IMAGE HISTOGRAM

Image Histogram

"An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance." - Image histogram (http://en.wikipedia.org/wiki/Image_histogram).

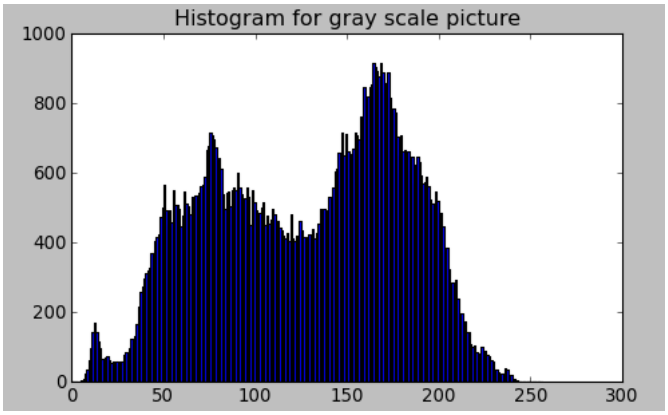
1. Histogram is a graphical representation of the intensity distribution of an image.
2. Histogram quantifies the number of pixels for each intensity value.

Here is a simple code for just loading the image:

```
import cv2
import numpy as np

gray_img = cv2.imread('images/SunsetGoldenGate.jpg', cv2.IMREAD_GRAYSCALE)
cv2.imshow('GoldenGate',gray_img)

while True:
    k = cv2.waitKey(0) & 0xFF
    if k == 27: break             # ESC key to exit
cv2.destroyAllWindows()
```



The code for histogram looks like this:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

gray_img = cv2.imread('images/GoldenGateSunset.png', cv2.IMREAD_GRAYSCALE)
cv2.imshow('GoldenGate',gray_img)
hist = cv2.calcHist([gray_img],[0],None,[256],[0,256])
plt.hist(gray_img.ravel(),256,[0,256])
plt.title('Histogram for gray scale picture')
plt.show()

while True:
    k = cv2.waitKey(0) & 0xFF
    if k == 27: break             # ESC key to exit
cv2.destroyAllWindows()
```

Note: This is how **ravel()** works, and it's equivalent of **reshape(-1)**.

```
>>> x = np.array([[1, 2, 3], [4, 5, 6]])
>>> print np.ravel(x)
[1 2 3 4 5 6]
>>> x.reshape(-1)
array([1, 2, 3, 4, 5, 6])
```

OpenCV 3 image and video processing with Python

OpenCV 3 with Python (/python/OpenCV_Python/pythc

Image - OpenCV BGR : Matplotlib RGB (/python/OpenCV_Python/pythc

Basic image operations - pixel access (/python/OpenCV_Python/pythc

iPython - Signal Processing with NumPy (/python/OpenCV_Python/pythc

Signal Processing with NumPy I - FFT and DFT for sine, square waves, unitpulse, and random signal (/python/OpenCV_Python/pythc

Signal Processing with NumPy II - Image Fourier Transform : FFT & DFT (/python/OpenCV_Python/pythc

Inverse Fourier Transform of an Image with low pass filter: cv2.idft() (/python/OpenCV_Python/pythc

Image Histogram (/python/OpenCV_Python/pythc

Video Capture and Switching colorspaces - RGB / HSV (/python/OpenCV_Python/pythc

Adaptive Thresholding - Otsu's clustering-based image

Histogram Terminology

Before using that function, we need to understand some terminologies related with histograms.

1. **bins** :The histogram above shows the number of pixels for every pixel value, from 0 to 255. In fact, we used 256 values (bins) to show the above histogram. It could be 8, 16, 32 etc. OpenCV uses **histSize** to refer to **bins**.
2. **dims** : It is the number of parameters for which we collect the data. In our case, we collect data based on intensity value. So, in our case, it is **1**.
3. **range** : It is the range of intensity values we want to measure. Normally, it is [0,256], ie all intensity values.

calcHist()

OpenCV comes with an in-built **cv2.calcHist()** function for histogram. So, it's time to look into the specific parameters related to the **cv2.calcHist()** function.

