resnet

April 24, 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]:

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

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

[6]:

```python
# 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          (None, 2048)                       0
 (GlobalAveragePooling2D)

 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]:

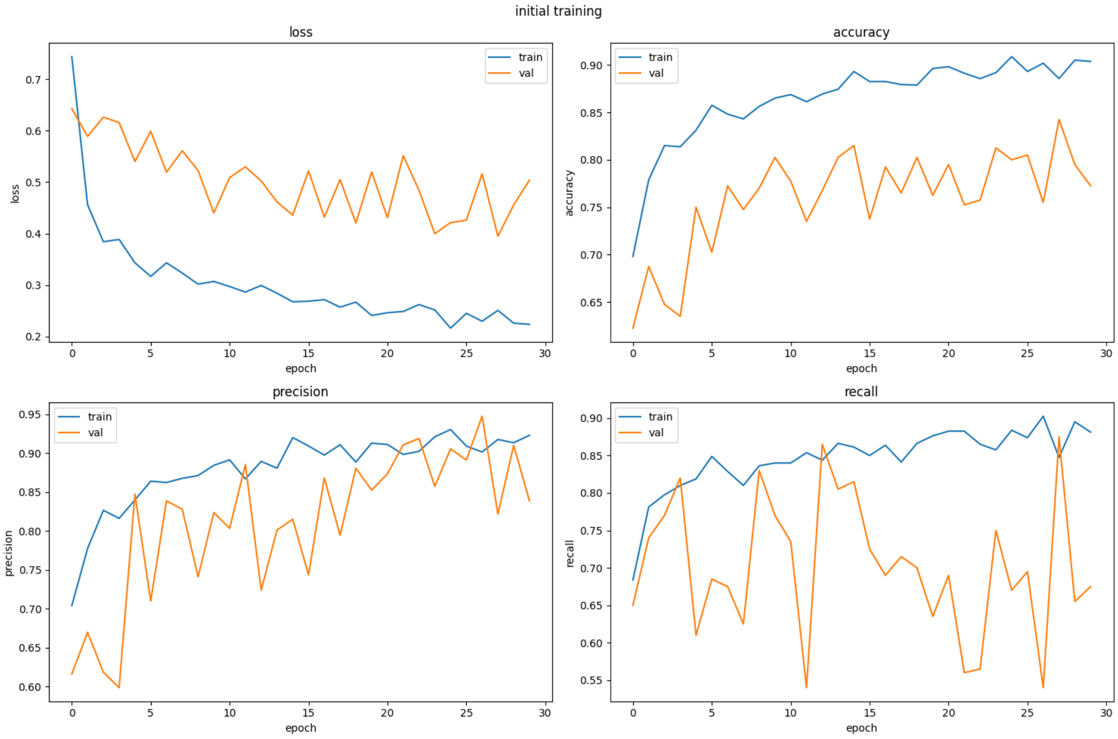
```
# train initial model
print("\ntraining initial model")
history = train_model(model, train_data_gen, val_data_gen)
plot_training_history(history, "initial training")
```

```
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                 46s 3s/step -
accuracy: 0.6598 - loss: 0.8021 - precision: 0.6607 - recall: 0.6413 -
val_accuracy: 0.6225 - val_loss: 0.6426 - val_precision: 0.6161 - val_recall:
0.6500
Epoch 2/30
13/13                 46s 4s/step -
accuracy: 0.7670 - loss: 0.4810 - precision: 0.7635 - recall: 0.7747 -
val_accuracy: 0.6875 - val_loss: 0.5889 - val_precision: 0.6697 - val_recall:
0.7400
Epoch 3/30
13/13                 44s 3s/step -
accuracy: 0.8029 - loss: 0.3951 - precision: 0.8228 - recall: 0.7899 -
val_accuracy: 0.6475 - val_loss: 0.6261 - val_precision: 0.6185 - val_recall:
0.7700
Epoch 4/30
13/13                 46s 3s/step -
accuracy: 0.8189 - loss: 0.3756 - precision: 0.8045 - recall: 0.8396 -
val_accuracy: 0.6350 - val_loss: 0.6155 - val_precision: 0.5985 - val_recall:
0.8200
Epoch 5/30
13/13                 46s 4s/step -
accuracy: 0.8337 - loss: 0.3315 - precision: 0.8182 - recall: 0.8388 -
val_accuracy: 0.7500 - val_loss: 0.5398 - val_precision: 0.8472 - val_recall:
0.6100
Epoch 6/30
13/13                 46s 4s/step -
accuracy: 0.8590 - loss: 0.3281 - precision: 0.8714 - recall: 0.8351 -
```

```
val_accuracy: 0.7025 - val_loss: 0.5988 - val_precision: 0.7098 - val_recall:
0.6850
Epoch 7/30
13/13           45s 3s/step -
accuracy: 0.8486 - loss: 0.3314 - precision: 0.8661 - recall: 0.8228 -
val_accuracy: 0.7725 - val_loss: 0.5191 - val_precision: 0.8385 - val_recall:
0.6750
Epoch 8/30
13/13           44s 3s/step -
accuracy: 0.8495 - loss: 0.3197 - precision: 0.8825 - recall: 0.8047 -
val_accuracy: 0.7475 - val_loss: 0.5609 - val_precision: 0.8278 - val_recall:
0.6250
Epoch 9/30
13/13           44s 3s/step -
accuracy: 0.8460 - loss: 0.3199 - precision: 0.8646 - recall: 0.8181 -
val_accuracy: 0.7700 - val_loss: 0.5224 - val_precision: 0.7411 - val_recall:
0.8300
Epoch 10/30
13/13           43s 3s/step -
accuracy: 0.8646 - loss: 0.2945 - precision: 0.8766 - recall: 0.8372 -
val_accuracy: 0.8025 - val_loss: 0.4400 - val_precision: 0.8235 - val_recall:
0.7700
Epoch 11/30
13/13           43s 3s/step -
accuracy: 0.8699 - loss: 0.2998 - precision: 0.8783 - recall: 0.8497 -
val_accuracy: 0.7775 - val_loss: 0.5087 - val_precision: 0.8033 - val_recall:
0.7350
Epoch 12/30
13/13           43s 3s/step -
accuracy: 0.8623 - loss: 0.2869 - precision: 0.8705 - recall: 0.8563 -
val_accuracy: 0.7350 - val_loss: 0.5296 - val_precision: 0.8852 - val_recall:
0.5400
Epoch 13/30
13/13           44s 3s/step -
accuracy: 0.8698 - loss: 0.3079 - precision: 0.8997 - recall: 0.8234 -
val_accuracy: 0.7675 - val_loss: 0.5023 - val_precision: 0.7238 - val_recall:
0.8650
Epoch 14/30
13/13           44s 3s/step -
accuracy: 0.8820 - loss: 0.2711 - precision: 0.8710 - recall: 0.8894 -
val_accuracy: 0.8025 - val_loss: 0.4616 - val_precision: 0.8010 - val_recall:
0.8050
Epoch 15/30
13/13           45s 3s/step -
accuracy: 0.8895 - loss: 0.2734 - precision: 0.9056 - recall: 0.8661 -
val_accuracy: 0.8150 - val_loss: 0.4357 - val_precision: 0.8150 - val_recall:
0.8150
Epoch 16/30
```

```
13/13           43s 3s/step -
accuracy: 0.8899 - loss: 0.2572 - precision: 0.9103 - recall: 0.8651 -
val_accuracy: 0.7375 - val_loss: 0.5218 - val_precision: 0.7436 - val_recall:
0.7250
Epoch 17/30
13/13           42s 3s/step -
accuracy: 0.8847 - loss: 0.2774 - precision: 0.8894 - recall: 0.8746 -
val_accuracy: 0.7925 - val_loss: 0.4319 - val_precision: 0.8679 - val_recall:
0.6900
Epoch 18/30
13/13           42s 3s/step -
accuracy: 0.8846 - loss: 0.2480 - precision: 0.9194 - recall: 0.8458 -
val_accuracy: 0.7650 - val_loss: 0.5049 - val_precision: 0.7944 - val_recall:
0.7150
Epoch 19/30
13/13           44s 3s/step -
accuracy: 0.8673 - loss: 0.2777 - precision: 0.8650 - recall: 0.8680 -
val_accuracy: 0.8025 - val_loss: 0.4204 - val_precision: 0.8805 - val_recall:
0.7000
Epoch 20/30
13/13           42s 3s/step -
accuracy: 0.8943 - loss: 0.2481 - precision: 0.9078 - recall: 0.8789 -
val_accuracy: 0.7625 - val_loss: 0.5196 - val_precision: 0.8523 - val_recall:
0.6350
Epoch 21/30
13/13           43s 3s/step -
accuracy: 0.9000 - loss: 0.2391 - precision: 0.9040 - recall: 0.8989 -
val_accuracy: 0.7950 - val_loss: 0.4307 - val_precision: 0.8734 - val_recall:
0.6900
Epoch 22/30
13/13           43s 3s/step -
accuracy: 0.8917 - loss: 0.2408 - precision: 0.8960 - recall: 0.8876 -
val_accuracy: 0.7525 - val_loss: 0.5514 - val_precision: 0.9106 - val_recall:
0.5600
Epoch 23/30
13/13           42s 3s/step -
accuracy: 0.8888 - loss: 0.2668 - precision: 0.9138 - recall: 0.8587 -
val_accuracy: 0.7575 - val_loss: 0.4840 - val_precision: 0.9187 - val_recall:
0.5650
Epoch 24/30
13/13           42s 3s/step -
accuracy: 0.8959 - loss: 0.2465 - precision: 0.9391 - recall: 0.8471 -
val_accuracy: 0.8125 - val_loss: 0.3996 - val_precision: 0.8571 - val_recall:
0.7500
Epoch 25/30
13/13           42s 3s/step -
accuracy: 0.9125 - loss: 0.2163 - precision: 0.9331 - recall: 0.8864 -
val_accuracy: 0.8000 - val_loss: 0.4212 - val_precision: 0.9054 - val_recall:
```

```
0.6700
Epoch 26/30
13/13            42s 3s/step -
accuracy: 0.8981 - loss: 0.2429 - precision: 0.9119 - recall: 0.8771 -
val_accuracy: 0.8050 - val_loss: 0.4260 - val_precision: 0.8910 - val_recall:
0.6950
Epoch 27/30
13/13            42s 3s/step -
accuracy: 0.9075 - loss: 0.2207 - precision: 0.9067 - recall: 0.9098 -
val_accuracy: 0.7550 - val_loss: 0.5162 - val_precision: 0.9474 - val_recall:
0.5400
Epoch 28/30
13/13            42s 3s/step -
accuracy: 0.8822 - loss: 0.2631 - precision: 0.9220 - recall: 0.8278 -
val_accuracy: 0.8425 - val_loss: 0.3949 - val_precision: 0.8216 - val_recall:
0.8750
Epoch 29/30
13/13            42s 3s/step -
accuracy: 0.9032 - loss: 0.2340 - precision: 0.9089 - recall: 0.8976 -
val_accuracy: 0.7950 - val_loss: 0.4556 - val_precision: 0.9097 - val_recall:
0.6550
Epoch 30/30
13/13            42s 3s/step -
accuracy: 0.9029 - loss: 0.2212 - precision: 0.9296 - recall: 0.8719 -
val_accuracy: 0.7725 - val_loss: 0.5037 - val_precision: 0.8385 - val_recall:
0.6750
```



[8]:

```python
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]:

```python
# 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]:

```python
# 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=0.0001, dropout_rate=0.4
Epoch 1/30
25/25                 46s 2s/step -
accuracy: 0.6585 - loss: 1.0620 - precision: 0.6895 - recall: 0.5814 -
val_accuracy: 0.7050 - val_loss: 0.6083 - val_precision: 0.6971 - val_recall:
0.7250
Epoch 2/30
25/25                 42s 2s/step -
accuracy: 0.7002 - loss: 0.6591 - precision: 0.7002 - recall: 0.6997 -
val_accuracy: 0.6700 - val_loss: 0.6149 - val_precision: 0.6954 - val_recall:
0.6050
Epoch 3/30
25/25                 42s 2s/step -
accuracy: 0.7544 - loss: 0.5469 - precision: 0.7579 - recall: 0.7235 -
val_accuracy: 0.7075 - val_loss: 0.6009 - val_precision: 0.7515 - val_recall:
0.6200
Epoch 4/30
```

```
25/25            42s 2s/step -
accuracy: 0.7719 - loss: 0.4810 - precision: 0.7722 - recall: 0.7586 -
val_accuracy: 0.7100 - val_loss: 0.5843 - val_precision: 0.7143 - val_recall:
0.7000
Epoch 5/30
25/25            43s 2s/step -
accuracy: 0.7831 - loss: 0.4635 - precision: 0.8029 - recall: 0.7576 -
val_accuracy: 0.7325 - val_loss: 0.5792 - val_precision: 0.7143 - val_recall:
0.7750
Epoch 6/30
25/25            43s 2s/step -
accuracy: 0.7903 - loss: 0.4486 - precision: 0.7891 - recall: 0.7900 -
val_accuracy: 0.6550 - val_loss: 0.6114 - val_precision: 0.6140 - val_recall:
0.8350
Epoch 7/30
25/25            42s 2s/step -
accuracy: 0.8294 - loss: 0.3850 - precision: 0.8289 - recall: 0.8357 -
val_accuracy: 0.6175 - val_loss: 0.6560 - val_precision: 0.5776 - val_recall:
0.8750
Epoch 8/30
25/25            43s 2s/step -
accuracy: 0.8118 - loss: 0.4000 - precision: 0.8071 - recall: 0.8027 -
val_accuracy: 0.7350 - val_loss: 0.5601 - val_precision: 0.7117 - val_recall:
0.7900
Epoch 9/30
25/25            43s 2s/step -
accuracy: 0.8101 - loss: 0.3783 - precision: 0.8178 - recall: 0.7982 -
val_accuracy: 0.7200 - val_loss: 0.5747 - val_precision: 0.6789 - val_recall:
0.8350
Epoch 10/30
25/25            43s 2s/step -
accuracy: 0.8325 - loss: 0.3692 - precision: 0.8238 - recall: 0.8522 -
val_accuracy: 0.7500 - val_loss: 0.5463 - val_precision: 0.7778 - val_recall:
0.7000
Epoch 11/30
25/25            43s 2s/step -
accuracy: 0.8449 - loss: 0.3512 - precision: 0.8688 - recall: 0.8125 -
val_accuracy: 0.7550 - val_loss: 0.5448 - val_precision: 0.7550 - val_recall:
0.7550
Epoch 12/30
25/25            40s 2s/step -
accuracy: 0.8363 - loss: 0.3340 - precision: 0.8397 - recall: 0.8382 -
val_accuracy: 0.7625 - val_loss: 0.5249 - val_precision: 0.8221 - val_recall:
0.6700
Epoch 13/30
25/25            42s 2s/step -
accuracy: 0.8547 - loss: 0.3075 - precision: 0.8690 - recall: 0.8326 -
val_accuracy: 0.7775 - val_loss: 0.5298 - val_precision: 0.8171 - val_recall:
```

```
0.7150
Epoch 14/30
25/25             42s 2s/step -
accuracy: 0.8442 - loss: 0.3321 - precision: 0.8561 - recall: 0.8275 -
val_accuracy: 0.7725 - val_loss: 0.5215 - val_precision: 0.7583 - val_recall:
0.8000
Epoch 15/30
25/25             42s 2s/step -
accuracy: 0.8324 - loss: 0.3455 - precision: 0.8411 - recall: 0.8280 -
val_accuracy: 0.7575 - val_loss: 0.5478 - val_precision: 0.7588 - val_recall:
0.7550
Epoch 16/30
25/25             41s 2s/step -
accuracy: 0.8476 - loss: 0.3376 - precision: 0.8723 - recall: 0.8255 -
val_accuracy: 0.7825 - val_loss: 0.4921 - val_precision: 0.8192 - val_recall:
0.7250
Epoch 17/30
25/25             42s 2s/step -
accuracy: 0.8759 - loss: 0.2984 - precision: 0.8936 - recall: 0.8431 -
val_accuracy: 0.7600 - val_loss: 0.4976 - val_precision: 0.8421 - val_recall:
0.6400
Epoch 18/30
25/25             40s 2s/step -
accuracy: 0.8453 - loss: 0.3224 - precision: 0.8548 - recall: 0.8289 -
val_accuracy: 0.7850 - val_loss: 0.5123 - val_precision: 0.7740 - val_recall:
0.8050
Epoch 19/30
25/25             40s 2s/step -
accuracy: 0.8747 - loss: 0.2953 - precision: 0.8815 - recall: 0.8679 -
val_accuracy: 0.7750 - val_loss: 0.4822 - val_precision: 0.8235 - val_recall:
0.7000
Epoch 20/30
25/25             41s 2s/step -
accuracy: 0.8562 - loss: 0.3215 - precision: 0.8675 - recall: 0.8319 -
val_accuracy: 0.7950 - val_loss: 0.4708 - val_precision: 0.8315 - val_recall:
0.7400
Epoch 21/30
25/25             40s 2s/step -
accuracy: 0.8820 - loss: 0.2754 - precision: 0.8915 - recall: 0.8693 -
val_accuracy: 0.7675 - val_loss: 0.5016 - val_precision: 0.8323 - val_recall:
0.6700
Epoch 22/30
25/25             40s 2s/step -
accuracy: 0.8604 - loss: 0.3033 - precision: 0.8727 - recall: 0.8425 -
val_accuracy: 0.8050 - val_loss: 0.4719 - val_precision: 0.8466 - val_recall:
0.7450
Epoch 23/30
25/25             41s 2s/step -
```

```
accuracy: 0.8775 - loss: 0.2769 - precision: 0.9043 - recall: 0.8491 -
val_accuracy: 0.7925 - val_loss: 0.4829 - val_precision: 0.8545 - val_recall:
0.7050
Epoch 24/30
25/25            41s 2s/step -
accuracy: 0.8691 - loss: 0.2991 - precision: 0.8766 - recall: 0.8583 -
val_accuracy: 0.8125 - val_loss: 0.4481 - val_precision: 0.8342 - val_recall:
0.7800
Epoch 25/30
25/25            40s 2s/step -
accuracy: 0.8902 - loss: 0.2694 - precision: 0.9098 - recall: 0.8751 -
val_accuracy: 0.7750 - val_loss: 0.4591 - val_precision: 0.8716 - val_recall:
0.6450
Epoch 26/30
25/25            40s 2s/step -
accuracy: 0.8885 - loss: 0.2710 - precision: 0.9108 - recall: 0.8598 -
val_accuracy: 0.8025 - val_loss: 0.4454 - val_precision: 0.8418 - val_recall:
0.7450
Epoch 27/30
25/25            40s 2s/step -
accuracy: 0.8830 - loss: 0.2704 - precision: 0.9012 - recall: 0.8617 -
val_accuracy: 0.8175 - val_loss: 0.4277 - val_precision: 0.8588 - val_recall:
0.7600
Epoch 28/30
25/25            41s 2s/step -
accuracy: 0.8742 - loss: 0.2795 - precision: 0.8909 - recall: 0.8617 -
val_accuracy: 0.7800 - val_loss: 0.4569 - val_precision: 0.8636 - val_recall:
0.6650
Epoch 29/30
25/25            41s 2s/step -
accuracy: 0.8881 - loss: 0.2651 - precision: 0.9054 - recall: 0.8678 -
val_accuracy: 0.7900 - val_loss: 0.4484 - val_precision: 0.8625 - val_recall:
0.6900
Epoch 30/30
25/25            44s 2s/step -
accuracy: 0.8642 - loss: 0.2834 - precision: 0.8890 - recall: 0.8319 -
val_accuracy: 0.7975 - val_loss: 0.4654 - val_precision: 0.8148 - val_recall:
0.7700
training with batch_size=16, learning_rate=0.0001, dropout_rate=0.3
Epoch 1/30
100/100            48s 452ms/step -
accuracy: 0.7453 - loss: 0.6873 - precision: 0.7436 - recall: 0.7306 -
val_accuracy: 0.6500 - val_loss: 0.6068 - val_precision: 0.6351 - val_recall:
0.7050
Epoch 2/30
100/100            44s 445ms/step -
accuracy: 0.7857 - loss: 0.5066 - precision: 0.7717 - recall: 0.7955 -
val_accuracy: 0.7400 - val_loss: 0.5599 - val_precision: 0.7308 - val_recall:
```

```
0.7600
Epoch 3/30
100/100            45s 447ms/step -
accuracy: 0.8203 - loss: 0.4034 - precision: 0.8170 - recall: 0.8332 -
val_accuracy: 0.7000 - val_loss: 0.5690 - val_precision: 0.6852 - val_recall:
0.7400
Epoch 4/30
100/100            45s 455ms/step -
accuracy: 0.8010 - loss: 0.4083 - precision: 0.7964 - recall: 0.8007 -
val_accuracy: 0.7500 - val_loss: 0.5430 - val_precision: 0.8289 - val_recall:
0.6300
Epoch 5/30
100/100            46s 463ms/step -
accuracy: 0.8519 - loss: 0.3441 - precision: 0.8633 - recall: 0.8246 -
val_accuracy: 0.7775 - val_loss: 0.5235 - val_precision: 0.8447 - val_recall:
0.6800
Epoch 6/30
100/100            48s 485ms/step -
accuracy: 0.8491 - loss: 0.3525 - precision: 0.8570 - recall: 0.8358 -
val_accuracy: 0.6300 - val_loss: 0.6364 - val_precision: 0.5861 - val_recall:
0.8850
Epoch 7/30
100/100            47s 472ms/step -
accuracy: 0.8268 - loss: 0.3744 - precision: 0.8181 - recall: 0.8441 -
val_accuracy: 0.7800 - val_loss: 0.5050 - val_precision: 0.7667 - val_recall:
0.8050
Epoch 8/30
100/100            47s 474ms/step -
accuracy: 0.8359 - loss: 0.3457 - precision: 0.8354 - recall: 0.8387 -
val_accuracy: 0.7850 - val_loss: 0.4894 - val_precision: 0.7969 - val_recall:
0.7650
Epoch 9/30
100/100            49s 493ms/step -
accuracy: 0.8601 - loss: 0.3008 - precision: 0.8724 - recall: 0.8432 -
val_accuracy: 0.7900 - val_loss: 0.4644 - val_precision: 0.8372 - val_recall:
0.7200
Epoch 10/30
100/100            46s 465ms/step -
accuracy: 0.8650 - loss: 0.3146 - precision: 0.8751 - recall: 0.8583 -
val_accuracy: 0.7600 - val_loss: 0.4979 - val_precision: 0.8611 - val_recall:
0.6200
Epoch 11/30
100/100            46s 461ms/step -
accuracy: 0.8678 - loss: 0.3122 - precision: 0.8873 - recall: 0.8498 -
val_accuracy: 0.7775 - val_loss: 0.4930 - val_precision: 0.8208 - val_recall:
0.7100
Epoch 12/30
100/100            46s 460ms/step -
```

```
accuracy: 0.8652 - loss: 0.3111 - precision: 0.8850 - recall: 0.8495 -
val_accuracy: 0.7575 - val_loss: 0.4963 - val_precision: 0.8503 - val_recall:
0.6250
Epoch 13/30
100/100             46s 463ms/step -
accuracy: 0.8577 - loss: 0.2977 - precision: 0.8700 - recall: 0.8237 -
val_accuracy: 0.7825 - val_loss: 0.4557 - val_precision: 0.8553 - val_recall:
0.6800
Epoch 14/30
100/100             47s 474ms/step -
accuracy: 0.8843 - loss: 0.2706 - precision: 0.9099 - recall: 0.8575 -
val_accuracy: 0.8025 - val_loss: 0.4501 - val_precision: 0.8343 - val_recall:
0.7550
Epoch 15/30
100/100             47s 470ms/step -
accuracy: 0.8584 - loss: 0.2948 - precision: 0.8695 - recall: 0.8494 -
val_accuracy: 0.7825 - val_loss: 0.4509 - val_precision: 0.8844 - val_recall:
0.6500
Epoch 16/30
100/100             46s 465ms/step -
accuracy: 0.8865 - loss: 0.2786 - precision: 0.9120 - recall: 0.8608 -
val_accuracy: 0.8125 - val_loss: 0.4199 - val_precision: 0.8531 - val_recall:
0.7550
Epoch 17/30
100/100             46s 465ms/step -
accuracy: 0.8921 - loss: 0.2624 - precision: 0.9014 - recall: 0.8880 -
val_accuracy: 0.7925 - val_loss: 0.4283 - val_precision: 0.8589 - val_recall:
0.7000
Epoch 18/30
100/100             46s 457ms/step -
accuracy: 0.9057 - loss: 0.2392 - precision: 0.9224 - recall: 0.8829 -
val_accuracy: 0.8000 - val_loss: 0.4195 - val_precision: 0.8659 - val_recall:
0.7100
Epoch 19/30
100/100             46s 459ms/step -
accuracy: 0.8798 - loss: 0.2708 - precision: 0.9127 - recall: 0.8453 -
val_accuracy: 0.8325 - val_loss: 0.3967 - val_precision: 0.8800 - val_recall:
0.7700
Epoch 20/30
100/100             47s 471ms/step -
accuracy: 0.8799 - loss: 0.2867 - precision: 0.9043 - recall: 0.8385 -
val_accuracy: 0.8000 - val_loss: 0.4309 - val_precision: 0.8947 - val_recall:
0.6800
Epoch 21/30
100/100             46s 462ms/step -
accuracy: 0.8627 - loss: 0.2649 - precision: 0.8903 - recall: 0.8192 -
val_accuracy: 0.8275 - val_loss: 0.4326 - val_precision: 0.8359 - val_recall:
0.8150
```

```
Epoch 22/30
100/100            47s 473ms/step -
accuracy: 0.9122 - loss: 0.2237 - precision: 0.9288 - recall: 0.8972 -
val_accuracy: 0.8000 - val_loss: 0.4233 - val_precision: 0.8947 - val_recall:
0.6800
Epoch 23/30
100/100            46s 461ms/step -
accuracy: 0.8992 - loss: 0.2355 - precision: 0.9147 - recall: 0.8753 -
val_accuracy: 0.8325 - val_loss: 0.4296 - val_precision: 0.8152 - val_recall:
0.8600
Epoch 24/30
100/100            47s 473ms/step -
accuracy: 0.9056 - loss: 0.2487 - precision: 0.9182 - recall: 0.8895 -
val_accuracy: 0.8250 - val_loss: 0.4226 - val_precision: 0.8693 - val_recall:
0.7650
training with batch_size=64, learning_rate=0.0001, dropout_rate=0.2
Epoch 1/30
25/25            46s 2s/step -
accuracy: 0.7163 - loss: 0.7432 - precision: 0.7398 - recall: 0.6684 -
val_accuracy: 0.7050 - val_loss: 0.5958 - val_precision: 0.7030 - val_recall:
0.7100
Epoch 2/30
25/25            42s 2s/step -
accuracy: 0.7618 - loss: 0.5137 - precision: 0.7494 - recall: 0.7931 -
val_accuracy: 0.7100 - val_loss: 0.5738 - val_precision: 0.7958 - val_recall:
0.5650
Epoch 3/30
25/25            41s 2s/step -
accuracy: 0.8156 - loss: 0.3855 - precision: 0.8281 - recall: 0.8010 -
val_accuracy: 0.6975 - val_loss: 0.5709 - val_precision: 0.6740 - val_recall:
0.7650
Epoch 4/30
25/25            41s 2s/step -
accuracy: 0.8017 - loss: 0.3962 - precision: 0.7850 - recall: 0.8154 -
val_accuracy: 0.7425 - val_loss: 0.5324 - val_precision: 0.7870 - val_recall:
0.6650
Epoch 5/30
25/25            41s 2s/step -
accuracy: 0.8282 - loss: 0.3863 - precision: 0.8392 - recall: 0.8100 -
val_accuracy: 0.7300 - val_loss: 0.5288 - val_precision: 0.8333 - val_recall:
0.5750
Epoch 6/30
25/25            41s 2s/step -
accuracy: 0.8237 - loss: 0.3614 - precision: 0.8456 - recall: 0.8022 -
val_accuracy: 0.7350 - val_loss: 0.5108 - val_precision: 0.7901 - val_recall:
0.6400
Epoch 7/30
25/25            42s 2s/step -
```

```
accuracy: 0.8392 - loss: 0.3552 - precision: 0.8474 - recall: 0.8204 -
val_accuracy: 0.7600 - val_loss: 0.4991 - val_precision: 0.8291 - val_recall:
0.6550
Epoch 8/30
25/25            41s 2s/step -
accuracy: 0.8494 - loss: 0.3251 - precision: 0.8682 - recall: 0.8269 -
val_accuracy: 0.7225 - val_loss: 0.5348 - val_precision: 0.8504 - val_recall:
0.5400
Epoch 9/30
25/25            41s 2s/step -
accuracy: 0.8526 - loss: 0.3213 - precision: 0.8575 - recall: 0.8316 -
val_accuracy: 0.7725 - val_loss: 0.4847 - val_precision: 0.8114 - val_recall:
0.7100
Epoch 10/30
25/25            41s 2s/step -
accuracy: 0.8872 - loss: 0.2999 - precision: 0.9029 - recall: 0.8694 -
val_accuracy: 0.7550 - val_loss: 0.4971 - val_precision: 0.8228 - val_recall:
0.6500
Epoch 11/30
25/25            41s 2s/step -
accuracy: 0.8705 - loss: 0.2908 - precision: 0.8700 - recall: 0.8702 -
val_accuracy: 0.7350 - val_loss: 0.5210 - val_precision: 0.8561 - val_recall:
0.5650
Epoch 12/30
25/25            42s 2s/step -
accuracy: 0.8438 - loss: 0.3274 - precision: 0.8562 - recall: 0.8388 -
val_accuracy: 0.7850 - val_loss: 0.4833 - val_precision: 0.7938 - val_recall:
0.7700
Epoch 13/30
25/25            45s 2s/step -
accuracy: 0.8578 - loss: 0.3017 - precision: 0.8557 - recall: 0.8675 -
val_accuracy: 0.7600 - val_loss: 0.4790 - val_precision: 0.8562 - val_recall:
0.6250
Epoch 14/30
25/25            41s 2s/step -
accuracy: 0.8657 - loss: 0.2863 - precision: 0.8813 - recall: 0.8393 -
val_accuracy: 0.7775 - val_loss: 0.4567 - val_precision: 0.8581 - val_recall:
0.6650
Epoch 15/30
25/25            42s 2s/step -
accuracy: 0.8742 - loss: 0.3000 - precision: 0.8851 - recall: 0.8657 -
val_accuracy: 0.7125 - val_loss: 0.5772 - val_precision: 0.8972 - val_recall:
0.4800
Epoch 16/30
25/25            42s 2s/step -
accuracy: 0.8911 - loss: 0.2556 - precision: 0.9144 - recall: 0.8662 -
val_accuracy: 0.7950 - val_loss: 0.4512 - val_precision: 0.8315 - val_recall:
0.7400
```

```
Epoch 17/30
25/25            43s 2s/step -
accuracy: 0.8797 - loss: 0.2900 - precision: 0.9026 - recall: 0.8478 -
val_accuracy: 0.7900 - val_loss: 0.4667 - val_precision: 0.7990 - val_recall:
0.7750
Epoch 18/30
25/25            42s 2s/step -
accuracy: 0.8963 - loss: 0.2363 - precision: 0.8939 - recall: 0.9046 -
val_accuracy: 0.7600 - val_loss: 0.4805 - val_precision: 0.8714 - val_recall:
0.6100
Epoch 19/30
25/25            43s 2s/step -
accuracy: 0.8874 - loss: 0.2465 - precision: 0.9272 - recall: 0.8446 -
val_accuracy: 0.7475 - val_loss: 0.4948 - val_precision: 0.8837 - val_recall:
0.5700
Epoch 20/30
25/25            42s 2s/step -
accuracy: 0.8796 - loss: 0.2771 - precision: 0.9080 - recall: 0.8429 -
val_accuracy: 0.8075 - val_loss: 0.4528 - val_precision: 0.8030 - val_recall:
0.8150
Epoch 21/30
25/25            42s 2s/step -
accuracy: 0.8812 - loss: 0.2682 - precision: 0.8721 - recall: 0.8850 -
val_accuracy: 0.8100 - val_loss: 0.4193 - val_precision: 0.8605 - val_recall:
0.7400
Epoch 22/30
25/25            41s 2s/step -
accuracy: 0.8817 - loss: 0.2600 - precision: 0.8807 - recall: 0.8803 -
val_accuracy: 0.8025 - val_loss: 0.4251 - val_precision: 0.8758 - val_recall:
0.7050
Epoch 23/30
25/25            41s 2s/step -
accuracy: 0.8889 - loss: 0.2715 - precision: 0.9050 - recall: 0.8765 -
val_accuracy: 0.7475 - val_loss: 0.5253 - val_precision: 0.9160 - val_recall:
0.5450
Epoch 24/30
25/25            41s 2s/step -
accuracy: 0.9092 - loss: 0.2294 - precision: 0.9312 - recall: 0.8767 -
val_accuracy: 0.8175 - val_loss: 0.4339 - val_precision: 0.8470 - val_recall:
0.7750
Epoch 25/30
25/25            41s 2s/step -
accuracy: 0.8923 - loss: 0.2603 - precision: 0.9051 - recall: 0.8729 -
val_accuracy: 0.8175 - val_loss: 0.4271 - val_precision: 0.8256 - val_recall:
0.8050
Epoch 26/30
25/25            41s 2s/step -
accuracy: 0.9144 - loss: 0.2188 - precision: 0.9161 - recall: 0.9140 -
```

