

## Project Report

### 1. Effects to achieve

This project focuses on image segmentation based on 3 methods, respectively Ostu's method, Kmeans algorithm and Gaussian Mixture Model.

### 2. Details of algorithm(All algorithms are implemented by myself)

- i) In Ostu's method, I used the method covered in class and just pick a threshold that maximizes the between-class variance.
- ii) In Kmeans algorithm, I optimize the initialization using Kmeans++ initialization method proposed by David Arthur and Sergei Vassilvitskii. The model is encapsulated in **kmeans.py**
- iii) In Gaussian Mixture Model, I implement the EM algorithm, which iteratively computing E-step and M-step. E-step computes the mixture probabilities for all data points while M-step updates the means, covariance matrices and prior probabilities. The initialization schemes also use kmeans algorithm that only runs one step in order to achieve good initial parameter settings. The model is encapsulated in **EM.py**

### 3. Instruction on how to run this program

#### Step 1:

This is an image processing web application which are embedded in Flask framework. Before running this web app, please make sure that you install Flask backend services.

To install Flask, just type "**pip install Flask**" in your command line or terminal.

Other packages include **cv2**, **numpy** and **scipy**.

#### Step 2:

To run this program, type in your command line "**cd 'root directory of this application(where app.py resides)'**". Inside your root directory, type in your command line "**python app.py**", then the application will be hosted on your localhost, and port by default is 8080, as shown in Fig 1.

(If port 8080 is preempted by another application, you can change this configuration in **app.py**. Find this line **app.run(host='localhost', port=8080, debug=True)**, and change the port number to any other available ports from 0 to 65535.)

```
C:\Users\LXL>cd Desktop\image processing website
C:\Users\LXL\Desktop\image processing website>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 653-179-533
* Running on http://localhost:8080/ (Press CTRL+C to quit)
```

### Fig 1 Start running your application

Step 3: Open up the localhost in your browser, I recommend you use Chrome or FireFox. After you open this website, you will see the page shown in Fig 3.

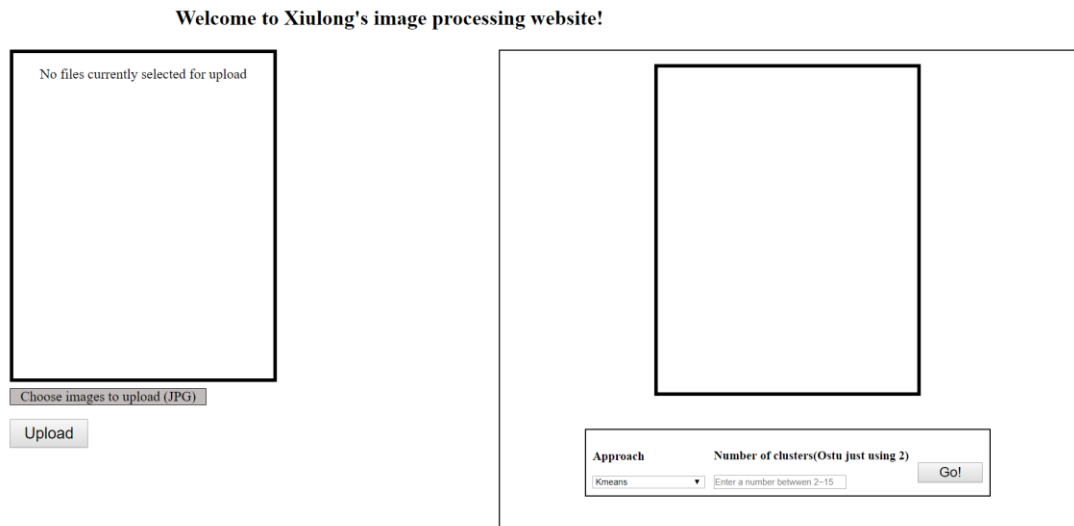


Fig 3 GUI of this website

Step 4: Upload your image to process. On the left part, you can upload your images by clicking “Choose images to upload(JPG)”. Images with jpg or jpeg extensions are required. After choosing your image, you will be able to view this image inside the left frame like shown in Fig 4.

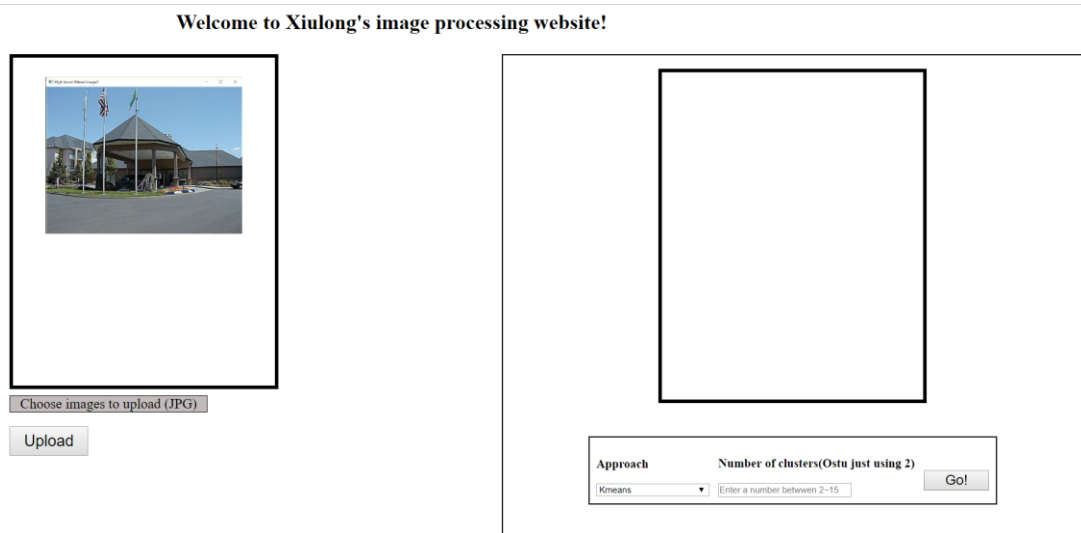


Fig 4 Choose your image to upload

Then click “Upload” button, you will see a message right to the button telling you the image has been successfully uploaded.

Step 5: Process your image. You can choose one of the three methods to process

your image by selecting from the drop-down menu. They are kmeans, Ostu and GMM. For kmeans and GMM, you can enter the number of clusters you want to segment, the range is between 2 to 15. For Ostu's method, the algorithm just use binary segmentation, so you don't need to enter the number of clusters. Once you are ready, click "Go!" to process your image.

Step 6: Wait for the process to complete. The speed of process depends on number of clusters, the choice of algorithm, and size of image. To save your time, we recommend you choose images with relatively small size, and choose a small value for the number of clusters. For Ostu's method, it can be done instantly. For Kmeans and GMM, it may take up to 3 minutes for larger images. Feel free to make a cup of coffee while you're waiting. You can actually see some temporary printing results for kmeans and GMM in your command line. The algorithm runs at most 10 iterations.

Step 7: Upon completion, the result image will show inside the right frame like shown in Fig 5.

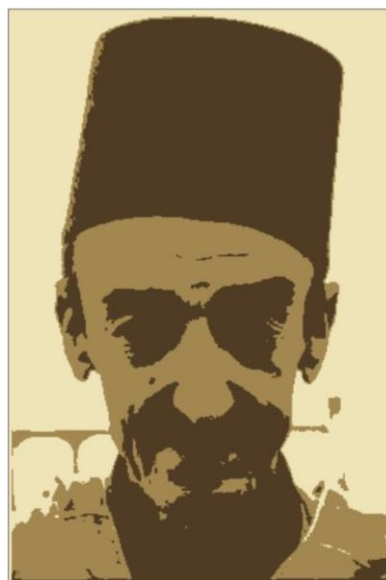


Fig 5 Kmeans using  $k=3$

#### 4. Conclusion

In this project, I get better understanding of how these basic cluster algorithms work by implement all details by myself. Also, building this tiny full stack system enables me to have a better understanding of how to build a web application. In future, I will implement more advanced algorithm like graph cut, which will involve more user interaction and will produce more robust segmentations that user expects.