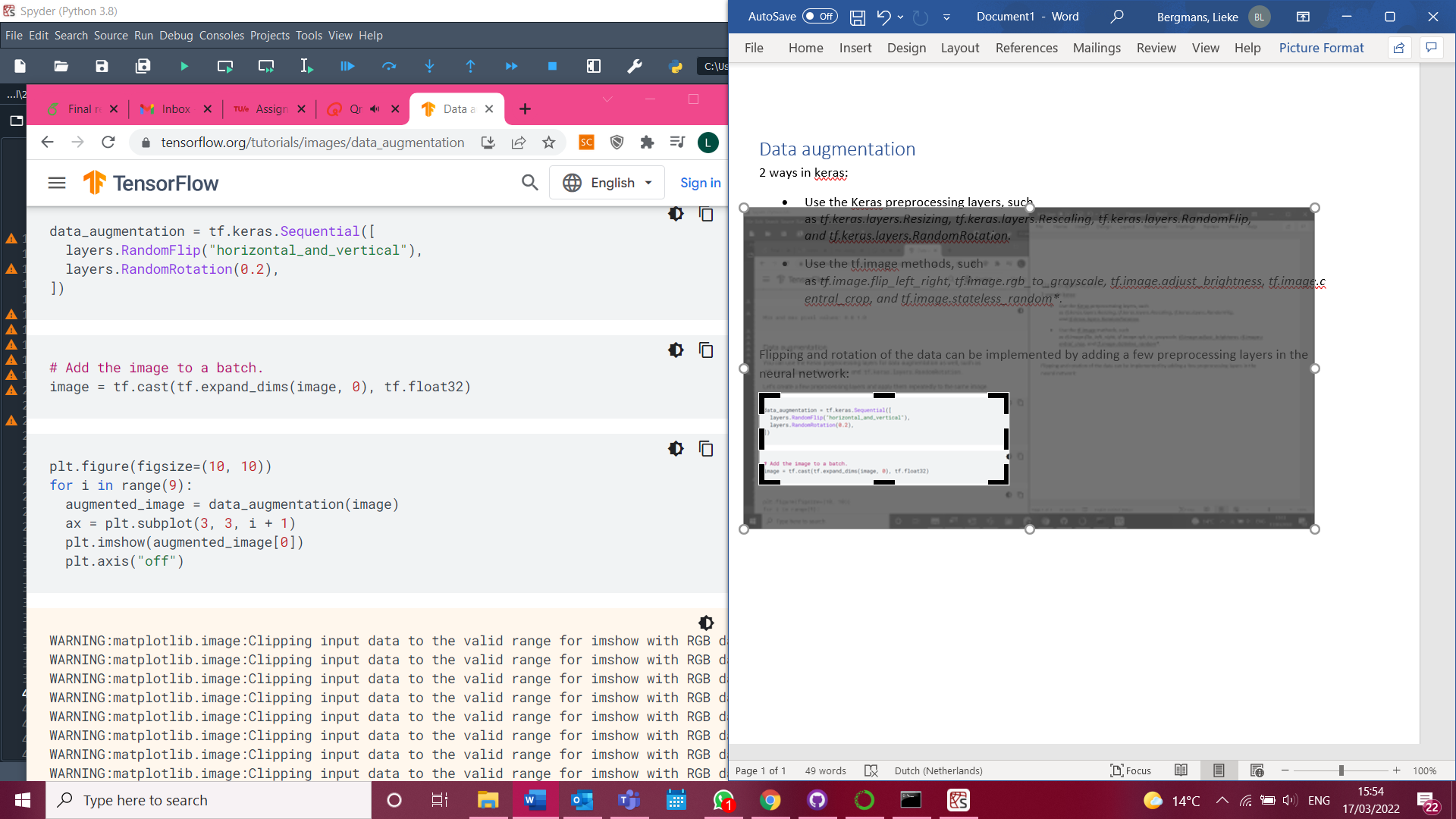
# Data augmentation

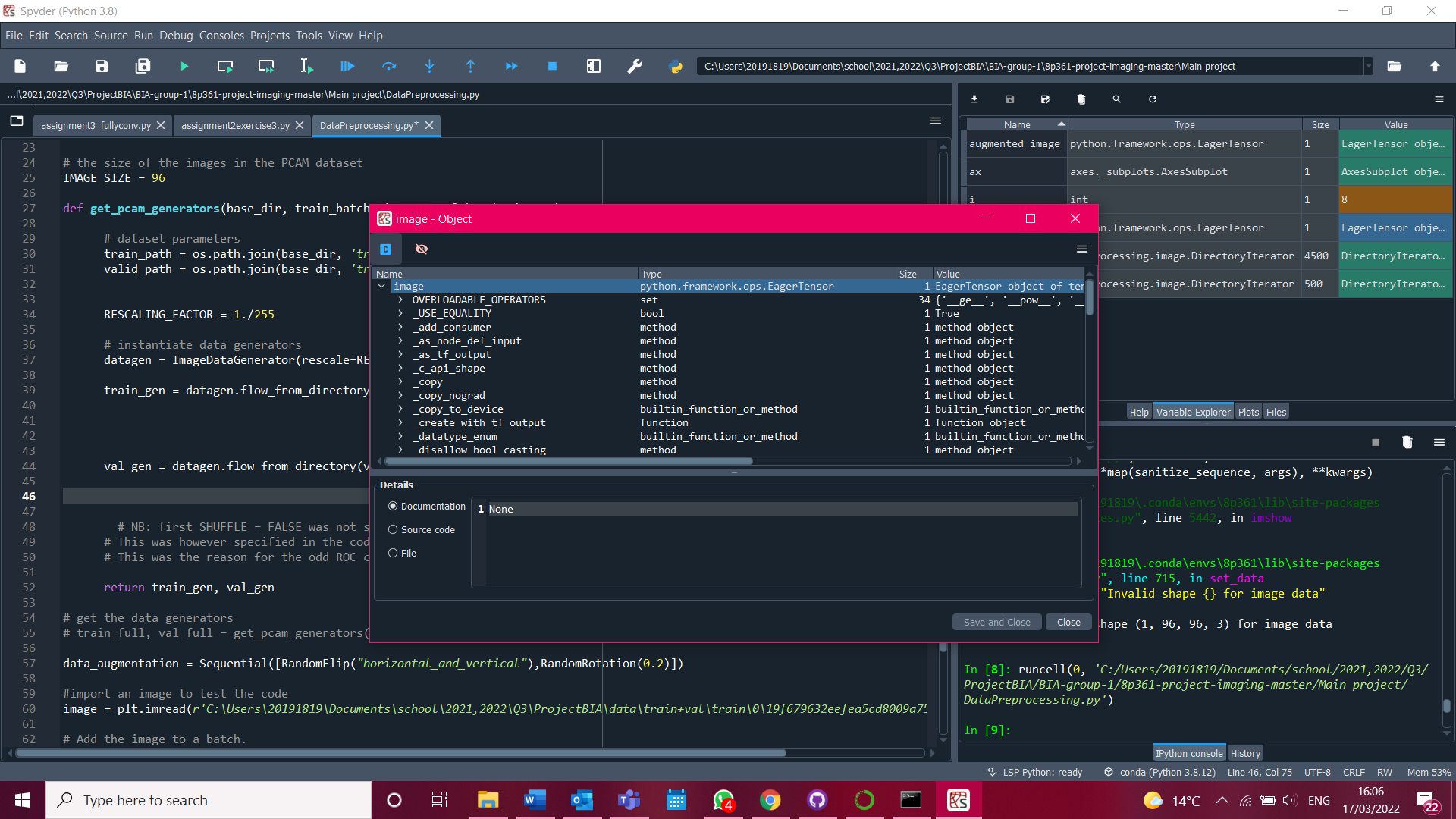
2 ways in keras:

* Use the Keras preprocessing layers, such as [*tf.keras.layers.Resizing*](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Resizing)*,*[*tf.keras.layers.Rescaling*](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Rescaling)*,*[*tf.keras.layers.RandomFlip*](https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomFlip)*, and*[*tf.keras.layers.RandomRotation*](https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomRotation)*.*
* Use the [tf.image](https://www.tensorflow.org/api_docs/python/tf/image) methods, such as [*tf.image.flip\_left\_right*](https://www.tensorflow.org/api_docs/python/tf/image/flip_left_right)*,*[*tf.image.rgb\_to\_grayscale*](https://www.tensorflow.org/api_docs/python/tf/image/rgb_to_grayscale)*,*[*tf.image.adjust\_brightness*](https://www.tensorflow.org/api_docs/python/tf/image/adjust_brightness)*, [tf.image.central\_crop](https://www.tensorflow.org/api_docs/python/tf/image/central_crop), and tf.image.stateless\_random\*.*

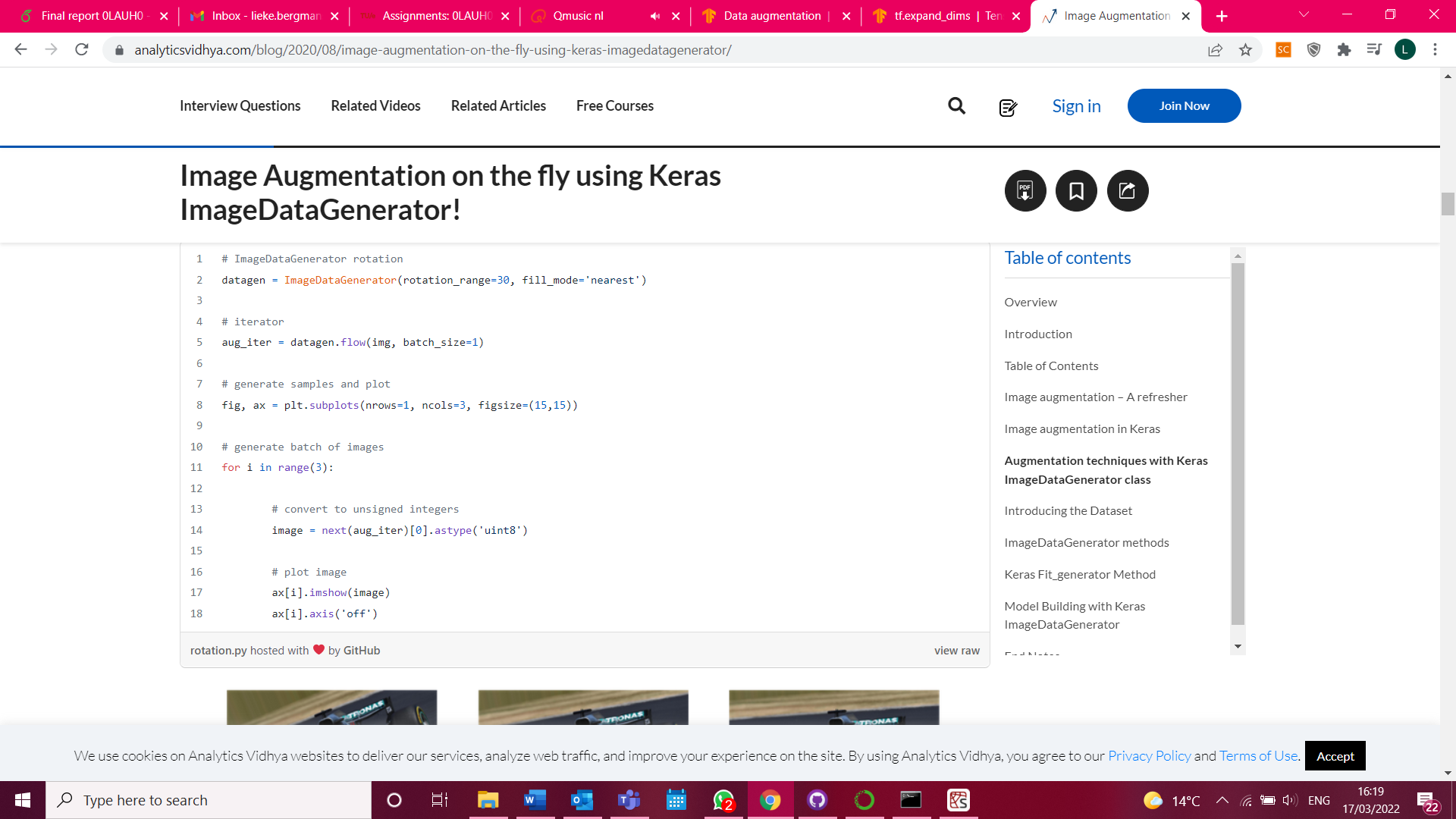
Flipping and rotation of the data can be implemented by adding a few preprocessing layers in the neural network:



This seems simple, however, our data seems to be somewhat complex with a lot of additional information:



Omdat de methode hierboven lastig bleek te zijn, heb ik gekeken naar methodes om in de ImageDataGenerator rotatie etc. toe te passen zodat de aanpak van de data hetzelfde blijft als in de assignments:



<https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>