Semantic Image Classification and Segmentation of Cancer MRI Images

Project Implementation ITCS 5122

GROUP MEMBER:

LIKHITH AALLA 801075869

ABISHEK SURYA 800985589

SHARMISTHA HALDAR 801061221

YASHWANTH TADIMALLA 801083447

ABSTRACT

The biggest problem the Health care Industry is facing in today's world is the low rate early detection of Cancer. To help better this situation we developed a detection algorithm which would detect the type of cancer based on the CT Scan of the person and help with the area that is affected identification. This would not only help patients but also the health care personnel in easier and immediate reaction at a necessary pace when dealing with large volumes of data.

To implement this mechanism, we are using SVM classifier and Skimage (Python Library) as a platform to train our system for Semantic image Classification and therefore enhance the overall performance of the system. The final released system would work with utmost accuracy and precision possible within its limitations of available hardware and software.

The four main categories of cancer that we are considering for this system are Glioblastoma Multiforme (Brain), Squamous lung cancer (Lung), Parathyroid cystic (Neck) and Clear cell Kidney carcinoma (kidney) and related datasets of anonymous patients.

This system when used at its maximum potential would significantly reduce the risk factor of cancer spreading to large no: of cells within the body and it can lead to any other organs to fail like cardiovascular (CV) diseases. Early detection result in availability of more time for the doctors to diagnose and provide required treatment and other related medication.

INTRODUCTION

In simple terms CANCER is a disease in which abnormal cells divide uncontrollably and destroy body tissue causing in bodily function and organs failure leading to fatality. Possible signs and symptoms include a lump, abnormal bleeding, prolonged cough, unexplained weight loss and a change in bowel movements. While these symptoms may indicate cancer, they may have other causes. Over 100 types of cancers affect humans.

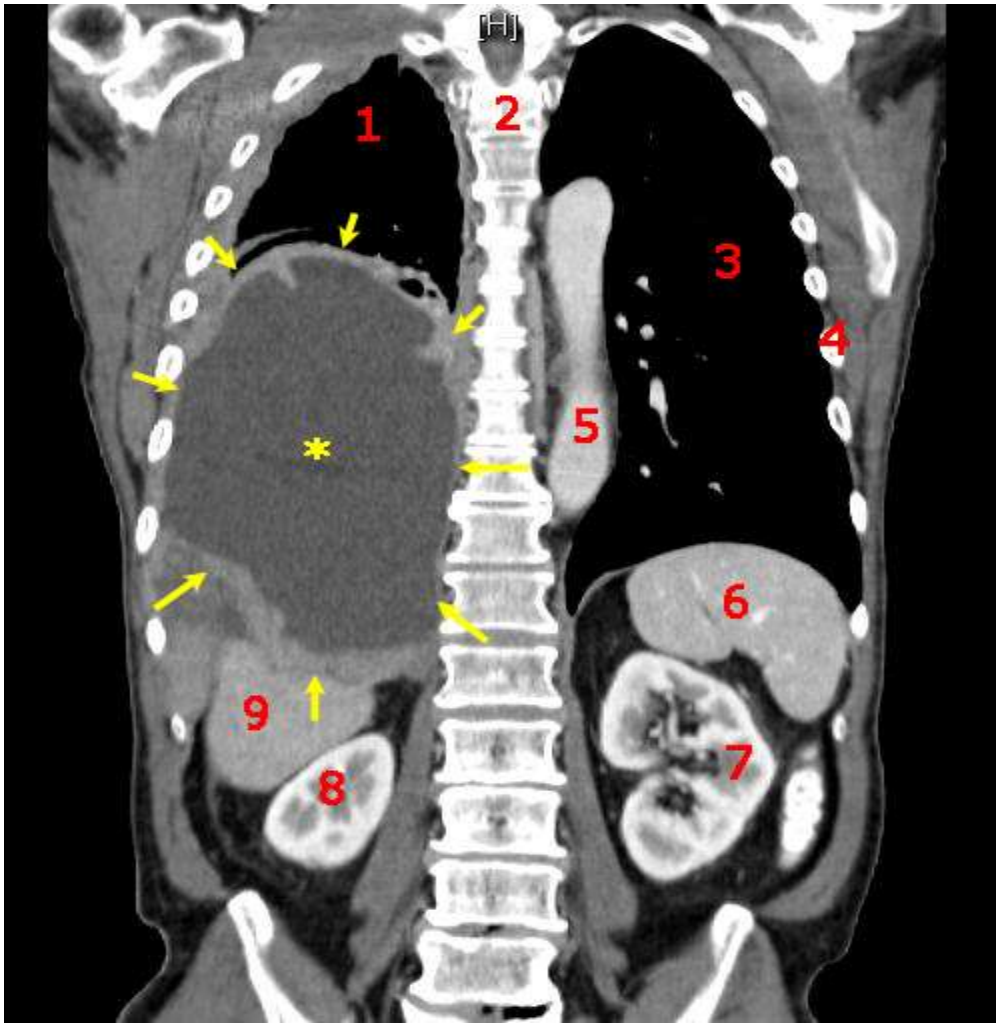


Image showing all the possible forms of cancer possible in a human torso CT Scan.
Legend: the malignant mesothelioma is indicated by yellow arrows, the central pleural effusion is marked with a yellow star. (1) right lung, (2) spine, (3) left lung, (4) ribs, (5) aorta, (6) spleen, (7) left kidney, (8) right kidney, (9) liver.

Although cancer can spread to any organ and any node, some of them more common than other. The major categories are Bone, Brain, Lung, Kidney/Liver, and Neck (Thyroid).

Types of cancer and pre-requisite information about the datasets:

Glioblastoma Multiforme (GBM) is a fast-growing type of malignant brain tumor that is the most common brain tumor in adults. In 2010, more than 22,000 Americans were estimated to have been diagnosed and 13,140 were estimated to have died from brain and other nervous system cancers.1 GBM accounts for about 15 percent of all brain tumors and occurs in adults between the ages of 45 to 70 years.2 Patients with GBM have a poor prognosis and usually survive less than 15 months following diagnosis. Currently there are no effective long-term treatments for this disease.

Small cell lung cancer (SCLC) or Non-small cell lung cancer (NSCLC): Worldwide, lung cancer has one of the highest incidence rates and the highest mortality rate of any cancer. lung cancer has the lowest 5-year relative survival rate. Lung cancer is classified as one of two main histologic types: small cell lung cancer (SCLC) or non-small cell lung cancer (NSCLC).
NSCLC is the more common type, found in approximately 85% to 90% of patients with lung cancer, while SCLC is found in approximately 10% to 15% of patients. NSCLC can be further divided into different subtypes, including nonsquamous and squamous NSCLC. With an estimated 450,000 cases worldwide in 2012, **squamous NSCLC** is the second most common subtype of NSCLC.

Chromophobe renal cell carcinoma: is a type of kidney cancer. This type of cancer forms in the cells lining the small tubules in the kidney. These small tubules help filter waste from the blood, making urine. All types of kidney cancer are different, making it even more important to characterize each one.[1] In 2012, it is estimated that there will be 64,770 new cases of kidney cancer and 13,570 deaths as a result of this disease. Chromophobe kidney cancer accounts for five percent of these cancer cases.

Thyroid cancer: According to the National Cancer Institute, there are over 56,000 new cases of thyroid cancer in the US each year, and the majority of those diagnosed are papillary thyroid cancer—the most common type of thyroid cancer. Females are more likely to have thyroid cancer at a ratio of 3:1. Thyroid cancer can occur in any age group, although it is most common after age 30, and its aggressiveness increases significantly in older patients. Approximately 1.2 percent of all men and women will be diagnosed with thyroid cancer during the course of their lifetime.

DATASET

TCIA is a service which de-identifies and hosts a large archive of medical images of cancer accessible for public download. The data are organized as "Collections", typically patients related by a common disease (e.g. lung cancer), image modality (MRI, CT, etc) or research focus. DICOM is the primary file format used by TCIA for image storage. Supporting data related to the images such as patient outcomes, treatment details, genomics, pathology, and expert analyses are also provided when available.

The Cancer Imaging Program (CIP) is one of four Programs in the Division of Cancer Treatment and Diagnosis (DCTD) of the National Cancer Institute.

http://www.cancerimagingarchive.net

https://www.cancer.gov/about-nci/organization/ccg/research/computational-genomics/gdc/tcga-infographic

CT SCAN

Computed Tomography, commonly known as CT or CAT scanning, is a non-invasive diagnostic tool. CT uses a specialized form of X-ray, coupled with computer technology, to produce cross-sectional images (slices) of soft tissue, organs, bone and blood vessels in any area of the body.

WHY IS CT USED?

CT scans are used to check the size and structure of an organ or other soft tissue and determine if it's infected, solid or filled with fluid. The scans are used to diagnose tumors, cancers, spinal injuries, heart disease, vascular conditions, brain disorders and various other abnormalities within the body. CT scans also are used to rapidly diagnose traumatic injuries and to guide several minimally invasive procedures such as needle biopsies, catheter placement, fluid drainage and duct and vessel stenting.

HOW DOES CT WORK?

CT uses X-rays to detect and record the amount of radiation absorbed by different tissues. During a CT scan, an X-ray tube focuses a precise beam of energy on a section of the body. A computer analyzes the readings from X-rays taken at thousands of different points and converts the information into images radiologists and other doctors use to analyze internal organs and tissue.

IMPLEMENTATION

For performing data analysis languages like R, Python can be used. The dataset that is considered for cancer image classification is large set, Python would be best choice as large datasets can be dealt easily by it. Ipython or Jupyter notebook is an interactive IDE to perform python operations. We can perform command line operations using jupyter.

The procedure followed:

1) DATASET SELECTION

   Choosing which dataset to work on and analyze the final result that can be estimated from the various kinds of datasets available.

2) FINALIZING CANCER IMAGE COLLECTIONS

   Finalizing on the different types of cancer datasets that are available at our dispense and selecting few among them.

3) DATA PREPROCESSING

   Data Cleaning and formatting the dataset according to the requirements of the system.

4) FEATURE EXTRACTION

   We identified the important features in each image based on which we classified the cancer images.

   We extracted following features:

   - Area
   - Perimeter
   - Convex_area
   - Mean_intensity
   - Max_intensity
   - Solidity
   - Orientation
   - Eccentricity
   - Equivalent_diameter
   - Euler_number
   - Extent

- Filled_area
- Major_axis
- Minor_axis

5) VISUALIZING DATA DISTRIBUTION

The distribution of a statistical data set (or a population) is a listing or function showing all the possible values (or intervals) of the data and how often they occur. When a distribution of categorical data is organized, you see the number or percentage of individuals in each group.

