Done by Sini Rahuman

```python
# ============================================================
# STEP 1 — Mount Google Drive
# ============================================================

from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```python
# ============================================================
# STEP 2 — Set Paths + Create Output Folders (Drive)
# ============================================================

import os

# ✅ Your CLEAN dataset split folder (already duplicate removed)
SPLIT_ROOT = "/content/drive/MyDrive/LungCancer_Project/split_70_15_15"
TRAIN_DIR = os.path.join(SPLIT_ROOT, "train")
VALID_DIR = os.path.join(SPLIT_ROOT, "valid")
TEST_DIR  = os.path.join(SPLIT_ROOT, "test")

print("SPLIT_ROOT exists:", os.path.exists(SPLIT_ROOT))
print("Train exists:", os.path.exists(TRAIN_DIR))
print("Valid exists:", os.path.exists(VALID_DIR))
print("Test exists :", os.path.exists(TEST_DIR))
print("Split contents:", os.listdir(SPLIT_ROOT))

# ✅ Hybrid outputs saved permanently in Drive
HYBRID_BASE = "/content/drive/MyDrive/Hybrid_CNN_Transformer"
MODEL_DIR = os.path.join(HYBRID_BASE, "models")
FIG_DIR   = os.path.join(HYBRID_BASE, "figures")
XAI_DIR   = os.path.join(HYBRID_BASE, "xai")
LOG_DIR   = os.path.join(HYBRID_BASE, "logs")

for d in [HYBRID_BASE, MODEL_DIR, FIG_DIR, XAI_DIR, LOG_DIR]:
    os.makedirs(d, exist_ok=True)

print("✅ Outputs will save to:", HYBRID_BASE)
```

```
SPLIT_ROOT exists: True
Train exists: True
Valid exists: True
Test exists : True
Split contents: ['train', 'valid', 'test']
✅ Outputs will save to: /content/drive/MyDrive/Hybrid_CNN_Transformer
```

```python
# ============================================================
# STEP 3 — Install Required Libraries
# ============================================================

!pip install -q shap scikit-learn
```

```python
# ============================================================
# STEP 4 — Confirm GPU
# ============================================================

import tensorflow as tf
print("GPU devices:", tf.config.list_physical_devices("GPU"))
```

```
GPU devices: [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

```python
# ============================================================
# STEP 5 — Speed Optimization (Mixed Precision)
# ============================================================

from tensorflow.keras import mixed_precision
mixed_precision.set_global_policy("mixed_float16")
```

```
print("✅ Mixed precision policy:", mixed_precision.global_policy())
```

```
✅ Mixed precision policy: <DTypePolicy "mixed_float16">
```

```python
# ==========================================================
# STEP 6 — Import Libraries
# ==========================================================

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras import layers
import os, json, glob
import cv2
```

```python
# ==========================================================
# STEP 7 — Training Configuration
# ==========================================================

IMG_SIZE = 160
BATCH_SIZE = 16              # smaller = faster + stable
NUM_CLASSES = 3
EPOCHS = 8
AUTOTUNE = tf.data.AUTOTUNE
```

```python
# ==========================================================
# STEP 8 — Load CLEAN Dataset from Drive
# ==========================================================

train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    TRAIN_DIR,
    image_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    label_mode="categorical"
)

valid_ds = tf.keras.preprocessing.image_dataset_from_directory(
    VALID_DIR,
    image_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    label_mode="categorical"
)

test_ds = tf.keras.preprocessing.image_dataset_from_directory(
    TEST_DIR,
    image_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    label_mode="categorical",
    shuffle=False
)

class_names = train_ds.class_names
print("✅ Classes:", class_names)

with open(os.path.join(LOG_DIR, "class_names.json"), "w") as f:
    json.dump(class_names, f, indent=2)
```

```
Found 9934 files belonging to 3 classes.
Found 2130 files belonging to 3 classes.
Found 2131 files belonging to 3 classes.
✅ Classes: ['lung_aca', 'lung_n', 'lung_scc']
```

```python
# ==========================================================
# STEP 9 — Preprocess + Cache + Prefetch (Speed)
# ==========================================================

preprocess = tf.keras.applications.efficientnet.preprocess_input

def prep(ds):
    ds = ds.map(lambda x, y: (preprocess(x), y), num_parallel_calls=AUTOTUNE)
```

```
        ds = ds.cache().prefetch(AUTOTUNE)
        return ds

train_ds = prep(train_ds)
valid_ds = prep(valid_ds)
test_ds  = prep(test_ds)
```

```
# ============================================================
# STEP 10 — Transformer Encoder Block
# ============================================================

def transformer_encoder(x, num_heads=2, key_dim=16, mlp_dim=128, dropout=0.1):
    x1 = layers.LayerNormalization(epsilon=1e-6)(x)
    attn = layers.MultiHeadAttention(num_heads=num_heads, key_dim=key_dim, dropout=dropout)(x1, x1)
    x2 = layers.Add()([x, attn])

    x3 = layers.LayerNormalization(epsilon=1e-6)(x2)
    mlp = layers.Dense(mlp_dim, activation="gelu")(x3)
    mlp = layers.Dropout(dropout)(mlp)
    mlp = layers.Dense(x.shape[-1])(mlp)
    return layers.Add()([x2, mlp])
```

```
# ============================================================
# STEP 11 — Build Hybrid CNN–Transformer Model
# ============================================================

def build_hybrid():
    inputs = keras.Input(shape=(IMG_SIZE, IMG_SIZE, 3), name="image")

    backbone = tf.keras.applications.EfficientNetB0(
        include_top=False,
        weights="imagenet",
        input_tensor=inputs
    )
    backbone.trainable = False

    feat = backbone.output
    tokens = layers.Reshape((-1, feat.shape[-1]))(feat)

    # ✅  Only 1 transformer block (fast, still "hybrid")
    x = transformer_encoder(tokens, num_heads=2, key_dim=16, mlp_dim=128, dropout=0.1)

    x = layers.LayerNormalization(epsilon=1e-6)(x)
    x = layers.GlobalAveragePooling1D()(x)
    x = layers.Dropout(0.3)(x)

