a c c c	<pre>m=cv.imread('D:\Semester 4\Image processing\Homework\Homework2\spider.png',cv.IMREAD_GRAYSCALE) ssert im is not None v.namedWindow('Image',cv.WINDOW_AUTOSIZE) v.imshow('Image',im) v.waitKey(0) amma = [0.2,0.8,1,1.2,2] ig, ax = plt.subplots() lt.imshow(im, 'gray') lt.title('Original image in grayscale')</pre>
	<pre>or i in gamma: t = np.array([(p/255)**i*255 for p in range (0,256)]).astype(np.uint8) g = cv.LUT(im,t) fig, ax = plt.subplots() ax.plot(t) fig = plt.subplots() plt.imshow(g, 'gray') plt.show()</pre>
	cv.namedWindow('Image',cv.WINDOW_AUTOSIZE) cv.imshow('Image',g) cv.waitKey(0) v.destroyAllWindows() Original image in grayscale
2 2 3 3 4	
1	
1 1 2 2 3	0 50 100 150 200 250 00 - 50 - 50 - 50 - 50 - 50 - 50 - 5
2	
1	
2 3 3 4	50 - 100 - 200 300 400 500 600 700 - 100 -
1	
1 2 2 3 3	
1	50
1 2 2	
2	50 - 100 200 300 400 500 600 700 50 - 100 200 300 400 500 600 700 50 - 100 - 100 200 300 400 500 600 700
1	
In []: t	1= np.linspace (0,100,50) 2= np.linspace (100,255,150) 3= np.linspace (255,255, 56)
f a a a g	<pre>= np.concatenate((t1,t2,t3), axis =0).astype(np.uint8) ig,ax = plt.subplots() x.plot(t) x.set_aspect('equal') ssert len(t) ==256 = cv.LUT(im,t) v.namedWindow('Image',cv.WINDOW_AUTOSIZE) v.imshow('Image',g)</pre>
f p p p p	<pre>v.waitKey(0) v.destroyAllWindows() ig,ax = plt.subplots(2,1,figsize=(10,8)) lt.subplot(2,1,1) lt.imshow(im, 'gray') lt.title('Original') lt.subplot(2,1,2) lt.imshow(g, 'gray') lt.title('Intesity transform image') ext(0.5, 1.0, 'Intesity transform image')</pre>
1	
1 1 2 2	Original Original
3 4 1	00 - 00 - 100 200 300 400 500 600 700 Intesity transform image
In []: i	mport numpy as np mport cv2 as cv mport matplotlib.pyplot as plt
h g h f a a	m=cv.calcHist([im], [0], None, [256], [0,256]) = cv.equalizeHist(im) ist_m = cv.calcHist([im], [0], None, [256], [0,256]) = cv.equalizeHist(im) ist_g = cv.calcHist([g], [0], None, [256], [0,256]) ist_g = cv.calcHist([g], [in], None, [256], None, None, [256], None, N
f p p p	<pre>v.waitKey(0) v.destroyAllWindows() ig = plt.subplots(2,1,figsize = (10,8)) lt.subplot(1,2,1) lt.imshow(im, "gray") lt.title("Original image") lt.subplot(1,2,2) lt.imshow(g, "gray") lt.title("Histogram equalize image") ext(0.5, 1.0, 'Histogram equalize image')</pre>
1 1 1	7500 - 25
1 1 1	0 1
1	2500 -
In []: i	00 -
h (s s	<pre>ssert im is not None sv0_im = cv.cvtColor(im, cv.CoLoR_BGR2HSV) h0, s0, v0) = cv.split(hsv0_im) 1= np.linspace (0,100,75) 2= np.linspace (100,225,125) 3= np.linspace (225,255, 56) = np.concatenate((s1,s2,s3), axis =0).astype(np.uint8) ig, ax = plt.subplots()</pre>
a a s h b	<pre>x.plot(s) x.set_aspect('equal') ssert len(s) ==256 new= cv.LUT(s0,s) sv1_im = cv.merge([h0,snew,v0]) gr_im1 = cv.cvtColor(hsv1_im, cv.Color_Hsv2BGR) v.namedWindow('Image',cv.WINDOW_AUTOSIZE) v.imshow('Image',bgr_im1) v.waitKey(0) v.destroyAllwindows()</pre>
f p p p	ig = plt.subplots(2,1,figsize=(10,8)) lt.subplot(2,1,1) lt.imshow(cv.cvtColor(hsv0_im, cv.COLOR_HSV2RGB)) lt.title("Original image") lt.subplot(2,1,2) lt.imshow(cv.cvtColor(hsv1_im, cv.COLOR_HSV2RGB)) lt.title("Saturation transformed image") ext(0.5, 1.0, 'Saturation transformed image')
1	
2	Original image Output Output
1	00 - 100 200 300 400 500 600 700 800 Saturation transformed image
In []: i a h	m=cv.imread('D:\Semester 4\Image processing\Homework\Homework2\zion_pass.jpg') ssert im is not None sv0_im = cv.cvtColor(im, cv.CoLOR_BGR2HSV)
h h h f a a	h0, s0, v0) = cv.split(hsv0_im) 1= np.linspace (0,75,25) 2= np.linspace (75,100,50) 3= np.linspace (100,150, 106) 4= np.linspace (150,255, 75) = np.concatenate((h1,h2,h3,h4), axis =0).astype(np.uint8) ig,ax = plt.subplots() x.plot(h) x.set_aspect('equal')
h b c c c	<pre>ssert len(h) ==256 new= cv.LUT(h0,h) sv2_im = cv.merge([hnew,s0,v0]) gr_im2 = cv.cvtColor(hsv2_im, cv.CoLoR_HSV2BGR) v.namedWindow('Image',cv.WINDOW_AUTOSIZE) v.imshow('Image',bgr_im2) v.imshow('Image',bgr_im2) v.waitKey(0) v.destroyAllWindows() ig = plt.subplot(2,1,figsize=(10,8)) lt subplot(2,1,figsize=(10,8))</pre>
p p p p p p p	<pre>lt.subplot(2,1,1) lt.imshow(cv.cvtColor(hsv0_im, cv.COLOR_HSV2RGB)) lt.title("Original image") lt.subplot(2,1,2) lt.imshow(cv.cvtColor(hsv2_im, cv.COLOR_HSV2RGB)) lt.title("Hue transformed image") ext(0.5, 1.0, 'Hue transformed image') 50</pre>
1	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
	Original image
2 3 4	
2 3 4 5	
2 3 4 5 1 2 3	00

Name = Devindi De Silva

Index number = 190128H

import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt