Computer Vision Challenge

Proposal

Han Zucheng	Tian Jian	Wang Huiyu	Yan Xu	Zhang Xiaoang
03739137	03749328	03746516	03750567	03743260

July 11, 2021

Impact of human activities on the landscape from above

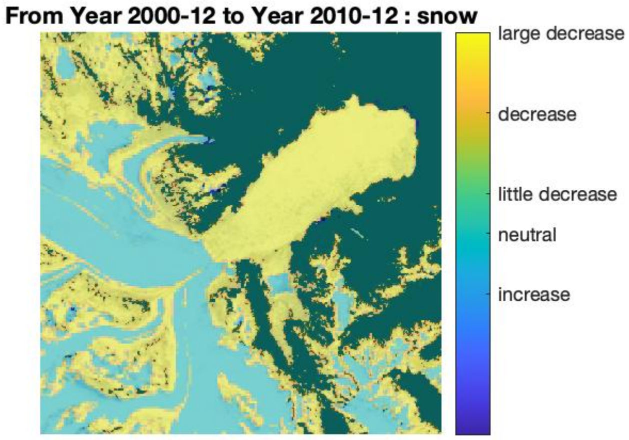

Figure 1: Heatmap

Contents

1 Introduction

Since the existence of the earth, the landform has never stopped change. The causes of the variation of the landform are not only natural, such as the eruption of volcanos or plate drift, but also deeply related to human behaviors. Due to the increase of the global population and human productivity all these years, the change of the landform speeds up, which leads to the degradation of the natural environment. The expansion of the cities causes the loss of arable land; deforestation leads to land desertification; industrial pollution brings the air pollution and the emission of Greenhouse gases results in global warming, followed by glaciers melting, the rising of sea level, and the worldwide floods. Aiming at environment protection, our software focuses on the identification and analysis of the landform change topographically by processing the satellite images of the same area continuously. In this process, visualization and quantitative analysis will be applied.

2 Image processing algorithm

The idea of the image processing part is mainly divided into three parts. Use SURF and KAZE to find correspondences points, and Ransac eliminates outliers. Finally, the similar parts of the pictures are overlapped, by the result of the four-point algorithm. The flow chart is as follows.

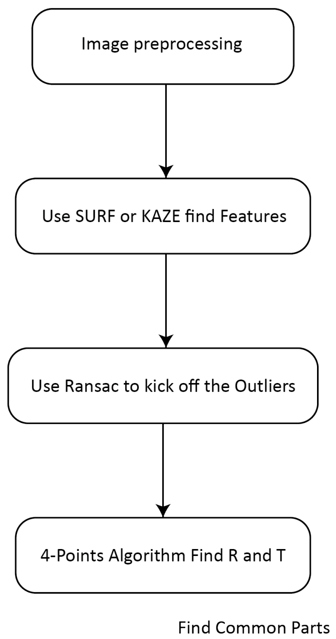

Figure 2: Flow chart

2.1 SURF

Like the Harris Detector that we have learned from the computer vision lecture, Speeded Up Robust Features (SURF) is also a method to capture features of images. As its name indicates, it is a speeded up version of another robust feature detector called SIFT[1]. Compared to Harris and SIFT, SURF can not only capture more robust correspondences or do further tasks like finding the same area of more than two satellite images, but also can be considered as a faster option when you run our MATLAB App. For the sake of completeness we will have a short theoretical view of it.
Unlike Harris Detector, the Hessian matrix and its determinant plays the important role for finding features in SURF:

$$\mathcal{H}(\mathrm{x},\sigma) = \begin{bmatrix} L_{xx}(\mathrm{x},\sigma) & L_{xy}(\mathrm{x},\sigma) \\ L_{xy}(\mathrm{x},\sigma) & L_{yy}(\mathrm{x},\sigma) \end{bmatrix}$$

where $L_{xx}(\mathrm{x},\sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I in point $\mathbf{x}$, and similarly for $L_{xy}(\mathrm{x},\sigma)$ and $L_{yy}(\mathrm{x},\sigma)$. By blurring (filtering) the image of different degrees and thus obtaining the Gaussian Pyramid, the difference of Gaussian can be computed to find the interest points (extreme points) of different scales[1].

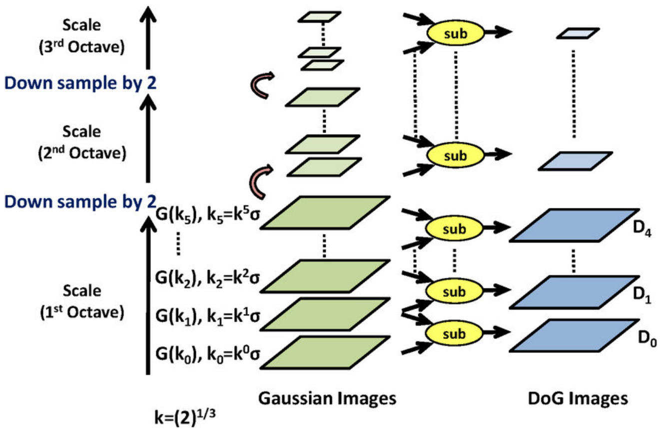

Figure 3: Gaussian pyramid and DoG pyramid[2]

For speeding up of the filtering the use of the original image should be replaced by its integral image[3], so that the computational cost is stripped down from O(n) to O(1).
Secondly, instead of Gaussian filters, the box filters are used to reduce the computational cost. According to H. Bay, T. Tuytelaars, and L. Van Gool[4], the box filters approximate second order Gaussian derivatives, and can be evaluated very fast using integral images, independently of size[4]. Besides, the descriptor size of SURF can also be reduced in comparison to SIFT, depending on the size of subregion of the region around the interest points. Details can be found in the reference[4]. The result of the robust correspondences are shown in the following figure for the example of Columbia Glacier.

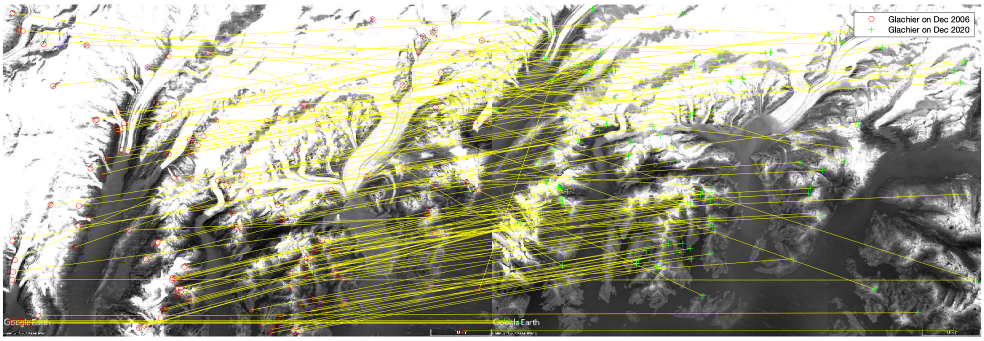

Figure 4: robust correspondences

As we can see, although the correspondences are meant to be robust, there are still some (sometimes even many) of them which are not really correspondences. Obviously, they are not suitable for the 4-point algorithm in the further step for finding out the camera rotation and translation. Therefore we must apply RanSac Algorithm to the correspondences to kick off those outliers, which are not really robust correspondences.

