

UAS
PENGOLAHAN CITRA DAN VISI KOMPUTER



NAMA : ARLAN ADY PRATAMA
NIM : 1941720042

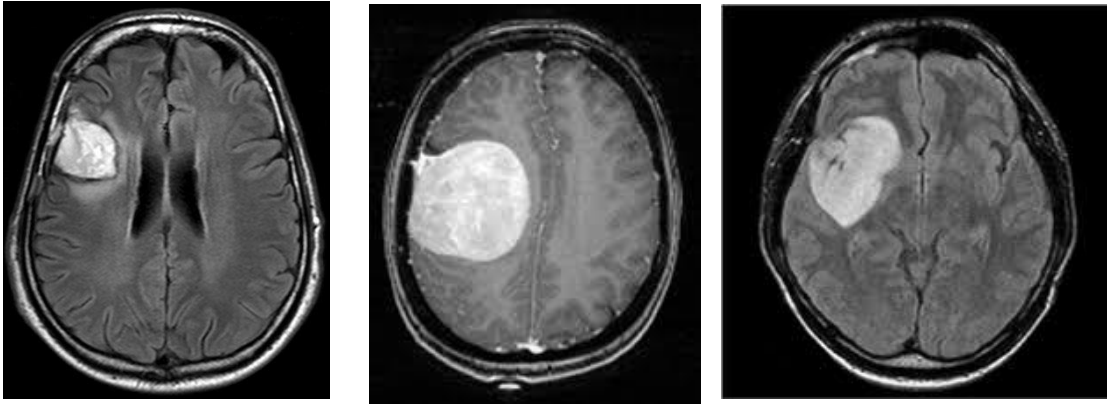
JURUSAN TEKNOLOGI INFORMASI
D-IV TEKNIK INFORMATIKA
POLITEKNIK NEGERI MALANG
2022

Subject

Menentukan letak tumor yang ada pada otak menggunakan Template Matching.

Dataset

Data yang digunakan yaitu gambar tumor yang ada di otak, seperti gambar di bawah ini.



Metode

Template Matching

Pada bagian ini kita akan menggunakan salah satu fungsi OpenCV yang digunakan untuk meringkas beberapa langkah, yaitu deteksi keypoint, matching dan deteksi tepi untuk mendeteksi posisi tumor dari serangkaian gambar. Algoritma ini disebut sebagai 'Template Matching'. OpenCV menyediakan fungsi `cv2.matchTemplate()` untuk keperluan Template Matching ini dengan berbagai metode deteksi berdasarkan tingkat kemiripan antara `template` atau gambar yang dicari dengan objek pada gambar yang tersedia, misalnya dengan menghitung rerata korelasi antar nilai piksel pada template tersebut dengan objek yang dicari. Latihan berikut menunjukkan aplikasi dari berbagai metode tersebut untuk mencari gambar yang diberikan.

Sourecode

1. Menyambungkan ke drive untuk akses gambar.

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

2. Import library

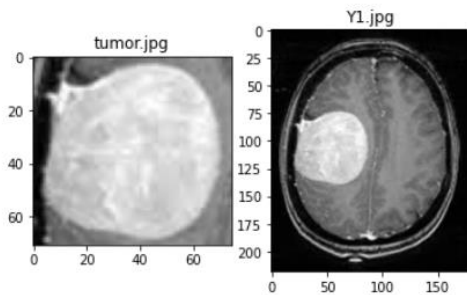
```
from matplotlib import pyplot as plt
import cv2
```

3. Panggil dan konversi warna agar sesuai dengan Matplotlib

```
parkir_kosong = cv2.imread('/content/drive/MyDrive/tugas visi data komputer/uas_car/tumor.jpg')  
parkir_kosong = cv2.cvtColor(parkir_kosong, cv2.COLOR_BGR2RGB)
```

```
parkir_ad = cv2.imread('/content/drive/MyDrive/tugas visi data komputer/uas_car/Y1.jpg')  
parkir_ad = cv2.cvtColor(parkir_ad, cv2.COLOR_BGR2RGB)
```

```
[ ] plt.subplot(121),plt.imshow(parkir_kosong), plt.title('tumor.jpg')  
plt.subplot(122),plt.imshow(parkir_ad), plt.title('Y1.jpg')  
plt.show()
```



4. ukuran template. ukuran ini akan digunakan untuk menggambar kotak

```
w, h = template1.shape[::-1]
```

5. menggunakan metode COEFF-NORMALIZED

```
res = cv2.matchTemplate(img_gray,template1,cv2.TM_CCOEFF_NORMED)
```

6. Nilai threshold atau ambang batas deteksi kemiripan titik.

7. Lakukan eksperimen dengan merubah nilai ini

```
threshold = 0.7  
loc = np.where(res >= threshold)
```

8. membuat array kosong untuk menyimpan lokasi-lokasi dari hasil deteksi

```
lspoint=[]
lspoint2=[]
count = 0 # untuk menyimpan jumlah matching yang ditemukan
for pt in zip(*loc[::-1]):
    ## jika sudah ada, skip lokasi tersebut
    if pt[0] not in lspoint and pt[1] not in lspoint2:
        ## gambar persegi warna kuning dengan ketebalan dua poin
        cv2.rectangle(img_rgb, pt, (pt[0] + w, pt[1] + h), (0,255,255), 2)
        for i in range(((pt[0])-9), ((pt[0])+9),1):
            ## tambahkan koordinat x ke list
            lspoint.append(i)
        for k in range(((pt[1])-9), ((pt[1])+9),1):
            ## tambahkan koordinat y ke list
            lspoint2.append(k)
        count+=1 ### berapa jumlah matching yang ditemukan?
    else:
        continue
print ("Jumlah objek ditemukan ", count)
```

9. tampilkan dengan imshow

