



In [2]: **import** kagglehub

```
# Download latest version
path = kagglehub.dataset_download("abhinav099802/eye-disease-image-dataset")

print("Path to dataset files:", path)
```

Downloading from https://www.kaggle.com/api/v1/datasets/download/abhinav099802/eye-disease-image-dataset?dataset_version_number=3...

100%|██████████| 3.67G/3.67G [00:39<00:00, 99.2MB/s]

Extracting files...

Path to dataset files: /root/.cache/kagglehub/datasets/abhinav099802/eye-disease-image-dataset/versions/3

In []:

```
In [5]: import os
import shutil
from sklearn.model_selection import train_test_split

dataset_root = path # This is /root/.cache/kagglehub/.../versions/3

# Define the base directory where images are stored
original_data_path = os.path.join(dataset_root, 'data', 'Original_Dataset')
augmented_data_path = os.path.join(dataset_root, 'data', 'Augmented_Dataset')

# Get the list of actual classes (e.g., 'Cataract', 'Normal')
# We can get them from either Original_Dataset or Augmented_Dataset, assuming
if os.path.exists(original_data_path):
    classes = os.listdir(original_data_path)
elif os.path.exists(augmented_data_path):
    classes = os.listdir(augmented_data_path)
else:
    raise FileNotFoundError("Neither Original_Dataset nor Augmented_Dataset found")

base_split = "/content/eye_split"

# Create target directories for train, val, test splits for each class
for split_type in ["train", "val", "test"]:
    for cls in classes:
        os.makedirs(os.path.join(base_split, split_type, cls), exist_ok=True)

# Process each class
for cls in classes:
    all_images_for_class = []

    # Collect images from Original_Dataset for the current class
    original_class_path = os.path.join(original_data_path, cls)
    if os.path.exists(original_class_path):
        for img_name in os.listdir(original_class_path):
            all_images_for_class.append(os.path.join(original_class_path, img_name))

    # Collect images from Augmented_Dataset for the current class
```

```

augmented_class_path = os.path.join(augmented_data_path, cls)
if os.path.exists(augmented_class_path):
    for img_name in os.listdir(augmented_class_path):
        all_images_for_class.append(os.path.join(augmented_class_path, img_name))

if not all_images_for_class:
    print(f"No images found for class: {cls}. Skipping split for this class")
    continue

# Perform the train/val/test split on the collected image paths
# Ensure there are enough samples for splitting
if len(all_images_for_class) < 3: # Need at least 3 for 10% test and 10% val
    print(f"Warning: Not enough images ({len(all_images_for_class)}) for class {cls}")
    train_images = all_images_for_class
    val_images = []
    test_images = []
else:
    # First split into train_val and test
    train_val_images, test_images = train_test_split(all_images_for_class, test_size=0.1)
    # Then split train_val into train and val (val will be 10% of original)
    train_images, val_images = train_test_split(train_val_images, test_size=0.1)

# Copy images to their respective split directories
for img_path in train_images:
    shutil.copy(img_path, os.path.join(base_split, "train", cls, os.path.basename(img_path)))

for img_path in val_images:
    shutil.copy(img_path, os.path.join(base_split, "val", cls, os.path.basename(img_path)))

for img_path in test_images:
    shutil.copy(img_path, os.path.join(base_split, "test", cls, os.path.basename(img_path)))

print("Dataset splitting complete.")

```

Dataset splitting complete.

DATA AUGMENTATION

In [9]: **from** tensorflow.keras.preprocessing.image **import** ImageDataGenerator

```

IMG_SIZE = (224, 224)
BATCH = 32

train_gen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True
).flow_from_directory(
    base_split + "/train",

```

```

        target_size=IMG_SIZE,
        batch_size=BATCH,
        class_mode="categorical"
    )

    val_gen = ImageDataGenerator(rescale=1./255).flow_from_directory(
        base_split + "/val",
        target_size=IMG_SIZE,
        batch_size=BATCH,
        class_mode="categorical"
    )

    test_gen = ImageDataGenerator(rescale=1./255).flow_from_directory(
        base_split + "/test",
        target_size=IMG_SIZE,
        batch_size=BATCH,
        class_mode="categorical",
        shuffle=False
    )

```

Found 13865 images belonging to 11 classes.

