

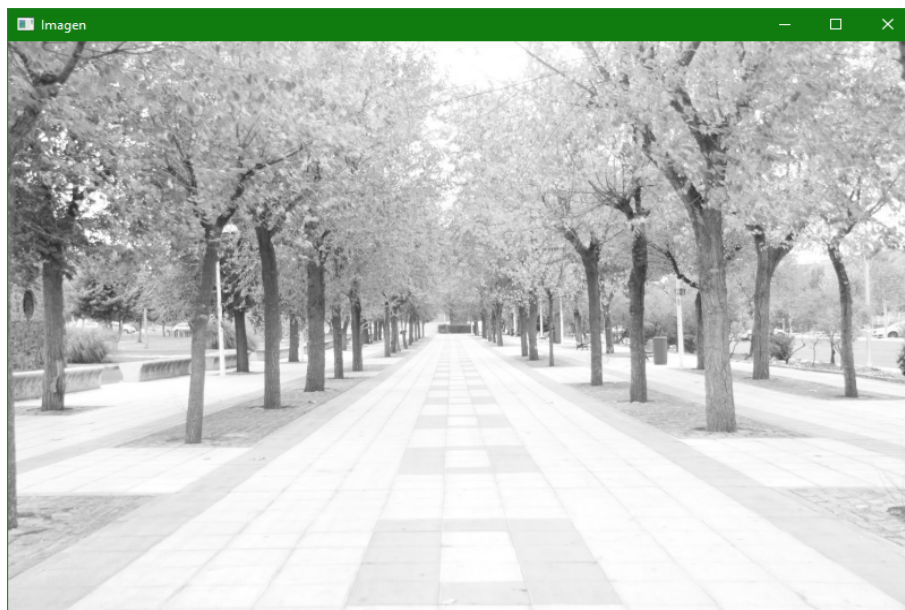
# Algoritmo 4

Muñoz Nuñez Ian Emmanuel

Histograma de la imagen

Visión Robótica

*Imagen usada para el algoritmo*



## *Punto 1*

```
# Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
import cv2
# Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
import matplotlib.pyplot as plt
# Se importa la libreria imutils para modificar el tamaño de la imagen
import imutils

# Se lee la imagen 'imagen.jpg' en escala de grises
imagen = cv2.imread("imagen.jpg", 0)
# Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
imagen = imutils.resize(imagen, 800, 800)

# Se asignan a M y N el tamaño de la matriz de la imagen
M, N = imagen.shape[0:2]

# Se crea el histograma de la imagen con el canal 0, sin mascara, con 256 puntos
# y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
# sin subarreglos. Todo el arreglo se divide entre el area de la imagen
histograma = cv2.calcHist([imagen], [0], None, [256], [0, 256]).flatten()/(M*N)

# Se muestra la imagen con opencv con el titulo de 'Imagen'
cv2.imshow("Imagen", imagen)

# Se agrega el histograma calculado al pyplot
plt.bar(range(len(histograma)), histograma)
# Se muestra el histograma con el pyplot
plt.show()

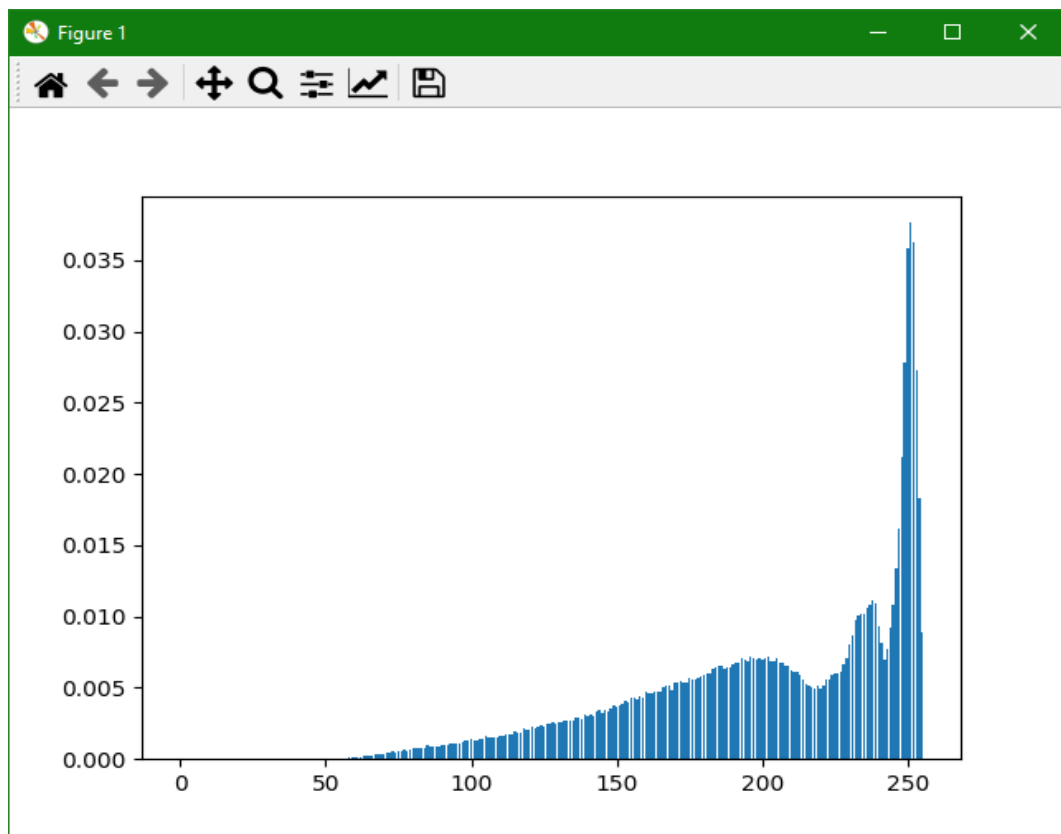
# cv2 espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el algoritmo del punto número 1, primero se importan las librerías necesarias, como ***opencv***, ***matplotlib*** y ***imutils***. Después se lee la imagen 'imagen.jpg' que se encuentra en el mismo directorio del programa y se ajusta a un tamaño que no sea mayor a 800 tanto de ancho como de alto, se asignan los valores del ancho y del alto a las variables M y N. Luego se calcula el histograma de la imagen y se divide en 256 puntos. Por último se muestran la imagen con ***opencv*** y el histograma con ***pyplot***, al final ***opencv*** espera a que el usuario presione una tecla para terminar el programa.

## *Código del algoritmo del punto 1*

```
1 # Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
2 import cv2
3 # Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
4 import matplotlib.pyplot as plt
5 # Se importa la libreria imutils para modificar el tamaño de la imagen
6 import imutils
7
8 # Se lee la imagen 'imagen.jpg' en escala de grises
9 imagen = cv2.imread("imagen.jpg", 0)
10 # Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
11 imagen = imutils.resize(imagen, 800, 800)
12
13 # Se asignan a M y N el tamaño de la matriz de la imagen
14 M, N = imagen.shape[0:2]
15
16 # Se crea el histograma de la imagen con el canal 0, sin mascara, con 256 puntos
17 # y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
18 # sin subarreglos. Todo el arreglo se divide entre el area de la imagen
19 histograma = cv2.calcHist([imagen], [0], None, [256], [0, 256]).flatten()/(M*N)
20
21 # Se muestra la imagen con opencv con el titulo de 'Imagen'
22 cv2.imshow("Imagen", imagen)
23
24 # Se agrega el histograma calculado al pyplot
25 plt.bar(range(len(histograma)), histograma)
26 # Se muestra el histograma con el pyplot
27 plt.show()
28
29 # cv2 espera a que el usuario presione una tecla para terminar el programa
30 cv2.waitKey(0)
31
```

## *Histograma de la imagen del punto 1*



## *Punto 2*

```
# Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
import cv2
# Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
import matplotlib.pyplot as plt
# Se importa la libreria imutils para modificar el tamaño de la imagen
import imutils

# Se lee la imagen 'imagen.jpg' en escala de grises
imagen = cv2.imread('imagen.jpg', 0)
# Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
imagen = imutils.resize(imagen, 800, 800)

# Se asignan a M y N el tamaño de la matriz de la imagen
M, N = imagen.shape[0:2]

# Se crea el histograma de la imagen con el canal 0, sin mascara, con terciles
# y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
# sin subarreglos. Todo el arreglo se divide entre el area de la imagen
histograma = cv2.calcHist([imagen], [0], None, [3], [0, 256]).flatten()/(M*N)

