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.

1. **MaxPooling Layers (MaxPooling2D):**

Reduce spatial dimensions using a pool size of (2, 2) after each convolutional layer.

1. **Dropout Layers:**

Add dropout (20%, 30% rates) after convolution and pooling layers to prevent overfitting.

1. **Flatten Layer:**

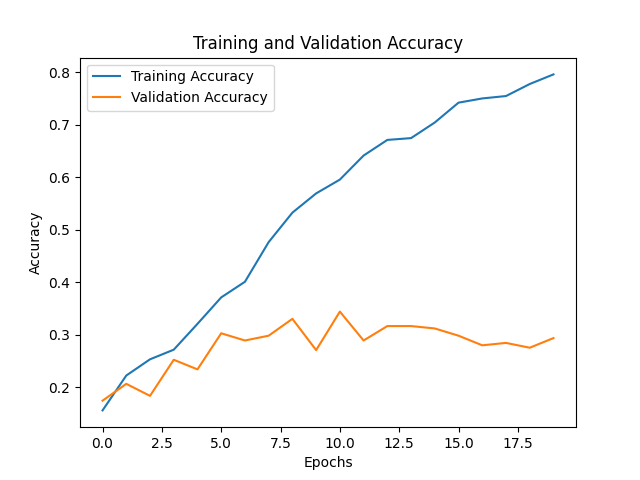
Flatten the 2D feature maps into 1D arrays for fully connected layers.

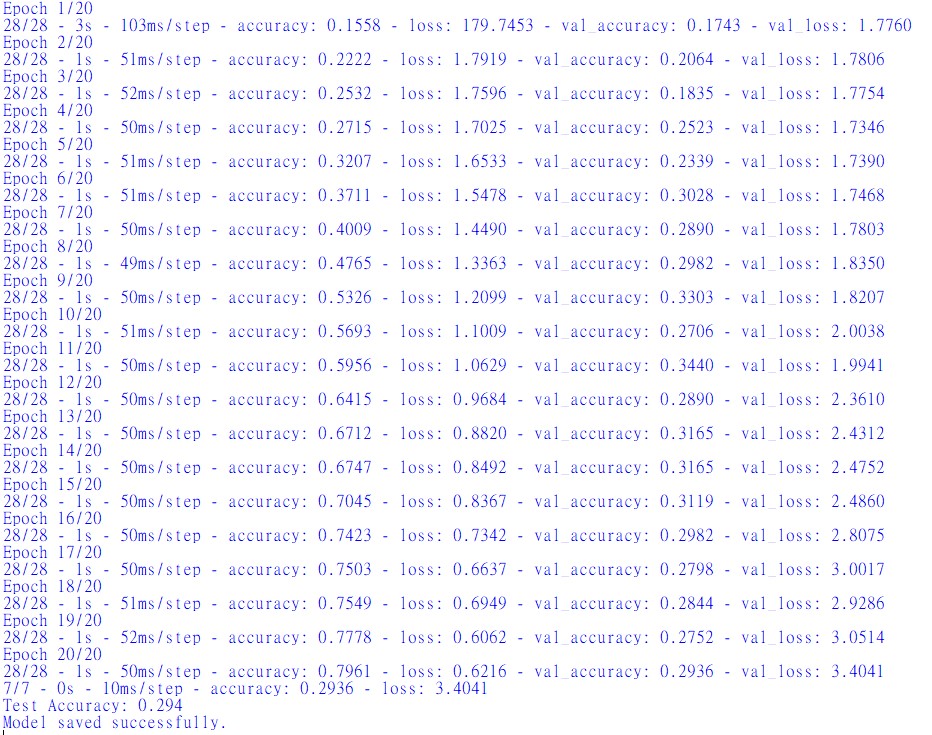
1. **Dense Layers:**

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.





**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.

1. **MaxPooling Layers (MaxPooling2D):**

Pooling layers with size (2, 2) to downsample spatial dimensions.

1. **Dropout Layers:**

Dropout (20% and 30%) to reduce overfitting by randomly deactivating neurons during training.

1. **Flatten Layer:**

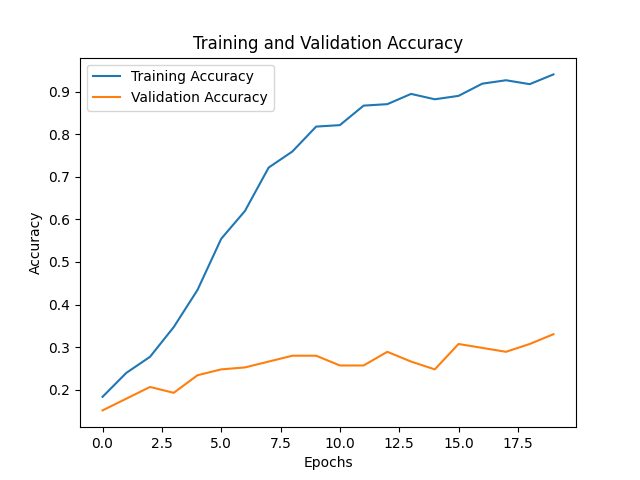
Converts the 2D feature maps into 1D vectors for the Dense layers.

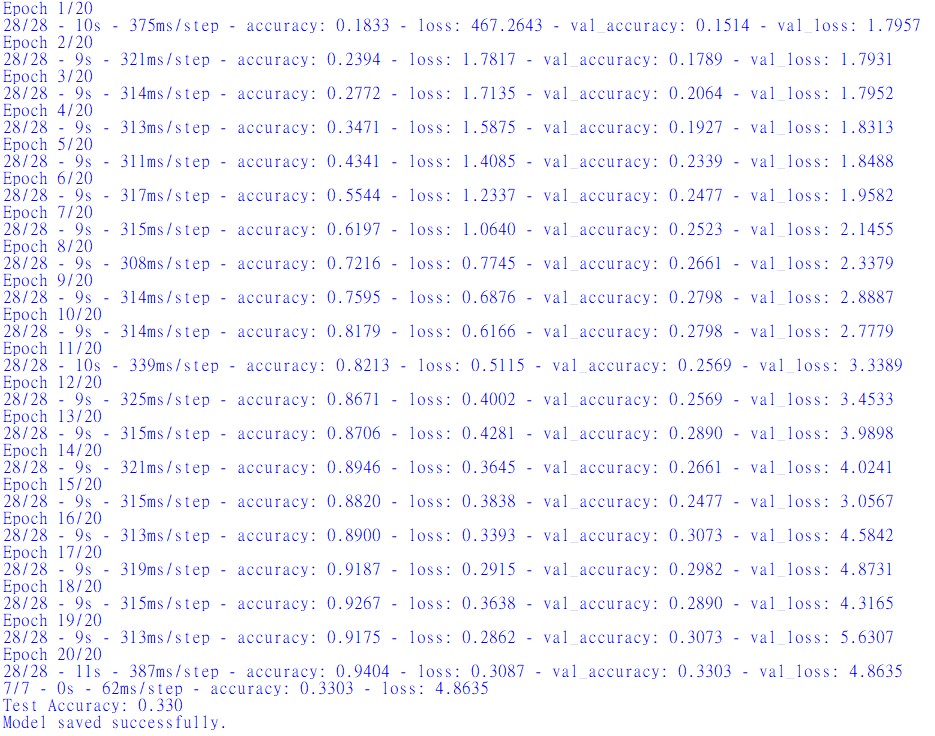
1. **Dense Layers:**

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.





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.