2.2 Ransac

Ransac stands for random sample consensus[5].It's a quite simple but highly effective algorithm that it can be used if data is affected by outliers. As we all know the real-world sensors data will never be perfect and they are quite often affected by outliers. So we need Ransac to group data points into an entire set and into an outlier set so that one can forget about the outliers and work with your inliers. The whole process is mainly a simple three-step procedure.

1. First step is sampling. We sample a subset of data points and we consider those data points to be inliers for this iteration and all the computations that we do are based on these sampled data points.

2. Step number two is a task specific step. We'll take those potential inliers and we'll compute the model parameters or solve the tasks that we actually want to solve To solve the problem of false matches after Surf, we also introduces RANSAC algorithm based on Sampson distance.

$$d_{Sampson}(x_1, x_2) = \frac{(x_2^T H x_1)^2}{\|\widehat{e_3} H x_1\|^2 + \|x_2 H \widehat{e_3}\|^2}$$

3. The third step is a scoring step where we see how many of the remaining data points will support this model. By simply counting we can compute the score and the repeat this process over and over again.Take the model that is best supported by the data which then tells us which are the inlier points which are the outlier points. Of course

we are supposed to know how often should we actually do the process.
We can compute it like this.

$$s = \frac{log(1-P)}{log(1-(1-\epsilon)^k)}$$

After we know the probability of success P, the outlier ratio in our data points epsilon and the number of sampling points k, we can easily calculate how many trials T we need.

So Ransac is a frequently used algorithm, whenever you work with real-world Center data. For example to compute visual odometry that means estimating the motion of a camera through the environment by taking into account the corresponding points in pairs of consecutive images so that are then can compute the relative orientation of the camera just based on those correspondences and given that this approach is kind of sensitive to outliers so you need to make sure we have only inliers in order to estimate the trajectory well and Ransac is an approach that helps me get rid of those outliers here.

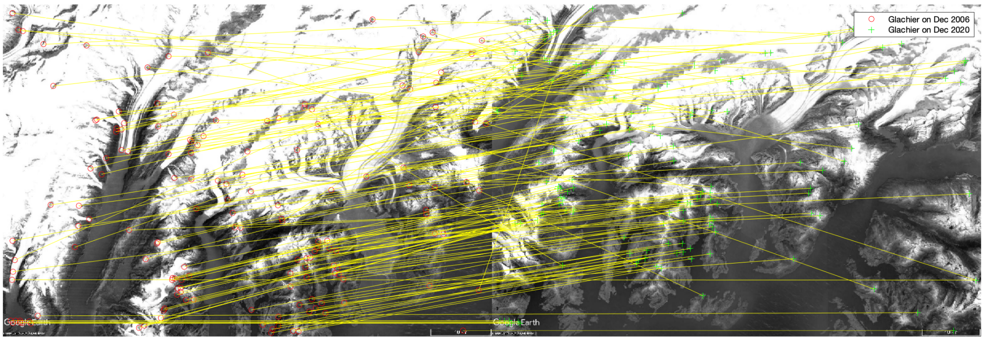

Figure 5: SURF Correspondences before RanSac

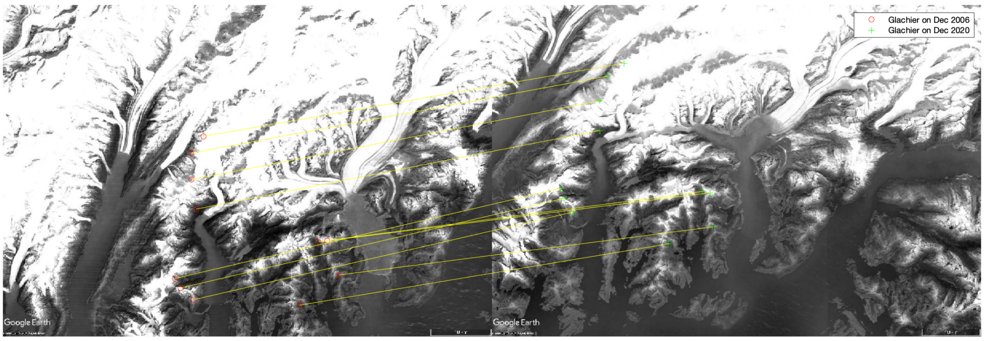

Figure 6: SURF Correspondences after RanSac

2.3 KAZE

We encountered difficulties in matching some types of pictures by using the SURF algorithm. In order to improve the situation that the recognition of some image types(Frauenkirche, Wiesn) through SURF algorithm is not ideal, we decided to use KAZE[6] method. KAZE detect method is a novel multiscale 2D feature detection and description algorithm in nonlinear scale spaces. Compared with the SIFT algorithm that construct the Gaussian scale space of an image, it detects and describes the 2D features in a nonlinear scale space by means of nonlinear diffusion filtering. The computation time of KAZE is longer than SURF, but it shows a step forward in performance both in detection and description.

2.3.1 KAZE algorithm detector and descriptor

Different from the SURF algorithm, KAZE uses a nonlinear diffusion filtering method. The nonlinear diffusion filtering method regards the change of image brightness (L) on different scales as the divergence of some form of flow function, which can be described by nonlinear partial differential equations.

$$\frac{\partial L}{\partial t} = div(c(x,y,t)\nabla L)$$

By setting an appropriate conduction function c(x,y,t), the diffusion can be adapted to the local structure of the image. Normally we usually adopt this conduction function construction method. The brightness L of the image can be calculated iteratively.

$$g_2 = \frac{1}{1 + \frac{|\nabla L_\sigma|^2}{k^2}}$$

$$L^{i+1} = (I - (t_{i+1} - t_i)\sum_{L=1}^{m} A_l(L^i))^{-1} L^i$$

Similar to the SURF algorithm, the KAZE algorithm obtains the feature valves of the points through the Hessian matrix. And the descriptor of KAZE algorithm is M-SURF. In the algorithm test, we found that we can obtain more image matching points through the KAZE algorithm.

$$L_{Hessian} = \sigma^2(L_{xx}Lyy - L_{xy}^2)$$

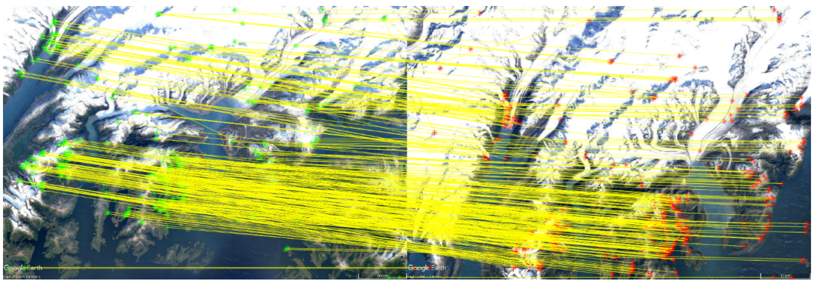

Figure 7: KAZE Correspondences

2.3.2 KAZE algorithm performance

The calculation time of the KAZE algorithm is much slower than SURF, but the calculation accuracy is much better than the SIFT and SURF algorithms.[7]

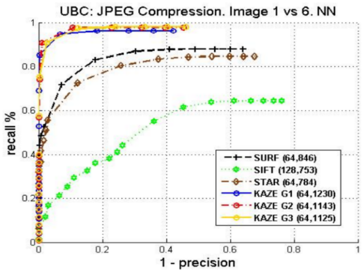

Figure 8: The performance of algorithms

The first difficulty for our team was that we need to find a way to improve the robustness of the algorithm.There are different types of pictures in the data set provided by the professor. Our initial algorithm can effectively identify pictures of natural environments such as Dubai or Columbia Glacier, but the recognition effect for urban environments and single buildings is not ideal, and correct image registration is almost impossible. So there is no way to talk about the subsequent analysis. In order to improve the accuracy of image registration, we decided to use the KAZE algorithm by searching literature. The KAZE algorithm obtains better calculation accuracy at the expense of calculation speed. When the SURF algorithm can't correctly identify the image, the KAZE algorithm often solves this problem easily. In the software designed by our group, the user can decide to use the KAZE or SURF algorithm when detecting the highlights of the image by selecting the mode SLOW or QUICK.

2.4 4-points algorithm

In order to overlap and compare the images, we need to find the camera rotation and translation through a suitable algorithm.
Actually, we have learned the 8-point algorithm and the 4-point algorithm in the computer vision course. Through these algorithms, we can obtain the rotation matrix R and the transform matrix T to achieve image registration.
However, the points from the eight-point algorithm can't be in a same planar, otherwise the essential matrix E cannot be solved. Therefore, the four-point algorithm is more suitable for Google satellite images. With the help of the knowledge that we learned in the Computer Vision course, we can construct the following equations to solve matrix H.

$$H = (R + \frac{1}{d} T n^T)$$

$$A = \begin{bmatrix} B_1^T \\ B_2^T \\ \vdots \\ B_n^T \end{bmatrix} B = kran(P_1, \widehat{P_2})$$

$$P_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \widehat{P_2} = \begin{bmatrix} 0 & -z_2 & y_2 \\ z_2 & 0 & -x_2 \\ -y_2 & x_2 & 0 \end{bmatrix}$$

$$AH_s = 0$$

Assuming that a point $A(x_1, y_1)$ is the point of image 1, and the corresponding point $B(x_2, y_2)$ belongs to the point of image 2. We can get the projection of image 2 on image 1 by Homography matrix H.

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} * \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

By finding the matching part between two images, we finally completed image registration and found the common part between the two images. In addition, it is difficult for us to obtain the specific parameters of the camera on the satellite. So the calibration matrix K can't be calculated. The camera calibration step is not performed.

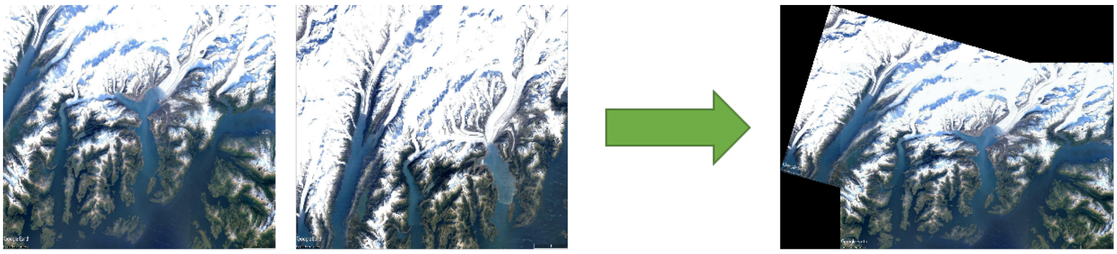

Figure 9: Image Registration

In practical applications, because the number of image matching points that we get is generally much greater than 4 points. This is a challenge for us. It will lead the matrix H to have innumerable solutions, so we need to select the most appropriate 4 points. In the actual program, after obtaining the matrix H by randomly selecting 4 points, we calculate the Shannon distances of other matching points. The closer the distance is, it indicates that the homography matrix H can better describe the transformation relationship between images.[8]

3 Visualization

We hope to adopt more visualization methods so that users can feel the changes in the environment from different angles. The Heat map and Bar Map are based on the analysis of K-means Clustering results. The Highlights can be obtained by arbitrary switching of two algorithms, namely Subtraction and PCA. In addition, we will also display the similar parts between the pictures obtained in the previous chapter in the form of automatic switching slides, that is, Time Lapse. The flow chart is as follows.

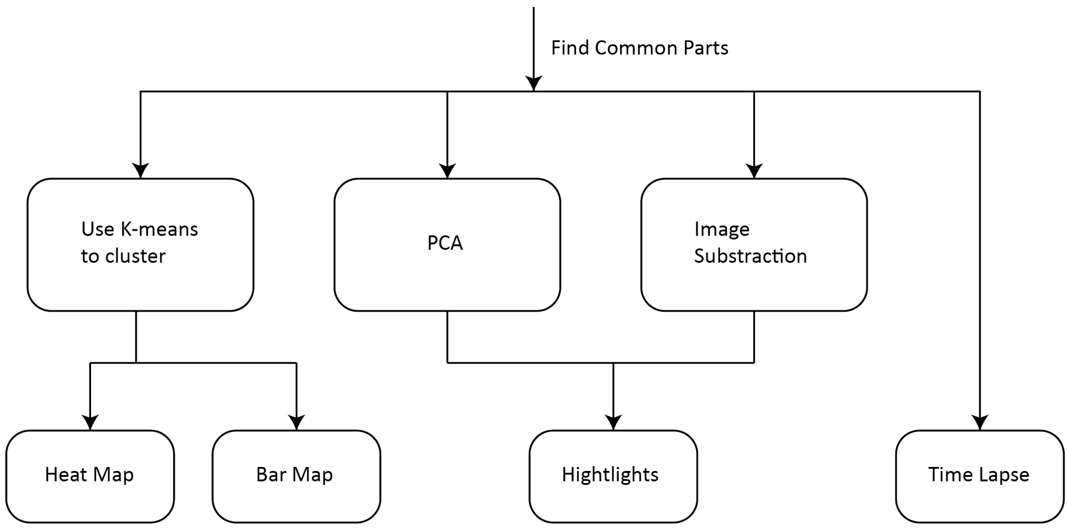

Figure 10: Flow chart

3.1 PCA

3.1.1 Finding different highlights of image by subtraction method

First of all, we tried to use the direct subtraction of the two pictures to get the highlights between them. Through the four-point algorithm, we can effectively perform image registration and find the same part of the two images. After subtracting the same part of the image, we can find the different Highlights of the images.[9]

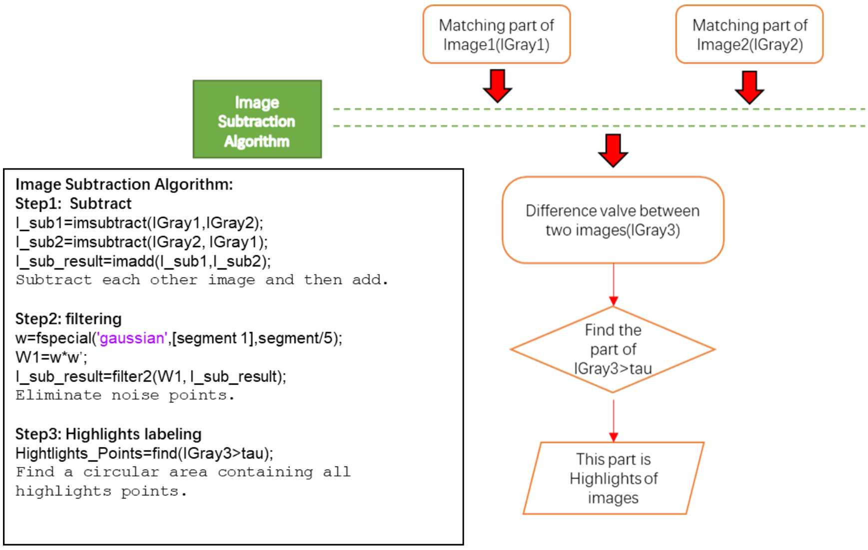

Figure 11: Image Subtraction Algorithm

With the help of this method we can clearly find the obvious difference between the two pictures. For example, in the image type of Columbia Glacier, ice and snow in the same area are the most obvious high-lights between the images.
However, this method is very easy to affected by the noise interference. In addition, this method pays more attention to the difference Highlights between images rather than the characteristics of the image itself. For this problem we need to choose the PCA method as a better algorithm.

3.1.2 PCA algorithm to analyze the highlights of images

PCA algorithm(principal component analysis) is normally used to explore data analysis and generate predictive models. It is very common method that we use in image recognition by projecting each data point onto only the first few principal components to obtain lower dimensional data.[10][11]
Through dimensionality reduction, we can refine the important information points in the image, and then we regard the information as the highlights of the image. Assuming that the

resolution of the image is $m*n$, we can construct a matrix$X = \{x_1, x_2, x_3, \cdots, x_{m*n}\}$, and we calculate the covariance matrix Cov.

$$\mu = \sum_{j=1}^{n} x_j, H = \frac{1}{\sqrt{n-1}}[x_1 - \mu, x_2 - \mu, \cdots, x_{m*n} - \mu]$$

$$Cov = HH^T$$

Calculating the eigenvalues and eigenvectors of matrix Cov, and we take the eigenvectors corresponding to the first h largest eigenvalues to construct matrix E. The matrix E can reduce the dimensionality of the image data.

$$E^T X = Y$$

Through the matrix Y, we can find the highlight points in the image. By comparing the highlights from different images, we can better analyze the changes in the region with time lapsing. Compared with the sub-traction method, PCA algorithm can better express the characteristics of the image itself.

3.1.3 Visualization 1: Highlights

In our software the user can choose two ways mentioned above by switching the mode(normal or PCA) to analyze the highlights of images.**(large/little)**

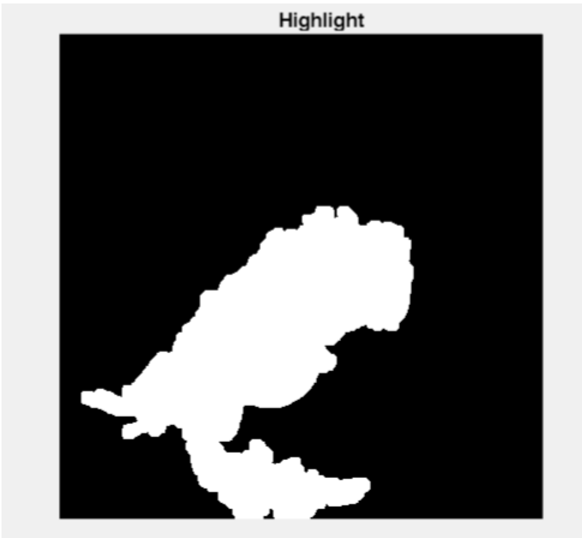

Figure 12: Highlight

3.2 K-means Clustering

In the previous steps, we have already found the same area of the satellite images. Now it is time to find out the changes from year to year (or month to month). For landscapes in the nature, it is an intuitive to distinguish different objects (like snow and dry land) by their colors. Therefore our approach is to use K-means clustering to divide the same area into segments, where the dominante colors of the same area are extracted.

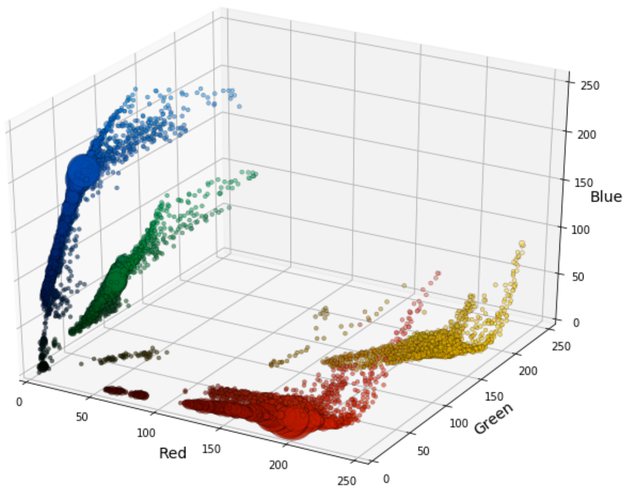

Figure 13: K-means Clustering[12]

By choosing the number of clusters, e.g., three starting points(pixels) will be randomly chosen and labeled with distinctive indices. In RGB color space the algorithm tries to group the pixels according to their RGB values and minimize the difference of their features (RGB values) in each cluster. After some iterations, the (nearly) white pixels are extracted from others as we can see in the following figure the comparison to the original image. In the following text we will call such clusters like cluster 1 and cluster 2 as color mask.

Through the results of K-means Clustering, we can clearly see the environment represented by each color classification in the window. We empower users to name each category and use them in subsequent processing. We believe that the results after clustering can satisfy users' free choice in the **city/countryside/rivers mode**.

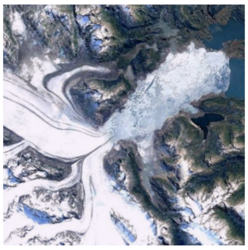

Figure 14: Original Image

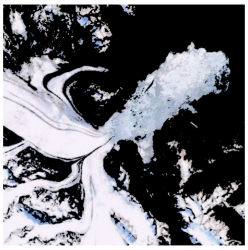

Figure 15: Cluster 1: Glacier

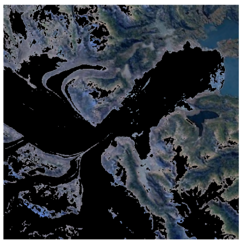

Figure 16: Cluster 2: All Other Things

We have also tried out some methods to enhance the clustering results. For example, we have tried to project the RGB colors into L*a*b* color space[3] and expected, that the deep and light colors can be better distinguished from each other. However, the result shows that the distinguishing between deep and light color might not be a really good idea as we can see in the following resulted images, the white snow in the shadow are separated from the snow under ‚normal' illumination, which is of course unwanted.
One problem we had by applying the k-means is that for e.g. the ‚white' cluster might be not all given the same index label (e.g. ‚1) for all the images from different year like the following

figure shows. However, they must be aligned in the same column so that the user can better see in the plot the comparison and changes. To address this problem, the software computes the correlation of one cluster to all clusters in the previous year to find which one from the previous year would match at most.
Even though it might still sometimes come to mismatching when the changes from one year to the next is too large. Because what we have extracted is the color in-formation, the average intensity of one cluster will also be compared as an addition criterion to find the actual ‚best match' to the clusters of the previous year. For the Dataset ‚Columbia Glacier', this approach works well and in the resulted subplot the cluster ‚snow' are all in the same column.

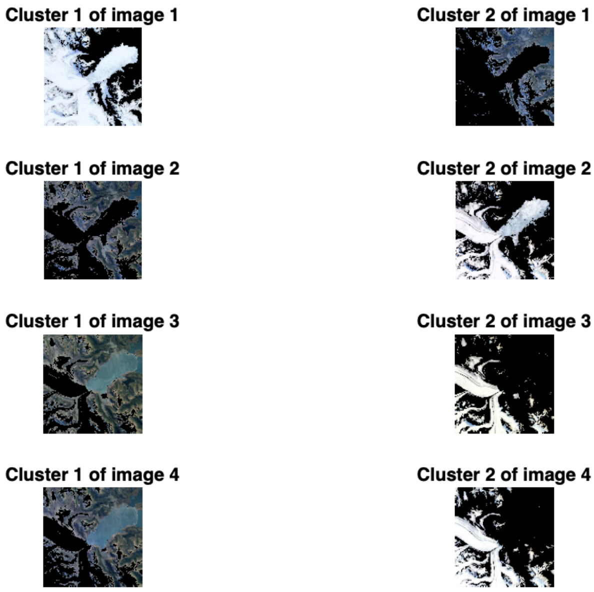

Figure 17: Mismatching

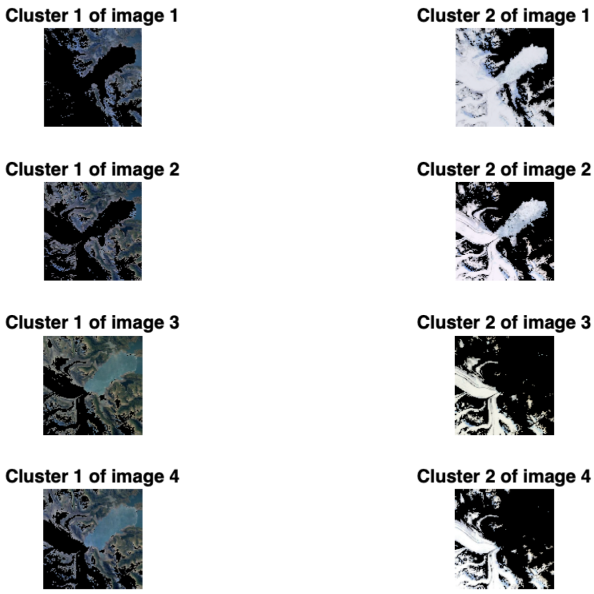

Figure 18: K-means result

3.2.1 Visualization 2: Heat Map

After that we have finished distinguishing the colors and clustered them into different classes, now it is easy to find out the changes of the clusters from one date to another. The first visualization method we use is the heat map, which plots colors from cold to warm to indicate the degrees of increase/decrease. We can easily see the size of the change between the two pictures.**(large/little)**

As we can see in the following figure, the vanishing parts of the Glacier are plotted in warm colors in this visualization mode.

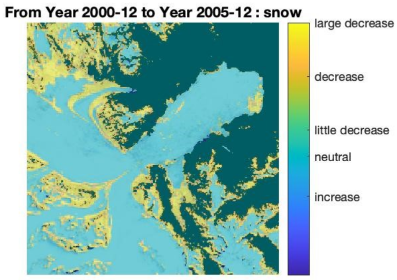

Figure 19: Heat Map

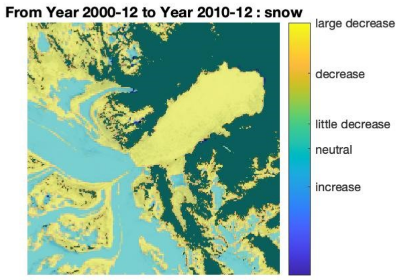

Figure 20: Heat Map

3.2.2 Visualization 3: Bar Map

Based on the K-means Clustering, we can display each part represented by color in the form of Bar Map. The abscissa represents time, and the ordinate represents the proportion of each part. From the changes in the bar graph over time, people can easily feel the melting of glaciers, the movement of coastlines and the impact of human activities on nature.**(quick/slow)**

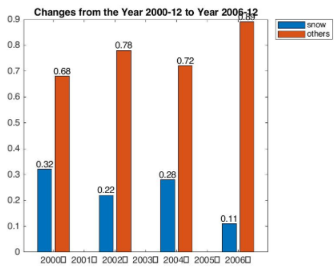

Figure 21: Bar Map

3.2.3 Visualization 4: Time Lapse

After having satellite images of different ages, we can't help asking ourselves, why not quickly display these images in order? Showing the unified picture to users will leave a deep impression on them. Because Time Lapse can intuitively make people aware of the changing speed of the environment under the satellite image. We like this display method and provide users with three different picture switching speeds.**(quick/slow)**

4 GUI Manual

When we run the file “main.m”, a user interface will automatically show up. This interface contains the following functions: select pictures from Datasets; compare pictures to find common parts; Use k-means method to classify the different components in the image and Visualize changes in four different ways. Users can click the “Readme” button to get information of Program and GUI instruction.

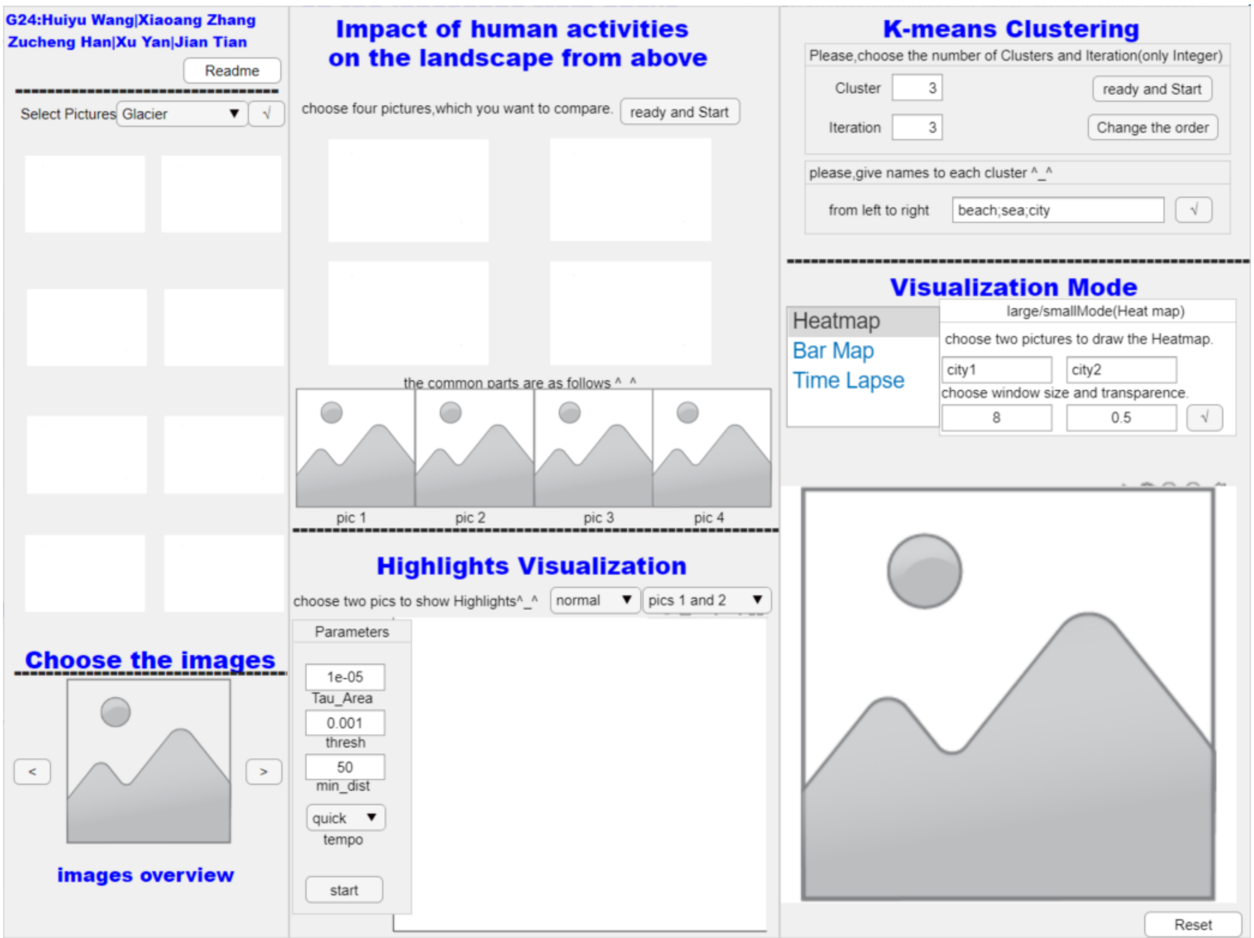

Figure 22: GUI design

4.1 Select Images

First of all, we have to choose a location on the earth through the drop-down menu at the top left of the interface, Then analyze the changes in this area over a period of time, Here users can choose provided images from Dataset, or use their own images. The window below can overview the selected images.

4.2 Compare Pictures

In this section users should select 2 to 4 images from the selected images set, and the selected images will be displayed at the top middle of the interface. After the images was selected, the user clicks "ready and start", the common part of the images will be found by using 4 point algorithms.

4.3 K-means Clustering

In this part, we give users great autonomy and freedom. users can determine the number of clusters and iterations by themselves and they can also name the clusters according to their needs.It should be noted here that the more iterations the more accurate, but the more time will be taken.

4.4 Visualisation Mode

We provide users with four visualization methods. The changes in the same location of 2 pictures can be shown in form of Heat map or Bar chart or Time slide in the window at bottom right corner of interface. Users can also see the Highlights of changesthe threshold and other parameters can be controlled by users.

5 Results in Pictures

Here we show three examples of using our program as results. First example: Columbia Glacier

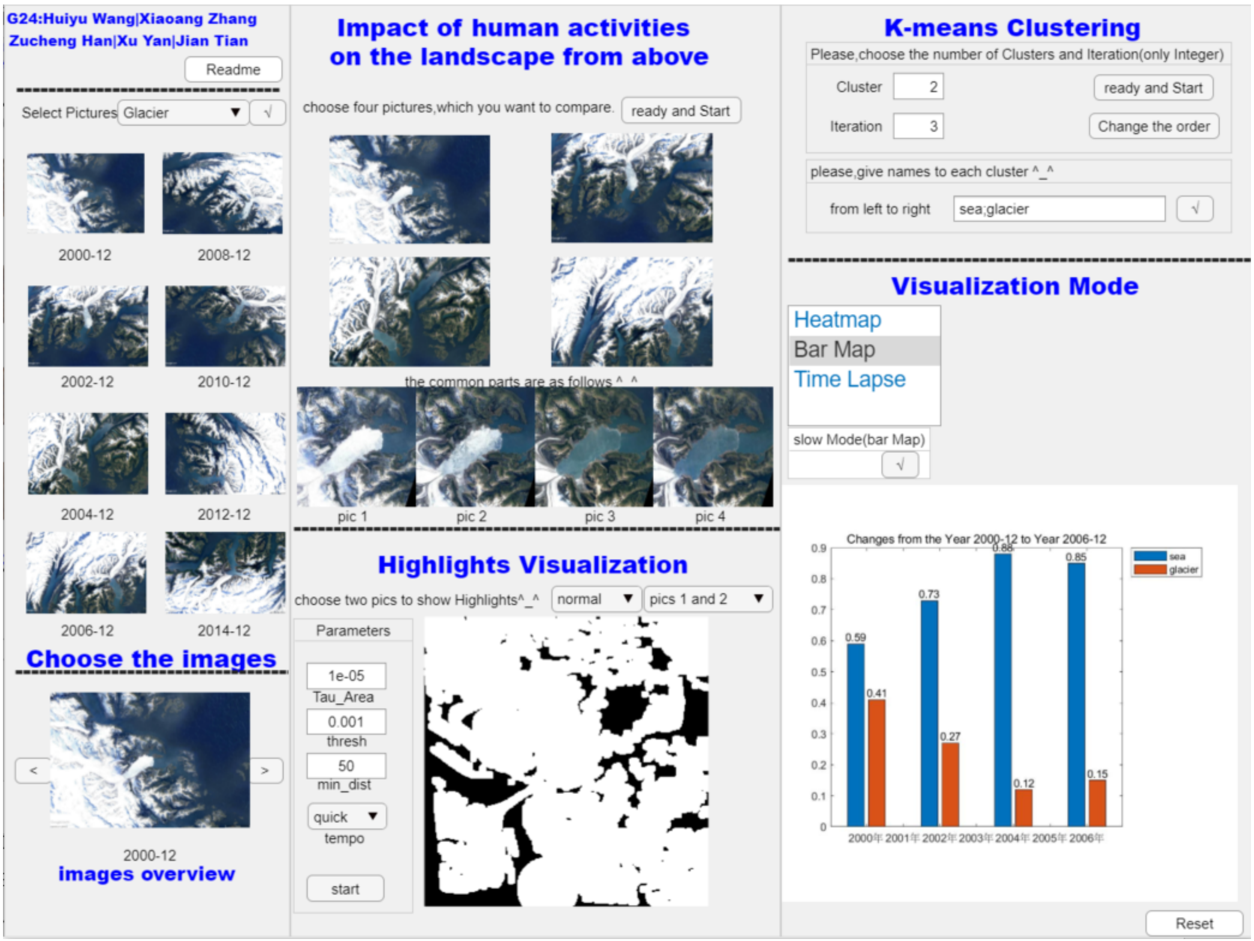

Figure 23: Analysis of the landform changes in the area of Columbia Glacier

In this example, we have selected four satellite images from 2000 to 2006. First, we find the common parts of these four images and project them into the same coordinate system. Then we find the changes between the images and visualize them by four different ways.

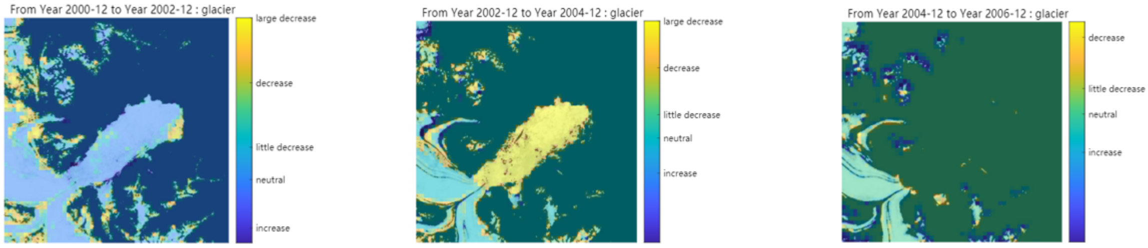

Figure 24: From the heat map, we can see that the area of glaciers is decreasing

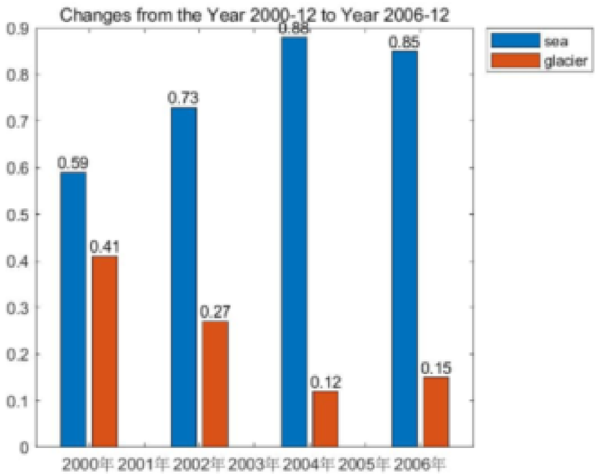

Figure 25: With bar chart we can see the changes of multiple landforms

From the above figures, we can see that in the area of Columbia Glacier , from 2000 to 2006, the area of Ice decreased, and the area of oceans and hills expanded. The reason for this is maybe that, due to the large amounts of CO2 emissions, the temperature of the earth is increasing year by year, which leads to the melting of glaciers and the continuous expansion of the ocean area.

Second example: Dubai

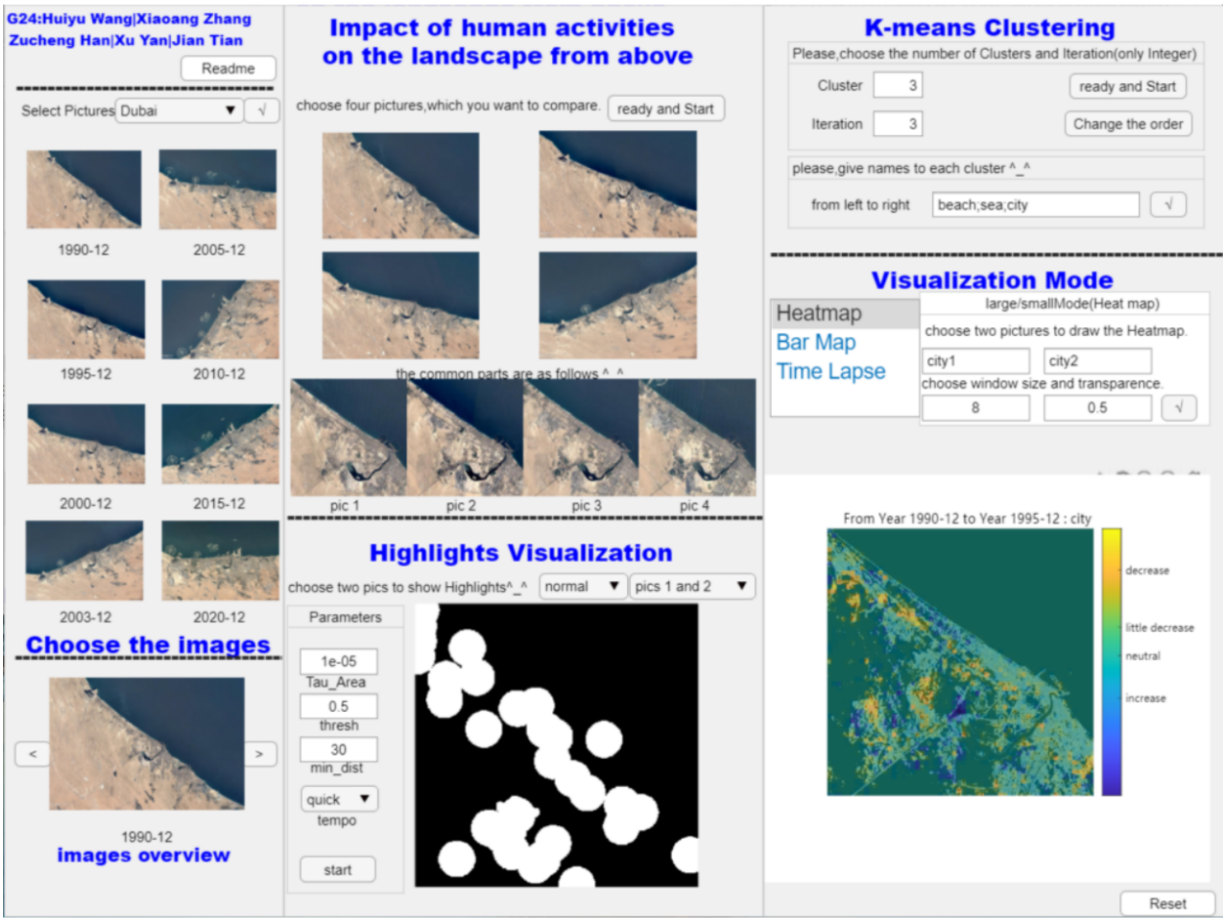

Figure 26: Analyzing the pictures of Dubai

From highlights visualization, man can see that, in the area of Dubai, the major changes are mainly near the coast, from 1990 to 2020 people built many new facilities along the coastline.

Third example: source region of the Yangtze River in China.

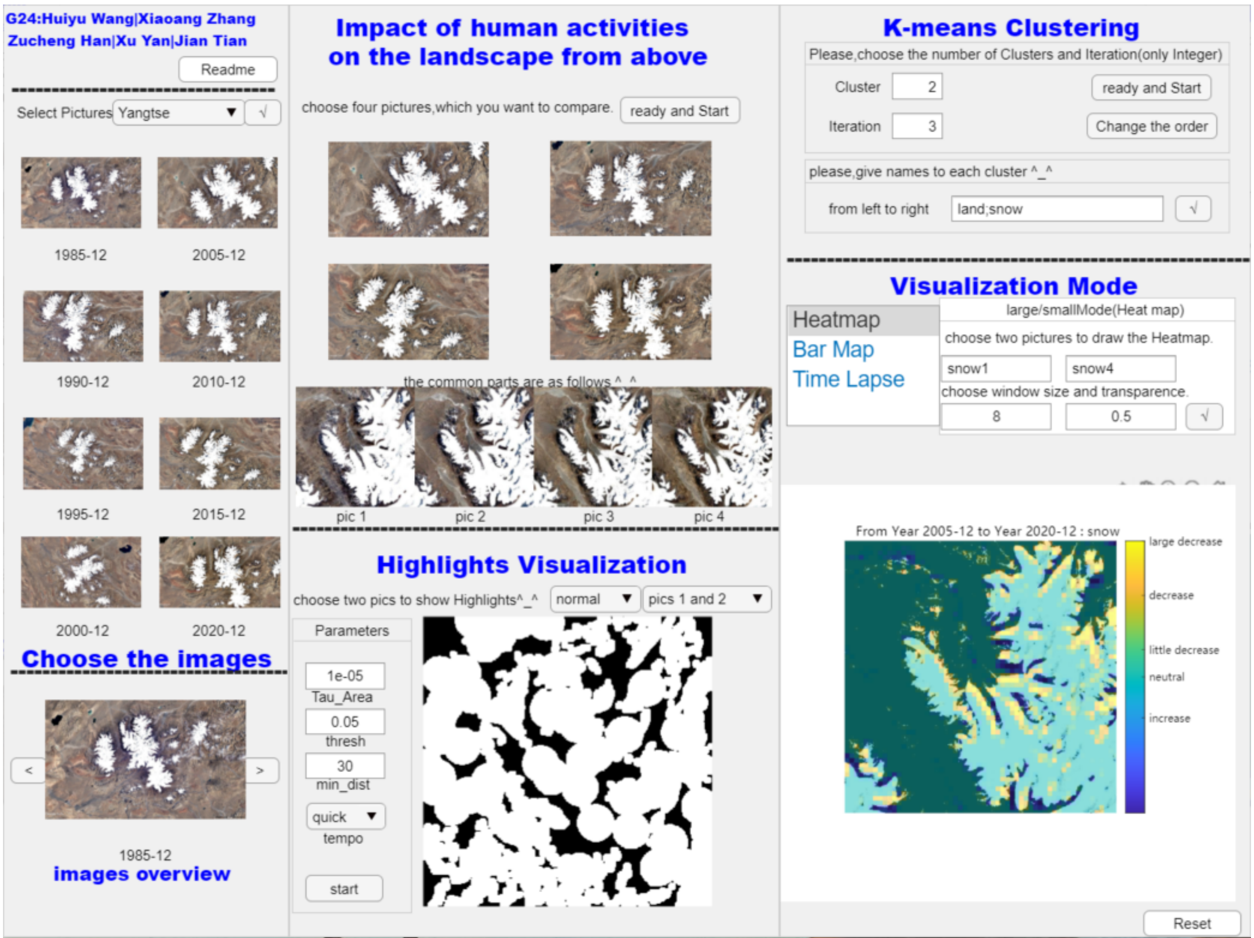

Figure 27: Analyzing of pictures in the source of yangtze river

In this example, we selected eight images with a long time span1995-2020. With the bar chart, we can clearly see that the area of the snow mountains in this area did not change much before 2000. After 2000, the area of the snow began to become smaller and the river's area is also getting smaller.

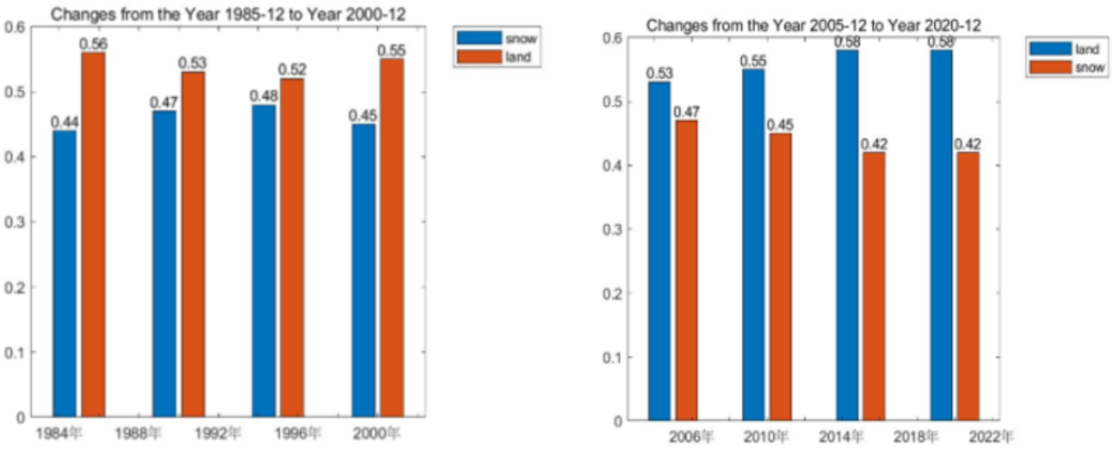

Figure 28: Bar chart

After 2000, the snow on the snow-capped mountains was gradually melting. This is obviously related to the increase in temperature. We can imagine that once all the snow melted, there would be no Yangtze River any more.

6 Conclusion

The CV Challenge is a huge project. No one can complete everything independently. Teamwork, effective communication and team member exchanges their thoughts are the key to solving the problem. After numerous programs debugging and modification, we finally solved the difficulties and challenges that we are faced with.

List of Figures

References

[1] David G Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[2] Feng-Cheng Huang et al. “High-performance SIFT hardware accelerator for real-time image feature extraction”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.3 (2011), pp. 340–351.

[3] MATLAB Group. *MATLAB documentation: Color-Based Segmentation Using the L*a*b* Color Space*. `https://www.mathworks.com/help/images/color-based-segmentation-using-k-means-clustering.html/`.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features”. In: *European conference on computer vision*. Springer. 2006, pp. 404–417.

[5] Yi Ma et al. *An invitation to 3-d vision: from images to geometric models*. Vol. 26. Springer Science & Business Media, 2012.

[6] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. "KAZE features". In: *European conference on computer vision*. Springer. 2012, pp. 214–227.

[7] Shaharyar Ahmed Khan Tareen and Zahra Saleem. "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk". In: *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE. 2018, pp. 1–10.

[8] Ke Yan. *image mosaic using SIFT*. https://www.mathworks.com/matlabcentral/fileexchange/30849-image-mosaic-using-sift//.

[9] Daniel Báscones, Carlos González, and Daniel Mozos. "Hyperspectral image compression using vector quantization, PCA and JPEG2000". In: *Remote sensing* 10.6 (2018), p. 907.

[10] Turgay Celik. "Unsupervised change detection in satellite images using principal component analysis and k-means clustering". In: *IEEE Geoscience and Remote Sensing Letters* 6.4 (2009), pp. 772–776.

[11] rulixiang. *ChangeDetectionPCAKmeans*. https://github.com/rulixiang/ChangeDetectionPCAKmeans//.

[12] tranberg. *Determining dominant colors in images using clustering*. https://tberg.dk/post/determining-dominant-colors//.