# Se muestra la imagen con opencv con el titulo de 'Imagen'
cv2.imshow('Imagen', imagen)

# Se agrega el histograma calculado al pyplot
plt.bar(range(len(histograma)), histograma)
# Se muestra el histograma con el pyplot
plt.show()

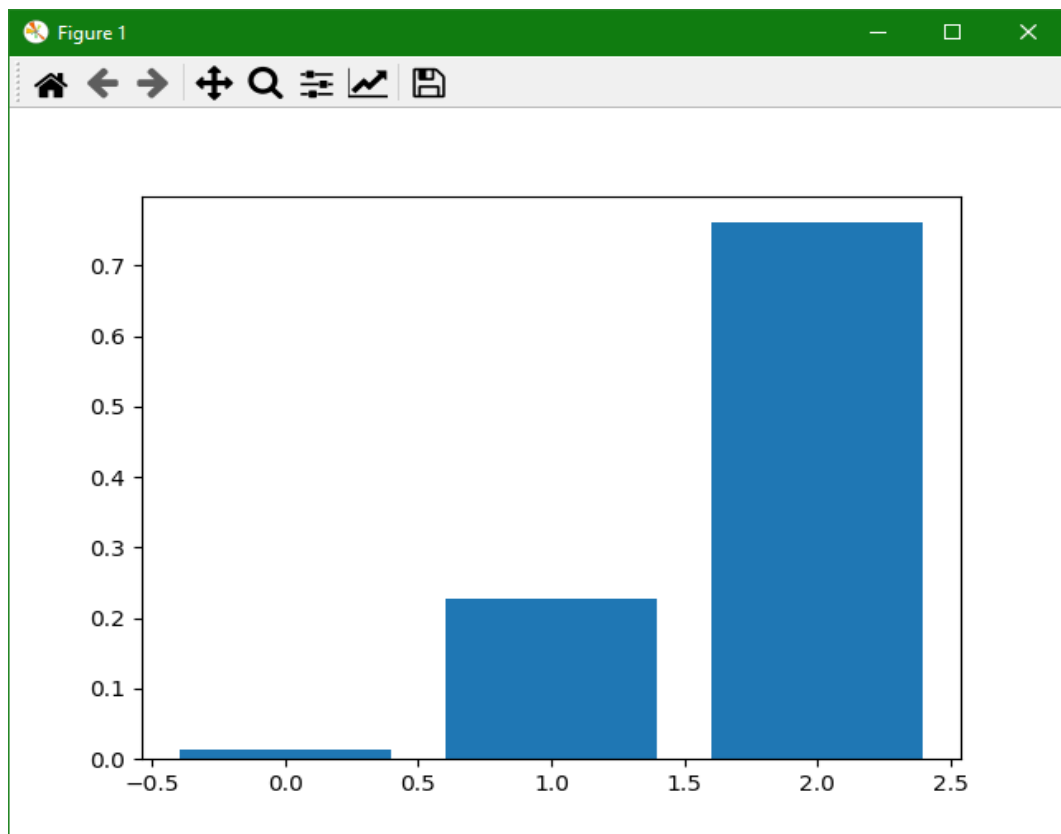
# cv2 espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el algoritmo del punto número 2, primero se importan las librerías necesarias, como ***opencv***, ***matplotlib*** y ***imutils***. Después se lee la imagen 'imagen.jpg' que se encuentra en el mismo directorio del programa y se ajusta a un tamaño que no sea mayor a 800, tanto de ancho como de alto, se asignan los valores del ancho y del alto a las variables M y N. Luego se calcula el histograma de la imagen y se divide en terciles. Por último se muestra la imagen con ***opencv*** y el histograma con ***pyplot***, al final ***opencv*** espera a que el usuario presione una tecla para termina el programa.

