate=0.001, dropout_rate=0.3, batch_size=64

Fold 1 Validation Loss: 0.4857, Accuracy: 0.9000

Fold 2 Validation Loss: 0.3491, Accuracy: 0.7500

Fold 3 Validation Loss: 0.2661, Accuracy: 0.9500

Fold 4 Validation Loss: 0.3416, Accuracy: 0.8000

Fold 5 Validation Loss: 0.5720, Accuracy: 0.9500

Average validation accuracy for this combination: 0.8700

Trying combination: learning_rate=0.001, dropout_rate=0.2, batch_size=8

Fold 1 Validation Loss: 0.2257, Accuracy: 0.9500

Fold 2 Validation Loss: 0.4170, Accuracy: 0.8500

Fold 3 Validation Loss: 0.2580, Accuracy: 0.9000

Fold 4 Validation Loss: 0.2767, Accuracy: 0.8000

Fold 5 Validation Loss: 0.2551, Accuracy: 0.9500

Average validation accuracy for this combination: 0.8900

Trying combination: learning_rate=0.001, dropout_rate=0.2, batch_size=16

Fold 1 Validation Loss: 0.1391, Accuracy: 0.9500

Fold 2 Validation Loss: 0.4031, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0866, Accuracy: 0.9500

Fold 4 Validation Loss: 0.0813, Accuracy: 1.0000

Fold 5 Validation Loss: 0.1018, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9300

Trying combination: learning_rate=0.001, dropout_rate=0.2, batch_size=32

Fold 1 Validation Loss: 0.1320, Accuracy: 1.0000

Fold 2 Validation Loss: 0.8804, Accuracy: 0.7500

Fold 3 Validation Loss: 0.0694, Accuracy: 1.0000

Fold 4 Validation Loss: 0.2088, Accuracy: 0.9500

Fold 5 Validation Loss: 0.1423, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9300

Trying combination: learning_rate=0.001, dropout_rate=0.2, batch_size=64

Fold 1 Validation Loss: 0.4694, Accuracy: 0.9000

Fold 2 Validation Loss: 0.6177, Accuracy: 0.8500

Fold 3 Validation Loss: 0.3142, Accuracy: 0.8500

Fold 4 Validation Loss: 0.5553, Accuracy: 0.6500

Fold 5 Validation Loss: 0.5301, Accuracy: 0.6500

Average validation accuracy for this combination: 0.7800

Trying combination: learning_rate=0.0005, dropout_rate=0.5, batch_size=8

Fold 1 Validation Loss: 0.0116, Accuracy: 1.0000

Fold 2 Validation Loss: 0.4948, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1870, Accuracy: 0.9500

Fold 4 Validation Loss: 0.0493, Accuracy: 1.0000

Fold 5 Validation Loss: 0.1188, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9500

Trying combination: learning_rate=0.0005, dropout_rate=0.5, batch_size=16

Fold 1 Validation Loss: 0.1153, Accuracy: 0.9500

Fold 2 Validation Loss: 0.5240, Accuracy: 0.8500

Fold 3 Validation Loss: 0.0594, Accuracy: 0.9500

Fold 4 Validation Loss: 0.1141, Accuracy: 1.0000

Fold 5 Validation Loss: 0.1897, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9400

Trying combination: learning_rate=0.0005, dropout_rate=0.5, batch_size=32

Fold 1 Validation Loss: 0.3806, Accuracy: 0.8500

Fold 2 Validation Loss: 0.3982, Accuracy: 0.8000

Fold 3 Validation Loss: 0.1880, Accuracy: 0.9500

Fold 4 Validation Loss: 0.6699, Accuracy: 0.7500

Fold 5 Validation Loss: 0.1139, Accuracy: 1.0000

Average validation accuracy for this combination: 0.8700

Trying combination: learning_rate=0.0005, dropout_rate=0.5, batch_size=64

Fold 1 Validation Loss: 0.2420, Accuracy: 1.0000

Fold 2 Validation Loss: 0.3515, Accuracy: 0.9000

Fold 3 Validation Loss: 0.2872, Accuracy: 0.9500

Fold 4 Validation Loss: 0.5344, Accuracy: 0.9000

Fold 5 Validation Loss: 0.4679, Accuracy: 0.8500

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.0005, dropout_rate=0.4, batch_size=8

