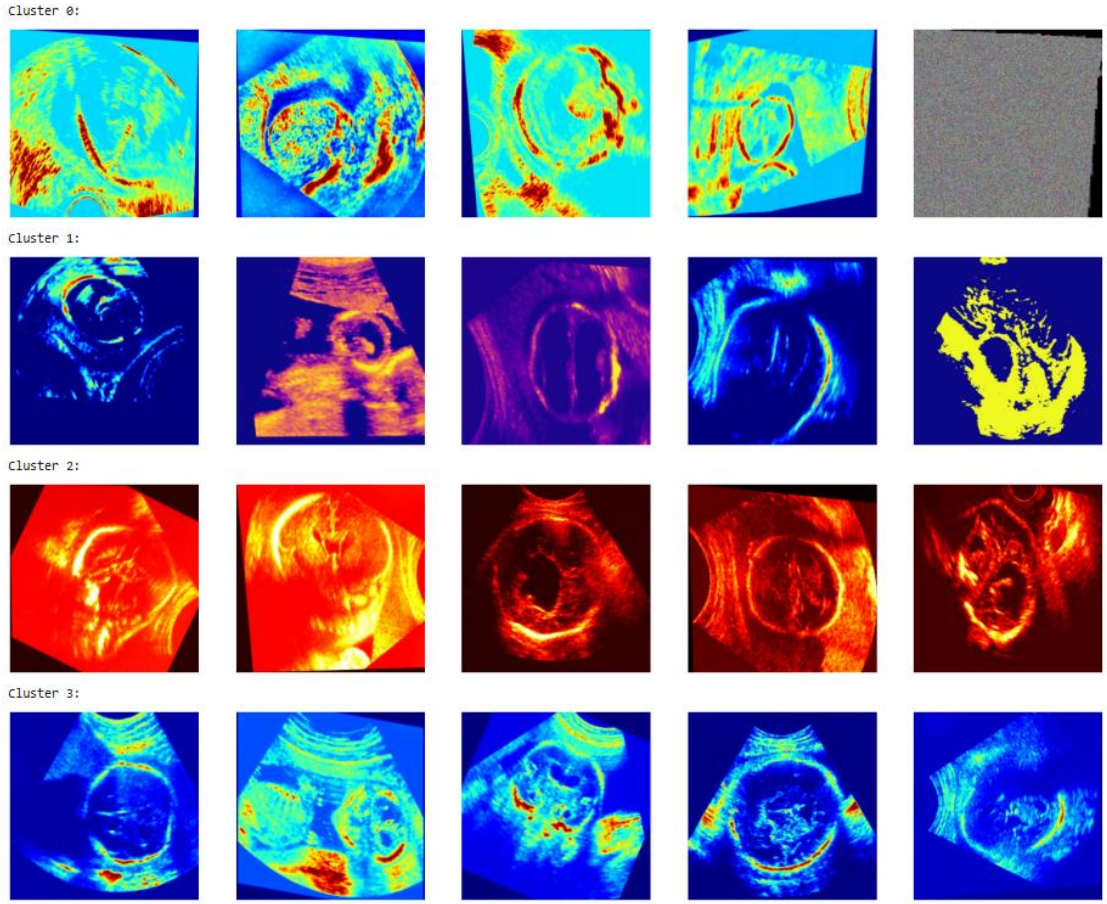
Medical Image Classification Using Multi-Head Attention and LSTM




```
import numpy as np
import pandas as pd
import os

base_path = '/kaggle/input/medical-imaging-fetal-colorized-new-dataset-umrict/Fetal Head Abnormalities Classification/Fetal Head Abnormalities Classification/data'

categories = [
    "3D_Rendering",
    "3D_Volume_Rendering",
    "Adaptive_Histogram_Equalization",
    "Alpha_Blending",
    "Basic_Color_Map",
    "Contrast_Stretching",
    "Edge_Detection",
    "Gamma_Correction",
    "Gaussian_Blur",
    "Heatmap_Visualization",
```

```
    "Interactive_Segmentation",
    "LUT_Color_Map",
    "Random_Color_Palette"
]

image_paths = []
labels = []

for category in categories:
    category_path = os.path.join(base_path, category)
    for image_name in os.listdir(category_path):
        image_path = os.path.join(category_path, image_name)
        image_paths.append(image_path)
        labels.append(category)

df = pd.DataFrame({
    "image_path": image_paths,
    "label": labels
})

df.head()
```

	image_path	label
0	/kaggle/input/medical-imaging-fetal-colorized-...	3D_Rendering
1	/kaggle/input/medical-imaging-fetal-colorized-...	3D_Rendering
2	/kaggle/input/medical-imaging-fetal-colorized-...	3D_Rendering
3	/kaggle/input/medical-imaging-fetal-colorized-...	3D_Rendering
4	/kaggle/input/medical-imaging-fetal-colorized-...	3D_Rendering

```
df.tail()
```

label	image_path
12657	/kaggle/input/medical-imaging-fetal-colorized-...
Random_Color_Palette	
12658	/kaggle/input/medical-imaging-fetal-colorized-...
Random_Color_Palette	
12659	/kaggle/input/medical-imaging-fetal-colorized-...
Random_Color_Palette	
12660	/kaggle/input/medical-imaging-fetal-colorized-...
Random_Color_Palette	
12661	/kaggle/input/medical-imaging-fetal-colorized-...
Random_Color_Palette	

```
df.shape
```

```
(12662, 2)
```

```
df.columns
```

```
Index(['image_path', 'label'], dtype='object')
```

```
df.duplicated().sum()
```

```
0
```

```
df.isnull().sum()
```

```
image_path    0
label         0
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12662 entries, 0 to 12661
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   image_path  12662 non-null  object
 1   label       12662 non-null  object
dtypes: object(2)
memory usage: 198.0+ KB
```

```
df['label'].unique()
```

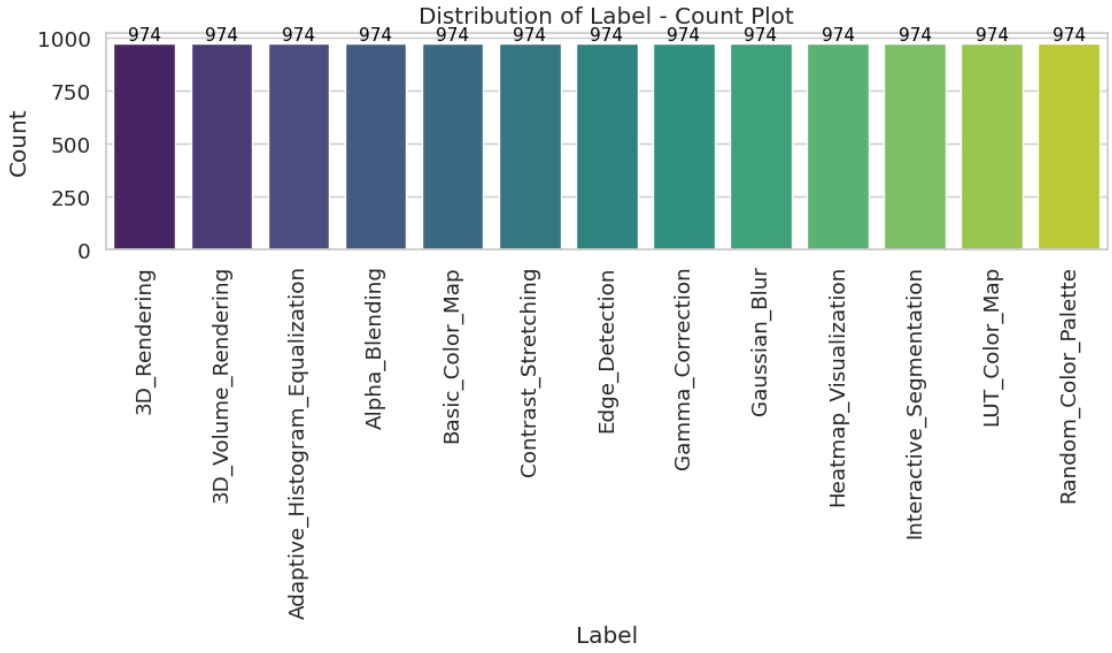
```
array(['3D_Rendering', '3D_Volume_Rendering',
       'Adaptive_Histogram_Equalization', 'Alpha_Blending',
       'Basic_Color_Map', 'Contrast_Stretching', 'Edge_Detection',
       'Gamma_Correction', 'Gaussian_Blur', 'Heatmap_Visualization',
       'Interactive_Segmentation', 'LUT_Color_Map',
       'Random_Color_Palette'], dtype=object)
```

