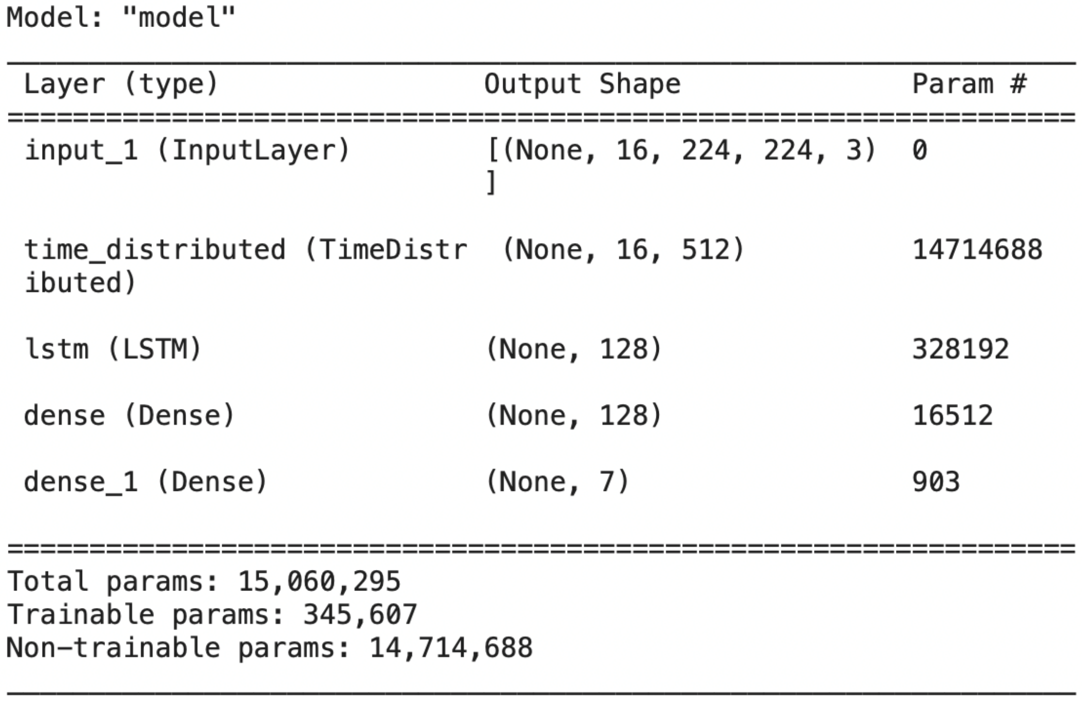
A description of the project：

The goal of this project is to build up a neural network that does classification of emotions on a given video of human facial expressions.
The training dataset of this project is huge, so we tried several different methods to optimize memory and batch processing.
We use feature fusion from two C3D (3D convolution) branches to predict video emotion. Also score fusion of one C3D and VGG16+LSTM.
Each branch is target-specific and efficient to catch facial features.

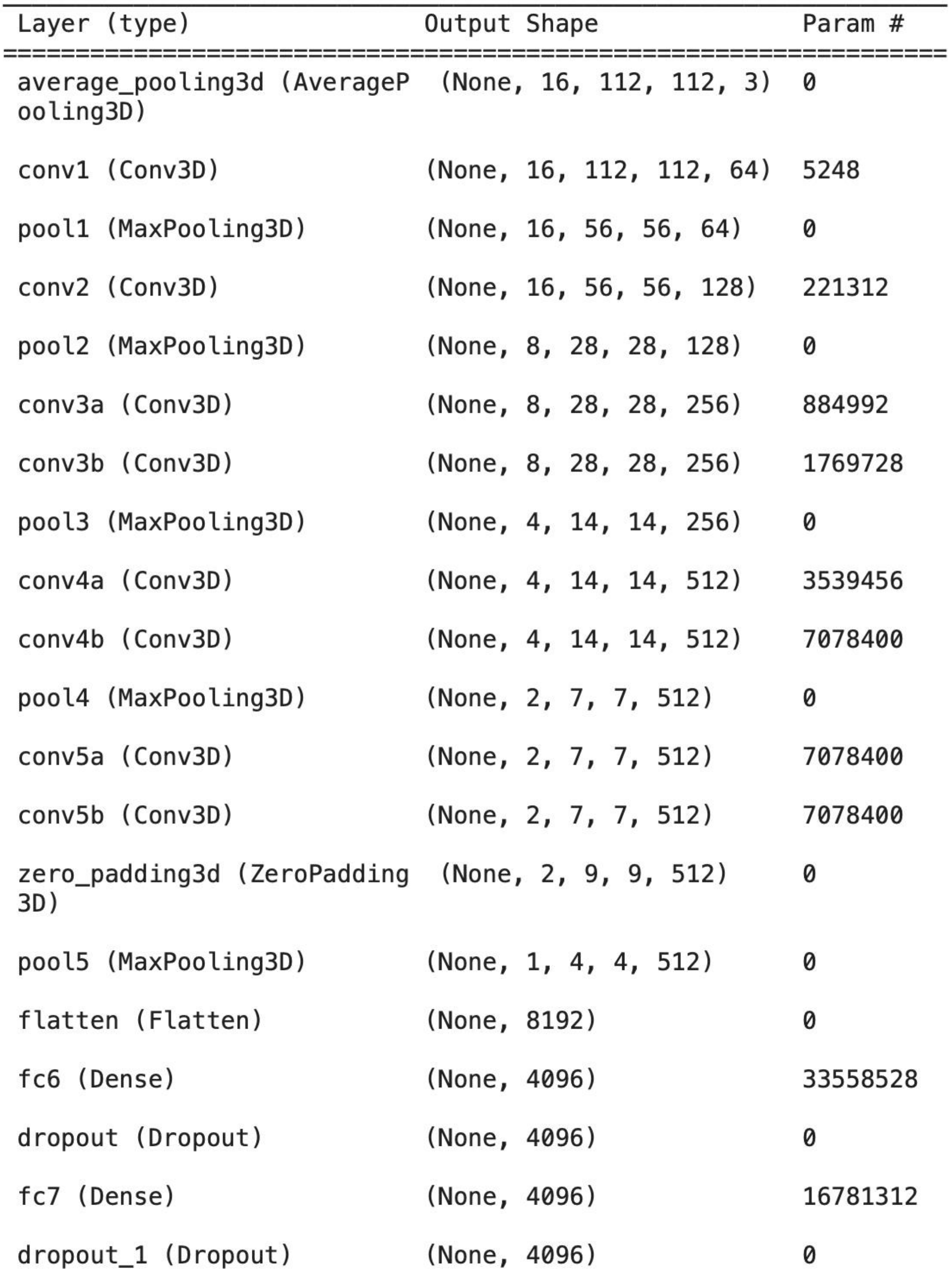
A description of the repository：
C3Dmodel.ipynb: model fusion of two C3D to do classification of emotions on a given video
c3dmodel.py: C3Dmodel.ipynb corresponding python file
d1.pkl: training and testing dataset(This file is too large to be uploaded to github)
labels.xlsx: training and testing labels
score_fusion.py: score fusion of 1 C3D model + (LSTM+vgg)
single_c3dmodel.py: helper function for score_fusion.py
vgg+lstm.py: helper function for score_fusion.py

Example commands to execute the code:
We run our code in google colab. Load C3Dmodel.ipynb directly to colab and run could train the model
Running cmd 'python ' + ‘file_names’ directly on other files.

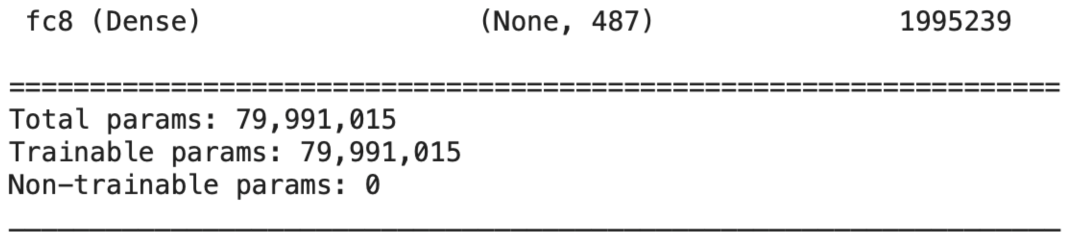
Results (including charts/tables) and your observations:
Score fusion:



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 16, 224, 224, 3)  0
                             ]

 time_distributed (TimeDistr  (None, 16, 512)          14714688
 ibuted)

 lstm (LSTM)                 (None, 128)               328192

 dense (Dense)               (None, 128)               16512

 dense_1 (Dense)             (None, 7)                 903

=================================================================
Total params: 15,060,295
Trainable params: 345,607
Non-trainable params: 14,714,688
_________________________________________________________________
```

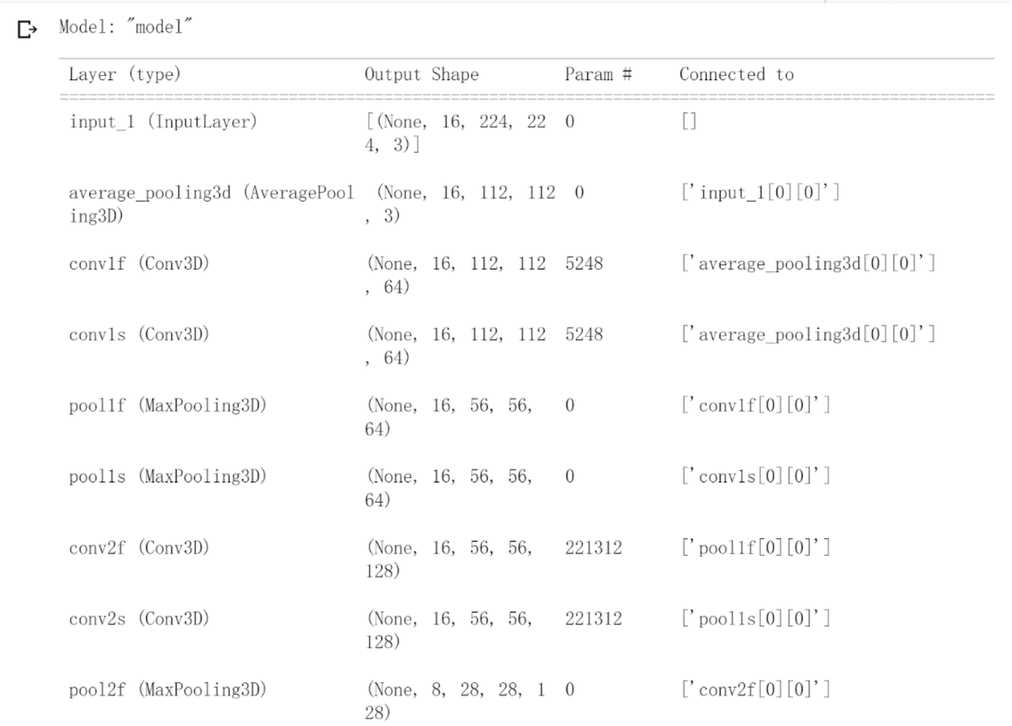


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 average_pooling3d (AverageP  (None, 16, 112, 112, 3)  0         
 ooling3D)                                                       
                                                                 
 conv1 (Conv3D)              (None, 16, 112, 112, 64)  5248      
                                                                 
 pool1 (MaxPooling3D)        (None, 16, 56, 56, 64)    0         
                                                                 
 conv2 (Conv3D)              (None, 16, 56, 56, 128)   221312    
                                                                 
 pool2 (MaxPooling3D)        (None, 8, 28, 28, 128)    0         
                                                                 
 conv3a (Conv3D)             (None, 8, 28, 28, 256)    884992    
                                                                 
 conv3b (Conv3D)             (None, 8, 28, 28, 256)    1769728   
                                                                 
 pool3 (MaxPooling3D)        (None, 4, 14, 14, 256)    0         
                                                                 
 conv4a (Conv3D)             (None, 4, 14, 14, 512)    3539456   
                                                                 
 conv4b (Conv3D)             (None, 4, 14, 14, 512)    7078400   
                                                                 
 pool4 (MaxPooling3D)        (None, 2, 7, 7, 512)      0         
                                                                 
 conv5a (Conv3D)             (None, 2, 7, 7, 512)      7078400   
                                                                 
 conv5b (Conv3D)             (None, 2, 7, 7, 512)      7078400   
                                                                 
 zero_padding3d (ZeroPadding  (None, 2, 9, 9, 512)     0         
 3D)                                                             
                                                                 
 pool5 (MaxPooling3D)        (None, 1, 4, 4, 512)      0         
                                                                 
 flatten (Flatten)           (None, 8192)              0         
                                                                 
 fc6 (Dense)                 (None, 4096)              33558528  
                                                                 
 dropout (Dropout)           (None, 4096)              0         
                                                                 
 fc7 (Dense)                 (None, 4096)              16781312  
                                                                 
 dropout_1 (Dropout)         (None, 4096)              0         
```



```
 fc8 (Dense)                 (None, 487)               1995239

=================================================================
Total params: 79,991,015
Trainable params: 79,991,015
Non-trainable params: 0
_________________________________________________________________
```

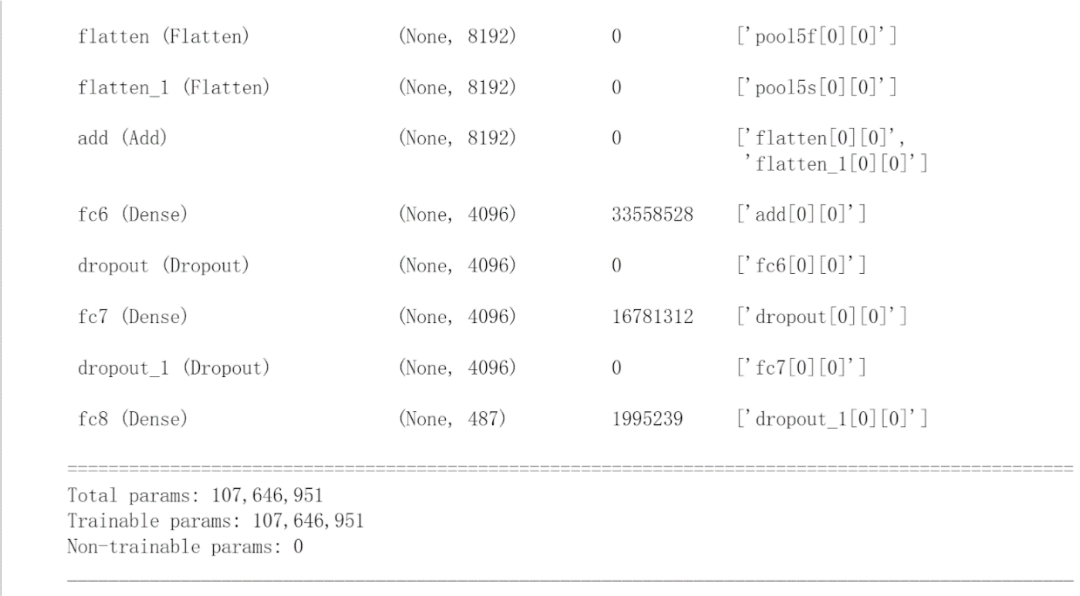
Model fusion



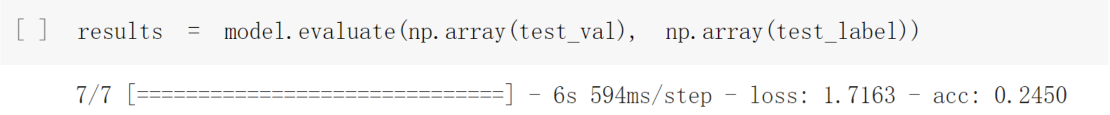
```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 16, 224, 22  0           []
                                4, 3)]

 average_pooling3d (AveragePool  (None, 16, 112, 112  0          ['input_1[0][0]']
 ing3D)                         , 3)

 conv1f (Conv3D)                (None, 16, 112, 112  5248        ['average_pooling3d[0][0]']
                                , 64)

 conv1s (Conv3D)                (None, 16, 112, 112  5248        ['average_pooling3d[0][0]']
                                , 64)

 pool1f (MaxPooling3D)          (None, 16, 56, 56,   0           ['conv1f[0][0]']
                                64)

 pool1s (MaxPooling3D)          (None, 16, 56, 56,   0           ['conv1s[0][0]']
                                64)

 conv2f (Conv3D)                (None, 16, 56, 56,   221312      ['pool1f[0][0]']
                                128)

 conv2s (Conv3D)                (None, 16, 56, 56,   221312      ['pool1s[0][0]']
                                128)

 pool2f (MaxPooling3D)          (None, 8, 28, 28, 1  0           ['conv2f[0][0]']
                                28)
```

```
pool2s (MaxPooling3D)          (None, 8, 28, 28, 1  0          [’conv2s[0][0]’]
                               28)

conv3af (Conv3D)               (None, 8, 28, 28, 2  884992     [’pool2f[0][0]’]
                               56)

conv3as (Conv3D)               (None, 8, 28, 28, 2  884992     [’pool2s[0][0]’]
                               56)

conv3bf (Conv3D)               (None, 8, 28, 28, 2  1769728    [’conv3af[0][0]’]
                               56)

conv3bs (Conv3D)               (None, 8, 28, 28, 2  1769728    [’conv3as[0][0]’]
                               56)

pool3f (MaxPooling3D)          (None, 4, 14, 14, 2  0          [’conv3bf[0][0]’]
                               56)

pool3s (MaxPooling3D)          (None, 4, 14, 14, 2  0          [’conv3bs[0][0]’]
                               56)

conv4af (Conv3D)               (None, 4, 14, 14, 5  3539456    [’pool3f[0][0]’]
                               12)

conv4as (Conv3D)               (None, 4, 14, 14, 5  3539456    [’pool3s[0][0]’]
                               12)

conv4bf (Conv3D)               (None, 4, 14, 14, 5  7078400    [’conv4af[0][0]’]
                               12)
```

```
conv4bs (Conv3D)               (None, 4, 14, 14, 5  7078400     ['conv4as[0][0]']
                               12)

pool4f (MaxPooling3D)          (None, 2, 7, 7, 512  0           ['conv4bf[0][0]']
                               )

pool4s (MaxPooling3D)          (None, 2, 7, 7, 512  0           ['conv4bs[0][0]']
                               )

