✓ **Congratulations! You passed!**

TO PASS 80% or higher

Keep Learning

Retake the assignment in **7h 57m**

GRADE
100%

Graded Quiz: Test your Project Understanding

LATEST SUBMISSION GRADE

100%

1. In the Facial Expression Recognition project that we just completed, we used the Keras API with TensorFlow as its backend. True or False?

1 / 1 point

● True

○ False

✓ **Correct**
Correct! While Keras can use Theano or CNTK as backend as well, we used the TensorFlow implementation of the Keras API.

2. Which of following is true about the FER 2013 dataset of images we used to train the CNN model? (Select all that apply)

1 / 1 point

□ The dataset contains RGB images of faces

☑ The dataset contains grayscale images of faces

✓ **Correct**
Correct! There is no channel information in the images, making them grayscale.

☑ Each image has 48 rows and 48 columns

✓ **Correct**
Correct! The data contains grayscale 48x48 images of faces.

☑ The dataset contains seven categories of facial expressions

✓ **Correct**
Correct! The objective is to classify each face based on the emotion shown in the facial expression into one of seven categories:

- 0=Angry
- 1=Disgust,
- 2=Fear,
- 3=Happy,
- 4=Sad,
- 5=Surprise,
- 6=Neutral

3. The Facial Expression Recognition model we created is comprised of 6 dense, fully connected layers. True or False?

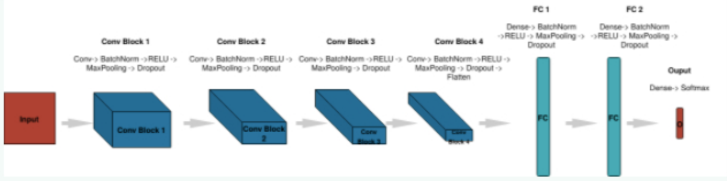
1 / 1 point

○ True

● False

✓ **Correct**

Correct! The model we created is comprised of 4 convolution blocks followed by 2 fully connected layers. Please refer to the model architecture block diagram:


Conv Block 1
Conv-> BatchNorm ->RELU -> MaxPooling -> Dropout
Conv Block 2
Conv-> BatchNorm ->RELU -> MaxPooling -> Dropout
Conv Block 3
Conv-> BatchNorm ->RELU -> MaxPooling -> Dropout
Conv Block 4
Conv-> BatchNorm ->RELU -> MaxPooling -> Dropout -> Flatten
FC 1
Dense-> BatchNorm ->RELU -> MaxPooling -> Dropout
FC 2
Dense-> BatchNorm ->RELU -> MaxPooling -> Dropout
Ouput
Dense-> Softmax
Input
Conv Block 1
Conv Block 2
Conv Block 3
Conv Block 4
FC
FC
O

4. What technology, API, or framework did we use for face detection?

1 / 1 point

○ Keras

● OpenCV

○ Flask

✓ **Correct**

Correct! We did not re-invent the wheel by implementing our own face detection module. Instead, we used one of OpenCV's pre-trained XML classifier for faces.

5. What are the two types of regularization techniques that you used in this project? (Select all that apply)

1 / 1 point

□ Weight regularization

☑ Dropout regularization

✓ **Correct**

Correct! We used dropout regularization in Task 4 of this hands-on project.

☑ Batch normalization

✓ **Correct**

Correct! We used batch normalization in Task 4 of this hands-on project.

6. Select all statements that are true:

1 / 1 point

☑ In order to get a predicted class, we used the **np.argmax()** function on the predictions. This is because the predictions are 7 dimensional vectors and we want to use the class with the highest probability as our final prediction.

✓ **Correct**

□ The **argmax** function displays the image along with its prediction.

☑ The probability scores of a prediction sum up to a total of 1.0

✓ **Correct**

Correct. The probability scores for the 7 classes will sum up to 1 and the class with the highest probability score is used as the final predicted class.

7. What does the following code block accomplish?

1 / 1 point

```
model_json = model.to_json()
with open("model.json", "w") as json_file:
    json_file.write(model_json)
```

○ The code saves the trained model's weights to disk as a JSON file.

● The code uses a JSON string to store the "config" or model architecture.

✓ **Correct**

Correct! We were interested in the architecture of the model, and did not need to save the weight values or the optimizer, since that was taken care of during training in Task 5.

8. In the hands-on project, which statement(s) would apply to the **softmax** activation function? (Select all that apply)

1 / 1 point

☑ It gives us probability scores for the classes.

✓ **Correct**

Correct, the **softmax** activation gives us probability scores for all the classes which sum up to a total of 1. The class with the highest probability score is then used as our final prediction.

☑ It can be used as an activation function for the output layer in classification problems.

✓ **Correct**

Correct! Since this activation function gives us probability scores for all the classes, it is suitable to be used as an output activation function for classification problems.

□ It gives us linear output for input values higher than 0 and for input values less than 0, the output is set to 0.

9. Which function returns the history object which has training metrics like training accuracy and loss?

1 / 1 point

● model.fit()

○ model.compile()

✓ **Correct**

Correct!

10. In the hands-on project, the trained model recognized facial expressions in videos saved to disk. Is it true that you need to change just one line of code to classify facial expressions in real-time from your webcam stream?

1 / 1 point

● True

○ False

✓ **Correct**

Correct! You need to edit **camera.py** and change the argument of the **cv2.VideoCapture()** function from the path of a video file to **0**. Changing it to **0** means OpenCV will use your default video input source. Namely, your webcam, if you have one.