6) CLASSIFIER TRAINING

- Image classification uses the quantitative spectral information contained in an image, which is related to the composition or condition of the target surface. There are several core principles of image analysis that pertain specifically to the extraction of information and features from remotely sensed data.
- The algorithm used was SVM, we compared the features of the images to identify the different parts of the body and classify the type of cancer present.
    I. Importing libraries
    II. Importing the Dataset
    III. Exploratory Data Analysis
    IV. Data Preprocessing
    V. Training the Algorithm
    VI. Making Predictions
    VII. Evaluating the Algorithm
    VIII. The Evaluation Results

7) CANCER IMAGE PREDICTION

After training the classifier we have used 30 images to predict the type of cancer.

We got the following readings using confusion matrix and classification report.

- Precision of 85%
- Recall of 73%
- F1-score of 73%

Implementation Screen Shots:

Showing the Length of the Training Dataset used.

In [136]:
```
%matplotlib inline
import matplotlib
matplotlib.rcParams['image.interpolation'] = 'nearest'
import numpy as np
import matplotlib.pyplot as plt
```

In [137]:
```
from skimage import io
ic = io.ImageCollection('FINAL_TRAINING_DATA_SET/*.jpg')
```

In [138]:
```
len(ic)
```

Out[138]: 352

Features selected for Training the System.

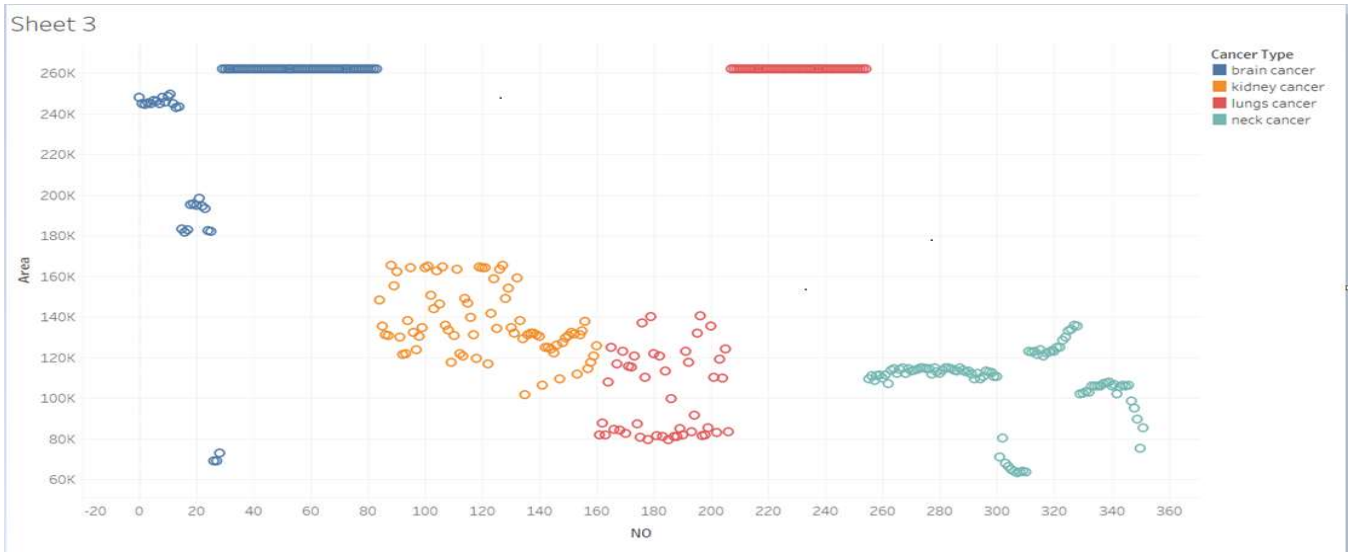
In [143]:
```
data
```

Out[143]:

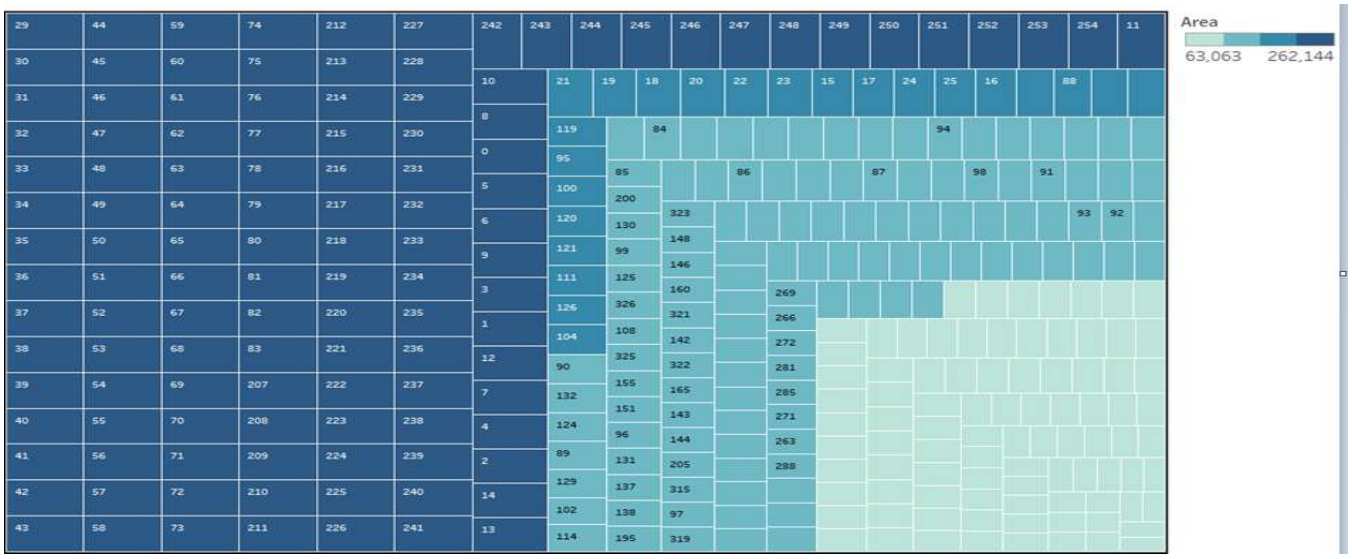
	Area	Perimeter	Convex_area	Mean_intensity	Max_intensity	Solidity	Orientation	Ecentricity	Equivalent_diameter	Euler_number	Extent	Filled
0	248040.0	2020.818326	250347.0	0.118741	1.000000	0.990785	0.056273	0.267082	561.973609	1.0	0.978693	248
1	245074.0	2020.374675	247200.0	0.111002	1.000000	0.991400	0.004282	0.268391	558.603534	1.0	0.980861	245
2	244409.0	2020.303607	246865.0	0.115094	0.996078	0.990051	-0.014933	0.252621	557.845143	1.0	0.978199	244
3	245214.0	2017.569589	247449.0	0.128792	1.000000	0.990968	0.010156	0.256561	558.763064	1.0	0.979414	245
4	244751.0	2008.818326	246945.0	0.117734	0.992157	0.991115	-0.043728	0.254744	558.235301	1.0	0.977565	244
5	246292.0	2008.770599	248305.0	0.137100	1.000000	0.991893	0.083597	0.265703	559.989923	1.0	0.983720	246
6	246124.0	2026.617316	249747.0	0.118072	1.000000	0.985493	0.024471	0.252858	559.798901	1.0	0.971133	246
7	244923.0	2017.883297	247468.0	0.125802	0.992157	0.989716	0.009468	0.251387	558.431418	1.0	0.978252	244
8	248120.0	2008.048773	249964.0	0.105714	1.000000	0.992623	0.098374	0.259873	562.064228	1.0	0.979009	248
9	245740.0	2007.291414	247207.0	0.105764	1.000000	0.994066	0.008756	0.258889	559.362035	1.0	0.981515	245
10	248551.0	2034.475180	250651.0	0.083834	1.000000	0.991622	0.085061	0.263184	562.552186	1.0	0.980709	248
11	249559.0	2014.285317	251437.0	0.091547	0.992157	0.992531	0.049245	0.269472	563.691749	1.0	0.984687	249
12	244980.0	2012.682287	247233.0	0.142212	0.988235	0.990887	0.090509	0.286475	558.496395	1.0	0.980485	244
337	107035.0	3437.803353	147541.0	0.356726	1.000000	0.610274	0.004143	0.952317	441.094524	0.0	0.978582	130
338	107394.0	3399.395237	146531.0	0.345348	0.980392	0.605630	0.002451	0.954493	440.371182	0.0	0.966821	128
339	107752.0	3320.318072	144820.0	0.345045	0.979085	0.622715	0.008334	0.956392	441.727735	0.0	0.984403	125
340	106091.0	3259.584053	140798.0	0.345527	0.951634	0.630074	0.009162	0.961176	437.584278	0.0	0.992358	118
341	106644.0	3332.667314	143848.0	0.350901	0.954248	0.616160	0.008376	0.958824	437.410579	0.0	0.972665	122
342	102124.0	2525.528137	128886.0	0.325157	0.909804	0.629657	0.015636	0.968696	430.333678	2.0	0.996847	102
343	105411.0	2550.794119	132082.0	0.322680	0.928105	0.612599	0.018795	0.967678	435.215833	2.0	0.994971	106
344	106210.0	3149.879472	139032.0	0.325090	0.924183	0.603471	0.011987	0.963122	436.326729	0.0	0.986025	115
345	105817.0	3073.465258	137551.0	0.321046	0.921569	0.601025	0.015370	0.964796	435.965237	0.0	0.990301	112
346	106317.0	2888.179941	134513.0	0.324982	0.935948	0.631342	0.015783	0.966565	437.098833	0.0	1.001816	109
347	98692.0	2496.498700	124752.0	0.319938	0.908497	0.604718	0.017814	0.969609	424.380607	2.0	0.987936	99
348	95141.0	2487.498700	120445.0	0.318287	0.917647	0.614850	0.015199	0.970098	417.307548	2.0	0.963502	95
349	89724.0	2792.912914	127447.0	0.538307	1.000000	0.704010	-0.008127	0.881900	337.994297	-2.0	0.689930	127
350	75298.0	2987.334270	127455.0	0.564394	1.000000	0.590781	-0.011964	0.881390	309.632671	-1.0	0.581290	96
351	85498.0	2870.333224	127268.0	0.558081	1.000000	0.671795	-0.010939	0.880380	329.938532	-2.0	0.660033	127

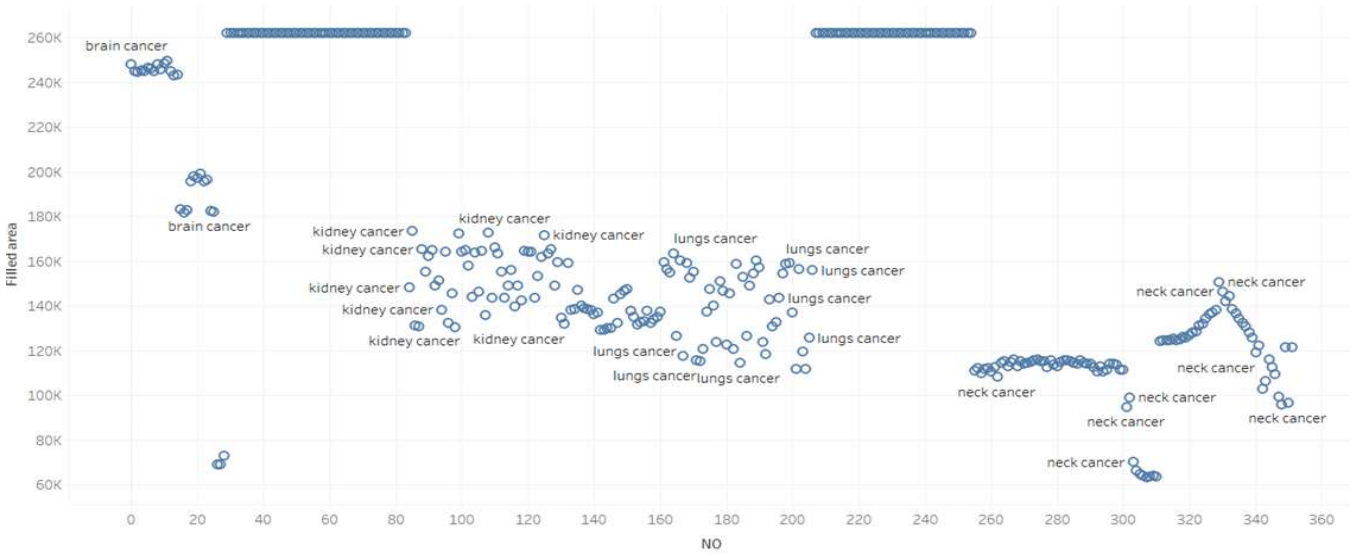
352 rows × 14 columns

Visualization of data distribution (Tableau)



Area occupied by different types





Feature values for Predicted images.

```
In [151]: data_predict.head()
```

Out[151]:

	Area	Perimeter	Convex_area	Mean_intensity	Max_intensity	Solidity	Orientation	Ecentricity	Equivalent_diameter	Euler_number	Extent	Filled_a
0	183891.0	1743.764502	184914.0	0.161281	0.996078	0.994468	1.567289	0.639682	483.877353	1.0	0.981003	1838
1	182283.0	1733.622366	183482.0	0.179323	1.000000	0.993465	1.569285	0.641850	481.757121	1.0	0.981546	1822
2	182931.0	1743.249783	184029.0	0.168321	1.000000	0.994034	1.567897	0.640573	482.612664	1.0	0.982444	1829
3	262144.0	2044.000000	262144.0	0.230589	1.000000	1.000000	0.785398	0.000000	577.730134	1.0	1.000000	2621
4	262144.0	2044.000000	262144.0	0.227378	1.000000	1.000000	0.785398	0.000000	577.730134	1.0	1.000000	2621

Final Predicted results of the system.

```
In [152]: result = clf.predict(data_predict[:])
          result
```

Out[152]:

```
array(['neck cancer', 'neck cancer', 'neck cancer', 'brain cancer',
       'brain cancer', 'brain cancer', 'brain cancer', 'brain cancer',
       'kidney cancer', 'kidney cancer', 'kidney cancer', 'kidney cancer',
       'neck cancer', 'neck cancer', 'neck cancer', 'neck cancer',
       'neck cancer', 'lungs cancer', 'lungs cancer', 'lungs cancer',
       'lungs cancer', 'lungs cancer', 'neck cancer', 'neck cancer',
       'neck cancer', 'neck cancer', 'neck cancer', 'neck cancer',
       'neck cancer', 'neck cancer'], dtype=object)
```

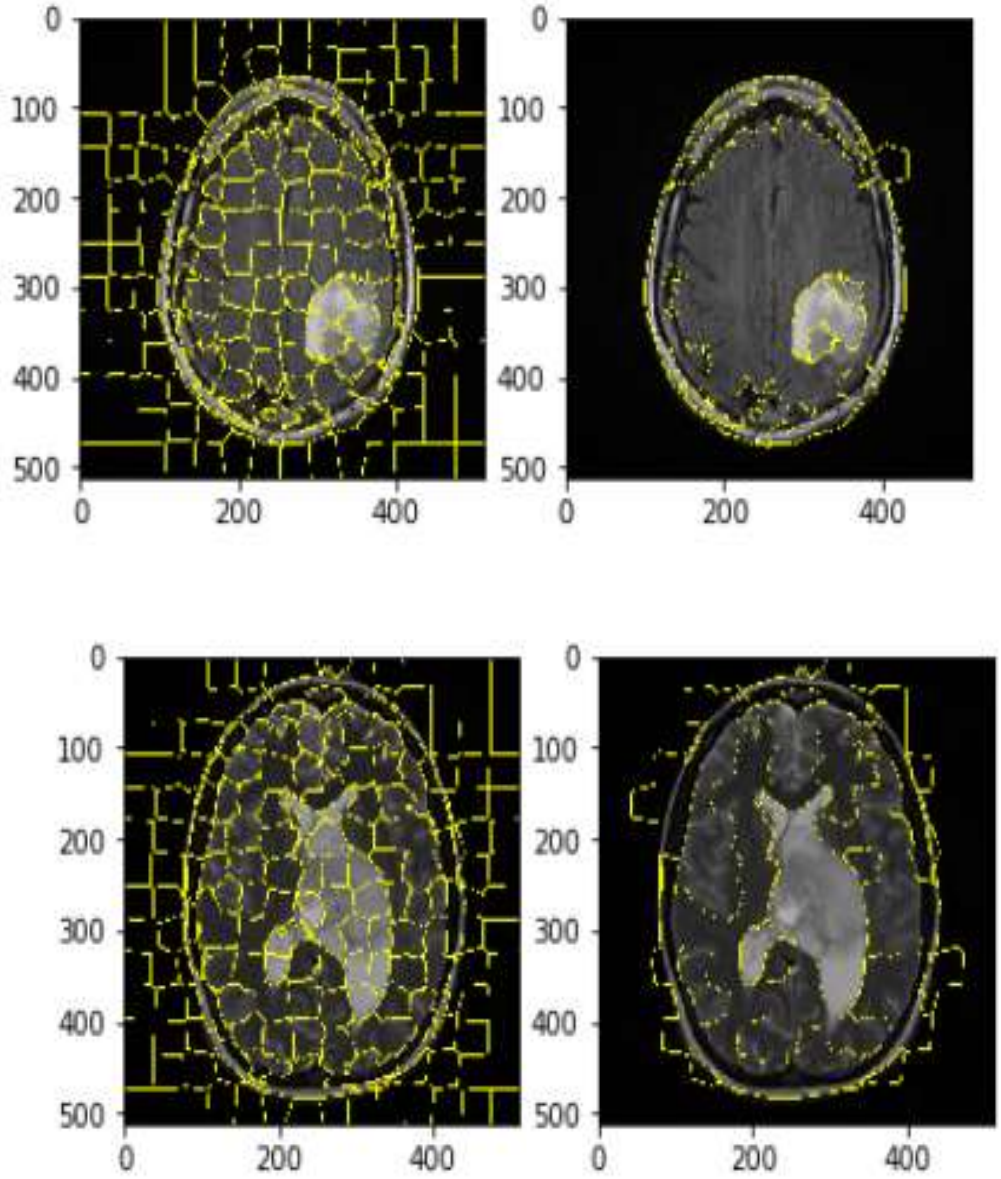
```
[[5 0 0 0]
 [0 4 0 0]
 [0 0 5 0]
 [3 0 5 8]]
              precision    recall  f1-score   support

 brain cancer       0.62      1.00      0.77         5
kidney cancer       1.00      1.00      1.00         4
 lungs cancer       0.50      1.00      0.67         5
  neck cancer       1.00      0.50      0.67        16

  avg / total       0.85      0.73      0.73        30
```

SEGMENTATION

SLIC (Simple Linear Iterative Clustering) is a segmentation algorithm which clusters pixels in both space and color. Therefore, regions of space that are similar in color will end up in the same segment.

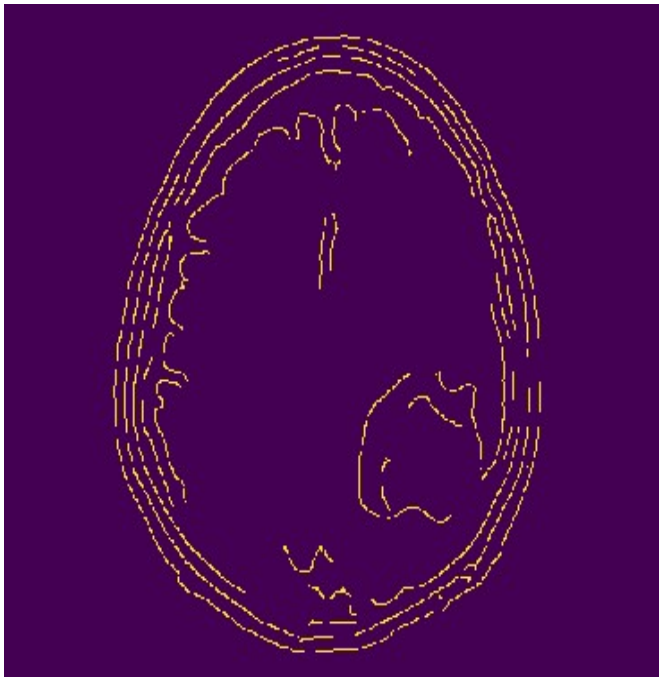
SLIC is a superpixel algorithm, which segments an image into patches (superpixels) of neighboring pixels with a similar color. SLIC also works in the Lab colorspace. The compactness parameter controls the relative importance of the distance in image- and color-space.

After the super-pixel segmentation (which is also called oversegmentation, because we end up with more segments that we want to), we can add a second clustering step to join superpixels belonging to the same region.



Canny Edge Detection:

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.



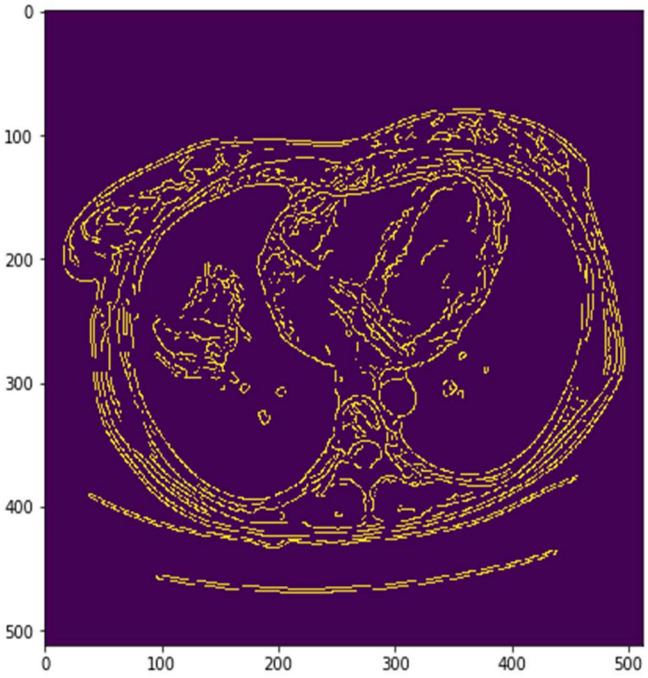


Image Classification Using Threshold:

Image binarization is a common operation. For grayscale images, finding the best threshold for binarization can be a manual operation. Alternatively, algorithms can select a threshold value automatically; which is convenient for computer vision, or for batch-processing a series of images.

Otsu algorithm is the most famous thresholding algorithm. It maximizes the variance between the two segmented groups of pixels. Therefore, it is can be interpreted as a clustering algorithm. Samples are pixels and have a single feature, which is their grayscale value.

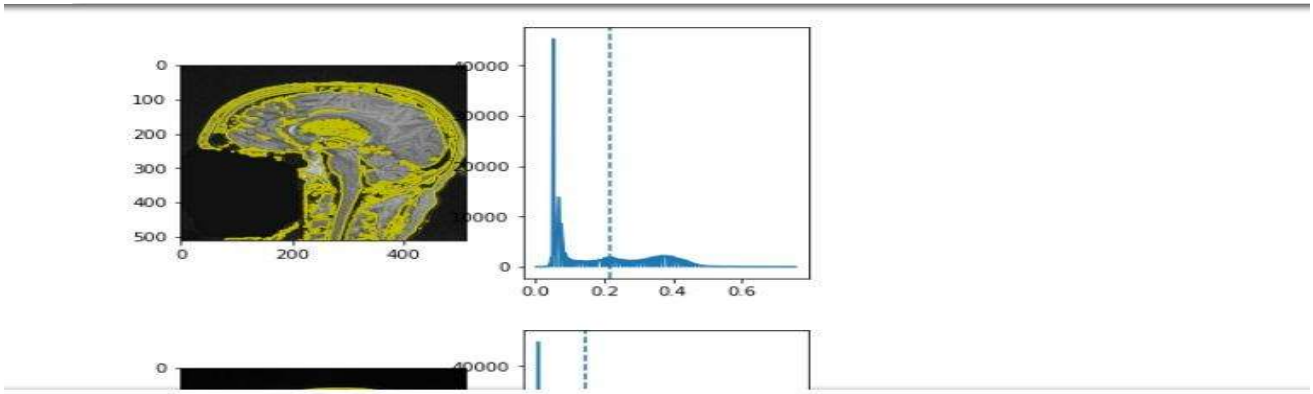


Image Cleaning

If we use the denoising + thresholding approach, the result of the thresholding is not completely what we want: small objects are detected, and small holes exist in the objects. Such defects of the segmentation can be amended, using the knowledge that no small holes should exist, and that blobs have a minimal size.

Utility functions to modify binary images are found in the morphology submodule. Although mathematical morphology encompasses a large set of possible operations, we will only see here how to remove small objects.

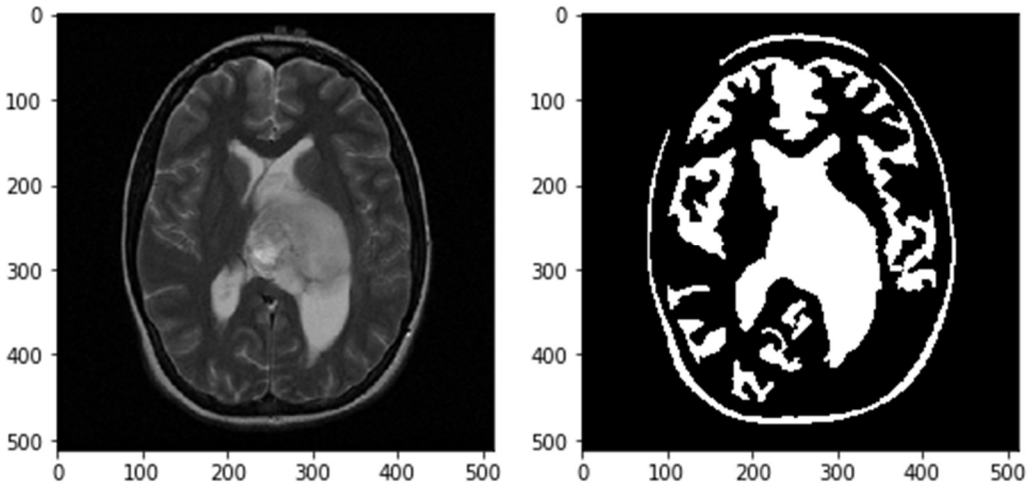


Image Classification Using K-Means:

K-means method has been shown to be effective in producing good clustering results for many practical applications. K-means method is well known for its relatively simple implementation and decent results. However, a direct algorithm of k-means method requires time proportional to the product of the number of documents (vectors) and number of clusters per iteration. This is computationally very expensive especially for large datasets

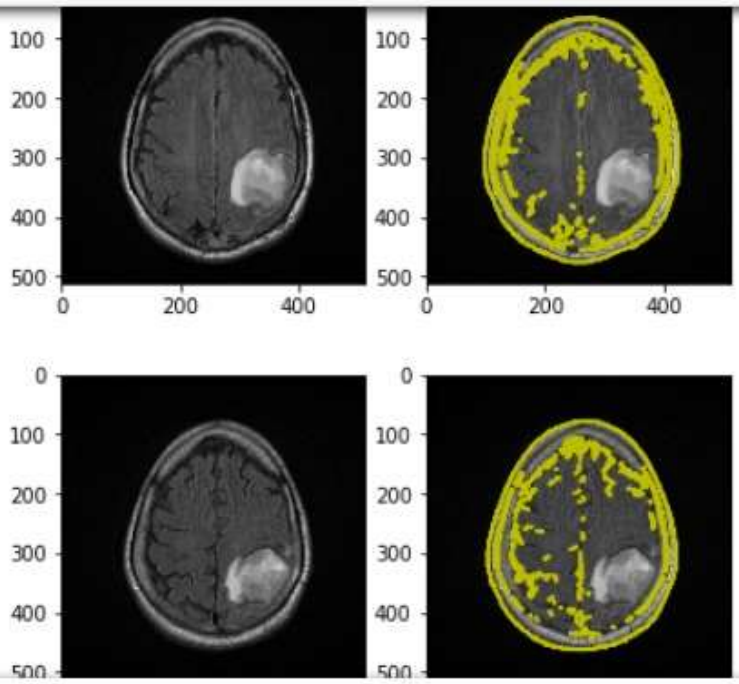
K-means Clustering

k-means clustering uses the Euclidean distance in feature space to cluster samples. If we want to cluster together pixels of similar color, the RGB space is not well suited since it mixes together information about color and light intensity. Therefore, we first transform the RGB image into Lab colorspace, and only use the color channels (a and b) for clustering.

In [23]:
```
from sklearn.cluster import KMeans
```

In [24]:
```
ic_seg_images = io.ImageCollection('Segmenting_image_data_set/*.jpg')
```

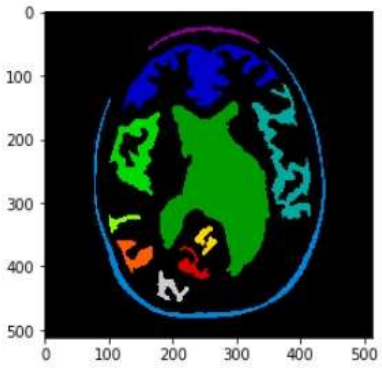
In [25]:
```
for i, image in enumerate(ic_seg_images):
    im_lab = color.rgb2lab(image)
    data = np.array([im_lab[..., 1].ravel(), im_lab[..., 2].ravel()])
    kmeans = KMeans(n_clusters=2, random_state=0).fit(data.T)
    segmentation = kmeans.labels_.reshape(image.shape[:-1])
    fig, axes = plt.subplots(1, 2)
    axes[0].imshow(image)
    axes[1].imshow(image)
    axes[1].contour(segmentation, colors='y')
```



Extracting objects from image

Measuring Region Properties.

```
Out[23]: <matplotlib.image.AxesImage at 0x2665bde18d0>
```



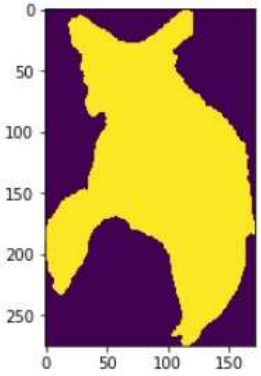

```
In [25]: props = measure.regionprops(labels, brain)
         for prop in props:
             print ('Area = ',prop.area, '  Perimeter = ', prop.perimeter)
```

```
Area =  1087   Perimeter =  397.56349186104046
Area =  8833   Perimeter =  1209.579869251966
Area =  5970   Perimeter =  2096.1139238614837
Area =  5588   Perimeter =  1105.4427842621187
```

Extracting effected Region.

```
In [27]: plt.imshow(props[4].image)
```

```
Out[27]: <matplotlib.image.AxesImage at 0x2665c2bc780>
```




```
In [28]: areas = np.array([prop.area for prop in props])
         perimeters = np.array([prop.perimeter for prop in props])
```

REFERENCES

[1] https://wiki.cancerimagingarchive.net/display/Public/CPTAC-GBM
[2] https://wiki.cancerimagingarchive.net/display/Public/RIDER+NEURO+MRI
[3] https://wiki.cancerimagingarchive.net/display/Public/REMBRANDT
[4] https://wiki.cancerimagingarchive.net/display/Public/CPTAC-CCRCC
[5] https://wiki.cancerimagingarchive.net/display/Public/TCGA-KICH
[6] https://wiki.cancerimagingarchive.net/display/Public/CPTAC-LSCC#e4e04de0b6f5475696e68126ac21cbee
[7] https://wiki.cancerimagingarchive.net/display/Public/TCGA-HNSC
[8] https://wiki.cancerimagingarchive.net/display/Public/CPTAC-HNSCC