```
val_accuracy: 0.7700 - val_loss: 0.4836 - val_precision: 0.9091 - val_recall:
0.6000
training with batch_size=16, learning_rate=0.0001, dropout_rate=0.4
Epoch 1/30
100/100            49s 462ms/step -
accuracy: 0.6957 - loss: 0.8553 - precision: 0.7421 - recall: 0.6213 -
val_accuracy: 0.6825 - val_loss: 0.6071 - val_precision: 0.6667 - val_recall:
0.7300
Epoch 2/30
100/100            46s 458ms/step -
accuracy: 0.7527 - loss: 0.5208 - precision: 0.7433 - recall: 0.7779 -
val_accuracy: 0.6575 - val_loss: 0.6168 - val_precision: 0.7442 - val_recall:
0.4800
Epoch 3/30
100/100            46s 459ms/step -
accuracy: 0.7799 - loss: 0.4781 - precision: 0.7865 - recall: 0.7745 -
val_accuracy: 0.7375 - val_loss: 0.5479 - val_precision: 0.7273 - val_recall:
0.7600
Epoch 4/30
100/100            46s 463ms/step -
accuracy: 0.8130 - loss: 0.4282 - precision: 0.8058 - recall: 0.8254 -
val_accuracy: 0.7300 - val_loss: 0.5635 - val_precision: 0.8026 - val_recall:
0.6100
Epoch 5/30
100/100            46s 459ms/step -
accuracy: 0.8118 - loss: 0.4173 - precision: 0.8299 - recall: 0.7903 -
val_accuracy: 0.7175 - val_loss: 0.5708 - val_precision: 0.8222 - val_recall:
0.5550
Epoch 6/30
100/100            46s 457ms/step -
accuracy: 0.8174 - loss: 0.3813 - precision: 0.8437 - recall: 0.7941 -
val_accuracy: 0.7600 - val_loss: 0.5226 - val_precision: 0.7737 - val_recall:
0.7350
Epoch 7/30
100/100            46s 463ms/step -
accuracy: 0.8311 - loss: 0.3725 - precision: 0.8339 - recall: 0.8188 -
val_accuracy: 0.7550 - val_loss: 0.5127 - val_precision: 0.8000 - val_recall:
0.6800
Epoch 8/30
100/100            46s 461ms/step -
accuracy: 0.8405 - loss: 0.3554 - precision: 0.8429 - recall: 0.8313 -
val_accuracy: 0.7725 - val_loss: 0.5031 - val_precision: 0.8079 - val_recall:
0.7150
Epoch 9/30
100/100            46s 459ms/step -
accuracy: 0.8413 - loss: 0.3433 - precision: 0.8473 - recall: 0.8248 -
val_accuracy: 0.7575 - val_loss: 0.5035 - val_precision: 0.8503 - val_recall:
0.6250
```

```
Epoch 10/30
100/100            46s 459ms/step -
accuracy: 0.8622 - loss: 0.3348 - precision: 0.8722 - recall: 0.8285 -
val_accuracy: 0.7775 - val_loss: 0.5063 - val_precision: 0.8208 - val_recall:
0.7100
Epoch 11/30
100/100            45s 454ms/step -
accuracy: 0.8632 - loss: 0.3250 - precision: 0.8749 - recall: 0.8489 -
val_accuracy: 0.7650 - val_loss: 0.5038 - val_precision: 0.7431 - val_recall:
0.8100
Epoch 12/30
100/100            46s 455ms/step -
accuracy: 0.8333 - loss: 0.3515 - precision: 0.8397 - recall: 0.8237 -
val_accuracy: 0.7575 - val_loss: 0.5041 - val_precision: 0.8503 - val_recall:
0.6250
Epoch 13/30
100/100            46s 458ms/step -
accuracy: 0.8501 - loss: 0.3314 - precision: 0.8649 - recall: 0.8546 -
val_accuracy: 0.7475 - val_loss: 0.5262 - val_precision: 0.8613 - val_recall:
0.5900
training with batch_size=32, learning_rate=1e-05, dropout_rate=0.4
Epoch 1/30
50/50            44s 841ms/step -
accuracy: 0.6094 - loss: 1.5066 - precision: 0.6039 - recall: 0.7029 -
val_accuracy: 0.6000 - val_loss: 0.7544 - val_precision: 0.7564 - val_recall:
0.2950
Epoch 2/30
50/50            43s 857ms/step -
accuracy: 0.5989 - loss: 0.9387 - precision: 0.6039 - recall: 0.5507 -
val_accuracy: 0.6375 - val_loss: 0.7161 - val_precision: 0.7523 - val_recall:
0.4100
Epoch 3/30
50/50            41s 829ms/step -
accuracy: 0.6162 - loss: 0.9043 - precision: 0.5833 - recall: 0.6123 -
val_accuracy: 0.6650 - val_loss: 0.7099 - val_precision: 0.7845 - val_recall:
0.4550
Epoch 4/30
50/50            41s 820ms/step -
accuracy: 0.6576 - loss: 0.7976 - precision: 0.6569 - recall: 0.6598 -
val_accuracy: 0.6900 - val_loss: 0.6826 - val_precision: 0.7879 - val_recall:
0.5200
Epoch 5/30
50/50            47s 940ms/step -
accuracy: 0.6890 - loss: 0.6731 - precision: 0.6712 - recall: 0.6957 -
val_accuracy: 0.7150 - val_loss: 0.6716 - val_precision: 0.7829 - val_recall:
0.5950
Epoch 6/30
50/50            42s 834ms/step -
```

```
accuracy: 0.7119 - loss: 0.7290 - precision: 0.7091 - recall: 0.7456 -
val_accuracy: 0.7100 - val_loss: 0.6619 - val_precision: 0.7800 - val_recall:
0.5850
Epoch 7/30
50/50            42s 835ms/step -
accuracy: 0.7087 - loss: 0.6438 - precision: 0.7213 - recall: 0.7250 -
val_accuracy: 0.7075 - val_loss: 0.6573 - val_precision: 0.7748 - val_recall:
0.5850
Epoch 8/30
50/50            43s 851ms/step -
accuracy: 0.7374 - loss: 0.5930 - precision: 0.7355 - recall: 0.7378 -
val_accuracy: 0.7200 - val_loss: 0.6247 - val_precision: 0.7619 - val_recall:
0.6400
Epoch 9/30
50/50            44s 875ms/step -
accuracy: 0.7044 - loss: 0.6138 - precision: 0.6955 - recall: 0.7529 -
val_accuracy: 0.6950 - val_loss: 0.6366 - val_precision: 0.7566 - val_recall:
0.5750
Epoch 10/30
50/50            43s 845ms/step -
accuracy: 0.7356 - loss: 0.6086 - precision: 0.7483 - recall: 0.7382 -
val_accuracy: 0.7100 - val_loss: 0.6196 - val_precision: 0.7593 - val_recall:
0.6150
Epoch 11/30
50/50            42s 833ms/step -
accuracy: 0.7335 - loss: 0.5697 - precision: 0.7382 - recall: 0.7362 -
val_accuracy: 0.7175 - val_loss: 0.6138 - val_precision: 0.7574 - val_recall:
0.6400
Epoch 12/30
50/50            42s 845ms/step -
accuracy: 0.7323 - loss: 0.5542 - precision: 0.7311 - recall: 0.7445 -
val_accuracy: 0.7150 - val_loss: 0.6140 - val_precision: 0.7654 - val_recall:
0.6200
Epoch 13/30
50/50            43s 870ms/step -
accuracy: 0.7715 - loss: 0.5236 - precision: 0.7797 - recall: 0.7612 -
val_accuracy: 0.7150 - val_loss: 0.6118 - val_precision: 0.7590 - val_recall:
0.6300
Epoch 14/30
50/50            41s 813ms/step -
accuracy: 0.7538 - loss: 0.5179 - precision: 0.7735 - recall: 0.7516 -
val_accuracy: 0.7200 - val_loss: 0.6077 - val_precision: 0.7619 - val_recall:
0.6400
Epoch 15/30
50/50            41s 812ms/step -
accuracy: 0.7908 - loss: 0.4747 - precision: 0.7887 - recall: 0.7917 -
val_accuracy: 0.7125 - val_loss: 0.6010 - val_precision: 0.7297 - val_recall:
0.6750
```

