

Práctica #2



Materia: Sistemas de Visión Artificial

Grupo: 7°E1

Isaac Alejandro Gutiérrez Huerta 19110198

22/04/2022

Práctica #2

Objetivo:

Elegir 2 imágenes de nuestra elección y utilizarlas para realizar las siguientes operaciones:

La suma
La resta
La división
La Multiplicación
Logaritmo natural
Raíz
Derivada
Potencia
Conjunción
Disyunción
Negación
Traslación
Escalado
Rotación
Traslación A fin.
Transpuesta
Proyección.

Mostrar 2 imágenes al pulsar La primera letra de su nombre las imágenes se pondrán a los lados y a centro aparecerá la operación.

Código:

#Isaac Alejandro Gutiérrez Huerta 19110198 7E1

#Sistemas de Visión Artificial

```
import numpy as np
from PIL import Image
from matplotlib import pyplot as plt
import cv2 #opencv
import os
import math
```

```

os.system("cls")

img1=cv2.imread('Guepardo.png',1)
img2=cv2.imread('Aguila.png',1)

img_orig = np.concatenate((img1, img2), axis=1)

cv2.imshow("Imagenes originales",img_orig)
k=0
while (k!=105):
    k=cv2.waitKey(0)
    if k == ord('i'): # wait for 'i' key to continue
        cv2.destroyAllWindows()

#-----SUMA
suma1=cv2.add(img1,img2)
cv2.imwrite("Suma1.png",suma1)
suma1 = np.concatenate((img1, suma1, img2), axis=1)

suma2=img1+img2
suma2 = np.concatenate((img1, suma2, img2), axis=1)

suma3=np.array(img1)+np.array(img2)
suma3=np.concatenate((img1, suma3, img2), axis=1)

suma=np.concatenate((suma1, suma2, suma3), axis=0)
cv2.imshow("Suma",suma)
cv2.imwrite("Suma.png",suma)
cv2.waitKey(3000)
cv2.destroyAllWindows()

```

#-----RESTA

```
resta1=cv2.subtract(img1,img2)
```

```
cv2.imwrite("Resta1.png",resta1)
```

```
resta1 = np.concatenate((img1, resta1, img2), axis=1)
```

```
resta2=img1-img2
```

```
resta2 = np.concatenate((img1, resta2, img2), axis=1)
```

```
resta3=np.array(img1)-np.array(img2)
```

```
resta3=np.concatenate((img1, resta3, img2), axis=1)
```

```
resta=np.concatenate((resta1, resta2, resta3), axis=0)
```

```
cv2.imshow("Resta",resta)
```

```
cv2.imwrite("Resta.png",resta)
```

```
cv2.waitKey(3000)
```

```
cv2.destroyAllWindows()
```

#-----MULTIPLICACIÓN

```
mult1=cv2.multiply(img1,img2)
```

```
cv2.imwrite("Multiplicacion1.png",mult1)
```

```
mult1= np.concatenate((img1, mult1, img2), axis=1)
```

```
mult2=img1*img2
```

```
mult2= np.concatenate((img1, mult2, img2), axis=1)
```

```
mult=np.concatenate((mult1, mult2), axis=0)
```

```
cv2.imshow("Multiplicacion",mult)
```

```
cv2.imwrite("Multiplicacion.png",mult)
```

```
cv2.waitKey(3000)
```

```
cv2.destroyAllWindows()
```

#-----DIVISIÓN

```

div1=cv2.divide(img1,img2)
cv2.imwrite("Division1.png",div1)
div1= np.concatenate((img1, div1, img2), axis=1)

div2=img1/img2
div2= np.concatenate((img1, div2, img2), axis=1)

div=np.concatenate((div1, div2),axis=0)
cv2.imshow("Division",div)
cv2.imwrite("Division.png",div1)
cv2.waitKey(3000)
cv2.destroyAllWindows()

#-----LOGARÍTMO NATURAL
c = 255 / np.log(1 + np.max(img1))
log11 = c * (np.log(img1 + 1))
log11 = np.array(log11, dtype = np.uint8)
cv2.imwrite("LogaritmoNatural1.png",log11)
c = 255 / np.log(1 + np.max(img2))
log12 = c * (np.log(img2 + 1))
log12 = np.array(log12, dtype = np.uint8)
cv2.imwrite("LogaritmoNatural2.png",log12)

log1=np.concatenate((img1, log11, log12, img2), axis=1)

log21=np.log(img1)
log21 = np.asarray(log21, dtype="uint8")
log22=np.log(img2)
log22 = np.asarray(log22, dtype="uint8")
log2=np.concatenate((img1, log21, log22, img2), axis=1)

log=np.concatenate((log1, log2), axis=0)

```

```
cv2.imwrite("LogaritmoNatural.png",log)
```

```
cv2.imshow("Logaritmo Natural",log)
```

```
cv2.waitKey(3000)
```

```
cv2.destroyAllWindows()
```

```
#-----RAÍZ
```

```
sqrt21=img1**(0.5)
```

```
cv2.imwrite("RaizCuadrada1.png",sqrt21)
```

```
sqrt22=img2**(0.5)
```

```
cv2.imwrite("RaizCuadrada2.png",sqrt22)
```

```
sqrt2=np.concatenate((img1,sqrt21,sqrt22,img2),axis=1)
```

```
sqrt31=np.sqrt(img1)
```

```
sqrt31 = np.asarray(sqrt31, dtype="uint8")
```

```
sqrt32=np.sqrt(img2)
```

```
sqrt32 = np.asarray(sqrt32, dtype="uint8")
```

```
sqrt3=np.concatenate((img1,sqrt31,sqrt32,img2),axis=1)
```

```
sqrt=np.concatenate((sqrt2, sqrt3),axis=0)
```

```
cv2.imshow("Raiz Cuadrada",sqrt)
```

```
cv2.imwrite("Raiz Cuadrada.png",sqrt)
```

```
cv2.waitKey(3000)
```

```
cv2.destroyAllWindows()
```

```
#-----POTENCIA
```

```
pot11=cv2.pow(img1,2)
```

```
cv2.imwrite("Potencia1.png",pot11)
```

```
pot12=cv2.pow(img2,2)
```

```
cv2.imwrite("Potencia2.png",pot12)
```

```
pot1=np.concatenate((img1,pot11,pot12,img2),axis=1)
```

```
pot21=img1**2
pot22=img2**2
pot2=np.concatenate((img1,pot21,pot22,img2),axis=1)
```

```
pot31=np.power(img1, 2)
pot32=np.power(img2, 2)
pot3=np.concatenate((img1,pot31,pot32,img2),axis=1)
```

```
pot=np.concatenate((pot1,pot2,pot3),axis=0)
cv2.imshow("Potencia",pot)
cv2.imwrite("Potencia.png",pot)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----DERIVADA

```
der1=cv2.Sobel(img1,ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5)
cv2.imwrite("Derivada1.png",der1)
der2=cv2.Sobel(img2,ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5)
cv2.imwrite("Derivada2.png",der2)
der=np.concatenate((img1,der1,der2,img2),axis=1)
```

```
cv2.imshow("Derivada",der)
cv2.imwrite("Derivada.png",der)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----CONJUNCIÓN

```
and1=cv2.bitwise_and(img1,img2)
cv2.imwrite("Conjuncion1.png",and1)
and1=np.concatenate((img1,and1,img2),axis=1)
```

```
cv2.imshow("Conjuncion",and1)
cv2.imwrite("Conjuncion.png",and1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----DISYUNCIÓN

```
or1=cv2.bitwise_or(img1,img2)
cv2.imwrite("Disyuncion1.png",or1)
or1=np.concatenate((img1,or1,img2),axis=1)
```

```
cv2.imshow("Disyuncion",or1)
cv2.imwrite("Disyuncion.png",or1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----NEGACIÓN

```
not11=cv2.bitwise_not(img1)
cv2.imwrite("Negacion1.png",not11)
not12=cv2.bitwise_not(img2)
cv2.imwrite("Negacion2.png",not12)
not1=np.concatenate((img1,not11,not12,img2),axis=1)
```

```
cv2.imshow("Negacion",not1)
cv2.imwrite("Negacion.png",not1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----TRASLACIÓN

```
ancho=img1.shape[1]
alto=img1.shape[0]
M1=np.float32([[1,0,50],[0,1,25]])
M2=np.float32([[1,0,-50],[0,1,25]])
```



```
tras11=cv2.warpAffine(img1,M1,(ancho,alto))
cv2.imwrite("Traslacion1.png",tras11)
tras12=cv2.warpAffine(img2,M2,(ancho,alto))
cv2.imwrite("Traslacion2.png",tras12)
tras1=np.concatenate((img1,tras11,tras12,img2),axis=1)
```

```
cv2.imshow("Traslacion",tras1)
cv2.imwrite("Traslacion.png",tras1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----ESCALADO

```
porc_escalado=50
ancho2=int(img1.shape[1]*porc_escalado/100)
alto2=int(img1.shape[0]*porc_escalado/100)
escal1=(ancho2,alto2)
escal11=cv2.resize(img1,escal1)
cv2.imwrite("Escalado1.png",escal11)
escal12=cv2.resize(img2,escal1)
cv2.imwrite("Escalado2.png",escal12)
escal1=np.concatenate((escal11,escal12),axis=1)
```

```
cv2.imshow("Escalado",escal1)
cv2.imwrite("Escalado.png",escal1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----ROTACIÓN

```
rot=cv2.getRotationMatrix2D((ancho/2,alto/2),45,1)
rot11=cv2.warpAffine(img1,rot,(ancho,alto))
cv2.imwrite('Rotacion1.png',rot11)
rot12=cv2.warpAffine(img2,rot,(ancho,alto))
```

```
cv2.imwrite('Rotacion2.png',rot12)
rot1=np.concatenate((img1,rot11,rot12,img2),axis=1)
```

```
cv2.imshow('Rotacion',rot1)
cv2.imwrite('Rotacion.png',rot1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----TRASLACIÓN A FIN

```
rows,cols = img1.shape[:2]
pts1 = np.float32([[50,50],[200,50],[50,200]])
pts2 = np.float32([[10,100],[200,50],[100,250]])
M = cv2.getAffineTransform(pts1,pts2)
trasafin11 = cv2.warpAffine(img1,M,(rows,cols))
trasafin11 = cv2.resize(trasafin11,(cols,rows))
cv2.imwrite('TraslacionAFin1.png',trasafin11)
```

```
rows,cols = img2.shape[:2]
pts1 = np.float32([[50,50],[200,50],[50,200]])
pts2 = np.float32([[10,100],[200,50],[100,250]])
M = cv2.getAffineTransform(pts1,pts2)
trasafin12 = cv2.warpAffine(img2,M,(rows,cols))
trasafin12 = cv2.resize(trasafin12,(cols,rows))
cv2.imwrite('TraslacionAFin2.png',trasafin12)
```

```
trasafin1=np.concatenate((img1,trasafin11,trasafin12,img2),axis=1)
```

```
cv2.imshow('Traslacion A Fin',trasafin1)
cv2.imwrite('TraslacionAFin.png',trasafin1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----TRANSPUESTA

```
trans=cv2.getRotationMatrix2D((ancho/2,alto/2),90,1)
trans11=cv2.warpAffine(img1,trans,(ancho,alto))
cv2.imwrite('Transpuesta1.png',trans11)
trans12=cv2.warpAffine(img2,trans,(ancho,alto))
cv2.imwrite('Transpuesta2.png',trans12)
trans1=np.concatenate((img1,trans11,trans12,img2),axis=1)
```

```
cv2.imshow('Transpuesta',trans1)
cv2.imwrite('Transpuesta.png',trans1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----PROYECCIÓN

```
pts1=np.float32([[200, 0],[350,0],[28,200],[350,210]])
pts2=np.float32([[0,0],[480,0],[0,270],[480,270]])
M=cv2.getPerspectiveTransform(pts1,pts2)
proy11=cv2.warpPerspective(img1,M,(cols,rows))
cv2.imwrite('Proyeccion1.png',proy11)
proy12=cv2.warpPerspective(img2,M,(cols,rows))
cv2.imwrite('Proyeccion2.png',proy12)
proy1=np.concatenate((img1,proy11,proy12,img2),axis=1)
cv2.imshow('Proyeccion',proy1)
cv2.imwrite('Proyeccion.png',proy1)
cv2.waitKey(3000)
cv2.destroyAllWindows()
```

#-----

```
os.system("cls")
```

Resultados de Impresión:

A continuación, se muestran algunos de los resultados:

Suma:



Resta:



Multiplicación:



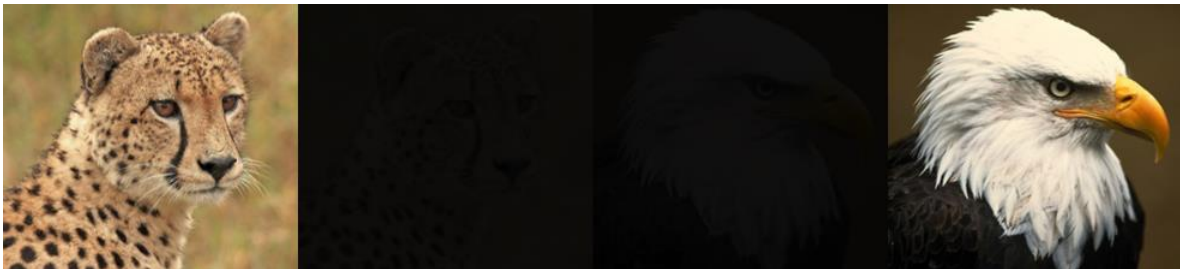
División:



Logaritmo Natural:



Raíz Cuadrada:



Potencia:



Derivada:



Conjunción:



Disyunción:



Negación:



Traslación:



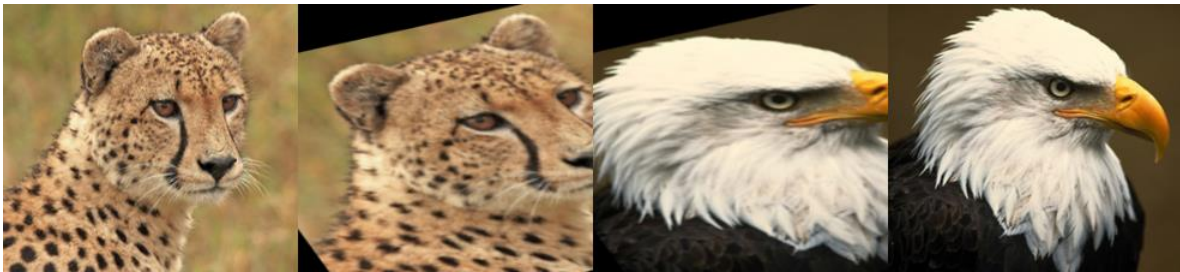
Escalado:



Rotación:



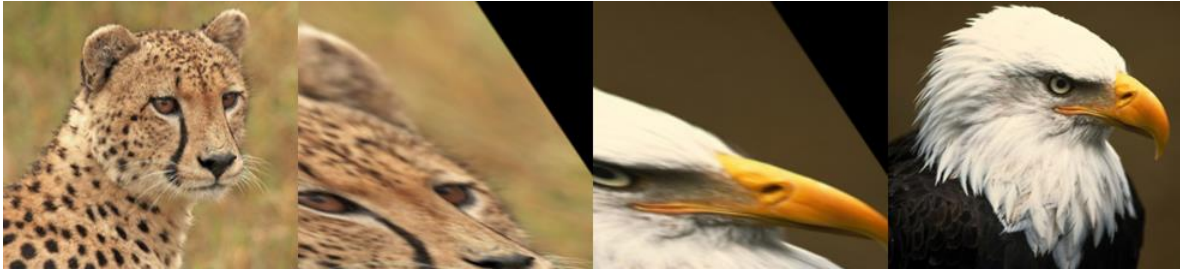
Traslación a fin:



Transpuesta:



Proyección:



Conclusiones:

Algo que me pareció interesante fue el hecho de aprender a realizar todas las operaciones ya que supongo que nos serán de gran utilidad para prácticas posteriores. Asimismo, lamentablemente no pude encontrar métodos distintos para realizar algunas operaciones, me resultó complicado encontrar información al respecto y su aplicación muchas veces no me dio los resultados esperados.

Sin embargo, lo que más me gustó fue poder visualizar como interactúan las imágenes entre ellas en cada una de las operaciones que, al estar investigando, pude observar que algunas de estas son utilizadas en prácticas más avanzadas e visión artificial.

Enlace GitHub:

<https://github.com/IsaacGutierrezCETI/Practica-2-VA.-Operaciones-con-imagenes>