

100%

✓ Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE 100%

1/1 point

Style Transfer

LATEST SUBMISSION GRADE

1.	In Neural Style Transfer when initializing the <i>generated image</i> from the <i>content image</i> , which of the following is true? Check all that apply.	1/1 point
	Initially the <i>style loss</i> will be equal or close to zero because both, the <i>content</i> and <i>generated</i> , images are the same.	
	Your goal for the <i>generated image</i> is to increase the <i>content loss</i> and decrease the <i>style loss</i> , while keeping the overall <i>accumulated loss</i> low.	
	Correct Correct! Since the generated image is initialized from the content image, you want it to inherit attributes from the style image (reduce style loss), but also not lose all of its attributes inherited from the content image (increase content loss).	
	Your goal for the <i>generated image</i> is to increase the <i>style loss</i> and decrease the <i>content loss</i> while keeping the overall <i>accumulated loss</i> low.	
	Initially the <i>content loss</i> will be equal or close to zero because both the <i>content image</i> and <i>generated</i> image are the same image.	
	✓ Correct Correct!	
2.	What does tf.keras.applications.vgg19.preprocess_input do?	1 / 1 point
	The function sets the pixel values of an image between 0 and 1.	
	The function centers the distribution of pixel values of an image around zero.	
	✓ CorrectCorrect! This is called standardization.	
3.	From which part of a CNN architecture can you extract the "content" of an image?	1 / 1 point
	The initial layers of the architecture.	
	From the deeper layers of the architecture.	
	Correct Correct! If you recall the lecture we used only the deeper layer of the CNN for computing <i>content loss</i> because that layer holds the information of the <i>content</i> of an image.	

Generated image

Content image

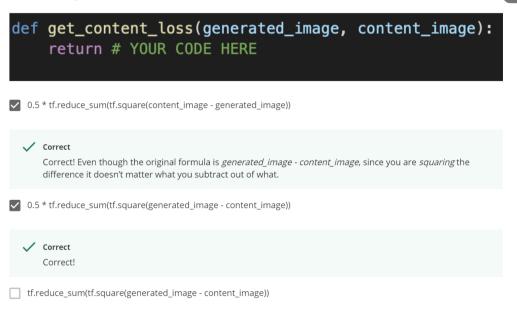
4. Consider the values given in the image below and calculate the *content loss* value.



✓ Correct
Correct!

5. Fill in the missing code below:

1 / 1 point



6. Consider the following code snippet. How will you include *Total Loss Variation* in it? Use TensorFlow as *tf.*

1/1 point

(Answer in the format, $\mathbf{x} + \mathbf{y}(\mathbf{z})$, considering python's spacing convention)

tf.reduce_sum(tf.square(content_image - generated_image))

total_variation_weight * tf.image.total_variation(image)

✓ Correct!