## *Código del algoritmo del punto 2*

```
1 # Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
2 import cv2
3 # Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
4 import matplotlib.pyplot as plt
5 # Se importa la libreria imutils para modificar el tamaño de la imagen
6 import imutils
7
8 # Se lee la imagen 'imagen.jpg' en escala de grises
9 imagen = cv2.imread("imagen.jpg", 0)
10 # Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
11 imagen = imutils.resize(imagen, 800, 800)
12
13 # Se asignan a M y N el tamaño de la matriz de la imagen
14 M, N = imagen.shape[0:2]
15
16 # Se crea el histograma de la imagen con el canal 0, sin mascara, con terciles
17 # y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
18 # sin subarreglos. Todo el arreglo se divide entre el area de la imagen
19 histograma = cv2.calcHist([imagen], [0], None, [3], [0, 256]).flatten()/(M*N)
20
21 # Se muestra la imagen con opencv con el titulo de 'Imagen'
22 cv2.imshow("Imagen", imagen)
23
24 # Se agrega el histograma calculado al pyplot
25 plt.bar(range(len(histograma)), histograma)
26 # Se muestra el histograma con el pyplot
27 plt.show()
28
29 # cv2 espera a que el usuario presione una tecla para terminar el programa
30 cv2.waitKey(0)
31
```

## *Histograma de la imagen del punto 2*



### *Punto 3*

```
# Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
import cv2
# Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
import matplotlib.pyplot as plt
# Se importa la libreria imutils para modificar el tamaño de la imagen
import imutils

# Se lee la imagen 'imagen.jpg' en escala de grises
imagen = cv2.imread('imagen.jpg', 0)
# Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
imagen = imutils.resize(imagen, 800, 800)

# Se asignan a M y N el tamaño de la matriz de la imagen
M, N = imagen.shape[0:2]

# Se crea el histograma de la imagen con el canal 0, sin mascara, con cuartiles
# y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
# sin subarreglos. Todo el arreglo se divide entre el area de la imagen
histograma = cv2.calcHist([imagen], [0], None, [4], [0, 256]).flatten()/(M*N)

# Se muestra la imagen con opencv con el titulo de 'Imagen'
cv2.imshow('Imagen', imagen)

# Se agrega el histograma calculado al pyplot
plt.bar(range(len(histograma)), histograma)
# Se muestra el histograma con el pyplot
plt.show()

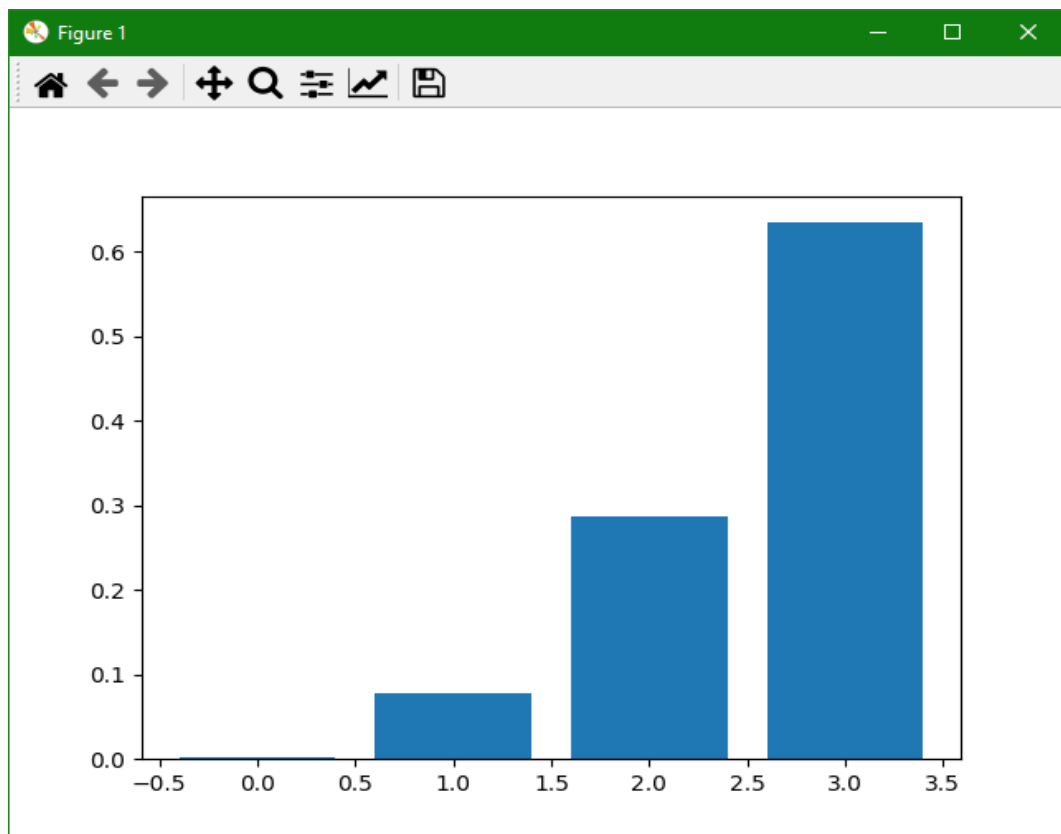
# cv2 espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el algoritmo del punto número 3, primero se importan las librerías necesarias, como ***opencv***, ***matplotlib*** y ***imutils***. Después se lee la imagen 'imagen.jpg' que se encuentra en el mismo directorio del programa y se ajusta a un tamaño que no sea mayor a 800, tanto de ancho como de alto, se asignan los valores del ancho y del alto a las variables M y N. Luego se calcula el histograma de la imagen y se divide en cuartiles. Por último se muestra la imagen con ***opencv*** y el histograma con ***pyplot***, al final ***opencv*** espera a que el usuario presione una tecla para termina el programa.

## *Código del algoritmo del punto 3*

```
1 # Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
2 import cv2
3 # Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
4 import matplotlib.pyplot as plt
5 # Se importa la libreria imutils para modificar el tamaño de la imagen
6 import imutils
7
8 # Se lee la imagen 'imagen.jpg' en escala de grises
9 imagen = cv2.imread("imagen.jpg", 0)
10 # Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
11 imagen = imutils.resize(imagen, 800, 800)
12
13 # Se asignan a M y N el tamaño de la matriz de la imagen
14 M, N = imagen.shape[0:2]
15
16 # Se crea el histograma de la imagen con el canal 0, sin mascara, con cuartiles
17 # y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
18 # sin subarreglos. Todo el arreglo se divide entre el area de la imagen
19 histograma = cv2.calcHist([imagen], [0], None, [4], [0, 256]).flatten()/(M*N)
20
21 # Se muestra la imagen con opencv con el titulo de 'Imagen'
22 cv2.imshow("Imagen", imagen)
23
24 # Se agrega el histograma calculado al pyplot
25 plt.bar(range(len(histograma)), histograma)
26 # Se muestra el histograma con el pyplot
27 plt.show()
28
29 # cv2 espera a que el usuario presione una tecla para terminar el programa
30 cv2.waitKey(0)
31
```

## *Histograma de la imagen del punto 3*



## *Punto 4*

```
# Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
import cv2
# Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
import matplotlib.pyplot as plt
# Se importa la libreria imutils para modificar el tamaño de la imagen
import imutils

# Se lee la imagen 'imagen.jpg' en escala de grises
imagen = cv2.imread('imagen.jpg', 0)
# Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
imagen = imutils.resize(imagen, 800, 800)

# Se asignan a M y N el tamaño de la matriz de la imagen
M, N = imagen.shape[0:2]

# Se crea el histograma de la imagen con el canal 0, sin mascara, con quintiles
# y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
# sin subarreglos. Todo el arreglo se divide entre el area de la imagen
histograma = cv2.calcHist([imagen], [0], None, [5], [0, 256]).flatten()/(M*N)

# Se muestra la imagen con opencv con el titulo de 'Imagen'
cv2.imshow('Imagen', imagen)

# Se agrega el histograma calculado al pyplot
plt.bar(range(len(histograma)), histograma)
# Se muestra el histograma con el pyplot
plt.show()

# cv2 espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

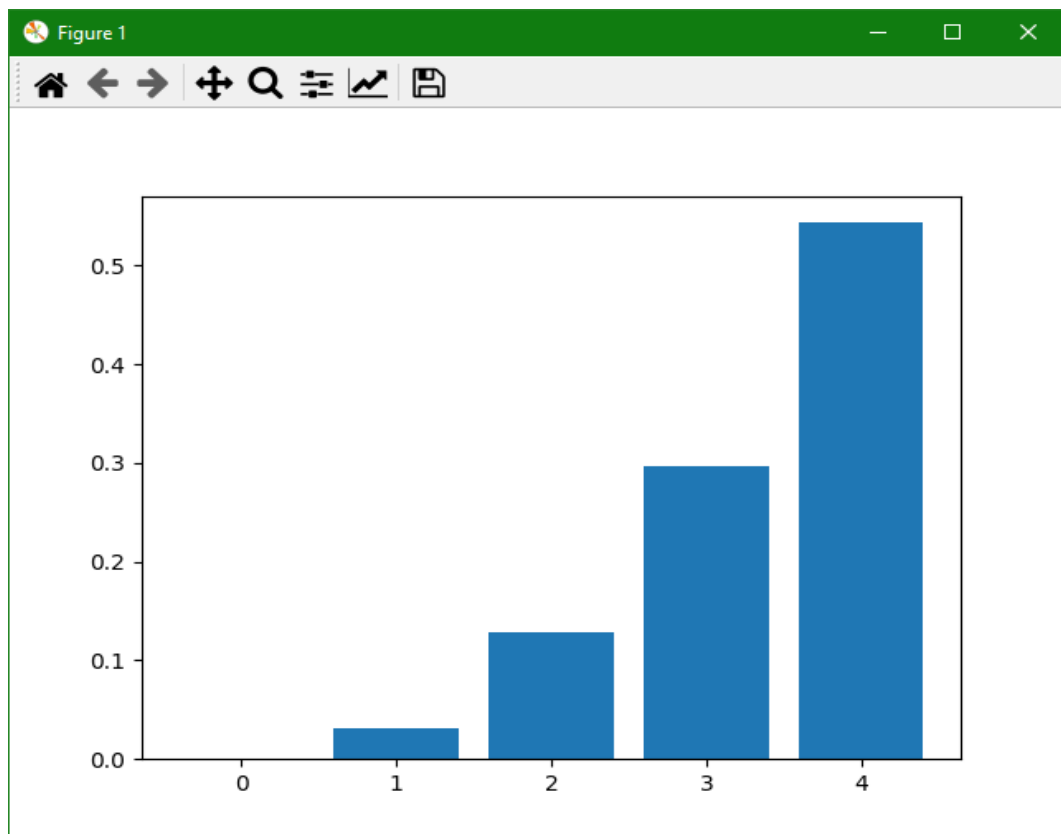
Para el algoritmo del punto número 4, primero se importan las librerías necesarias, como ***opencv***, ***matplotlib*** y ***imutils***. Después se lee la imagen 'imagen.jpg' que se encuentra en el mismo directorio del programa y se ajusta a un tamaño que no sea mayor a 800, tanto de ancho como de alto, se asignan los valores del ancho y del alto a las variables M y N. Luego se calcula el histograma de la imagen y se divide en quintiles. Por último se muestra la imagen con ***opencv*** y el histograma con ***pyplot***, al final ***opencv*** espera a que el usuario presione una tecla para termina el programa.



## *Código del algoritmo del punto 4*

```
1 # Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
2 import cv2
3 # Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
4 import matplotlib.pyplot as plt
5 # Se importa la libreria imutils para modificar el tamaño de la imagen
6 import imutils
7
8 # Se lee la imagen 'imagen.jpg' en escala de grises
9 imagen = cv2.imread("imagen.jpg", 0)
10 # Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
11 imagen = imutils.resize(imagen, 800, 800)
12
13 # Se asignan a M y N el tamaño de la matriz de la imagen
14 M, N = imagen.shape[0:2]
15
16 # Se crea el histograma de la imagen con el canal 0, sin mascara, con quintiles
17 # y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
18 # sin subarreglos. Todo el arreglo se divide entre el area de la imagen
19 histograma = cv2.calcHist([imagen], [0], None, [5], [0, 256]).flatten()/(M*N)
20
21 # Se muestra la imagen con opencv con el titulo de 'Imagen'
22 cv2.imshow("Imagen", imagen)
23
24 # Se agrega el histograma calculado al pyplot
25 plt.bar(range(len(histograma)), histograma)
26 # Se muestra el histograma con el pyplot
27 plt.show()
28
29 # cv2 espera a que el usuario presione una tecla para terminar el programa
30 cv2.waitKey(0)
31
```

## *Histograma de la imagen del punto 4*



## *Punto 5*

```
# Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
import cv2
# Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
import matplotlib.pyplot as plt
# Se importa la libreria imutils para modificar el tamaño de la imagen
import imutils

# Se lee la imagen 'imagen.jpg' en escala de grises
imagen = cv2.imread('imagen.jpg', 0)
# Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
imagen = imutils.resize(imagen, 800, 800)

# Se asignan a M y N el tamaño de la matriz de la imagen
M, N = imagen.shape[0:2]

# Se crea el histograma de la imagen con el canal 0, sin mascara, con deciles
# y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
# sin subarreglos. Todo el arreglo se divide entre el area de la imagen
histograma = cv2.calcHist([imagen], [0], None, [10], [0, 256]).flatten()/(M*N)

# Se muestra la imagen con opencv con el titulo de 'Imagen'
cv2.imshow('Imagen', imagen)

# Se agrega el histograma calculado al pyplot
plt.bar(range(len(histograma)), histograma)
# Se muestra el histograma con el pyplot
plt.show()

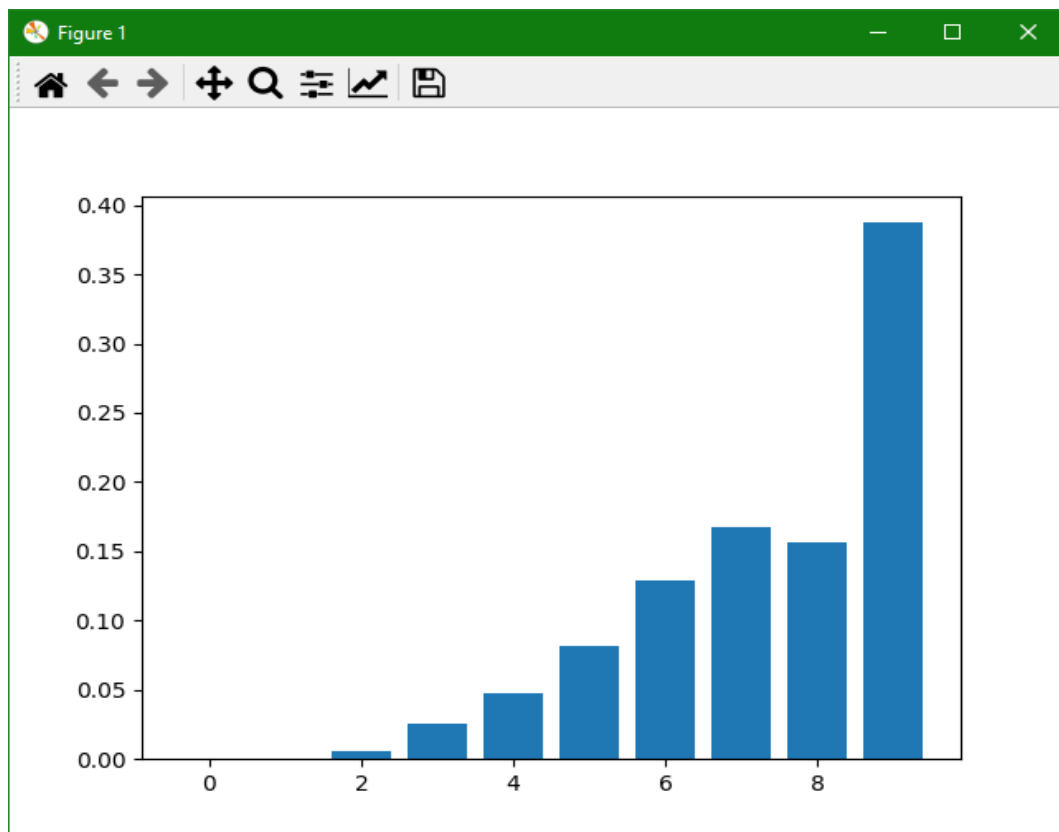
# cv2 espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el algoritmo del punto número 5, primero se importan las librerías necesarias, como *opencv*, *matplotlib* y *imutils*. Después se lee la imagen 'imagen.jpg' que se encuentra en el mismo directorio del programa y se ajusta a un tamaño que no sea mayor a 800, tanto de ancho como de alto, se asignan los valores del ancho y del alto a las variables M y N. Luego se calcula el histograma de la imagen y se divide en deciles. Por último se muestra la imagen con *opencv* y el histograma con *pyplot*, al final *opencv* espera a que el usuario presione una tecla para termina el programa.

## *Código del algoritmo del punto 5*

```
1 # Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
2 import cv2
3 # Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
4 import matplotlib.pyplot as plt
5 # Se importa la libreria imutils para modificar el tamaño de la imagen
6 import imutils
7
8 # Se lee la imagen 'imagen.jpg' en escala de grises
9 imagen = cv2.imread("imagen.jpg", 0)
10 # Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
11 imagen = imutils.resize(imagen, 800, 800)
12
13 # Se asignan a M y N el tamaño de la matriz de la imagen
14 M, N = imagen.shape[0:2]
15
16 # Se crea el histograma de la imagen con el canal 0, sin mascara, con deciles
17 # y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
18 # sin subarreglos. Todo el arreglo se divide entre el area de la imagen
19 histograma = cv2.calcHist([imagen], [0], None, [10], [0, 256]).flatten()/(M*N)
20
21 # Se muestra la imagen con opencv con el titulo de 'Imagen'
22 cv2.imshow("Imagen", imagen)
23
24 # Se agrega el histograma calculado al pyplot
25 plt.bar(range(len(histograma)), histograma)
26 # Se muestra el histograma con el pyplot
27 plt.show()
28
29 # cv2 espera a que el usuario presione una tecla para terminar el programa
30 cv2.waitKey(0)
31
```

## *Histograma de la imagen del punto 5*



## *Punto 6*

```
# Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
import cv2
# Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
import matplotlib.pyplot as plt
# Se importa la libreria imutils para modificar el tamaño de la imagen
import imutils

# Se lee la imagen 'imagen.jpg' en escala de grises
imagen = cv2.imread('imagen.jpg', 0)
# Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
imagen = imutils.resize(imagen, 800, 800)

# Se asignan a M y N el tamaño de la matriz de la imagen
M, N = imagen.shape[0:2]

# Se crea el histograma de la imagen con el canal 0, sin mascara, con percentiles
# y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
# sin subarreglos. Todo el arreglo se divide entre el area de la imagen
histograma = cv2.calcHist([imagen], [0], None, [100], [0, 256]).flatten()/(M*N)

# Se muestra la imagen con opencv con el titulo de 'Imagen'
cv2.imshow('Imagen', imagen)

# Se agrega el histograma calculado al pyplot
plt.bar(range(len(histograma)), histograma)
# Se muestra el histograma con el pyplot
plt.show()

# cv2 espera a que el usuario presione una tecla para terminar el programa
cv2.waitKey(0)
```

Para el algoritmo del punto número 6, primero se importan las librerías necesarias, como ***opencv***, ***matplotlib*** y ***imutils***. Después se lee la imagen 'imagen.jpg' que se encuentra en el mismo directorio del programa y se ajusta a un tamaño que no sea mayor a 800, tanto de ancho como de alto, se asignan los valores del ancho y del alto a las variables M y N. Luego se calcula el histograma de la imagen y se divide en percentiles. Por último se muestra la imagen con ***opencv*** y el histograma con ***pyplot***, al final ***opencv*** espera a que el usuario presione una tecla para termina el programa.

## *Código del algoritmo del punto 6*

```
1 # Se importa la libreria cv2 para poder leer, mostrar y manipular la imagen
2 import cv2
3 # Se importa la libreria matplotlib.pyplot para calcular el histograma y mostrarlo
4 import matplotlib.pyplot as plt
5 # Se importa la libreria imutils para modificar el tamaño de la imagen
6 import imutils
7
8 # Se lee la imagen 'imagen.jpg' en escala de grises
9 imagen = cv2.imread("imagen.jpg", 0)
10 # Se cambia el tamaño de la imagen por y se ajusta a un ancho y alto no mayores a 800
11 imagen = imutils.resize(imagen, 800, 800)
12
13 # Se asignan a M y N el tamaño de la matriz de la imagen
14 M, N = imagen.shape[0:2]
15
16 # Se crea el histograma de la imagen con el canal 0, sin mascara, con percentiles
17 # y con un rango de 0 a 256, se usa la funcion 'flatten' para regresar un arreglo
18 # sin subarreglos. Todo el arreglo se divide entre el area de la imagen
19 histograma = cv2.calcHist([imagen], [0], None, [100], [0, 256]).flatten()/(M*N)
20
21 # Se muestra la imagen con opencv con el titulo de 'Imagen'
22 cv2.imshow("Imagen", imagen)
23
24 # Se agrega el histograma calculado al pyplot
25 plt.bar(range(len(histograma)), histograma)
26 # Se muestra el histograma con el pyplot
27 plt.show()
28
29 # cv2 espera a que el usuario presione una tecla para terminar el programa
30 cv2.waitKey(0)
31
```

## *Histograma de la imagen del punto 6*