    # IMPORTANT: dtype float32 for final softmax with mixed precision
    outputs = layers.Dense(NUM_CLASSES, activation="softmax", dtype="float32")(x)

    model = keras.Model(inputs, outputs, name="Hybrid_EffNetB0_Transformer")
    return model

hybrid_model = build_hybrid()
hybrid_model.summary()
```

Model: "Hybrid_EffNetB0_Transformer"

Layer (type)	Output Shape	Param #	Connected to
image (InputLayer)	(None, 160, 160, 3)	0	-
rescaling (Rescaling)	(None, 160, 160, 3)	0	image[0][0]
normalization (Normalization)	(None, 160, 160, 3)	7	rescaling[0][0]
rescaling_1 (Rescaling)	(None, 160, 160, 3)	0	normalization[0]…
stem_conv_pad (ZeroPadding2D)	(None, 161, 161, 3)	0	rescaling_1[0][0]
stem_conv (Conv2D)	(None, 80, 80, 32)	864	stem_conv_pad[0]…
stem_bn (BatchNormalizatio…)	(None, 80, 80, 32)	128	stem_conv[0][0]
stem_activation (Activation)	(None, 80, 80, 32)	0	stem_bn[0][0]
block1a_dwconv (DepthwiseConv2D)	(None, 80, 80, 32)	288	stem_activation[…
block1a_bn (BatchNormalizatio…)	(None, 80, 80, 32)	128	block1a_dwconv[0…
block1a_activation (Activation)	(None, 80, 80, 32)	0	block1a_bn[0][0]
block1a_se_squeeze (GlobalAveragePool…)	(None, 32)	0	block1a_activati…
block1a_se_reshape (Reshape)	(None, 1, 1, 32)	0	block1a_se_squee…
block1a_se_reduce (Conv2D)	(None, 1, 1, 8)	264	block1a_se_resha…
block1a_se_expand (Conv2D)	(None, 1, 1, 32)	288	block1a_se_reduc…
block1a_se_excite (Multiply)	(None, 80, 80, 32)	0	block1a_activati… block1a_se_expan…
block1a_project_co… (Conv2D)	(None, 80, 80, 16)	512	block1a_se_excit…
block1a_project_bn (BatchNormalizatio…)	(None, 80, 80, 16)	64	block1a_project_…
block2a_expand_conv (Conv2D)	(None, 80, 80, 96)	1,536	block1a_project_…
block2a_expand_bn (BatchNormalizatio…)	(None, 80, 80, 96)	384	block2a_expand_c…
block2a_expand_act… (Activation)	(None, 80, 80, 96)	0	block2a_expand_b…
block2a_dwconv_pad (ZeroPadding2D)	(None, 81, 81, 96)	0	block2a_expand_a…
block2a_dwconv (DepthwiseConv2D)	(None, 40, 40, 96)	864	block2a_dwconv_p…
block2a_bn (BatchNormalizatio…)	(None, 40, 40, 96)	384	block2a_dwconv[0…
block2a_activation (Activation)	(None, 40, 40, 96)	0	block2a_bn[0][0]
block2a_se_squeeze (GlobalAveragePool…)	(None, 96)	0	block2a_activati…
block2a_se_reshape (Reshape)	(None, 1, 1, 96)	0	block2a_se_squee…

block2a_se_reduce (Conv2D)	(None, 1, 1, 4)	388	block2a_se_resha…
block2a_se_expand (Conv2D)	(None, 1, 1, 96)	480	block2a_se_reduc…
block2a_se_excite (Multiply)	(None, 40, 40, 96)	0	block2a_activati… block2a_se_expan…
block2a_project_co… (Conv2D)	(None, 40, 40, 24)	2,304	block2a_se_excit…
block2a_project_bn (BatchNormalizatio…)	(None, 40, 40, 24)	96	block2a_project_…
block2b_expand_conv (Conv2D)	(None, 40, 40, 144)	3,456	block2a_project_…
block2b_expand_bn (BatchNormalizatio…)	(None, 40, 40, 144)	576	block2b_expand_c…
block2b_expand_act… (Activation)	(None, 40, 40, 144)	0	block2b_expand_b…
block2b_dwconv (DepthwiseConv2D)	(None, 40, 40, 144)	1,296	block2b_expand_a…
block2b_bn (BatchNormalizatio…)	(None, 40, 40, 144)	576	block2b_dwconv[0…
block2b_activation (Activation)	(None, 40, 40, 144)	0	block2b_bn[0][0]
block2b_se_squeeze (GlobalAveragePool…)	(None, 144)	0	block2b_activati…
block2b_se_reshape (Reshape)	(None, 1, 1, 144)	0	block2b_se_squee…
block2b_se_reduce (Conv2D)	(None, 1, 1, 6)	870	block2b_se_resha…
block2b_se_expand (Conv2D)	(None, 1, 1, 144)	1,008	block2b_se_reduc…
block2b_se_excite (Multiply)	(None, 40, 40, 144)	0	block2b_activati… block2b_se_expan…
block2b_project_co… (Conv2D)	(None, 40, 40, 24)	3,456	block2b_se_excit…
block2b_project_bn (BatchNormalizatio…)	(None, 40, 40, 24)	96	block2b_project_…
block2b_drop (Dropout)	(None, 40, 40, 24)	0	block2b_project_…
block2b_add (Add)	(None, 40, 40, 24)	0	block2b_drop[0][… block2a_project_…
block3a_expand_conv (Conv2D)	(None, 40, 40, 144)	3,456	block2b_add[0][0]
block3a_expand_bn (BatchNormalizatio…)	(None, 40, 40, 144)	576	block3a_expand_c…
block3a_expand_act… (Activation)	(None, 40, 40, 144)	0	block3a_expand_b…
block3a_dwconv_pad (ZeroPadding2D)	(None, 43, 43, 144)	0	block3a_expand_a…
block3a_dwconv (DepthwiseConv2D)	(None, 20, 20, 144)	3,600	block3a_dwconv_p…
block3a_bn (BatchNormalizatio…)	(None, 20, 20, 144)	576	block3a_dwconv[0…
block3a_activation (Activation)	(None, 20, 20, 144)	0	block3a_bn[0][0]
block3a_se_squeeze (GlobalAveragePool…)	(None, 144)	0	block3a_activati…
block3a_se_reshape (Reshape)	(None, 1, 1, 144)	0	block3a_se_squee…

Layer	Output Shape	Param #	Connected to
(Reshape)			
block3a_se_reduce (Conv2D)	(None, 1, 1, 6)	870	block3a_se_resha…
block3a_se_expand (Conv2D)	(None, 1, 1, 144)	1,008	block3a_se_reduc…
block3a_se_excite (Multiply)	(None, 20, 20, 144)	0	block3a_activati… block3a_se_expan…
block3a_project_co… (Conv2D)	(None, 20, 20, 40)	5,760	block3a_se_excit…
block3a_project_bn (BatchNormalizatio…)	(None, 20, 20, 40)	160	block3a_project_…
block3b_expand_conv (Conv2D)	(None, 20, 20, 240)	9,600	block3a_project_…
block3b_expand_bn (BatchNormalizatio…)	(None, 20, 20, 240)	960	block3b_expand_c…
block3b_expand_act… (Activation)	(None, 20, 20, 240)	0	block3b_expand_b…
block3b_dwconv (DepthwiseConv2D)	(None, 20, 20, 240)	6,000	block3b_expand_a…
block3b_bn (BatchNormalizatio…)	(None, 20, 20, 240)	960	block3b_dwconv[0…
block3b_activation (Activation)	(None, 20, 20, 240)	0	block3b_bn[0][0]
block3b_se_squeeze (GlobalAveragePool…)	(None, 240)	0	block3b_activati…
block3b_se_reshape (Reshape)	(None, 1, 1, 240)	0	block3b_se_squee…
block3b_se_reduce (Conv2D)	(None, 1, 1, 10)	2,410	block3b_se_resha…
block3b_se_expand (Conv2D)	(None, 1, 1, 240)	2,640	block3b_se_reduc…
block3b_se_excite (Multiply)	(None, 20, 20, 240)	0	block3b_activati… block3b_se_expan…
block3b_project_co… (Conv2D)	(None, 20, 20, 40)	9,600	block3b_se_excit…
block3b_project_bn (BatchNormalizatio…)	(None, 20, 20, 40)	160	block3b_project_…
block3b_drop (Dropout)	(None, 20, 20, 40)	0	block3b_project_…
block3b_add (Add)	(None, 20, 20, 40)	0	block3b_drop[0][… block3a_project_…
block4a_expand_conv (Conv2D)	(None, 20, 20, 240)	9,600	block3b_add[0][0]
block4a_expand_bn (BatchNormalizatio…)	(None, 20, 20, 240)	960	block4a_expand_c…
block4a_expand_act… (Activation)	(None, 20, 20, 240)	0	block4a_expand_b…
block4a_dwconv_pad (ZeroPadding2D)	(None, 21, 21, 240)	0	block4a_expand_a…
block4a_dwconv (DepthwiseConv2D)	(None, 10, 10, 240)	2,160	block4a_dwconv_p…
block4a_bn (BatchNormalizatio…)	(None, 10, 10, 240)	960	block4a_dwconv[0…
block4a_activation (Activation)	(None, 10, 10, 240)	0	block4a_bn[0][0]
block4a_se_squeeze (GlobalAveragePool…)	(None, 240)	0	block4a_activati…
block4a_se_reshape	(None, 1, 1, 240)	0	block4a_se_squee…

Layer (type)	Output Shape	Param #	Connected to
(Reshape)			
block4a_se_reduce (Conv2D)	(None, 1, 1, 10)	2,410	block4a_se_resha…
block4a_se_expand (Conv2D)	(None, 1, 1, 240)	2,640	block4a_se_reduc…
block4a_se_excite (Multiply)	(None, 10, 10, 240)	0	block4a_activati… block4a_se_expan…
block4a_project_co… (Conv2D)	(None, 10, 10, 80)	19,200	block4a_se_excit…
block4a_project_bn (BatchNormalizatio…)	(None, 10, 10, 80)	320	block4a_project_…
block4b_expand_conv (Conv2D)	(None, 10, 10, 480)	38,400	block4a_project_…
block4b_expand_bn (BatchNormalizatio…)	(None, 10, 10, 480)	1,920	block4b_expand_c…
block4b_expand_act… (Activation)	(None, 10, 10, 480)	0	block4b_expand_b…
block4b_dwconv (DepthwiseConv2D)	(None, 10, 10, 480)	4,320	block4b_expand_a…
block4b_bn (BatchNormalizatio…)	(None, 10, 10, 480)	1,920	block4b_dwconv[0…
block4b_activation (Activation)	(None, 10, 10, 480)	0	block4b_bn[0][0]
block4b_se_squeeze (GlobalAveragePool…)	(None, 480)	0	block4b_activati…
block4b_se_reshape (Reshape)	(None, 1, 1, 480)	0	block4b_se_squee…
block4b_se_reduce (Conv2D)	(None, 1, 1, 20)	9,620	block4b_se_resha…
block4b_se_expand (Conv2D)	(None, 1, 1, 480)	10,080	block4b_se_reduc…
block4b_se_excite (Multiply)	(None, 10, 10, 480)	0	block4b_activati… block4b_se_expan…
block4b_project_co… (Conv2D)	(None, 10, 10, 80)	38,400	block4b_se_excit…
block4b_project_bn (BatchNormalizatio…)	(None, 10, 10, 80)	320	block4b_project_…
block4b_drop (Dropout)	(None, 10, 10, 80)	0	block4b_project_…
block4b_add (Add)	(None, 10, 10, 80)	0	block4b_drop[0][… block4a_project_…
block4c_expand_conv (Conv2D)	(None, 10, 10, 480)	38,400	block4b_add[0][0]
block4c_expand_bn (BatchNormalizatio…)	(None, 10, 10, 480)	1,920	block4c_expand_c…
block4c_expand_act… (Activation)	(None, 10, 10, 480)	0	block4c_expand_b…
block4c_dwconv (DepthwiseConv2D)	(None, 10, 10, 480)	4,320	block4c_expand_a…
block4c_bn (BatchNormalizatio…)	(None, 10, 10, 480)	1,920	block4c_dwconv[0…
block4c_activation (Activation)	(None, 10, 10, 480)	0	block4c_bn[0][0]
block4c_se_squeeze (GlobalAveragePool…)	(None, 480)	0	block4c_activati…
block4c_se_reshape (Reshape)	(None, 1, 1, 480)	0	block4c_se_squee…
block4c_se_reduce	(None, 1, 1, 20)	9,620	block4c_se_resha…

Layer (type)	Output Shape	Param #	Connected to
block4c_se_reduce (Conv2D)	(None, 1, 1, 20)	9,620	block4c_se_resha...
block4c_se_expand (Conv2D)	(None, 1, 1, 480)	10,080	block4c_se_reduc...
block4c_se_excite (Multiply)	(None, 10, 10, 480)	0	block4c_activati... block4c_se_expan...
block4c_project_co... (Conv2D)	(None, 10, 10, 80)	38,400	block4c_se_excit...
block4c_project_bn (BatchNormalizatio...)	(None, 10, 10, 80)	320	block4c_project_...
block4c_drop (Dropout)	(None, 10, 10, 80)	0	block4c_project_...
block4c_add (Add)	(None, 10, 10, 80)	0	block4c_drop[0][... block4b_add[0][0]
block5a_expand_conv (Conv2D)	(None, 10, 10, 480)	38,400	block4c_add[0][0]
block5a_expand_bn (BatchNormalizatio...)	(None, 10, 10, 480)	1,920	block5a_expand_c...
block5a_expand_act... (Activation)	(None, 10, 10, 480)	0	block5a_expand_b...
block5a_dwconv (DepthwiseConv2D)	(None, 10, 10, 480)	12,000	block5a_expand_a...
block5a_bn (BatchNormalizatio...)	(None, 10, 10, 480)	1,920	block5a_dwconv[0...
block5a_activation (Activation)	(None, 10, 10, 480)	0	block5a_bn[0][0]
block5a_se_squeeze (GlobalAveragePool...)	(None, 480)	0	block5a_activati...
block5a_se_reshape (Reshape)	(None, 1, 1, 480)	0	block5a_se_squee...
block5a_se_reduce (Conv2D)	(None, 1, 1, 20)	9,620	block5a_se_resha...
block5a_se_expand (Conv2D)	(None, 1, 1, 480)	10,080	block5a_se_reduc...
block5a_se_excite (Multiply)	(None, 10, 10, 480)	0	block5a_activati... block5a_se_expan...
block5a_project_co... (Conv2D)	(None, 10, 10, 112)	53,760	block5a_se_excit...
block5a_project_bn (BatchNormalizatio...)	(None, 10, 10, 112)	448	block5a_project_...
block5b_expand_conv (Conv2D)	(None, 10, 10, 672)	75,264	block5a_project_...
block5b_expand_bn (BatchNormalizatio...)	(None, 10, 10, 672)	2,688	block5b_expand_c...
block5b_expand_act... (Activation)	(None, 10, 10, 672)	0	block5b_expand_b...
block5b_dwconv (DepthwiseConv2D)	(None, 10, 10, 672)	16,800	block5b_expand_a...
block5b_bn (BatchNormalizatio...)	(None, 10, 10, 672)	2,688	block5b_dwconv[0...
block5b_activation (Activation)	(None, 10, 10, 672)	0	block5b_bn[0][0]
block5b_se_squeeze (GlobalAveragePool...)	(None, 672)	0	block5b_activati...
block5b_se_reshape (Reshape)	(None, 1, 1, 672)	0	block5b_se_squee...
block5b_se_reduce (Conv2D)	(None, 1, 1, 28)	18,844	block5b_se_resha...

Layer	Output Shape	Param #	Connected to
block5b_se_expand (Conv2D)	(None, 1, 1, 672)	19,488	block5b_se_reduc…
block5b_se_excite (Multiply)	(None, 10, 10, 672)	0	block5b_activati… block5b_se_expan…
block5b_project_co… (Conv2D)	(None, 10, 10, 112)	75,264	block5b_se_excit…
block5b_project_bn (BatchNormalizatio…)	(None, 10, 10, 112)	448	block5b_project_…
block5b_drop (Dropout)	(None, 10, 10, 112)	0	block5b_project_…
block5b_add (Add)	(None, 10, 10, 112)	0	block5b_drop[0][… block5a_project_…
block5c_expand_conv (Conv2D)	(None, 10, 10, 672)	75,264	block5b_add[0][0]
block5c_expand_bn (BatchNormalizatio…)	(None, 10, 10, 672)	2,688	block5c_expand_c…
block5c_expand_act… (Activation)	(None, 10, 10, 672)	0	block5c_expand_b…
block5c_dwconv (DepthwiseConv2D)	(None, 10, 10, 672)	16,800	block5c_expand_a…
block5c_bn (BatchNormalizatio…)	(None, 10, 10, 672)	2,688	block5c_dwconv[0…
block5c_activation (Activation)	(None, 10, 10, 672)	0	block5c_bn[0][0]
block5c_se_squeeze (GlobalAveragePool…)	(None, 672)	0	block5c_activati…
block5c_se_reshape (Reshape)	(None, 1, 1, 672)	0	block5c_se_squee…
block5c_se_reduce (Conv2D)	(None, 1, 1, 28)	18,844	block5c_se_resha…
block5c_se_expand (Conv2D)	(None, 1, 1, 672)	19,488	block5c_se_reduc…
block5c_se_excite (Multiply)	(None, 10, 10, 672)	0	block5c_activati… block5c_se_expan…
block5c_project_co… (Conv2D)	(None, 10, 10, 112)	75,264	block5c_se_excit…
block5c_project_bn (BatchNormalizatio…)	(None, 10, 10, 112)	448	block5c_project_…
block5c_drop (Dropout)	(None, 10, 10, 112)	0	block5c_project_…
block5c_add (Add)	(None, 10, 10, 112)	0	block5c_drop[0][… block5b_add[0][0]
block6a_expand_conv (Conv2D)	(None, 10, 10, 672)	75,264	block5c_add[0][0]
block6a_expand_bn (BatchNormalizatio…)	(None, 10, 10, 672)	2,688	block6a_expand_c…
block6a_expand_act… (Activation)	(None, 10, 10, 672)	0	block6a_expand_b…
block6a_dwconv_pad (ZeroPadding2D)	(None, 13, 13, 672)	0	block6a_expand_a…
block6a_dwconv (DepthwiseConv2D)	(None, 5, 5, 672)	16,800	block6a_dwconv_p…
block6a_bn (BatchNormalizatio…)	(None, 5, 5, 672)	2,688	block6a_dwconv[0…
block6a_activation (Activation)	(None, 5, 5, 672)	0	block6a_bn[0][0]
block6a_se_squeeze (GlobalAveragePool…)	(None, 672)	0	block6a_activati…

block6a_se_reshape (Reshape)	(None, 1, 1, 672)	0	block6a_se_squee…
block6a_se_reduce (Conv2D)	(None, 1, 1, 28)	18,844	block6a_se_resha…
block6a_se_expand (Conv2D)	(None, 1, 1, 672)	19,488	block6a_se_reduc…
block6a_se_excite (Multiply)	(None, 5, 5, 672)	0	block6a_activati… block6a_se_expan…
block6a_project_co… (Conv2D)	(None, 5, 5, 192)	129,024	block6a_se_excit…
block6a_project_bn (BatchNormalizatio…)	(None, 5, 5, 192)	768	block6a_project_…
block6b_expand_conv (Conv2D)	(None, 5, 5, 1152)	221,184	block6a_project_…
block6b_expand_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block6b_expand_c…
block6b_expand_act… (Activation)	(None, 5, 5, 1152)	0	block6b_expand_b…
block6b_dwconv (DepthwiseConv2D)	(None, 5, 5, 1152)	28,800	block6b_expand_a…
block6b_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block6b_dwconv[0…
block6b_activation (Activation)	(None, 5, 5, 1152)	0	block6b_bn[0][0]
block6b_se_squeeze (GlobalAveragePool…)	(None, 1152)	0	block6b_activati…
block6b_se_reshape (Reshape)	(None, 1, 1, 1152)	0	block6b_se_squee…
block6b_se_reduce (Conv2D)	(None, 1, 1, 48)	55,344	block6b_se_resha…
block6b_se_expand (Conv2D)	(None, 1, 1, 1152)	56,448	block6b_se_reduc…
block6b_se_excite (Multiply)	(None, 5, 5, 1152)	0	block6b_activati… block6b_se_expan…
block6b_project_co… (Conv2D)	(None, 5, 5, 192)	221,184	block6b_se_excit…
block6b_project_bn (BatchNormalizatio…)	(None, 5, 5, 192)	768	block6b_project_…
block6b_drop (Dropout)	(None, 5, 5, 192)	0	block6b_project_…
block6b_add (Add)	(None, 5, 5, 192)	0	block6b_drop[0][… block6a_project_…
block6c_expand_conv (Conv2D)	(None, 5, 5, 1152)	221,184	block6b_add[0][0]
block6c_expand_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block6c_expand_c…
block6c_expand_act… (Activation)	(None, 5, 5, 1152)	0	block6c_expand_b…
block6c_dwconv (DepthwiseConv2D)	(None, 5, 5, 1152)	28,800	block6c_expand_a…
block6c_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block6c_dwconv[0…
block6c_activation (Activation)	(None, 5, 5, 1152)	0	block6c_bn[0][0]
block6c_se_squeeze (GlobalAveragePool…)	(None, 1152)	0	block6c_activati…
block6c_se_reshape (Reshape)	(None, 1, 1, 1152)	0	block6c_se_squee…

Layer	Output Shape	Param #	Connected to
block6c_se_reduce (Conv2D)	(None, 1, 1, 48)	55,344	block6c_se_resha…
block6c_se_expand (Conv2D)	(None, 1, 1, 1152)	56,448	block6c_se_reduc…
block6c_se_excite (Multiply)	(None, 5, 5, 1152)	0	block6c_activati… block6c_se_expan…
block6c_project_co… (Conv2D)	(None, 5, 5, 192)	221,184	block6c_se_excit…
block6c_project_bn (BatchNormalizatio…)	(None, 5, 5, 192)	768	block6c_project_…
block6c_drop (Dropout)	(None, 5, 5, 192)	0	block6c_project_…
block6c_add (Add)	(None, 5, 5, 192)	0	block6c_drop[0][… block6b_add[0][0]
block6d_expand_conv (Conv2D)	(None, 5, 5, 1152)	221,184	block6c_add[0][0]
block6d_expand_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block6d_expand_c…
block6d_expand_act… (Activation)	(None, 5, 5, 1152)	0	block6d_expand_b…
block6d_dwconv (DepthwiseConv2D)	(None, 5, 5, 1152)	28,800	block6d_expand_a…
block6d_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block6d_dwconv[0…
block6d_activation (Activation)	(None, 5, 5, 1152)	0	block6d_bn[0][0]
block6d_se_squeeze (GlobalAveragePool…)	(None, 1152)	0	block6d_activati…
block6d_se_reshape (Reshape)	(None, 1, 1, 1152)	0	block6d_se_squee…
block6d_se_reduce (Conv2D)	(None, 1, 1, 48)	55,344	block6d_se_resha…
block6d_se_expand (Conv2D)	(None, 1, 1, 1152)	56,448	block6d_se_reduc…
block6d_se_excite (Multiply)	(None, 5, 5, 1152)	0	block6d_activati… block6d_se_expan…
block6d_project_co… (Conv2D)	(None, 5, 5, 192)	221,184	block6d_se_excit…
block6d_project_bn (BatchNormalizatio…)	(None, 5, 5, 192)	768	block6d_project_…
block6d_drop (Dropout)	(None, 5, 5, 192)	0	block6d_project_…
block6d_add (Add)	(None, 5, 5, 192)	0	block6d_drop[0][… block6c_add[0][0]
block7a_expand_conv (Conv2D)	(None, 5, 5, 1152)	221,184	block6d_add[0][0]
block7a_expand_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block7a_expand_c…
block7a_expand_act… (Activation)	(None, 5, 5, 1152)	0	block7a_expand_b…
block7a_dwconv (DepthwiseConv2D)	(None, 5, 5, 1152)	10,368	block7a_expand_a…
block7a_bn (BatchNormalizatio…)	(None, 5, 5, 1152)	4,608	block7a_dwconv[0…
block7a_activation (Activation)	(None, 5, 5, 1152)	0	block7a_bn[0][0]
block7a_se_squeeze (GlobalAveragePool…	(None, 1152)	0	block7a_activati…

```python
# =========================================================
# STEP 12 — Compile Model
# =========================================================

hybrid_model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-3),
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)
```

```python
# =========================================================
# STEP 13 — Callbacks (Save best model to Drive)
# =========================================================

BEST_MODEL_PATH = os.path.join(MODEL_DIR, "best_hybrid.keras")

callbacks = [
    tf.keras.callbacks.EarlyStopping(monitor="val_accuracy", patience=3, restore_best_weights=True),
    tf.keras.callbacks.ReduceLROnPlateau(monitor="val_loss", factor=0.5, patience=2),
    tf.keras.callbacks.ModelCheckpoint(BEST_MODEL_PATH, monitor="val_accuracy", save_best_only=True)
]

print("✅ Best model will save to:", BEST_MODEL_PATH)
```

✅ Best model will save to: /content/drive/MyDrive/Hybrid_CNN_Transformer/models/best_hybrid.keras

```python
# =========================================================
# STEP 14 — Train Stage 1 (Frozen Backbone)
# =========================================================

history1 = hybrid_model.fit(
    train_ds,
    validation_data=valid_ds,
    epochs=EPOCHS,
    callbacks=callbacks,
    verbose=1
)

with open(os.path.join(LOG_DIR, "history_stage1.json"), "w") as f:
    json.dump(history1.history, f, indent=2)

print("✅ Saved history to Drive")
print("✅ Best model exists?", os.path.exists(BEST_MODEL_PATH))
```

