baseline

April 22, 2025

[1]:
```
import os
import itertools
import random
import data_pipeline as pipeline
import pandas as pd
import numpy as np
import tensorflow as tf
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from tensorflow.keras.metrics import BinaryAccuracy, Precision, Recall
from tensorflow.keras.callbacks import EarlyStopping
```

[2]:
```
PROCESSED_DIR = "../data/processed"
IMG_HEIGHT = IMG_WIDTH = 224
NUM_CHANNELS = 3
NUM_CLASSES = 2
EPOCHS = 30
BATCH_SIZE = 128
```

[3]:
```
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
random.seed(SEED)
```

[4]:
```
metrics = [
    BinaryAccuracy(name="accuracy"),
    Precision(name="precision"),
    Recall(name="recall"),
]
```

[5]:
```
train_dir = os.path.join(PROCESSED_DIR, "train")
val_dir = os.path.join(PROCESSED_DIR, "val")
test_dir = os.path.join(PROCESSED_DIR, "test")

all_paths = pipeline.get_image_paths(PROCESSED_DIR)
train_paths = [path for path in all_paths if "/train/" in path]
mean, std = pipeline.calc_mean_std(train_paths)
```

```
# use generators from data_pipeline for training, validation, and testing
print("loading train/val/test generators from data_pipeline")
train_data_gen, val_data_gen, test_data_gen, test_data_gen_raw = pipeline.
 ↪load_data(
    train_dir, val_dir, test_dir, mean, std
)
```

```
loading train/val/test generators from data_pipeline
creating train generator
Found 1600 images belonging to 2 classes.
creating validation generator
Found 400 images belonging to 2 classes.
creating test generator (normalized)
Found 200 images belonging to 2 classes.
creating test generator (raw)
Found 200 images belonging to 2 classes.
```

[6]:
```
image_size = (IMG_HEIGHT, IMG_WIDTH)
input_shape = (IMG_HEIGHT, IMG_WIDTH, NUM_CHANNELS)
```

[7]:
```
class_names = list(train_data_gen.class_indices.keys())
print(f"class names found: {class_names}")
```

```
class names found: ['COVID', 'NORMAL']
```

[8]:
```
def build_model(input_shape):
    """
    build keras sequential model

    params
    ------
    input_shape: tuple
        shape of input images (height, width, channels)

    returns
    -------
    model: tf.keras.Model
        compiled keras model
    """
    model = tf.keras.Sequential(
        [
            tf.keras.layers.Input(shape=input_shape),
            # convolutional
            tf.keras.layers.Conv2D(32, (3, 3), activation="relu",␣
 ↪padding="same"),
            tf.keras.layers.MaxPooling2D((2, 2)),
```

```
            tf.keras.layers.Conv2D(64, (3, 3), activation="relu",␣
↪padding="same"),
            tf.keras.layers.MaxPooling2D((2, 2)),
            tf.keras.layers.Conv2D(128, (3, 3), activation="relu",␣
↪padding="same"),
            tf.keras.layers.MaxPooling2D((2, 2)),
            # fully connected
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(128, activation="relu"),
            tf.keras.layers.Dropout(0.3),  # random value
            tf.keras.layers.Dense(64, activation="relu"),
            tf.keras.layers.Dropout(0.3),  # random value
            tf.keras.layers.Dense(1, activation="sigmoid"),
        ]
    )
    return model
```

[9]:
```
def train_model(model, train_data, val_data, epochs=EPOCHS):
    # compile the model
    model.compile(
        optimizer="adam",
        loss="binary_crossentropy",
        metrics=metrics,
    )
    # train the model
    history = model.fit(train_data, epochs=epochs, validation_data=val_data,␣
↪verbose=2)
    return history
```

[10]:
```
# build model
model = build_model(input_shape)

# print model summary
print("model architecture:")
model.summary()
```

```
model architecture:
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	896
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_1 (Conv2D)	(None, 112, 112, 64)	18,496

max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
conv2d_2 (Conv2D)	(None, 56, 56, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 128)	0
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 128)	12,845,184
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8,256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

Total params: 12,946,753 (49.39 MB)

Trainable params: 12,946,753 (49.39 MB)

Non-trainable params: 0 (0.00 B)

[11]:
```
# train the model
history = train_model(model, train_data_gen, val_data_gen, EPOCHS)
```

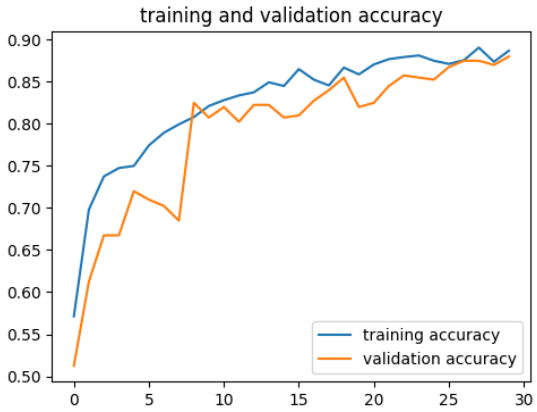
```
/opt/anaconda3/envs/ml-2025/lib/python3.12/site-
packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will be
ignored.
  self._warn_if_super_not_called()
Epoch 1/30
13/13 - 26s - 2s/step - accuracy: 0.5713 - loss: 0.8630 - precision: 0.5513 -
recall: 0.7663 - val_accuracy: 0.5125 - val_loss: 0.7130 - val_precision: 0.5148
- val_recall: 0.4350
Epoch 2/30
13/13 - 24s - 2s/step - accuracy: 0.6981 - loss: 0.5913 - precision: 0.6931 -
recall: 0.7113 - val_accuracy: 0.6125 - val_loss: 0.6641 - val_precision: 0.6166
- val_recall: 0.5950
Epoch 3/30
```

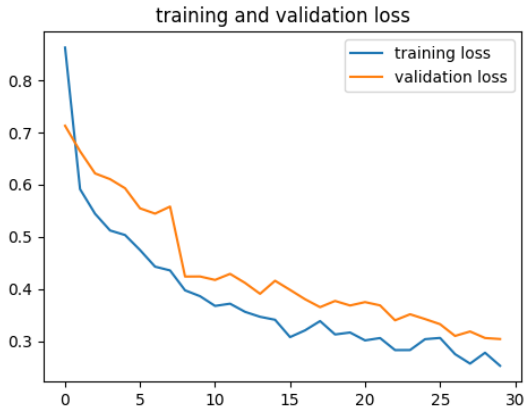
```
13/13 - 25s - 2s/step - accuracy: 0.7375 - loss: 0.5441 - precision: 0.7387 -
recall: 0.7350 - val_accuracy: 0.6675 - val_loss: 0.6216 - val_precision: 0.6959
- val_recall: 0.5950
Epoch 4/30
13/13 - 23s - 2s/step - accuracy: 0.7475 - loss: 0.5121 - precision: 0.7494 -
recall: 0.7437 - val_accuracy: 0.6675 - val_loss: 0.6106 - val_precision: 0.8602
- val_recall: 0.4000
Epoch 5/30
13/13 - 25s - 2s/step - accuracy: 0.7500 - loss: 0.5034 - precision: 0.7833 -
recall: 0.6913 - val_accuracy: 0.7200 - val_loss: 0.5931 - val_precision: 0.7444
- val_recall: 0.6700
Epoch 6/30
13/13 - 25s - 2s/step - accuracy: 0.7744 - loss: 0.4745 - precision: 0.8105 -
recall: 0.7163 - val_accuracy: 0.7100 - val_loss: 0.5545 - val_precision: 0.6707
- val_recall: 0.8250
Epoch 7/30
13/13 - 24s - 2s/step - accuracy: 0.7894 - loss: 0.4427 - precision: 0.7847 -
recall: 0.7975 - val_accuracy: 0.7025 - val_loss: 0.5444 - val_precision: 0.8785
- val_recall: 0.4700
Epoch 8/30
13/13 - 23s - 2s/step - accuracy: 0.7994 - loss: 0.4354 - precision: 0.8548 -
recall: 0.7212 - val_accuracy: 0.6850 - val_loss: 0.5579 - val_precision: 0.6194
- val_recall: 0.9600
Epoch 9/30
13/13 - 24s - 2s/step - accuracy: 0.8081 - loss: 0.3972 - precision: 0.8101 -
recall: 0.8050 - val_accuracy: 0.8250 - val_loss: 0.4237 - val_precision: 0.8218
- val_recall: 0.8300
Epoch 10/30
13/13 - 24s - 2s/step - accuracy: 0.8213 - loss: 0.3860 - precision: 0.8391 -
recall: 0.7950 - val_accuracy: 0.8075 - val_loss: 0.4238 - val_precision: 0.8060
- val_recall: 0.8100
Epoch 11/30
13/13 - 24s - 2s/step - accuracy: 0.8281 - loss: 0.3674 - precision: 0.8431 -
recall: 0.8062 - val_accuracy: 0.8200 - val_loss: 0.4172 - val_precision: 0.8441
- val_recall: 0.7850
Epoch 12/30
13/13 - 24s - 2s/step - accuracy: 0.8338 - loss: 0.3716 - precision: 0.8678 -
recall: 0.7875 - val_accuracy: 0.8025 - val_loss: 0.4288 - val_precision: 0.7951
- val_recall: 0.8150
Epoch 13/30
13/13 - 24s - 2s/step - accuracy: 0.8375 - loss: 0.3560 - precision: 0.8590 -
recall: 0.8075 - val_accuracy: 0.8225 - val_loss: 0.4115 - val_precision: 0.8274
- val_recall: 0.8150
Epoch 14/30
13/13 - 24s - 2s/step - accuracy: 0.8494 - loss: 0.3465 - precision: 0.8712 -
recall: 0.8200 - val_accuracy: 0.8225 - val_loss: 0.3906 - val_precision: 0.8116
- val_recall: 0.8400
Epoch 15/30
```

```
13/13 - 24s - 2s/step - accuracy: 0.8450 - loss: 0.3408 - precision: 0.8594 -
recall: 0.8250 - val_accuracy: 0.8075 - val_loss: 0.4157 - val_precision: 0.7595
- val_recall: 0.9000
Epoch 16/30
13/13 - 24s - 2s/step - accuracy: 0.8650 - loss: 0.3075 - precision: 0.8715 -
recall: 0.8562 - val_accuracy: 0.8100 - val_loss: 0.3981 - val_precision: 0.7818
- val_recall: 0.8600
Epoch 17/30
13/13 - 24s - 2s/step - accuracy: 0.8525 - loss: 0.3205 - precision: 0.8643 -
recall: 0.8363 - val_accuracy: 0.8275 - val_loss: 0.3802 - val_precision: 0.8047
- val_recall: 0.8650
Epoch 18/30
13/13 - 24s - 2s/step - accuracy: 0.8456 - loss: 0.3383 - precision: 0.8605 -
recall: 0.8250 - val_accuracy: 0.8400 - val_loss: 0.3650 - val_precision: 0.8505
- val_recall: 0.8250
Epoch 19/30
13/13 - 25s - 2s/step - accuracy: 0.8669 - loss: 0.3126 - precision: 0.8817 -
recall: 0.8475 - val_accuracy: 0.8550 - val_loss: 0.3769 - val_precision: 0.8698
- val_recall: 0.8350
Epoch 20/30
13/13 - 24s - 2s/step - accuracy: 0.8587 - loss: 0.3164 - precision: 0.8786 -
recall: 0.8325 - val_accuracy: 0.8200 - val_loss: 0.3682 - val_precision: 0.7689
- val_recall: 0.9150
Epoch 21/30
13/13 - 24s - 2s/step - accuracy: 0.8706 - loss: 0.3012 - precision: 0.8720 -
recall: 0.8687 - val_accuracy: 0.8250 - val_loss: 0.3747 - val_precision: 0.7686
- val_recall: 0.9300
Epoch 22/30
13/13 - 24s - 2s/step - accuracy: 0.8769 - loss: 0.3058 - precision: 0.8972 -
recall: 0.8512 - val_accuracy: 0.8450 - val_loss: 0.3684 - val_precision: 0.8108
- val_recall: 0.9000
Epoch 23/30
13/13 - 24s - 2s/step - accuracy: 0.8794 - loss: 0.2826 - precision: 0.8936 -
recall: 0.8612 - val_accuracy: 0.8575 - val_loss: 0.3397 - val_precision: 0.8421
- val_recall: 0.8800
Epoch 24/30
13/13 - 24s - 2s/step - accuracy: 0.8813 - loss: 0.2827 - precision: 0.9201 -
recall: 0.8350 - val_accuracy: 0.8550 - val_loss: 0.3513 - val_precision: 0.8198
- val_recall: 0.9100
Epoch 25/30
13/13 - 23s - 2s/step - accuracy: 0.8750 - loss: 0.3035 - precision: 0.8741 -
recall: 0.8763 - val_accuracy: 0.8525 - val_loss: 0.3422 - val_precision: 0.8895
- val_recall: 0.8050
Epoch 26/30
13/13 - 24s - 2s/step - accuracy: 0.8712 - loss: 0.3060 - precision: 0.9024 -
recall: 0.8325 - val_accuracy: 0.8675 - val_loss: 0.3323 - val_precision: 0.8419
- val_recall: 0.9050
Epoch 27/30
```

```
13/13 - 24s - 2s/step - accuracy: 0.8756 - loss: 0.2748 - precision: 0.8848 -
recall: 0.8637 - val_accuracy: 0.8750 - val_loss: 0.3098 - val_precision: 0.8788
- val_recall: 0.8700
Epoch 28/30
13/13 - 24s - 2s/step - accuracy: 0.8906 - loss: 0.2567 - precision: 0.9139 -
recall: 0.8625 - val_accuracy: 0.8750 - val_loss: 0.3183 - val_precision: 0.8947
- val_recall: 0.8500
Epoch 29/30
13/13 - 24s - 2s/step - accuracy: 0.8737 - loss: 0.2776 - precision: 0.8833 -
recall: 0.8612 - val_accuracy: 0.8700 - val_loss: 0.3058 - val_precision: 0.8558
- val_recall: 0.8900
Epoch 30/30
13/13 - 24s - 2s/step - accuracy: 0.8869 - loss: 0.2525 - precision: 0.9025 -
recall: 0.8675 - val_accuracy: 0.8800 - val_loss: 0.3038 - val_precision: 0.8689
- val_recall: 0.8950
```

[12]:
```python
# plot training history
acc = history.history["accuracy"]
val_acc = history.history["val_accuracy"]
loss = history.history["loss"]
val_loss = history.history["val_loss"]

epochs_range = range(EPOCHS)

plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label="training accuracy")
plt.plot(epochs_range, val_acc, label="validation accuracy")
plt.legend(loc="lower right")
plt.title("training and validation accuracy")

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label="training loss")
plt.plot(epochs_range, val_loss, label="validation loss")
plt.legend(loc="upper right")
plt.title("training and validation loss")
plt.show()
```





[13]:
```
# evaluate the model on the validation set after training
print("\nevaluating model on validation data after training")
results = model.evaluate(val_data_gen, verbose=1)
print(f"final validation loss: {results[0]}")
print(f"final validation accuracy: {results[1]}")
print(f"final validation precision: {results[2]}")
print(f"final validation recall: {results[3]}")
```

```
evaluating model on validation data after training
4/4             2s 341ms/step -
accuracy: 0.8700 - loss: 0.3069 - precision: 0.6495 - recall: 0.7170
final validation loss: 0.30384504795074463
final validation accuracy: 0.8799999952316284
final validation precision: 0.8689320683479309
final validation recall: 0.8949999809265137
```

1 with hyperparam tuning

[14]:
```
def build_model(input_shape, filters_conv1, units_dense1, dropout_rate):
    """
    build keras sequential model

    params
    ------
    input_shape: tuple
        shape of input images (height, width, channels)
    filters_conv1: int
        number of filters in the first convolutional layer
    units_dense1: int
        number of units in the first dense layer
    dropout_rate: float
```

```
        dropout rate for dropout layers

    returns
    -------
    model: tf.keras.Model
        keras model (not compiled)
    """
    model = tf.keras.Sequential(
        [
            tf.keras.layers.Input(shape=input_shape),
            # convolutional
            tf.keras.layers.Conv2D(
                filters_conv1, (3, 3), activation="relu", padding="same"
            ),
            tf.keras.layers.MaxPooling2D((2, 2)),
            tf.keras.layers.Conv2D(64, (3, 3), activation="relu",␣
↪padding="same"),
            tf.keras.layers.MaxPooling2D((2, 2)),
            tf.keras.layers.Conv2D(128, (3, 3), activation="relu",␣
↪padding="same"),
            tf.keras.layers.MaxPooling2D((2, 2)),
            # fully connected
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(units_dense1, activation="relu"),
            tf.keras.layers.Dropout(dropout_rate),
            tf.keras.layers.Dense(64, activation="relu"),
            tf.keras.layers.Dropout(dropout_rate),
            tf.keras.layers.Dense(1, activation="sigmoid"),
        ]
    )
    return model
```

[15]:
```
# define search space
learning_rates = [1e-3, 1e-4, 1e-5]
filters_conv1_list = [16, 32, 64]
units_dense1_list = [64, 128, 256]
dropout_rates = [0.2, 0.3, 0.4, 0.5]
```

[16]:
```
# create all possible combinations (full grid)
param_grid = list(
    itertools.product(
        learning_rates, filters_conv1_list, units_dense1_list, dropout_rates
    )
)
```

[17]:
```
# define number of combinations to randomly sample
num_combinations_to_test = 10
```

```
# randomly sample combinations
sampled_params = random.sample(param_grid, num_combinations_to_test)
print(
    f"randomly sampling {num_combinations_to_test} combinations from␣
 ↪{len(param_grid)} total."
)
```

```
randomly sampling 10 combinations from 108 total.
```

[18]:
```
# store results
results_list = []
```

[19]:
```
# early stopping callback
early_stopping = EarlyStopping(
    monitor="val_loss", patience=10, restore_best_weights=True, verbose=1
)
```

[20]:
```
print("starting hyperparameter tuning")

for lr, filters1, units1, dr in sampled_params:
    print(
        f"testing: lr={lr}, filters_conv1={filters1}, units_dense1={units1},␣
 ↪dropout_rate={dr}"
    )

    # build model
    input_shape = (IMG_HEIGHT, IMG_WIDTH, NUM_CHANNELS)
    model = build_model(
        input_shape,
        filters_conv1=filters1,
        units_dense1=units1,
        dropout_rate=dr,
    )

    # compile model
    optimizer = tf.keras.optimizers.Adam(learning_rate=lr)
    model.compile(optimizer=optimizer, loss="binary_crossentropy",␣
 ↪metrics=metrics)

    # train the model
    history = model.fit(
        train_data_gen,
        epochs=EPOCHS,
        validation_data=val_data_gen,
        callbacks=[early_stopping],
        verbose=2,
```

```
    )

    # evaluate the model on the validation set using best weights from early␣
 ↪stopping
    print("evaluating best model from this run")
    eval_results = model.evaluate(val_data_gen, verbose=1)

    run_results = {
        "learning_rate": lr,
        "filters_conv1": filters1,
        "units_dense1": units1,
        "dropout_rate": dr,
        "val_loss": eval_results[0],
        "val_accuracy": eval_results[1],
        "val_precision": eval_results[2],
        "val_recall": eval_results[3],
        "epochs_trained": len(history.epoch),
    }
    results_list.append(run_results)

print("hyperparameter tuning finished.")
```

```
starting hyperparameter tuning
testing: lr=0.001, filters_conv1=16, units_dense1=64, dropout_rate=0.2
Epoch 1/30
13/13 - 19s - 1s/step - accuracy: 0.6310 - loss: 1.1767 - precision: 0.6035 -
recall: 0.7640 - val_accuracy: 0.5625 - val_loss: 0.6716 - val_precision: 0.5415
- val_recall: 0.8150
Epoch 2/30
13/13 - 19s - 1s/step - accuracy: 0.6981 - loss: 0.5868 - precision: 0.6736 -
recall: 0.7688 - val_accuracy: 0.6475 - val_loss: 0.6733 - val_precision: 0.7521
- val_recall: 0.4400
Epoch 3/30
13/13 - 18s - 1s/step - accuracy: 0.7387 - loss: 0.5293 - precision: 0.7520 -
recall: 0.7125 - val_accuracy: 0.6175 - val_loss: 0.6433 - val_precision: 0.6009
- val_recall: 0.7000
Epoch 4/30
13/13 - 18s - 1s/step - accuracy: 0.7538 - loss: 0.5182 - precision: 0.7699 -
recall: 0.7237 - val_accuracy: 0.7000 - val_loss: 0.5993 - val_precision: 0.7410
- val_recall: 0.6150
Epoch 5/30
13/13 - 18s - 1s/step - accuracy: 0.7719 - loss: 0.4981 - precision: 0.7675 -
recall: 0.7800 - val_accuracy: 0.7050 - val_loss: 0.5779 - val_precision: 0.6934
- val_recall: 0.7350
Epoch 6/30
13/13 - 19s - 1s/step - accuracy: 0.7862 - loss: 0.4581 - precision: 0.8021 -
recall: 0.7600 - val_accuracy: 0.7625 - val_loss: 0.5366 - val_precision: 0.8261
- val_recall: 0.6650
```

```
Epoch 7/30
13/13 - 20s - 2s/step - accuracy: 0.7912 - loss: 0.4508 - precision: 0.8026 -
recall: 0.7725 - val_accuracy: 0.7350 - val_loss: 0.5502 - val_precision: 0.7260
- val_recall: 0.7550
Epoch 8/30
13/13 - 18s - 1s/step - accuracy: 0.8175 - loss: 0.4300 - precision: 0.8405 -
recall: 0.7837 - val_accuracy: 0.7800 - val_loss: 0.4850 - val_precision: 0.8043
- val_recall: 0.7400
Epoch 9/30
13/13 - 18s - 1s/step - accuracy: 0.8062 - loss: 0.4173 - precision: 0.8141 -
recall: 0.7937 - val_accuracy: 0.7875 - val_loss: 0.4724 - val_precision: 0.7602
- val_recall: 0.8400
Epoch 10/30
13/13 - 18s - 1s/step - accuracy: 0.8213 - loss: 0.3945 - precision: 0.8261 -
recall: 0.8138 - val_accuracy: 0.7875 - val_loss: 0.4706 - val_precision: 0.8324
- val_recall: 0.7200
Epoch 11/30
13/13 - 18s - 1s/step - accuracy: 0.8263 - loss: 0.3708 - precision: 0.8434 -
recall: 0.8012 - val_accuracy: 0.7900 - val_loss: 0.4509 - val_precision: 0.7990
- val_recall: 0.7750
Epoch 12/30
13/13 - 18s - 1s/step - accuracy: 0.8431 - loss: 0.3626 - precision: 0.8797 -
recall: 0.7950 - val_accuracy: 0.8225 - val_loss: 0.4198 - val_precision: 0.8377
- val_recall: 0.8000
Epoch 13/30
13/13 - 18s - 1s/step - accuracy: 0.8388 - loss: 0.3583 - precision: 0.8575 -
recall: 0.8125 - val_accuracy: 0.8075 - val_loss: 0.4663 - val_precision: 0.8555
- val_recall: 0.7400
Epoch 14/30
13/13 - 18s - 1s/step - accuracy: 0.8419 - loss: 0.3498 - precision: 0.8661 -
recall: 0.8087 - val_accuracy: 0.8325 - val_loss: 0.3922 - val_precision: 0.8519
- val_recall: 0.8050
Epoch 15/30
13/13 - 18s - 1s/step - accuracy: 0.8494 - loss: 0.3419 - precision: 0.8702 -
recall: 0.8213 - val_accuracy: 0.8125 - val_loss: 0.4541 - val_precision: 0.8205
- val_recall: 0.8000
Epoch 16/30
13/13 - 18s - 1s/step - accuracy: 0.8525 - loss: 0.3300 - precision: 0.8720 -
recall: 0.8263 - val_accuracy: 0.8475 - val_loss: 0.3921 - val_precision: 0.8390
- val_recall: 0.8600
Epoch 17/30
13/13 - 18s - 1s/step - accuracy: 0.8619 - loss: 0.3271 - precision: 0.8794 -
recall: 0.8388 - val_accuracy: 0.8200 - val_loss: 0.4081 - val_precision: 0.8556
- val_recall: 0.7700
Epoch 18/30
13/13 - 18s - 1s/step - accuracy: 0.8562 - loss: 0.3250 - precision: 0.8770 -
recall: 0.8288 - val_accuracy: 0.8125 - val_loss: 0.4170 - val_precision: 0.7682
- val_recall: 0.8950
```

```
Epoch 19/30
13/13 - 18s - 1s/step - accuracy: 0.8550 - loss: 0.3301 - precision: 0.8650 -
recall: 0.8413 - val_accuracy: 0.8375 - val_loss: 0.3919 - val_precision: 0.8358
- val_recall: 0.8400
Epoch 20/30
13/13 - 18s - 1s/step - accuracy: 0.8612 - loss: 0.3128 - precision: 0.8927 -
recall: 0.8213 - val_accuracy: 0.8450 - val_loss: 0.3698 - val_precision: 0.8382
- val_recall: 0.8550
Epoch 21/30
13/13 - 18s - 1s/step - accuracy: 0.8512 - loss: 0.3257 - precision: 0.8737 -
recall: 0.8213 - val_accuracy: 0.8550 - val_loss: 0.3548 - val_precision: 0.8586
- val_recall: 0.8500
Epoch 22/30
13/13 - 18s - 1s/step - accuracy: 0.8669 - loss: 0.2893 - precision: 0.8929 -
recall: 0.8338 - val_accuracy: 0.8200 - val_loss: 0.4220 - val_precision: 0.7667
- val_recall: 0.9200
Epoch 23/30
13/13 - 18s - 1s/step - accuracy: 0.8669 - loss: 0.2949 - precision: 0.8758 -
recall: 0.8550 - val_accuracy: 0.8450 - val_loss: 0.3405 - val_precision: 0.8594
- val_recall: 0.8250
Epoch 24/30
13/13 - 18s - 1s/step - accuracy: 0.8600 - loss: 0.2993 - precision: 0.8750 -
recall: 0.8400 - val_accuracy: 0.8125 - val_loss: 0.4078 - val_precision: 0.7572
- val_recall: 0.9200
Epoch 25/30
13/13 - 18s - 1s/step - accuracy: 0.8781 - loss: 0.2913 - precision: 0.8815 -
recall: 0.8737 - val_accuracy: 0.8575 - val_loss: 0.3297 - val_precision: 0.8488
- val_recall: 0.8700
Epoch 26/30
13/13 - 18s - 1s/step - accuracy: 0.8612 - loss: 0.3188 - precision: 0.8905 -
recall: 0.8238 - val_accuracy: 0.8575 - val_loss: 0.3486 - val_precision: 0.8705
- val_recall: 0.8400
Epoch 27/30
13/13 - 18s - 1s/step - accuracy: 0.8644 - loss: 0.2914 - precision: 0.8830 -
recall: 0.8400 - val_accuracy: 0.8375 - val_loss: 0.4050 - val_precision: 0.7824
- val_recall: 0.9350
Epoch 28/30
13/13 - 18s - 1s/step - accuracy: 0.8687 - loss: 0.2989 - precision: 0.8744 -
recall: 0.8612 - val_accuracy: 0.8650 - val_loss: 0.3160 - val_precision: 0.8510
- val_recall: 0.8850
Epoch 29/30
13/13 - 18s - 1s/step - accuracy: 0.8631 - loss: 0.2987 - precision: 0.8848 -
recall: 0.8350 - val_accuracy: 0.8625 - val_loss: 0.3144 - val_precision: 0.9143
- val_recall: 0.8000
Epoch 30/30
13/13 - 18s - 1s/step - accuracy: 0.8794 - loss: 0.2764 - precision: 0.8957 -
recall: 0.8587 - val_accuracy: 0.8600 - val_loss: 0.3355 - val_precision: 0.8243
- val_recall: 0.9150
```

```
Restoring model weights from the end of the best epoch: 29.
evaluating best model from this run
4/4             1s 255ms/step -
accuracy: 0.8833 - loss: 0.2909 - precision: 0.6965 - recall: 0.6380
testing: lr=0.0001, filters_conv1=16, units_dense1=128, dropout_rate=0.4
Epoch 1/30
13/13 - 20s - 2s/step - accuracy: 0.6610 - loss: 0.6457 - precision: 0.6458 -
recall: 0.7130 - val_accuracy: 0.6175 - val_loss: 0.6699 - val_precision: 0.7009
- val_recall: 0.4100
Epoch 2/30
13/13 - 18s - 1s/step - accuracy: 0.7025 - loss: 0.5679 - precision: 0.6942 -
recall: 0.7237 - val_accuracy: 0.6275 - val_loss: 0.6722 - val_precision: 0.6378
- val_recall: 0.5900
Epoch 3/30
13/13 - 18s - 1s/step - accuracy: 0.7212 - loss: 0.5464 - precision: 0.7287 -
recall: 0.7050 - val_accuracy: 0.6425 - val_loss: 0.6620 - val_precision: 0.6557
- val_recall: 0.6000
Epoch 4/30
13/13 - 19s - 1s/step - accuracy: 0.7425 - loss: 0.5280 - precision: 0.7474 -
recall: 0.7325 - val_accuracy: 0.6400 - val_loss: 0.6375 - val_precision: 0.6273
- val_recall: 0.6900
Epoch 5/30
13/13 - 18s - 1s/step - accuracy: 0.7506 - loss: 0.5144 - precision: 0.7670 -
recall: 0.7200 - val_accuracy: 0.6600 - val_loss: 0.6295 - val_precision: 0.6481
- val_recall: 0.7000
Epoch 6/30
13/13 - 18s - 1s/step - accuracy: 0.7563 - loss: 0.5183 - precision: 0.7778 -
recall: 0.7175 - val_accuracy: 0.6700 - val_loss: 0.6032 - val_precision: 0.6478
- val_recall: 0.7450
Epoch 7/30
13/13 - 19s - 1s/step - accuracy: 0.7819 - loss: 0.4867 - precision: 0.7902 -
recall: 0.7675 - val_accuracy: 0.7350 - val_loss: 0.5640 - val_precision: 0.7701
- val_recall: 0.6700
Epoch 8/30
13/13 - 18s - 1s/step - accuracy: 0.7669 - loss: 0.4845 - precision: 0.7921 -
recall: 0.7237 - val_accuracy: 0.7350 - val_loss: 0.5506 - val_precision: 0.7398
- val_recall: 0.7250
Epoch 9/30
13/13 - 19s - 1s/step - accuracy: 0.7869 - loss: 0.4708 - precision: 0.8097 -
recall: 0.7500 - val_accuracy: 0.7600 - val_loss: 0.5398 - val_precision: 0.8210
- val_recall: 0.6650
Epoch 10/30
13/13 - 19s - 1s/step - accuracy: 0.7887 - loss: 0.4647 - precision: 0.7954 -
recall: 0.7775 - val_accuracy: 0.7250 - val_loss: 0.5425 - val_precision: 0.7368
- val_recall: 0.7000
Epoch 11/30
13/13 - 18s - 1s/step - accuracy: 0.7825 - loss: 0.4700 - precision: 0.8229 -
recall: 0.7200 - val_accuracy: 0.7450 - val_loss: 0.5372 - val_precision: 0.7475
```