```
cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])
```

In the code, we used:

```
hist = cv2.calcHist([gray_img],[0],None,[256],[0,256])
```

The parameters are:

1. **images**: source image of type uint8 or float32. it should be given in as a list, ie, **[gray_img]**.
2. **channels**: it is also given in as a list []. It the index of channel for which we calculate histogram. For example, if input is grayscale image, its value is **[0]**. For color image, you can pass [0],[1] or [2] to calculate histogram of blue,green or red channel, respectively.
3. **mask**: mask image. To find histogram of full image, it is set as **None**. However, if we want to get histogram of specific region of image, we should create a mask image for that and give it as mask.
4. **histSize**: this represents our BIN count. Need to be given in []. For full scale, we pass **[256]**.
5. **ranges**: Normally, it is **[0,256]**.

thresholding (/python/OpenCV_Python/pythc

Edge Detection - Sobel and Laplacian Kernels (/python/OpenCV_Python/pythc

Canny Edge Detection (/python/OpenCV_Python/pythc

Hough Transform - Circles (/python/OpenCV_Python/pythc

Watershed Algorithm : Marker-based Segmentation I (/python/OpenCV_Python/pythc

Watershed Algorithm : Marker-based Segmentation II (/python/OpenCV_Python/pythc

Image noise reduction : Non-local Means denoising algorithm (/python/OpenCV_Python/pythc local_Means_Denoising_Algorith

Image object detection : Face detection using Haar Cascade Classifiers (/python/OpenCV_Python/pythc

Image segmentation - Foreground extraction Grabcut algorithm based on graph cuts (/python/OpenCV_Python/pythc

Image Reconstruction - Inpainting (Interpolation) - Fast Marching Methods (/python/OpenCV_Python/pythc

Video : Mean shift object tracking (/python/OpenCV_Python/pythc

Machine Learning : Clustering - K-Means clustering I (/python/OpenCV_Python/pythc Means_Clustering_Vector_Quan

Machine Learning : Clustering - K-Means clustering II (/python/OpenCV_Python/pythc Means_Clustering_Vector_Quan

Machine Learning :

Classification - k-nearest neighbors (k-NN) algorithm (/python/OpenCV_Python/pythc nearest_neighbors_k-NN.php)

NumPy - np.histogram()

NumPy also provides us a function for histogram, **np.histogram()**. So, we can use NumPy fucntion instead of OpenCV function:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

gray_img = cv2.imread('images/GoldenGateSunset.png', cv2.IMREAD_GRAYSCALE)
cv2.imshow('GoldenGate',gray_img)
#hist = cv2.calcHist([gray_img],[0],None,[256],[0,256])
hist,bins = np.histogram(gray_img,256,[0,256])

plt.hist(gray_img.ravel(),256,[0,256])
plt.title('Histogram for gray scale picture')
plt.show()

while True:
    k = cv2.waitKey(0) & 0xFF
    if k == 27: break             # ESC key to exit
cv2.destroyAllWindows()
```

Other parts of the code remain untouched, and it gives us the same histogram.

Histogram for color image

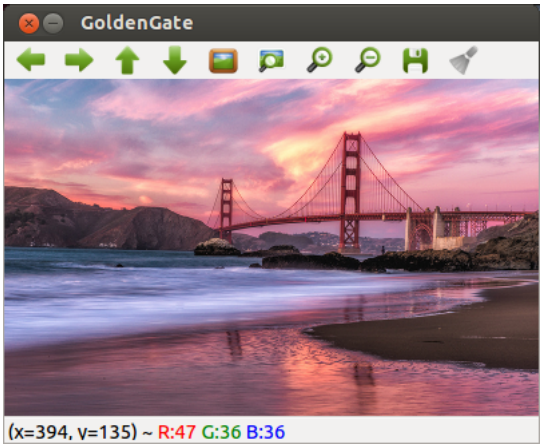
Let's draw RGB histogram:

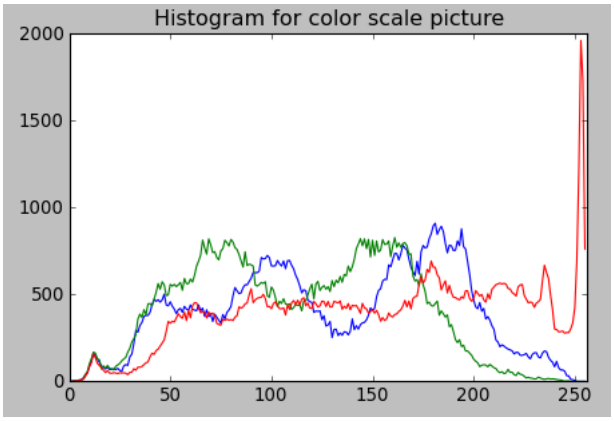
Python tutorial

Python Home (/python/pytut.php)

Introduction (/python/python_introduction.p

Running Python Programs (os, sys, import)





The code:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('images/GoldenGateSunset.png', -1)
cv2.imshow('GoldenGate',img)

color = ('b','g','r')
for channel,col in enumerate(color):
    histr = cv2.calcHist([img],[channel],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.xlim([0,256])
plt.title('Histogram for color scale picture')
plt.show()

while True:
    k = cv2.waitKey(0) & 0xFF
    if k == 27: break             # ESC key to exit
cv2.destroyAllWindows()
```

(/python/python_running.php)

Modules and IDLE (Import, Reload, exec) (/python/python_modules_idle.p

Object Types - Numbers, Strings, and None (/python/python_numbers_strin

Strings - Escape Sequence, Raw String, and Slicing (/python/python_strings.php)

Strings - Methods (/python/python_strings_metho

Formatting Strings - expressions and method calls (/python/python_string_formatt

Files and os.path (/python/python_files.php)

Traversing directories recursively (/python/python_traversing_dire

Subprocess Module (/python/python_subprocess_m

Regular Expressions with Python (/python/python_regularExpress

Object Types - Lists (/python/python_lists.php)

Object Types - Dictionaries and Tuples (/python/python_dictionaries_tu

Functions def, *args, **kargs (/python/python_functions_def.

Functions lambda (/python/python_functions_lam

Built-in Functions (/python/python_functions_built

map, filter, and reduce (/python/python_fncs_map_filte

Decorators (/python/python_decorators.ph

List Comprehension

(/python/python_list_comprehe

Sets (union/intersection) and itertools - Jaccard coefficient and shingling to check plagiarism (/python/python_sets_union_int

Hashing (Hash tables and hashlib) (/python/python_hash_tables_h

Dictionary Comprehension with zip (/python/python_dictionary_con

The yield keyword (/python/python_function_with_

Generator Functions and Expressions (/python/python_generators.ph

generator.send() method (/python/python_function_with_

Iterators (/python/python_iterators.php)

Classes and Instances (__init__, __call__, etc.) (/python/python_classes_instan

if__name__ == '__main__' (/python/python_if__name__equ

argparse (/python/python_argparse.php)

Exceptions (/python/python_try_except_fina

@static method vs class method (/python/python_differences_be

Private attributes and private methods (/python/python_private_attribu

bits, bytes, bitstring, and constBitStream (/python/python_bits_bytes_bits

json.dump(s) and json.load(s) (/python/python-json-dumps-loads-file-read-write.php)

Python Object Serialization - pickle and json (/python/python_serialization_p

Python Object Serialization - yaml and json (/python/python_yaml_json_con

Priority queue and heap queue data structure (/python/python_PriorityQueue_

Graph data structure (/python/python_graph_data_st

Dijkstra's shortest path algorithm (/python/python_Dijkstras_Shor

Prim's spanning tree algorithm (/python/python_Prims_Spannir

Closure (/python/python_closure.php)

Functional programming in Python (/python/python_functional_pro

Remote running a local file using ssh (/python/python_ssh_remote_ru

SQLite 3 - A. Connecting to DB, create/drop table, and insert data into a table (/python/python_sqlite_connect

SQLite 3 - B. Selecting, updating and deleting data (/python/python_sqlite_select_u

MongoDB with PyMongo I - Installing MongoDB ... (/python/MongoDB_PyMongo/p

Python HTTP Web Services - urllib, httplib2 (/python/python_http_web_serv

Web scraping with Selenium for checking domain availability (/python/python_Web_scraping_

REST API : Http Requests for Humans with Flask

(/python/python-REST-API-Http-Requests-for-Humans-with-Flask.php)

Blog app with Tornado (/python/Tornado/Python_Torna

Multithreading ... (/python/Multithread/python_m

Python Network Programming I - Basic Server / Client : A Basics (/python/python_network_progr

Python Network Programming I - Basic Server / Client : B File Transfer (/python/python_network_progr

Python Network Programming II - Chat Server / Client (/python/python_network_progr

Python Network Programming III - Echo Server using socketserver network framework (/python/python_network_progr

Python Network Programming IV - Asynchronous Request Handling : ThreadingMixIn and ForkingMixIn (/python/python_network_progr

Python Interview Questions I (/python/python_interview_ques

Python Interview Questions II (/python/python_interview_ques

Python Interview Questions III (/python/python_interview_ques

Python Interview Questions IV (/python/python_interview_ques

Python Interview Questions V (/python/python_interview_ques

Image processing with Python image library Pillow (/python/python_image_process

Python and C++ with SIP (/python/python_cpp_sip.php)

PyDev with Eclipse
(/python/pydev_eclipse_plugin_i

Matplotlib
(/python/python_matplotlib.php

Redis with Python
(/python/python_redis_with_pyt

NumPy array basics A
(/python/python_numpy_array_

NumPy Matrix and Linear Algebra
(/python/python_numpy_matrix

Pandas with NumPy and Matplotlib
(/python/python_Pandas_NumP

Celluar Automata
(/python/python_cellular_autom

Batch gradient descent algorithm
(/python/python_numpy_batch_

Longest Common Substring Algorithm
(/python/python_longest_comm

Python Unit Test - TDD using unittest.TestCase class
(/python/python_unit_testing.ph

Simple tool - Google page ranking by keywords
(/python/python_site_page_rank

Google App Hello World
(/python/GoogleApp/python_Gc

Google App webapp2 and WSGI
(/python/GoogleApp/python_Gc

Uploading Google App Hello World
(/python/GoogleApp/python_Gc

Python 2 vs Python 3
(/python/python_differences_Py

virtualenv and virtualenvwrapper
(/python/python_virtualenv_virt

Uploading a big file to AWS S3

using boto module
(/DevOps/AWS/aws_S3_uploadi

Scheduled stopping and
starting an AWS instance
(/DevOps/AWS/aws_stopping_st

Cloudera CDH5 - Scheduled
stopping and starting services
(/Hadoop/BigData_hadoop_CDH

Removing Cloud Files -
Rackspace API with curl and
subprocess
(/python/python_Rackspace_API

Checking if a process is
running/hanging and stop/run
a scheduled task on Windows
(/python/python-Windows-
Check-if-a-Process-is-Running-
Hanging-Schtasks-Run-
Stop.php)

Apache Spark 1.3 with PySpark
(Spark Python API) Shell
(/Hadoop/BigData_hadoop_Apa

Apache Spark 1.2 Streaming
(/Hadoop/BigData_hadoop_Apa

bottle 0.12.7 - Fast and simple
WSGI-micro framework for
small web-applications ...
(/python/Bottle/Python_Bottle_F

Flask app with Apache WSGI
on Ubuntu14/CentOS7 ...
(/python/Flask/Python_Flask_Blo

Selenium WebDriver
(/python/python_Selenium_Web

Fabric - streamlining the use
of SSH for application
deployment
(/python/Fabric/python_Fabric.p

Ansible Quick Preview - Setting
up web servers with Nginx,
configure enviroments, and
deploy an App
(/DevOps/Ansible/Ansible_Settin

Neural Networks with
backpropagation for XOR
using one hidden layer
(/python/python_Neural_Netwo

NLP - NLTK (Natural Language Toolkit) ... (/python/NLTK/NLTK_install.php

RabbitMQ(Message broker server) and Celery(Task queue) ... (/python/RabbitMQ_Celery/pyth

OpenCV3 and Matplotlib ... (/python/OpenCV_Python/pythc

Simple tool - Concatenating slides using FFmpeg ... (/FFMpeg/ffmpeg_fade_in_fade_

iPython - Signal Processing with NumPy (/python/OpenCV_Python/pythc

iPython and Jupyter - Install Jupyter, iPython Notebook, drawing with Matplotlib, and publishing it to Github (/python/IPython/IPython_Jupyt

iPython and Jupyter Notebook with Embedded D3.js (/python/IPython/iPython_Jupyt

Downloading YouTube videos using youtube-dl embedded with Python (/VideoStreaming/YouTube/yout dl-embedding.php)

Machine Learning : scikit-learn ... (/python/scikit-learn/scikit_machine_learning_S

Django 1.6/1.8 Web Framework ... (/python/Django/Python_Django

CONTACT

BogoToBogo
contactus@bogotobogo.com (mailto:#)

FOLLOW BOGOTOBOGO

(https://www.facebook.com/KHongSanFrancisco) (https://twitter.com/KHongTwit) (https://plus.google.com/u/0/+KHongSanFrancisco/posts)

ABOUT US (/ABOUT_US.PHP)

contactus@bogotobogo.com (mailto:#)

Golden Gate Ave, San Francisco, CA 94115

Golden Gate Ave, San Francisco, CA 94115