conv5af (Conv3D)               (None, 2, 7, 7, 512  7078400     ['pool4f[0][0]']
                               )

conv5as (Conv3D)               (None, 2, 7, 7, 512  7078400     ['pool4s[0][0]']
                               )

conv5bf (Conv3D)               (None, 2, 7, 7, 512  7078400     ['conv5af[0][0]']
                               )

conv5bs (Conv3D)               (None, 2, 7, 7, 512  7078400     ['conv5as[0][0]']
                               )

zero_padding3d (ZeroPadding3D)  (None, 2, 9, 9, 512  0          ['conv5bf[0][0]']
                               )

zero_padding3d_1 (ZeroPadding3  (None, 2, 9, 9, 512  0          ['conv5bs[0][0]']
D)                             )

pool5f (MaxPooling3D)          (None, 1, 4, 4, 512  0           ['zero_padding3d[0][0]']
                               )

pool5s (MaxPooling3D)          (None, 1, 4, 4, 512  0           ['zero_padding3d_1[0][0]']
                               )
```

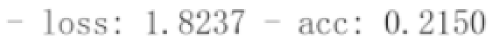


```
 flatten (Flatten)              (None, 8192)         0           ['pool5f[0][0]']

 flatten_1 (Flatten)            (None, 8192)         0           ['pool5s[0][0]']

 add (Add)                      (None, 8192)         0           ['flatten[0][0]',
                                                                  'flatten_1[0][0]']

 fc6 (Dense)                    (None, 4096)         33558528    ['add[0][0]']

 dropout (Dropout)              (None, 4096)         0           ['fc6[0][0]']

 fc7 (Dense)                    (None, 4096)         16781312    ['dropout[0][0]']

 dropout_1 (Dropout)            (None, 4096)         0           ['fc7[0][0]']

 fc8 (Dense)                    (None, 487)          1995239     ['dropout_1[0][0]']

==================================================================================================
Total params: 107,646,951
Trainable params: 107,646,951
Non-trainable params: 0
__________________________________________________________________________________________________
```



```
[ ]  results = model.evaluate(np.array(test_val), np.array(test_label))

     7/7 [==============================] - 6s 594ms/step - loss: 1.7163 - acc: 0.2450
```

Testing accuracy on double C3D model: 24.5% accuracy



```
- loss: 1.8237 - acc: 0.2150
```

Testing accuracy on one C3D model: 21.5%

Observation:

Using feature fusion does not harm the accuracy much, but saves nearly 40% parameters compared to score fusion. Double C3D model has a little improvement in terms of accuracy compared with single C3D model. The training of both networks does not perform very well on video emotion classification problem, only achieving 25 percentage accuracy. This may due to lack of training data since we only train on 1800 videos. If more data is available, accuracy could be improved.