

## APPENDIX A USER STUDY DESIGN

### A. Tasks

#### 1) Task 1:

```
[ ] '''
Change the shape of a tensor without changing the value.
input = tf.Tensor([[[ 7  9  1  6]
                    [[12  4  3  3]
                    [[11 11 12  0]
                    [[11  1  0  8]]], shape=(4, 1, 4), dtype=int32)
output = tf.Tensor([[[ 7  9  1  6]
                    [12  4  3  3]
                    [11 11 12  0]
                    [11  1  0  8]]], shape=(4, 4), dtype=int32)
'''
```

#### 2) Task 2:

```
[ ] '''
Compute matrix multiplication of
three tensors input1, input2, and input3.
input1: tf.Tensor([[6]
                  [6]], shape=(2, 1), dtype=int32)
input2: tf.Tensor([[6 5]], shape=(1, 2), dtype=int32)
input3: tf.Tensor([[9]
                  [8]], shape=(2, 1), dtype=int32)
output: tf.Tensor([[564]
                  [564]], shape=(2, 1), dtype=int32)
'''
```

#### 3) Task 3:

```
[ ] '''
Merge three tensors having the same first dimension,
across the second dimension.

input1: tf.Tensor([[7]
                  [9]], shape=(2, 1), dtype=int32)
input2: tf.Tensor([[18  3]
                  [19 14]], shape=(2, 2), dtype=int32)
input3: tf.Tensor([[ 4 12 17]
                  [ 4  3  7]], shape=(2, 3), dtype=int32)
output: tf.Tensor([[ 7 18  3  4 12 17]
                  [ 9 19 14  4  3  7]], shape=(2, 6), dtype=int32)
'''
```

4) Task 4:

```
[ ] '''
Select rows from tensor input1
if the corresponding value in tensor input2 is True.

input1: tf.Tensor([[ 7  9]
                   [ 1  6]
                   [12  4]], shape=(3, 2), dtype=int32)
input2: tf.Tensor([False  True  True], shape=(3,), dtype=bool)
output: tf.Tensor([[ 1  6]
                   [12  4]], shape=(2, 2), dtype=int32)
'''
```

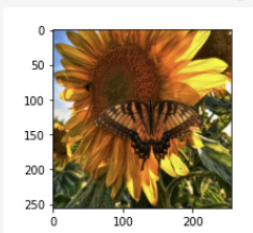
5) Task 5:

```
[ ] '''
Divide tensor input1 with tensor input2 and
only keep the integer part of the output.

input1: tf.Tensor([[7 9]
                   [1 6]], shape=(2, 2), dtype=int32)
input2: tf.Tensor([[3 3]
                   [4 4]], shape=(2, 2), dtype=int32)
output: tf.Tensor([[2 3]
                   [0 1]], shape=(2, 2), dtype=int32)
'''
```

6) Task 6:

```
[ ] '''
Change the shape of the image from [256, 256] into [200, 500] without distortion.
input: a Tensorflow Tensor of an RGB image with shape [256, 256]
```



output: a Tensorflow Tensor of the same image with shape [200, 500]



```
'''
```

7) Task 7:

```
[ ] '''
Compute a convolution given an input tensor input1,
and a kernel tensor input2.

input1 (input): tf.Tensor([[[1.]
                        [0.]
                        [2.]
                        [3.]
                        [0.]]], shape=(1, 5, 1), dtype=float32)

input2 (kernel): tf.Tensor([[[2.]
                        [[1.]]
                        [[3.]]], shape=(3, 1, 1), dtype=float32)

output: tf.Tensor([[[ 8.]
                    [11.]
                    [ 7.]]], shape=(1, 3, 1), dtype=float32)
...
'''
```

8) Task 8:

```
[ ] '''
Compute softmax cross entropy loss between the input1 and input2
of multi-class classification model.

input1 (target): tf.Tensor([2 0 1], shape=(3,), dtype=int32)
input2 (logits): tf.Tensor([[0.1 0.  1. ]
                        [5.  0.8 1. ]
                        [1.9 2.1 0.7]],
                        shape=(3, 3), dtype=float32)
output: tf.Tensor([1.2939507  0.04070133 0.87471807],
                  shape=(3,), dtype=float32)
...
'''
```

## B. User Study Task Assignments

Participants	Task IDs			
	(Treatment cond. in <b>bold</b> )			
	1st	2nd	3rd	4th
P1	2	<b>6</b>	1	<b>5</b>
P2	<b>6</b>	<b>7</b>	5	8
P3	5	<b>2</b>	6	<b>1</b>
P4	<b>4</b>	<b>8</b>	3	7
P5	<b>7</b>	6	<b>2</b>	4
P6	4	1	<b>6</b>	<b>8</b>
P7	6	2	<b>4</b>	<b>7</b>
P8	<b>1</b>	4	2	<b>3</b>
P9	<b>3</b>	8	<b>5</b>	1
P10	3	<b>1</b>	4	<b>2</b>
P11	8	<b>4</b>	<b>1</b>	6
P12	<b>5</b>	7	<b>3</b>	2
P13	<b>2</b>	5	<b>7</b>	3
P14	1	<b>3</b>	<b>8</b>	5
P15	7	<b>5</b>	8	<b>6</b>
P16	<b>8</b>	3	7	<b>4</b>