

Training the model

This step can be done with only one line of code. The `.fit` method of the `Sequential` class has several parameters that we must take a look at. First, we can pass in a value to `x`, the parameter that takes in the training dataset of images. We have this stored in the variable `train`.

The next argument we must take a look at is `validation_data`, which will be used to determine the accuracy of the classifier on separate test data. We have this data stored in the variable `test`.

The final parameter that we are going to take a look at is `epochs`, which is the number of dataset iterations our CNN will train for. For this, we are going to input 100, as this number of epochs ensures that our classifier will be well-trained without overfitting (I determined this by testing several different numbers, but you can feel free to experiment yourself as well).

Great! Since we now know the variables that we will input, let's implement this into code.

```
cnn.fit(x=train, validation_data=test, epochs=100, batch_size=32)
```

If we run our program now, we should receive a validation accuracy (denoted by `val_accuracy`) of around 98 percent.

The final thing that we're going to do is save our model so that we can use it to classify single images.

Fortunately, this can be done with `Sequential`'s `.save` method, which will store the model in its own directory. Let's go ahead and put this into our code.

```
cnn.save('model', save_format='tf')
```

Testing on Single Images

Now that we're done training and testing our neural network, let's create another file called test.py in which we create and define a function that will use the neural network to classify single input images as either a wildfire or not a wildfire.

Let's start by importing all the classes we will need for this section.

```
from tensorflow.keras.models import load_model
```

```
import numpy as np
```

```
from tensorflow.keras.preprocessing.image import load_img
```

```
from tensorflow.keras.preprocessing.image import img_to_array
```

Now, let's figure out which images we will select from the test dataset to test our network performance on.

Here are the images that I chose:

