Mental Health Assessment Classification Using Neural Network

1)Data Collection and Organization.

-Gather your raw images and organize them into a structured folder hierarchy.

-Use Roboflow to upload your images and annotations, if available, and organize them into datasets.

2)Data Preprocessing:

-Roboflow provides tools for preprocessing images, such as resizing, cropping, and adjusting image quality. You can apply these preprocessing techniques to ensure uniformity and optimize the images for training.

3)Data Augmentation:

-Roboflow offers a wide range of data augmentation options to increase the variability of your dataset. These include transformations like flipping, rotation, scaling, translation, and introducing noise.

-Experiment with different augmentation techniques to enhance the robustness of your model and improve its performance on unseen data. Exporting the Dataset:

Once processed and augmented dataset to satisfaction, export it from Roboflow in the format suitable for YOLOv8 training.

Training YOLOv8: -Follow the steps outlined previously for configuring, training, and evaluating the YOLOv8 model using TensorFlow. Use the exported dataset from Roboflow as input to the training pipeline.

IMPORTNG LIBRARIES

In [20]:
```
import tensorflow as tf
from tensorflow.keras import models, layers
from tensorflow.keras.utils import image_dataset_from_directory
import matplotlib.pyplot as plt
import numpy
```

Converting Images To Tensors To Visualize and See Number of Classes

In [38]:
```
'''
Use the image_dataset_from_directory function to load the dataset.
The subset parameter is set to 'training' to get the training dataset.
The validation_split parameter is set to 0.2 to split 20% of the data for validation
'''

train_ds = image_dataset_from_directory(
     r"C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\archive\EmotionsDataset_Splitted\data\train",
     shuffle =True,
     image_size = (400,400),
     batch_size =20)

valid_ds = image_dataset_from_directory(
     r"C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\archive\EmotionsDataset_Splitted\data\train",
     shuffle =True,
     image_size = (400,400),
     batch_size = 20)

test_ds = image_dataset_from_directory(
     r"C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\archive\EmotionsDataset_Splitted\data\test",
     shuffle =True,
     image_size = (400,400),
     batch_size = 20)
```

```
Found 3152 files belonging to 4 classes.
Found 3152 files belonging to 4 classes.
Found 788 files belonging to 4 classes.
```

In [39]:
```
class_names = train_ds.class_names
class_names    # for printing class names
```

Out[39]:
```
['angry', 'happy', 'nothing', 'sad']
```

In [40]:
```
plt.figure(figsize=(15,15))
for image_batch,label_batch in train_ds.take(5):
    for i in range(9):
        plt.subplot(4,3,i+1)
        plt.imshow(image_batch[i].numpy().astype('uint8'))
        plt.title(class_names[label_batch[i]])
```

In [90]:
```
# Looking the number of data-pointpresent in each set (data_point* batch_size = number of images)
print("number of batches in Training dataset are : ",len(train_ds))
print("number of batches in Testing dataset are : ",len(test_ds))
print("number of batches in Validation dataset are : ",len(valid_ds))
```

```
number of batches in Training dataset are :  127
number of batches in Testing dataset are :  40
number of batches in Validation dataset are :  32
```

In [91]:
```
for image_batch,label_batch in train_ds.take(1):
    print(image_batch.shape)
    print(label_batch.numpy())  # here 3,9,3,7 ......  are class labels
```

```
(20, 224, 224, 3)
[2 3 3 3 3 1 2 3 0 2 2 2 1 3 2 1 2 2 0 1]
```

Pre-Processing

In [92]:
```
"""
cache(): This line stores the training dataset. Caching keeps the dataset in memory, which helps speed up subse

shuffle(1000): This line shuffles the training data. Shuffling randomly orders the dataset, which can assist en

Prefetch() improves model performance by lowering the time spent waiting for data to be loaded. The prefetch me

"""
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size = tf.data.AUTOTUNE)
val_ds = valid_ds.cache().shuffle(1000).prefetch(buffer_size = tf.data.AUTOTUNE)
test_ds = test_ds.cache().shuffle(1000).prefetch(buffer_size = tf.data.AUTOTUNE)
```

In [93]:
```
"""Adding Layers for Standardizing Image Pixel Values and Resizing the images to desired range"""
from tensorflow.keras.layers import Resizing, Rescaling ,RandomFlip, RandomRotation

resize_and_rescale = tf.keras.Sequential([
    Resizing(224, 224),
    Rescaling(1.0 / 255)
])
 # Standardizing size and scale of pixels
```

DATA AUGMENTATION

In [94]:
```
"""Adding layers for Generating different views of Object so that if anyone capture image at different angle or
from tensorflow.keras.layers import RandomFlip, RandomRotation
data_augmentation = tf.keras.Sequential([
    RandomFlip("horizontal"),
    RandomFlip("vertical"),
    RandomRotation(0.2)
]) # Generating different views of same leaf so that if anyone capture image at different angle or side it shou
```

MODELLING USING CONVOLUTIONAL NEURAL NETWORK

In [95]:
```
input_shape = (20,224,224,3) # (Batches , X, Y, Channel)  -->> Channel = 3 -> RGB
n_classes = 8
model = models.Sequential([
    resize_and_rescale,
    data_augmentation,
    layers.Conv2D(32,(3,3),activation="relu",input_shape = input_shape),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(32,kernel_size = (3,3),activation="relu"),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,kernel_size = (3,3),activation="relu"),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(64,kernel_size = (3,3),activation = "relu"),
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(32,kernel_size = (3,3),activation = "relu"),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(64,activation = "relu"),
    layers.Dense(n_classes,activation = "softmax")
])

model.build(input_shape = input_shape)
```

In [96]:
```
model.summary()  # summary of neural network architecture
```

Model: "sequential_22"

Layer (type)	Output Shape	Param #
sequential_20 (Sequential)	(20, 224, 224, 3)	0
sequential_21 (Sequential)	(20, 224, 224, 3)	0
conv2d_70 (Conv2D)	(20, 222, 222, 32)	896
max_pooling2d_70 (MaxPooling2D)	(20, 111, 111, 32)	0
conv2d_71 (Conv2D)	(20, 109, 109, 32)	9,248
max_pooling2d_71 (MaxPooling2D)	(20, 54, 54, 32)	0
conv2d_72 (Conv2D)	(20, 52, 52, 64)	18,496
max_pooling2d_72 (MaxPooling2D)	(20, 26, 26, 64)	0
conv2d_73 (Conv2D)	(20, 24, 24, 64)	36,928
max_pooling2d_73 (MaxPooling2D)	(20, 12, 12, 64)	0
conv2d_74 (Conv2D)	(20, 10, 10, 32)	18,464
max_pooling2d_74 (MaxPooling2D)	(20, 5, 5, 32)	0
flatten_14 (Flatten)	(20, 800)	0
dense_39 (Dense)	(20, 64)	51,264
dense_40 (Dense)	(20, 8)	520

Total params: 135,816 (530.53 KB)

Trainable params: 135,816 (530.53 KB)

Non-trainable params: 0 (0.00 B)

In [97]:
```
model.compile(
    optimizer = 'adam',
    loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits = False),
    metrics = ['accuracy']
)
## keras.losses.SparseCategoricalCrossentropy
```

In [98]:
```
history = model.fit(
    train_ds,
    epochs = 30,
    batch_size = 20,
    verbose = 1,
    validation_data = valid_ds
)
```

```
Epoch 1/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 38s 281ms/step - accuracy: 0.4409 - loss: 1.2440 - val_accuracy: 0.5317 - val_loss:
0.9485
Epoch 2/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 33s 259ms/step - accuracy: 0.5384 - loss: 0.9049 - val_accuracy: 0.5524 - val_loss:
0.8938
Epoch 3/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 277ms/step - accuracy: 0.5544 - loss: 0.8614 - val_accuracy: 0.5365 - val_loss:
0.8717
Epoch 4/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 36s 281ms/step - accuracy: 0.5360 - loss: 0.8629 - val_accuracy: 0.5413 - val_loss:
0.8542
Epoch 5/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 321ms/step - accuracy: 0.5301 - loss: 0.8816 - val_accuracy: 0.5778 - val_loss:
0.8723
Epoch 6/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 39s 307ms/step - accuracy: 0.5862 - loss: 0.8136 - val_accuracy: 0.5540 - val_loss:
0.9082
Epoch 7/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 31s 245ms/step - accuracy: 0.5723 - loss: 0.8262 - val_accuracy: 0.5778 - val_loss:
0.8930
Epoch 8/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 41s 320ms/step - accuracy: 0.5919 - loss: 0.8100 - val_accuracy: 0.6175 - val_loss:
0.7953
Epoch 9/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 33s 264ms/step - accuracy: 0.5692 - loss: 0.8892 - val_accuracy: 0.6000 - val_loss:
0.8165
Epoch 10/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 38s 299ms/step - accuracy: 0.6218 - loss: 0.8206 - val_accuracy: 0.5714 - val_loss:
0.8426
Epoch 11/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 38s 298ms/step - accuracy: 0.6202 - loss: 0.7726 - val_accuracy: 0.6206 - val_loss:
0.7940
Epoch 12/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 36s 286ms/step - accuracy: 0.6196 - loss: 0.7820 - val_accuracy: 0.6333 - val_loss:
0.8014
Epoch 13/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 36s 284ms/step - accuracy: 0.6236 - loss: 0.7764 - val_accuracy: 0.6238 - val_loss:
0.7693
Epoch 14/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 270ms/step - accuracy: 0.6247 - loss: 0.7809 - val_accuracy: 0.6492 - val_loss:
0.7663
Epoch 15/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 265ms/step - accuracy: 0.6275 - loss: 0.7531 - val_accuracy: 0.6460 - val_loss:
0.7647
Epoch 16/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 33s 257ms/step - accuracy: 0.6595 - loss: 0.7383 - val_accuracy: 0.6365 - val_loss:
0.7972
Epoch 17/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 271ms/step - accuracy: 0.6584 - loss: 0.7302 - val_accuracy: 0.6000 - val_loss:
0.7945
Epoch 18/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 39s 302ms/step - accuracy: 0.6469 - loss: 0.7410 - val_accuracy: 0.6397 - val_loss:
0.7560
Epoch 19/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 37s 288ms/step - accuracy: 0.6334 - loss: 0.7395 - val_accuracy: 0.6143 - val_loss:
0.7835
Epoch 20/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 36s 283ms/step - accuracy: 0.6627 - loss: 0.7145 - val_accuracy: 0.6651 - val_loss:
0.7298
Epoch 21/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 43s 338ms/step - accuracy: 0.6656 - loss: 0.7114 - val_accuracy: 0.6444 - val_loss:
0.7494
Epoch 22/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 56s 442ms/step - accuracy: 0.6532 - loss: 0.7081 - val_accuracy: 0.6683 - val_loss:
0.7271
Epoch 23/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 424ms/step - accuracy: 0.6597 - loss: 0.7061 - val_accuracy: 0.6698 - val_loss:
0.7049
Epoch 24/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 428ms/step - accuracy: 0.6749 - loss: 0.7001 - val_accuracy: 0.6540 - val_loss:
0.7330
Epoch 25/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 423ms/step - accuracy: 0.6882 - loss: 0.6806 - val_accuracy: 0.6222 - val_loss:
0.8063
```

```
Epoch 26/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 55s 434ms/step - accuracy: 0.6559 - loss: 0.7321 - val_accuracy: 0.6095 - val_loss:
0.8604
Epoch 27/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 428ms/step - accuracy: 0.6614 - loss: 0.7203 - val_accuracy: 0.6381 - val_loss:
0.7679
Epoch 28/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 55s 432ms/step - accuracy: 0.6792 - loss: 0.6879 - val_accuracy: 0.6730 - val_loss:
0.7001
Epoch 29/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.6854 - loss: 0.6476 - val_accuracy: 0.6460 - val_loss:
0.7523
Epoch 30/30
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.6722 - loss: 0.6754 - val_accuracy: 0.7048 - val_loss:
0.7183
```

In [101…
```
model.fit(train_ds,batch_size = 20 , verbose=1 , validation_data = valid_ds , initial_epoch = 30, epochs = 50)
```

```
Epoch 31/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 55s 433ms/step - accuracy: 0.7051 - loss: 0.6543 - val_accuracy: 0.6968 - val_loss:
0.7202
Epoch 32/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.7005 - loss: 0.6464 - val_accuracy: 0.6794 - val_loss:
0.7290
Epoch 33/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 427ms/step - accuracy: 0.6985 - loss: 0.6530 - val_accuracy: 0.6921 - val_loss:
0.7267
Epoch 34/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.7058 - loss: 0.6331 - val_accuracy: 0.6730 - val_loss:
0.7155
Epoch 35/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 429ms/step - accuracy: 0.7137 - loss: 0.6337 - val_accuracy: 0.6524 - val_loss:
0.7951
Epoch 36/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 426ms/step - accuracy: 0.7089 - loss: 0.6494 - val_accuracy: 0.6730 - val_loss:
0.7476
Epoch 37/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.7338 - loss: 0.6055 - val_accuracy: 0.6746 - val_loss:
0.6964
Epoch 38/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 423ms/step - accuracy: 0.7051 - loss: 0.6266 - val_accuracy: 0.6937 - val_loss:
0.7021
Epoch 39/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 424ms/step - accuracy: 0.7059 - loss: 0.6476 - val_accuracy: 0.6921 - val_loss:
0.7032
Epoch 40/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.7229 - loss: 0.6299 - val_accuracy: 0.6841 - val_loss:
0.7130
Epoch 41/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.7372 - loss: 0.6127 - val_accuracy: 0.6841 - val_loss:
0.7902
Epoch 42/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 426ms/step - accuracy: 0.7227 - loss: 0.6117 - val_accuracy: 0.6587 - val_loss:
0.8258
Epoch 43/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 53s 421ms/step - accuracy: 0.7178 - loss: 0.6281 - val_accuracy: 0.7032 - val_loss:
0.7178
Epoch 44/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 427ms/step - accuracy: 0.7422 - loss: 0.5908 - val_accuracy: 0.6873 - val_loss:
0.7098
Epoch 45/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 55s 429ms/step - accuracy: 0.7430 - loss: 0.5889 - val_accuracy: 0.6635 - val_loss:
0.9924
Epoch 46/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 425ms/step - accuracy: 0.7180 - loss: 0.6206 - val_accuracy: 0.6921 - val_loss:
0.6904
Epoch 47/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 429ms/step - accuracy: 0.7309 - loss: 0.5927 - val_accuracy: 0.6857 - val_loss:
0.7191
Epoch 48/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 429ms/step - accuracy: 0.7316 - loss: 0.6191 - val_accuracy: 0.6619 - val_loss:
0.7645
Epoch 49/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 427ms/step - accuracy: 0.7144 - loss: 0.5992 - val_accuracy: 0.6921 - val_loss:
0.6806
Epoch 50/50
127/127 ━━━━━━━━━━━━━━━━━━━━ 54s 427ms/step - accuracy: 0.7397 - loss: 0.5964 - val_accuracy: 0.6857 - val_loss:
0.7039
```