```
- val_recall: 0.7400
Epoch 12/30
13/13 - 18s - 1s/step - accuracy: 0.7869 - loss: 0.4596 - precision: 0.8048 -
recall: 0.7575 - val_accuracy: 0.7425 - val_loss: 0.5198 - val_precision: 0.7389
- val_recall: 0.7500
Epoch 13/30
13/13 - 19s - 1s/step - accuracy: 0.8150 - loss: 0.4105 - precision: 0.8281 -
recall: 0.7950 - val_accuracy: 0.7600 - val_loss: 0.4849 - val_precision: 0.7430
- val_recall: 0.7950
Epoch 14/30
13/13 - 18s - 1s/step - accuracy: 0.7931 - loss: 0.4479 - precision: 0.8182 -
recall: 0.7538 - val_accuracy: 0.7450 - val_loss: 0.5186 - val_precision: 0.7094
- val_recall: 0.8300
Epoch 15/30
13/13 - 19s - 1s/step - accuracy: 0.7994 - loss: 0.4257 - precision: 0.8259 -
recall: 0.7588 - val_accuracy: 0.7550 - val_loss: 0.5149 - val_precision: 0.7429
- val_recall: 0.7800
Epoch 16/30
13/13 - 19s - 1s/step - accuracy: 0.7987 - loss: 0.4316 - precision: 0.8161 -
recall: 0.7713 - val_accuracy: 0.7675 - val_loss: 0.5045 - val_precision: 0.7399
- val_recall: 0.8250
Epoch 17/30
13/13 - 18s - 1s/step - accuracy: 0.8075 - loss: 0.4232 - precision: 0.8306 -
recall: 0.7725 - val_accuracy: 0.7575 - val_loss: 0.4892 - val_precision: 0.7289
- val_recall: 0.8200
Epoch 18/30
13/13 - 19s - 1s/step - accuracy: 0.8044 - loss: 0.4179 - precision: 0.8277 -
recall: 0.7688 - val_accuracy: 0.7575 - val_loss: 0.4646 - val_precision: 0.7191
- val_recall: 0.8450
Epoch 19/30
13/13 - 19s - 1s/step - accuracy: 0.8156 - loss: 0.4120 - precision: 0.8176 -
recall: 0.8125 - val_accuracy: 0.7700 - val_loss: 0.4730 - val_precision: 0.7523
- val_recall: 0.8050
Epoch 20/30
13/13 - 19s - 1s/step - accuracy: 0.8181 - loss: 0.3985 - precision: 0.8284 -
recall: 0.8025 - val_accuracy: 0.7900 - val_loss: 0.4613 - val_precision: 0.7929
- val_recall: 0.7850
Epoch 21/30
13/13 - 19s - 1s/step - accuracy: 0.8231 - loss: 0.3922 - precision: 0.8615 -
recall: 0.7700 - val_accuracy: 0.8025 - val_loss: 0.4589 - val_precision: 0.7814
- val_recall: 0.8400
Epoch 22/30
13/13 - 19s - 1s/step - accuracy: 0.8481 - loss: 0.3701 - precision: 0.8521 -
recall: 0.8425 - val_accuracy: 0.7675 - val_loss: 0.4523 - val_precision: 0.7336
- val_recall: 0.8400
Epoch 23/30
13/13 - 19s - 1s/step - accuracy: 0.8338 - loss: 0.3802 - precision: 0.8363 -
recall: 0.8300 - val_accuracy: 0.7900 - val_loss: 0.4500 - val_precision: 0.7377
```

```
- val_recall: 0.9000
Epoch 24/30
13/13 - 18s - 1s/step - accuracy: 0.8313 - loss: 0.3863 - precision: 0.8433 -
recall: 0.8138 - val_accuracy: 0.8100 - val_loss: 0.4486 - val_precision: 0.8069
- val_recall: 0.8150
Epoch 25/30
13/13 - 19s - 1s/step - accuracy: 0.8175 - loss: 0.3807 - precision: 0.8442 -
recall: 0.7788 - val_accuracy: 0.7950 - val_loss: 0.4488 - val_precision: 0.7658
- val_recall: 0.8500
Epoch 26/30
13/13 - 19s - 1s/step - accuracy: 0.8325 - loss: 0.3768 - precision: 0.8402 -
recall: 0.8213 - val_accuracy: 0.7825 - val_loss: 0.4559 - val_precision: 0.7446
- val_recall: 0.8600
Epoch 27/30
13/13 - 19s - 1s/step - accuracy: 0.8338 - loss: 0.3741 - precision: 0.8504 -
recall: 0.8100 - val_accuracy: 0.8200 - val_loss: 0.4230 - val_precision: 0.8077
- val_recall: 0.8400
Epoch 28/30
13/13 - 19s - 1s/step - accuracy: 0.8450 - loss: 0.3699 - precision: 0.8781 -
recall: 0.8012 - val_accuracy: 0.7875 - val_loss: 0.4441 - val_precision: 0.7366
- val_recall: 0.8950
Epoch 29/30
13/13 - 19s - 1s/step - accuracy: 0.8313 - loss: 0.3774 - precision: 0.8240 -
recall: 0.8425 - val_accuracy: 0.8275 - val_loss: 0.4154 - val_precision: 0.8075
- val_recall: 0.8600
Epoch 30/30
13/13 - 18s - 1s/step - accuracy: 0.8425 - loss: 0.3659 - precision: 0.8764 -
recall: 0.7975 - val_accuracy: 0.8075 - val_loss: 0.4061 - val_precision: 0.7834
- val_recall: 0.8500
Restoring model weights from the end of the best epoch: 30.
evaluating best model from this run
4/4             1s 256ms/step -
accuracy: 0.8003 - loss: 0.4252 - precision: 0.5660 - recall: 0.6739
testing: lr=1e-05, filters_conv1=64, units_dense1=64, dropout_rate=0.4
Epoch 1/30
13/13 - 35s - 3s/step - accuracy: 0.6235 - loss: 0.6715 - precision: 0.5879 -
recall: 0.8260 - val_accuracy: 0.5800 - val_loss: 0.6713 - val_precision: 0.5606
- val_recall: 0.7400
Epoch 2/30
13/13 - 33s - 3s/step - accuracy: 0.6388 - loss: 0.6422 - precision: 0.6047 -
recall: 0.8012 - val_accuracy: 0.6250 - val_loss: 0.6656 - val_precision: 0.6453
- val_recall: 0.5550
Epoch 3/30
13/13 - 33s - 3s/step - accuracy: 0.6569 - loss: 0.6233 - precision: 0.6306 -
recall: 0.7575 - val_accuracy: 0.6325 - val_loss: 0.6592 - val_precision: 0.6417
- val_recall: 0.6000
Epoch 4/30
13/13 - 33s - 3s/step - accuracy: 0.6938 - loss: 0.6048 - precision: 0.6692 -
```

```
recall: 0.7663 - val_accuracy: 0.6325 - val_loss: 0.6627 - val_precision: 0.6667
- val_recall: 0.5300
Epoch 5/30
13/13 - 33s - 3s/step - accuracy: 0.6950 - loss: 0.5982 - precision: 0.6749 -
recall: 0.7525 - val_accuracy: 0.6200 - val_loss: 0.6546 - val_precision: 0.6250
- val_recall: 0.6000
Epoch 6/30
13/13 - 33s - 3s/step - accuracy: 0.7106 - loss: 0.5890 - precision: 0.7004 -
recall: 0.7362 - val_accuracy: 0.6350 - val_loss: 0.6549 - val_precision: 0.6467
- val_recall: 0.5950
Epoch 7/30
13/13 - 33s - 3s/step - accuracy: 0.6906 - loss: 0.5797 - precision: 0.6645 -
recall: 0.7700 - val_accuracy: 0.6250 - val_loss: 0.6524 - val_precision: 0.6214
- val_recall: 0.6400
Epoch 8/30
13/13 - 33s - 3s/step - accuracy: 0.7125 - loss: 0.5709 - precision: 0.6954 -
recall: 0.7563 - val_accuracy: 0.6400 - val_loss: 0.6512 - val_precision: 0.6556
- val_recall: 0.5900
Epoch 9/30
13/13 - 33s - 3s/step - accuracy: 0.7019 - loss: 0.5644 - precision: 0.6858 -
recall: 0.7450 - val_accuracy: 0.6275 - val_loss: 0.6539 - val_precision: 0.6232
- val_recall: 0.6450
Epoch 10/30
13/13 - 33s - 3s/step - accuracy: 0.7169 - loss: 0.5594 - precision: 0.7025 -
recall: 0.7525 - val_accuracy: 0.6375 - val_loss: 0.6490 - val_precision: 0.6396
- val_recall: 0.6300
Epoch 11/30
13/13 - 33s - 3s/step - accuracy: 0.7188 - loss: 0.5648 - precision: 0.7145 -
recall: 0.7287 - val_accuracy: 0.6350 - val_loss: 0.6438 - val_precision: 0.6350
- val_recall: 0.6350
Epoch 12/30
13/13 - 33s - 3s/step - accuracy: 0.7188 - loss: 0.5649 - precision: 0.7124 -
recall: 0.7337 - val_accuracy: 0.6150 - val_loss: 0.6420 - val_precision: 0.6055
- val_recall: 0.6600
Epoch 13/30
13/13 - 33s - 3s/step - accuracy: 0.7375 - loss: 0.5529 - precision: 0.7351 -
recall: 0.7425 - val_accuracy: 0.6550 - val_loss: 0.6382 - val_precision: 0.6632
- val_recall: 0.6300
Epoch 14/30
13/13 - 33s - 3s/step - accuracy: 0.7269 - loss: 0.5500 - precision: 0.7260 -
recall: 0.7287 - val_accuracy: 0.6200 - val_loss: 0.6435 - val_precision: 0.6111
- val_recall: 0.6600
Epoch 15/30
13/13 - 33s - 3s/step - accuracy: 0.7156 - loss: 0.5587 - precision: 0.6976 -
recall: 0.7613 - val_accuracy: 0.6375 - val_loss: 0.6371 - val_precision: 0.6267
- val_recall: 0.6800
Epoch 16/30
13/13 - 33s - 3s/step - accuracy: 0.7350 - loss: 0.5404 - precision: 0.7368 -
```

```
recall: 0.7312 - val_accuracy: 0.6400 - val_loss: 0.6305 - val_precision: 0.6308
- val_recall: 0.6750
Epoch 17/30
13/13 - 33s - 3s/step - accuracy: 0.7400 - loss: 0.5351 - precision: 0.7297 -
recall: 0.7625 - val_accuracy: 0.6425 - val_loss: 0.6299 - val_precision: 0.6290
- val_recall: 0.6950
Epoch 18/30
13/13 - 33s - 3s/step - accuracy: 0.7469 - loss: 0.5281 - precision: 0.7447 -
recall: 0.7513 - val_accuracy: 0.6375 - val_loss: 0.6301 - val_precision: 0.6267
- val_recall: 0.6800
Epoch 19/30
13/13 - 33s - 3s/step - accuracy: 0.7356 - loss: 0.5387 - precision: 0.7313 -
recall: 0.7450 - val_accuracy: 0.6575 - val_loss: 0.6245 - val_precision: 0.6438
- val_recall: 0.7050
Epoch 20/30
13/13 - 33s - 3s/step - accuracy: 0.7513 - loss: 0.5225 - precision: 0.7457 -
recall: 0.7625 - val_accuracy: 0.6500 - val_loss: 0.6239 - val_precision: 0.6351
- val_recall: 0.7050
Epoch 21/30
13/13 - 33s - 3s/step - accuracy: 0.7394 - loss: 0.5175 - precision: 0.7433 -
recall: 0.7312 - val_accuracy: 0.6225 - val_loss: 0.6318 - val_precision: 0.6000
- val_recall: 0.7350
Epoch 22/30
13/13 - 33s - 3s/step - accuracy: 0.7513 - loss: 0.5208 - precision: 0.7513 -
recall: 0.7513 - val_accuracy: 0.6650 - val_loss: 0.6121 - val_precision: 0.6650
- val_recall: 0.6650
Epoch 23/30
13/13 - 33s - 3s/step - accuracy: 0.7525 - loss: 0.5163 - precision: 0.7610 -
recall: 0.7362 - val_accuracy: 0.6700 - val_loss: 0.6022 - val_precision: 0.6491
- val_recall: 0.7400
Epoch 24/30
13/13 - 33s - 3s/step - accuracy: 0.7487 - loss: 0.5273 - precision: 0.7532 -
recall: 0.7400 - val_accuracy: 0.6550 - val_loss: 0.6150 - val_precision: 0.6250
- val_recall: 0.7750
Epoch 25/30
13/13 - 33s - 3s/step - accuracy: 0.7594 - loss: 0.5169 - precision: 0.7720 -
recall: 0.7362 - val_accuracy: 0.6850 - val_loss: 0.6026 - val_precision: 0.6637
- val_recall: 0.7500
Epoch 26/30
13/13 - 33s - 3s/step - accuracy: 0.7469 - loss: 0.5124 - precision: 0.7441 -
recall: 0.7525 - val_accuracy: 0.6425 - val_loss: 0.6250 - val_precision: 0.6059
- val_recall: 0.8150
Epoch 27/30
13/13 - 33s - 3s/step - accuracy: 0.7594 - loss: 0.5187 - precision: 0.7604 -
recall: 0.7575 - val_accuracy: 0.6700 - val_loss: 0.6007 - val_precision: 0.6589
- val_recall: 0.7050
Epoch 28/30
13/13 - 33s - 3s/step - accuracy: 0.7706 - loss: 0.5059 - precision: 0.7758 -
```

```
recall: 0.7613 - val_accuracy: 0.6575 - val_loss: 0.6101 - val_precision: 0.6275
- val_recall: 0.7750
Epoch 29/30
13/13 - 34s - 3s/step - accuracy: 0.7625 - loss: 0.5131 - precision: 0.7658 -
recall: 0.7563 - val_accuracy: 0.6850 - val_loss: 0.5969 - val_precision: 0.6729
- val_recall: 0.7200
Epoch 30/30
13/13 - 42s - 3s/step - accuracy: 0.7725 - loss: 0.4980 - precision: 0.7853 -
recall: 0.7500 - val_accuracy: 0.6825 - val_loss: 0.5986 - val_precision: 0.6515
- val_recall: 0.7850
Restoring model weights from the end of the best epoch: 29.
evaluating best model from this run
4/4             2s 503ms/step -
accuracy: 0.6836 - loss: 0.5952 - precision: 0.4718 - recall: 0.5719
testing: lr=0.001, filters_conv1=32, units_dense1=128, dropout_rate=0.2
Epoch 1/30
13/13 - 25s - 2s/step - accuracy: 0.5375 - loss: 1.4318 - precision: 0.5326 -
recall: 0.6130 - val_accuracy: 0.5175 - val_loss: 0.6776 - val_precision: 0.5090
- val_recall: 0.9950
Epoch 2/30
13/13 - 23s - 2s/step - accuracy: 0.6931 - loss: 0.5904 - precision: 0.6681 -
recall: 0.7675 - val_accuracy: 0.6325 - val_loss: 0.6711 - val_precision: 0.7154
- val_recall: 0.4400
Epoch 3/30
13/13 - 23s - 2s/step - accuracy: 0.7237 - loss: 0.5424 - precision: 0.7210 -
recall: 0.7300 - val_accuracy: 0.6275 - val_loss: 0.6340 - val_precision: 0.6123
- val_recall: 0.6950
Epoch 4/30
13/13 - 23s - 2s/step - accuracy: 0.7581 - loss: 0.5072 - precision: 0.7772 -
recall: 0.7237 - val_accuracy: 0.7000 - val_loss: 0.5908 - val_precision: 0.7667
- val_recall: 0.5750
Epoch 5/30
13/13 - 23s - 2s/step - accuracy: 0.7487 - loss: 0.5205 - precision: 0.7618 -
recall: 0.7237 - val_accuracy: 0.7050 - val_loss: 0.5958 - val_precision: 0.7470
- val_recall: 0.6200
Epoch 6/30
13/13 - 23s - 2s/step - accuracy: 0.7681 - loss: 0.4922 - precision: 0.7983 -
recall: 0.7175 - val_accuracy: 0.6550 - val_loss: 0.6233 - val_precision: 0.6140
- val_recall: 0.8350
Epoch 7/30
13/13 - 23s - 2s/step - accuracy: 0.7744 - loss: 0.4854 - precision: 0.7923 -
recall: 0.7437 - val_accuracy: 0.6625 - val_loss: 0.6111 - val_precision: 0.6117
- val_recall: 0.8900
Epoch 8/30
13/13 - 23s - 2s/step - accuracy: 0.7931 - loss: 0.4656 - precision: 0.8123 -
recall: 0.7625 - val_accuracy: 0.7425 - val_loss: 0.5494 - val_precision: 0.7904
- val_recall: 0.6600
Epoch 9/30
```

```
13/13 - 23s - 2s/step - accuracy: 0.7881 - loss: 0.4729 - precision: 0.8188 -
recall: 0.7400 - val_accuracy: 0.6975 - val_loss: 0.5823 - val_precision: 0.6502
- val_recall: 0.8550
Epoch 10/30
13/13 - 23s - 2s/step - accuracy: 0.7862 - loss: 0.4575 - precision: 0.8120 -
recall: 0.7450 - val_accuracy: 0.7125 - val_loss: 0.5785 - val_precision: 0.6778
- val_recall: 0.8100
Epoch 11/30
13/13 - 23s - 2s/step - accuracy: 0.8094 - loss: 0.4339 - precision: 0.8377 -
recall: 0.7675 - val_accuracy: 0.6650 - val_loss: 0.5982 - val_precision: 0.6122
- val_recall: 0.9000
Epoch 12/30
13/13 - 23s - 2s/step - accuracy: 0.8119 - loss: 0.4116 - precision: 0.8162 -
recall: 0.8050 - val_accuracy: 0.7825 - val_loss: 0.4851 - val_precision: 0.7703
- val_recall: 0.8050
Epoch 13/30
13/13 - 23s - 2s/step - accuracy: 0.8188 - loss: 0.3998 - precision: 0.8474 -
recall: 0.7775 - val_accuracy: 0.7450 - val_loss: 0.5096 - val_precision: 0.6725
- val_recall: 0.9550
Epoch 14/30
13/13 - 23s - 2s/step - accuracy: 0.8169 - loss: 0.4093 - precision: 0.8254 -
recall: 0.8037 - val_accuracy: 0.8050 - val_loss: 0.4409 - val_precision: 0.7990
- val_recall: 0.8150
Epoch 15/30
13/13 - 23s - 2s/step - accuracy: 0.8431 - loss: 0.3576 - precision: 0.8665 -
recall: 0.8112 - val_accuracy: 0.8225 - val_loss: 0.4274 - val_precision: 0.7972
- val_recall: 0.8650
Epoch 16/30
13/13 - 23s - 2s/step - accuracy: 0.8325 - loss: 0.3643 - precision: 0.8384 -
recall: 0.8238 - val_accuracy: 0.8200 - val_loss: 0.4338 - val_precision: 0.8404
- val_recall: 0.7900
Epoch 17/30
13/13 - 23s - 2s/step - accuracy: 0.8581 - loss: 0.3372 - precision: 0.8898 -
recall: 0.8175 - val_accuracy: 0.7950 - val_loss: 0.4336 - val_precision: 0.7458
- val_recall: 0.8950
Epoch 18/30
13/13 - 23s - 2s/step - accuracy: 0.8594 - loss: 0.3181 - precision: 0.8635 -
recall: 0.8537 - val_accuracy: 0.8100 - val_loss: 0.4229 - val_precision: 0.7541
- val_recall: 0.9200
Epoch 19/30
13/13 - 24s - 2s/step - accuracy: 0.8456 - loss: 0.3488 - precision: 0.8522 -
recall: 0.8363 - val_accuracy: 0.8375 - val_loss: 0.3977 - val_precision: 0.8325
- val_recall: 0.8450
Epoch 20/30
13/13 - 23s - 2s/step - accuracy: 0.8500 - loss: 0.3210 - precision: 0.8836 -
recall: 0.8062 - val_accuracy: 0.8225 - val_loss: 0.3834 - val_precision: 0.8028
- val_recall: 0.8550
Epoch 21/30
```

```
13/13 - 23s - 2s/step - accuracy: 0.8600 - loss: 0.3117 - precision: 0.8600 -
recall: 0.8600 - val_accuracy: 0.8450 - val_loss: 0.3653 - val_precision: 0.8416
- val_recall: 0.8500
Epoch 22/30
13/13 - 23s - 2s/step - accuracy: 0.8544 - loss: 0.3246 - precision: 0.8765 -
recall: 0.8250 - val_accuracy: 0.8450 - val_loss: 0.3762 - val_precision: 0.8136
- val_recall: 0.8950
Epoch 23/30
13/13 - 23s - 2s/step - accuracy: 0.8806 - loss: 0.3004 - precision: 0.8960 -
recall: 0.8612 - val_accuracy: 0.8250 - val_loss: 0.3978 - val_precision: 0.7928
- val_recall: 0.8800
Epoch 24/30
13/13 - 23s - 2s/step - accuracy: 0.8650 - loss: 0.3131 - precision: 0.8946 -
recall: 0.8275 - val_accuracy: 0.8375 - val_loss: 0.3962 - val_precision: 0.8169
- val_recall: 0.8700
Epoch 25/30
13/13 - 23s - 2s/step - accuracy: 0.8763 - loss: 0.2868 - precision: 0.8849 -
recall: 0.8650 - val_accuracy: 0.8325 - val_loss: 0.3581 - val_precision: 0.7982
- val_recall: 0.8900
Epoch 26/30
13/13 - 23s - 2s/step - accuracy: 0.8625 - loss: 0.3059 - precision: 0.8671 -
recall: 0.8562 - val_accuracy: 0.8050 - val_loss: 0.4123 - val_precision: 0.7798
- val_recall: 0.8500
Epoch 27/30
13/13 - 23s - 2s/step - accuracy: 0.8731 - loss: 0.3016 - precision: 0.8996 -
recall: 0.8400 - val_accuracy: 0.8625 - val_loss: 0.3352 - val_precision: 0.8718
- val_recall: 0.8500
Epoch 28/30
13/13 - 23s - 2s/step - accuracy: 0.8681 - loss: 0.3065 - precision: 0.8695 -
recall: 0.8662 - val_accuracy: 0.8275 - val_loss: 0.3748 - val_precision: 0.7741
- val_recall: 0.9250
Epoch 29/30
13/13 - 23s - 2s/step - accuracy: 0.8712 - loss: 0.2923 - precision: 0.8960 -
recall: 0.8400 - val_accuracy: 0.8600 - val_loss: 0.3546 - val_precision: 0.8711
- val_recall: 0.8450
Epoch 30/30
13/13 - 23s - 2s/step - accuracy: 0.8750 - loss: 0.2880 - precision: 0.9011 -
recall: 0.8425 - val_accuracy: 0.8375 - val_loss: 0.3461 - val_precision: 0.8000
- val_recall: 0.9000
Restoring model weights from the end of the best epoch: 27.
evaluating best model from this run
4/4             2s 334ms/step -
accuracy: 0.8624 - loss: 0.3311 - precision: 0.6508 - recall: 0.6749
testing: lr=1e-05, filters_conv1=16, units_dense1=256, dropout_rate=0.3
Epoch 1/30
13/13 - 20s - 2s/step - accuracy: 0.6515 - loss: 0.6409 - precision: 0.6168 -
recall: 0.8000 - val_accuracy: 0.5775 - val_loss: 0.6814 - val_precision: 0.6281
- val_recall: 0.3800
```

```
Epoch 2/30
13/13 - 18s - 1s/step - accuracy: 0.7156 - loss: 0.5739 - precision: 0.7138 -
recall: 0.7200 - val_accuracy: 0.6100 - val_loss: 0.6777 - val_precision: 0.6294
- val_recall: 0.5350
Epoch 3/30
13/13 - 18s - 1s/step - accuracy: 0.7138 - loss: 0.5680 - precision: 0.7209 -
recall: 0.6975 - val_accuracy: 0.6300 - val_loss: 0.6674 - val_precision: 0.6444
- val_recall: 0.5800
Epoch 4/30
13/13 - 19s - 1s/step - accuracy: 0.7212 - loss: 0.5587 - precision: 0.7117 -
recall: 0.7437 - val_accuracy: 0.6275 - val_loss: 0.6569 - val_precision: 0.6564
- val_recall: 0.5350
Epoch 5/30
13/13 - 18s - 1s/step - accuracy: 0.7331 - loss: 0.5449 - precision: 0.7464 -
recall: 0.7063 - val_accuracy: 0.6475 - val_loss: 0.6463 - val_precision: 0.6578
- val_recall: 0.6150
Epoch 6/30
13/13 - 19s - 1s/step - accuracy: 0.7600 - loss: 0.5293 - precision: 0.7730 -
recall: 0.7362 - val_accuracy: 0.6500 - val_loss: 0.6433 - val_precision: 0.6531
- val_recall: 0.6400
Epoch 7/30
13/13 - 19s - 1s/step - accuracy: 0.7550 - loss: 0.5138 - precision: 0.7649 -
recall: 0.7362 - val_accuracy: 0.6550 - val_loss: 0.6333 - val_precision: 0.6566
- val_recall: 0.6500
Epoch 8/30
13/13 - 19s - 1s/step - accuracy: 0.7519 - loss: 0.5256 - precision: 0.7607 -
recall: 0.7350 - val_accuracy: 0.6600 - val_loss: 0.6206 - val_precision: 0.6739
- val_recall: 0.6200
Epoch 9/30
13/13 - 19s - 1s/step - accuracy: 0.7437 - loss: 0.5312 - precision: 0.7384 -
recall: 0.7550 - val_accuracy: 0.6700 - val_loss: 0.6103 - val_precision: 0.6635
- val_recall: 0.6900
Epoch 10/30
13/13 - 18s - 1s/step - accuracy: 0.7675 - loss: 0.5094 - precision: 0.7892 -
recall: 0.7300 - val_accuracy: 0.6650 - val_loss: 0.6073 - val_precision: 0.6447
- val_recall: 0.7350
Epoch 11/30
13/13 - 19s - 1s/step - accuracy: 0.7694 - loss: 0.4892 - precision: 0.7657 -
recall: 0.7763 - val_accuracy: 0.6900 - val_loss: 0.5985 - val_precision: 0.7209
- val_recall: 0.6200
Epoch 12/30
13/13 - 18s - 1s/step - accuracy: 0.7756 - loss: 0.4899 - precision: 0.7860 -
recall: 0.7575 - val_accuracy: 0.6725 - val_loss: 0.5954 - val_precision: 0.6751
- val_recall: 0.6650
Epoch 13/30
13/13 - 18s - 1s/step - accuracy: 0.7756 - loss: 0.4860 - precision: 0.7831 -
recall: 0.7625 - val_accuracy: 0.7050 - val_loss: 0.5851 - val_precision: 0.7253
- val_recall: 0.6600
```

```
Epoch 14/30
13/13 - 19s - 1s/step - accuracy: 0.7875 - loss: 0.4686 - precision: 0.8018 -
recall: 0.7638 - val_accuracy: 0.6750 - val_loss: 0.5916 - val_precision: 0.6535
- val_recall: 0.7450
Epoch 15/30
13/13 - 18s - 1s/step - accuracy: 0.7812 - loss: 0.4796 - precision: 0.7799 -
recall: 0.7837 - val_accuracy: 0.7000 - val_loss: 0.5692 - val_precision: 0.6942
- val_recall: 0.7150
Epoch 16/30
13/13 - 18s - 1s/step - accuracy: 0.7887 - loss: 0.4766 - precision: 0.8164 -
recall: 0.7450 - val_accuracy: 0.7325 - val_loss: 0.5479 - val_precision: 0.7183
- val_recall: 0.7650
Epoch 17/30
13/13 - 19s - 1s/step - accuracy: 0.7837 - loss: 0.4783 - precision: 0.7933 -
recall: 0.7675 - val_accuracy: 0.7375 - val_loss: 0.5442 - val_precision: 0.7411
- val_recall: 0.7300
Epoch 18/30
13/13 - 18s - 1s/step - accuracy: 0.7850 - loss: 0.4730 - precision: 0.8065 -
recall: 0.7500 - val_accuracy: 0.7575 - val_loss: 0.5324 - val_precision: 0.7441
- val_recall: 0.7850
Epoch 19/30
13/13 - 18s - 1s/step - accuracy: 0.7944 - loss: 0.4529 - precision: 0.8213 -
recall: 0.7525 - val_accuracy: 0.7125 - val_loss: 0.5574 - val_precision: 0.6707
- val_recall: 0.8350
Epoch 20/30
13/13 - 18s - 1s/step - accuracy: 0.8006 - loss: 0.4464 - precision: 0.8254 -
recall: 0.7625 - val_accuracy: 0.7525 - val_loss: 0.5246 - val_precision: 0.7393
- val_recall: 0.7800
Epoch 21/30
13/13 - 18s - 1s/step - accuracy: 0.7837 - loss: 0.4509 - precision: 0.7873 -
recall: 0.7775 - val_accuracy: 0.7550 - val_loss: 0.5169 - val_precision: 0.7361
- val_recall: 0.7950
Epoch 22/30
13/13 - 18s - 1s/step - accuracy: 0.7975 - loss: 0.4340 - precision: 0.8083 -
recall: 0.7800 - val_accuracy: 0.7300 - val_loss: 0.5368 - val_precision: 0.6797
- val_recall: 0.8700
Epoch 23/30
13/13 - 18s - 1s/step - accuracy: 0.7912 - loss: 0.4446 - precision: 0.8026 -
recall: 0.7725 - val_accuracy: 0.7700 - val_loss: 0.5142 - val_precision: 0.7596
- val_recall: 0.7900
Epoch 24/30
13/13 - 18s - 1s/step - accuracy: 0.8081 - loss: 0.4359 - precision: 0.8205 -
recall: 0.7887 - val_accuracy: 0.7700 - val_loss: 0.5087 - val_precision: 0.7596
- val_recall: 0.7900
Epoch 25/30
13/13 - 19s - 1s/step - accuracy: 0.8012 - loss: 0.4344 - precision: 0.8213 -
recall: 0.7700 - val_accuracy: 0.7575 - val_loss: 0.5091 - val_precision: 0.7119
- val_recall: 0.8650
```

```
Epoch 26/30
13/13 - 18s - 1s/step - accuracy: 0.8156 - loss: 0.4280 - precision: 0.8389 -
recall: 0.7812 - val_accuracy: 0.7725 - val_loss: 0.4912 - val_precision: 0.7422
- val_recall: 0.8350
Epoch 27/30
13/13 - 18s - 1s/step - accuracy: 0.8075 - loss: 0.4313 - precision: 0.8195 -
recall: 0.7887 - val_accuracy: 0.7750 - val_loss: 0.4872 - val_precision: 0.7434
- val_recall: 0.8400
Epoch 28/30
13/13 - 19s - 1s/step - accuracy: 0.8269 - loss: 0.4121 - precision: 0.8392 -
recall: 0.8087 - val_accuracy: 0.7600 - val_loss: 0.4938 - val_precision: 0.7149
- val_recall: 0.8650
Epoch 29/30
13/13 - 18s - 1s/step - accuracy: 0.8169 - loss: 0.4116 - precision: 0.8229 -
recall: 0.8075 - val_accuracy: 0.7875 - val_loss: 0.4790 - val_precision: 0.7833
- val_recall: 0.7950
Epoch 30/30
13/13 - 19s - 1s/step - accuracy: 0.8194 - loss: 0.4142 - precision: 0.8288 -
recall: 0.8050 - val_accuracy: 0.7950 - val_loss: 0.4681 - val_precision: 0.7783
- val_recall: 0.8250
Restoring model weights from the end of the best epoch: 30.
evaluating best model from this run
4/4             1s 267ms/step -
accuracy: 0.7956 - loss: 0.4687 - precision: 0.5619 - recall: 0.6559
testing: lr=0.001, filters_conv1=64, units_dense1=256, dropout_rate=0.3
Epoch 1/30
13/13 - 38s - 3s/step - accuracy: 0.6330 - loss: 1.1855 - precision: 0.6155 -
recall: 0.7090 - val_accuracy: 0.5850 - val_loss: 0.7027 - val_precision: 0.6037
- val_recall: 0.4950
Epoch 2/30
13/13 - 36s - 3s/step - accuracy: 0.7194 - loss: 0.5693 - precision: 0.7153 -
recall: 0.7287 - val_accuracy: 0.6400 - val_loss: 0.6573 - val_precision: 0.6346
- val_recall: 0.6600
Epoch 3/30
13/13 - 36s - 3s/step - accuracy: 0.7569 - loss: 0.5241 - precision: 0.7652 -
recall: 0.7412 - val_accuracy: 0.5975 - val_loss: 0.6680 - val_precision: 0.5670
- val_recall: 0.8250
Epoch 4/30
13/13 - 36s - 3s/step - accuracy: 0.7456 - loss: 0.5278 - precision: 0.7293 -
recall: 0.7812 - val_accuracy: 0.6800 - val_loss: 0.6032 - val_precision: 0.6622
- val_recall: 0.7350
Epoch 5/30
13/13 - 36s - 3s/step - accuracy: 0.7738 - loss: 0.4963 - precision: 0.7724 -
recall: 0.7763 - val_accuracy: 0.7325 - val_loss: 0.5623 - val_precision: 0.8298
- val_recall: 0.5850
Epoch 6/30
13/13 - 35s - 3s/step - accuracy: 0.7812 - loss: 0.4854 - precision: 0.8082 -
recall: 0.7375 - val_accuracy: 0.7300 - val_loss: 0.5644 - val_precision: 0.7233
```

```
- val_recall: 0.7450
Epoch 7/30
13/13 - 36s - 3s/step - accuracy: 0.8106 - loss: 0.4259 - precision: 0.8300 -
recall: 0.7812 - val_accuracy: 0.7725 - val_loss: 0.4754 - val_precision: 0.7739
- val_recall: 0.7700
Epoch 8/30
13/13 - 36s - 3s/step - accuracy: 0.8231 - loss: 0.3854 - precision: 0.8489 -
recall: 0.7862 - val_accuracy: 0.8050 - val_loss: 0.4352 - val_precision: 0.8352
- val_recall: 0.7600
Epoch 9/30
13/13 - 36s - 3s/step - accuracy: 0.8356 - loss: 0.3601 - precision: 0.8585 -
recall: 0.8037 - val_accuracy: 0.8075 - val_loss: 0.3955 - val_precision: 0.8398
- val_recall: 0.7600
Epoch 10/30
13/13 - 36s - 3s/step - accuracy: 0.8250 - loss: 0.3894 - precision: 0.8448 -
recall: 0.7962 - val_accuracy: 0.7775 - val_loss: 0.4646 - val_precision: 0.7362
- val_recall: 0.8650
Epoch 11/30
13/13 - 36s - 3s/step - accuracy: 0.8400 - loss: 0.3356 - precision: 0.8598 -
recall: 0.8125 - val_accuracy: 0.8150 - val_loss: 0.4028 - val_precision: 0.8119
- val_recall: 0.8200
Epoch 12/30
13/13 - 36s - 3s/step - accuracy: 0.8475 - loss: 0.3373 - precision: 0.8677 -
recall: 0.8200 - val_accuracy: 0.8000 - val_loss: 0.4304 - val_precision: 0.7941
- val_recall: 0.8100
Epoch 13/30
13/13 - 36s - 3s/step - accuracy: 0.8612 - loss: 0.3272 - precision: 0.8813 -
recall: 0.8350 - val_accuracy: 0.8575 - val_loss: 0.3661 - val_precision: 0.8667
- val_recall: 0.8450
Epoch 14/30
13/13 - 36s - 3s/step - accuracy: 0.8475 - loss: 0.3409 - precision: 0.8727 -
recall: 0.8138 - val_accuracy: 0.7675 - val_loss: 0.4540 - val_precision: 0.7166
- val_recall: 0.8850
Epoch 15/30
13/13 - 36s - 3s/step - accuracy: 0.8456 - loss: 0.3487 - precision: 0.8478 -
recall: 0.8425 - val_accuracy: 0.8375 - val_loss: 0.4034 - val_precision: 0.9245
- val_recall: 0.7350
Epoch 16/30
13/13 - 36s - 3s/step - accuracy: 0.8581 - loss: 0.3244 - precision: 0.8755 -
recall: 0.8350 - val_accuracy: 0.8425 - val_loss: 0.3683 - val_precision: 0.8743
- val_recall: 0.8000
Epoch 17/30
13/13 - 36s - 3s/step - accuracy: 0.8625 - loss: 0.2967 - precision: 0.8856 -
recall: 0.8325 - val_accuracy: 0.8400 - val_loss: 0.3641 - val_precision: 0.8333
- val_recall: 0.8500
Epoch 18/30
13/13 - 36s - 3s/step - accuracy: 0.8712 - loss: 0.2820 - precision: 0.8817 -
recall: 0.8575 - val_accuracy: 0.8625 - val_loss: 0.3377 - val_precision: 0.8756
```

```
- val_recall: 0.8450
Epoch 19/30
13/13 - 36s - 3s/step - accuracy: 0.8694 - loss: 0.2776 - precision: 0.8813 -
recall: 0.8537 - val_accuracy: 0.8675 - val_loss: 0.3311 - val_precision: 0.8657
- val_recall: 0.8700
Epoch 20/30
13/13 - 35s - 3s/step - accuracy: 0.8763 - loss: 0.3036 - precision: 0.8810 -
recall: 0.8700 - val_accuracy: 0.8650 - val_loss: 0.3499 - val_precision: 0.9101
- val_recall: 0.8100
Epoch 21/30
13/13 - 36s - 3s/step - accuracy: 0.8881 - loss: 0.2683 - precision: 0.9113 -
recall: 0.8600 - val_accuracy: 0.8650 - val_loss: 0.3321 - val_precision: 0.8443
- val_recall: 0.8950
Epoch 22/30
13/13 - 36s - 3s/step - accuracy: 0.8881 - loss: 0.2643 - precision: 0.9069 -
recall: 0.8650 - val_accuracy: 0.8250 - val_loss: 0.3717 - val_precision: 0.7778
- val_recall: 0.9100
Epoch 23/30
13/13 - 36s - 3s/step - accuracy: 0.8913 - loss: 0.2760 - precision: 0.8982 -
recall: 0.8825 - val_accuracy: 0.8425 - val_loss: 0.3522 - val_precision: 0.8044
- val_recall: 0.9050
Epoch 24/30
13/13 - 36s - 3s/step - accuracy: 0.8869 - loss: 0.2620 - precision: 0.8994 -
recall: 0.8712 - val_accuracy: 0.8575 - val_loss: 0.3377 - val_precision: 0.8389
- val_recall: 0.8850
Epoch 25/30
13/13 - 37s - 3s/step - accuracy: 0.8919 - loss: 0.2604 - precision: 0.8994 -
recall: 0.8825 - val_accuracy: 0.8775 - val_loss: 0.3062 - val_precision: 0.9266
- val_recall: 0.8200
Epoch 26/30
13/13 - 36s - 3s/step - accuracy: 0.9038 - loss: 0.2270 - precision: 0.9152 -
recall: 0.8900 - val_accuracy: 0.8400 - val_loss: 0.3565 - val_precision: 0.7957
- val_recall: 0.9150
Epoch 27/30
13/13 - 36s - 3s/step - accuracy: 0.8975 - loss: 0.2514 - precision: 0.8936 -
recall: 0.9025 - val_accuracy: 0.8775 - val_loss: 0.3076 - val_precision: 0.8832
- val_recall: 0.8700
Epoch 28/30
13/13 - 36s - 3s/step - accuracy: 0.9119 - loss: 0.2243 - precision: 0.9176 -
recall: 0.9050 - val_accuracy: 0.8550 - val_loss: 0.3186 - val_precision: 0.8227
- val_recall: 0.9050
Epoch 29/30
13/13 - 36s - 3s/step - accuracy: 0.8956 - loss: 0.2549 - precision: 0.9001 -
recall: 0.8900 - val_accuracy: 0.8725 - val_loss: 0.2982 - val_precision: 0.9116
- val_recall: 0.8250
Epoch 30/30
13/13 - 36s - 3s/step - accuracy: 0.8875 - loss: 0.2560 - precision: 0.9036 -
recall: 0.8675 - val_accuracy: 0.8275 - val_loss: 0.4164 - val_precision: 0.7610
```

```
- val_recall: 0.9550
Restoring model weights from the end of the best epoch: 29.
evaluating best model from this run
4/4             2s 504ms/step -
accuracy: 0.8912 - loss: 0.2638 - precision: 0.6940 - recall: 0.6595
testing: lr=0.001, filters_conv1=32, units_dense1=256, dropout_rate=0.2
Epoch 1/30
13/13 - 25s - 2s/step - accuracy: 0.5985 - loss: 1.2758 - precision: 0.5763 -
recall: 0.7440 - val_accuracy: 0.5300 - val_loss: 0.6839 - val_precision: 0.5161
- val_recall: 0.9600
Epoch 2/30
13/13 - 24s - 2s/step - accuracy: 0.6862 - loss: 0.5838 - precision: 0.6737 -
recall: 0.7225 - val_accuracy: 0.6550 - val_loss: 0.6651 - val_precision: 0.6962
- val_recall: 0.5500
Epoch 3/30
13/13 - 24s - 2s/step - accuracy: 0.7194 - loss: 0.5476 - precision: 0.7097 -
recall: 0.7425 - val_accuracy: 0.6000 - val_loss: 0.6560 - val_precision: 0.5725
- val_recall: 0.7900
Epoch 4/30
13/13 - 24s - 2s/step - accuracy: 0.7513 - loss: 0.5215 - precision: 0.7597 -
recall: 0.7350 - val_accuracy: 0.6600 - val_loss: 0.6190 - val_precision: 0.6798
- val_recall: 0.6050
Epoch 5/30
13/13 - 23s - 2s/step - accuracy: 0.7738 - loss: 0.4983 - precision: 0.8050 -
recall: 0.7225 - val_accuracy: 0.7000 - val_loss: 0.5831 - val_precision: 0.7062
- val_recall: 0.6850
Epoch 6/30
13/13 - 24s - 2s/step - accuracy: 0.7912 - loss: 0.4618 - precision: 0.8123 -
recall: 0.7575 - val_accuracy: 0.6825 - val_loss: 0.5846 - val_precision: 0.6367
- val_recall: 0.8500
Epoch 7/30
13/13 - 24s - 2s/step - accuracy: 0.7925 - loss: 0.4619 - precision: 0.8015 -
recall: 0.7775 - val_accuracy: 0.7625 - val_loss: 0.5024 - val_precision: 0.7561
- val_recall: 0.7750
Epoch 8/30
13/13 - 24s - 2s/step - accuracy: 0.8000 - loss: 0.4352 - precision: 0.8085 -
recall: 0.7862 - val_accuracy: 0.7725 - val_loss: 0.5127 - val_precision: 0.8150
- val_recall: 0.7050
Epoch 9/30
13/13 - 24s - 2s/step - accuracy: 0.8037 - loss: 0.4318 - precision: 0.8172 -
recall: 0.7825 - val_accuracy: 0.6550 - val_loss: 0.5966 - val_precision: 0.6076
- val_recall: 0.8750
Epoch 10/30
13/13 - 23s - 2s/step - accuracy: 0.8188 - loss: 0.4089 - precision: 0.8303 -
recall: 0.8012 - val_accuracy: 0.7475 - val_loss: 0.4994 - val_precision: 0.6774
- val_recall: 0.9450
Epoch 11/30
13/13 - 23s - 2s/step - accuracy: 0.8256 - loss: 0.3944 - precision: 0.8361 -
```

```
recall: 0.8100 - val_accuracy: 0.7950 - val_loss: 0.4577 - val_precision: 0.7731
- val_recall: 0.8350
Epoch 12/30
13/13 - 24s - 2s/step - accuracy: 0.8350 - loss: 0.3660 - precision: 0.8517 -
recall: 0.8112 - val_accuracy: 0.7700 - val_loss: 0.4548 - val_precision: 0.7288
- val_recall: 0.8600
Epoch 13/30
13/13 - 24s - 2s/step - accuracy: 0.8375 - loss: 0.3534 - precision: 0.8600 -
recall: 0.8062 - val_accuracy: 0.8150 - val_loss: 0.4118 - val_precision: 0.8351
- val_recall: 0.7850
Epoch 14/30
13/13 - 23s - 2s/step - accuracy: 0.8319 - loss: 0.3693 - precision: 0.8306 -
recall: 0.8338 - val_accuracy: 0.8075 - val_loss: 0.4096 - val_precision: 0.7531
- val_recall: 0.9150
Epoch 15/30
13/13 - 23s - 2s/step - accuracy: 0.8625 - loss: 0.3243 - precision: 0.8856 -
recall: 0.8325 - val_accuracy: 0.7900 - val_loss: 0.4097 - val_precision: 0.7589
- val_recall: 0.8500
Epoch 16/30
13/13 - 24s - 2s/step - accuracy: 0.8631 - loss: 0.3193 - precision: 0.8701 -
recall: 0.8537 - val_accuracy: 0.8275 - val_loss: 0.3916 - val_precision: 0.8164
- val_recall: 0.8450
Epoch 17/30
13/13 - 23s - 2s/step - accuracy: 0.8600 - loss: 0.3044 - precision: 0.8861 -
recall: 0.8263 - val_accuracy: 0.8000 - val_loss: 0.4044 - val_precision: 0.7459
- val_recall: 0.9100
Epoch 18/30
13/13 - 24s - 2s/step - accuracy: 0.8637 - loss: 0.3054 - precision: 0.8601 -
recall: 0.8687 - val_accuracy: 0.8425 - val_loss: 0.3647 - val_precision: 0.8703
- val_recall: 0.8050
Epoch 19/30
13/13 - 24s - 2s/step - accuracy: 0.8575 - loss: 0.3207 - precision: 0.8886 -
recall: 0.8175 - val_accuracy: 0.8400 - val_loss: 0.3894 - val_precision: 0.7881
- val_recall: 0.9300
Epoch 20/30
13/13 - 24s - 2s/step - accuracy: 0.8725 - loss: 0.2985 - precision: 0.8725 -
recall: 0.8725 - val_accuracy: 0.8475 - val_loss: 0.3562 - val_precision: 0.8458
- val_recall: 0.8500
Epoch 21/30
13/13 - 23s - 2s/step - accuracy: 0.8838 - loss: 0.2770 - precision: 0.8977 -
recall: 0.8662 - val_accuracy: 0.8425 - val_loss: 0.3793 - val_precision: 0.8216
- val_recall: 0.8750
Epoch 22/30
13/13 - 24s - 2s/step - accuracy: 0.8838 - loss: 0.2773 - precision: 0.9018 -
recall: 0.8612 - val_accuracy: 0.8500 - val_loss: 0.3628 - val_precision: 0.8070
- val_recall: 0.9200
Epoch 23/30
13/13 - 25s - 2s/step - accuracy: 0.8800 - loss: 0.2773 - precision: 0.8762 -
```

```
recall: 0.8850 - val_accuracy: 0.8650 - val_loss: 0.3094 - val_precision: 0.8724
- val_recall: 0.8550
Epoch 24/30
13/13 - 24s - 2s/step - accuracy: 0.8831 - loss: 0.2655 - precision: 0.9081 -
recall: 0.8525 - val_accuracy: 0.8425 - val_loss: 0.3599 - val_precision: 0.8018
- val_recall: 0.9100
Epoch 25/30
13/13 - 23s - 2s/step - accuracy: 0.8819 - loss: 0.2959 - precision: 0.8932 -
recall: 0.8675 - val_accuracy: 0.8375 - val_loss: 0.3807 - val_precision: 0.8169
- val_recall: 0.8700
Epoch 26/30
13/13 - 23s - 2s/step - accuracy: 0.8856 - loss: 0.2695 - precision: 0.8832 -
recall: 0.8888 - val_accuracy: 0.8750 - val_loss: 0.3086 - val_precision: 0.9076
- val_recall: 0.8350
Epoch 27/30
13/13 - 23s - 2s/step - accuracy: 0.8788 - loss: 0.2860 - precision: 0.8905 -
recall: 0.8637 - val_accuracy: 0.8775 - val_loss: 0.3066 - val_precision: 0.8683
- val_recall: 0.8900
Epoch 28/30
13/13 - 23s - 2s/step - accuracy: 0.8950 - loss: 0.2593 - precision: 0.8970 -
recall: 0.8925 - val_accuracy: 0.8550 - val_loss: 0.3405 - val_precision: 0.8698
- val_recall: 0.8350
Epoch 29/30
13/13 - 23s - 2s/step - accuracy: 0.8819 - loss: 0.2680 - precision: 0.8912 -
recall: 0.8700 - val_accuracy: 0.8550 - val_loss: 0.3350 - val_precision: 0.8349
- val_recall: 0.8850
Epoch 30/30
13/13 - 24s - 2s/step - accuracy: 0.8825 - loss: 0.2706 - precision: 0.8883 -
recall: 0.8750 - val_accuracy: 0.8700 - val_loss: 0.3291 - val_precision: 0.8491
- val_recall: 0.9000
Restoring model weights from the end of the best epoch: 27.
evaluating best model from this run
4/4             2s 350ms/step -
accuracy: 0.8711 - loss: 0.3178 - precision: 0.6489 - recall: 0.7128
testing: lr=1e-05, filters_conv1=32, units_dense1=256, dropout_rate=0.4
Epoch 1/30
13/13 - 25s - 2s/step - accuracy: 0.6730 - loss: 0.6429 - precision: 0.6510 -
recall: 0.7460 - val_accuracy: 0.6375 - val_loss: 0.6406 - val_precision: 0.6425
- val_recall: 0.6200
Epoch 2/30
13/13 - 24s - 2s/step - accuracy: 0.6925 - loss: 0.5883 - precision: 0.6754 -
recall: 0.7412 - val_accuracy: 0.6225 - val_loss: 0.6424 - val_precision: 0.6150
- val_recall: 0.6550
Epoch 3/30
13/13 - 24s - 2s/step - accuracy: 0.7269 - loss: 0.5572 - precision: 0.7070 -
recall: 0.7750 - val_accuracy: 0.6525 - val_loss: 0.6383 - val_precision: 0.6649
- val_recall: 0.6150
Epoch 4/30
```

```
13/13 - 24s - 2s/step - accuracy: 0.7381 - loss: 0.5316 - precision: 0.7228 -
recall: 0.7725 - val_accuracy: 0.6225 - val_loss: 0.6354 - val_precision: 0.6043
- val_recall: 0.7100
Epoch 5/30
13/13 - 24s - 2s/step - accuracy: 0.7294 - loss: 0.5358 - precision: 0.7078 -
recall: 0.7812 - val_accuracy: 0.6225 - val_loss: 0.6337 - val_precision: 0.5984
- val_recall: 0.7450
Epoch 6/30
13/13 - 24s - 2s/step - accuracy: 0.7506 - loss: 0.5281 - precision: 0.7485 -
recall: 0.7550 - val_accuracy: 0.6475 - val_loss: 0.6183 - val_precision: 0.6311
- val_recall: 0.7100
Epoch 7/30
13/13 - 24s - 2s/step - accuracy: 0.7394 - loss: 0.5216 - precision: 0.7229 -
recall: 0.7763 - val_accuracy: 0.6700 - val_loss: 0.6053 - val_precision: 0.6635
- val_recall: 0.6900
Epoch 8/30
13/13 - 24s - 2s/step - accuracy: 0.7569 - loss: 0.5247 - precision: 0.7638 -
recall: 0.7437 - val_accuracy: 0.6175 - val_loss: 0.6299 - val_precision: 0.5874
- val_recall: 0.7900
Epoch 9/30
13/13 - 26s - 2s/step - accuracy: 0.7506 - loss: 0.5186 - precision: 0.7436 -
recall: 0.7650 - val_accuracy: 0.6950 - val_loss: 0.5901 - val_precision: 0.7167
- val_recall: 0.6450
Epoch 10/30
13/13 - 25s - 2s/step - accuracy: 0.7700 - loss: 0.4984 - precision: 0.7734 -
recall: 0.7638 - val_accuracy: 0.6325 - val_loss: 0.6112 - val_precision: 0.6039
- val_recall: 0.7700
Epoch 11/30
13/13 - 25s - 2s/step - accuracy: 0.7625 - loss: 0.5103 - precision: 0.7756 -
recall: 0.7387 - val_accuracy: 0.6775 - val_loss: 0.5859 - val_precision: 0.6550
- val_recall: 0.7500
Epoch 12/30
13/13 - 25s - 2s/step - accuracy: 0.7769 - loss: 0.4938 - precision: 0.7843 -
recall: 0.7638 - val_accuracy: 0.6500 - val_loss: 0.5968 - val_precision: 0.6172
- val_recall: 0.7900
Epoch 13/30
13/13 - 24s - 2s/step - accuracy: 0.7837 - loss: 0.4819 - precision: 0.7956 -
recall: 0.7638 - val_accuracy: 0.6875 - val_loss: 0.5733 - val_precision: 0.6623
- val_recall: 0.7650
Epoch 14/30
13/13 - 24s - 2s/step - accuracy: 0.7681 - loss: 0.4894 - precision: 0.7625 -
recall: 0.7788 - val_accuracy: 0.7200 - val_loss: 0.5546 - val_precision: 0.7037
- val_recall: 0.7600
Epoch 15/30
13/13 - 24s - 2s/step - accuracy: 0.7931 - loss: 0.4766 - precision: 0.8098 -
recall: 0.7663 - val_accuracy: 0.6950 - val_loss: 0.5597 - val_precision: 0.6653
- val_recall: 0.7850
Epoch 16/30
```

```
13/13 - 24s - 2s/step - accuracy: 0.7831 - loss: 0.4770 - precision: 0.8032 -
recall: 0.7500 - val_accuracy: 0.6725 - val_loss: 0.5675 - val_precision: 0.6408
- val_recall: 0.7850
Epoch 17/30
13/13 - 24s - 2s/step - accuracy: 0.7887 - loss: 0.4634 - precision: 0.8056 -
recall: 0.7613 - val_accuracy: 0.7175 - val_loss: 0.5456 - val_precision: 0.7023
- val_recall: 0.7550
Epoch 18/30
13/13 - 24s - 2s/step - accuracy: 0.7956 - loss: 0.4641 - precision: 0.7953 -
recall: 0.7962 - val_accuracy: 0.7475 - val_loss: 0.5287 - val_precision: 0.7592
- val_recall: 0.7250
Epoch 19/30
13/13 - 24s - 2s/step - accuracy: 0.7900 - loss: 0.4654 - precision: 0.8135 -
recall: 0.7525 - val_accuracy: 0.6675 - val_loss: 0.5842 - val_precision: 0.6159
- val_recall: 0.8900
Epoch 20/30
13/13 - 24s - 2s/step - accuracy: 0.7806 - loss: 0.4685 - precision: 0.7904 -
recall: 0.7638 - val_accuracy: 0.7350 - val_loss: 0.5288 - val_precision: 0.7098
- val_recall: 0.7950
Epoch 21/30
13/13 - 24s - 2s/step - accuracy: 0.8056 - loss: 0.4453 - precision: 0.8007 -
recall: 0.8138 - val_accuracy: 0.7200 - val_loss: 0.5280 - val_precision: 0.6947
- val_recall: 0.7850
Epoch 22/30
13/13 - 24s - 2s/step - accuracy: 0.8012 - loss: 0.4495 - precision: 0.8222 -
recall: 0.7688 - val_accuracy: 0.7150 - val_loss: 0.5337 - val_precision: 0.6777
- val_recall: 0.8200
Epoch 23/30
13/13 - 24s - 2s/step - accuracy: 0.8025 - loss: 0.4392 - precision: 0.8201 -
recall: 0.7750 - val_accuracy: 0.7150 - val_loss: 0.5276 - val_precision: 0.6706
- val_recall: 0.8450
Epoch 24/30
13/13 - 24s - 2s/step - accuracy: 0.7819 - loss: 0.4428 - precision: 0.7932 -
recall: 0.7625 - val_accuracy: 0.7250 - val_loss: 0.5188 - val_precision: 0.6815
- val_recall: 0.8450
Epoch 25/30
13/13 - 24s - 2s/step - accuracy: 0.7956 - loss: 0.4560 - precision: 0.8192 -
recall: 0.7588 - val_accuracy: 0.7375 - val_loss: 0.5328 - val_precision: 0.6753
- val_recall: 0.9150
Epoch 26/30
13/13 - 24s - 2s/step - accuracy: 0.8006 - loss: 0.4374 - precision: 0.8048 -
recall: 0.7937 - val_accuracy: 0.7550 - val_loss: 0.5031 - val_precision: 0.7429
- val_recall: 0.7800
Epoch 27/30
13/13 - 24s - 2s/step - accuracy: 0.8106 - loss: 0.4362 - precision: 0.8182 -
recall: 0.7987 - val_accuracy: 0.7425 - val_loss: 0.5078 - val_precision: 0.7175
- val_recall: 0.8000
Epoch 28/30
```

```
13/13 - 24s - 2s/step - accuracy: 0.8056 - loss: 0.4382 - precision: 0.8230 -
recall: 0.7788 - val_accuracy: 0.7300 - val_loss: 0.5205 - val_precision: 0.6769
- val_recall: 0.8800
Epoch 29/30
13/13 - 24s - 2s/step - accuracy: 0.8050 - loss: 0.4306 - precision: 0.8112 -
recall: 0.7950 - val_accuracy: 0.7600 - val_loss: 0.4926 - val_precision: 0.7430
- val_recall: 0.7950
Epoch 30/30
13/13 - 24s - 2s/step - accuracy: 0.8156 - loss: 0.4294 - precision: 0.8344 -
recall: 0.7875 - val_accuracy: 0.7250 - val_loss: 0.5349 - val_precision: 0.6630
- val_recall: 0.9150
Restoring model weights from the end of the best epoch: 29.
evaluating best model from this run
4/4             2s 368ms/step -
accuracy: 0.7616 - loss: 0.4860 - precision: 0.5338 - recall: 0.6421
testing: lr=0.0001, filters_conv1=32, units_dense1=256, dropout_rate=0.2
Epoch 1/30
13/13 - 25s - 2s/step - accuracy: 0.6680 - loss: 0.6443 - precision: 0.6591 -
recall: 0.6960 - val_accuracy: 0.6000 - val_loss: 0.7045 - val_precision: 0.5990
- val_recall: 0.6050
Epoch 2/30
13/13 - 24s - 2s/step - accuracy: 0.7156 - loss: 0.5680 - precision: 0.7027 -
recall: 0.7475 - val_accuracy: 0.5900 - val_loss: 0.6542 - val_precision: 0.5667
- val_recall: 0.7650
Epoch 3/30
13/13 - 24s - 2s/step - accuracy: 0.7356 - loss: 0.5363 - precision: 0.7371 -
recall: 0.7325 - val_accuracy: 0.5975 - val_loss: 0.6836 - val_precision: 0.5684
- val_recall: 0.8100
Epoch 4/30
13/13 - 24s - 2s/step - accuracy: 0.7638 - loss: 0.5037 - precision: 0.7698 -
recall: 0.7525 - val_accuracy: 0.7075 - val_loss: 0.5929 - val_precision: 0.7085
- val_recall: 0.7050
Epoch 5/30
13/13 - 24s - 2s/step - accuracy: 0.7912 - loss: 0.4686 - precision: 0.8174 -
recall: 0.7500 - val_accuracy: 0.6925 - val_loss: 0.5762 - val_precision: 0.6559
- val_recall: 0.8100
Epoch 6/30
13/13 - 24s - 2s/step - accuracy: 0.7638 - loss: 0.4898 - precision: 0.7678 -
recall: 0.7563 - val_accuracy: 0.6700 - val_loss: 0.5941 - val_precision: 0.6328
- val_recall: 0.8100
Epoch 7/30
13/13 - 24s - 2s/step - accuracy: 0.8019 - loss: 0.4484 - precision: 0.8173 -
recall: 0.7775 - val_accuracy: 0.7425 - val_loss: 0.5295 - val_precision: 0.7064
- val_recall: 0.8300
Epoch 8/30
13/13 - 24s - 2s/step - accuracy: 0.8175 - loss: 0.4292 - precision: 0.8414 -
recall: 0.7825 - val_accuracy: 0.7375 - val_loss: 0.5114 - val_precision: 0.7093
- val_recall: 0.8050
```

```
Epoch 9/30
13/13 - 24s - 2s/step - accuracy: 0.7900 - loss: 0.4489 - precision: 0.8053 -
recall: 0.7650 - val_accuracy: 0.7625 - val_loss: 0.4931 - val_precision: 0.7178
- val_recall: 0.8650
Epoch 10/30
13/13 - 24s - 2s/step - accuracy: 0.8037 - loss: 0.4140 - precision: 0.7914 -
recall: 0.8250 - val_accuracy: 0.7400 - val_loss: 0.4902 - val_precision: 0.7162
- val_recall: 0.7950
Epoch 11/30
13/13 - 24s - 2s/step - accuracy: 0.8231 - loss: 0.3968 - precision: 0.8362 -
recall: 0.8037 - val_accuracy: 0.7800 - val_loss: 0.4513 - val_precision: 0.7772
- val_recall: 0.7850
Epoch 12/30
13/13 - 24s - 2s/step - accuracy: 0.8144 - loss: 0.4011 - precision: 0.8279 -
recall: 0.7937 - val_accuracy: 0.7700 - val_loss: 0.4689 - val_precision: 0.7195
- val_recall: 0.8850
Epoch 13/30
13/13 - 24s - 2s/step - accuracy: 0.8444 - loss: 0.3603 - precision: 0.8649 -
recall: 0.8163 - val_accuracy: 0.7975 - val_loss: 0.4377 - val_precision: 0.7598
- val_recall: 0.8700
Epoch 14/30
13/13 - 24s - 2s/step - accuracy: 0.8388 - loss: 0.3542 - precision: 0.8465 -
recall: 0.8275 - val_accuracy: 0.7125 - val_loss: 0.5447 - val_precision: 0.8829
- val_recall: 0.4900
Epoch 15/30
13/13 - 26s - 2s/step - accuracy: 0.8338 - loss: 0.3654 - precision: 0.8551 -
recall: 0.8037 - val_accuracy: 0.7850 - val_loss: 0.4374 - val_precision: 0.7436
- val_recall: 0.8700
Epoch 16/30
13/13 - 24s - 2s/step - accuracy: 0.8369 - loss: 0.3567 - precision: 0.8390 -
recall: 0.8338 - val_accuracy: 0.8000 - val_loss: 0.4435 - val_precision: 0.7439
- val_recall: 0.9150
Epoch 17/30
13/13 - 24s - 2s/step - accuracy: 0.8481 - loss: 0.3450 - precision: 0.8631 -
recall: 0.8275 - val_accuracy: 0.8325 - val_loss: 0.3819 - val_precision: 0.8308
- val_recall: 0.8350
Epoch 18/30
13/13 - 24s - 2s/step - accuracy: 0.8556 - loss: 0.3273 - precision: 0.8482 -
recall: 0.8662 - val_accuracy: 0.8200 - val_loss: 0.3781 - val_precision: 0.8441
- val_recall: 0.7850
Epoch 19/30
13/13 - 24s - 2s/step - accuracy: 0.8469 - loss: 0.3428 - precision: 0.8715 -
recall: 0.8138 - val_accuracy: 0.8150 - val_loss: 0.3732 - val_precision: 0.8058
- val_recall: 0.8300
Epoch 20/30
13/13 - 24s - 2s/step - accuracy: 0.8369 - loss: 0.3413 - precision: 0.8348 -
recall: 0.8400 - val_accuracy: 0.8300 - val_loss: 0.3814 - val_precision: 0.8367
- val_recall: 0.8200
```

```
Epoch 21/30
13/13 - 24s - 2s/step - accuracy: 0.8650 - loss: 0.3280 - precision: 0.8893 -
recall: 0.8338 - val_accuracy: 0.8450 - val_loss: 0.3627 - val_precision: 0.8382
- val_recall: 0.8550
Epoch 22/30
13/13 - 24s - 2s/step - accuracy: 0.8669 - loss: 0.3196 - precision: 0.8768 -
recall: 0.8537 - val_accuracy: 0.8325 - val_loss: 0.4056 - val_precision: 0.8009
- val_recall: 0.8850
Epoch 23/30
13/13 - 24s - 2s/step - accuracy: 0.8562 - loss: 0.3158 - precision: 0.8682 -
recall: 0.8400 - val_accuracy: 0.8150 - val_loss: 0.3821 - val_precision: 0.7763
- val_recall: 0.8850
Epoch 24/30
13/13 - 24s - 2s/step - accuracy: 0.8656 - loss: 0.3240 - precision: 0.8736 -
recall: 0.8550 - val_accuracy: 0.8175 - val_loss: 0.3829 - val_precision: 0.8629
- val_recall: 0.7550
Epoch 25/30
13/13 - 24s - 2s/step - accuracy: 0.8594 - loss: 0.3112 - precision: 0.8818 -
recall: 0.8300 - val_accuracy: 0.8425 - val_loss: 0.3650 - val_precision: 0.8186
- val_recall: 0.8800
Epoch 26/30
13/13 - 24s - 2s/step - accuracy: 0.8581 - loss: 0.3029 - precision: 0.8668 -
recall: 0.8462 - val_accuracy: 0.8425 - val_loss: 0.3552 - val_precision: 0.8408
- val_recall: 0.8450
Epoch 27/30
13/13 - 24s - 2s/step - accuracy: 0.8788 - loss: 0.2902 - precision: 0.8816 -
recall: 0.8750 - val_accuracy: 0.8650 - val_loss: 0.3435 - val_precision: 0.8802
- val_recall: 0.8450
Epoch 28/30
13/13 - 24s - 2s/step - accuracy: 0.8650 - loss: 0.3111 - precision: 0.8659 -
recall: 0.8637 - val_accuracy: 0.8425 - val_loss: 0.3795 - val_precision: 0.8216
- val_recall: 0.8750
Epoch 29/30
13/13 - 24s - 2s/step - accuracy: 0.8769 - loss: 0.2932 - precision: 0.8860 -
recall: 0.8650 - val_accuracy: 0.8325 - val_loss: 0.3514 - val_precision: 0.8009
- val_recall: 0.8850
Epoch 30/30
13/13 - 24s - 2s/step - accuracy: 0.8700 - loss: 0.3035 - precision: 0.8844 -
recall: 0.8512 - val_accuracy: 0.8350 - val_loss: 0.3500 - val_precision: 0.8160
- val_recall: 0.8650
Restoring model weights from the end of the best epoch: 27.
evaluating best model from this run
4/4             2s 360ms/step -
accuracy: 0.8726 - loss: 0.3357 - precision: 0.6607 - recall: 0.6754
testing: lr=0.0001, filters_conv1=64, units_dense1=256, dropout_rate=0.4
Epoch 1/30
13/13 - 39s - 3s/step - accuracy: 0.6285 - loss: 0.7064 - precision: 0.6124 -
recall: 0.7000 - val_accuracy: 0.6475 - val_loss: 0.6505 - val_precision: 0.6468
```

```
- val_recall: 0.6500
Epoch 2/30
13/13 - 38s - 3s/step - accuracy: 0.6731 - loss: 0.6069 - precision: 0.6572 -
recall: 0.7237 - val_accuracy: 0.5800 - val_loss: 0.6617 - val_precision: 0.5567
- val_recall: 0.7850
Epoch 3/30
13/13 - 39s - 3s/step - accuracy: 0.6969 - loss: 0.5857 - precision: 0.6829 -
recall: 0.7350 - val_accuracy: 0.6225 - val_loss: 0.6490 - val_precision: 0.6089
- val_recall: 0.6850
Epoch 4/30
13/13 - 39s - 3s/step - accuracy: 0.7262 - loss: 0.5487 - precision: 0.7029 -
recall: 0.7837 - val_accuracy: 0.6425 - val_loss: 0.6423 - val_precision: 0.6377
- val_recall: 0.6600
Epoch 5/30
13/13 - 38s - 3s/step - accuracy: 0.7206 - loss: 0.5389 - precision: 0.7040 -
recall: 0.7613 - val_accuracy: 0.6875 - val_loss: 0.6134 - val_precision: 0.6777
- val_recall: 0.7150
Epoch 6/30
13/13 - 38s - 3s/step - accuracy: 0.7469 - loss: 0.5084 - precision: 0.7400 -
recall: 0.7613 - val_accuracy: 0.7075 - val_loss: 0.5900 - val_precision: 0.6878
- val_recall: 0.7600
Epoch 7/30
13/13 - 48s - 4s/step - accuracy: 0.7444 - loss: 0.4975 - precision: 0.7276 -
recall: 0.7812 - val_accuracy: 0.7125 - val_loss: 0.5933 - val_precision: 0.7673
- val_recall: 0.6100
Epoch 8/30
13/13 - 44s - 3s/step - accuracy: 0.7619 - loss: 0.4934 - precision: 0.7609 -
recall: 0.7638 - val_accuracy: 0.7325 - val_loss: 0.5642 - val_precision: 0.7487
- val_recall: 0.7000
Epoch 9/30
13/13 - 39s - 3s/step - accuracy: 0.7656 - loss: 0.4734 - precision: 0.7653 -
recall: 0.7663 - val_accuracy: 0.7575 - val_loss: 0.5477 - val_precision: 0.7943
- val_recall: 0.6950
Epoch 10/30
13/13 - 38s - 3s/step - accuracy: 0.7862 - loss: 0.4638 - precision: 0.7921 -
recall: 0.7763 - val_accuracy: 0.7775 - val_loss: 0.5381 - val_precision: 0.8323
- val_recall: 0.6950
Epoch 11/30
13/13 - 39s - 3s/step - accuracy: 0.7881 - loss: 0.4581 - precision: 0.7892 -
recall: 0.7862 - val_accuracy: 0.7825 - val_loss: 0.5225 - val_precision: 0.8156
- val_recall: 0.7300
Epoch 12/30
13/13 - 37s - 3s/step - accuracy: 0.7950 - loss: 0.4507 - precision: 0.8018 -
recall: 0.7837 - val_accuracy: 0.7450 - val_loss: 0.5205 - val_precision: 0.6960
- val_recall: 0.8700
Epoch 13/30
13/13 - 37s - 3s/step - accuracy: 0.8019 - loss: 0.4351 - precision: 0.8030 -
recall: 0.8000 - val_accuracy: 0.7850 - val_loss: 0.5112 - val_precision: 0.7591
```

```
- val_recall: 0.8350
Epoch 14/30
13/13 - 37s - 3s/step - accuracy: 0.8094 - loss: 0.4287 - precision: 0.8278 -
recall: 0.7812 - val_accuracy: 0.7800 - val_loss: 0.4988 - val_precision: 0.7188
- val_recall: 0.9200
Epoch 15/30
13/13 - 37s - 3s/step - accuracy: 0.7819 - loss: 0.4388 - precision: 0.7753 -
recall: 0.7937 - val_accuracy: 0.7750 - val_loss: 0.4872 - val_precision: 0.7670
- val_recall: 0.7900
Epoch 16/30
13/13 - 37s - 3s/step - accuracy: 0.8081 - loss: 0.4201 - precision: 0.7981 -
recall: 0.8250 - val_accuracy: 0.8025 - val_loss: 0.4802 - val_precision: 0.7531
- val_recall: 0.9000
Epoch 17/30
13/13 - 37s - 3s/step - accuracy: 0.8175 - loss: 0.3984 - precision: 0.8265 -
recall: 0.8037 - val_accuracy: 0.7700 - val_loss: 0.4902 - val_precision: 0.7700
- val_recall: 0.7700
Epoch 18/30
13/13 - 37s - 3s/step - accuracy: 0.8150 - loss: 0.4012 - precision: 0.8333 -
recall: 0.7875 - val_accuracy: 0.8075 - val_loss: 0.4400 - val_precision: 0.7887
- val_recall: 0.8400
Epoch 19/30
13/13 - 37s - 3s/step - accuracy: 0.8244 - loss: 0.3979 - precision: 0.8323 -
recall: 0.8125 - val_accuracy: 0.8000 - val_loss: 0.4404 - val_precision: 0.8061
- val_recall: 0.7900
Epoch 20/30
13/13 - 37s - 3s/step - accuracy: 0.8231 - loss: 0.3825 - precision: 0.8397 -
recall: 0.7987 - val_accuracy: 0.8100 - val_loss: 0.4379 - val_precision: 0.7844
- val_recall: 0.8550
Epoch 21/30
13/13 - 37s - 3s/step - accuracy: 0.8300 - loss: 0.3650 - precision: 0.8492 -
recall: 0.8025 - val_accuracy: 0.7925 - val_loss: 0.4108 - val_precision: 0.8095
- val_recall: 0.7650
Epoch 22/30
13/13 - 37s - 3s/step - accuracy: 0.8269 - loss: 0.3616 - precision: 0.8257 -
recall: 0.8288 - val_accuracy: 0.7950 - val_loss: 0.4538 - val_precision: 0.7500
- val_recall: 0.8850
Epoch 23/30
13/13 - 37s - 3s/step - accuracy: 0.8294 - loss: 0.3668 - precision: 0.8357 -
recall: 0.8200 - val_accuracy: 0.8275 - val_loss: 0.4148 - val_precision: 0.7964
- val_recall: 0.8800
Epoch 24/30
13/13 - 37s - 3s/step - accuracy: 0.8456 - loss: 0.3576 - precision: 0.8742 -
recall: 0.8075 - val_accuracy: 0.8100 - val_loss: 0.4072 - val_precision: 0.8444
- val_recall: 0.7600
Epoch 25/30
13/13 - 37s - 3s/step - accuracy: 0.8338 - loss: 0.3703 - precision: 0.8532 -
recall: 0.8062 - val_accuracy: 0.8125 - val_loss: 0.4191 - val_precision: 0.8079
```

```
- val_recall: 0.8200
Epoch 26/30
13/13 - 37s - 3s/step - accuracy: 0.8325 - loss: 0.3636 - precision: 0.8317 -
recall: 0.8338 - val_accuracy: 0.8275 - val_loss: 0.4304 - val_precision: 0.8743
- val_recall: 0.7650
Epoch 27/30
13/13 - 39s - 3s/step - accuracy: 0.8356 - loss: 0.3678 - precision: 0.8456 -
recall: 0.8213 - val_accuracy: 0.8200 - val_loss: 0.4133 - val_precision: 0.8077
- val_recall: 0.8400
Epoch 28/30
13/13 - 37s - 3s/step - accuracy: 0.8438 - loss: 0.3466 - precision: 0.8553 -
recall: 0.8275 - val_accuracy: 0.8175 - val_loss: 0.3786 - val_precision: 0.8470
- val_recall: 0.7750
Epoch 29/30
13/13 - 37s - 3s/step - accuracy: 0.8469 - loss: 0.3558 - precision: 0.8526 -
recall: 0.8388 - val_accuracy: 0.8300 - val_loss: 0.4057 - val_precision: 0.7845
- val_recall: 0.9100
Epoch 30/30
13/13 - 37s - 3s/step - accuracy: 0.8500 - loss: 0.3489 - precision: 0.8636 -
recall: 0.8313 - val_accuracy: 0.8075 - val_loss: 0.4124 - val_precision: 0.7709
- val_recall: 0.8750
Restoring model weights from the end of the best epoch: 28.
evaluating best model from this run
4/4             2s 522ms/step -
accuracy: 0.8322 - loss: 0.3633 - precision: 0.6262 - recall: 0.6165
hyperparameter tuning finished.
```

[21]:
```python
# convert results to dataframe
results_df = pd.DataFrame(results_list)

# sort by validation accuracy (descending)
results_df = results_df.sort_values(by="val_accuracy", ascending=False)

# print results
print("\ntuning results summary:")
print(results_df)

# save results to csv
results_filename = "../results/hyperparameter_tuning_results_random_search.csv"
results_df.to_csv(results_filename, index=False)
print(f"\nresults saved to {results_filename}")
```

```
tuning results summary:
   learning_rate  filters_conv1  units_dense1  dropout_rate  val_loss  \
6        0.00100             32           256           0.2  0.306606
5        0.00100             64           256           0.3  0.298156
8        0.00010             32           256           0.2  0.343528
```

```
0        0.00100             16             64           0.2  0.314409
3        0.00100             32            128           0.2  0.335180
9        0.00010             64            256           0.4  0.378565
1        0.00010             16            128           0.4  0.406113
4        0.00001             16            256           0.3  0.468071
7        0.00001             32            256           0.4  0.492562
2        0.00001             64             64           0.4  0.596940

   val_accuracy  val_precision  val_recall  epochs_trained
6        0.8775       0.868293       0.890              30
5        0.8725       0.911602       0.825              30
8        0.8650       0.880208       0.845              30
0        0.8625       0.914286       0.800              30
3        0.8625       0.871795       0.850              30
9        0.8175       0.846995       0.775              30
1        0.8075       0.783410       0.850              30
4        0.7950       0.778302       0.825              30
7        0.7600       0.742991       0.795              30
2        0.6850       0.672897       0.720              30

results saved to ../results/hyperparameter_tuning_results_random_search.csv
```

[22]:
```python
# concatenate train and val data
X_all = []
y_all = []

train_data_gen.reset()
val_data_gen.reset()

for batch_x, batch_y in train_data_gen:
    X_all.append(batch_x)
    y_all.append(batch_y)
    if len(X_all) * BATCH_SIZE >= train_data_gen.samples:
        break

for batch_x, batch_y in val_data_gen:
    X_all.append(batch_x)
    y_all.append(batch_y)
    if len(X_all) * BATCH_SIZE >= val_data_gen.samples + train_data_gen.samples:
        break

X_all = np.concatenate(X_all)
y_all = np.concatenate(y_all)

# create dataset
train_val_ds = (
    tf.data.Dataset.from_tensor_slices((X_all, y_all))
```

```
    .shuffle(1000)
    .batch(BATCH_SIZE)
    .prefetch(tf.data.AUTOTUNE)
)
```

[23]:
```
print("\nretraining the best model on the combined train+validation data")
best_params = results_df.iloc[0]
best_lr = best_params["learning_rate"]
best_filters_conv1 = int(
    best_params["filters_conv1"]
)  # cast to int, wasn't working before
best_units_dense1 = int(
    best_params["units_dense1"]
)  # cast to int, wasn't working before
best_dropout_rate = best_params["dropout_rate"]
print(
    f"best hyperparameters found: lr={best_lr},␣
 ↪filters_conv1={best_filters_conv1}, units_dense1={best_units_dense1},␣
 ↪dropout_rate={best_dropout_rate}"
)
```

```
retraining the best model on the combined train+validation data
best hyperparameters found: lr=0.001, filters_conv1=32, units_dense1=256,
dropout_rate=0.2
```

[24]:
```
# build the best model
final_model = build_model(
    input_shape,
    filters_conv1=best_filters_conv1,
    units_dense1=best_units_dense1,
    dropout_rate=best_dropout_rate,
)

# compile the best model
optimizer = tf.keras.optimizers.Adam(learning_rate=best_lr)
final_model.compile(optimizer=optimizer, loss="binary_crossentropy",␣
 ↪metrics=metrics)

# define steps per epoch
steps_per_epoch = (train_data_gen.samples + val_data_gen.samples) // BATCH_SIZE

# train the best model on the combined train+val data
final_history = final_model.fit(
    train_val_ds,
    epochs=EPOCHS,
    steps_per_epoch=steps_per_epoch,
```

```
    verbose=2,
)
```

```
Epoch 1/30
15/15 - 27s - 2s/step - accuracy: 0.5707 - loss: 1.5957 - precision: 0.5830 -
recall: 0.4757
Epoch 2/30
15/15 - 1s - 55ms/step - accuracy: 0.5469 - loss: 0.6493 - precision: 0.5246 -
recall: 1.0000
Epoch 3/30
```

```
/opt/anaconda3/envs/ml-2025/lib/python3.12/site-
packages/keras/src/trainers/epoch_iterator.py:107: UserWarning: Your input ran
out of data; interrupting training. Make sure that your dataset or generator can
generate at least `steps_per_epoch * epochs` batches. You may need to use the
`.repeat()` function when building your dataset.
  self._interrupted_warning()
```

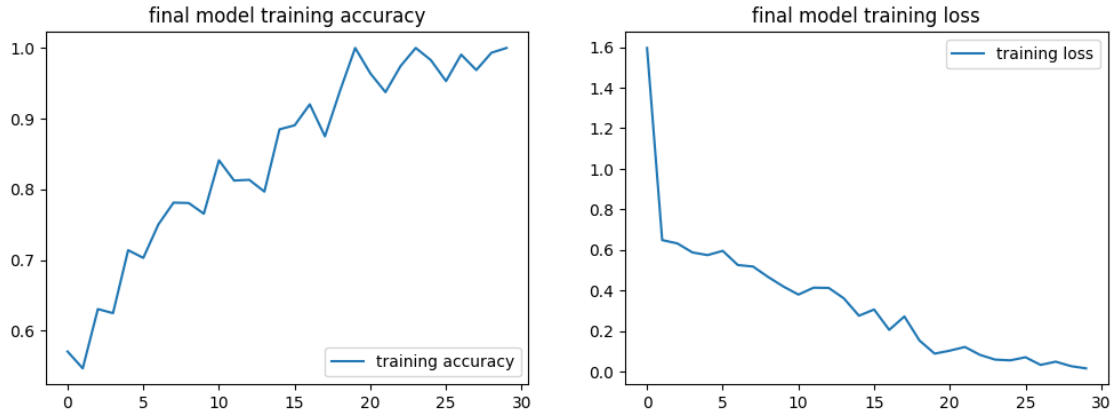
```
15/15 - 25s - 2s/step - accuracy: 0.6307 - loss: 0.6330 - precision: 0.6019 -
recall: 0.7518
Epoch 4/30
15/15 - 1s - 55ms/step - accuracy: 0.6250 - loss: 0.5886 - precision: 0.6098 -
recall: 0.7576
Epoch 5/30
15/15 - 25s - 2s/step - accuracy: 0.7141 - loss: 0.5754 - precision: 0.7040 -
recall: 0.7284
Epoch 6/30
15/15 - 1s - 70ms/step - accuracy: 0.7031 - loss: 0.5965 - precision: 0.7586 -
recall: 0.6471
Epoch 7/30
15/15 - 27s - 2s/step - accuracy: 0.7505 - loss: 0.5264 - precision: 0.7714 -
recall: 0.7075
Epoch 8/30
15/15 - 1s - 57ms/step - accuracy: 0.7812 - loss: 0.5192 - precision: 0.7353 -
recall: 0.8333
Epoch 9/30
15/15 - 25s - 2s/step - accuracy: 0.7807 - loss: 0.4676 - precision: 0.7984 -
recall: 0.7474
Epoch 10/30
15/15 - 1s - 62ms/step - accuracy: 0.7656 - loss: 0.4210 - precision: 0.7586 -
recall: 0.7333
Epoch 11/30
15/15 - 25s - 2s/step - accuracy: 0.8411 - loss: 0.3809 - precision: 0.8575 -
recall: 0.8151
Epoch 12/30
15/15 - 1s - 56ms/step - accuracy: 0.8125 - loss: 0.4149 - precision: 0.7778 -
recall: 0.8750
Epoch 13/30
15/15 - 25s - 2s/step - accuracy: 0.8135 - loss: 0.4139 - precision: 0.8295 -
```

```
recall: 0.7859
Epoch 14/30
15/15 - 1s - 55ms/step - accuracy: 0.7969 - loss: 0.3625 - precision: 0.7812 -
recall: 0.8065
Epoch 15/30
15/15 - 26s - 2s/step - accuracy: 0.8849 - loss: 0.2768 - precision: 0.9120 -
recall: 0.8496
Epoch 16/30
15/15 - 1s - 64ms/step - accuracy: 0.8906 - loss: 0.3074 - precision: 0.9643 -
recall: 0.8182
Epoch 17/30
15/15 - 28s - 2s/step - accuracy: 0.9203 - loss: 0.2072 - precision: 0.9344 -
recall: 0.9018
Epoch 18/30
15/15 - 1s - 57ms/step - accuracy: 0.8750 - loss: 0.2731 - precision: 0.9394 -
recall: 0.8378
Epoch 19/30
15/15 - 27s - 2s/step - accuracy: 0.9396 - loss: 0.1546 - precision: 0.9422 -
recall: 0.9363
Epoch 20/30
15/15 - 1s - 59ms/step - accuracy: 1.0000 - loss: 0.0902 - precision: 1.0000 -
recall: 1.0000
Epoch 21/30
15/15 - 26s - 2s/step - accuracy: 0.9641 - loss: 0.1047 - precision: 0.9720 -
recall: 0.9546
Epoch 22/30
15/15 - 1s - 56ms/step - accuracy: 0.9375 - loss: 0.1230 - precision: 1.0000 -
recall: 0.8919
Epoch 23/30
15/15 - 26s - 2s/step - accuracy: 0.9745 - loss: 0.0838 - precision: 0.9718 -
recall: 0.9769
Epoch 24/30
15/15 - 1s - 56ms/step - accuracy: 1.0000 - loss: 0.0606 - precision: 1.0000 -
recall: 1.0000
Epoch 25/30
15/15 - 25s - 2s/step - accuracy: 0.9828 - loss: 0.0570 - precision: 0.9893 -
recall: 0.9758
Epoch 26/30
15/15 - 1s - 55ms/step - accuracy: 0.9531 - loss: 0.0723 - precision: 0.9688 -
recall: 0.9394
Epoch 27/30
15/15 - 25s - 2s/step - accuracy: 0.9906 - loss: 0.0346 - precision: 0.9875 -
recall: 0.9937
Epoch 28/30
15/15 - 1s - 56ms/step - accuracy: 0.9688 - loss: 0.0505 - precision: 0.9643 -
recall: 0.9643
Epoch 29/30
15/15 - 25s - 2s/step - accuracy: 0.9932 - loss: 0.0286 - precision: 0.9937 -
```

```
recall: 0.9927
Epoch 30/30
15/15 - 1s - 56ms/step - accuracy: 1.0000 - loss: 0.0175 - precision: 1.0000 -
recall: 1.0000
```

[25]:
```python
print("\nplotting final training history")
final_acc = final_history.history["accuracy"]
final_loss = final_history.history["loss"]

plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, final_acc, label="training accuracy")
plt.legend(loc="lower right")
plt.title("final model training accuracy")

plt.subplot(1, 2, 2)
plt.plot(epochs_range, final_loss, label="training loss")
plt.legend(loc="upper right")
plt.title("final model training loss")
plt.show()
```

```
plotting final training history
```



[26]:
```python
model_save_path = "../models/baseline_model.keras"
print(f"\nsaving final model to {model_save_path}")
final_model.save(model_save_path)

print("loading model back")
loaded_model = tf.keras.models.load_model(model_save_path)
print("model loaded successfully")
```

```
saving final model to ../models/baseline_model.keras
loading model back
model loaded successfully
```

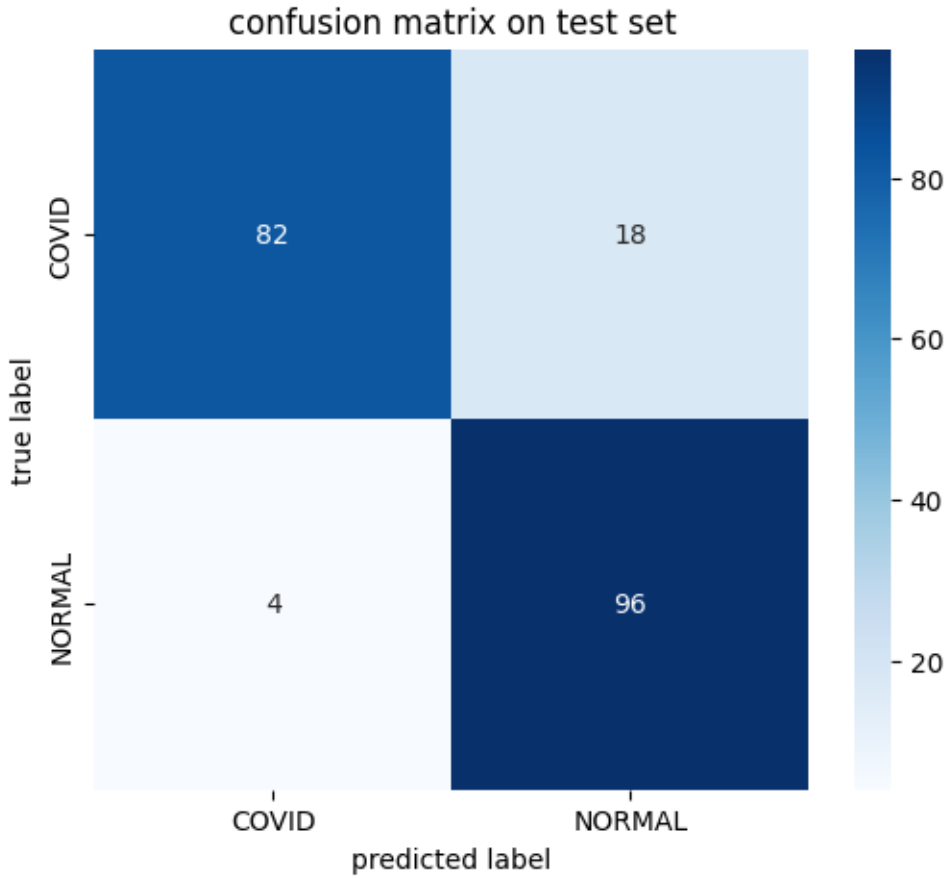
[27]:
```
print("\nevaluating loaded model on test data")
test_results = loaded_model.evaluate(test_data_gen, verbose=1)
print(f"test loss: {test_results[0]}")
print(f"test accuracy: {test_results[1]}")
print(f"test precision: {test_results[2]}")
print(f"test recall: {test_results[3]}")
```

```
evaluating loaded model on test data

/opt/anaconda3/envs/ml-2025/lib/python3.12/site-
packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will be
ignored.
  self._warn_if_super_not_called()

2/2             1s 288ms/step -
accuracy: 0.8772 - loss: 0.5137 - precision: 0.7614 - recall: 0.9614
test loss: 0.4800443947315216
test accuracy: 0.8899999856948853
test precision: 0.8421052694320679
test recall: 0.9599999785423279
```

[28]:
```
print("\ngenerating confusion matrix")
# get predictions (probabilities)
y_pred_prob = loaded_model.predict(test_data_gen)
# convert probabilities to binary predictions
y_pred = (y_pred_prob > 0.5).astype(int).flatten()
# get true labels
y_true = test_data_gen.classes
# calculate confusion matrix
cm = confusion_matrix(y_true, y_pred)
# plot confusion matrix
plt.figure(figsize=(6, 5))
sns.heatmap(
    cm,
    annot=True,
    fmt="d",
    cmap="Blues",
    xticklabels=class_names,
    yticklabels=class_names,
)
plt.xlabel("predicted label")
```

```
plt.ylabel("true label")
plt.title("confusion matrix on test set")
plt.show()
```

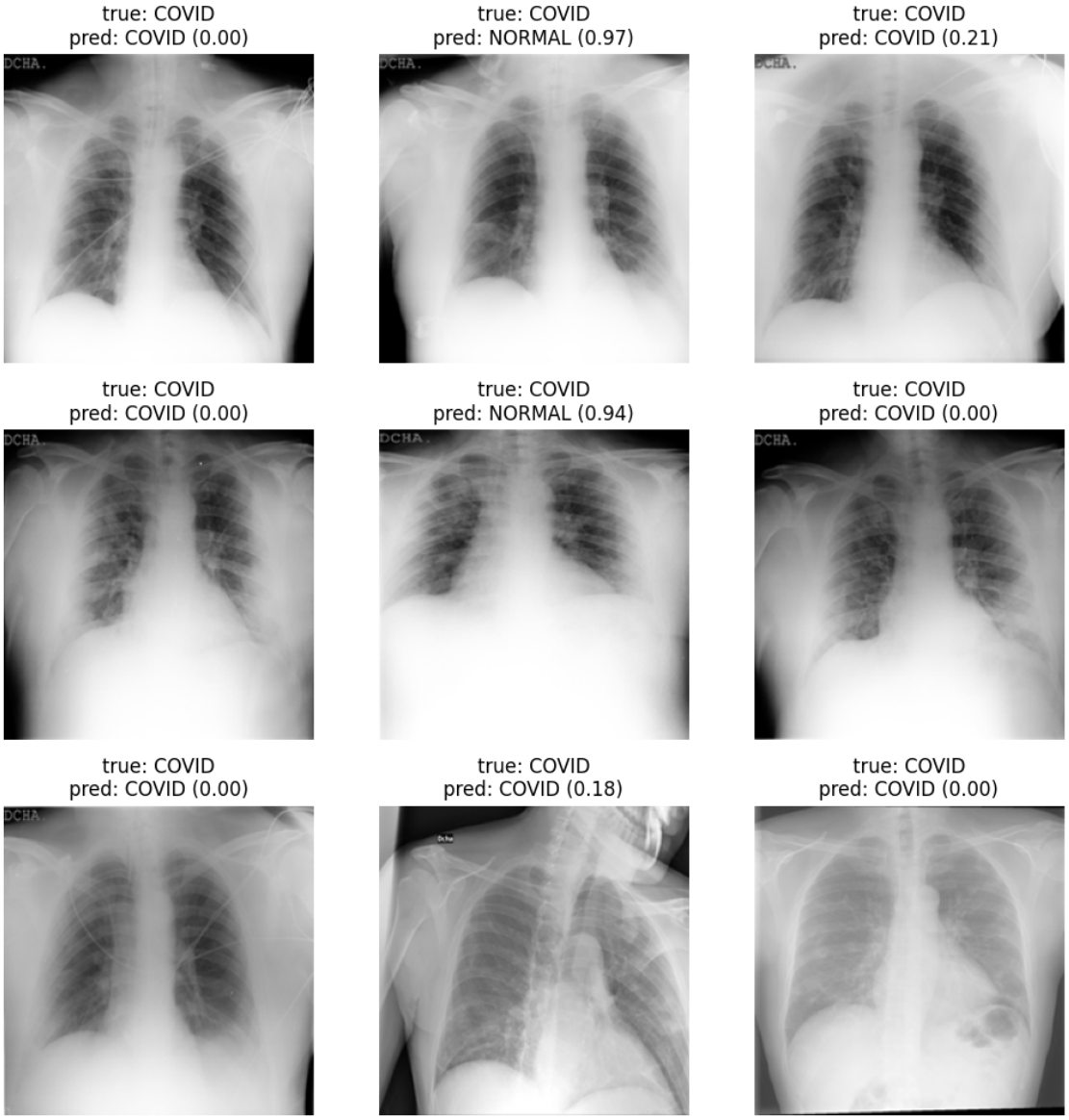
```
generating confusion matrix
2/2             1s 306ms/step
```



[29]:
```
print("\nplotting sample test images with predictions")
# get a batch of raw (unnormalized) images and labels
images_raw, labels_raw = next(iter(test_data_gen_raw))
# get predictions for this batch using the normalized data generator
# important: ensure we use the *same batch* for predictions
# we need to reset the generator to be sure we get the same batch
test_data_gen.reset()
images_norm, _ = next(iter(test_data_gen))  # get normalized images for␣
 ↪prediction
batch_pred_prob = loaded_model.predict(images_norm)
batch_pred = (batch_pred_prob > 0.5).astype(int).flatten()
```

```
plt.figure(figsize=(10, 10))
for i in range(9):  # display 9 samples
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images_raw[i].astype("uint8"))  # display raw image
    true_label = class_names[int(labels_raw[i])]  # cast to int
    pred_label = class_names[batch_pred[i]]
    prob = batch_pred_prob[i][0]
    plt.title(f"true: {true_label}\npred: {pred_label} ({prob:.2f})")
    plt.axis("off")
plt.tight_layout()
plt.show()
```

```
plotting sample test images with predictions
4/4             1s 122ms/step
```


true: COVID
pred: COVID (0.00)
true: COVID
pred: NORMAL (0.97)
true: COVID
pred: COVID (0.21)
true: COVID
pred: COVID (0.00)
true: COVID
pred: NORMAL (0.94)
true: COVID
pred: COVID (0.00)
true: COVID
pred: COVID (0.00)
true: COVID
pred: COVID (0.18)
true: COVID
pred: COVID (0.00)