```
Epoch 1/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 2266s 4s/step - accuracy: 0.8999 - loss: 0.3009 - val_accuracy: 0.9718 - val_loss: 0.0926 - learnin
Epoch 2/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 7s 11ms/step - accuracy: 0.9561 - loss: 0.1367 - val_accuracy: 0.9624 - val_loss: 0.1057 - learning
Epoch 3/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 7s 12ms/step - accuracy: 0.9685 - loss: 0.0939 - val_accuracy: 0.9812 - val_loss: 0.0509 - learning
Epoch 4/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 7s 12ms/step - accuracy: 0.9753 - loss: 0.0720 - val_accuracy: 0.9822 - val_loss: 0.0427 - learning
Epoch 5/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 6s 10ms/step - accuracy: 0.9770 - loss: 0.0726 - val_accuracy: 0.9803 - val_loss: 0.0546 - learning
Epoch 6/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 7s 12ms/step - accuracy: 0.9808 - loss: 0.0520 - val_accuracy: 0.9859 - val_loss: 0.0410 - learning
Epoch 7/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 7s 12ms/step - accuracy: 0.9808 - loss: 0.0648 - val_accuracy: 0.9873 - val_loss: 0.0350 - learning
Epoch 8/8
621/621 ━━━━━━━━━━━━━━━━━━━━ 7s 11ms/step - accuracy: 0.9817 - loss: 0.0521 - val_accuracy: 0.9897 - val_loss: 0.0334 - learning
✅ Saved history to Drive
✅ Best model exists? True
```

```python
# =========================================================
# STEP 15 — Evaluate Best Model on Test Set
# =========================================================

best_model = tf.keras.models.load_model(BEST_MODEL_PATH)

test_loss, test_acc = best_model.evaluate(test_ds, verbose=1)
print("✅ Test Accuracy:", test_acc)

with open(os.path.join(LOG_DIR, "test_results.txt"), "w") as f:
```

```
        f.write(f"test_loss: {test_loss}\n")
        f.write(f"test_accuracy: {test_acc}\n")
```

134/134 ──────────────── **401s** 3s/step - accuracy: 0.9922 - loss: 0.0252
✅ Test Accuracy: 0.9831064939498901

```
# ============================================================
# STEP 16 — Save y_true, y_pred, y_prob (Drive)
# ============================================================

import numpy as np

y_true, y_pred = [], []
y_prob_list = []

for xb, yb in test_ds:
    probs = best_model.predict(xb, verbose=0)
    y_prob_list.append(probs)
    y_true.extend(np.argmax(yb.numpy(), axis=1))
    y_pred.extend(np.argmax(probs, axis=1))

y_true = np.array(y_true)
y_pred = np.array(y_pred)
y_prob = np.vstack(y_prob_list)

np.save(os.path.join(LOG_DIR, "y_true.npy"), y_true)
np.save(os.path.join(LOG_DIR, "y_pred.npy"), y_pred)
np.save(os.path.join(LOG_DIR, "y_prob.npy"), y_prob)

print("✅ Saved y_true/y_pred/y_prob to:", LOG_DIR)
```

✅ Saved y_true/y_pred/y_prob to: /content/drive/MyDrive/Hybrid_CNN_Transformer/logs

```
# ============================================================
# STEP 17 — Classification Report + Confusion Matrix (Save)
# ============================================================

from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay

report = classification_report(y_true, y_pred, target_names=class_names, digits=4)
print(report)

with open(os.path.join(LOG_DIR, "classification_report.txt"), "w") as f:
    f.write(report)

cm = confusion_matrix(y_true, y_pred)

plt.figure()
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)
disp.plot(values_format="d")
plt.title("Confusion Matrix — Hybrid CNN-Transformer")

cm_path = os.path.join(FIG_DIR, "confusion_matrix.png")
plt.savefig(cm_path, dpi=300, bbox_inches="tight")
plt.show()

print("✅ Saved:", cm_path)
```
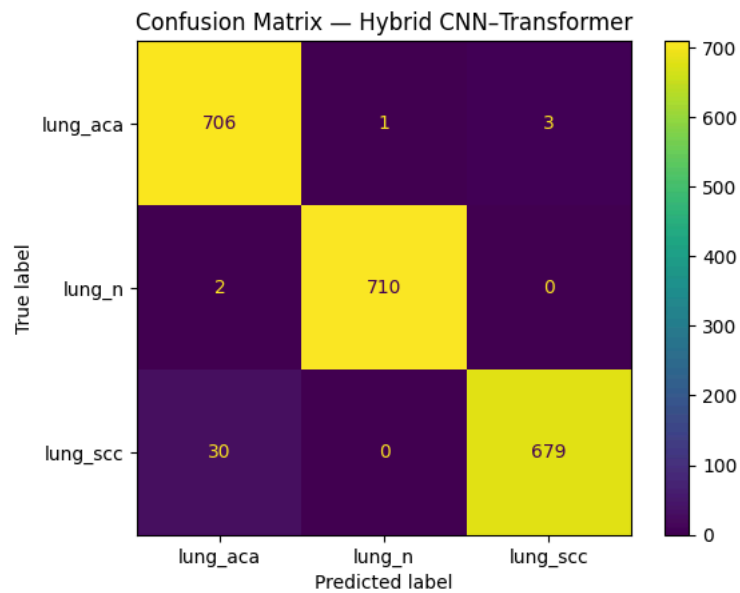
```
               precision    recall  f1-score   support

    lung_aca      0.9566    0.9944    0.9751       710
      lung_n      0.9986    0.9972    0.9979       712
    lung_scc      0.9956    0.9577    0.9763       709

    accuracy                          0.9831      2131
   macro avg      0.9836    0.9831    0.9831      2131
weighted avg      0.9836    0.9831    0.9831      2131
```

```
<Figure size 640x480 with 0 Axes>
```


Confusion Matrix — Hybrid CNN–Transformer

✅ Saved: /content/drive/MyDrive/Hybrid_CNN_Transformer/figures/confusion_matrix.png

```python
# ============================================================
# STEP 18 — ROC Curve (OvR) Save
# ============================================================

from sklearn.preprocessing import label_binarize
from sklearn.metrics import roc_curve, auc

y_true_bin = label_binarize(y_true, classes=list(range(NUM_CLASSES)))

plt.figure()
for i in range(NUM_CLASSES):
    fpr, tpr, _ = roc_curve(y_true_bin[:, i], y_prob[:, i])
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, label=f"{class_names[i]} (AUC={roc_auc:.4f})")

plt.plot([0, 1], [0, 1], linestyle="--")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve (OvR) — Hybrid CNN-Transformer")
plt.legend()

roc_path = os.path.join(FIG_DIR, "roc_curve.png")
plt.savefig(roc_path, dpi=300, bbox_inches="tight")
plt.show()