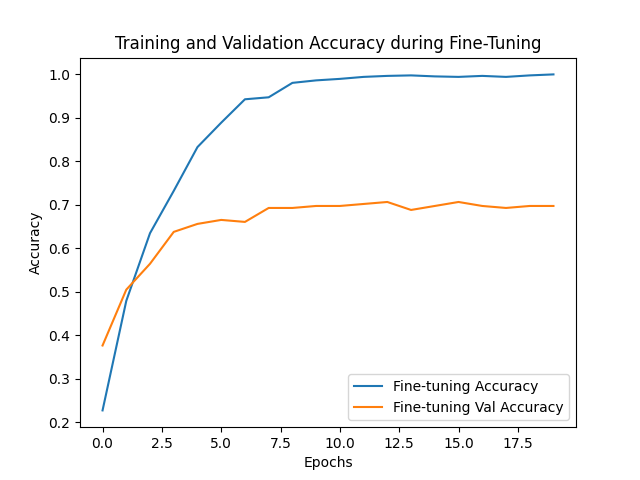
**Added Layers:**

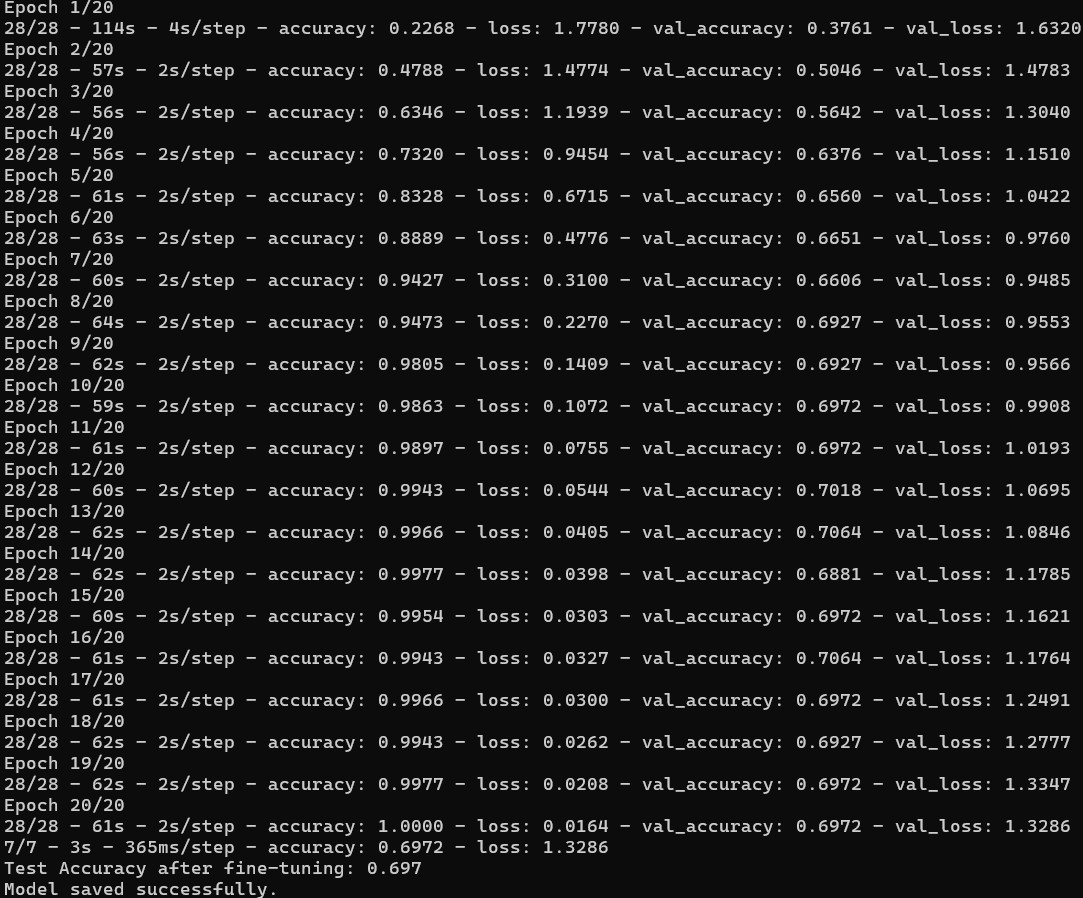
**GlobalAveragePooling2D:** To reduce spatial dimensions to a vector.

**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.





|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | 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



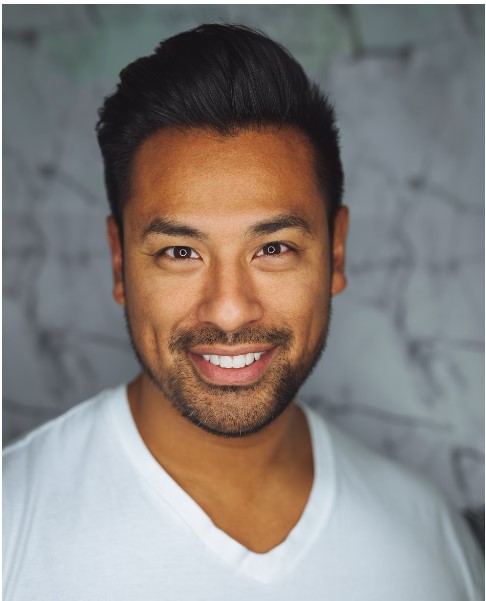
Pic 3 happy



Pic 4 happy



Pic 5 sad



Pic 6 happy



Pic 7 happy



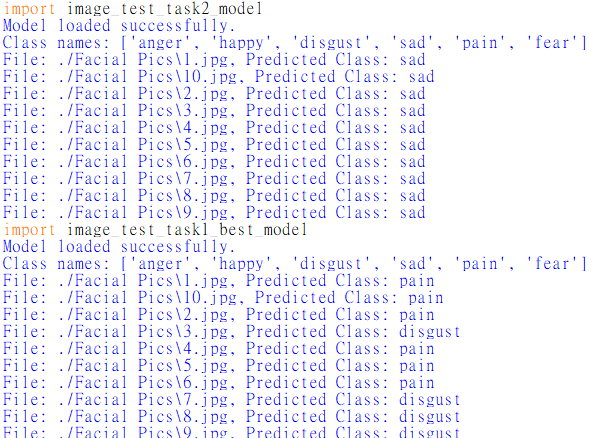
Pic 8 happy



Pic 9 sad



Pic 10 happy



#### ****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.