Team F: Rescue Rangers

Juncheng Zhang

Teammates:	Karthik Ramachandran
	Sumit Saxena
	Xiaoyang Liu

ILR10

4/06/2017

1、Individual Progress

1.1、Overview

During the past few weeks, my primary role was migrating the current algorithm from Matlab to Python, so that the code could be called by the python server and run at a faster speed in our software. I migrated the preprocessing part to the python platform, which finds the human candidates in aerial images, and combined it with the classification part I wrote before the last PR. Then, I tested the integrated algorithm on thermal dataset, and the result seemed to be comparable with the result of our Matlab version.

Also, I helped Xiaoyang improve the fusing method for RGB and thermal detection algorithms. After that, the integrated human detection algorithm had less false positives compared with the previous version.

1.2、Code migration to Python

For the preprocessing part, basically I used canny edge detection[1] to find contours of possible human and implemented Morphological Transformations to connect each contour and fill them[2]. Then, I found each connected pixels and only consider those connected areas with the same shape as human beings. The results of images after morphological transformations and after the integrated detection algorithm are shown below:

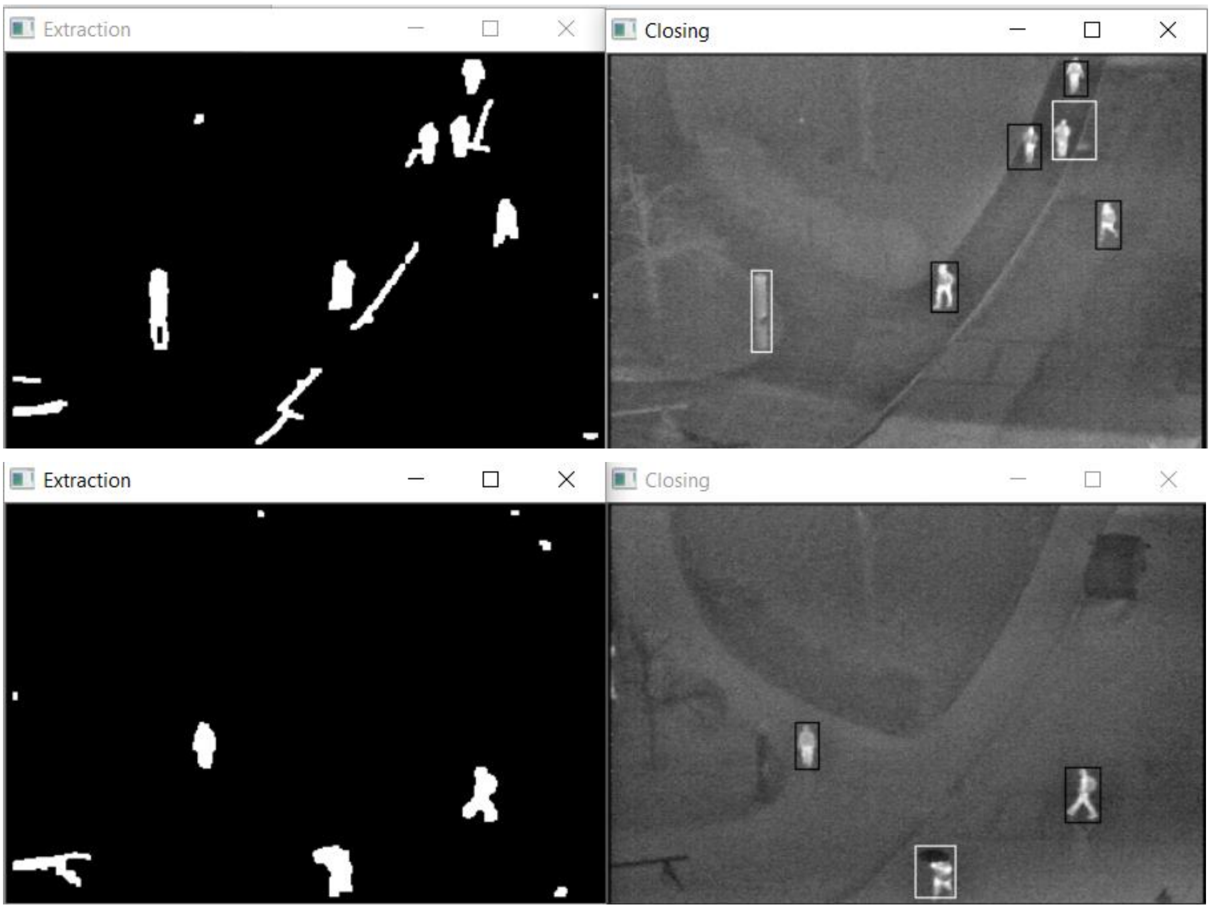

Figure1 result of test images

The images on the left are the results after preprocessing, which shows that the edge detection with morphological operations can successfully extract contours of potential human candidates. The images on the right demonstrate the classification results, and the black rectangles mean the classified humans, while the white represent non-human beings after classifications.

1.3、Improvement of human detection algorithm

Also, I tried to improve the performance of our integration of RGB and thermal human detection algorithms with Xiaoyang. Essentially, we reduced the number of false positives through the following two approaches:

1. Add drifts for mapping the ROIs in Thermal and RGB images
2. Integrate the ROIs from both the algorithms

By implementing the above two methods, the bounding boxes in RGB and thermal images could be matched without misregistration. Since the candidates will only be considered as humans if their bounding boxes in RGB and thermal images are both classified as human beings, the likelihood of false positives drops dramatically. One example of detected human beings in RGB image and thermal images is shown below:

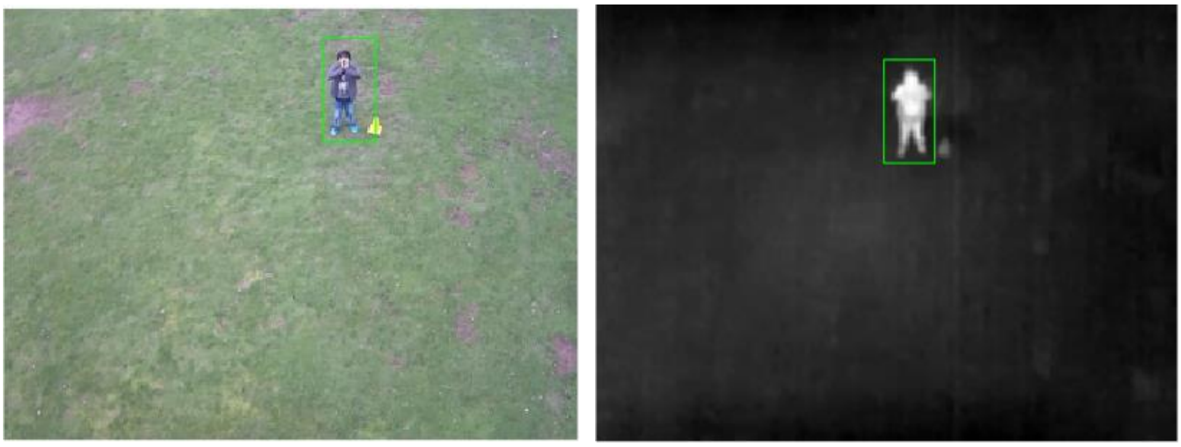

Figure2 result of detected huamn

As we can see from the image above, even though there is a clear shift between locations of bounding boxes in RGB image and thermal image, the algorithm can still match those two bounding boxes and detect the human being successfully. A video about implementing the improved human detection algorithm on one of our collected video can be found in the following link:
https://drive.google.com/file/d/0B19Cauta5rmFeHdVV3RYdUNJUWc/view?usp=sharing

As can be seen in the video, all the human beings shown in the video clip can be detected, and there are limited number of false positives. Since most false positives only appear in one single frame, and human beings can be detected in consecutive frames, the number of false positives can be further pruned if we take that into consideration.

2、Challenges

The main challenge I faced last week was to rewrite the preprocessing part to propose possible human candidates in aerial images. The difficult part was that even though I could find similar functions in OpenCV as the functions used in Matlab code, they were not exactly the same. Because of that, I had to figure out the usage of each possible functions for the same purpose, and decide which one I should use for our case. Finally, the code in Python showed similar results as the Matlab code even though they were not using the same methods. (e.g. Python code used canny edge detection method, while the Matlab code used sobel edge detection method).

3、TeamWork

After the previous progress review, our team discussed the plan for the next few weeks, and broke the work down as follows:

Table1 Work distribution form

Member	Work
Karthik Ramachandran	Develop Processing Pipeline
Sumit Saxena	Integrate GPS estimation with Signature detection
Juncheng Zhang	Code migration to Python; Improve human detection algorithm
Xiaoyang Liu	Add output stream of the detection algorithm; Improve human detection algorithm

The team worked with great coordination during execution of the entire task, and went to NREC together for one more data collection. We worked on different components of our whole system separately, and each of us did a good job.

4、Future Plans

Right now, we have most components of our whole system working. Before the next progress review, we will have an integrated end-to-end system based on the data processing pipeline, and hopefully can present a rehearsal of SVE. Also, we are going to explore some deep learning approaches for our human detection algorithms.

For my personal task, I will finish migrating the current Matlab code to the Python platform, and try to keep improving the performance of the current algorithm.

5、Reference

[1]https://www.packtpub.com/mapt/book/application-development/9781785283932/2/ch02lvl1sec20/edge-detection

[2] http://stackoverflow.com/questions/10316057/filling-holes-inside-a-binary-object