Found 2109 images belonging to 11 classes.

Found 2102 images belonging to 11 classes.

In [6]: **from** tensorflow.keras.preprocessing.image **import** ImageDataGenerator

```

IMG_SIZE = (224, 224)
BATCH = 32

train_gen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    zoom_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True
).flow_from_directory(
    base_split + "/train",
    target_size=IMG_SIZE,
    batch_size=BATCH,
    class_mode="categorical"
)

val_gen = ImageDataGenerator(rescale=1./255).flow_from_directory(
    base_split + "/val",
    target_size=IMG_SIZE,
    batch_size=BATCH,
    class_mode="categorical"
)

test_gen = ImageDataGenerator(rescale=1./255).flow_from_directory(
    base_split + "/test",
    target_size=IMG_SIZE,
    batch_size=BATCH,

```

```

        class_mode="categorical",
        shuffle=False
    )

```

Found 13865 images belonging to 11 classes.

Found 2109 images belonging to 11 classes.

Found 2102 images belonging to 11 classes.

```

In [7]: from tensorflow.keras.applications import MobileNetV2
        from tensorflow.keras.layers import Dense, Dropout, GlobalAveragePooling2D
        from tensorflow.keras.models import Model
        from tensorflow.keras.optimizers import Adam

        base = MobileNetV2(weights="imagenet", include_top=False, input_shape=(224,224,3))
        base.trainable = False # freeze first

        x = GlobalAveragePooling2D()(base.output)
        x = Dense(256, activation="relu")(x)
        x = Dropout(0.5)(x)
        output = Dense(train_gen.num_classes, activation="softmax")(x)

        model = Model(inputs=base.input, outputs=output)

        model.compile(
            optimizer=Adam(1e-3),
            loss="categorical_crossentropy",
            metrics=["accuracy"]
        )

        model.summary()

```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet_v2/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_1.0_224_no_top.h5

9406464/9406464 ————— 0s 0us/step

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 224, 224, 3)	0	-
Conv1 (Conv2D)	(None, 112, 112, 32)	864	input_layer[0][0]
bn_Conv1 (BatchNormalizatio...	(None, 112, 112, 32)	128	Conv1[0][0]
Conv1_relu (ReLU)	(None, 112, 112, 32)	0	bn_Conv1[0][0]
expanded_conv_dept... (DepthwiseConv2D)	(None, 112, 112, 32)	288	Conv1_relu[0][0]
expanded_conv_dept... (BatchNormalizatio...	(None, 112, 112, 32)	128	expanded_conv_de...
expanded_conv_dept... (ReLU)	(None, 112, 112, 32)	0	expanded_conv_de...
expanded_conv_proj... (Conv2D)	(None, 112, 112, 16)	512	expanded_conv_de...
expanded_conv_proj... (BatchNormalizatio...	(None, 112, 112, 16)	64	expanded_conv_pr...
block_1_expand (Conv2D)	(None, 112, 112, 96)	1,536	expanded_conv_pr...
block_1_expand_BN (BatchNormalizatio...	(None, 112, 112, 96)	384	block_1_expand[0...
block_1_expand_relu (ReLU)	(None, 112, 112, 96)	0	block_1_expand_B...
block_1_pad (ZeroPadding2D)	(None, 113, 113, 96)	0	block_1_expand_r...
block_1_depthwise (DepthwiseConv2D)	(None, 56, 56, 96)	864	block_1_pad[0][0]
block_1_depthwise_... (BatchNormalizatio...	(None, 56, 56, 96)	384	block_1_depthwis...
block_1_depthwise_... (ReLU)	(None, 56, 56, 96)	0	block_1_depthwis...
block_1_project (Conv2D)	(None, 56, 56, 24)	2,304	block_1_depthwis...
block_1_project_BN (BatchNormalizatio...	(None, 56, 56, 24)	96	block_1_project[...

block_2_expand (Conv2D)	(None, 56, 56, 144)	3,456	block_1_project_...
block_2_expand_BN (BatchNormalizatio...	(None, 56, 56, 144)	576	block_2_expand[0...
block_2_expand_relu (ReLU)	(None, 56, 56, 144)	0	block_2_expand_B...
block_2_depthwise (DepthwiseConv2D)	(None, 56, 56, 144)	1,296	block_2_expand_r...
block_2_depthwise_... (BatchNormalizatio...	(None, 56, 56, 144)	576	block_2_depthwis...
block_2_depthwise_... (ReLU)	(None, 56, 56, 144)	0	block_2_depthwis...
block_2_project (Conv2D)	(None, 56, 56, 24)	3,456	block_2_depthwis...
block_2_project_BN (BatchNormalizatio...	(None, 56, 56, 24)	96	block_2_project[...
block_2_add (Add)	(None, 56, 56, 24)	0	block_1_project_... block_2_project_...
block_3_expand (Conv2D)	(None, 56, 56, 144)	3,456	block_2_add[0][0]
block_3_expand_BN (BatchNormalizatio...	(None, 56, 56, 144)	576	block_3_expand[0...
block_3_expand_relu (ReLU)	(None, 56, 56, 144)	0	block_3_expand_B...
block_3_pad (ZeroPadding2D)	(None, 57, 57, 144)	0	block_3_expand_r...
block_3_depthwise (DepthwiseConv2D)	(None, 28, 28, 144)	1,296	block_3_pad[0][0]
block_3_depthwise_... (BatchNormalizatio...	(None, 28, 28, 144)	576	block_3_depthwis...
block_3_depthwise_... (ReLU)	(None, 28, 28, 144)	0	block_3_depthwis...
block_3_project (Conv2D)	(None, 28, 28, 32)	4,608	block_3_depthwis...
block_3_project_BN (BatchNormalizatio...	(None, 28, 28, 32)	128	block_3_project[...
block_4_expand	(None, 28, 28,	6,144	block_3_project_...

(Conv2D)	192)		
block_4_expand_BN (BatchNormalizatio...	(None, 28, 28, 192)	768	block_4_expand[0...
block_4_expand_relu (ReLU)	(None, 28, 28, 192)	0	block_4_expand_B...
block_4_depthwise (DepthwiseConv2D)	(None, 28, 28, 192)	1,728	block_4_expand_r...
block_4_depthwise_... (BatchNormalizatio...	(None, 28, 28, 192)	768	block_4_depthwis...
block_4_depthwise_... (ReLU)	(None, 28, 28, 192)	0	block_4_depthwis...
block_4_project (Conv2D)	(None, 28, 28, 32)	6,144	block_4_depthwis...
block_4_project_BN (BatchNormalizatio...	(None, 28, 28, 32)	128	block_4_project[...
block_4_add (Add)	(None, 28, 28, 32)	0	block_3_project_... block_4_project_...
block_5_expand (Conv2D)	(None, 28, 28, 192)	6,144	block_4_add[0][0]
block_5_expand_BN (BatchNormalizatio...	(None, 28, 28, 192)	768	block_5_expand[0...
block_5_expand_relu (ReLU)	(None, 28, 28, 192)	0	block_5_expand_B...
block_5_depthwise (DepthwiseConv2D)	(None, 28, 28, 192)	1,728	block_5_expand_r...
block_5_depthwise_... (BatchNormalizatio...	(None, 28, 28, 192)	768	block_5_depthwis...
block_5_depthwise_... (ReLU)	(None, 28, 28, 192)	0	block_5_depthwis...
block_5_project (Conv2D)	(None, 28, 28, 32)	6,144	block_5_depthwis...
block_5_project_BN (BatchNormalizatio...	(None, 28, 28, 32)	128	block_5_project[...
block_5_add (Add)	(None, 28, 28, 32)	0	block_4_add[0][0... block_5_project_...
block_6_expand (Conv2D)	(None, 28, 28, 192)	6,144	block_5_add[0][0]

block_6_expand_BN (BatchNormalizatio...	(None, 28, 28, 192)	768	block_6_expand[0...
block_6_expand_relu (ReLU)	(None, 28, 28, 192)	0	block_6_expand_B...
block_6_pad (ZeroPadding2D)	(None, 29, 29, 192)	0	block_6_expand_r...
block_6_depthwise (DepthwiseConv2D)	(None, 14, 14, 192)	1,728	block_6_pad[0][0]
block_6_depthwise_...	(None, 14, 14, 192)	768	block_6_depthwis...
block_6_depthwise_...	(None, 14, 14, 192)	0	block_6_depthwis...
block_6_project (Conv2D)	(None, 14, 14, 64)	12,288	block_6_depthwis...
block_6_project_BN (BatchNormalizatio...	(None, 14, 14, 64)	256	block_6_project[...
block_7_expand (Conv2D)	(None, 14, 14, 384)	24,576	block_6_project_...
block_7_expand_BN (BatchNormalizatio...	(None, 14, 14, 384)	1,536	block_7_expand[0...
block_7_expand_relu (ReLU)	(None, 14, 14, 384)	0	block_7_expand_B...
block_7_depthwise (DepthwiseConv2D)	(None, 14, 14, 384)	3,456	block_7_expand_r...
block_7_depthwise_...	(None, 14, 14, 384)	1,536	block_7_depthwis...
block_7_depthwise_...	(None, 14, 14, 384)	0	block_7_depthwis...
block_7_project (Conv2D)	(None, 14, 14, 64)	24,576	block_7_depthwis...
block_7_project_BN (BatchNormalizatio...	(None, 14, 14, 64)	256	block_7_project[...
block_7_add (Add)	(None, 14, 14, 64)	0	block_6_project_... block_7_project_...
block_8_expand (Conv2D)	(None, 14, 14, 384)	24,576	block_7_add[0][0]
block_8_expand_BN (BatchNormalizatio...	(None, 14, 14, 384)	1,536	block_8_expand[0...

block_8_expand_relu (ReLU)	(None, 14, 14, 384)	0	block_8_expand_B...
block_8_depthwise (DepthwiseConv2D)	(None, 14, 14, 384)	3,456	block_8_expand_r...
block_8_depthwise_... (BatchNormalizatio...	(None, 14, 14, 384)	1,536	block_8_depthwis...
block_8_depthwise_... (ReLU)	(None, 14, 14, 384)	0	block_8_depthwis...
block_8_project (Conv2D)	(None, 14, 14, 64)	24,576	block_8_depthwis...
block_8_project_BN (BatchNormalizatio...	(None, 14, 14, 64)	256	block_8_project[...
block_8_add (Add)	(None, 14, 14, 64)	0	block_7_add[0][0... block_8_project_...
block_9_expand (Conv2D)	(None, 14, 14, 384)	24,576	block_8_add[0][0]
block_9_expand_BN (BatchNormalizatio...	(None, 14, 14, 384)	1,536	block_9_expand[0...
block_9_expand_relu (ReLU)	(None, 14, 14, 384)	0	block_9_expand_B...
block_9_depthwise (DepthwiseConv2D)	(None, 14, 14, 384)	3,456	block_9_expand_r...
block_9_depthwise_... (BatchNormalizatio...	(None, 14, 14, 384)	1,536	block_9_depthwis...
block_9_depthwise_... (ReLU)	(None, 14, 14, 384)	0	block_9_depthwis...
block_9_project (Conv2D)	(None, 14, 14, 64)	24,576	block_9_depthwis...
block_9_project_BN (BatchNormalizatio...	(None, 14, 14, 64)	256	block_9_project[...
block_9_add (Add)	(None, 14, 14, 64)	0	block_8_add[0][0... block_9_project_...
block_10_expand (Conv2D)	(None, 14, 14, 384)	24,576	block_9_add[0][0]
block_10_expand_BN (BatchNormalizatio...	(None, 14, 14, 384)	1,536	block_10_expand[...
block_10_expand_re...	(None, 14, 14,	0	block_10_expand_...

(ReLU)	384)		
block_10_depthwise (DepthwiseConv2D)	(None, 14, 14, 384)	3,456	block_10_expand_...
block_10_depthwise... (BatchNormalizatio...	(None, 14, 14, 384)	1,536	block_10_depthwi...
block_10_depthwise... (ReLU)	(None, 14, 14, 384)	0	block_10_depthwi...
block_10_project (Conv2D)	(None, 14, 14, 96)	36,864	block_10_depthwi...
block_10_project_BN (BatchNormalizatio...	(None, 14, 14, 96)	384	block_10_project...
block_11_expand (Conv2D)	(None, 14, 14, 576)	55,296	block_10_project...
block_11_expand_BN (BatchNormalizatio...	(None, 14, 14, 576)	2,304	block_11_expand[...
block_11_expand_re... (ReLU)	(None, 14, 14, 576)	0	block_11_expand_...
block_11_depthwise (DepthwiseConv2D)	(None, 14, 14, 576)	5,184	block_11_expand_...
block_11_depthwise... (BatchNormalizatio...	(None, 14, 14, 576)	2,304	block_11_depthwi...
block_11_depthwise... (ReLU)	(None, 14, 14, 576)	0	block_11_depthwi...
block_11_project (Conv2D)	(None, 14, 14, 96)	55,296	block_11_depthwi...
block_11_project_BN (BatchNormalizatio...	(None, 14, 14, 96)	384	block_11_project...
block_11_add (Add)	(None, 14, 14, 96)	0	block_10_project... block_11_project...
block_12_expand (Conv2D)	(None, 14, 14, 576)	55,296	block_11_add[0][...
block_12_expand_BN (BatchNormalizatio...	(None, 14, 14, 576)	2,304	block_12_expand[...
block_12_expand_re... (ReLU)	(None, 14, 14, 576)	0	block_12_expand_...
block_12_depthwise (DepthwiseConv2D)	(None, 14, 14, 576)	5,184	block_12_expand_...

block_12_depthwise... (BatchNormalizatio...	(None, 14, 14, 576)	2,304	block_12_depthwi...
block_12_depthwise... (ReLU)	(None, 14, 14, 576)	0	block_12_depthwi...
block_12_project (Conv2D)	(None, 14, 14, 96)	55,296	block_12_depthwi...
block_12_project_BN (BatchNormalizatio...	(None, 14, 14, 96)	384	block_12_project...
block_12_add (Add)	(None, 14, 14, 96)	0	block_11_add[0][... block_12_project...
block_13_expand (Conv2D)	(None, 14, 14, 576)	55,296	block_12_add[0][...
block_13_expand_BN (BatchNormalizatio...	(None, 14, 14, 576)	2,304	block_13_expand[...
block_13_expand_re... (ReLU)	(None, 14, 14, 576)	0	block_13_expand_...
block_13_pad (ZeroPadding2D)	(None, 15, 15, 576)	0	block_13_expand_...
block_13_depthwise (DepthwiseConv2D)	(None, 7, 7, 576)	5,184	block_13_pad[0][...
block_13_depthwise... (BatchNormalizatio...	(None, 7, 7, 576)	2,304	block_13_depthwi...
block_13_depthwise... (ReLU)	(None, 7, 7, 576)	0	block_13_depthwi...
block_13_project (Conv2D)	(None, 7, 7, 160)	92,160	block_13_depthwi...
block_13_project_BN (BatchNormalizatio...	(None, 7, 7, 160)	640	block_13_project...
block_14_expand (Conv2D)	(None, 7, 7, 960)	153,600	block_13_project...
block_14_expand_BN (BatchNormalizatio...	(None, 7, 7, 960)	3,840	block_14_expand[...
block_14_expand_re... (ReLU)	(None, 7, 7, 960)	0	block_14_expand_...
block_14_depthwise (DepthwiseConv2D)	(None, 7, 7, 960)	8,640	block_14_expand_...
block_14_depthwise... (BatchNormalizatio...	(None, 7, 7, 960)	3,840	block_14_depthwi...

block_14_depthwise... (ReLU)	(None, 7, 7, 960)	0	block_14_depthwi...
block_14_project (Conv2D)	(None, 7, 7, 160)	153,600	block_14_depthwi...
block_14_project_BN (BatchNormalizatio...	(None, 7, 7, 160)	640	block_14_project...
block_14_add (Add)	(None, 7, 7, 160)	0	block_13_project... block_14_project...
block_15_expand (Conv2D)	(None, 7, 7, 960)	153,600	block_14_add[0][...
block_15_expand_BN (BatchNormalizatio...	(None, 7, 7, 960)	3,840	block_15_expand[...
block_15_expand_re... (ReLU)	(None, 7, 7, 960)	0	block_15_expand_...
block_15_depthwise (DepthwiseConv2D)	(None, 7, 7, 960)	8,640	block_15_expand_...
block_15_depthwise... (BatchNormalizatio...	(None, 7, 7, 960)	3,840	block_15_depthwi...
block_15_depthwise... (ReLU)	(None, 7, 7, 960)	0	block_15_depthwi...
block_15_project (Conv2D)	(None, 7, 7, 160)	153,600	block_15_depthwi...
block_15_project_BN (BatchNormalizatio...	(None, 7, 7, 160)	640	block_15_project...
block_15_add (Add)	(None, 7, 7, 160)	0	block_14_add[0][...] block_15_project...
block_16_expand (Conv2D)	(None, 7, 7, 960)	153,600	block_15_add[0][...
block_16_expand_BN (BatchNormalizatio...	(None, 7, 7, 960)	3,840	block_16_expand[...
block_16_expand_re... (ReLU)	(None, 7, 7, 960)	0	block_16_expand_...
block_16_depthwise (DepthwiseConv2D)	(None, 7, 7, 960)	8,640	block_16_expand_...
block_16_depthwise... (BatchNormalizatio...	(None, 7, 7, 960)	3,840	block_16_depthwi...
block_16_depthwise...	(None, 7, 7, 960)	0	block_16_depthwi...

(ReLU)			
block_16_project (Conv2D)	(None, 7, 7, 320)	307,200	block_16_depthwi...
block_16_project_BN (BatchNormalizatio...	(None, 7, 7, 320)	1,280	block_16_project...
Conv_1 (Conv2D)	(None, 7, 7, 1280)	409,600	block_16_project...
Conv_1_bn (BatchNormalizatio...	(None, 7, 7, 1280)	5,120	Conv_1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	Conv_1_bn[0][0]
global_average_poo... (GlobalAveragePool...	(None, 1280)	0	out_relu[0][0]
dense (Dense)	(None, 256)	327,936	global_average_p...
dropout (Dropout)	(None, 256)	0	dense[0][0]
dense_1 (Dense)	(None, 11)	2,827	dropout[0][0]

Total params: 2,588,747 (9.88 MB)

Trainable params: 330,763 (1.26 MB)

Non-trainable params: 2,257,984 (8.61 MB)

```
In [11]: history = model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=10
)
```

Epoch 1/10
434/434 ————— **1081s** 2s/step - accuracy: 0.6497 - loss: 0.9454 - val_accuracy: 0.7221 - val_loss: 0.7476
Epoch 2/10
434/434 ————— **1065s** 2s/step - accuracy: 0.6521 - loss: 0.9253 - val_accuracy: 0.7018 - val_loss: 0.8011
Epoch 3/10
434/434 ————— **1046s** 2s/step - accuracy: 0.6573 - loss: 0.9160 - val_accuracy: 0.7041 - val_loss: 0.7583
Epoch 4/10
434/434 ————— **1049s** 2s/step - accuracy: 0.6657 - loss: 0.9042 - val_accuracy: 0.7122 - val_loss: 0.7780
Epoch 5/10
434/434 ————— **1088s** 3s/step - accuracy: 0.6554 - loss: 0.9152 - val_accuracy: 0.7250 - val_loss: 0.7403
Epoch 6/10
434/434 ————— **1054s** 2s/step - accuracy: 0.6602 - loss: 0.9041 - val_accuracy: 0.7131 - val_loss: 0.7515
Epoch 7/10
434/434 ————— **1055s** 2s/step - accuracy: 0.6619 - loss: 0.8886 - val_accuracy: 0.7136 - val_loss: 0.7332
Epoch 8/10
434/434 ————— **1040s** 2s/step - accuracy: 0.6683 - loss: 0.8901 - val_accuracy: 0.7141 - val_loss: 0.7336
Epoch 9/10
434/434 ————— **1098s** 3s/step - accuracy: 0.6583 - loss: 0.9050 - val_accuracy: 0.7112 - val_loss: 0.7502
Epoch 10/10
434/434 ————— **1054s** 2s/step - accuracy: 0.6724 - loss: 0.8724 - val_accuracy: 0.7236 - val_loss: 0.7243

```
In [12]: test_loss, test_acc = model.evaluate(test_gen)
         print(f"Test Accuracy = {test_acc*100:.2f}%")
```

66/66 ————— **121s** 2s/step - accuracy: 0.7283 - loss: 0.6986
Test Accuracy = 71.50%

```
In [13]: import numpy as np
         from sklearn.metrics import confusion_matrix
         import seaborn as sns
         import matplotlib.pyplot as plt

         true_labels = test_gen.classes
         pred_prob = model.predict(test_gen)
         pred_labels = np.argmax(pred_prob, axis=1)

         class_names = list(test_gen.class_indices.keys())
         labels = list(range(len(class_names)))

         cm = confusion_matrix(true_labels, pred_labels, labels=labels)
         cm_norm = cm.astype("float") / cm.sum(axis=1)[:, np.newaxis]

         plt.figure(figsize=(12,9))
         sns.heatmap(cm_norm, annot=True, cmap="Blues", fmt=".2f",
```

```

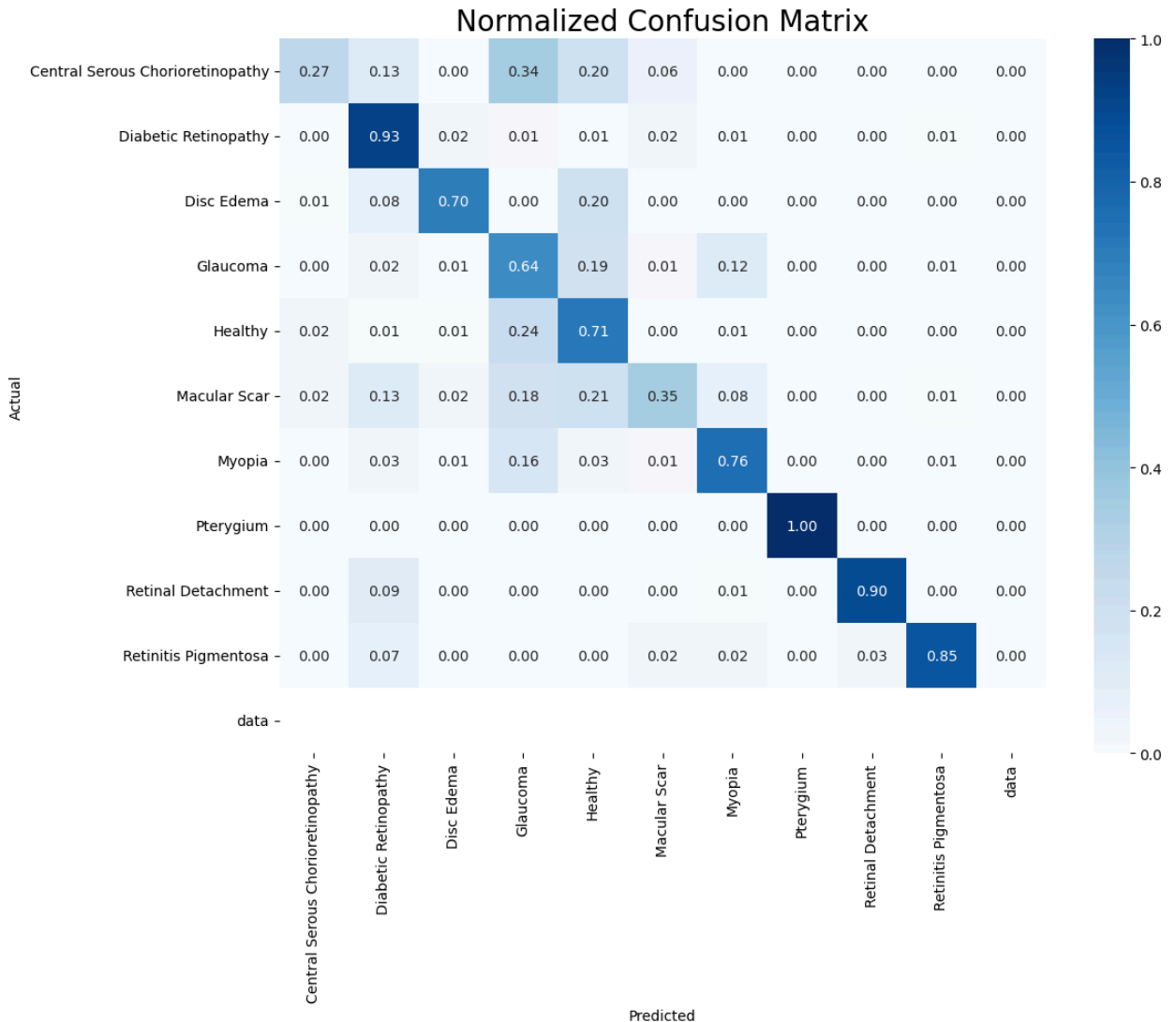
xticklabels=class_names, yticklabels=class_names)
plt.title("Normalized Confusion Matrix", fontsize=20)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

```

66/66 ————— 126s 2s/step

/tmp/ipython-input-1982483493.py:14: RuntimeWarning: invalid value encountered in divide

```
cm_norm = cm.astype("float") / cm.sum(axis=1)[:, np.newaxis]
```



```

In [15]: from sklearn.metrics import classification_report
import pandas as pd
import numpy as np

# Get the mapping from class index to class name
idx_to_class = {v: k for k, v in test_gen.class_indices.items()}

# Get the unique labels present in the test set's true labels
unique_true_labels = np.unique(true_labels)

```

```

# Create a list of target names corresponding to these unique labels
# Ensure the order is consistent with the label indices
report_target_names = [idx_to_class[i] for i in sorted(unique_true_labels)]

report = classification_report(true_labels, pred_labels, labels=unique_true_labels)
df = pd.DataFrame(report).transpose().round(3)

df.style.background_gradient(cmap="Blues").set_properties(**{'text-align': 'center'})

```

Out[15]:

	precision	recall	f1-score	support
Central Serous Chorioretinopathy	0.559000	0.271000	0.365000	70.000000
Diabetic Retinopathy	0.847000	0.929000	0.886000	481.000000
Disc Edema	0.747000	0.705000	0.725000	88.000000
Glaucoma	0.570000	0.643000	0.604000	412.000000
Healthy	0.598000	0.713000	0.650000	352.000000
Macular Scar	0.776000	0.355000	0.487000	234.000000
Myopia	0.733000	0.757000	0.745000	272.000000
Pterygium	1.000000	1.000000	1.000000	11.000000
Retinal Detachment	0.962000	0.895000	0.928000	86.000000
Retinitis Pigmentosa	0.882000	0.854000	0.868000	96.000000
accuracy	0.715000	0.715000	0.715000	0.715000
macro avg	0.767000	0.712000	0.726000	2102.000000
weighted avg	0.721000	0.715000	0.706000	2102.000000