print("✅ Saved:", roc_path)
```

ROC Curve (OvR) — Hybrid CNN–Transformer
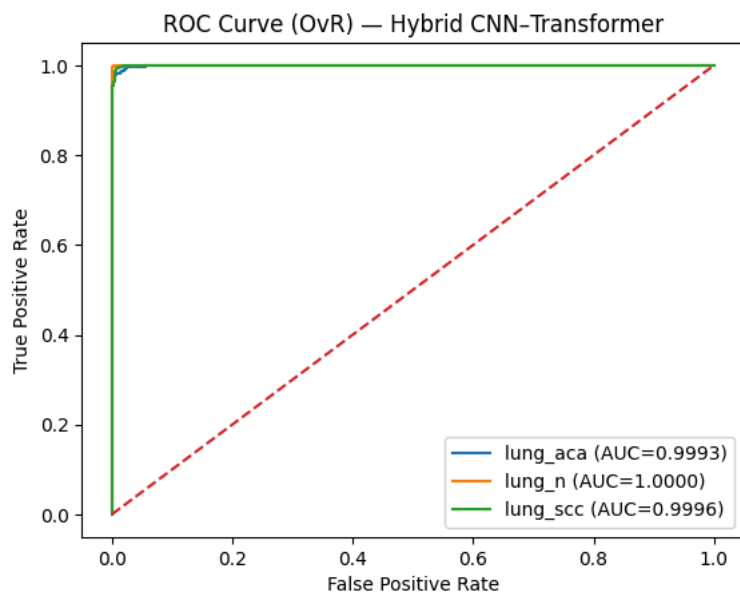
✅ Saved: /content/drive/MyDrive/Hybrid_CNN_Transformer/figures/roc_curve.png

```
# ===========================================================
# STEP 19 — Grad-CAM Save 10 Images per Class
# ===========================================================

TARGET_LAYER = "top_conv"

grad_model = tf.keras.models.Model(
    inputs=best_model.inputs,
    outputs=[best_model.get_layer(TARGET_LAYER).output, best_model.output]
)

def make_gradcam_heatmap(img_array):
    with tf.GradientTape() as tape:
        conv_outputs, preds = grad_model(img_array)
        class_idx = tf.argmax(preds[0])
        loss = preds[:, class_idx]
    grads = tape.gradient(loss, conv_outputs)
    pooled_grads = tf.reduce_mean(grads, axis=(0, 1, 2))
    conv_outputs = conv_outputs[0]
    heatmap = tf.reduce_sum(pooled_grads * conv_outputs, axis=-1)
    heatmap = tf.maximum(heatmap, 0) / (tf.reduce_max(heatmap) + 1e-8)
    return heatmap.numpy()

def save_overlay(img_path, out_path, alpha=0.4):
    img_bgr = cv2.imread(img_path)
    img_bgr = cv2.resize(img_bgr, (IMG_SIZE, IMG_SIZE))
    img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

    arr = np.expand_dims(img_rgb.astype(np.float32), axis=0)
    arr = preprocess(arr)

    heatmap = make_gradcam_heatmap(arr)
    heatmap = cv2.resize(heatmap, (IMG_SIZE, IMG_SIZE))
    heatmap_u8 = np.uint8(255 * heatmap)
    heatmap_color = cv2.applyColorMap(heatmap_u8, cv2.COLORMAP_JET)
    heatmap_color = cv2.cvtColor(heatmap_color, cv2.COLOR_BGR2RGB)

    overlay = (img_rgb * (1 - alpha) + heatmap_color * alpha).astype(np.uint8)
    cv2.imwrite(out_path, cv2.cvtColor(overlay, cv2.COLOR_RGB2BGR))

for cls in class_names:
    cls_imgs = sorted(glob.glob(os.path.join(TEST_DIR, cls, "*.jpg")))[:10]
    for i, p in enumerate(cls_imgs, 1):
        out = os.path.join(XAI_DIR, f"{cls}_gradcam_{i}.png")
        save_overlay(p, out)

print("✅ Saved Grad-CAM images to:", XAI_DIR)
```

✅ Saved Grad-CAM images to: /content/drive/MyDrive/Hybrid_CNN_Transformer/xai

```
# ============================================================
# DISPLAY A — Show saved Grad-CAM images from Drive
# ============================================================

import os, glob
import matplotlib.pyplot as plt
import cv2

XAI_DIR = "/content/drive/MyDrive/Hybrid_CNN_Transformer/xai"

imgs = sorted(glob.glob(os.path.join(XAI_DIR, "*.png")))[:12]  # show first 12
print("Found Grad-CAM images:", len(glob.glob(os.path.join(XAI_DIR, '*.png'))))

plt.figure(figsize=(14, 8))
for i, p in enumerate(imgs, 1):
    img = cv2.cvtColor(cv2.imread(p), cv2.COLOR_BGR2RGB)
    plt.subplot(3, 4, i)
    plt.imshow(img)
    plt.title(os.path.basename(p), fontsize=8)
    plt.axis("off")
plt.tight_layout()
plt.show()
```

```
Found Grad-CAM images: 0
<Figure size 1400x800 with 0 Axes>
```

```
# ============================================================
# STEP 1 — Detect actual image extensions in test folder
# ============================================================

import os, glob

test_root = "/content/drive/MyDrive/LungCancer_Project/split_70_15_15/test"

all_files = glob.glob(os.path.join(test_root, "*", "*"))
exts = sorted(set([os.path.splitext(f)[1].lower() for f in all_files]))

print("Total files:", len(all_files))
print("Detected extensions:", exts)

# Show some sample filenames
print("Sample files:", all_files[:10])
```

```
Total files: 2131
Detected extensions: ['.jpeg']
Sample files: ['/content/drive/MyDrive/LungCancer_Project/split_70_15_15/test/lung_aca/lungaca3438.jpeg', '/content/drive/MyDriv
```

```
# ============================================================
# FIXED DISPLAY — Robust Grad-CAM Visualization
# ============================================================

import os, glob, numpy as np, tensorflow as tf, cv2
import matplotlib.pyplot as plt

# ---------------- Pick image ----------------
chosen_class = "lung_aca"     # change if needed
cdir = os.path.join(TEST_DIR, chosen_class)
img_path = sorted(glob.glob(os.path.join(cdir, "*.jpeg")))[0]
print("Using image:", img_path)

# ---------------- Target layer ----------------
TARGET_LAYER = "top_conv"
print("Using TARGET_LAYER:", TARGET_LAYER)

grad_model = tf.keras.models.Model(
    inputs=best_model.inputs,
    outputs=[best_model.get_layer(TARGET_LAYER).output,
             best_model.output]
)

# ---------------- GradCAM function ----------------
def make_gradcam_heatmap(img_array):
```

```python
    with tf.GradientTape() as tape:
        conv_outputs, preds = grad_model(img_array, training=False)
        class_idx = tf.argmax(preds[0])
        loss = preds[:, class_idx]

    grads = tape.gradient(loss, conv_outputs)

    # SAFETY: convert to numpy + clean
    conv_outputs = conv_outputs[0].numpy()
    grads = grads[0].numpy()

    # Global average pooling on gradients
    weights = np.mean(grads, axis=(0, 1))

    # Weighted sum
    heatmap = np.sum(weights * conv_outputs, axis=-1)

    # ReLU + normalize
    heatmap = np.maximum(heatmap, 0)
    if np.max(heatmap) != 0:
        heatmap /= np.max(heatmap)

    return heatmap, preds.numpy()

# ---------------- Read image ----------------
img_bgr = cv2.imread(img_path)
img_bgr = cv2.resize(img_bgr, (IMG_SIZE, IMG_SIZE))
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

img_array = np.expand_dims(img_rgb.astype(np.float32), axis=0)
img_array = preprocess(img_array)

# ---------------- Compute heatmap ----------------
heatmap, preds = make_gradcam_heatmap(img_array)

print("Heatmap shape:", heatmap.shape)
print("Heatmap min/max:", heatmap.min(), heatmap.max())

# ---------------- Resize safely ----------------
heatmap_resized = cv2.resize(
    heatmap.astype(np.float32),
    (IMG_SIZE, IMG_SIZE),
    interpolation=cv2.INTER_LINEAR
)

heatmap_u8 = np.uint8(255 * heatmap_resized)
heatmap_color = cv2.applyColorMap(heatmap_u8, cv2.COLORMAP_JET)
heatmap_color = cv2.cvtColor(heatmap_color, cv2.COLOR_BGR2RGB)

# ---------------- Overlay ----------------
alpha = 0.4
overlay = (img_rgb * (1 - alpha) + heatmap_color * alpha).astype(np.uint8)

pred_class = class_names[int(np.argmax(preds[0]))]
print("Predicted:", pred_class)
print("Probabilities:", preds[0])

# ---------------- Display ----------------
plt.figure(figsize=(12,4))

plt.subplot(1,3,1)
plt.imshow(img_rgb)
plt.title("Original")
plt.axis("off")

plt.subplot(1,3,2)
plt.imshow(heatmap_resized, cmap="jet")
plt.title("Heatmap")
plt.axis("off")

plt.subplot(1,3,3)
plt.imshow(overlay)
plt.title(f"Overlay (Pred: {pred_class})")
plt.axis("off")

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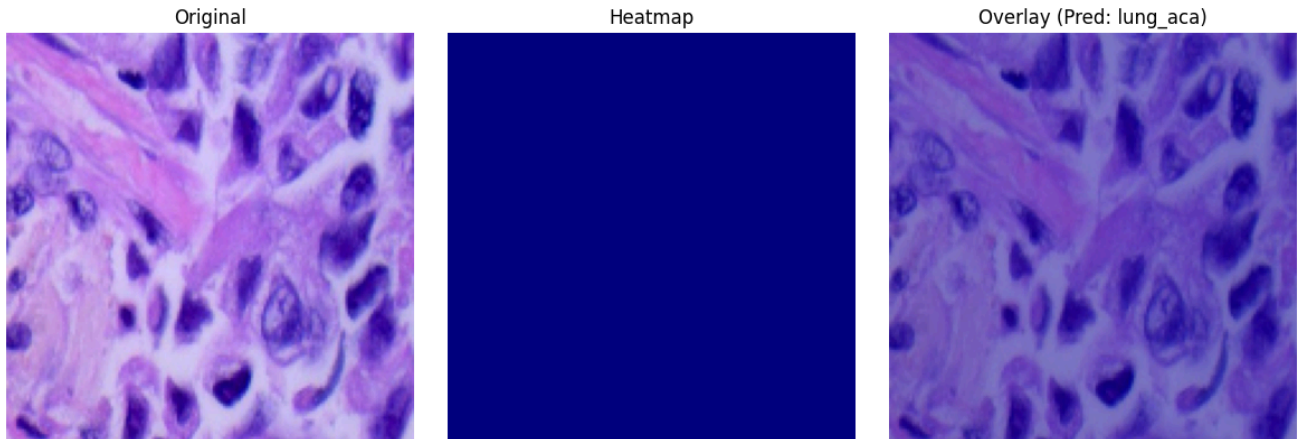
```
plt.show()
```

```
Using image: /content/drive/MyDrive/LungCancer_Project/split_70_15_15/test/lung_aca/lungaca1005.jpeg
Using TARGET_LAYER: top_conv
Heatmap shape: (5, 5)
Heatmap min/max: 0.0 0.0
Predicted: lung_aca
Probabilities: [9.9999857e-01 6.8156417e-08 1.3230793e-06]
```



```python
# ============================================================
# IMPROVED Grad-CAM — log(prob) target + safe normalization
# ============================================================

import os, glob, numpy as np, tensorflow as tf, cv2
import matplotlib.pyplot as plt

chosen_class = "lung_aca"
img_path = sorted(glob.glob(os.path.join(TEST_DIR, chosen_class, "*.jpeg")))[0]
print("Using image:", img_path)

TARGET_LAYER = "top_conv"

grad_model = tf.keras.models.Model(
    inputs=best_model.inputs,
    outputs=[best_model.get_layer(TARGET_LAYER).output, best_model.output]
)

def make_gradcam_heatmap_logprob(img_array, eps=1e-8):
    with tf.GradientTape() as tape:
        conv_outputs, preds = grad_model(img_array, training=False)
        class_idx = tf.argmax(preds[0])
        # Use log(prob) to avoid gradient collapse at extreme probs
        score = tf.math.log(preds[:, class_idx] + eps)

    grads = tape.gradient(score, conv_outputs)

    conv_outputs = conv_outputs[0].numpy()
    grads = grads[0].numpy()

    weights = np.mean(grads, axis=(0, 1))
    heatmap = np.sum(weights * conv_outputs, axis=-1)

    # Use absolute value before ReLU to avoid all-zero case
    heatmap = np.abs(heatmap)

    # Normalize safely
    heatmap -= heatmap.min()
    denom = heatmap.max() + eps
    heatmap = heatmap / denom

    return heatmap, preds.numpy()

# Read image
img_bgr = cv2.imread(img_path)
img_bgr = cv2.resize(img_bgr, (IMG_SIZE, IMG_SIZE))
```

```
img_bgr = cv2.resize(img_bgr, (IMG_SIZE, IMG_SIZE))
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

img_array = np.expand_dims(img_rgb.astype(np.float32), axis=0)
img_array = preprocess(img_array)

heatmap, preds = make_gradcam_heatmap_logprob(img_array)

print("Heatmap shape:", heatmap.shape)
```