Out[101… `<keras.src.callbacks.history.History at 0x28e44a7c220>`

In [102…
```
model.fit(train_ds,batch_size = 20 , verbose=1 , validation_data = valid_ds , initial_epoch = 50, epochs = 100)
```

```
Epoch 51/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 39s 307ms/step - accuracy: 0.7445 - loss: 0.5756 - val_accuracy: 0.6937 - val_loss:
```

```
0.7131
Epoch 52/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 277ms/step - accuracy: 0.7029 - loss: 0.6163 - val_accuracy: 0.6905 - val_loss:
0.6989
Epoch 53/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 280ms/step - accuracy: 0.7255 - loss: 0.5865 - val_accuracy: 0.7032 - val_loss:
0.7208
Epoch 54/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 277ms/step - accuracy: 0.7379 - loss: 0.5767 - val_accuracy: 0.7063 - val_loss:
0.7560
Epoch 55/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7272 - loss: 0.5894 - val_accuracy: 0.7111 - val_loss:
0.6718
Epoch 56/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 278ms/step - accuracy: 0.7502 - loss: 0.5440 - val_accuracy: 0.7016 - val_loss:
0.7384
Epoch 57/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 277ms/step - accuracy: 0.7619 - loss: 0.5425 - val_accuracy: 0.7143 - val_loss:
0.7112
Epoch 58/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 36s 280ms/step - accuracy: 0.7533 - loss: 0.5775 - val_accuracy: 0.7143 - val_loss:
0.6698
Epoch 59/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7570 - loss: 0.5433 - val_accuracy: 0.6921 - val_loss:
0.7189
Epoch 60/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 278ms/step - accuracy: 0.7535 - loss: 0.5699 - val_accuracy: 0.7063 - val_loss:
0.6993
Epoch 61/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 277ms/step - accuracy: 0.7584 - loss: 0.5437 - val_accuracy: 0.7000 - val_loss:
0.7294
Epoch 62/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 278ms/step - accuracy: 0.7680 - loss: 0.5269 - val_accuracy: 0.7032 - val_loss:
0.6600
Epoch 63/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7578 - loss: 0.5453 - val_accuracy: 0.7063 - val_loss:
0.7207
Epoch 64/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 36s 282ms/step - accuracy: 0.7643 - loss: 0.5240 - val_accuracy: 0.6921 - val_loss:
0.7861
Epoch 65/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 280ms/step - accuracy: 0.7546 - loss: 0.5316 - val_accuracy: 0.7048 - val_loss:
0.7326
Epoch 66/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 277ms/step - accuracy: 0.7439 - loss: 0.5831 - val_accuracy: 0.7143 - val_loss:
0.6593
Epoch 67/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 278ms/step - accuracy: 0.7643 - loss: 0.5314 - val_accuracy: 0.6905 - val_loss:
0.7158
Epoch 68/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7510 - loss: 0.5503 - val_accuracy: 0.7254 - val_loss:
0.6801
Epoch 69/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 36s 280ms/step - accuracy: 0.7587 - loss: 0.5496 - val_accuracy: 0.7190 - val_loss:
0.6920
Epoch 70/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7649 - loss: 0.5224 - val_accuracy: 0.7048 - val_loss:
0.7113
Epoch 71/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7656 - loss: 0.5260 - val_accuracy: 0.6952 - val_loss:
0.7178
Epoch 72/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7695 - loss: 0.5345 - val_accuracy: 0.7175 - val_loss:
0.6751
Epoch 73/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 278ms/step - accuracy: 0.7690 - loss: 0.5425 - val_accuracy: 0.7111 - val_loss:
0.7160
Epoch 74/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 278ms/step - accuracy: 0.7840 - loss: 0.5014 - val_accuracy: 0.7095 - val_loss:
0.6902
Epoch 75/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7740 - loss: 0.4926 - val_accuracy: 0.7095 - val_loss:
0.6853
Epoch 76/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7833 - loss: 0.4942 - val_accuracy: 0.7143 - val_loss:
0.6624
Epoch 77/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7572 - loss: 0.5419 - val_accuracy: 0.6619 - val_loss:
1.0704
Epoch 78/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7794 - loss: 0.5387 - val_accuracy: 0.7222 - val_loss:
0.6473
Epoch 79/100
```

```
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 275ms/step - accuracy: 0.7740 - loss: 0.5220 - val_accuracy: 0.7127 - val_loss:
0.7483
Epoch 80/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 274ms/step - accuracy: 0.7869 - loss: 0.5199 - val_accuracy: 0.7190 - val_loss:
0.6848
Epoch 81/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 275ms/step - accuracy: 0.7777 - loss: 0.5126 - val_accuracy: 0.7143 - val_loss:
0.6622
Epoch 82/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 271ms/step - accuracy: 0.7610 - loss: 0.5334 - val_accuracy: 0.7270 - val_loss:
0.7058
Epoch 83/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 271ms/step - accuracy: 0.7708 - loss: 0.5080 - val_accuracy: 0.7317 - val_loss:
0.6482
Epoch 84/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 271ms/step - accuracy: 0.7687 - loss: 0.5087 - val_accuracy: 0.7397 - val_loss:
0.6399
Epoch 85/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 272ms/step - accuracy: 0.7906 - loss: 0.4961 - val_accuracy: 0.7349 - val_loss:
0.6804
Epoch 86/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 274ms/step - accuracy: 0.7870 - loss: 0.4913 - val_accuracy: 0.7143 - val_loss:
0.7265
Epoch 87/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 272ms/step - accuracy: 0.7683 - loss: 0.5221 - val_accuracy: 0.7317 - val_loss:
0.6699
Epoch 88/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 275ms/step - accuracy: 0.8002 - loss: 0.4692 - val_accuracy: 0.7317 - val_loss:
0.6783
Epoch 89/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 273ms/step - accuracy: 0.7751 - loss: 0.4929 - val_accuracy: 0.7381 - val_loss:
0.6635
Epoch 90/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 272ms/step - accuracy: 0.7988 - loss: 0.4727 - val_accuracy: 0.7270 - val_loss:
0.6202
Epoch 91/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 271ms/step - accuracy: 0.7903 - loss: 0.4893 - val_accuracy: 0.7317 - val_loss:
0.6842
Epoch 92/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 273ms/step - accuracy: 0.7739 - loss: 0.4963 - val_accuracy: 0.7317 - val_loss:
0.7301
Epoch 93/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 270ms/step - accuracy: 0.7924 - loss: 0.4791 - val_accuracy: 0.7159 - val_loss:
0.6918
Epoch 94/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 270ms/step - accuracy: 0.7990 - loss: 0.4637 - val_accuracy: 0.7317 - val_loss:
0.6753
Epoch 95/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 272ms/step - accuracy: 0.8053 - loss: 0.4549 - val_accuracy: 0.7222 - val_loss:
0.7534
Epoch 96/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 272ms/step - accuracy: 0.7951 - loss: 0.4807 - val_accuracy: 0.7349 - val_loss:
0.6879
Epoch 97/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 277ms/step - accuracy: 0.7866 - loss: 0.4611 - val_accuracy: 0.7079 - val_loss:
0.7571
Epoch 98/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 35s 279ms/step - accuracy: 0.7911 - loss: 0.4889 - val_accuracy: 0.7238 - val_loss:
0.8377
Epoch 99/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 271ms/step - accuracy: 0.7614 - loss: 0.5347 - val_accuracy: 0.7000 - val_loss:
0.7529
Epoch 100/100
127/127 ━━━━━━━━━━━━━━━━━━━━ 34s 270ms/step - accuracy: 0.7853 - loss: 0.4870 - val_accuracy: 0.7270 - val_loss:
0.7285
```

Out[102… <keras.src.callbacks.history.History at 0x28e44a7c670>

In [104…
```
scores = model.evaluate(test_ds)
scores
```

```
40/40 ━━━━━━━━━━━━━━━━━━━━ 6s 141ms/step - accuracy: 0.7362 - loss: 0.7030
```

Out[104… [0.6980661153793335, 0.7411167621612549]

In [105…
```
history
history.params
history.history.keys()
```

Out[105… dict_keys(['accuracy', 'loss', 'val_accuracy', 'val_loss'])

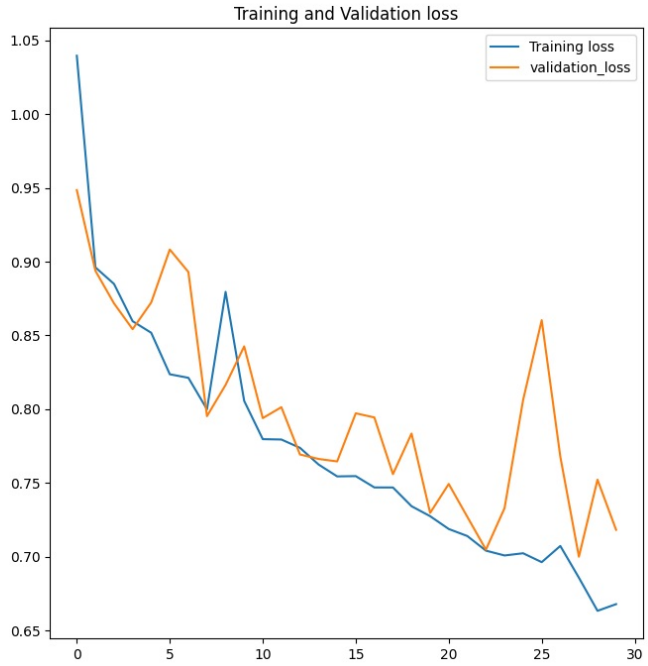
In [106…
```
acc = history.history["accuracy"]
val_acc= history.history["val_accuracy"]
```

```
loss = history.history["loss"]
val_loss = history.history["val_loss"]
```

In [107…
```
plt.figure(figsize=(17,8))
plt.subplot(1,2,1)
plt.plot(range(30),acc,label="Training Accuracy")
plt.plot(range(30),val_acc,label = "validation_accuracy")
plt.legend(loc="lower right")
plt.title("Training and Validation Accuracy")

plt.subplot(1,2,2)
plt.plot(range(30),loss,label="Training loss")
plt.plot(range(30),val_loss,label = "validation_loss")
plt.legend(loc="upper right")
plt.title("Training and Validation loss")
```

Out[107… `Text(0.5, 1.0, 'Training and Validation loss')`

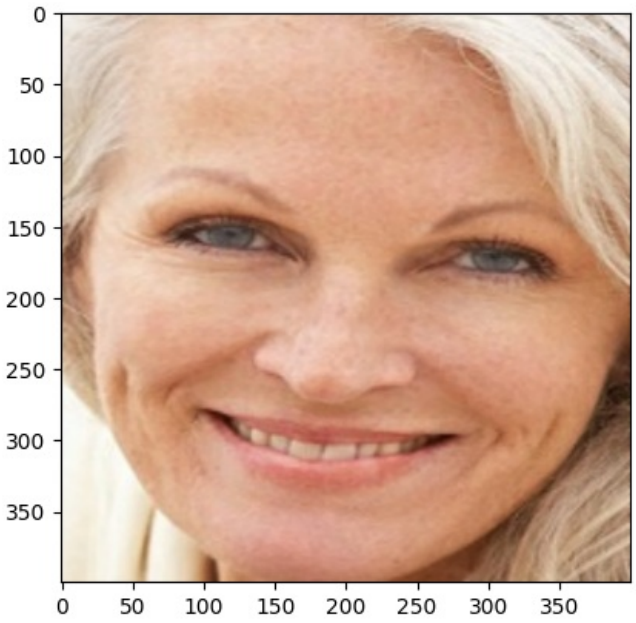


In [109…
```
for image_batch,label_batch in test_ds.take(1):
    first_image = image_batch[0].numpy().astype("uint8")
    first_label = label_batch[0].numpy()

    print('first image to predict')
    plt.imshow(first_image)
    print('actual label : ',class_names[first_label])

    batch_prediction = model.predict(image_batch)
    print(class_names[numpy.argmax(batch_prediction[0])])
```

```
first image to predict
actual label :  happy
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 218ms/step
happy
```



In [110…

```
## Defining functions for predicting test dataset results

def predict(model,img):
    img_arry = tf.keras.preprocessing.image.img_to_array((img))
    img_array = tf.expand_dims(img_arry,0)  ## creating a batch
    # prediction
    prediction = model.predict(img_array)

    pred_class = class_names[numpy.argmax(prediction[0])]
    confidence = round(100*(numpy.max(prediction[0])),2)
    return pred_class , confidence
```

In [111…

```
plt.figure(figsize = (15,15))
for images,label in test_ds.take(1):
    for i in range(12):
        ax = plt.subplot(3,4,i+1)
        plt.imshow(images[i].numpy().astype('uint8'))

        pred_class , confidence = predict(model,images[i])
        actual_class = class_names[label[i]]

        plt.title(f"Actual : {actual_class}, \n Predicted : {pred_class}. \n Confidence : {confidence} %")
        plt.axis('off')
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 393ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
```

Actual : angry,
Predicted : sad.
Confidence : 85.18 %

Actual : nothing,
Predicted : nothing.
Confidence : 100.0 %

Actual : happy,
Predicted : happy.
Confidence : 98.69 %

Actual : angry,
Predicted : sad.
Confidence : 65.91 %

Actual : angry,
Predicted : sad.
Confidence : 38.49 %

Actual : happy,
Predicted : happy.
Confidence : 98.18 %

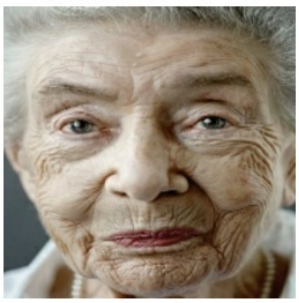
Actual : angry,
Predicted : angry.
Confidence : 99.75 %

Actual : happy,
Predicted : happy.
Confidence : 45.25 %

Actual : angry,
Predicted : happy.
Confidence : 57.81 %

Actual : nothing,
Predicted : nothing.
Confidence : 100.0 %

Actual : angry,
Predicted : happy.
Confidence : 63.34 %



Actual : angry,
Predicted : angry.
Confidence : 99.3 %

In [113…
```
# import os
# model_version = max([int(i) for i in os.listdir(r"") + [0]])+1
# model.save(f'{model_version}')
```

In [ ]:
```
```

```
        plt.axis('off')
```

Streamlining Data Preparation with Roboflow

This project leverages Roboflow's robust capabilities for data preprocessing and data augmentation, ensuring a well-conditioned dataset for optimal model performance.

- **Preprocessing:**
  - Standardized image dimensions for efficient model training.
  - Adjustments like grayscale conversion or contrast normalization for consistency.
  - Potential for object isolation or background removal for focused learning.
- **Data Augmentation:**
  - Artificial variations of existing images to increase dataset size and diversity.
  - Techniques like random cropping, flipping, rotation, and noise injection to simulate real-world variations.
  - Enhanced model generalizability and robustness to unseen data.

By employing Roboflow's user-friendly interface, we've streamlined the data preparation process, saving valuable time and resources. This well-prepared dataset lays the foundation for a superior machine learning model.

In [5]:
```
from roboflow import Roboflow
rf = Roboflow(api_key="[redacted]")
project = rf.workspace("emmo").project("human_emmotion")
version = project.version(1)
dataset = version.download("clip")
```

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in human_emmotion-1 to clip:: 100%|[redacted]| 438073/438073 [11:44<00:00, 621.83it/s]
Extracting Dataset Version Zip to human_emmotion-1 in clip:: 100%|[redacted]| 9038/9038 [00:20<00:00, 447.92it/s]
```

Modelling

Harnessing YOLOv8 for Emotion Classification

This project delves into the application of YOLOv8 for emotion classification.

- **YOLOv8:**
  - Powerful object detection framework with facial recognition capabilities.
  - Detects and isolates faces within images, enabling emotion analysis.
  - Can be customized to classify specific emotions based on facial feature extraction and classification layers.

By leveraging YOLOv8's object detection prowess for faces and incorporating emotion classification layers, this project aims to achieve accurate identification of emotions from images.

In [1]:
```
from ultralytics import YOLO
model = YOLO("yolov8n-cls.pt")
```

```
Downloading https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n-cls.pt to 'yolov8n-cls.pt'...
100%|██████████| 5.30M/5.30M [00:09<00:00, 559kB/s]
```

In [6]:
```
model.train(data = r"human_emmotion-1" , epochs =50)
```

```
New https://pypi.org/project/ultralytics/8.1.34 available 😃 Update with 'pip install -U ultralytics'
Ultralytics YOLOv8.1.29  Python-3.10.11 torch-2.2.1+cpu CPU (11th Gen Intel Core(TM) i5-11300H 3.10GHz)
engine\trainer: task=classify, mode=train, model=yolov8n-cls.pt, data=human_emmotion-1, epochs=50, time=None, patience=100, batch=16, imgsz=224, save=True, save_period=-1, cache=False, device=None, workers=0, project=None, name=train2, exist_ok=False, pretrained=True, optimizer=auto, verbose=True, seed=0, deterministic=True, single_cls=False, rect=False, cos_lr=False, close_mosaic=10, resume=False, amp=True, fraction=1.0, profile=False, freeze=None, multi_scale=False, overlap_mask=True, mask_ratio=4, dropout=0.0, val=True, split=val, save_json=False, save_hybrid=False, conf=None, iou=0.7, max_det=300, half=False, dnn=False, plots=True, source=None, vid_stride=1, stream_buffer=False, visualize=False, augment=False, agnostic_nms=False, classes=None, retina_masks=False, embed=None, show=False, save_frames=False, save_txt=False, save_conf=False, save_crop=False, show_labels=True, show_conf=True, show_boxes=True, line_width=None, format=torchscript, keras=False, optimize=False, int8=False, dynamic=False, simplify=False, opset=None, workspace=4, nms=False, lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.00005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.0, box=7.5, cls=0.5, dfl=1.5, pose=12.0, kobj=1.0, label_smoothing=0.0, nbs=64, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0, copy_paste=0.0, auto_augment=randaugment, erasing=0.4, crop_fraction=1.0, cfg=None, tracker=botsort.yaml, save_dir=runs\classify\train2
train: C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\train... found 8325 images in 4 classes ✅
val: None...
test: C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\test... found 348 images in 4 classes ✅

                   from  n    params  module                                       arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
  2                  -1  1      7360  ultralytics.nn.modules.block.C2f             [32, 32, 1, True]
  3                  -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
  4                  -1  2     49664  ultralytics.nn.modules.block.C2f             [64, 64, 2, True]
  5                  -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
  6                  -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  8                  -1  1    460288  ultralytics.nn.modules.block.C2f             [256, 256, 1, True]
  9                  -1  1    335364  ultralytics.nn.modules.head.Classify         [256, 4]
YOLOv8n-cls summary: 99 layers, 1443412 parameters, 1443412 gradients, 3.4 GFLOPs
Transferred 158/158 items from pretrained weights
TensorBoard: Start with 'tensorboard --logdir runs\classify\train2', view at http://localhost:6006/
train: Scanning C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\train... 8325 images, 0 corrupt
train: New cache created: C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\train.cache
val: Scanning C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\test... 348 images, 0 corrupt: 10
val: New cache created: C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\test.cache
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.000714, momentum=0.9) with parameter groups 26 weight(decay=0.0), 27 weight(decay=0.0005), 27 bias(decay=0.0)
TensorBoard: model graph visualization added ✅
Image sizes 224 train, 224 val
Using 0 dataloader workers
Logging results to runs\classify\train2
Starting training for 50 epochs...

      Epoch    GPU_mem       loss  Instances       Size
       1/50         0G     0.5196          5        224: 100%|██████████| 521/521 [13:37<00:00,  1.57s/it]
              classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:17<00:00,  1.59s/it]
                  all      0.908          1

      Epoch    GPU_mem       loss  Instances       Size
```

```
      2/50         0G     0.4337          5        224: 100%|██████████| 521/521 [11:18<00:00,  1.30s/it]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:09<00:00,  1.21it/s]
                     all      0.905          1

      Epoch    GPU_mem       loss  Instances       Size
      3/50         0G     0.4095          5        224: 100%|██████████| 521/521 [07:24<00:00,  1.17it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.25it/s]
                     all      0.891          1

      Epoch    GPU_mem       loss  Instances       Size
      4/50         0G      0.401          5        224: 100%|██████████| 521/521 [06:54<00:00,  1.26it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.34it/s]
                     all      0.874          1

      Epoch    GPU_mem       loss  Instances       Size
      5/50         0G     0.3679          5        224: 100%|██████████| 521/521 [07:00<00:00,  1.24it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.34it/s]
                     all      0.897          1

      Epoch    GPU_mem       loss  Instances       Size
      6/50         0G      0.343          5        224: 100%|██████████| 521/521 [06:58<00:00,  1.25it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.33it/s]
                     all      0.853          1

      Epoch    GPU_mem       loss  Instances       Size
      7/50         0G      0.338          5        224: 100%|██████████| 521/521 [06:57<00:00,  1.25it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.32it/s]
                     all      0.879          1

      Epoch    GPU_mem       loss  Instances       Size
      8/50         0G     0.3139          5        224: 100%|██████████| 521/521 [06:56<00:00,  1.25it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.33it/s]
                     all      0.859          1

      Epoch    GPU_mem       loss  Instances       Size
      9/50         0G     0.3022          5        224: 100%|██████████| 521/521 [06:59<00:00,  1.24it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.32it/s]
                     all      0.865          1

      Epoch    GPU_mem       loss  Instances       Size
     10/50         0G     0.2823          5        224: 100%|██████████| 521/521 [06:56<00:00,  1.25it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.34it/s]
                     all      0.876          1

      Epoch    GPU_mem       loss  Instances       Size
     11/50         0G     0.2834          5        224: 100%|██████████| 521/521 [06:56<00:00,  1.25it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:08<00:00,  1.32it/s]
                     all      0.871          1

      Epoch    GPU_mem       loss  Instances       Size
     12/50         0G     0.2683          5        224: 100%|██████████| 521/521 [05:40<00:00,  1.53it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.91it/s]
                     all      0.885          1

      Epoch    GPU_mem       loss  Instances       Size
     13/50         0G      0.253          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.89it/s]
                     all      0.859          1

      Epoch    GPU_mem       loss  Instances       Size
     14/50         0G     0.2535          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.92it/s]
                     all      0.865          1

      Epoch    GPU_mem       loss  Instances       Size
     15/50         0G     0.2407          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.90it/s]
                     all      0.874          1

      Epoch    GPU_mem       loss  Instances       Size
     16/50         0G     0.2264          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.95it/s]
                     all      0.862          1

      Epoch    GPU_mem       loss  Instances       Size
     17/50         0G     0.2097          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.86it/s]
                     all      0.836          1

      Epoch    GPU_mem       loss  Instances       Size
     18/50         0G     0.2013          5        224: 100%|██████████| 521/521 [04:38<00:00,  1.87it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.88it/s]
                     all      0.874          1

      Epoch    GPU_mem       loss  Instances       Size
     19/50         0G     0.1969          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
                 classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.92it/s]
                     all      0.862          1
```

```
      Epoch    GPU_mem       loss  Instances       Size
      20/50         0G     0.2022          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  2.07it/s]
                    all      0.871          1

      Epoch    GPU_mem       loss  Instances       Size
      21/50         0G     0.1847          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.88it/s]
                    all      0.882          1

      Epoch    GPU_mem       loss  Instances       Size
      22/50         0G     0.1766          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  2.04it/s]
                    all      0.882          1

      Epoch    GPU_mem       loss  Instances       Size
      23/50         0G     0.1684          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.95it/s]
                    all      0.856          1

      Epoch    GPU_mem       loss  Instances       Size
      24/50         0G     0.1705          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.94it/s]
                    all      0.871          1

      Epoch    GPU_mem       loss  Instances       Size
      25/50         0G     0.1625          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.92it/s]
                    all      0.871          1

      Epoch    GPU_mem       loss  Instances       Size
      26/50         0G     0.1505          5        224: 100%|██████████| 521/521 [04:36<00:00,  1.89it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.89it/s]
                    all      0.888          1

      Epoch    GPU_mem       loss  Instances       Size
      27/50         0G     0.1489          5        224: 100%|██████████| 521/521 [04:40<00:00,  1.86it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.95it/s]
                    all      0.865          1

      Epoch    GPU_mem       loss  Instances       Size
      28/50         0G     0.1499          5        224: 100%|██████████| 521/521 [04:36<00:00,  1.88it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.90it/s]
                    all      0.868          1

      Epoch    GPU_mem       loss  Instances       Size
      29/50         0G     0.1485          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  2.16it/s]
                    all      0.859          1

      Epoch    GPU_mem       loss  Instances       Size
      30/50         0G     0.1352          5        224: 100%|██████████| 521/521 [04:38<00:00,  1.87it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.93it/s]
                    all      0.859          1

      Epoch    GPU_mem       loss  Instances       Size
      31/50         0G     0.1425          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.90it/s]
                    all      0.862          1

      Epoch    GPU_mem       loss  Instances       Size
      32/50         0G     0.1249          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  2.04it/s]
                    all      0.874          1

      Epoch    GPU_mem       loss  Instances       Size
      33/50         0G     0.1215          5        224: 100%|██████████| 521/521 [04:36<00:00,  1.88it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.89it/s]
                    all      0.865          1

      Epoch    GPU_mem       loss  Instances       Size
      34/50         0G     0.1199          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.89it/s]
                    all      0.879          1

      Epoch    GPU_mem       loss  Instances       Size
      35/50         0G     0.1147          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.90it/s]
                    all      0.885          1

      Epoch    GPU_mem       loss  Instances       Size
      36/50         0G      0.109          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
                classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.90it/s]
                    all      0.874          1

      Epoch    GPU_mem       loss  Instances       Size
```

```
      37/50         0G     0.1058          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]

               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.93it/s]
                   all      0.871          1
      Epoch    GPU_mem       loss  Instances       Size
      38/50         0G     0.1029          5        224: 100%|██████████| 521/521 [04:29<00:00,  1.93it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.98it/s]
                   all      0.879          1
      Epoch    GPU_mem       loss  Instances       Size
      39/50         0G    0.09994          5        224: 100%|██████████| 521/521 [04:31<00:00,  1.92it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.91it/s]
                   all      0.874          1
      Epoch    GPU_mem       loss  Instances       Size
      40/50         0G     0.0961          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.94it/s]
                   all      0.879          1
      Epoch    GPU_mem       loss  Instances       Size
      41/50         0G     0.0996          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.94it/s]
                   all      0.871          1
      Epoch    GPU_mem       loss  Instances       Size
      42/50         0G     0.0965          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.93it/s]
                   all      0.853          1
      Epoch    GPU_mem       loss  Instances       Size
      43/50         0G    0.09583          5        224: 100%|██████████| 521/521 [04:35<00:00,  1.89it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.90it/s]
                   all      0.865          1
      Epoch    GPU_mem       loss  Instances       Size
      44/50         0G    0.09209          5        224: 100%|██████████| 521/521 [04:32<00:00,  1.91it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.93it/s]
                   all      0.865          1
      Epoch    GPU_mem       loss  Instances       Size
      45/50         0G    0.09171          5        224: 100%|██████████| 521/521 [04:36<00:00,  1.88it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  2.06it/s]
                   all      0.865          1
      Epoch    GPU_mem       loss  Instances       Size
      46/50         0G    0.08352          5        224: 100%|██████████| 521/521 [04:40<00:00,  1.86it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.94it/s]
                   all      0.865          1
      Epoch    GPU_mem       loss  Instances       Size
      47/50         0G    0.08048          5        224: 100%|██████████| 521/521 [04:38<00:00,  1.87it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.98it/s]
                   all      0.865          1
      Epoch    GPU_mem       loss  Instances       Size
      48/50         0G    0.07895          5        224: 100%|██████████| 521/521 [04:37<00:00,  1.88it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.89it/s]
                   all      0.868          1
      Epoch    GPU_mem       loss  Instances       Size
      49/50         0G    0.08348          5        224: 100%|██████████| 521/521 [04:36<00:00,  1.88it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.94it/s]
                   all      0.862          1
      Epoch    GPU_mem       loss  Instances       Size
      50/50         0G    0.07728          5        224: 100%|██████████| 521/521 [04:34<00:00,  1.90it/s]
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  1.91it/s]
                   all      0.862          1

50 epochs completed in 4.569 hours.
Optimizer stripped from runs\classify\train2\weights\last.pt, 3.0MB
Optimizer stripped from runs\classify\train2\weights\best.pt, 3.0MB

Validating runs\classify\train2\weights\best.pt...
Ultralytics YOLOv8.1.29  Python-3.10.11 torch-2.2.1+cpu CPU (11th Gen Intel Core(TM) i5-11300H 3.10GHz)
YOLOv8n-cls summary (fused): 73 layers, 1440004 parameters, 0 gradients, 3.3 GFLOPs
WARNING ⚠  Dataset 'split=val' not found, using 'split=test' instead.
train: C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\train... found 8325 images in 4 c
lasses ✅
val: None...
test: C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\test... found 348 images in 4 clas
ses ✅
               classes   top1_acc   top5_acc: 100%|██████████| 11/11 [00:05<00:00,  2.01it/s]
                   all      0.908          1
Speed: 0.0ms preprocess, 7.1ms inference, 0.0ms loss, 0.0ms postprocess per image
Results saved to runs\classify\train2
Results saved to runs\classify\train2
```

Out[6]:
```
ultralytics.utils.metrics.ClassifyMetrics object with attributes:

confusion_matrix: <ultralytics.utils.metrics.ConfusionMatrix object at 0x000002AE17BFC3D0>
curves: []
curves_results: []
fitness: 0.9540229737758636
keys: ['metrics/accuracy_top1', 'metrics/accuracy_top5']
results_dict: {'metrics/accuracy_top1': 0.9080459475517273, 'metrics/accuracy_top5': 1.0, 'fitness': 0.9540229737758636}
save_dir: WindowsPath('runs/classify/train2')
speed: {'preprocess': 0.0, 'inference': 7.118260723420943, 'loss': 0.0, 'postprocess': 0.0}
task: 'classify'
top1: 0.9080459475517273
top5: 1.0
```

Model Result's

In [7]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [10]:
```
results_path = "runs/classify/train2/results.csv"
```

In [11]:
```
results = pd.read_csv(results_path)
```

In [12]:
```
results
```

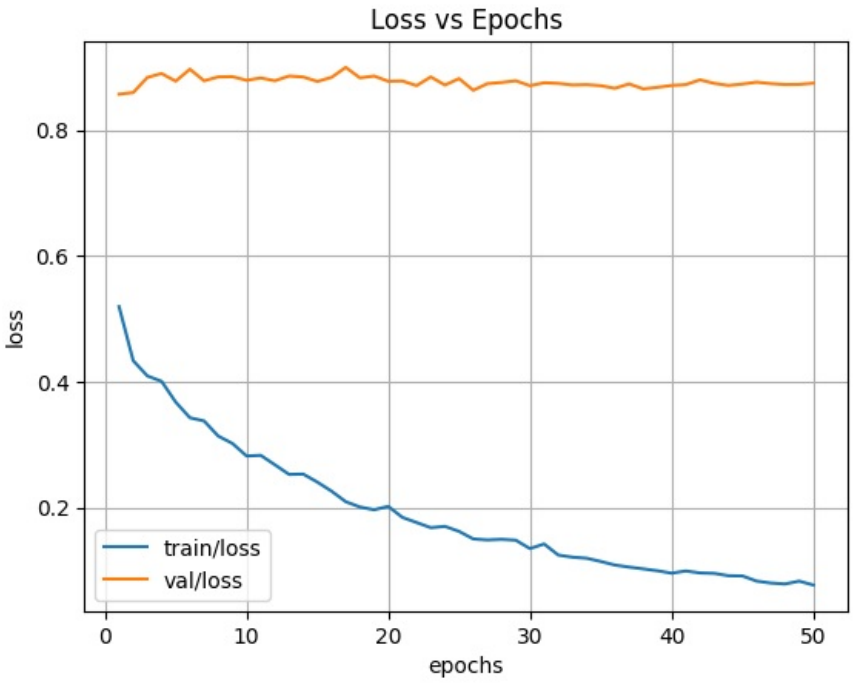
Out[12]:

	epoch	train/loss	metrics/accuracy_top1	metrics/accuracy_top5	val/loss	lr/pg0	lr/pg1	lr/pg2
0	1	0.51961	0.90805	1	0.85712	0.000238	0.000238	0.000238
1	2	0.43373	0.90517	1	0.85970	0.000466	0.000466	0.000466
2	3	0.40948	0.89080	1	0.88379	0.000685	0.000685	0.000685
3	4	0.40097	0.87356	1	0.88995	0.000672	0.000672	0.000672
4	5	0.36793	0.89655	1	0.87787	0.000672	0.000672	0.000672
5	6	0.34304	0.85345	1	0.89685	0.000657	0.000657	0.000657
6	7	0.33800	0.87931	1	0.87841	0.000643	0.000643	0.000643
7	8	0.31395	0.85920	1	0.88460	0.000629	0.000629	0.000629
8	9	0.30225	0.86494	1	0.88502	0.000615	0.000615	0.000615
9	10	0.28233	0.87644	1	0.87919	0.000601	0.000601	0.000601
10	11	0.28337	0.87069	1	0.88292	0.000587	0.000587	0.000587
11	12	0.26830	0.88506	1	0.87851	0.000573	0.000573	0.000573
12	13	0.25302	0.85920	1	0.88599	0.000558	0.000558	0.000558
13	14	0.25354	0.86494	1	0.88448	0.000544	0.000544	0.000544
14	15	0.24075	0.87356	1	0.87736	0.000530	0.000530	0.000530
15	16	0.22637	0.86207	1	0.88388	0.000516	0.000516	0.000516
16	17	0.20966	0.83621	1	0.89975	0.000502	0.000502	0.000502
17	18	0.20127	0.87356	1	0.88307	0.000488	0.000488	0.000488
18	19	0.19686	0.86207	1	0.88598	0.000474	0.000474	0.000474
19	20	0.20223	0.87069	1	0.87769	0.000460	0.000460	0.000460
20	21	0.18473	0.88218	1	0.87811	0.000445	0.000445	0.000445
21	22	0.17659	0.88218	1	0.87068	0.000431	0.000431	0.000431
22	23	0.16841	0.85632	1	0.88460	0.000417	0.000417	0.000417
23	24	0.17048	0.87069	1	0.87167	0.000403	0.000403	0.000403
24	25	0.16249	0.87069	1	0.88188	0.000389	0.000389	0.000389
25	26	0.15049	0.88793	1	0.86333	0.000375	0.000375	0.000375
26	27	0.14894	0.86494	1	0.87391	0.000361	0.000361	0.000361
27	28	0.14990	0.86782	1	0.87576	0.000346	0.000346	0.000346
28	29	0.14853	0.85920	1	0.87844	0.000332	0.000332	0.000332
29	30	0.13516	0.85920	1	0.87058	0.000318	0.000318	0.000318
30	31	0.14251	0.86207	1	0.87527	0.000304	0.000304	0.000304
31	32	0.12488	0.87356	1	0.87429	0.000290	0.000290	0.000290
32	33	0.12152	0.86494	1	0.87177	0.000276	0.000276	0.000276

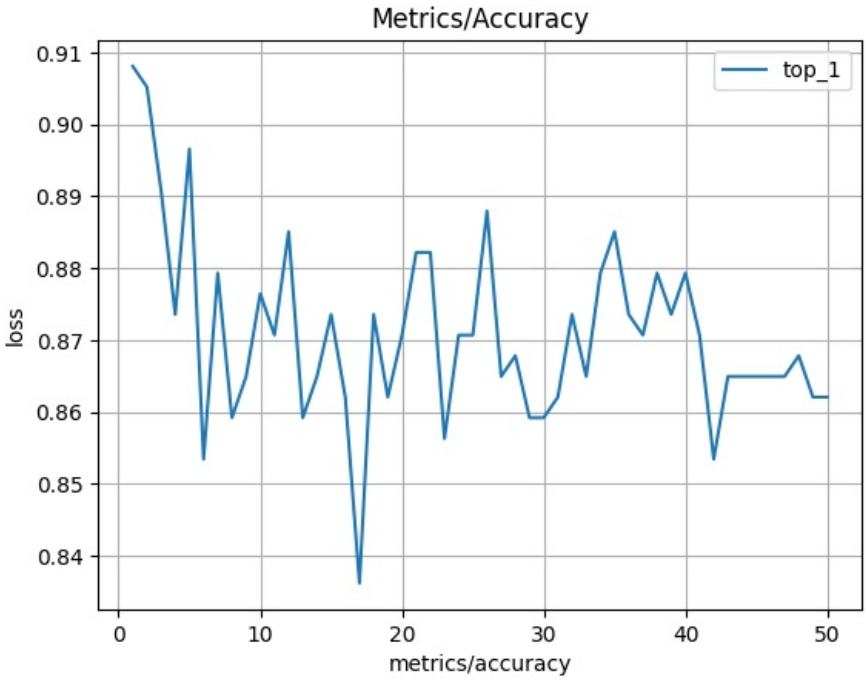
33	34	0.11986	0.87931	1	0.87229	0.000262	0.000262	0.000262
34	35	0.11470	0.88506	1	0.87062	0.000247	0.000247	0.000247
35	36	0.10901	0.87356	1	0.86651	0.000233	0.000233	0.000233
36	37	0.10579	0.87069	1	0.87348	0.000219	0.000219	0.000219
37	38	0.10293	0.87931	1	0.86556	0.000205	0.000205	0.000205
38	39	0.09994	0.87356	1	0.86805	0.000191	0.000191	0.000191
39	40	0.09610	0.87931	1	0.87097	0.000177	0.000177	0.000177
40	41	0.09960	0.87069	1	0.87234	0.000163	0.000163	0.000163
41	42	0.09650	0.85345	1	0.87993	0.000149	0.000149	0.000149
42	43	0.09583	0.86494	1	0.87435	0.000134	0.000134	0.000134
43	44	0.09209	0.86494	1	0.87104	0.000120	0.000120	0.000120
44	45	0.09171	0.86494	1	0.87321	0.000106	0.000106	0.000106
45	46	0.08352	0.86494	1	0.87626	0.000092	0.000092	0.000092
46	47	0.08048	0.86494	1	0.87394	0.000078	0.000078	0.000078
47	48	0.07895	0.86782	1	0.87248	0.000064	0.000064	0.000064
48	49	0.08348	0.86207	1	0.87266	0.000050	0.000050	0.000050
49	50	0.07728	0.86207	1	0.87461	0.000035	0.000035	0.000035

In [37]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.figure()
plt.plot(results['                epoch'], results['            train/loss'], label='train/loss')
plt.plot(results['                epoch'], results['              val/loss'], label='val/loss')
plt.grid()
plt.title('Loss vs Epochs')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend()

plt.show()
```



In [36]:
```
plt.figure()
plt.plot(results['                epoch'], results['  metrics/accuracy_top1'], label =  'top_1')
plt.grid()
plt.title('Metrics/Accuracy')
plt.ylabel('loss')
plt.xlabel('metrics/accuracy')
plt.legend()
plt.show()
```



In [61]:
```
import numpy as np
result = model.predict(source = "human_emmotion-1/test/happy/112_jpg.rf.04fdb5f724e90979af2c0c7c661dcd75.jpg")
img = plt.imread("human_emmotion-1/test/happy/112_jpg.rf.04fdb5f724e90979af2c0c7c661dcd75.jpg")
```

```
image 1/1 C:\Users\Satoshi\OneDrive\Desktop\mini-projects\Images\human_emmotion-1\test\happy\112_jpg.rf.04fdb5f7
24e90979af2c0c7c661dcd75.jpg: 224x224 happy 1.00, angry 0.00, sad 0.00, nothing 0.00, 45.8ms
Speed: 6.4ms preprocess, 45.8ms inference, 0.0ms postprocess per image at shape (1, 3, 224, 224)
```

In [63]:
```
plt.imshow(img)
plt.axis('off')
print('predicted class is : ',class_names[np.argmax(result[0].probs.data)])
```

```
predicted class is :  happy
```

In [ ]:

In [ ]: