Assignment 4 YuTing Chiu

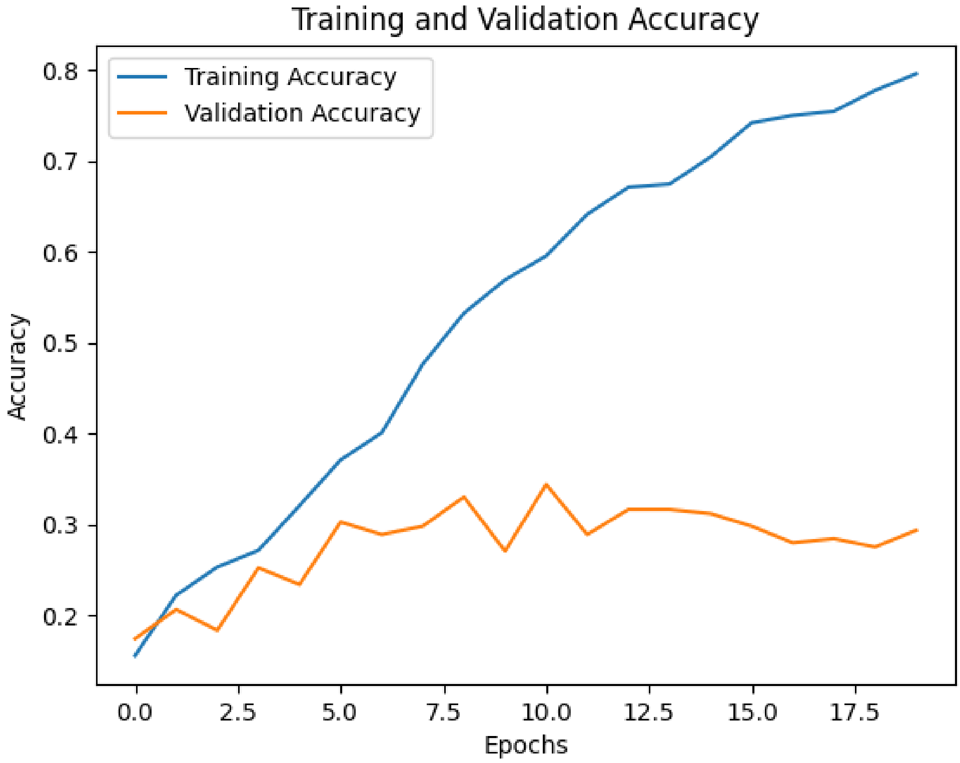
Task1 :

For Task1_model1:

The dataset is split into training (80%) and validation (20%) subsets by specifying validation_split=0.2

The CNN model is created using Sequential, with the following layers:

1. **Input Layer:** Accepts images of shape (100, 100, 3) (3 channels for RGB).
2. **Convolutional Layers (Conv2D):**
   Extract spatial features using 32 and 64 filters of size (3, 3), both activated by ReLU.
3. **MaxPooling Layers (MaxPooling2D):**
   Reduce spatial dimensions using a pool size of (2, 2) after each convolutional layer.
4. **Dropout Layers:**
   Add dropout (20%, 30% rates) after convolution and pooling layers to prevent overfitting.
5. **Flatten Layer:**
   Flatten the 2D feature maps into 1D arrays for fully connected layers.
6. **Dense Layers:**
   A fully connected layer with 128 neurons and ReLU activation.
   Another dropout (50%) to further reduce overfitting.
   The output layer with num_classes neurons and softmax activation for multi-class classification.




```
Epoch 1/20
28/28 - 3s - 103ms/step - accuracy: 0.1558 - loss: 179.7453 - val_accuracy: 0.1743 - val_loss: 1.7760
Epoch 2/20
28/28 - 1s - 51ms/step - accuracy: 0.2222 - loss: 1.7919 - val_accuracy: 0.2064 - val_loss: 1.7806
Epoch 3/20
28/28 - 1s - 52ms/step - accuracy: 0.2532 - loss: 1.7596 - val_accuracy: 0.1835 - val_loss: 1.7754
Epoch 4/20
28/28 - 1s - 50ms/step - accuracy: 0.2715 - loss: 1.7025 - val_accuracy: 0.2523 - val_loss: 1.7346
Epoch 5/20
28/28 - 1s - 51ms/step - accuracy: 0.3207 - loss: 1.6533 - val_accuracy: 0.2339 - val_loss: 1.7390
Epoch 6/20
28/28 - 1s - 51ms/step - accuracy: 0.3711 - loss: 1.5478 - val_accuracy: 0.3028 - val_loss: 1.7468
Epoch 7/20
28/28 - 1s - 50ms/step - accuracy: 0.4009 - loss: 1.4490 - val_accuracy: 0.2890 - val_loss: 1.7803
Epoch 8/20
28/28 - 1s - 49ms/step - accuracy: 0.4765 - loss: 1.3363 - val_accuracy: 0.2982 - val_loss: 1.8350
Epoch 9/20
28/28 - 1s - 50ms/step - accuracy: 0.5326 - loss: 1.2099 - val_accuracy: 0.3303 - val_loss: 1.8207
Epoch 10/20
28/28 - 1s - 51ms/step - accuracy: 0.5693 - loss: 1.1009 - val_accuracy: 0.2706 - val_loss: 2.0038
Epoch 11/20
28/28 - 1s - 50ms/step - accuracy: 0.5956 - loss: 1.0629 - val_accuracy: 0.3440 - val_loss: 1.9941
Epoch 12/20
28/28 - 1s - 50ms/step - accuracy: 0.6415 - loss: 0.9684 - val_accuracy: 0.2890 - val_loss: 2.3610
Epoch 13/20
28/28 - 1s - 50ms/step - accuracy: 0.6712 - loss: 0.8820 - val_accuracy: 0.3165 - val_loss: 2.4312
Epoch 14/20
28/28 - 1s - 50ms/step - accuracy: 0.6747 - loss: 0.8492 - val_accuracy: 0.3165 - val_loss: 2.4752
Epoch 15/20
28/28 - 1s - 50ms/step - accuracy: 0.7045 - loss: 0.8367 - val_accuracy: 0.3119 - val_loss: 2.4860
Epoch 16/20
28/28 - 1s - 50ms/step - accuracy: 0.7423 - loss: 0.7342 - val_accuracy: 0.2982 - val_loss: 2.8075
Epoch 17/20
28/28 - 1s - 50ms/step - accuracy: 0.7503 - loss: 0.6637 - val_accuracy: 0.2798 - val_loss: 3.0017
Epoch 18/20
28/28 - 1s - 51ms/step - accuracy: 0.7549 - loss: 0.6949 - val_accuracy: 0.2844 - val_loss: 2.9286
Epoch 19/20
28/28 - 1s - 52ms/step - accuracy: 0.7778 - loss: 0.6062 - val_accuracy: 0.2752 - val_loss: 3.0514
Epoch 20/20
28/28 - 1s - 50ms/step - accuracy: 0.7961 - loss: 0.6216 - val_accuracy: 0.2936 - val_loss: 3.4041
7/7 - 0s - 10ms/step - accuracy: 0.2936 - loss: 3.4041
Test Accuracy: 0.294
Model saved successfully.
```

For Task1_model2:

A simple CNN is built with the following layers:

1. **Input Layer:** Accepts images of size (224, 224, 3) (RGB format).
2. **Convolutional Layers (Conv2D):**

   Two convolutional layers with 32 and 64 filters respectively, each using a kernel size of (3, 3) and ReLU activation.
3. **MaxPooling Layers (MaxPooling2D):**

   Pooling layers with size (2, 2) to downsample spatial dimensions.
4. **Dropout Layers:**

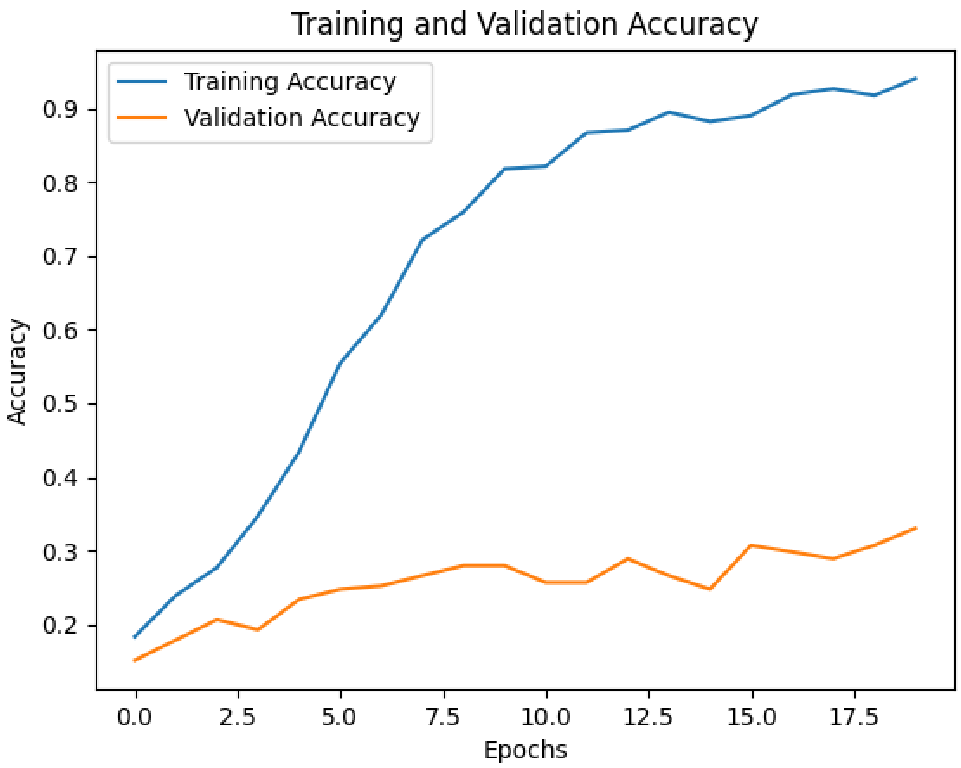
   Dropout (20% and 30%) to reduce overfitting by randomly deactivating neurons during training.
5. **Flatten Layer:**

   Converts the 2D feature maps into 1D vectors for the Dense layers.
6. **Dense Layers:**

   A fully connected layer with 128 neurons and ReLU activation.

   Another dropout (50%) to further reduce overfitting.

   The output layer has neurons equal to num_classes with softmax activation for multi-class classification.




```
Epoch 1/20
28/28 - 10s - 375ms/step - accuracy: 0.1833 - loss: 467.2643 - val_accuracy: 0.1514 - val_loss: 1.7957
Epoch 2/20
28/28 - 9s - 321ms/step - accuracy: 0.2394 - loss: 1.7817 - val_accuracy: 0.1789 - val_loss: 1.7931
Epoch 3/20
28/28 - 9s - 314ms/step - accuracy: 0.2772 - loss: 1.7135 - val_accuracy: 0.2064 - val_loss: 1.7952
Epoch 4/20
28/28 - 9s - 313ms/step - accuracy: 0.3471 - loss: 1.5875 - val_accuracy: 0.1927 - val_loss: 1.8313
Epoch 5/20
28/28 - 9s - 311ms/step - accuracy: 0.4341 - loss: 1.4085 - val_accuracy: 0.2339 - val_loss: 1.8488
Epoch 6/20
28/28 - 9s - 317ms/step - accuracy: 0.5544 - loss: 1.2337 - val_accuracy: 0.2477 - val_loss: 1.9582
Epoch 7/20
28/28 - 9s - 315ms/step - accuracy: 0.6197 - loss: 1.0640 - val_accuracy: 0.2523 - val_loss: 2.1455
Epoch 8/20
28/28 - 9s - 308ms/step - accuracy: 0.7216 - loss: 0.7745 - val_accuracy: 0.2661 - val_loss: 2.3379
Epoch 9/20
28/28 - 9s - 314ms/step - accuracy: 0.7595 - loss: 0.6876 - val_accuracy: 0.2798 - val_loss: 2.8887
Epoch 10/20
28/28 - 9s - 314ms/step - accuracy: 0.8179 - loss: 0.6166 - val_accuracy: 0.2798 - val_loss: 2.7779
Epoch 11/20
28/28 - 10s - 339ms/step - accuracy: 0.8213 - loss: 0.5115 - val_accuracy: 0.2569 - val_loss: 3.3389
Epoch 12/20
28/28 - 9s - 325ms/step - accuracy: 0.8671 - loss: 0.4002 - val_accuracy: 0.2569 - val_loss: 3.4533
Epoch 13/20
28/28 - 9s - 315ms/step - accuracy: 0.8706 - loss: 0.4281 - val_accuracy: 0.2890 - val_loss: 3.9898
Epoch 14/20
28/28 - 9s - 321ms/step - accuracy: 0.8946 - loss: 0.3645 - val_accuracy: 0.2661 - val_loss: 4.0241
Epoch 15/20
28/28 - 9s - 315ms/step - accuracy: 0.8820 - loss: 0.3838 - val_accuracy: 0.2477 - val_loss: 3.0567
Epoch 16/20
28/28 - 9s - 313ms/step - accuracy: 0.8900 - loss: 0.3393 - val_accuracy: 0.3073 - val_loss: 4.5842
Epoch 17/20
28/28 - 9s - 319ms/step - accuracy: 0.9187 - loss: 0.2915 - val_accuracy: 0.2982 - val_loss: 4.8731
Epoch 18/20
28/28 - 9s - 315ms/step - accuracy: 0.9267 - loss: 0.3638 - val_accuracy: 0.2890 - val_loss: 4.3165
Epoch 19/20
28/28 - 9s - 313ms/step - accuracy: 0.9175 - loss: 0.2862 - val_accuracy: 0.3073 - val_loss: 5.6307
Epoch 20/20
28/28 - 11s - 387ms/step - accuracy: 0.9404 - loss: 0.3087 - val_accuracy: 0.3303 - val_loss: 4.8635
7/7 - 0s - 62ms/step - accuracy: 0.3303 - loss: 4.8635
Test Accuracy: 0.330
Model saved successfully.
```

Comparison Between Two Models

Input Shape	(100, 100, 3)	(224, 224, 3)
Convolutional Layers	Conv2D with 32 filters and kernel size (3, 3)	Conv2D with 32 filters and kernel size (3, 3)
	Conv2D with 64 filters and kernel size (3, 3)	Conv2D with 64 filters and kernel size (3, 3)
Pooling Layers	Two MaxPooling2D layers with (2, 2) pooling.	Two MaxPooling2D layers with (2, 2) pooling.
Dropout Layers	Dropout layers after convolution and dense layers with rates: 0.2, 0.3, 0.5.	Dropout layers after convolution and dense layers with rates: 0.2, 0.3, 0.5.
Dense Layers	One fully connected layer with 128 neurons (ReLU) and a softmax output layer.	One fully connected layer with 128 neurons (ReLU) and a softmax output layer.
Output Layer	Softmax activation for multi-class classification.	Softmax activation for multi-class classification.

Aspect	Model 1	Model 2
Training Accuracy	79.6%	94.0%
Validation Accuracy	29.3%	33.0%
Test Accuracy	29.4%	33.0%
Test Loss	3.4041	4.8635
Training Speed	Faster (~50ms/step)	Slower (~315ms/step)
Input Size	(100x100)	(224x224)

Model 2 shows slightly better performance in terms of validation and test accuracy (33% vs. 29.3% for Model 1) due to its higher input resolution, but it suffers from significantly higher computational cost and overfitting, similar to Model 1. Both models have limited generalization, as indicated by the low validation and test accuracy. Improving data preprocessing (e.g., augmentation) and using transfer learning could help boost performance while addressing overfitting. For resource efficiency, Model 1 is preferable unless further optimizations are applied to Model 2.

Task2:

(i)Pre-trained Model and Added Layers

Pre-trained Model:

EfficientNetV2B3 with ImageNet weights.

Model Size: ~12 million parameters.

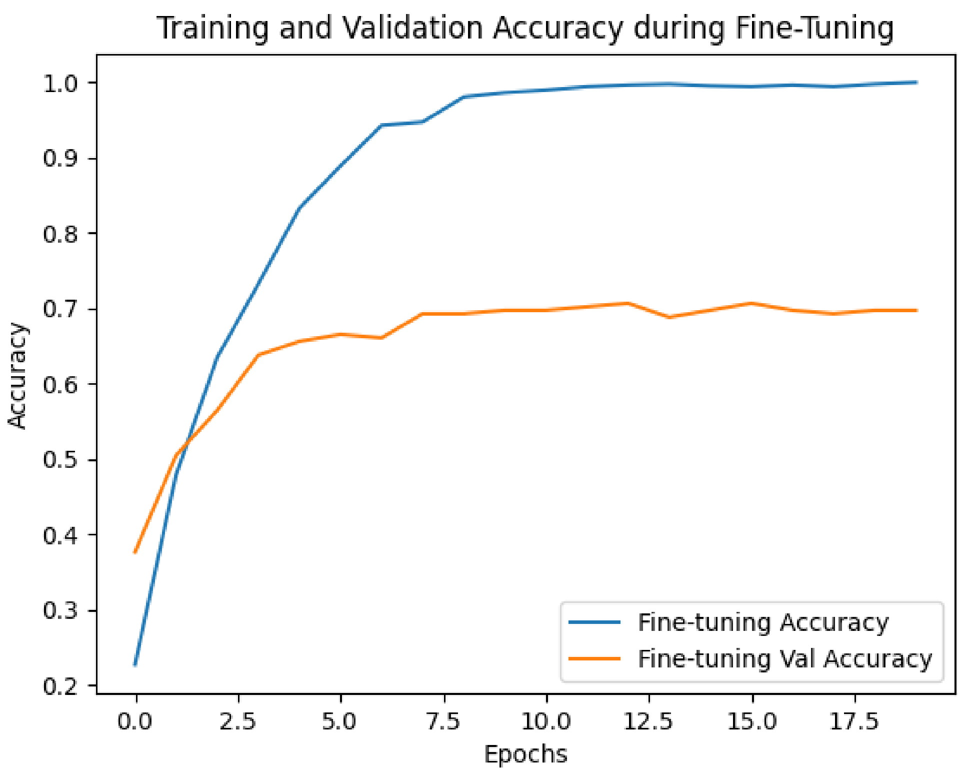
Added Layers:

GlobalAveragePooling2D: To reduce spatial dimensions to a vector.

Dense (128 neurons): Fully connected layer with ReLU activation.

Dropout (rate=0.5): Regularization to prevent overfitting.

Dense (output neurons = number of classes): Softmax layer for classification.



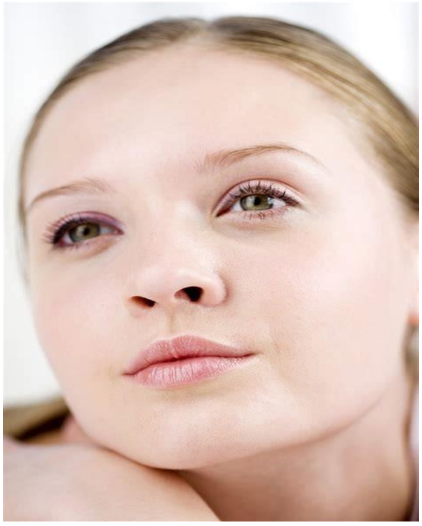
```
Epoch 1/20
28/28 - 114s - 4s/step - accuracy: 0.2268 - loss: 1.7780 - val_accuracy: 0.3761 - val_loss: 1.6320
Epoch 2/20
28/28 - 57s - 2s/step - accuracy: 0.4788 - loss: 1.4774 - val_accuracy: 0.5046 - val_loss: 1.4783
Epoch 3/20
28/28 - 56s - 2s/step - accuracy: 0.6346 - loss: 1.1939 - val_accuracy: 0.5642 - val_loss: 1.3040
Epoch 4/20
28/28 - 56s - 2s/step - accuracy: 0.7320 - loss: 0.9454 - val_accuracy: 0.6376 - val_loss: 1.1510
Epoch 5/20
28/28 - 61s - 2s/step - accuracy: 0.8328 - loss: 0.6715 - val_accuracy: 0.6560 - val_loss: 1.0422
Epoch 6/20
28/28 - 63s - 2s/step - accuracy: 0.8889 - loss: 0.4776 - val_accuracy: 0.6651 - val_loss: 0.9760
Epoch 7/20
28/28 - 60s - 2s/step - accuracy: 0.9427 - loss: 0.3100 - val_accuracy: 0.6606 - val_loss: 0.9485
Epoch 8/20
28/28 - 64s - 2s/step - accuracy: 0.9473 - loss: 0.2270 - val_accuracy: 0.6927 - val_loss: 0.9553
Epoch 9/20
28/28 - 62s - 2s/step - accuracy: 0.9805 - loss: 0.1409 - val_accuracy: 0.6927 - val_loss: 0.9566
Epoch 10/20
28/28 - 59s - 2s/step - accuracy: 0.9863 - loss: 0.1072 - val_accuracy: 0.6972 - val_loss: 0.9908
Epoch 11/20
28/28 - 61s - 2s/step - accuracy: 0.9897 - loss: 0.0755 - val_accuracy: 0.6972 - val_loss: 1.0193
Epoch 12/20
28/28 - 60s - 2s/step - accuracy: 0.9943 - loss: 0.0544 - val_accuracy: 0.7018 - val_loss: 1.0695
Epoch 13/20
28/28 - 62s - 2s/step - accuracy: 0.9966 - loss: 0.0405 - val_accuracy: 0.7064 - val_loss: 1.0846
Epoch 14/20
28/28 - 62s - 2s/step - accuracy: 0.9977 - loss: 0.0398 - val_accuracy: 0.6881 - val_loss: 1.1785
Epoch 15/20
28/28 - 60s - 2s/step - accuracy: 0.9954 - loss: 0.0303 - val_accuracy: 0.6972 - val_loss: 1.1621
Epoch 16/20
28/28 - 61s - 2s/step - accuracy: 0.9943 - loss: 0.0327 - val_accuracy: 0.7064 - val_loss: 1.1764
Epoch 17/20
28/28 - 61s - 2s/step - accuracy: 0.9966 - loss: 0.0300 - val_accuracy: 0.6972 - val_loss: 1.2491
Epoch 18/20
28/28 - 62s - 2s/step - accuracy: 0.9943 - loss: 0.0262 - val_accuracy: 0.6927 - val_loss: 1.2777
Epoch 19/20
28/28 - 62s - 2s/step - accuracy: 0.9977 - loss: 0.0208 - val_accuracy: 0.6972 - val_loss: 1.3347
Epoch 20/20
28/28 - 61s - 2s/step - accuracy: 1.0000 - loss: 0.0164 - val_accuracy: 0.6972 - val_loss: 1.3286
7/7 - 3s - 365ms/step - accuracy: 0.6972 - loss: 1.3286
Test Accuracy after fine-tuning: 0.697
Model saved successfully.
```

Model	Test Accuracy
Fine-Tuned Model (Task 2)	69.7%
Better Model (Task 1)	33.0%

The fine-tuned model using EfficientNetV2B3 significantly outperformed the Task 1 models, achieving a much higher test accuracy (69.7% vs. 33.0%). This improvement highlights the effectiveness of transfer learning, leveraging pre-trained features for better generalization. The added layers complemented the pre-trained model, adapting it to the specific dataset while preventing overfitting. This model is recommended for use, given sufficient computational resources.

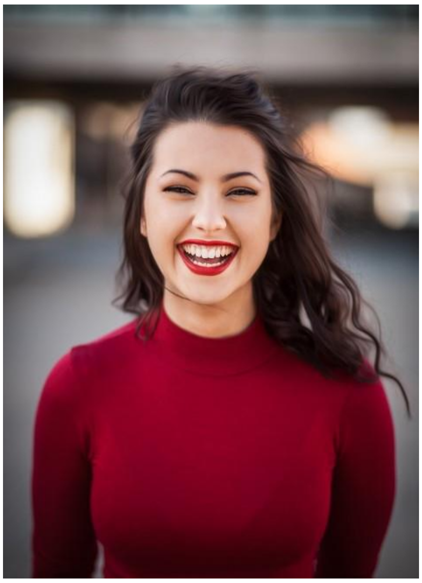
Task3:

Pic 1: Sad



Pic 2 happy

Pic 3 happy



Pic 4 happy

Pic 5 sad

Pic 6 happy

Pic 7 happy

Pic 8 happy

Pic 9 sad

Pic 10 happy

```
import image_test_task2_model
Model loaded successfully.
Class names: ['anger', 'happy', 'disgust', 'sad', 'pain', 'fear']
File: ./Facial Pics\1.jpg, Predicted Class: sad
File: ./Facial Pics\10.jpg, Predicted Class: sad
File: ./Facial Pics\2.jpg, Predicted Class: sad
File: ./Facial Pics\3.jpg, Predicted Class: sad
File: ./Facial Pics\4.jpg, Predicted Class: sad
File: ./Facial Pics\5.jpg, Predicted Class: sad
File: ./Facial Pics\6.jpg, Predicted Class: sad
File: ./Facial Pics\7.jpg, Predicted Class: sad
File: ./Facial Pics\8.jpg, Predicted Class: sad
File: ./Facial Pics\9.jpg, Predicted Class: sad
import image_test_task1_best_model
Model loaded successfully.
Class names: ['anger', 'happy', 'disgust', 'sad', 'pain', 'fear']
File: ./Facial Pics\1.jpg, Predicted Class: pain
File: ./Facial Pics\10.jpg, Predicted Class: pain
File: ./Facial Pics\2.jpg, Predicted Class: pain
File: ./Facial Pics\3.jpg, Predicted Class: disgust
File: ./Facial Pics\4.jpg, Predicted Class: pain
File: ./Facial Pics\5.jpg, Predicted Class: pain
File: ./Facial Pics\6.jpg, Predicted Class: pain
File: ./Facial Pics\7.jpg, Predicted Class: disgust
File: ./Facial Pics\8.jpg, Predicted Class: disgust
File: ./Facial Pics\9.jpg, Predicted Class: disgust
```

1. Task 1 Model Errors

Simpler Architecture: Task 1 model is too simple to capture the subtle features of emotions, leading to frequent misclassifications like "Pain" or "Disgust."

Poor Generalization: It struggles to generalize well to unseen data, likely overfitting to the training set.

Low Input Resolution: Even with 224x224, the model lacks the capacity to extract detailed features needed for emotion recognition.

2. Task 2 Model Errors

Class Bias: The Task 2 model predicts "Sad" for most images, possibly due to class imbalance in the training dataset.

Pretrained Features Limitation: Although it uses EfficientNetV2B3, which is pretrained on ImageNet, those features are better suited for object detection rather than subtle facial expressions.

Conclusion:

The original dataset may be too small to effectively train the models, resulting in both Task 1 and Task 2 models having low accuracy and struggling to distinguish images in the new dataset. The limited data likely caused insufficient learning and poor generalization, leading to biases and errors in predictions. Expanding the dataset with more diverse and balanced samples or using specialized emotion recognition datasets could significantly improve the models' performance.