```
Epoch 16/30
50/50             41s 814ms/step -
accuracy: 0.7888 - loss: 0.4721 - precision: 0.7882 - recall: 0.7939 -
val_accuracy: 0.7150 - val_loss: 0.5992 - val_precision: 0.7560 - val_recall:
0.6350
Epoch 17/30
50/50             41s 811ms/step -
accuracy: 0.7737 - loss: 0.4998 - precision: 0.7593 - recall: 0.7900 -
val_accuracy: 0.7125 - val_loss: 0.5967 - val_precision: 0.7545 - val_recall:
0.6300
Epoch 18/30
50/50             41s 811ms/step -
accuracy: 0.8076 - loss: 0.4268 - precision: 0.8141 - recall: 0.8017 -
val_accuracy: 0.7125 - val_loss: 0.5977 - val_precision: 0.7576 - val_recall:
0.6250
Epoch 19/30
50/50             41s 817ms/step -
accuracy: 0.7762 - loss: 0.4629 - precision: 0.7678 - recall: 0.7834 -
val_accuracy: 0.7025 - val_loss: 0.5985 - val_precision: 0.7396 - val_recall:
0.6250
Epoch 20/30
50/50             41s 817ms/step -
accuracy: 0.7978 - loss: 0.4657 - precision: 0.8117 - recall: 0.7746 -
val_accuracy: 0.7050 - val_loss: 0.5935 - val_precision: 0.7412 - val_recall:
0.6300
Epoch 21/30
50/50             40s 809ms/step -
accuracy: 0.7594 - loss: 0.4846 - precision: 0.7731 - recall: 0.7540 -
val_accuracy: 0.7075 - val_loss: 0.6008 - val_precision: 0.7610 - val_recall:
0.6050
Epoch 22/30
50/50             41s 810ms/step -
accuracy: 0.7919 - loss: 0.4667 - precision: 0.8140 - recall: 0.7799 -
val_accuracy: 0.7025 - val_loss: 0.5898 - val_precision: 0.7288 - val_recall:
0.6450
Epoch 23/30
50/50             41s 814ms/step -
accuracy: 0.7637 - loss: 0.4784 - precision: 0.7621 - recall: 0.7693 -
val_accuracy: 0.6925 - val_loss: 0.5885 - val_precision: 0.7151 - val_recall:
0.6400
Epoch 24/30
50/50             40s 809ms/step -
accuracy: 0.7801 - loss: 0.4660 - precision: 0.7847 - recall: 0.7932 -
val_accuracy: 0.7025 - val_loss: 0.5874 - val_precision: 0.7368 - val_recall:
0.6300
Epoch 25/30
50/50             43s 869ms/step -
accuracy: 0.8061 - loss: 0.4191 - precision: 0.8187 - recall: 0.7970 -
```

```
val_accuracy: 0.7075 - val_loss: 0.5854 - val_precision: 0.7485 - val_recall:
0.6250
Epoch 26/30
50/50           42s 834ms/step -
accuracy: 0.7697 - loss: 0.4962 - precision: 0.7840 - recall: 0.7496 -
val_accuracy: 0.7025 - val_loss: 0.5827 - val_precision: 0.7396 - val_recall:
0.6250
Epoch 27/30
50/50           41s 828ms/step -
accuracy: 0.8049 - loss: 0.4223 - precision: 0.8116 - recall: 0.7968 -
val_accuracy: 0.7000 - val_loss: 0.5774 - val_precision: 0.7273 - val_recall:
0.6400
Epoch 28/30
50/50           41s 820ms/step -
accuracy: 0.8063 - loss: 0.4281 - precision: 0.8077 - recall: 0.7997 -
val_accuracy: 0.7075 - val_loss: 0.5760 - val_precision: 0.7515 - val_recall:
0.6200
Epoch 29/30
50/50           41s 816ms/step -
accuracy: 0.7866 - loss: 0.4277 - precision: 0.7967 - recall: 0.7831 -
val_accuracy: 0.7100 - val_loss: 0.5730 - val_precision: 0.7561 - val_recall:
0.6200
Epoch 30/30
50/50           40s 806ms/step -
accuracy: 0.8070 - loss: 0.4109 - precision: 0.8160 - recall: 0.7995 -
val_accuracy: 0.7075 - val_loss: 0.5757 - val_precision: 0.7643 - val_recall:
0.6000

hyperparameter tuning results:
   batch_size  learning_rate  dropout_rate  val_loss  val_accuracy  \
0          64        0.00010           0.4  0.427747        0.8175
1          16        0.00010           0.3  0.396711        0.8325
2          64        0.00010           0.2  0.419310        0.8175
3          16        0.00010           0.4  0.503103        0.7775
4          32        0.00001           0.4  0.573049        0.7200

   val_precision  val_recall
0       0.871622       0.875
1       0.894737       0.885
2       0.915966       0.815
3       0.861314       0.810
4       0.787879       0.675
```

[11]:

```
# 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)
)
```

[14]:

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

# 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)
```

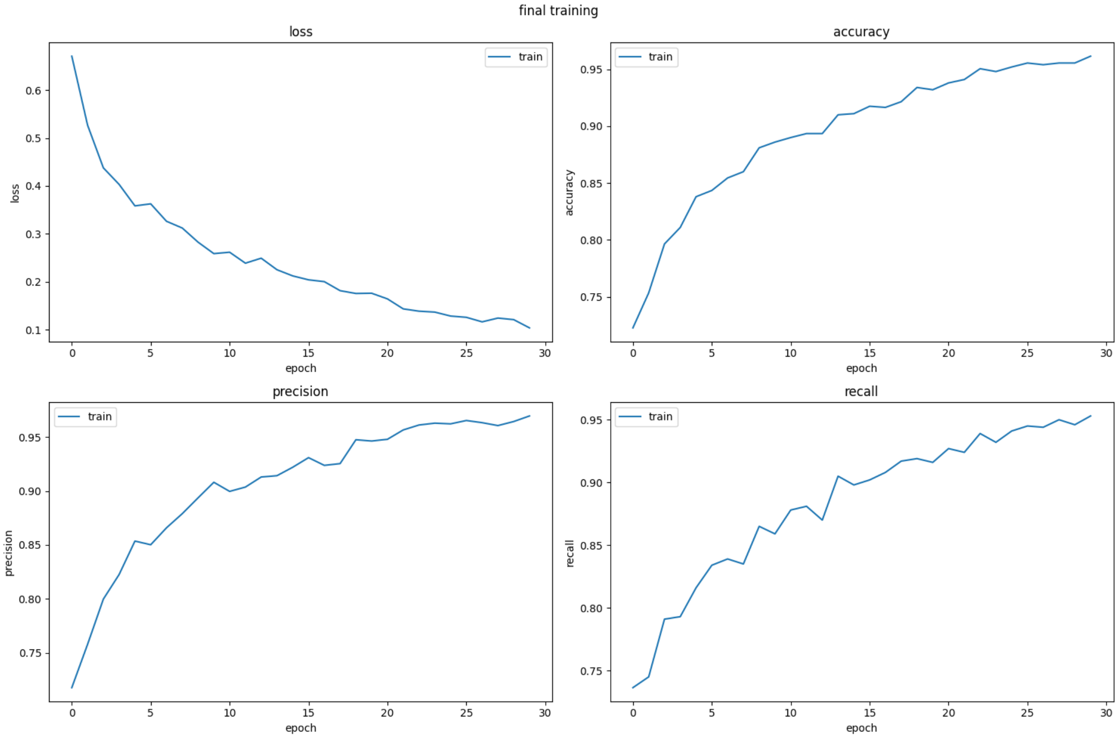
```
best hyperparameters: batch_size       16.000000
learning_rate     0.000100
dropout_rate      0.300000
val_loss          0.396711
val_accuracy      0.832500
val_precision     0.894737
val_recall        0.885000
Name: 1, dtype: float64

training final model
Epoch 1/30
63/63            45s 665ms/step -
accuracy: 0.7258 - loss: 0.7382 - precision: 0.7268 - recall: 0.7310
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)
```

```
63/63            40s 642ms/step -
accuracy: 0.7553 - loss: 0.5388 - precision: 0.7674 - recall: 0.7406
Epoch 3/30
63/63            40s 639ms/step -
accuracy: 0.7957 - loss: 0.4432 - precision: 0.7979 - recall: 0.8008
Epoch 4/30
63/63            41s 651ms/step -
accuracy: 0.8274 - loss: 0.3814 - precision: 0.8462 - recall: 0.7996
Epoch 5/30
63/63            40s 643ms/step -
accuracy: 0.8504 - loss: 0.3369 - precision: 0.8720 - recall: 0.8247
Epoch 6/30
63/63            47s 743ms/step -
accuracy: 0.8510 - loss: 0.3475 - precision: 0.8520 - recall: 0.8521
Epoch 7/30
63/63            44s 687ms/step -
accuracy: 0.8545 - loss: 0.3115 - precision: 0.8690 - recall: 0.8335
Epoch 8/30
63/63            42s 662ms/step -
accuracy: 0.8678 - loss: 0.3016 - precision: 0.8772 - recall: 0.8499
Epoch 9/30
63/63            42s 674ms/step -
accuracy: 0.8873 - loss: 0.2742 - precision: 0.8959 - recall: 0.8754
Epoch 10/30
63/63            42s 661ms/step -
accuracy: 0.8958 - loss: 0.2383 - precision: 0.9254 - recall: 0.8639
Epoch 11/30
```

```
63/63            43s 687ms/step -
accuracy: 0.8935 - loss: 0.2530 - precision: 0.9027 - recall: 0.8877
Epoch 12/30
63/63            42s 670ms/step -
accuracy: 0.8861 - loss: 0.2402 - precision: 0.9067 - recall: 0.8703
Epoch 13/30
63/63            43s 676ms/step -
accuracy: 0.9005 - loss: 0.2349 - precision: 0.9200 - recall: 0.8735
Epoch 14/30
63/63            43s 690ms/step -
accuracy: 0.8988 - loss: 0.2303 - precision: 0.9013 - recall: 0.8962
Epoch 15/30
63/63            41s 658ms/step -
accuracy: 0.9156 - loss: 0.2002 - precision: 0.9264 - recall: 0.9001
Epoch 16/30
63/63            42s 666ms/step -
accuracy: 0.9263 - loss: 0.1865 - precision: 0.9415 - recall: 0.9045
Epoch 17/30
63/63            44s 700ms/step -
accuracy: 0.9035 - loss: 0.2199 - precision: 0.9039 - recall: 0.9063
Epoch 18/30
63/63            42s 659ms/step -
accuracy: 0.9361 - loss: 0.1689 - precision: 0.9444 - recall: 0.9309
Epoch 19/30
63/63            41s 651ms/step -
accuracy: 0.9322 - loss: 0.1720 - precision: 0.9540 - recall: 0.9097
Epoch 20/30
63/63            45s 720ms/step -
accuracy: 0.9323 - loss: 0.1723 - precision: 0.9369 - recall: 0.9240
Epoch 21/30
63/63            43s 676ms/step -
accuracy: 0.9397 - loss: 0.1562 - precision: 0.9493 - recall: 0.9287
Epoch 22/30
63/63            43s 676ms/step -
accuracy: 0.9405 - loss: 0.1442 - precision: 0.9584 - recall: 0.9161
Epoch 23/30
63/63            42s 666ms/step -
accuracy: 0.9524 - loss: 0.1403 - precision: 0.9612 - recall: 0.9422
Epoch 24/30
63/63            41s 654ms/step -
accuracy: 0.9438 - loss: 0.1427 - precision: 0.9558 - recall: 0.9280
Epoch 25/30
63/63            41s 649ms/step -
accuracy: 0.9556 - loss: 0.1302 - precision: 0.9590 - recall: 0.9519
Epoch 26/30
63/63            43s 684ms/step -
accuracy: 0.9538 - loss: 0.1292 - precision: 0.9659 - recall: 0.9423
Epoch 27/30
```

```
63/63                42s 663ms/step -
accuracy: 0.9527 - loss: 0.1185 - precision: 0.9588 - recall: 0.9460
Epoch 28/30
63/63                41s 652ms/step -
accuracy: 0.9586 - loss: 0.1114 - precision: 0.9619 - recall: 0.9540
Epoch 29/30
63/63                40s 641ms/step -
accuracy: 0.9670 - loss: 0.1027 - precision: 0.9785 - recall: 0.9542
Epoch 30/30
63/63                41s 651ms/step -
accuracy: 0.9592 - loss: 0.1078 - precision: 0.9712 - recall: 0.9474
```

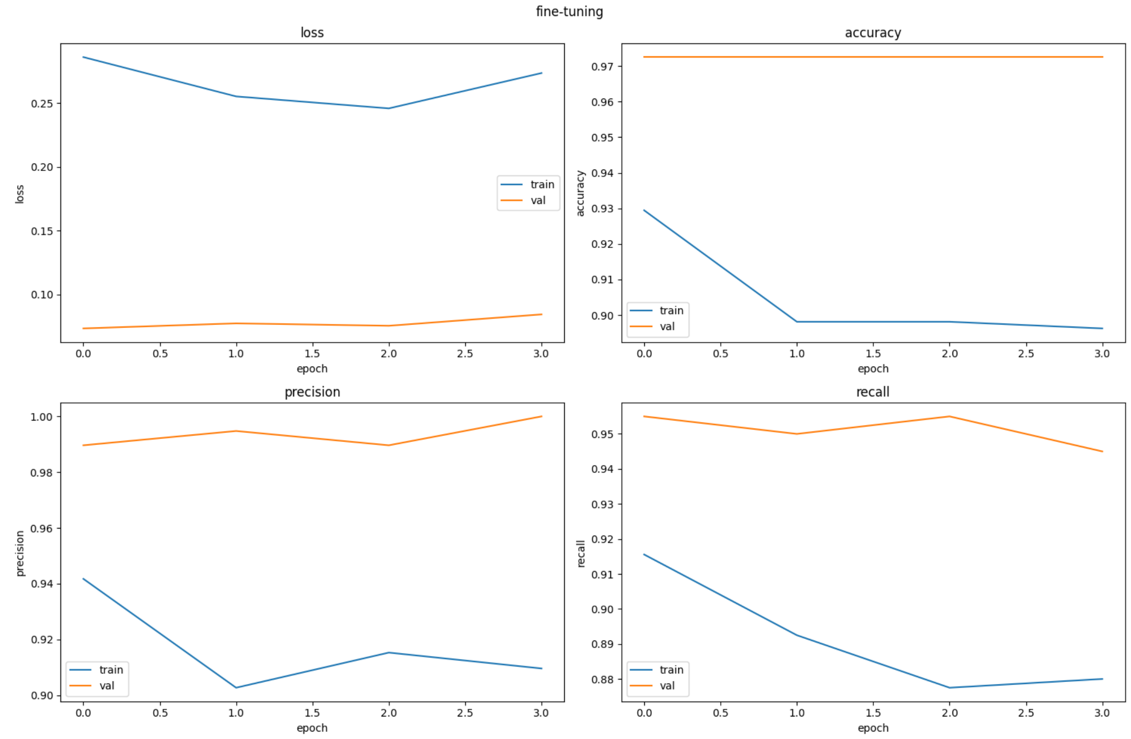


```
saving final model to ../models/final_resnet_model.keras
```

[15]:

```python
# 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
100/100            49s 467ms/step -
accuracy: 0.9445 - loss: 0.2261 - precision: 0.9541 - recall: 0.9336 -
val_accuracy: 0.9725 - val_loss: 0.0733 - val_precision: 0.9896 - val_recall:
0.9550
Epoch 2/30
100/100            47s 467ms/step -
accuracy: 0.9171 - loss: 0.2104 - precision: 0.9222 - recall: 0.9094 -
val_accuracy: 0.9725 - val_loss: 0.0773 - val_precision: 0.9948 - val_recall:
0.9500
Epoch 3/30
100/100            48s 479ms/step -
accuracy: 0.8979 - loss: 0.2428 - precision: 0.9209 - recall: 0.8662 -
val_accuracy: 0.9725 - val_loss: 0.0754 - val_precision: 0.9896 - val_recall:
0.9550
Epoch 4/30
100/100            47s 475ms/step -
accuracy: 0.8985 - loss: 0.2717 - precision: 0.9001 - recall: 0.8915 -
val_accuracy: 0.9725 - val_loss: 0.0843 - val_precision: 1.0000 - val_recall:
0.9450
```



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

[16]:

```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
```

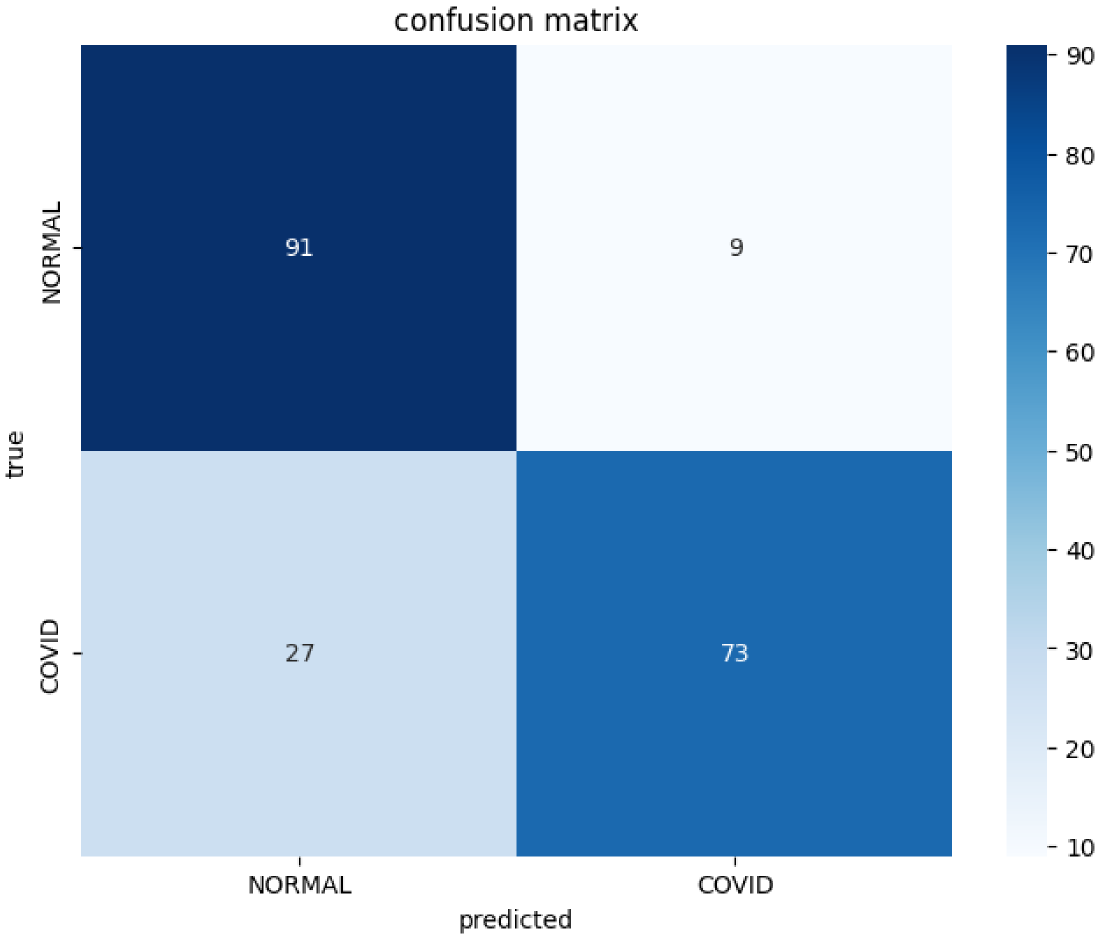
```
/opt/anaconda3/envs/ml-2025/lib/python3.12/site-
packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not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

```
2/2              5s 1s/step -
accuracy: 0.8331 - loss: 0.3571 - precision: 0.8197 - recall: 0.7129
test loss: 0.3972
test accuracy: 0.8200
test precision: 0.8902
test recall: 0.7300
```

[17]:

```python
# 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)
```

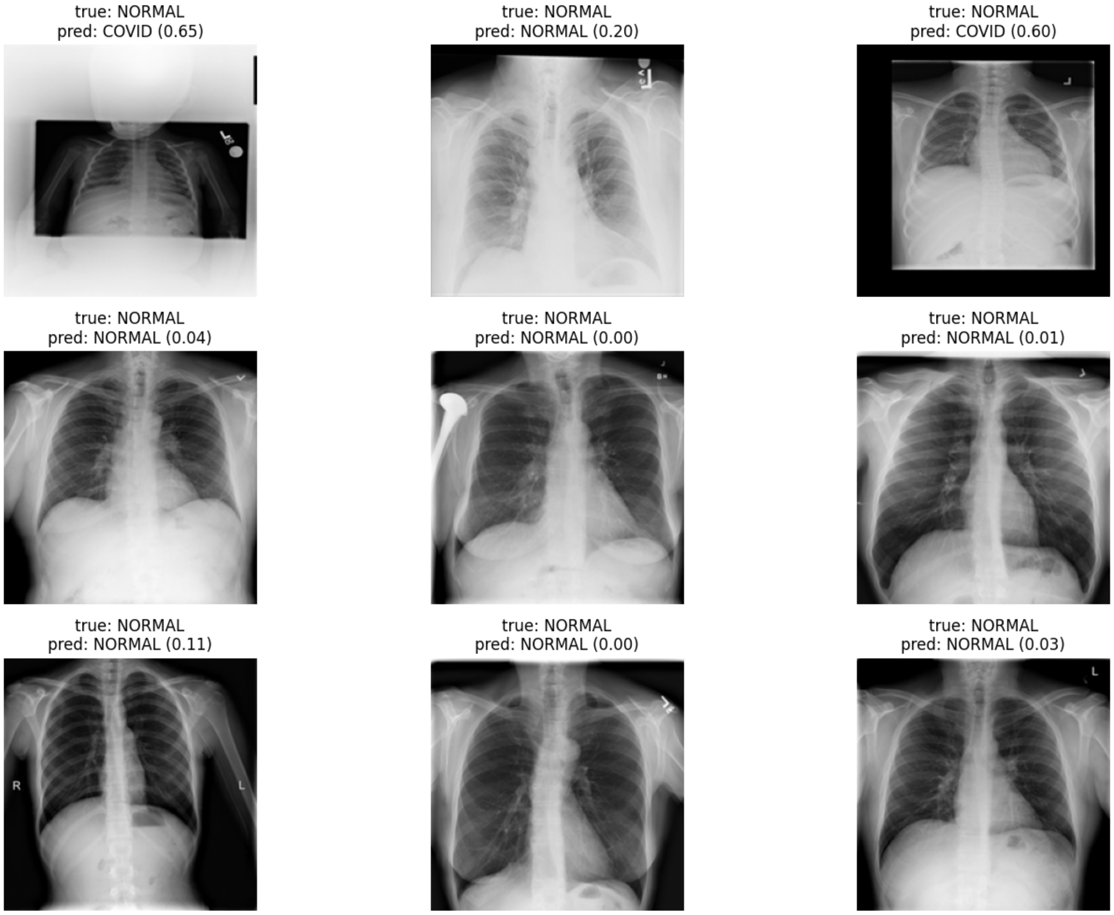
```
generating confusion matrix
2/2              6s 2s/step
```



[18]:

```
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
```



[ ]: