resnet

April 22, 2025

[1]:
```
import os
import itertools
import random
import numpy as np
import tensorflow as tf
import pandas as pd
import matplotlib.pyplot as plt
from resnet import (
    build_transfer_model,
    train_model,
    fine_tune_model,
    plot_training_history,
    plot_confusion_matrix,
    PROCESSED_DIR,
    IMG_HEIGHT,
    IMG_WIDTH,
    NUM_CHANNELS,
)
import data_pipeline as pipeline
```

[2]:
```
BATCH_SIZE = 32  # smaller batch size for transfer learning
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
```

[3]:
```
# get image paths, mean and std
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)
```

[4]:
```
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.
```

[5]:
```
# get class names
class_names = list(train_data_gen.class_indices.keys())
print(f"class names found: {class_names}")
```

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

[6]:
```
# build and train initial model
input_shape = (IMG_HEIGHT, IMG_WIDTH, NUM_CHANNELS)
model = build_transfer_model(input_shape)
model.summary()
```

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
resnet50v2 (Functional)	(None, 7, 7, 2048)	23,564,800
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 512)	1,049,088
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131,328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 1)	257

Total params: 24,745,473 (94.40 MB)

Trainable params: 1,180,673 (4.50 MB)

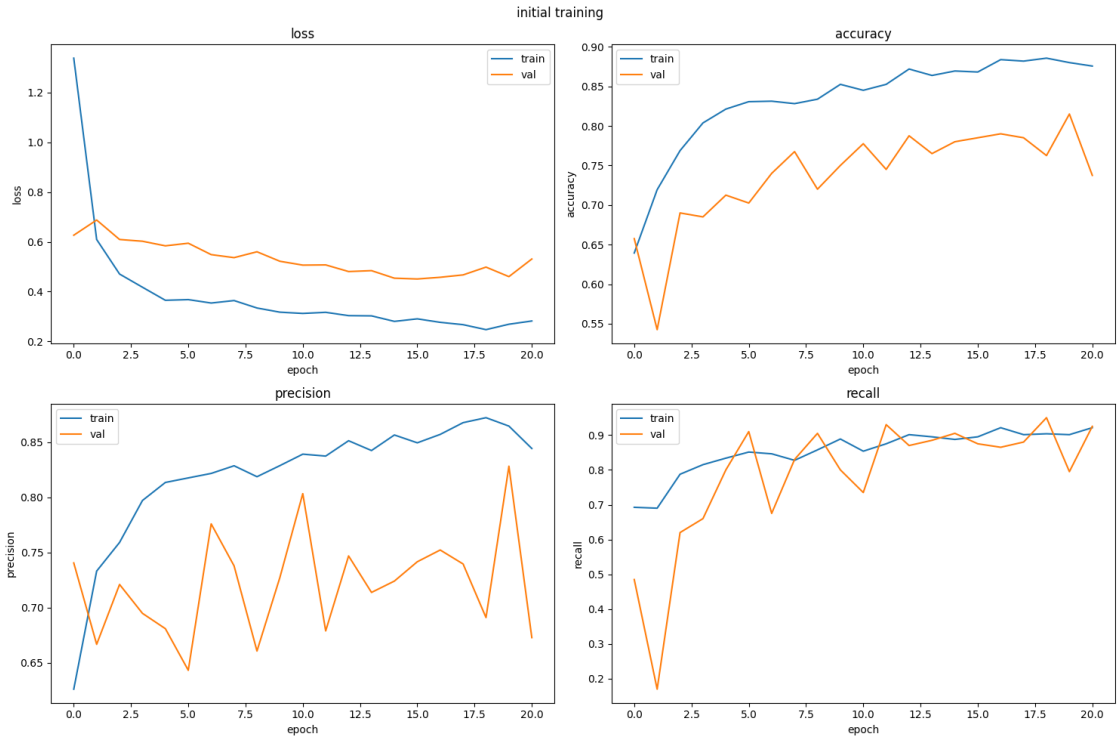
Non-trainable params: 23,564,800 (89.89 MB)

[7]:
```
# train initial model
print("\ntraining initial model")
history = train_model(model, train_data_gen, val_data_gen)
plot_training_history(history, "initial training")
```

```
training initial model
/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             44s 3s/step -
accuracy: 0.5746 - loss: 1.8392 - precision: 0.5420 - recall: 0.5806 -
val_accuracy: 0.6575 - val_loss: 0.6266 - val_precision: 0.7405 - val_recall:
0.4850
Epoch 2/30
13/13             40s 3s/step -
accuracy: 0.7120 - loss: 0.6339 - precision: 0.7419 - recall: 0.6422 -
val_accuracy: 0.5425 - val_loss: 0.6876 - val_precision: 0.6667 - val_recall:
0.1700
Epoch 3/30
13/13             40s 3s/step -
accuracy: 0.7535 - loss: 0.4982 - precision: 0.7486 - recall: 0.7616 -
val_accuracy: 0.6900 - val_loss: 0.6094 - val_precision: 0.7209 - val_recall:
0.6200
Epoch 4/30
13/13             40s 3s/step -
accuracy: 0.7889 - loss: 0.4372 - precision: 0.7774 - recall: 0.8030 -
val_accuracy: 0.6850 - val_loss: 0.6024 - val_precision: 0.6947 - val_recall:
0.6600
Epoch 5/30
13/13             40s 3s/step -
accuracy: 0.8163 - loss: 0.3724 - precision: 0.8126 - recall: 0.8260 -
val_accuracy: 0.7125 - val_loss: 0.5839 - val_precision: 0.6809 - val_recall:
0.8000
Epoch 6/30
13/13             40s 3s/step -
accuracy: 0.8247 - loss: 0.3752 - precision: 0.8081 - recall: 0.8411 -
```

```
val_accuracy: 0.7025 - val_loss: 0.5945 - val_precision: 0.6431 - val_recall:
0.9100
Epoch 7/30
13/13             40s 3s/step -
accuracy: 0.8239 - loss: 0.3705 - precision: 0.8146 - recall: 0.8436 -
val_accuracy: 0.7400 - val_loss: 0.5486 - val_precision: 0.7759 - val_recall:
0.6750
Epoch 8/30
13/13             40s 3s/step -
accuracy: 0.8299 - loss: 0.3608 - precision: 0.8266 - recall: 0.8286 -
val_accuracy: 0.7675 - val_loss: 0.5365 - val_precision: 0.7378 - val_recall:
0.8300
Epoch 9/30
13/13             40s 3s/step -
accuracy: 0.8346 - loss: 0.3314 - precision: 0.8120 - recall: 0.8741 -
val_accuracy: 0.7200 - val_loss: 0.5601 - val_precision: 0.6606 - val_recall:
0.9050
Epoch 10/30
13/13             40s 3s/step -
accuracy: 0.8510 - loss: 0.3338 - precision: 0.8252 - recall: 0.8868 -
val_accuracy: 0.7500 - val_loss: 0.5220 - val_precision: 0.7273 - val_recall:
0.8000
Epoch 11/30
13/13             40s 3s/step -
accuracy: 0.8558 - loss: 0.2989 - precision: 0.8565 - recall: 0.8603 -
val_accuracy: 0.7775 - val_loss: 0.5065 - val_precision: 0.8033 - val_recall:
0.7350
Epoch 12/30
13/13             40s 3s/step -
accuracy: 0.8499 - loss: 0.3146 - precision: 0.8393 - recall: 0.8658 -
val_accuracy: 0.7450 - val_loss: 0.5073 - val_precision: 0.6788 - val_recall:
0.9300
Epoch 13/30
13/13             40s 3s/step -
accuracy: 0.8618 - loss: 0.3188 - precision: 0.8311 - recall: 0.9065 -
val_accuracy: 0.7875 - val_loss: 0.4807 - val_precision: 0.7468 - val_recall:
0.8700
Epoch 14/30
13/13             40s 3s/step -
accuracy: 0.8726 - loss: 0.2979 - precision: 0.8530 - recall: 0.9047 -
val_accuracy: 0.7650 - val_loss: 0.4843 - val_precision: 0.7137 - val_recall:
0.8850
Epoch 15/30
13/13             40s 3s/step -
accuracy: 0.8695 - loss: 0.2832 - precision: 0.8572 - recall: 0.8805 -
val_accuracy: 0.7800 - val_loss: 0.4539 - val_precision: 0.7240 - val_recall:
0.9050
Epoch 16/30
```

```
13/13            40s 3s/step -
accuracy: 0.8630 - loss: 0.3054 - precision: 0.8323 - recall: 0.9100 -
val_accuracy: 0.7850 - val_loss: 0.4508 - val_precision: 0.7415 - val_recall:
0.8750
Epoch 17/30
13/13            40s 3s/step -
accuracy: 0.8922 - loss: 0.2716 - precision: 0.8699 - recall: 0.9331 -
val_accuracy: 0.7900 - val_loss: 0.4576 - val_precision: 0.7522 - val_recall:
0.8650
Epoch 18/30
13/13            40s 3s/step -
accuracy: 0.8738 - loss: 0.2698 - precision: 0.8630 - recall: 0.8836 -
val_accuracy: 0.7850 - val_loss: 0.4671 - val_precision: 0.7395 - val_recall:
0.8800
Epoch 19/30
13/13            40s 3s/step -
accuracy: 0.8857 - loss: 0.2576 - precision: 0.8658 - recall: 0.9031 -
val_accuracy: 0.7625 - val_loss: 0.4985 - val_precision: 0.6909 - val_recall:
0.9500
Epoch 20/30
13/13            40s 3s/step -
accuracy: 0.8824 - loss: 0.2645 - precision: 0.8571 - recall: 0.9218 -
val_accuracy: 0.8150 - val_loss: 0.4602 - val_precision: 0.8281 - val_recall:
0.7950
Epoch 21/30
13/13            40s 3s/step -
accuracy: 0.8757 - loss: 0.2740 - precision: 0.8584 - recall: 0.9131 -
val_accuracy: 0.7375 - val_loss: 0.5310 - val_precision: 0.6727 - val_recall:
0.9250
```



[8]:
```
model_save_path = "../models/initial_resnet_model.keras"
print(f"\nsaving final model to {model_save_path}")
model.save(model_save_path)
```

```
saving final model to ../models/initial_resnet_model.keras
```

[9]:
```
# hyperparams
batch_sizes = [16, 32, 64]
learning_rates = [0.001, 0.0001, 0.00001]
dropout_rates = [0.2, 0.3, 0.4]

# param grid
param_grid = list(itertools.product(batch_sizes, learning_rates, dropout_rates))

# define number of combinations to randomly sample
num_combinations_to_test = 5

# 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 5 combinations from 27 total.

[10]:
```
# hyperparameter tuning
print("\nperforming hyperparameter tuning")

results = []

for batch_size, learning_rate, dropout_rate in sampled_params:
    print(
        f"training with batch_size={batch_size}, learning_rate={learning_rate},␣
 ↪dropout_rate={dropout_rate}"
    )

    # update batch size in generators
    train_data_gen.batch_size = batch_size
    val_data_gen.batch_size = batch_size

    # build and train model
    model = build_transfer_model(input_shape, dropout_rate=dropout_rate)
    history = train_model(
        model, train_data_gen, val_data_gen, learning_rate=learning_rate
    )

    # get best validation metrics
    best_val_loss = min(history.history["val_loss"])
    best_val_acc = max(history.history["val_accuracy"])
    best_val_precision = max(history.history["val_precision"])
    best_val_recall = max(history.history["val_recall"])

    results.append(
        {
            "batch_size": batch_size,
            "learning_rate": learning_rate,
            "dropout_rate": dropout_rate,
            "val_loss": best_val_loss,
            "val_accuracy": best_val_acc,
            "val_precision": best_val_precision,
            "val_recall": best_val_recall,
        }
    )

# save results to csv
os.makedirs("../results", exist_ok=True)
results_df = pd.DataFrame(results)
results_df.to_csv("../results/transfer_learning_hyperparameter_tuning.csv",␣
 ↪index=False)
print("\nhyperparameter tuning results:")
```

```
print(results_df)
```

```
performing hyperparameter tuning
training with batch_size=64, learning_rate=1e-05, dropout_rate=0.2
Epoch 1/30
25/25             46s 2s/step -
accuracy: 0.6027 - loss: 0.9223 - precision: 0.5941 - recall: 0.6153 -
val_accuracy: 0.5900 - val_loss: 0.7056 - val_precision: 0.5526 - val_recall:
0.9450
Epoch 2/30
25/25             41s 2s/step -
accuracy: 0.6149 - loss: 0.7549 - precision: 0.6052 - recall: 0.7115 -
val_accuracy: 0.6725 - val_loss: 0.6292 - val_precision: 0.6342 - val_recall:
0.8150
Epoch 3/30
25/25             42s 2s/step -
accuracy: 0.6776 - loss: 0.6617 - precision: 0.6797 - recall: 0.6827 -
val_accuracy: 0.7000 - val_loss: 0.6072 - val_precision: 0.6667 - val_recall:
0.8000
Epoch 4/30
25/25             41s 2s/step -
accuracy: 0.7293 - loss: 0.5873 - precision: 0.7330 - recall: 0.7266 -
val_accuracy: 0.7025 - val_loss: 0.5994 - val_precision: 0.6653 - val_recall:
0.8150
Epoch 5/30
25/25             41s 2s/step -
accuracy: 0.7440 - loss: 0.5574 - precision: 0.7423 - recall: 0.7283 -
val_accuracy: 0.6925 - val_loss: 0.5942 - val_precision: 0.6611 - val_recall:
0.7900
Epoch 6/30
25/25             41s 2s/step -
accuracy: 0.7308 - loss: 0.5736 - precision: 0.7234 - recall: 0.7541 -
val_accuracy: 0.6925 - val_loss: 0.5946 - val_precision: 0.6624 - val_recall:
0.7850
Epoch 7/30
25/25             41s 2s/step -
accuracy: 0.7159 - loss: 0.5836 - precision: 0.6979 - recall: 0.7342 -
val_accuracy: 0.7125 - val_loss: 0.5866 - val_precision: 0.6923 - val_recall:
0.7650
Epoch 8/30
25/25             41s 2s/step -
accuracy: 0.7366 - loss: 0.5398 - precision: 0.7320 - recall: 0.7271 -
val_accuracy: 0.7050 - val_loss: 0.5862 - val_precision: 0.6830 - val_recall:
0.7650
Epoch 9/30
25/25             41s 2s/step -
accuracy: 0.7587 - loss: 0.4975 - precision: 0.7554 - recall: 0.7327 -
```

```
val_accuracy: 0.6900 - val_loss: 0.5885 - val_precision: 0.6583 - val_recall:
0.7900
Epoch 10/30
25/25             41s 2s/step -
accuracy: 0.7558 - loss: 0.5159 - precision: 0.7478 - recall: 0.7614 -
val_accuracy: 0.6925 - val_loss: 0.5820 - val_precision: 0.6624 - val_recall:
0.7850
Epoch 11/30
25/25             41s 2s/step -
accuracy: 0.7685 - loss: 0.4947 - precision: 0.7461 - recall: 0.7842 -
val_accuracy: 0.7075 - val_loss: 0.5738 - val_precision: 0.6895 - val_recall:
0.7550
Epoch 12/30
25/25             41s 2s/step -
accuracy: 0.7776 - loss: 0.4626 - precision: 0.7653 - recall: 0.7916 -
val_accuracy: 0.7200 - val_loss: 0.5725 - val_precision: 0.7056 - val_recall:
0.7550
Epoch 13/30
25/25             41s 2s/step -
accuracy: 0.7785 - loss: 0.4639 - precision: 0.7873 - recall: 0.7751 -
val_accuracy: 0.7150 - val_loss: 0.5748 - val_precision: 0.6991 - val_recall:
0.7550
Epoch 14/30
25/25             41s 2s/step -
accuracy: 0.7770 - loss: 0.4708 - precision: 0.7758 - recall: 0.7693 -
val_accuracy: 0.7000 - val_loss: 0.5753 - val_precision: 0.6770 - val_recall:
0.7650
Epoch 15/30
25/25             41s 2s/step -
accuracy: 0.7706 - loss: 0.4779 - precision: 0.7612 - recall: 0.7845 -
val_accuracy: 0.7225 - val_loss: 0.5717 - val_precision: 0.7129 - val_recall:
0.7450
Epoch 16/30
25/25             41s 2s/step -
accuracy: 0.7802 - loss: 0.4386 - precision: 0.7651 - recall: 0.7883 -
val_accuracy: 0.7250 - val_loss: 0.5663 - val_precision: 0.7163 - val_recall:
0.7450
Epoch 17/30
25/25             41s 2s/step -
accuracy: 0.8053 - loss: 0.4158 - precision: 0.8010 - recall: 0.8115 -
val_accuracy: 0.7275 - val_loss: 0.5630 - val_precision: 0.7198 - val_recall:
0.7450
Epoch 18/30
25/25             41s 2s/step -
accuracy: 0.8040 - loss: 0.4270 - precision: 0.8071 - recall: 0.8059 -
val_accuracy: 0.7300 - val_loss: 0.5655 - val_precision: 0.7130 - val_recall:
0.7700
Epoch 19/30
```

```
25/25            42s 2s/step -
accuracy: 0.7845 - loss: 0.4395 - precision: 0.7777 - recall: 0.7975 -
val_accuracy: 0.7200 - val_loss: 0.5638 - val_precision: 0.6982 - val_recall:
0.7750
Epoch 20/30
25/25            42s 2s/step -
accuracy: 0.7929 - loss: 0.4184 - precision: 0.7969 - recall: 0.8103 -
val_accuracy: 0.7225 - val_loss: 0.5586 - val_precision: 0.7109 - val_recall:
0.7500
Epoch 21/30
25/25            42s 2s/step -
accuracy: 0.7970 - loss: 0.4245 - precision: 0.7858 - recall: 0.7898 -
val_accuracy: 0.7200 - val_loss: 0.5599 - val_precision: 0.6947 - val_recall:
0.7850
Epoch 22/30
25/25            42s 2s/step -
accuracy: 0.8171 - loss: 0.3994 - precision: 0.7962 - recall: 0.8442 -
val_accuracy: 0.7375 - val_loss: 0.5530 - val_precision: 0.7209 - val_recall:
0.7750
Epoch 23/30
25/25            41s 2s/step -
accuracy: 0.7813 - loss: 0.4379 - precision: 0.7727 - recall: 0.7843 -
val_accuracy: 0.7375 - val_loss: 0.5509 - val_precision: 0.7230 - val_recall:
0.7700
Epoch 24/30
25/25            41s 2s/step -
accuracy: 0.8199 - loss: 0.3862 - precision: 0.8197 - recall: 0.8233 -
val_accuracy: 0.7400 - val_loss: 0.5513 - val_precision: 0.7143 - val_recall:
0.8000
Epoch 25/30
25/25            41s 2s/step -
accuracy: 0.8258 - loss: 0.3926 - precision: 0.8146 - recall: 0.8433 -
val_accuracy: 0.7375 - val_loss: 0.5476 - val_precision: 0.7251 - val_recall:
0.7650
Epoch 26/30
25/25            41s 2s/step -
accuracy: 0.8109 - loss: 0.4191 - precision: 0.8081 - recall: 0.8089 -
val_accuracy: 0.7400 - val_loss: 0.5475 - val_precision: 0.7124 - val_recall:
0.8050
Epoch 27/30
25/25            42s 2s/step -
accuracy: 0.8056 - loss: 0.3952 - precision: 0.7946 - recall: 0.8295 -
val_accuracy: 0.7450 - val_loss: 0.5407 - val_precision: 0.7290 - val_recall:
0.7800
Epoch 28/30
25/25            41s 2s/step -
accuracy: 0.8330 - loss: 0.3761 - precision: 0.8330 - recall: 0.8281 -
val_accuracy: 0.7325 - val_loss: 0.5359 - val_precision: 0.7123 - val_recall:
```

```
0.7800
Epoch 29/30
25/25             41s 2s/step -
accuracy: 0.8002 - loss: 0.4161 - precision: 0.7812 - recall: 0.8153 -
val_accuracy: 0.7450 - val_loss: 0.5369 - val_precision: 0.7112 - val_recall:
0.8250
Epoch 30/30
25/25             41s 2s/step -
accuracy: 0.8294 - loss: 0.3740 - precision: 0.8163 - recall: 0.8568 -
val_accuracy: 0.7475 - val_loss: 0.5311 - val_precision: 0.7368 - val_recall:
0.7700
training with batch_size=32, learning_rate=1e-05, dropout_rate=0.3
Epoch 1/30
50/50             45s 840ms/step -
accuracy: 0.6525 - loss: 0.9082 - precision: 0.6616 - recall: 0.6166 -
val_accuracy: 0.6450 - val_loss: 0.7077 - val_precision: 0.6058 - val_recall:
0.8300
Epoch 2/30
50/50             41s 813ms/step -
accuracy: 0.6721 - loss: 0.7559 - precision: 0.6362 - recall: 0.6919 -
val_accuracy: 0.6600 - val_loss: 0.6804 - val_precision: 0.6240 - val_recall:
0.8050
Epoch 3/30
50/50             41s 828ms/step -
accuracy: 0.6935 - loss: 0.6813 - precision: 0.6879 - recall: 0.7098 -
val_accuracy: 0.6700 - val_loss: 0.6622 - val_precision: 0.6417 - val_recall:
0.7700
Epoch 4/30
50/50             41s 826ms/step -
accuracy: 0.6985 - loss: 0.6587 - precision: 0.6940 - recall: 0.7064 -
val_accuracy: 0.6575 - val_loss: 0.6520 - val_precision: 0.6352 - val_recall:
0.7400
Epoch 5/30
50/50             41s 830ms/step -
accuracy: 0.7439 - loss: 0.5373 - precision: 0.7424 - recall: 0.7510 -
val_accuracy: 0.6850 - val_loss: 0.6395 - val_precision: 0.6832 - val_recall:
0.6900
Epoch 6/30
50/50             41s 820ms/step -
accuracy: 0.7533 - loss: 0.5558 - precision: 0.7721 - recall: 0.7539 -
val_accuracy: 0.6750 - val_loss: 0.6354 - val_precision: 0.6699 - val_recall:
0.6900
Epoch 7/30
50/50             41s 823ms/step -
accuracy: 0.7632 - loss: 0.5134 - precision: 0.7598 - recall: 0.7536 -
val_accuracy: 0.6750 - val_loss: 0.6302 - val_precision: 0.6716 - val_recall:
0.6850
Epoch 8/30
```

```
50/50            41s 822ms/step -
accuracy: 0.7540 - loss: 0.5294 - precision: 0.7631 - recall: 0.7168 -
val_accuracy: 0.6750 - val_loss: 0.6312 - val_precision: 0.6549 - val_recall:
0.7400
Epoch 9/30
50/50            41s 822ms/step -
accuracy: 0.7711 - loss: 0.5066 - precision: 0.7722 - recall: 0.7879 -
val_accuracy: 0.6800 - val_loss: 0.6210 - val_precision: 0.6800 - val_recall:
0.6800
Epoch 10/30
50/50            41s 822ms/step -
accuracy: 0.7697 - loss: 0.5075 - precision: 0.7994 - recall: 0.7464 -
val_accuracy: 0.6850 - val_loss: 0.6132 - val_precision: 0.6888 - val_recall:
0.6750
Epoch 11/30
50/50            41s 819ms/step -
accuracy: 0.7801 - loss: 0.4798 - precision: 0.7918 - recall: 0.7606 -
val_accuracy: 0.6975 - val_loss: 0.6032 - val_precision: 0.6854 - val_recall:
0.7300
Epoch 12/30
50/50            42s 832ms/step -
accuracy: 0.7834 - loss: 0.4743 - precision: 0.7648 - recall: 0.7871 -
val_accuracy: 0.6875 - val_loss: 0.5998 - val_precision: 0.6829 - val_recall:
0.7000
Epoch 13/30
50/50            41s 825ms/step -
accuracy: 0.7939 - loss: 0.4779 - precision: 0.7906 - recall: 0.8031 -
val_accuracy: 0.7000 - val_loss: 0.5957 - val_precision: 0.6869 - val_recall:
0.7350
Epoch 14/30
50/50            41s 829ms/step -
accuracy: 0.8012 - loss: 0.4600 - precision: 0.8067 - recall: 0.7822 -
val_accuracy: 0.7200 - val_loss: 0.5942 - val_precision: 0.6982 - val_recall:
0.7750
Epoch 15/30
50/50            41s 823ms/step -
accuracy: 0.8052 - loss: 0.4442 - precision: 0.7989 - recall: 0.8004 -
val_accuracy: 0.7050 - val_loss: 0.5940 - val_precision: 0.6916 - val_recall:
0.7400
Epoch 16/30
50/50            41s 828ms/step -
accuracy: 0.8124 - loss: 0.4495 - precision: 0.8366 - recall: 0.8122 -
val_accuracy: 0.7050 - val_loss: 0.5929 - val_precision: 0.7030 - val_recall:
0.7100
Epoch 17/30
50/50            41s 819ms/step -
accuracy: 0.7992 - loss: 0.4258 - precision: 0.8015 - recall: 0.7933 -
val_accuracy: 0.7050 - val_loss: 0.5917 - val_precision: 0.6952 - val_recall:
```

```
0.7300
Epoch 18/30
50/50             41s 823ms/step -
accuracy: 0.7931 - loss: 0.4287 - precision: 0.7917 - recall: 0.7893 -
val_accuracy: 0.7050 - val_loss: 0.5855 - val_precision: 0.6864 - val_recall:
0.7550
Epoch 19/30
50/50             41s 819ms/step -
accuracy: 0.7795 - loss: 0.4543 - precision: 0.7606 - recall: 0.8107 -
val_accuracy: 0.7125 - val_loss: 0.5845 - val_precision: 0.6977 - val_recall:
0.7500
Epoch 20/30
50/50             42s 832ms/step -
accuracy: 0.8111 - loss: 0.3904 - precision: 0.8029 - recall: 0.8282 -
val_accuracy: 0.7150 - val_loss: 0.5822 - val_precision: 0.7172 - val_recall:
0.7100
Epoch 21/30
50/50             41s 824ms/step -
accuracy: 0.8154 - loss: 0.4093 - precision: 0.8328 - recall: 0.7891 -
val_accuracy: 0.7100 - val_loss: 0.5800 - val_precision: 0.7000 - val_recall:
0.7350
Epoch 22/30
50/50             41s 830ms/step -
accuracy: 0.8184 - loss: 0.3951 - precision: 0.8213 - recall: 0.8219 -
val_accuracy: 0.7250 - val_loss: 0.5696 - val_precision: 0.7009 - val_recall:
0.7850
Epoch 23/30
50/50             41s 824ms/step -
accuracy: 0.8168 - loss: 0.4198 - precision: 0.8063 - recall: 0.8327 -
val_accuracy: 0.7225 - val_loss: 0.5628 - val_precision: 0.7051 - val_recall:
0.7650
Epoch 24/30
50/50             42s 832ms/step -
accuracy: 0.8079 - loss: 0.4247 - precision: 0.8031 - recall: 0.8023 -
val_accuracy: 0.7250 - val_loss: 0.5630 - val_precision: 0.6974 - val_recall:
0.7950
Epoch 25/30
50/50             41s 830ms/step -
accuracy: 0.8230 - loss: 0.4007 - precision: 0.8032 - recall: 0.8271 -
val_accuracy: 0.7225 - val_loss: 0.5744 - val_precision: 0.6894 - val_recall:
0.8100
Epoch 26/30
50/50             42s 835ms/step -
accuracy: 0.8026 - loss: 0.4289 - precision: 0.7979 - recall: 0.8243 -
val_accuracy: 0.7175 - val_loss: 0.5669 - val_precision: 0.6986 - val_recall:
0.7650
Epoch 27/30
50/50             42s 832ms/step -
```

```
accuracy: 0.8359 - loss: 0.3709 - precision: 0.8337 - recall: 0.8373 -
val_accuracy: 0.7125 - val_loss: 0.5643 - val_precision: 0.6889 - val_recall:
0.7750
Epoch 28/30
50/50             41s 827ms/step -
accuracy: 0.8374 - loss: 0.3646 - precision: 0.8285 - recall: 0.8574 -
val_accuracy: 0.7125 - val_loss: 0.5658 - val_precision: 0.7014 - val_recall:
0.7400
training with batch_size=64, learning_rate=1e-05, dropout_rate=0.4
Epoch 1/30
25/25             44s 2s/step -
accuracy: 0.5705 - loss: 1.2596 - precision: 0.5766 - recall: 0.5516 -
val_accuracy: 0.5425 - val_loss: 0.7211 - val_precision: 0.5245 - val_recall:
0.9100
Epoch 2/30
25/25             41s 2s/step -
accuracy: 0.5398 - loss: 1.0139 - precision: 0.5275 - recall: 0.6244 -
val_accuracy: 0.6125 - val_loss: 0.7094 - val_precision: 0.5738 - val_recall:
0.8750
Epoch 3/30
25/25             41s 2s/step -
accuracy: 0.5900 - loss: 0.9051 - precision: 0.5896 - recall: 0.6268 -
val_accuracy: 0.6725 - val_loss: 0.6972 - val_precision: 0.6245 - val_recall:
0.8650
Epoch 4/30
25/25             41s 2s/step -
accuracy: 0.6265 - loss: 0.8613 - precision: 0.6099 - recall: 0.6406 -
val_accuracy: 0.6950 - val_loss: 0.6923 - val_precision: 0.6535 - val_recall:
0.8300
Epoch 5/30
25/25             41s 2s/step -
accuracy: 0.6650 - loss: 0.8157 - precision: 0.6580 - recall: 0.6804 -
val_accuracy: 0.6925 - val_loss: 0.6587 - val_precision: 0.6758 - val_recall:
0.7400
Epoch 6/30
25/25             41s 2s/step -
accuracy: 0.6754 - loss: 0.7310 - precision: 0.6779 - recall: 0.6645 -
val_accuracy: 0.7000 - val_loss: 0.6637 - val_precision: 0.6754 - val_recall:
0.7700
Epoch 7/30
25/25             41s 2s/step -
accuracy: 0.6722 - loss: 0.7459 - precision: 0.6717 - recall: 0.6640 -
val_accuracy: 0.6900 - val_loss: 0.6519 - val_precision: 0.6759 - val_recall:
0.7300
Epoch 8/30
25/25             41s 2s/step -
accuracy: 0.7194 - loss: 0.6489 - precision: 0.7200 - recall: 0.6928 -
val_accuracy: 0.6975 - val_loss: 0.6414 - val_precision: 0.6872 - val_recall:
```

```
0.7250
Epoch 9/30
25/25             41s 2s/step -
accuracy: 0.6853 - loss: 0.6609 - precision: 0.6988 - recall: 0.6629 -
val_accuracy: 0.7150 - val_loss: 0.6398 - val_precision: 0.6955 - val_recall:
0.7650
Epoch 10/30
25/25             41s 2s/step -
accuracy: 0.6903 - loss: 0.7003 - precision: 0.6832 - recall: 0.6665 -
val_accuracy: 0.7175 - val_loss: 0.6362 - val_precision: 0.6968 - val_recall:
0.7700
Epoch 11/30
25/25             41s 2s/step -
accuracy: 0.7056 - loss: 0.6371 - precision: 0.7102 - recall: 0.6994 -
val_accuracy: 0.7100 - val_loss: 0.6208 - val_precision: 0.7000 - val_recall:
0.7350
Epoch 12/30
25/25             41s 2s/step -
accuracy: 0.6967 - loss: 0.6466 - precision: 0.7262 - recall: 0.6546 -
val_accuracy: 0.7150 - val_loss: 0.6220 - val_precision: 0.6972 - val_recall:
0.7600
Epoch 13/30
25/25             41s 2s/step -
accuracy: 0.7202 - loss: 0.6048 - precision: 0.7183 - recall: 0.7223 -
val_accuracy: 0.7225 - val_loss: 0.6142 - val_precision: 0.7032 - val_recall:
0.7700
Epoch 14/30
25/25             41s 2s/step -
accuracy: 0.7166 - loss: 0.6168 - precision: 0.7141 - recall: 0.6747 -
val_accuracy: 0.7200 - val_loss: 0.6223 - val_precision: 0.6930 - val_recall:
0.7900
Epoch 15/30
25/25             41s 2s/step -
accuracy: 0.7337 - loss: 0.6058 - precision: 0.7457 - recall: 0.7016 -
val_accuracy: 0.7250 - val_loss: 0.6331 - val_precision: 0.6875 - val_recall:
0.8250
Epoch 16/30
25/25             41s 2s/step -
accuracy: 0.7427 - loss: 0.5732 - precision: 0.7324 - recall: 0.7690 -
val_accuracy: 0.7200 - val_loss: 0.6170 - val_precision: 0.6930 - val_recall:
0.7900
Epoch 17/30
25/25             41s 2s/step -
accuracy: 0.7445 - loss: 0.5458 - precision: 0.7661 - recall: 0.7174 -
val_accuracy: 0.7275 - val_loss: 0.6088 - val_precision: 0.7059 - val_recall:
0.7800
Epoch 18/30
25/25             41s 2s/step -
```

```
accuracy: 0.7448 - loss: 0.5565 - precision: 0.7599 - recall: 0.7101 -
val_accuracy: 0.7225 - val_loss: 0.5945 - val_precision: 0.7032 - val_recall:
0.7700
Epoch 19/30
25/25             41s 2s/step -
accuracy: 0.7485 - loss: 0.5710 - precision: 0.7640 - recall: 0.7302 -
val_accuracy: 0.7175 - val_loss: 0.5902 - val_precision: 0.6968 - val_recall:
0.7700
Epoch 20/30
25/25             41s 2s/step -
accuracy: 0.7505 - loss: 0.5184 - precision: 0.7460 - recall: 0.7218 -
val_accuracy: 0.7225 - val_loss: 0.5945 - val_precision: 0.6926 - val_recall:
0.8000
Epoch 21/30
25/25             42s 2s/step -
accuracy: 0.7648 - loss: 0.5251 - precision: 0.7664 - recall: 0.7549 -
val_accuracy: 0.7175 - val_loss: 0.5948 - val_precision: 0.6883 - val_recall:
0.7950
Epoch 22/30
25/25             41s 2s/step -
accuracy: 0.7623 - loss: 0.5294 - precision: 0.7819 - recall: 0.7599 -
val_accuracy: 0.7225 - val_loss: 0.5897 - val_precision: 0.6996 - val_recall:
0.7800
Epoch 23/30
25/25             41s 2s/step -
accuracy: 0.7704 - loss: 0.4967 - precision: 0.7869 - recall: 0.7535 -
val_accuracy: 0.7125 - val_loss: 0.5842 - val_precision: 0.7033 - val_recall:
0.7350
Epoch 24/30
25/25             41s 2s/step -
accuracy: 0.7432 - loss: 0.5461 - precision: 0.7555 - recall: 0.7239 -
val_accuracy: 0.7125 - val_loss: 0.5815 - val_precision: 0.6941 - val_recall:
0.7600
Epoch 25/30
25/25             41s 2s/step -
accuracy: 0.7647 - loss: 0.5223 - precision: 0.7655 - recall: 0.7528 -
val_accuracy: 0.7125 - val_loss: 0.5808 - val_precision: 0.6959 - val_recall:
0.7550
Epoch 26/30
25/25             41s 2s/step -
accuracy: 0.7660 - loss: 0.4932 - precision: 0.7636 - recall: 0.7598 -
val_accuracy: 0.7150 - val_loss: 0.5792 - val_precision: 0.6972 - val_recall:
0.7600
Epoch 27/30
25/25             41s 2s/step -
accuracy: 0.7721 - loss: 0.4901 - precision: 0.7749 - recall: 0.7645 -
val_accuracy: 0.7100 - val_loss: 0.5761 - val_precision: 0.6892 - val_recall:
0.7650
```

```
Epoch 28/30
25/25             41s 2s/step -
accuracy: 0.7815 - loss: 0.4752 - precision: 0.7784 - recall: 0.7633 -
val_accuracy: 0.7200 - val_loss: 0.5704 - val_precision: 0.6947 - val_recall:
0.7850
Epoch 29/30
25/25             41s 2s/step -
accuracy: 0.7597 - loss: 0.4708 - precision: 0.7539 - recall: 0.7488 -
val_accuracy: 0.7225 - val_loss: 0.5692 - val_precision: 0.6960 - val_recall:
0.7900
Epoch 30/30
25/25             41s 2s/step -
accuracy: 0.7935 - loss: 0.4663 - precision: 0.7949 - recall: 0.7875 -
val_accuracy: 0.7175 - val_loss: 0.5680 - val_precision: 0.6916 - val_recall:
0.7850
training with batch_size=32, learning_rate=0.0001, dropout_rate=0.3
Epoch 1/30
50/50             45s 842ms/step -
accuracy: 0.6883 - loss: 0.7471 - precision: 0.6648 - recall: 0.7104 -
val_accuracy: 0.6950 - val_loss: 0.6174 - val_precision: 0.6711 - val_recall:
0.7650
Epoch 2/30
50/50             41s 830ms/step -
accuracy: 0.7583 - loss: 0.5230 - precision: 0.7423 - recall: 0.7739 -
val_accuracy: 0.6975 - val_loss: 0.6502 - val_precision: 0.6348 - val_recall:
0.9300
Epoch 3/30
50/50             41s 825ms/step -
accuracy: 0.7810 - loss: 0.4645 - precision: 0.7656 - recall: 0.8311 -
val_accuracy: 0.7475 - val_loss: 0.5433 - val_precision: 0.7037 - val_recall:
0.8550
Epoch 4/30
50/50             41s 830ms/step -
accuracy: 0.8210 - loss: 0.4157 - precision: 0.8168 - recall: 0.8538 -
val_accuracy: 0.7600 - val_loss: 0.5595 - val_precision: 0.7261 - val_recall:
0.8350
Epoch 5/30
50/50             41s 825ms/step -
accuracy: 0.8065 - loss: 0.4498 - precision: 0.8073 - recall: 0.8225 -
val_accuracy: 0.6575 - val_loss: 0.6029 - val_precision: 0.7838 - val_recall:
0.4350
Epoch 6/30
50/50             41s 830ms/step -
accuracy: 0.8145 - loss: 0.3920 - precision: 0.8096 - recall: 0.8076 -
val_accuracy: 0.7200 - val_loss: 0.6051 - val_precision: 0.6571 - val_recall:
0.9200
Epoch 7/30
50/50             41s 828ms/step -
```

```
accuracy: 0.8453 - loss: 0.3553 - precision: 0.8253 - recall: 0.8797 -
val_accuracy: 0.7550 - val_loss: 0.4994 - val_precision: 0.7125 - val_recall:
0.8550
Epoch 8/30
50/50            41s 821ms/step -
accuracy: 0.8547 - loss: 0.3413 - precision: 0.8453 - recall: 0.8708 -
val_accuracy: 0.7250 - val_loss: 0.5477 - val_precision: 0.6718 - val_recall:
0.8800
Epoch 9/30
50/50            41s 819ms/step -
accuracy: 0.8556 - loss: 0.3183 - precision: 0.8473 - recall: 0.8651 -
val_accuracy: 0.7750 - val_loss: 0.4795 - val_precision: 0.7292 - val_recall:
0.8750
Epoch 10/30
50/50            41s 824ms/step -
accuracy: 0.8536 - loss: 0.3520 - precision: 0.8458 - recall: 0.8669 -
val_accuracy: 0.7825 - val_loss: 0.4870 - val_precision: 0.7534 - val_recall:
0.8400
Epoch 11/30
50/50            41s 825ms/step -
accuracy: 0.8463 - loss: 0.3266 - precision: 0.8297 - recall: 0.8598 -
val_accuracy: 0.7800 - val_loss: 0.4745 - val_precision: 0.7917 - val_recall:
0.7600
Epoch 12/30
50/50            41s 826ms/step -
accuracy: 0.8781 - loss: 0.2925 - precision: 0.8654 - recall: 0.9057 -
val_accuracy: 0.7550 - val_loss: 0.4999 - val_precision: 0.6932 - val_recall:
0.9150
Epoch 13/30
50/50            41s 829ms/step -
accuracy: 0.8593 - loss: 0.3107 - precision: 0.8363 - recall: 0.8881 -
val_accuracy: 0.7800 - val_loss: 0.4649 - val_precision: 0.7545 - val_recall:
0.8300
Epoch 14/30
50/50            41s 825ms/step -
accuracy: 0.8650 - loss: 0.2893 - precision: 0.8544 - recall: 0.8824 -
val_accuracy: 0.7300 - val_loss: 0.5311 - val_precision: 0.6631 - val_recall:
0.9350
Epoch 15/30
50/50            41s 820ms/step -
accuracy: 0.8714 - loss: 0.2932 - precision: 0.8532 - recall: 0.9016 -
val_accuracy: 0.7850 - val_loss: 0.4838 - val_precision: 0.7938 - val_recall:
0.7700
Epoch 16/30
50/50            41s 822ms/step -
accuracy: 0.8744 - loss: 0.2796 - precision: 0.8696 - recall: 0.8838 -
val_accuracy: 0.7925 - val_loss: 0.4584 - val_precision: 0.7555 - val_recall:
0.8650
```

```
Epoch 17/30
50/50            41s 821ms/step -
accuracy: 0.8708 - loss: 0.2845 - precision: 0.8623 - recall: 0.8941 -
val_accuracy: 0.7925 - val_loss: 0.4463 - val_precision: 0.7532 - val_recall:
0.8700
Epoch 18/30
50/50            41s 823ms/step -
accuracy: 0.8726 - loss: 0.2847 - precision: 0.8653 - recall: 0.8812 -
val_accuracy: 0.7225 - val_loss: 0.5818 - val_precision: 0.6540 - val_recall:
0.9450
Epoch 19/30
50/50            41s 823ms/step -
accuracy: 0.8698 - loss: 0.2949 - precision: 0.8553 - recall: 0.8993 -
val_accuracy: 0.7850 - val_loss: 0.4444 - val_precision: 0.7436 - val_recall:
0.8700
Epoch 20/30
50/50            41s 824ms/step -
accuracy: 0.8863 - loss: 0.2558 - precision: 0.8749 - recall: 0.9000 -
val_accuracy: 0.7450 - val_loss: 0.4935 - val_precision: 0.6750 - val_recall:
0.9450
Epoch 21/30
50/50            41s 824ms/step -
accuracy: 0.8846 - loss: 0.2738 - precision: 0.8579 - recall: 0.9196 -
val_accuracy: 0.7475 - val_loss: 0.4817 - val_precision: 0.6800 - val_recall:
0.9350
Epoch 22/30
50/50            41s 824ms/step -
accuracy: 0.8865 - loss: 0.2618 - precision: 0.8775 - recall: 0.9020 -
val_accuracy: 0.8050 - val_loss: 0.4368 - val_precision: 0.7961 - val_recall:
0.8200
Epoch 23/30
50/50            41s 825ms/step -
accuracy: 0.9078 - loss: 0.2359 - precision: 0.8917 - recall: 0.9295 -
val_accuracy: 0.8025 - val_loss: 0.4490 - val_precision: 0.7738 - val_recall:
0.8550
Epoch 24/30
50/50            41s 823ms/step -
accuracy: 0.8927 - loss: 0.2430 - precision: 0.8882 - recall: 0.9010 -
val_accuracy: 0.7850 - val_loss: 0.4825 - val_precision: 0.8608 - val_recall:
0.6800
Epoch 25/30
50/50            41s 826ms/step -
accuracy: 0.8878 - loss: 0.2739 - precision: 0.9017 - recall: 0.8785 -
val_accuracy: 0.7725 - val_loss: 0.4887 - val_precision: 0.7121 - val_recall:
0.9150
Epoch 26/30
50/50            41s 822ms/step -
accuracy: 0.8823 - loss: 0.2572 - precision: 0.8464 - recall: 0.9223 -
```

```
val_accuracy: 0.7525 - val_loss: 0.4766 - val_precision: 0.6996 - val_recall:
0.8850
Epoch 27/30
50/50            41s 828ms/step -
accuracy: 0.8965 - loss: 0.2372 - precision: 0.8802 - recall: 0.9224 -
val_accuracy: 0.8375 - val_loss: 0.4194 - val_precision: 0.8462 - val_recall:
0.8250
Epoch 28/30
50/50            41s 825ms/step -
accuracy: 0.8957 - loss: 0.2254 - precision: 0.8797 - recall: 0.9242 -
val_accuracy: 0.7575 - val_loss: 0.4930 - val_precision: 0.6900 - val_recall:
0.9350
Epoch 29/30
50/50            41s 825ms/step -
accuracy: 0.8842 - loss: 0.2540 - precision: 0.8785 - recall: 0.8969 -
val_accuracy: 0.8300 - val_loss: 0.3942 - val_precision: 0.8333 - val_recall:
0.8250
Epoch 30/30
50/50            41s 827ms/step -
accuracy: 0.8919 - loss: 0.2357 - precision: 0.8927 - recall: 0.8913 -
val_accuracy: 0.7975 - val_loss: 0.4320 - val_precision: 0.7960 - val_recall:
0.8000
training with batch_size=64, learning_rate=0.001, dropout_rate=0.2
Epoch 1/30
25/25            44s 2s/step -
accuracy: 0.7352 - loss: 0.8453 - precision: 0.7387 - recall: 0.7332 -
val_accuracy: 0.6825 - val_loss: 0.6150 - val_precision: 0.6347 - val_recall:
0.8600
Epoch 2/30
25/25            41s 2s/step -
accuracy: 0.8088 - loss: 0.4031 - precision: 0.7988 - recall: 0.8347 -
val_accuracy: 0.6975 - val_loss: 0.5885 - val_precision: 0.6561 - val_recall:
0.8300
Epoch 3/30
25/25            41s 2s/step -
accuracy: 0.8098 - loss: 0.3745 - precision: 0.8146 - recall: 0.7957 -
val_accuracy: 0.7375 - val_loss: 0.5575 - val_precision: 0.6834 - val_recall:
0.8850
Epoch 4/30
25/25            41s 2s/step -
accuracy: 0.8657 - loss: 0.3262 - precision: 0.8454 - recall: 0.9080 -
val_accuracy: 0.7275 - val_loss: 0.5572 - val_precision: 0.6717 - val_recall:
0.8900
Epoch 5/30
25/25            41s 2s/step -
accuracy: 0.8484 - loss: 0.3398 - precision: 0.8406 - recall: 0.8773 -
val_accuracy: 0.7400 - val_loss: 0.5347 - val_precision: 0.7222 - val_recall:
0.7800
```

```
Epoch 6/30
25/25             41s 2s/step -
accuracy: 0.8585 - loss: 0.3141 - precision: 0.8532 - recall: 0.8600 -
val_accuracy: 0.7425 - val_loss: 0.5456 - val_precision: 0.6830 - val_recall:
0.9050
Epoch 7/30
25/25             41s 2s/step -
accuracy: 0.8560 - loss: 0.3354 - precision: 0.8527 - recall: 0.8655 -
val_accuracy: 0.6575 - val_loss: 0.6152 - val_precision: 0.7692 - val_recall:
0.4500
Epoch 8/30
25/25             41s 2s/step -
accuracy: 0.8517 - loss: 0.3017 - precision: 0.8473 - recall: 0.8659 -
val_accuracy: 0.7950 - val_loss: 0.4556 - val_precision: 0.7837 - val_recall:
0.8150
Epoch 9/30
25/25             41s 2s/step -
accuracy: 0.8615 - loss: 0.3043 - precision: 0.8500 - recall: 0.8882 -
val_accuracy: 0.8000 - val_loss: 0.4576 - val_precision: 0.8191 - val_recall:
0.7700
Epoch 10/30
25/25             41s 2s/step -
accuracy: 0.8558 - loss: 0.2978 - precision: 0.8483 - recall: 0.8722 -
val_accuracy: 0.7950 - val_loss: 0.4878 - val_precision: 0.8687 - val_recall:
0.6950
Epoch 11/30
25/25             41s 2s/step -
accuracy: 0.8602 - loss: 0.3234 - precision: 0.8574 - recall: 0.8738 -
val_accuracy: 0.7475 - val_loss: 0.5199 - val_precision: 0.7438 - val_recall:
0.7550
Epoch 12/30
25/25             41s 2s/step -
accuracy: 0.8751 - loss: 0.2766 - precision: 0.8763 - recall: 0.8747 -
val_accuracy: 0.7825 - val_loss: 0.4548 - val_precision: 0.7181 - val_recall:
0.9300
Epoch 13/30
25/25             41s 2s/step -
accuracy: 0.8750 - loss: 0.2612 - precision: 0.8522 - recall: 0.9018 -
val_accuracy: 0.8050 - val_loss: 0.4270 - val_precision: 0.7990 - val_recall:
0.8150
Epoch 14/30
25/25             41s 2s/step -
accuracy: 0.9081 - loss: 0.2353 - precision: 0.9077 - recall: 0.9087 -
val_accuracy: 0.7950 - val_loss: 0.4645 - val_precision: 0.8278 - val_recall:
0.7450
Epoch 15/30
25/25             41s 2s/step -
accuracy: 0.8823 - loss: 0.2713 - precision: 0.8812 - recall: 0.8832 -
```

```
val_accuracy: 0.7625 - val_loss: 0.5278 - val_precision: 0.6882 - val_recall:
0.9600
Epoch 16/30
25/25             41s 2s/step -
accuracy: 0.8899 - loss: 0.2703 - precision: 0.8711 - recall: 0.9170 -
val_accuracy: 0.7950 - val_loss: 0.4646 - val_precision: 0.8315 - val_recall:
0.7400
Epoch 17/30
25/25             41s 2s/step -
accuracy: 0.8774 - loss: 0.2797 - precision: 0.8774 - recall: 0.8730 -
val_accuracy: 0.7675 - val_loss: 0.5150 - val_precision: 0.7019 - val_recall:
0.9300
Epoch 18/30
25/25             41s 2s/step -
accuracy: 0.8838 - loss: 0.2471 - precision: 0.8705 - recall: 0.8924 -
val_accuracy: 0.7575 - val_loss: 0.4601 - val_precision: 0.6886 - val_recall:
0.9400

hyperparameter tuning results:
   batch_size  learning_rate  dropout_rate  val_loss  val_accuracy  \
0          64        0.00001           0.2  0.531070        0.7475
1          32        0.00001           0.3  0.562782        0.7250
2          64        0.00001           0.4  0.568049        0.7275
3          32        0.00010           0.3  0.394171        0.8375
4          64        0.00100           0.2  0.426978        0.8050

   val_precision  val_recall
0       0.736842       0.945
1       0.717172       0.830
2       0.705882       0.910
3       0.860759       0.945
4       0.868750       0.960
```

[11]:
```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)
)
```

[12]:
```
def plot_training_history(history, title):
    """
    plot training and validation metrics

    params
    ------
    history: tf.keras.callbacks.History
        training history
    title: str
        plot title
    """
    metrics = ["loss", "accuracy", "precision", "recall"]
    fig, axes = plt.subplots(2, 2, figsize=(15, 10))
    axes = axes.ravel()

    for idx, metric in enumerate(metrics):
        axes[idx].plot(history.history[metric], label="train")
        if f"val_{metric}" in history.history:
            axes[idx].plot(history.history[f"val_{metric}"], label="val")
        axes[idx].set_title(f"{metric}")
        axes[idx].set_xlabel("epoch")
        axes[idx].set_ylabel(metric)
        axes[idx].legend()

    plt.suptitle(title)
    plt.tight_layout()
    plt.show()
```

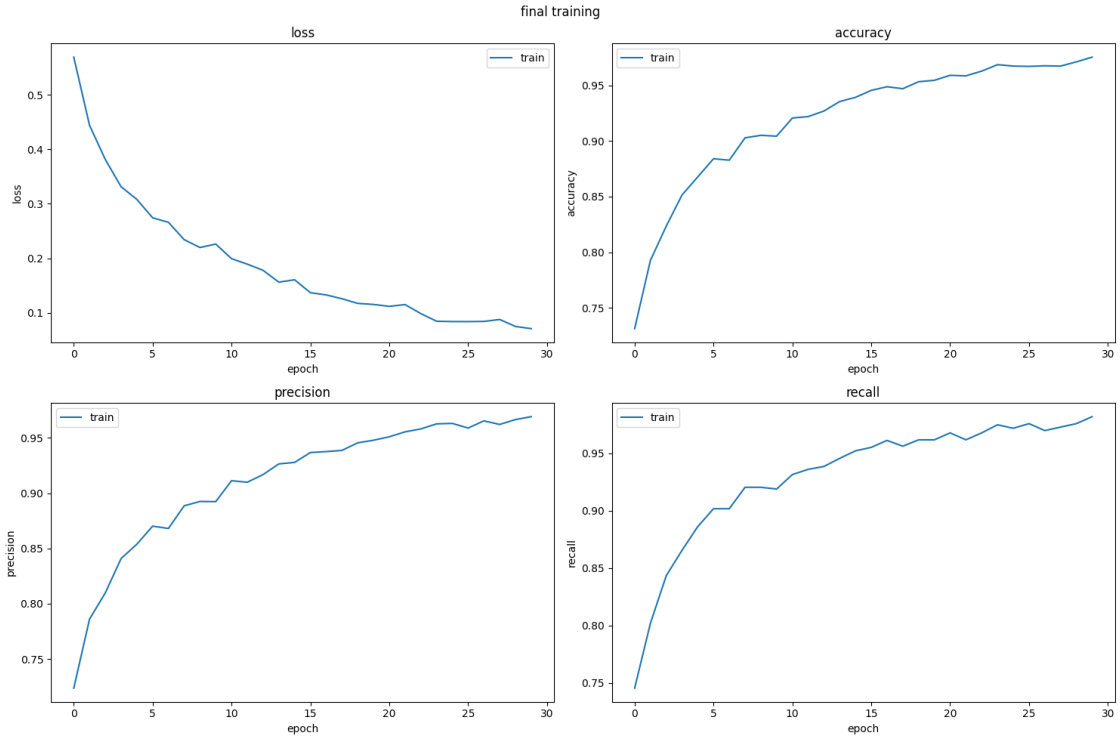
[13]:
```
# train final model with best hyperparameters
best_params = results_df.loc[results_df["val_loss"].idxmin()]
print(f"\nbest hyperparameters: {best_params}")
```

```python
# update batch size
train_data_gen.batch_size = int(best_params["batch_size"])
val_data_gen.batch_size = int(best_params["batch_size"])

# build and train model
print("\ntraining final model")
model = build_transfer_model(input_shape,␣
 ↪dropout_rate=best_params["dropout_rate"])
history = train_model(
    model, train_val_ds, None, learning_rate=best_params["learning_rate"]
)
plot_training_history(history, "final training")

# save final model
model_save_path = "../models/final_resnet_model.keras"
print(f"\nsaving final model to {model_save_path}")
model.save(model_save_path)
```

```
best hyperparameters: batch_size       32.000000
learning_rate     0.000100
dropout_rate      0.300000
val_loss          0.394171
val_accuracy      0.837500
val_precision     0.860759
val_recall        0.945000
Name: 3, dtype: float64

training final model
Epoch 1/30
125/125             82s 630ms/step -
accuracy: 0.7215 - loss: 0.6307 - precision: 0.7051 - recall: 0.7656
Epoch 2/30

/opt/anaconda3/envs/ml-2025/lib/python3.12/site-
packages/keras/src/callbacks/early_stopping.py:153: UserWarning: Early stopping
conditioned on metric `val_loss` which is not available. Available metrics are:
accuracy,loss,precision,recall
  current = self.get_monitor_value(logs)

125/125             78s 627ms/step -
accuracy: 0.7969 - loss: 0.4319 - precision: 0.7876 - recall: 0.8019
Epoch 3/30
125/125             79s 631ms/step -
accuracy: 0.8249 - loss: 0.3747 - precision: 0.8072 - recall: 0.8516
Epoch 4/30
125/125             80s 637ms/step -
accuracy: 0.8620 - loss: 0.3078 - precision: 0.8543 - recall: 0.8737
```

```
Epoch 5/30
125/125             80s 638ms/step -
accuracy: 0.8763 - loss: 0.2925 - precision: 0.8595 - recall: 0.9029
Epoch 6/30
125/125             79s 636ms/step -
accuracy: 0.8843 - loss: 0.2646 - precision: 0.8702 - recall: 0.9010
Epoch 7/30
125/125             79s 635ms/step -
accuracy: 0.8959 - loss: 0.2510 - precision: 0.8892 - recall: 0.9023
Epoch 8/30
125/125             79s 634ms/step -
accuracy: 0.9099 - loss: 0.2229 - precision: 0.8977 - recall: 0.9269
Epoch 9/30
125/125             79s 636ms/step -
accuracy: 0.9079 - loss: 0.2067 - precision: 0.8885 - recall: 0.9295
Epoch 10/30
125/125             81s 646ms/step -
accuracy: 0.9084 - loss: 0.2085 - precision: 0.8953 - recall: 0.9211
Epoch 11/30
125/125             79s 630ms/step -
accuracy: 0.9291 - loss: 0.1796 - precision: 0.9220 - recall: 0.9359
Epoch 12/30
125/125             79s 631ms/step -
accuracy: 0.9258 - loss: 0.1816 - precision: 0.9138 - recall: 0.9416
Epoch 13/30
125/125             79s 631ms/step -
accuracy: 0.9285 - loss: 0.1679 - precision: 0.9245 - recall: 0.9326
Epoch 14/30
125/125             79s 631ms/step -
accuracy: 0.9339 - loss: 0.1580 - precision: 0.9229 - recall: 0.9447
Epoch 15/30
125/125             79s 634ms/step -
accuracy: 0.9493 - loss: 0.1473 - precision: 0.9355 - recall: 0.9641
Epoch 16/30
125/125             79s 629ms/step -
accuracy: 0.9473 - loss: 0.1335 - precision: 0.9403 - recall: 0.9535
Epoch 17/30
125/125             79s 631ms/step -
accuracy: 0.9517 - loss: 0.1296 - precision: 0.9375 - recall: 0.9661
Epoch 18/30
125/125             79s 629ms/step -
accuracy: 0.9426 - loss: 0.1281 - precision: 0.9374 - recall: 0.9471
Epoch 19/30
125/125             79s 630ms/step -
accuracy: 0.9535 - loss: 0.1155 - precision: 0.9430 - recall: 0.9637
Epoch 20/30
125/125             79s 630ms/step -
accuracy: 0.9555 - loss: 0.1140 - precision: 0.9449 - recall: 0.9661
```

```
Epoch 21/30
125/125             79s 631ms/step -
accuracy: 0.9615 - loss: 0.1062 - precision: 0.9530 - recall: 0.9707
Epoch 22/30
125/125             79s 634ms/step -
accuracy: 0.9546 - loss: 0.1214 - precision: 0.9533 - recall: 0.9545
Epoch 23/30
125/125             79s 631ms/step -
accuracy: 0.9608 - loss: 0.0963 - precision: 0.9577 - recall: 0.9620
Epoch 24/30
125/125             79s 630ms/step -
accuracy: 0.9714 - loss: 0.0852 - precision: 0.9624 - recall: 0.9814
Epoch 25/30
125/125             79s 631ms/step -
accuracy: 0.9692 - loss: 0.0797 - precision: 0.9631 - recall: 0.9760
Epoch 26/30
125/125             79s 631ms/step -
accuracy: 0.9688 - loss: 0.0809 - precision: 0.9590 - recall: 0.9792
Epoch 27/30
125/125             79s 630ms/step -
accuracy: 0.9653 - loss: 0.0799 - precision: 0.9647 - recall: 0.9645
Epoch 28/30
125/125             79s 631ms/step -
accuracy: 0.9667 - loss: 0.0871 - precision: 0.9571 - recall: 0.9760
Epoch 29/30
125/125             79s 631ms/step -
accuracy: 0.9717 - loss: 0.0720 - precision: 0.9660 - recall: 0.9778
Epoch 30/30
125/125             79s 630ms/step -
accuracy: 0.9773 - loss: 0.0655 - precision: 0.9713 - recall: 0.9830
```

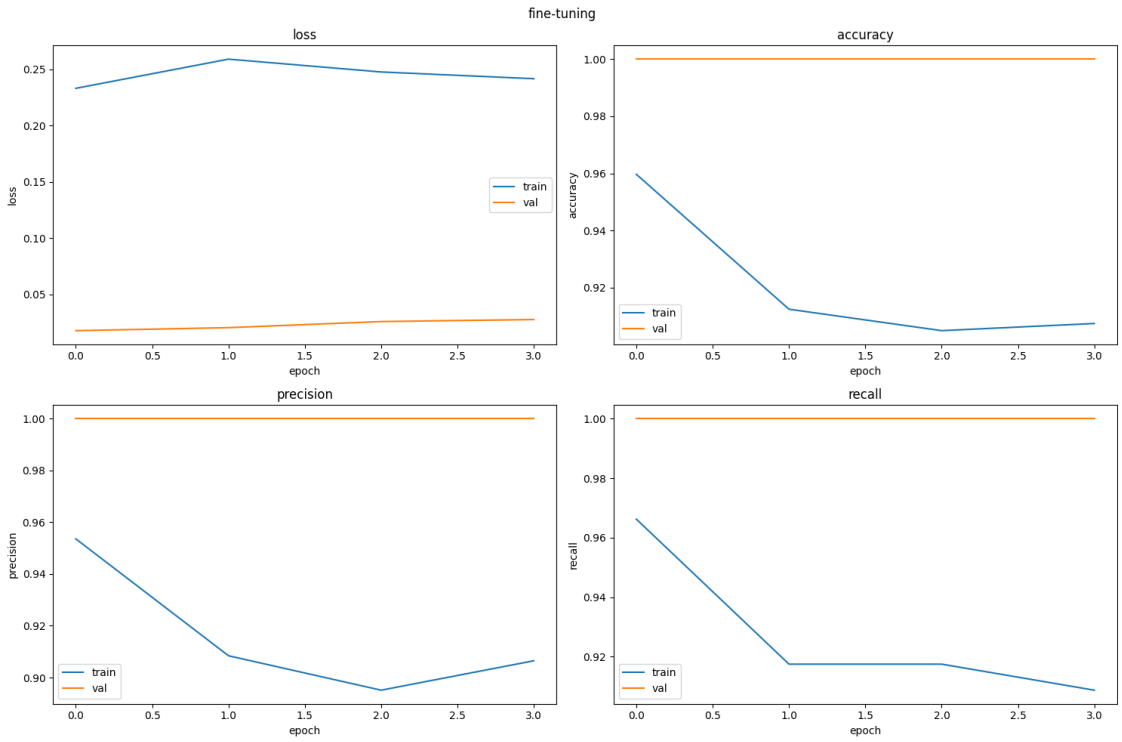



```
saving final model to ../models/final_resnet_model.keras
```

[14]:
```
# fine-tune model
print("\nfine-tuning")
history_fine = fine_tune_model(model, train_data_gen, val_data_gen)
plot_training_history(history_fine, "fine-tuning")

# save fine-tuned model
model_save_path = "../models/fine_tuned_resnet_model.keras"
print(f"\nsaving fine-tuned model to {model_save_path}")
model.save(model_save_path)
```

```
fine-tuning
Epoch 1/30
50/50              44s 835ms/step -
accuracy: 0.9661 - loss: 0.2402 - precision: 0.9597 - recall: 0.9730 -
val_accuracy: 1.0000 - val_loss: 0.0180 - val_precision: 1.0000 - val_recall:
1.0000
Epoch 2/30
50/50              41s 814ms/step -
accuracy: 0.9169 - loss: 0.2435 - precision: 0.9043 - recall: 0.9250 -
val_accuracy: 1.0000 - val_loss: 0.0206 - val_precision: 1.0000 - val_recall:
```

```
1.0000
Epoch 3/30
50/50              41s 815ms/step -
accuracy: 0.9153 - loss: 0.2247 - precision: 0.9120 - recall: 0.9224 -
val_accuracy: 1.0000 - val_loss: 0.0260 - val_precision: 1.0000 - val_recall:
1.0000
Epoch 4/30
50/50              41s 823ms/step -
accuracy: 0.9041 - loss: 0.2531 - precision: 0.9057 - recall: 0.9015 -
val_accuracy: 1.0000 - val_loss: 0.0278 - val_precision: 1.0000 - val_recall:
1.0000
```




```
saving fine-tuned model to ../models/fine_tuned_resnet_model.keras
```

[15]:
```python
# evaluate on test set
print("\nevaluating on test set")
test_loss, test_acc, test_precision, test_recall = model.evaluate(test_data_gen)
print(f"test loss: {test_loss:.4f}")
print(f"test accuracy: {test_acc:.4f}")
print(f"test precision: {test_precision:.4f}")
print(f"test recall: {test_recall:.4f}")
```

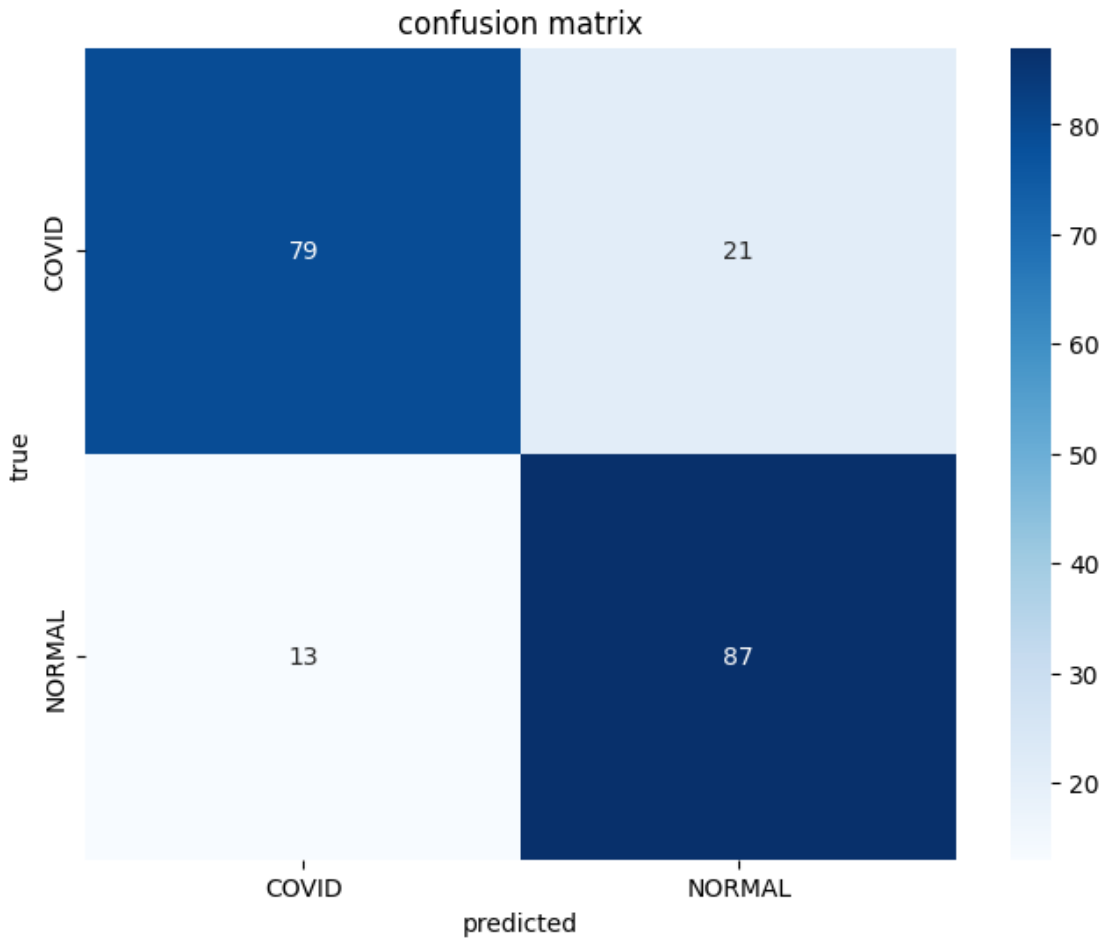
```
evaluating on test set
```

```
/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             4s 2s/step -
accuracy: 0.8190 - loss: 0.5796 - precision: 0.7113 - recall: 0.8538
test loss: 0.5247
test accuracy: 0.8300
test precision: 0.8056
test recall: 0.8700
```

[16]:
```
# plot confusion matrix
print("\ngenerating confusion matrix")
y_pred = model.predict(test_data_gen)
y_pred = (y_pred > 0.5).astype(int)
y_true = test_data_gen.classes

plot_confusion_matrix(y_true, y_pred, class_names)
```

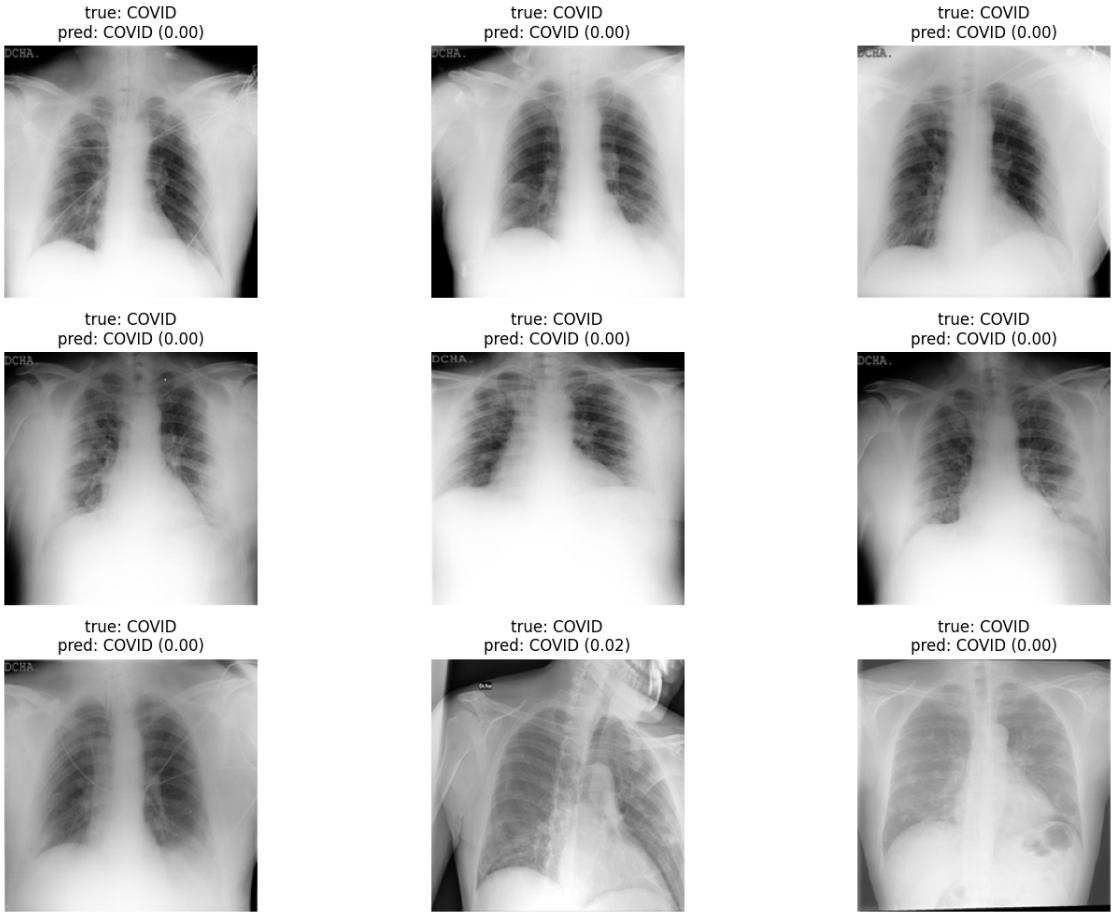
```
generating confusion matrix
2/2             6s 2s/step
```



[17]:
```
print("\nplotting sample predictions with raw images")

# get a batch of raw (unnormalized) images and labels
images_raw, labels_raw = next(iter(test_data_gen_raw))

# get the corresponding normalized batch for prediction
test_data_gen.reset()
images_norm, _ = next(iter(test_data_gen))

# make predictions
batch_pred_prob = model.predict(images_norm, verbose=0)
batch_pred = (batch_pred_prob > 0.5).astype(int).flatten()

# plot 9 samples
plt.figure(figsize=(15, 10))
for i in range(9):
    plt.subplot(3, 3, i + 1)
```

```
    plt.imshow(images_raw[i].astype("uint8"))
    true_class = class_names[int(labels_raw[i])]
    pred_class = class_names[batch_pred[i]]
    prob = batch_pred_prob[i][0]
    plt.title(f"true: {true_class}\npred: {pred_class} ({prob:.2f})")
    plt.axis("off")

plt.tight_layout()
plt.show()
```

```
plotting sample predictions with raw images
```



[ ]: