

Team F: Rescue Rangers

Juncheng Zhang

Teammates: Karthik Ramachandran

Sumit Saxena

Xiaoyang Liu



ILR09

3/23/2017

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 finally run in a faster speed in our pipeline. I first migrated the HOG+SVM to the python platform, which is the essential algorithm for our system, and tested it on our thermal dataset. The result seemed to be comparable with the result of our Matlab version.

Also, the team went to NREC for data collection for three times during the spring break, and successfully collected RGB, thermal and sound data. In addition, we tested various mounting configurations of microphone and camera, and tested various flight parameters. Lastly, I also helped Xiaoyang with the fusing method for RGB and thermal detection algorithms.

1.2、 Data Collection at NREC

Before the actual data collection, the team firstly need to figure out how to mount the camera and the microphone. For the camera mounting, we purchased the frame for GoPro Hero4, which has the same size as our Flir Duo, and also the suction cup for attaching the camera on the drone. The images of those two products are shown below:



Figure1 Frame for Gopro



Figure2 Suction Cup for Gopr

We firstly mounted the camera in a downward facing way, but the detection result was not satisfactory because of varied poses of humans in the view. Then, we adjusted the mounting so that the camera would have some angles, which improves the accuracy of human detection algorithm, since most humans are upright in the images given the angle of the camera. Also, after testing several configurations of mounting the microphone, we finally decided to hang it to keep it from being too close to the propellers.

Except for collecting RGB, thermal and sound data of human, we also collected other signatures, including mattresses with bright colors, and items with thermal features. Two samples of those possible signatures are shown below:



Figure3 Mattress with bright color



Figure4 Item with thermal features

1.3. Code migration to Python

In order to fasten the speed of our algorithm and make sure it can be called by our pipeline built in Python, I started to migrate the code from the Matlab platform to Python. Right now, I have already implemented HOG+SVM based on open source library, including `skimage.feature[1]` and `sklearn.svm[2]`. The default settings of extracted hog features in the `skimage.feature.hog` function is shown below:

```
def hog(image, orientations=9, pixels_per_cell=(8, 8),
        cells_per_block=(3, 3), visualise=False, transform_sqrt=False,
        feature_vector=True, normalise=None):
    """Extract Histogram of Oriented Gradients (HOG) for a given image.

    Compute a Histogram of Oriented Gradients (HOG) by

    1. (optional) global image normalisation
    2. computing the gradient image in x and y
    3. computing gradient histograms
    4. normalising across blocks
    5. flattening into a feature vector
```

Figure4 Default setting of `skimage.feature.hog`

I tested it using the thermal dataset, including 4728 positive images and 5430 negative images for training, and 850 positive images and 1060 negative images for testing. The result is shown in Figure5.

The default setting is apparently not the best parameters for hog features in our use case. I just used the same settings of the hog descriptors in the Matlab version of our code, which is

shown in Figure6. Using the same training set and test set, the classification accuracy improves a lot, as is shown in Figure 7.

```
In [31]: run test-object-detector
Calculating the descriptors for the positive samples and saving them
Positive features saved in ../data/features/pos
Calculating the descriptors for the negative samples and saving them
Negative features saved in ../data/features/neg
Completed calculating features from training images
Training a Linear SVM Classifier
Classifier saved to ../data/models/svm.model
Testing our classifier on positive images
0.830588235294
Testing our classifier on negative images
0.832075471698
```

Figure5 result of HOG+SVM using the default setting

```
[hog]
min_wdw_sz: [32, 64]
step_size: [10, 10]
orientations: 9
pixels_per_cell: [8, 8]
cells_per_block: [2, 2]
visualize: False
normalize: True

[nms]
threshold: .3

[paths]
pos_feat_ph: ../data/features/pos
neg_feat_ph: ../data/features/neg
model_path: ../data/models/svm.model
```

Figure6 Adjusted setting of skimage.feature.hog

```
In [1]: run test-object-detector
Calculating the descriptors for the positive samples and saving them
Positive features saved in ../data/features/pos
Calculating the descriptors for the negative samples and saving them
Negative features saved in ../data/features/neg
Completed calculating features from training images
Training a Linear SVM Classifier
Classifier saved to ../data/models/svm.model
Testing our classifier on positive images
0.894117647059
Testing our classifier on negative images
0.868867924528
```

Figure7 result of HOG+SVM using the adjusted setting

2、Challenges

The main challenge I faced last week was to understand the open-source Python code and to modify it for our use case. The example of object detector using HOG as a descriptor and linear SVM as the classifier can be found in [3]. This open source example gives me the direction about what library I should use in python and how should I structure the code.

However, I still had to understand the source code in the library so that I could tune the parameters of hog for our application. Finally, I could change the hog parameters easily through a configuration file and write testing scripts to test our algorithm using the thermal dataset we collected before.

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	Test sound detection algorithm on our collected dataset
Sumit Saxena	Develop other signature detection algorithm; Test GPS location reporting algorithm
Juncheng Zhang	Code migration to Python
Xiaoyang Liu	Test fused human detection algorithm on our collected dataset

The team worked with great coordination during execution of the entire task, and went to NREC together for three times for 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 each component of our whole system working. Before the next progress review, we plan to have an integrated end-to-end system based on the data processing pipeline. 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 improve the current fusion algorithm.

5、Reference

- [1]. <http://scikit-image.org/docs/dev/api/skimage.feature.html>
- [2]. <http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- [3]. <https://github.com/bikz05/object-detector>