

## Save the model

We're just going to do one final thing before we conclude this project. We're going to create yet another file which will use the `predict()` function we created to classify an image that the computer camera takes.

Let's start by importing all the libraries we will be using:

```
from cv2 import cv2
import time
from test import predict
```

Now, we will instantiate the `VideoCapture` class by inputting the camera we want to use (for most people with built-in cameras, this number would be 0).

```
cam = VideoCapture(0)
```

Now, we're going to create an infinite while loop since we want our program to take pictures at specified time intervals. Inside this loop, we're going to read the current frame of the video and ensure that the frame was properly captured. This can be done as shown below.

```
while True:
    is_error, img = cam.read()
    if not is_error:
        print('Error when grabbing frame')
        break
```

The `.read` method returns a list of which the first index contains a boolean; `True` signifies that the frame was properly captured, and `False` indicates that it was not. The second index of the list contains the frame itself.

If there is an error, our program will tell the user that there was an "Error when grabbing frame" and break out of the loop.

Now that we've completed this portion, let's write the image to a file by using OpenCV's `imwrite()` function.

```
while True:
```

```
    is_error, img = cam.read()
    if not is_error:
        print('Error when grabbing frame')
        break
    cv2.imwrite('image.png', img)
```

Finally, we'll call the `predict()` function we created and assign its return value to `prediction`. We'll then print this prediction and wait for 10 seconds before starting the next iteration of the loop.

```
while True:
```

```
    is_error, img = cam.read()
    if not is_error:
        print('Error when grabbing frame')
        break
```

```
    cv2.imwrite('image.png', img)
    prediction = predict('image.png')
    print(prediction)
    time.sleep(10)
```

Great! If we run our program now, we should see a series of 0's and 1's depending on whether our classifier categorized the picture taken as a wildfire or not a wildfire.

You may notice that the network classifies more images incorrectly when you take pictures around your home; this is because it was trained with pictures of nature and the natural environment. Thus, images of a home environment and/or any other non-natural environment will be more foreign to it.