Fold 1 Validation Loss: 0.1134, Accuracy: 0.9500

Fold 2 Validation Loss: 0.2888, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1011, Accuracy: 0.9500

Fold 4 Validation Loss: 0.0717, Accuracy: 1.0000

Fold 5 Validation Loss: 0.0078, Accuracy: 1.0000

Average validation accuracy for this combination: 0.9500

Trying combination: learning_rate=0.0005, dropout_rate=0.4, batch_size=16

Fold 1 Validation Loss: 0.0491, Accuracy: 1.0000

Fold 2 Validation Loss: 0.2232, Accuracy: 0.9000

Fold 3 Validation Loss: 0.0294, Accuracy: 1.0000

Fold 4 Validation Loss: 0.3158, Accuracy: 0.7500

Fold 5 Validation Loss: 0.0906, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.0005, dropout_rate=0.4, batch_size=32

Fold 1 Validation Loss: 0.1475, Accuracy: 1.0000

Fold 2 Validation Loss: 0.5391, Accuracy: 0.8000

Fold 3 Validation Loss: 0.1069, Accuracy: 1.0000

Fold 4 Validation Loss: 0.5405, Accuracy: 0.7500

Fold 5 Validation Loss: 0.2588, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9000

Trying combination: learning_rate=0.0005, dropout_rate=0.4, batch_size=64

Fold 1 Validation Loss: 0.4448, Accuracy: 0.8000

Fold 2 Validation Loss: 0.5829, Accuracy: 0.7500

Fold 3 Validation Loss: 0.4981, Accuracy: 0.8500

Fold 4 Validation Loss: 0.5121, Accuracy: 0.6000

Fold 5 Validation Loss: 0.5709, Accuracy: 0.6000

Average validation accuracy for this combination: 0.7200

Trying combination: learning_rate=0.0005, dropout_rate=0.3, batch_size=8

Fold 1 Validation Loss: 0.0591, Accuracy: 0.9500

Fold 2 Validation Loss: 0.2893, Accuracy: 0.8500

Fold 3 Validation Loss: 0.0115, Accuracy: 1.0000

Fold 4 Validation Loss: 0.1693, Accuracy: 0.9000

Fold 5 Validation Loss: 0.0196, Accuracy: 1.0000

Average validation accuracy for this combination: 0.9400

Trying combination: learning_rate=0.0005, dropout_rate=0.3, batch_size=16

Fold 1 Validation Loss: 0.0636, Accuracy: 1.0000

Fold 2 Validation Loss: 0.3038, Accuracy: 0.8500

Fold 3 Validation Loss: 0.0964, Accuracy: 1.0000

Fold 4 Validation Loss: 0.2650, Accuracy: 0.8500

Fold 5 Validation Loss: 0.0787, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9300

Trying combination: learning_rate=0.0005, dropout_rate=0.3, batch_size=32

Fold 1 Validation Loss: 0.2591, Accuracy: 0.9500

Fold 2 Validation Loss: 0.3735, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1644, Accuracy: 0.9500

Fold 4 Validation Loss: 0.3800, Accuracy: 0.8500

Fold 5 Validation Loss: 0.2366, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9100

Trying combination: learning_rate=0.0005, dropout_rate=0.3, batch_size=64

Fold 1 Validation Loss: 0.5044, Accuracy: 0.8500

Fold 2 Validation Loss: 0.5598, Accuracy: 0.6500

Fold 3 Validation Loss: 0.6290, Accuracy: 0.7000

Fold 4 Validation Loss: 0.4123, Accuracy: 0.8500

Fold 5 Validation Loss: 0.4467, Accuracy: 0.8500

Average validation accuracy for this combination: 0.7800

Trying combination: learning_rate=0.0005, dropout_rate=0.2, batch_size=8

Fold 1 Validation Loss: 0.0640, Accuracy: 0.9500

Fold 2 Validation Loss: 0.5140, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1624, Accuracy: 0.9500

Fold 4 Validation Loss: 0.4892, Accuracy: 0.7500

Fold 5 Validation Loss: 0.1256, Accuracy: 0.9500

Average validation accuracy for this combination: 0.8900

Trying combination: learning_rate=0.0005, dropout_rate=0.2, batch_size=16

Fold 1 Validation Loss: 0.0468, Accuracy: 1.0000

Fold 2 Validation Loss: 0.2813, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0264, Accuracy: 1.0000

Fold 4 Validation Loss: 0.3673, Accuracy: 0.8000

Fold 5 Validation Loss: 0.0208, Accuracy: 1.0000

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.0005, dropout_rate=0.2, batch_size=32

Fold 1 Validation Loss: 0.1961, Accuracy: 0.9000

Fold 2 Validation Loss: 0.4994, Accuracy: 0.7500

Fold 3 Validation Loss: 0.0740, Accuracy: 1.0000

Fold 4 Validation Loss: 0.1845, Accuracy: 0.9500

Fold 5 Validation Loss: 0.1236, Accuracy: 1.0000

verage validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.0005, dropout_rate=0.2, batch_size=64

Fold 1 Validation Loss: 0.2170, Accuracy: 1.0000

Fold 2 Validation Loss: 0.4802, Accuracy: 0.8000

Fold 3 Validation Loss: 0.5070, Accuracy: 0.7500

Fold 4 Validation Loss: 0.5078, Accuracy: 0.7500

Fold 5 Validation Loss: 0.2040, Accuracy: 0.9500

Average validation accuracy for this combination: 0.8500

Trying combination: learning_rate=0.0001, dropout_rate=0.5, batch_size=8

Fold 1 Validation Loss: 0.2381, Accuracy: 0.9000

Fold 2 Validation Loss: 0.3386, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1073, Accuracy: 1.0000

Fold 4 Validation Loss: 0.2002, Accuracy: 0.9500

Fold 5 Validation Loss: 0.3093, Accuracy: 0.9000

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.0001, dropout_rate=0.5, batch_size=16

Fold 1 Validation Loss: 0.2160, Accuracy: 0.9500

Fold 2 Validation Loss: 0.3933, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1326, Accuracy: 1.0000

Fold 4 Validation Loss: 0.1849, Accuracy: 0.9500

Fold 5 Validation Loss: 0.2624, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9400

Trying combination: learning_rate=0.0001, dropout_rate=0.5, batch_size=32

Fold 1 Validation Loss: 0.4849, Accuracy: 0.9500

Fold 2 Validation Loss: 0.4450, Accuracy: 0.7500

Fold 3 Validation Loss: 0.4801, Accuracy: 0.9500

Fold 4 Validation Loss: 0.4582, Accuracy: 0.8500

Fold 5 Validation Loss: 0.5325, Accuracy: 1.0000

Average validation accuracy for this combination: 0.9000

Trying combination: learning_rate=0.0001, dropout_rate=0.5, batch_size=64

Fold 1 Validation Loss: 0.6050, Accuracy: 0.7500

Fold 2 Validation Loss: 0.6165, Accuracy: 0.8500

Fold 3 Validation Loss: 0.5691, Accuracy: 0.9000

Fold 4 Validation Loss: 0.6300, Accuracy: 0.8000

Fold 5 Validation Loss: 0.5668, Accuracy: 0.9000

Average validation accuracy for this combination: 0.8400

Trying combination: learning_rate=0.0001, dropout_rate=0.4, batch_size=8

Fold 1 Validation Loss: 0.1135, Accuracy: 0.9500

Fold 2 Validation Loss: 0.4304, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0717, Accuracy: 1.0000

Fold 4 Validation Loss: 0.2129, Accuracy: 0.9500

Fold 5 Validation Loss: 0.2035, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9300

Trying combination: learning_rate=0.0001, dropout_rate=0.4, batch_size=16

Fold 1 Validation Loss: 0.1462, Accuracy: 1.0000

Fold 2 Validation Loss: 0.3423, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1530, Accuracy: 0.9500

Fold 4 Validation Loss: 0.1504, Accuracy: 1.0000

Fold 5 Validation Loss: 0.1955, Accuracy: 1.0000

Average validation accuracy for this combination: 0.9600

Trying combination: learning_rate=0.0001, dropout_rate=0.4, batch_size=32

Fold 1 Validation Loss: 0.3301, Accuracy: 0.9500

Fold 2 Validation Loss: 0.6168, Accuracy: 0.7500

Fold 3 Validation Loss: 0.4247, Accuracy: 0.9500

Fold 4 Validation Loss: 0.4625, Accuracy: 0.9000

Fold 5 Validation Loss: 0.4076, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9000

Trying combination: learning_rate=0.0001, dropout_rate=0.4, batch_size=64

Fold 1 Validation Loss: 0.5239, Accuracy: 0.9500

Fold 2 Validation Loss: 0.5598, Accuracy: 0.8000

Fold 3 Validation Loss: 0.3933, Accuracy: 1.0000

Fold 4 Validation Loss: 0.6106, Accuracy: 0.8000

Fold 5 Validation Loss: 0.5821, Accuracy: 0.9000

Average validation accuracy for this combination: 0.8900

Trying combination: learning_rate=0.0001, dropout_rate=0.3, batch_size=8

Fold 1 Validation Loss: 0.1262, Accuracy: 0.9500

Fold 2 Validation Loss: 0.5306, Accuracy: 0.8000

Fold 3 Validation Loss: 0.1387, Accuracy: 0.9500

Fold 4 Validation Loss: 0.2131, Accuracy: 0.8500

Fold 5 Validation Loss: 0.1517, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9000

Trying combination: learning_rate=0.0001, dropout_rate=0.3, batch_size=16

Fold 1 Validation Loss: 0.1543, Accuracy: 0.9500

Fold 2 Validation Loss: 0.3434, Accuracy: 0.7500

Fold 3 Validation Loss: 0.1064, Accuracy: 1.0000

Fold 4 Validation Loss: 0.1828, Accuracy: 1.0000

Fold 5 Validation Loss: 0.3490, Accuracy: 0.9000

Average validation accuracy for this combination: 0.9200

Trying combination: learning_rate=0.0001, dropout_rate=0.3, batch_size=32

Fold 1 Validation Loss: 0.4386, Accuracy: 0.9000

Fold 2 Validation Loss: 0.4060, Accuracy: 0.8000

Fold 3 Validation Loss: 0.2949, Accuracy: 0.9500

Fold 4 Validation Loss: 0.4683, Accuracy: 0.8000

Fold 5 Validation Loss: 0.4382, Accuracy: 0.9500

Average validation accuracy for this combination: 0.8800

Trying combination: learning_rate=0.0001, dropout_rate=0.3, batch_size=64

Fold 1 Validation Loss: 0.5483, Accuracy: 0.8500

Fold 2 Validation Loss: 0.5835, Accuracy: 0.7500

Fold 3 Validation Loss: 0.5551, Accuracy: 0.7500

Fold 4 Validation Loss: 0.5678, Accuracy: 0.9000

Fold 5 Validation Loss: 0.5121, Accuracy: 0.9000

Average validation accuracy for this combination: 0.8300

Trying combination: learning_rate=0.0001, dropout_rate=0.2, batch_size=8

Fold 1 Validation Loss: 0.2466, Accuracy: 0.9000

Fold 2 Validation Loss: 0.2614, Accuracy: 0.8500

Fold 3 Validation Loss: 0.1541, Accuracy: 0.9500

Fold 4 Validation Loss: 0.1752, Accuracy: 0.9500

Fold 5 Validation Loss: 0.2438, Accuracy: 0.9000

Average validation accuracy for this combination: 0.9100

Trying combination: learning_rate=0.0001, dropout_rate=0.2, batch_size=16

Fold 1 Validation Loss: 0.1343, Accuracy: 0.9500

Fold 2 Validation Loss: 0.3994, Accuracy: 0.8000

Fold 3 Validation Loss: 0.0941, Accuracy: 1.0000

Fold 4 Validation Loss: 0.2754, Accuracy: 0.8000

Fold 5 Validation Loss: 0.2161, Accuracy: 0.9500

Average validation accuracy for this combination: 0.9000

Trying combination: learning_rate=0.0001, dropout_rate=0.2, batch_size=32

Fold 1 Validation Loss: 0.4037, Accuracy: 0.8500

Fold 2 Validation Loss: 0.4234, Accuracy: 0.8000

Fold 3 Validation Loss: 0.3346, Accuracy: 0.9500

Fold 4 Validation Loss: 0.5427, Accuracy: 0.7000

Fold 5 Validation Loss: 0.4868, Accuracy: 0.9000

Average validation accuracy for this combination: 0.8400

Trying combination: learning_rate=0.0001, dropout_rate=0.2, batch_size=64

Fold 1 Validation Loss: 0.5422, Accuracy: 0.9500

Fold 2 Validation Loss: 0.5720, Accuracy: 0.8500

Fold 3 Validation Loss: 0.5668, Accuracy: 1.0000

Fold 4 Validation Loss: 0.6378, Accuracy: 0.6500

Fold 5 Validation Loss: 0.5279, Accuracy: 0.6500

Average validation accuracy for this combination: 0.8200

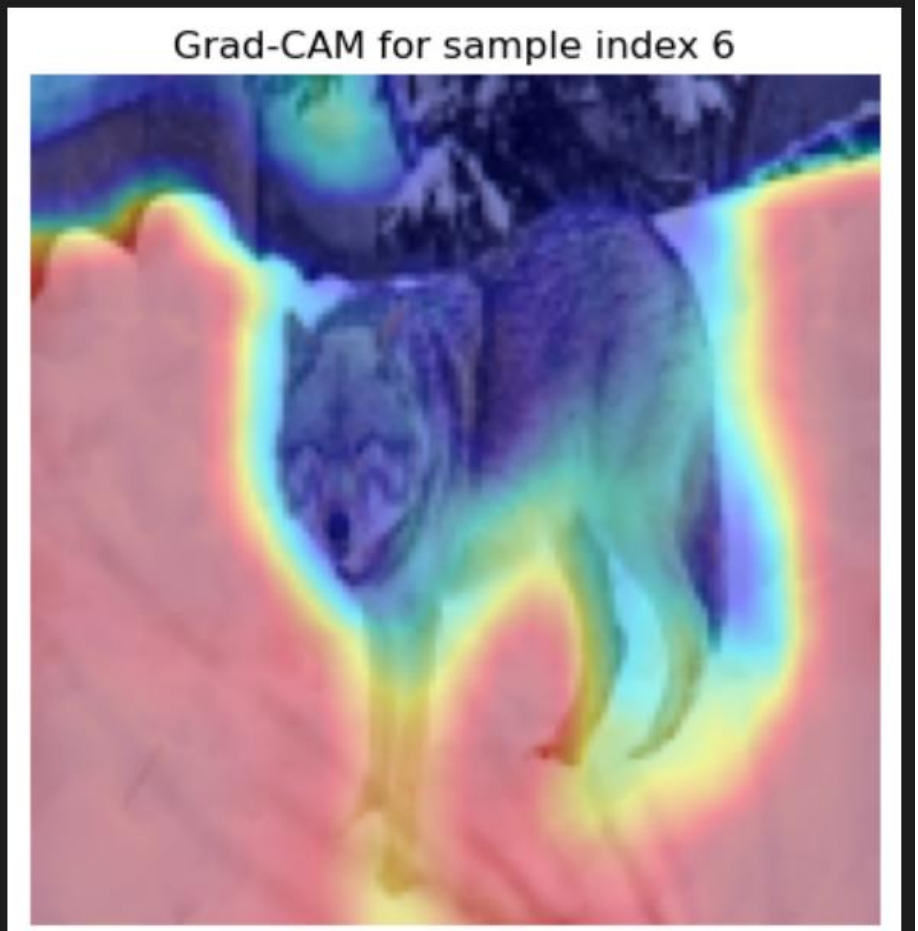
Correct Evaluation:

```
test_path = 'data/test/'  # Change this to your actual test path

# Check if the path exists and proceed with the test evaluation
if os.path.exists(test_path):
    X_test, y_test = load_and_preprocess_test(test_path)
    test_loss, test_acc = final_model.evaluate(X_test, y_test, verbose=1)
    print("Test Loss:", test_loss)
    print("Test Accuracy:", test_acc)
else:
    # Bonus 2: If no test set provided, create a clean test set of at least 10 images from the full data.
    indices_husky = [i for i, lbl in enumerate(labels_clean) if lbl == "husky"]
    indices_wolf = [i for i, lbl in enumerate(labels_clean) if lbl == "wolf"]
    selected_indices = indices_husky[:5] + indices_wolf[:5]
    X_test = X[selected_indices]
    y_test = y[selected_indices]
    test_loss, test_acc = final_model.evaluate(X_test, y_test, verbose=1)
    print("Created Clean Test Set Accuracy:", test_acc)
```

Bonus 1: Using Grad-CAM

```
Using last conv layer for Grad-CAM: conv2d_962
Grad-CAM visualization on 4 sample test images:
```



Bonus 2: Detect and Fix the data leakage

Data leakage occurs when information that should not be available during model training is inadvertently included, leading to overly optimistic performance and poor generalization to new data. In the context of the **Husky vs. Wolf classification problem**, a well-known case of data leakage is when the **background (e.g., snow vs. non-snow) is a stronger predictor than the actual animal features**.

```
Background focus ratio=0.23. Model is less likely to rely on background.
```

How the Data Leakage Occurs

In some datasets, **wolves** are often photographed in snowy backgrounds, while **huskies** are more commonly found in varied, non-snowy environments. A model trained on such a dataset may learn to classify **"snow" = wolf** and **"no snow" = husky** instead of actually distinguishing between the animals based on their physical characteristics. This is an example of **contextual data leakage** because background information is unintentionally influencing the classification.

Demonstrating the Data Leakage

1. **Train a model on the given dataset** and check its performance.
2. **Analyze which features contribute most to classification.** If the model heavily relies on background features rather than the animals themselves, it suggests data leakage.
3. **Use explainability tools like SHAP or Grad-CAM** to visualize which parts of the image the model is focusing on.

Fixing the Data Leakage

To remove the leakage and ensure the model classifies based on the animals rather than the background:

1. **Manually crop images** to focus only on the animal, removing background clues.
2. **Augment the dataset** by ensuring a balanced mix of wolves and huskies in both snowy and non-snowy environments.
3. **Apply background blurring or segmentation techniques** to neutralize background influence.
4. **Train the model with adversarial techniques** where a separate network learns to remove background influence.

By addressing the data leakage, the model will generalize better and correctly classify wolves and huskies based on their true distinguishing features rather than irrelevant background information.