```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import tensorflow as tf

import keras_preprocessing
from keras_preprocessing import image
from keras_preprocessing.image import ImageDataGenerator


```

```
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Input, Dropout

```

```
model = tf.keras.models.load_model("/content/drive/MyDrive/study/durhamcollege/GitInCloud/fir
model.save

```

```
<bound method Model.save of <keras.engine.functional.Functional object at 0x7f0595d78390>>
```

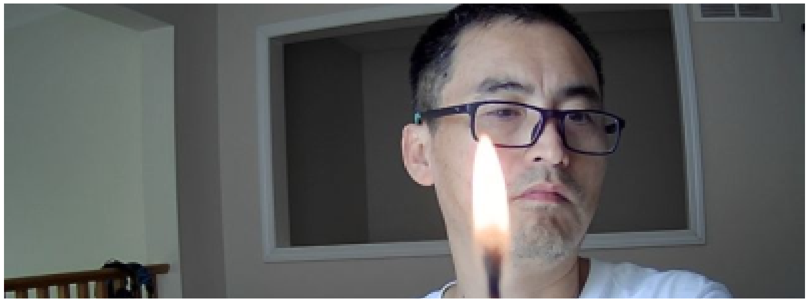
```
import imutils
import cv2
from google.colab.patches import cv2_imshow
from IPython.display import display, Javascript
from google.colab.output import eval_js
```

```
def take_photo(filename='photo.jpg', quality=0.8):
  js = Javascript('''
    async function takePhoto(quality) {
      const div = document.createElement('div');
      const capture = document.createElement('button');
      capture.textContent = 'Capture';
      div.appendChild(capture);

      const video = document.createElement('video');
      video.style.display = 'block';
      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);
      div.appendChild(video);
      video.srcObject = stream;
      await video.play();

      // Resize the output to fit the video element.
      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.
      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');
      canvas.width = video.videoWidth;
      canvas.height = video.videoHeight;
      canvas.getContext('2d').drawImage(video, 0, 0);
      stream.getVideoTracks()[0].stop();
      div.remove();
      return canvas.toDataURL('image/jpeg', quality);
    }
    ''')
  display(js)
  data = eval_js('takePhoto({})'.format(quality))
  binary = b64decode(data.split(',')[1])
  with open(filename, 'wb') as f:
    f.write(binary)
  return filename
```

```
image_file = take_photo()
```

```
#image = cv2.imread(image_file, cv2.IMREAD_UNCHANGED)
image = cv2.imread(image_file)

# resize it to have a maximum width of 400 pixels
image = imutils.resize(image, width=400)
img = tf.image.resize(image, (224,224))
(h, w) = image.shape[:2]
print(w,h)
cv2_imshow(image)


```

```
400 300
```



```
import numpy as np
x = tf.keras.utils.img_to_array(img)

x = np.expand_dims(x, axis=0) /255
classes = model.predict(x)
print(np.argmax(classes[0])==0, max(classes[0]))
```

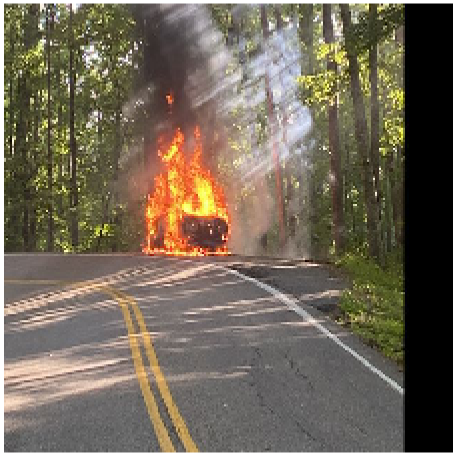
```
True 0.58499634
```

```
#predicting any random image
import numpy as np
from google.colab import files
from keras.preprocessing import image

uploaded = files.upload()
for fn in uploaded.keys():
  path = '/content/' + fn
  img = image.load_img(path, target_size=(224, 224))
  display(img)
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0) /255
  classes = model.predict(x)
  print(np.argmax(classes[0])==0, max(classes[0]))
```

Choose Files fireimage3.jpg

- **fireimage3.jpg**(image/jpeg) - 62776 bytes, last modified: 6/27/2022 - 100% done

```
Saving fireimage3.jpg to fireimage3 (1).jpg
```



```
True 0.8200393
```

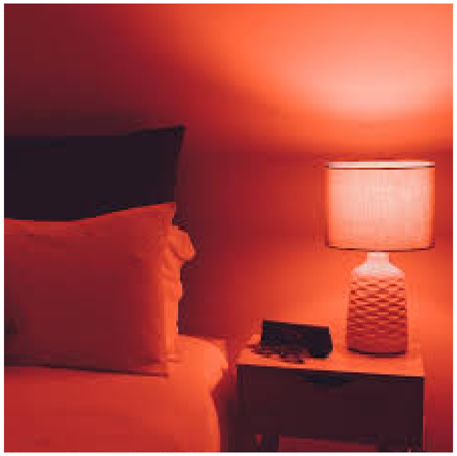
```
#predicting any random image
import numpy as np
from google.colab import files
from keras.preprocessing import image

uploaded = files.upload()
for fn in uploaded.keys():
  path = '/content/' + fn
  img = image.load_img(path, target_size=(224, 224))
  display(img)
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0) /255
  classes = model.predict(x)
  print(np.argmax(classes[0])==0, max(classes[0]))
```

```
    print(np.argmax(classes[0])==0, max(classes[0]))
```

Choose Files noffire2.jpg

- **noffire2.jpg**(image/jpeg) - 6175 bytes, last modified: 6/27/2022 - 100% done

```
Saving noffire2.jpg to noffire2 (1).jpg
```



```
False 0.8351341
```

✓ 15s completed at 10:22 AM ● ×