

```
In [2]: import cv2
import os
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

```
In [3]: def mean_saturation(img):
        hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV) #convert it to hsv
        return np.mean(hsv[:, :, 1])
```

```
In [4]: lst = os.listdir('data/toy_store_4_/resize5')
dct = {}
for img_name in lst:
    if not os.path.isdir(img_name):
        img = cv2.imread(os.path.join('data/toy_store_4_/resize5', img_name))
        dct[img_name] = mean_saturation(img)
df = pd.Series(dct)
display(df.argmin(), df.values.min())
display(df.argmax(), df.values.max())
```

C:\Users\xinrui zhan\Anaconda3\lib\site-packages\ipykernel\_launcher.py:8: FutureWarning:

The current behaviour of 'Series.argmin' is deprecated, use 'idxmin' instead.

The behavior of 'argmin' will be corrected to return the positional minimum in the future. For now, use 'series.values.argmin' or 'np.argmin(np.array(values))' to get the position of the minimum row.

'311.jpg'

66.02283902691511

C:\Users\xinrui zhan\Anaconda3\lib\site-packages\ipykernel\_launcher.py:9: FutureWarning:

The current behaviour of 'Series.argmax' is deprecated, use 'idxmax' instead.

The behavior of 'argmax' will be corrected to return the positional maximum in the future. For now, use 'series.values.argmax' or 'np.argmax(np.array(values))' to get the position of the maximum row.

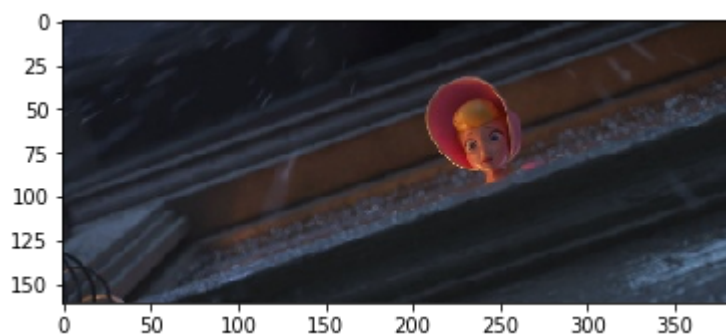
```
if __name__ == '__main__':
```

'119.jpg'

156.86248059006212

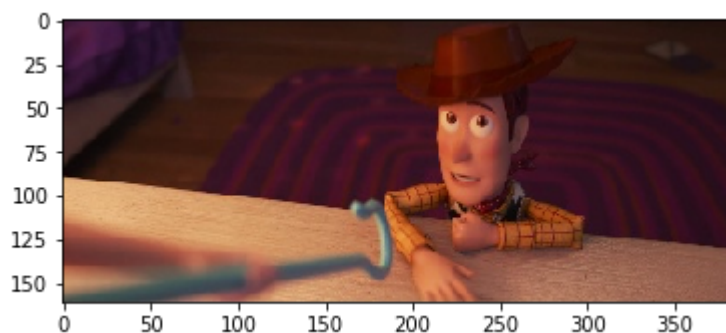
```
In [5]: plt.imshow(cv2.imread('data/toy_store_4_/resize5/311.jpg')[:, :, ::-1])
```

```
Out[5]: <matplotlib.image.AxesImage at 0x21e97d8c2b0>
```



```
In [6]: plt.imshow(cv2.imread('data/toy_store_4_/resize5/119.jpg')[:, :, ::-1])
```

```
Out[6]: <matplotlib.image.AxesImage at 0x21e97e27da0>
```



```
In [ ]:
```