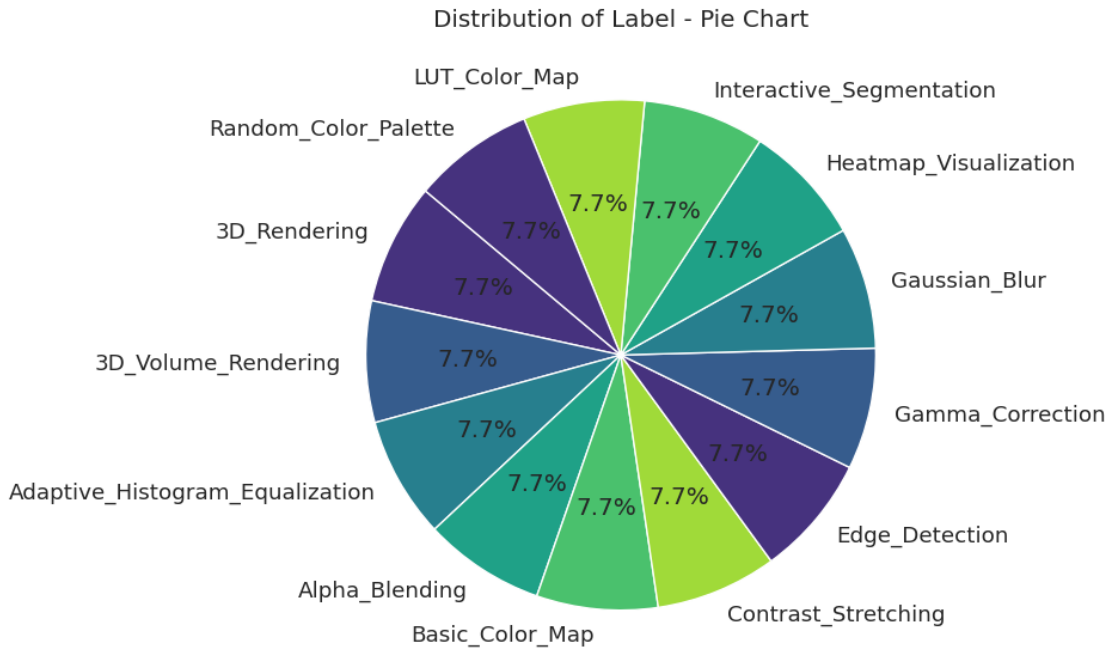
```
df['label'].value_counts()
```

```
label
3D_Rendering                        974
3D_Volume_Rendering                 974
Adaptive_Histogram_Equalization     974
Alpha_Blending                      974
Basic_Color_Map                     974
Contrast_Stretching                 974
Edge_Detection                      974
Gamma_Correction                    974
Gaussian_Blur                       974
Heatmap_Visualization               974
Interactive_Segmentation            974
LUT_Color_Map                       974
Random_Color_Palette                974
Name: count, dtype: int64
```

```
import seaborn as sns
import matplotlib.pyplot as plt
```

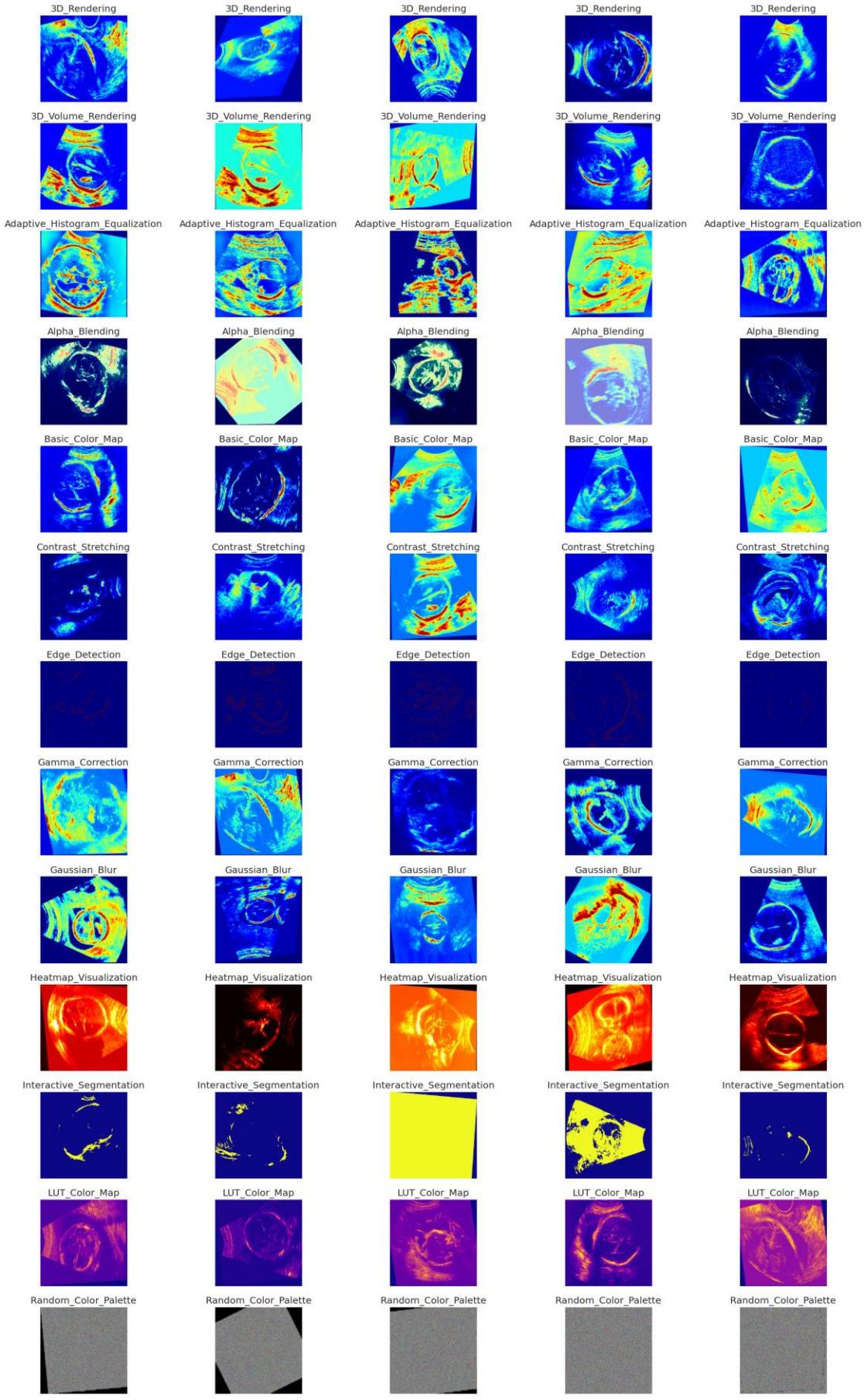
```
def visualize_label_distribution(df, label_column="label", figsize=(10, 6),
palette="viridis"):
    """
    Visualizes the distribution of labels in a DataFrame using count and pie
charts.

    Args:
        df (pd.DataFrame): The DataFrame containing the label data.
        label_column (str): The name of the column containing the labels.
Defaults to "label".
        figsize (tuple): The figure size for the plots. Defaults to (10, 6).
        palette (str): The color palette to use. Defaults to "viridis".
    """

    plt.figure(figsize=figsize)
    ax = sns.countplot(data=df, x=label_column, palette=palette)
    plt.title(f"Distribution of {label_column.capitalize()} - Count Plot")
    plt.xticks(rotation = 90)
    plt.xlabel(label_column.capitalize())
    plt.ylabel("Count")

    for p in ax.patches:
        ax.annotate(f'{int(p.get_height())}',
                    (p.get_x() + p.get_width() / 2., p.get_height()),
                    ha='center', va='center', fontsize=11, color='black',
xytext=(0, 5),
                    textcoords='offset points')
    plt.tight_layout()
    plt.show()


    label_counts = df[label_column].value_counts()
    plt.figure(figsize=figsize)
    plt.pie(label_counts, labels=label_counts.index, autopct='%1.1f%%',
startangle=140, colors=sns.color_palette(palette))
    plt.title(f"Distribution of {label_column.capitalize()} - Pie Chart")
    plt.tight_layout()
    plt.show()

visualize_label_distribution(df)
```






```
import cv2

num_images = 5

plt.figure(figsize=(20, 30))

for i, category in enumerate(categories):
    category_images = df[df['label'] ==
category]['image_path'].iloc[:num_images]
```

```
    for j, img_path in enumerate(category_images):

        img = cv2.imread(img_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        plt.subplot(len(categories), num_images, i * num_images + j + 1)
        plt.imshow(img)
        plt.axis('off')
        plt.title(category)

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


3D_Rendering
3D_Rendering
3D_Rendering
3D_Rendering
3D_Rendering
3D_Volume_Rendering
3D_Volume_Rendering
3D_Volume_Rendering
3D_Volume_Rendering
3D_Volume_Rendering
Adaptive_Histogram_Equalization
Adaptive_Histogram_Equalization
Adaptive_Histogram_Equalization
Adaptive_Histogram_Equalization
Adaptive_Histogram_Equalization
Alpha_Blending
Alpha_Blending
Alpha_Blending
Alpha_Blending
Alpha_Blending
Basic_Color_Map
Basic_Color_Map
Basic_Color_Map
Basic_Color_Map
Basic_Color_Map
Contrast_Stretching
Contrast_Stretching
Contrast_Stretching
Contrast_Stretching
Contrast_Stretching
Edge_Detection
Edge_Detection
Edge_Detection
Edge_Detection
Edge_Detection
Gamma_Correction
Gamma_Correction
Gamma_Correction
Gamma_Correction
Gamma_Correction
Gaussian_Blur
Gaussian_Blur
Gaussian_Blur
Gaussian_Blur
Gaussian_Blur
Heatmap_Visualization
Heatmap_Visualization
Heatmap_Visualization
Heatmap_Visualization
Heatmap_Visualization
Interactive_Segmentation
Interactive_Segmentation
Interactive_Segmentation
Interactive_Segmentation
Interactive_Segmentation
LUT_Color_Map
LUT_Color_Map
LUT_Color_Map
LUT_Color_Map
LUT_Color_Map
Random_Color_Palette
Random_Color_Palette
Random_Color_Palette
Random_Color_Palette
Random_Color_Palette

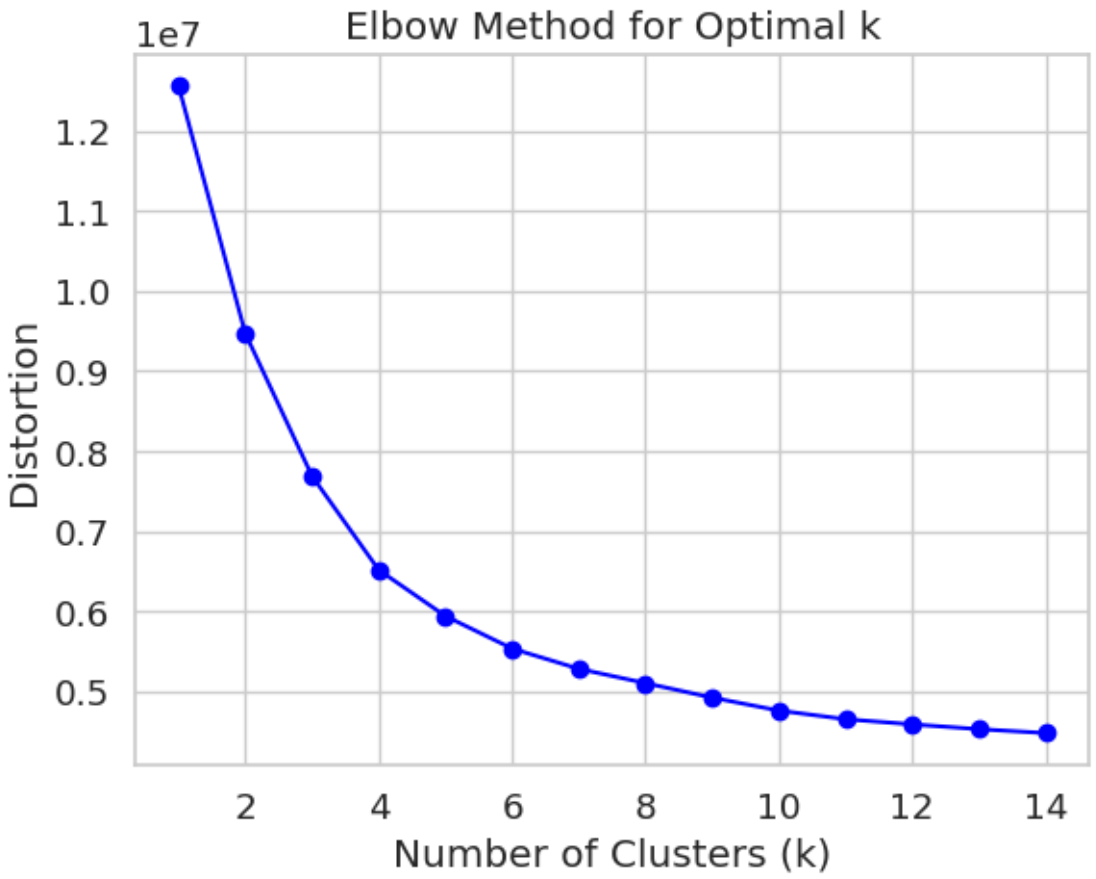
```
from PIL import Image

def preprocess_image(img_path, target_size=(64, 64)):
    img = Image.open(img_path).resize(target_size)
    img_array = np.array(img) / 255.0
    return img_array.flatten()
X = np.array([preprocess_image(img_path) for img_path in df['image_path']])
print(X.shape)
```

```
(12662, 12288)
```

```
from sklearn.decomposition import PCA

pca = PCA(n_components=100)
X_reduced = pca.fit_transform(X)
print(X_reduced.shape)
```

```
(12662, 100)
```

```
from sklearn.cluster import KMeans

distortions = []
K = range(1, 15)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_reduced)
    distortions.append(kmeans.inertia_)

plt.plot(K, distortions, 'bo-')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Distortion')
plt.title('Elbow Method for Optimal k')
plt.show()
```




```
k = 4
kmeans = KMeans(n_clusters=k, random_state=42)
df['cluster'] = kmeans.fit_predict(X_reduced)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

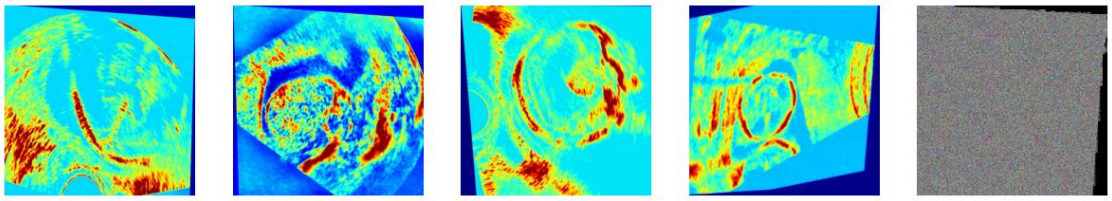
```
df
```

```
                                              image_path  \
0      /kaggle/input/medical-imaging-fetal-colorized-...
1      /kaggle/input/medical-imaging-fetal-colorized-...
2      /kaggle/input/medical-imaging-fetal-colorized-...
3      /kaggle/input/medical-imaging-fetal-colorized-...
4      /kaggle/input/medical-imaging-fetal-colorized-...
...                                                  ...
12657  /kaggle/input/medical-imaging-fetal-colorized-...
12658  /kaggle/input/medical-imaging-fetal-colorized-...
12659  /kaggle/input/medical-imaging-fetal-colorized-...
12660  /kaggle/input/medical-imaging-fetal-colorized-...
12661  /kaggle/input/medical-imaging-fetal-colorized-...
```

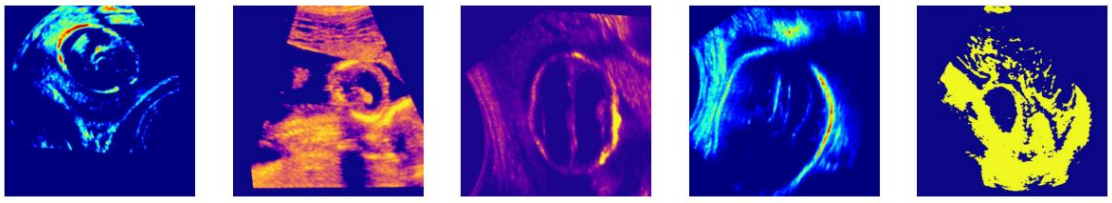
```
                     label  cluster
0              3D_Rendering        3
1              3D_Rendering        3
2              3D_Rendering        3
3              3D_Rendering        1
4              3D_Rendering        3
...                     ...      ...
12657  Random_Color_Palette        0
12658  Random_Color_Palette        0
12659  Random_Color_Palette        0
12660  Random_Color_Palette        0
12661  Random_Color_Palette        1

[12662 rows x 3 columns]
```

```python
for cluster in range(k):
    print(f"Cluster {cluster}:")
    cluster_samples = df[df['cluster'] == cluster]['image_path'].sample(5,
random_state=42)
    plt.figure(figsize=(15, 3))
    for i, img_path in enumerate(cluster_samples):
        plt.subplot(1, 5, i + 1)
        img = Image.open(img_path)
        plt.imshow(img)
        plt.axis('off')
    plt.show()
```

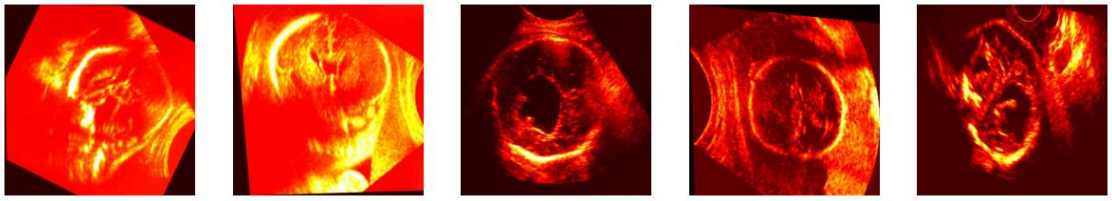
Cluster 0:



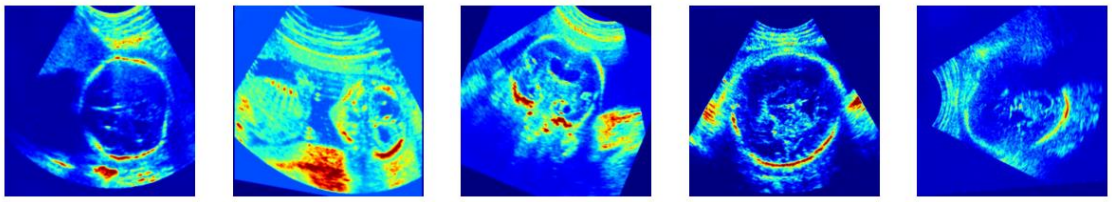
Cluster 1:



Cluster 2:



Cluster 3:



```
df['cluster'].unique()
```

```
array([3, 1, 0, 2], dtype=int32)
```

```
df['cluster'].value_counts()
```

```
cluster
3    4685
1    4379
0    2626
2     972
Name: count, dtype: int64
```

```
df = df[['image_path', 'cluster']]
```

```
from imblearn.over_sampling import RandomOverSampler

ros = RandomOverSampler(random_state=42)
X_resampled, y_resampled = ros.fit_resample(df[['image_path']],
df['cluster'])

df_resampled = pd.DataFrame(X_resampled, columns=['image_path'])
df_resampled['category_encoded'] = y_resampled

print("\nClass distribution after oversampling:")
print(df_resampled['category_encoded'].value_counts())
```

```
Class distribution after oversampling:
category_encoded
3    4685
1    4685
0    4685
2    4685
Name: count, dtype: int64
```

```
df_resampled['category_encoded'] =
df_resampled['category_encoded'].astype(str)

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
Activation, Dropout, BatchNormalization
from tensorflow.keras import regularizers

import warnings
warnings.filterwarnings("ignore")

print ('check')

check

train_df_new, temp_df_new = train_test_split(
    df_resampled,
    train_size=0.8,
    shuffle=True,
    random_state=42,
    stratify=df_resampled['category_encoded']
)

valid_df_new, test_df_new = train_test_split(
    temp_df_new,
    test_size=0.5,
    shuffle=True,
    random_state=42,
    stratify=temp_df_new['category_encoded']
)

batch_size = 16
img_size = (224, 224)
channels = 3
img_shape = (img_size[0], img_size[1], channels)

tr_gen = ImageDataGenerator(rescale=1./255)
ts_gen = ImageDataGenerator(rescale=1./255)

train_gen_new = tr_gen.flow_from_dataframe(
    train_df_new,
    x_col='image_path',
    y_col='category_encoded',
```

```
    target_size=img_size,
    class_mode='sparse',
    color_mode='rgb',
    shuffle=True,
    batch_size=batch_size
)

valid_gen_new = ts_gen.flow_from_dataframe(
    valid_df_new,
    x_col='image_path',
    y_col='category_encoded',
    target_size=img_size,
    class_mode='sparse',
    color_mode='rgb',
    shuffle=True,
    batch_size=batch_size
)

test_gen_new = ts_gen.flow_from_dataframe(
    test_df_new,
    x_col='image_path',
    y_col='category_encoded',
    target_size=img_size,
    class_mode='sparse',
    color_mode='rgb',
    shuffle=False,
    batch_size=batch_size
)
```

```
Found 14992 validated image filenames belonging to 4 classes.
Found 1874 validated image filenames belonging to 4 classes.
Found 1874 validated image filenames belonging to 4 classes.
```

```
import tensorflow as tf

print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
```

```
Num GPUs Available:  2
```

```
gpus = tf.config.list_physical_devices('GPU')
if gpus:
    try:
        for gpu in gpus:
            tf.config.experimental.set_memory_growth(gpu, True)
        print("GPU is set for TensorFlow")
    except RuntimeError as e:
        print(e)
```

```
GPU is set for TensorFlow
```

```
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
```

```
early_stopping = EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)

from tensorflow.keras import layers, models

from tensorflow.keras.applications import MobileNet
from tensorflow.keras.models import Model
from tensorflow.keras.layers import (
    GlobalAveragePooling2D, Dense, Dropout,
    BatchNormalization, GaussianNoise, Input,
    MultiHeadAttention, Reshape, LSTM
)
from tensorflow.keras.optimizers import Adam

def create_mobilenet_lstm_model(input_shape, num_classes=4,
learning_rate=0.0001):
    inputs = Input(shape=input_shape, name="Input_Layer")

    base_model = MobileNet(weights='imagenet', input_tensor=inputs,
include_top=False)
    base_model.trainable = False

    x = base_model.output

    height, width, channels = x.shape[1], x.shape[2], x.shape[3]
    x = Reshape((height * width, channels), name="Reshape_to_Sequence")(x)

    attention_output = MultiHeadAttention(
        num_heads=8, key_dim=channels, name="Multi_Head_Attention"
    )(x, x)

    attention_output = Reshape((height, width, channels),
name="Reshape_to_Spatial")(attention_output)

    x = GaussianNoise(0.25, name="Gaussian_Noise_1")(attention_output)
    x = GlobalAveragePooling2D(name="Global_Avg_Pooling")(x)

    x = Reshape((1, -1), name="Reshape_for_LSTM")(x)

    x = LSTM(128, return_sequences=False, name="LSTM_Layer")(x)

    outputs = Dense(num_classes, activation='softmax',
name="Output_Layer")(x)

    model = Model(inputs=inputs, outputs=outputs, name="MobileNet_with_LSTM")

    model.compile(
        optimizer=Adam(learning_rate=learning_rate),
        loss='sparse_categorical_crossentropy',
```

```
        metrics=['accuracy']
    )

    return model

input_shape = (224, 224, 3)
cnn_model = create_mobilenet_lstm_model(input_shape, num_classes=4,
learning_rate=0.0001)

cnn_model.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/mobilenet/mobilenet_1_0_224_tf_no_top.h5
17225924/17225924 ━━━━━━━━━━━━━━━━━━━━0s 0us/step

Model: "MobileNet_with_LSTM"
```

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]
conv1_bn (BatchNormalization)	(None, 112, 112, 32)	128	conv1[0][0]
conv1_relu (ReLU)	(None, 112, 112, 32)	0	conv1_bn[0][0]
conv_dw_1 (DepthwiseConv2D)	(None, 112, 112, 32)	288	conv1_relu[0][0]
conv_dw_1_bn (BatchNormalization)	(None, 112, 112, 32)	128	conv_dw_1[0][0]

conv_dw_1_relu (ReLU)	(None, 112, 112, 32)	0	conv_dw_1_bn[0][0]
conv_pw_1 (Conv2D)	(None, 112, 112, 64)	2,048	conv_dw_1_relu[0][0]
conv_pw_1_bn (BatchNormalization)	(None, 112, 112, 64)	256	conv_pw_1[0][0]
conv_pw_1_relu (ReLU)	(None, 112, 112, 64)	0	conv_pw_1_bn[0][0]
conv_pad_2 (ZeroPadding2D)	(None, 113, 113, 64)	0	conv_pw_1_relu[0][0]
conv_dw_2 (DepthwiseConv2D)	(None, 56, 56, 64)	576	conv_pad_2[0][0]
conv_dw_2_bn (BatchNormalization)	(None, 56, 56, 64)	256	conv_dw_2[0][0]
conv_dw_2_relu (ReLU)	(None, 56, 56, 64)	0	conv_dw_2_bn[0][0]
conv_pw_2 (Conv2D)	(None, 56, 56, 128)	8,192	conv_dw_2_relu[0][0]
conv_pw_2_bn (BatchNormalization)	(None, 56, 56, 128)	512	conv_pw_2[0][0]

conv_pw_2_relu (ReLU)	(None, 56, 56, 128)	0	conv_pw_2_bn[0][0]
conv_dw_3 (DepthwiseConv2D)	(None, 56, 56, 128)	1,152	conv_pw_2_relu[0][0]
conv_dw_3_bn (BatchNormalization)	(None, 56, 56, 128)	512	conv_dw_3[0][0]
conv_dw_3_relu (ReLU)	(None, 56, 56, 128)	0	conv_dw_3_bn[0][0]
conv_pw_3 (Conv2D)	(None, 56, 56, 128)	16,384	conv_dw_3_relu[0][0]
conv_pw_3_bn (BatchNormalization)	(None, 56, 56, 128)	512	conv_pw_3[0][0]
conv_pw_3_relu (ReLU)	(None, 56, 56, 128)	0	conv_pw_3_bn[0][0]
conv_pad_4 (ZeroPadding2D)	(None, 57, 57, 128)	0	conv_pw_3_relu[0][0]
conv_dw_4 (DepthwiseConv2D)	(None, 28, 28, 128)	1,152	conv_pad_4[0][0]
conv_dw_4_bn	(None, 28, 28, 128)	512	

```
conv_dw_4[0][0]        │
│ (BatchNormalization)      │                           │                 │
│
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_dw_4_relu (ReLU)     │ (None, 28, 28, 128)      │               0 │
conv_dw_4_bn[0][0]     │
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_pw_4 (Conv2D)        │ (None, 28, 28, 256)      │          32,768 │
conv_dw_4_relu[0][0]   │
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_pw_4_bn              │ (None, 28, 28, 256)      │           1,024 │
conv_pw_4[0][0]        │
│ (BatchNormalization)      │                           │                 │
│
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_pw_4_relu (ReLU)     │ (None, 28, 28, 256)      │               0 │
conv_pw_4_bn[0][0]     │
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_dw_5                 │ (None, 28, 28, 256)      │           2,304 │
conv_pw_4_relu[0][0]   │
│ (DepthwiseConv2D)         │                           │                 │
│
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_dw_5_bn              │ (None, 28, 28, 256)      │           1,024 │
conv_dw_5[0][0]        │
│ (BatchNormalization)      │                           │                 │
│
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_dw_5_relu (ReLU)     │ (None, 28, 28, 256)      │               0 │
conv_dw_5_bn[0][0]     │
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_pw_5 (Conv2D)        │ (None, 28, 28, 256)      │          65,536 │
conv_dw_5_relu[0][0]   │
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_pw_5_bn              │ (None, 28, 28, 256)      │           1,024 │
conv_pw_5[0][0]        │
│ (BatchNormalization)      │                           │                 │
│
├──────────────────────────┼──────────────────────────┼─────────────────┼──────
──────────────────┤
│ conv_pw_5_relu (ReLU)     │ (None, 28, 28, 256)      │               0 │
```

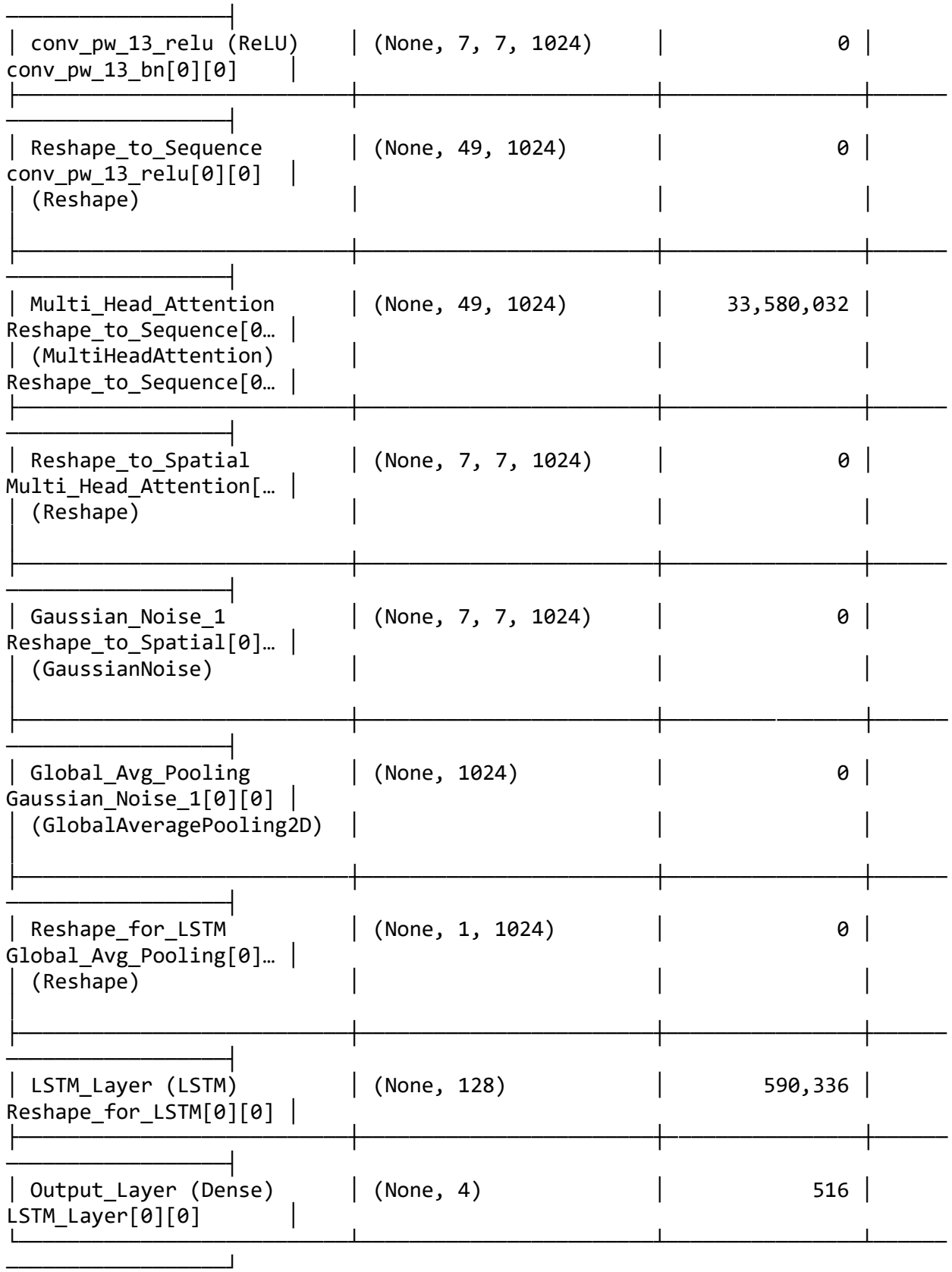
conv_pw_5_bn[0][0] |
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_pad_6 | (None, 29, 29, 256) | 0 |
conv_pw_5_relu[0][0] |
| (ZeroPadding2D) | | |
|
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_dw_6 | (None, 14, 14, 256) | 2,304 |
conv_pad_6[0][0] |
| (DepthwiseConv2D) | | |
|
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_dw_6_bn | (None, 14, 14, 256) | 1,024 |
conv_dw_6[0][0] |
| (BatchNormalization) | | |
|
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_dw_6_relu (ReLU) | (None, 14, 14, 256) | 0 |
conv_dw_6_bn[0][0] |
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_pw_6 (Conv2D) | (None, 14, 14, 512) | 131,072 |
conv_dw_6_relu[0][0] |
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_pw_6_bn | (None, 14, 14, 512) | 2,048 |
conv_pw_6[0][0] |
| (BatchNormalization) | | |
|
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_pw_6_relu (ReLU) | (None, 14, 14, 512) | 0 |
conv_pw_6_bn[0][0] |
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_dw_7 | (None, 14, 14, 512) | 4,608 |
conv_pw_6_relu[0][0] |
| (DepthwiseConv2D) | | |
|
├──────────────────────────┼──────────────────────────┼──────────────┼──────
──────────────────┤
| conv_dw_7_bn | (None, 14, 14, 512) | 2,048 |
conv_dw_7[0][0] |
| (BatchNormalization) | | |
|
├──────────────────────────┼──────────────────────────┼──────────────┼──────

```
──────────────────┤
│ conv_dw_7_relu (ReLU)     │ (None, 14, 14, 512)    │              0 │
conv_dw_7_bn[0][0]     │
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_pw_7 (Conv2D)        │ (None, 14, 14, 512)    │        262,144 │
conv_dw_7_relu[0][0]   │
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_pw_7_bn              │ (None, 14, 14, 512)    │          2,048 │
conv_pw_7[0][0]        │
│ (BatchNormalization)      │                        │                │
│
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_pw_7_relu (ReLU)     │ (None, 14, 14, 512)    │              0 │
conv_pw_7_bn[0][0]     │
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_dw_8                 │ (None, 14, 14, 512)    │          4,608 │
conv_pw_7_relu[0][0]   │
│ (DepthwiseConv2D)         │                        │                │
│
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_dw_8_bn              │ (None, 14, 14, 512)    │          2,048 │
conv_dw_8[0][0]        │
│ (BatchNormalization)      │                        │                │
│
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_dw_8_relu (ReLU)     │ (None, 14, 14, 512)    │              0 │
conv_dw_8_bn[0][0]     │
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_pw_8 (Conv2D)        │ (None, 14, 14, 512)    │        262,144 │
conv_dw_8_relu[0][0]   │
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_pw_8_bn              │ (None, 14, 14, 512)    │          2,048 │
conv_pw_8[0][0]        │
│ (BatchNormalization)      │                        │                │
│
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_pw_8_relu (ReLU)     │ (None, 14, 14, 512)    │              0 │
conv_pw_8_bn[0][0]     │
├──────────────────────────┼────────────────────────┼────────────────┼─────
──────────────────┤
│ conv_dw_9                 │ (None, 14, 14, 512)    │          4,608 │
```

```
conv_pw_8_relu[0][0]   │
│ (DepthwiseConv2D)         │                        │               │
│
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_dw_9_bn              │ (None, 14, 14, 512)    │         2,048 │
conv_dw_9[0][0]        │
│ (BatchNormalization)      │                        │               │
│
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_dw_9_relu (ReLU)     │ (None, 14, 14, 512)    │             0 │
conv_dw_9_bn[0][0]     │
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_pw_9 (Conv2D)        │ (None, 14, 14, 512)    │       262,144 │
conv_dw_9_relu[0][0]   │
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_pw_9_bn              │ (None, 14, 14, 512)    │         2,048 │
conv_pw_9[0][0]        │
│ (BatchNormalization)      │                        │               │
│
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_pw_9_relu (ReLU)     │ (None, 14, 14, 512)    │             0 │
conv_pw_9_bn[0][0]     │
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_dw_10                │ (None, 14, 14, 512)    │         4,608 │
conv_pw_9_relu[0][0]   │
│ (DepthwiseConv2D)         │                        │               │
│
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_dw_10_bn             │ (None, 14, 14, 512)    │         2,048 │
conv_dw_10[0][0]       │
│ (BatchNormalization)      │                        │               │
│
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_dw_10_relu (ReLU)    │ (None, 14, 14, 512)    │             0 │
conv_dw_10_bn[0][0]    │
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_pw_10 (Conv2D)       │ (None, 14, 14, 512)    │       262,144 │
conv_dw_10_relu[0][0]  │
├──────────────────────────┼────────────────────────┼───────────────┼──────
──────────┤
│ conv_pw_10_bn             │ (None, 14, 14, 512)    │         2,048 │
```

```
conv_pw_10[0][0]       │
│ (BatchNormalization)      │                           │                 │
│
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_pw_10_relu (ReLU)    │ (None, 14, 14, 512)      │               0 │
conv_pw_10_bn[0][0]    │
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_dw_11                │ (None, 14, 14, 512)      │           4,608 │
conv_pw_10_relu[0][0]  │
│ (DepthwiseConv2D)         │                          │                 │
│
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_dw_11_bn             │ (None, 14, 14, 512)      │           2,048 │
conv_dw_11[0][0]       │
│ (BatchNormalization)      │                          │                 │
│
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_dw_11_relu (ReLU)    │ (None, 14, 14, 512)      │               0 │
conv_dw_11_bn[0][0]    │
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_pw_11 (Conv2D)       │ (None, 14, 14, 512)      │         262,144 │
conv_dw_11_relu[0][0]  │
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_pw_11_bn             │ (None, 14, 14, 512)      │           2,048 │
conv_pw_11[0][0]       │
│ (BatchNormalization)      │                          │                 │
│
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_pw_11_relu (ReLU)    │ (None, 14, 14, 512)      │               0 │
conv_pw_11_bn[0][0]    │
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_pad_12               │ (None, 15, 15, 512)      │               0 │
conv_pw_11_relu[0][0]  │
│ (ZeroPadding2D)           │                          │                 │
│
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
──────────────────┤
│ conv_dw_12                │ (None, 7, 7, 512)        │           4,608 │
conv_pad_12[0][0]      │
│ (DepthwiseConv2D)         │                          │                 │
│
├──────────────────────────────┼──────────────────────────┼─────────────────┼─────
```

```
──────────────────┤
│ conv_dw_12_bn             │ (None, 7, 7, 512)      │           2,048 │
conv_dw_12[0][0]       │
│ (BatchNormalization)      │                        │                 │
│
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_dw_12_relu (ReLU)    │ (None, 7, 7, 512)      │               0 │
conv_dw_12_bn[0][0]    │
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_pw_12 (Conv2D)       │ (None, 7, 7, 1024)     │         524,288 │
conv_dw_12_relu[0][0]  │
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_pw_12_bn             │ (None, 7, 7, 1024)     │           4,096 │
conv_pw_12[0][0]       │
│ (BatchNormalization)      │                        │                 │
│
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_pw_12_relu (ReLU)    │ (None, 7, 7, 1024)     │               0 │
conv_pw_12_bn[0][0]    │
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_dw_13                │ (None, 7, 7, 1024)     │           9,216 │
conv_pw_12_relu[0][0]  │
│ (DepthwiseConv2D)         │                        │                 │
│
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_dw_13_bn             │ (None, 7, 7, 1024)     │           4,096 │
conv_dw_13[0][0]       │
│ (BatchNormalization)      │                        │                 │
│
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_dw_13_relu (ReLU)    │ (None, 7, 7, 1024)     │               0 │
conv_dw_13_bn[0][0]    │
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_pw_13 (Conv2D)       │ (None, 7, 7, 1024)     │       1,048,576 │
conv_dw_13_relu[0][0]  │
├──────────────────────────┼────────────────────────┼─────────────────┼────
──────────────────┤
│ conv_pw_13_bn             │ (None, 7, 7, 1024)     │           4,096 │
conv_pw_13[0][0]       │
│ (BatchNormalization)      │                        │                 │
│
├──────────────────────────┼────────────────────────┼─────────────────┼────
```

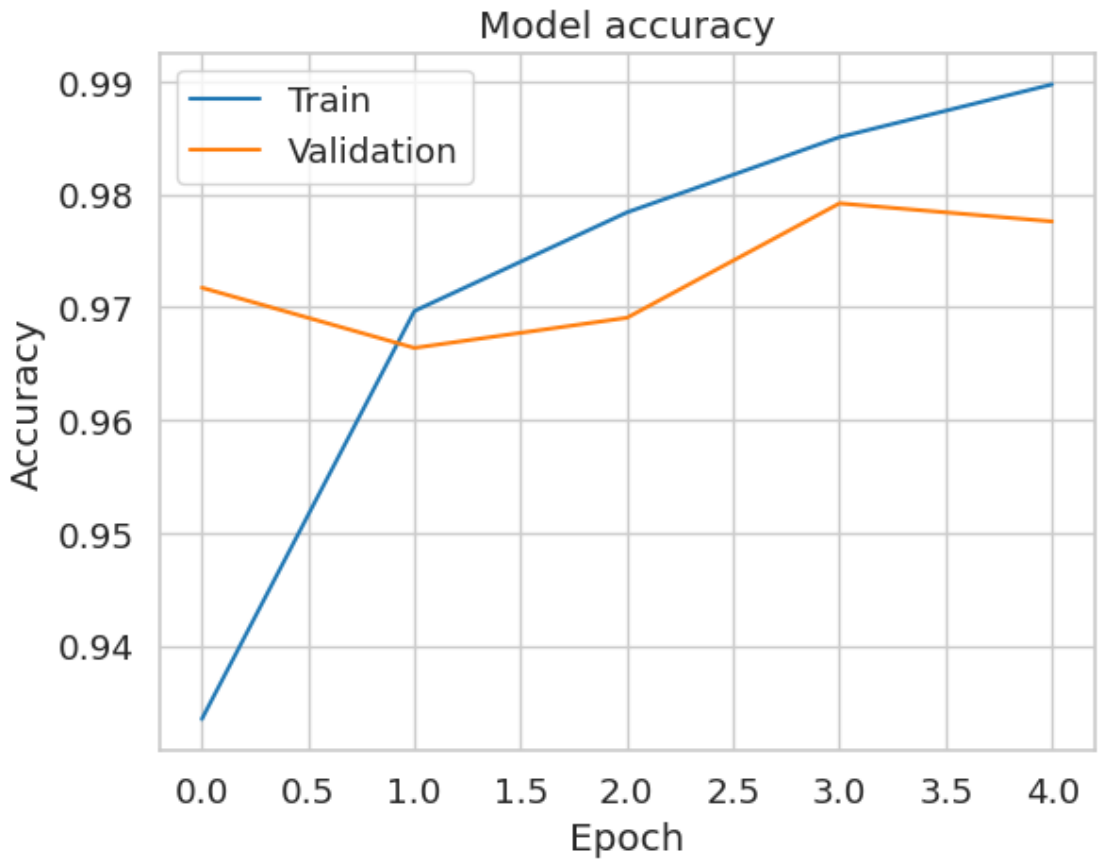


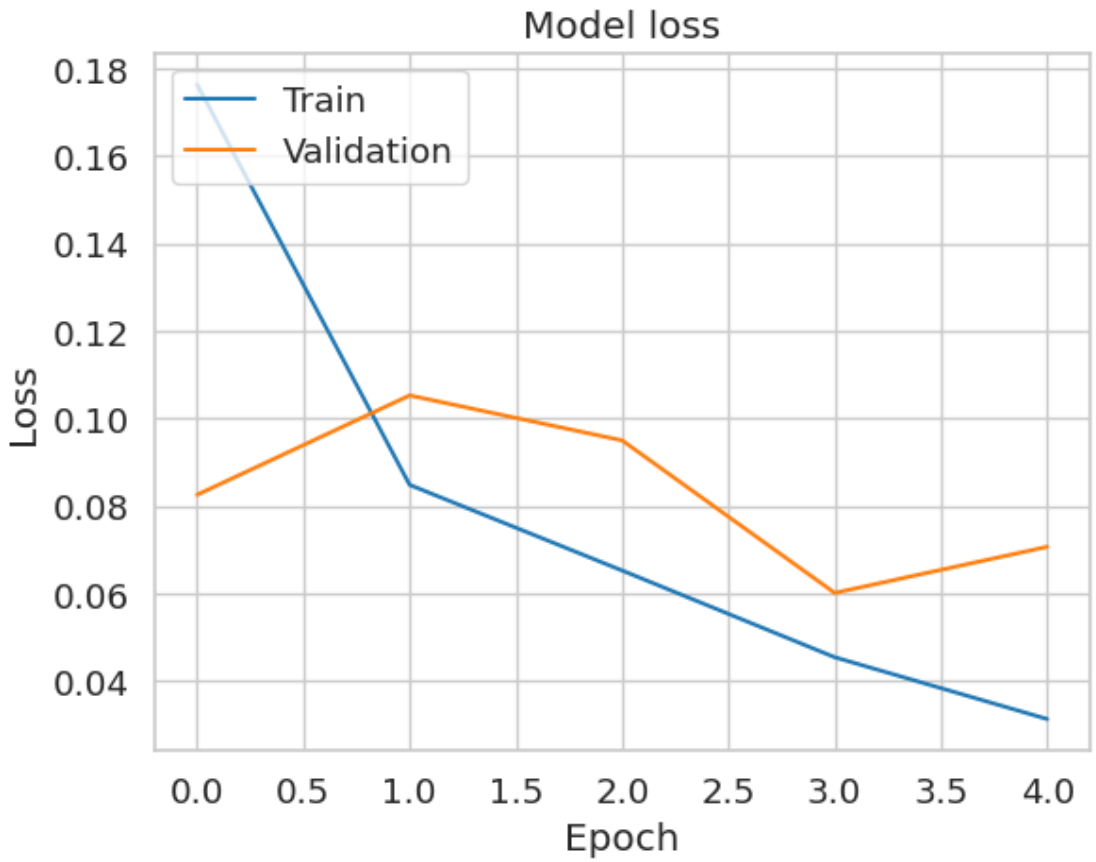
```
 Total params: 37,399,748 (142.67 MB)

 Trainable params: 34,170,884 (130.35 MB)

 Non-trainable params: 3,228,864 (12.32 MB)
```

```
history = cnn_model.fit(
    train_gen_new,
    validation_data=valid_gen_new,
    epochs=5,
    batch_size=16,
    verbose=1
)
```

```
Epoch 1/5
937/937 ━━━━━━━━━━━━━━━━━━━━101s 100ms/step - accuracy: 0.8971 - loss:
0.2696 - val_accuracy: 0.9717 - val_loss: 0.0825
Epoch 2/5
937/937 ━━━━━━━━━━━━━━━━━━━━99s 105ms/step - accuracy: 0.9670 - loss: 0.0909
- val_accuracy: 0.9664 - val_loss: 0.1052
Epoch 3/5
937/937 ━━━━━━━━━━━━━━━━━━━━99s 105ms/step - accuracy: 0.9795 - loss: 0.0607
- val_accuracy: 0.9691 - val_loss: 0.0950
Epoch 4/5
937/937 ━━━━━━━━━━━━━━━━━━━━98s 104ms/step - accuracy: 0.9863 - loss: 0.0431
- val_accuracy: 0.9792 - val_loss: 0.0601
Epoch 5/5
937/937 ━━━━━━━━━━━━━━━━━━━━98s 104ms/step - accuracy: 0.9897 - loss: 0.0322
- val_accuracy: 0.9776 - val_loss: 0.0707
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```


Model accuracy
Train
Validation
Accuracy
0.99
0.98
0.97
0.96
0.95
0.94
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epoch




```
test_labels = test_gen_new.classes
predictions = cnn_model.predict(test_gen_new)
predicted_classes = np.argmax(predictions, axis=1)
```

```
118/118 ━━━━━━━━━━━━━━━━━━━━9s 74ms/step
```

```
report = classification_report(test_labels, predicted_classes,
target_names=list(test_gen_new.class_indices.keys()))
print(report)
```

```
              precision    recall  f1-score   support

           0       0.96      0.99      0.98       469
           1       0.97      0.99      0.98       468
           2       1.00      1.00      1.00       469
           3       0.99      0.94      0.97       468

    accuracy                           0.98      1874
   macro avg       0.98      0.98      0.98      1874
weighted avg       0.98      0.98      0.98      1874
```

```
conf_matrix = confusion_matrix(test_labels, predicted_classes)
```

```
conf_matrix

array([[465,   2,   0,   2],
       [  5, 462,   0,   1],
       [  0,   0, 469,   0],
       [ 14,  14,   0, 440]])
```

```
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=list(test_gen_new.class_indices.keys()),
yticklabels=list(test_gen_new.class_indices.keys()))
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```