Goal

1. The result of t-SNE visualisation for the two video sequences.
2. An idea for labelling different objects for images, based on the t-SNE results.

My approach

1. Visualizing one video sequence for testing

I follow Karpathy's method, use Caffe's trained 1000 class model to extract 4096 features from fc7 layer. The values are normalized to 0~1 and passed through the Barnes-Hut t-SNE. Following Karpathy's code, I implement a function that maps images onto a grid map according to their xy coordinates in Python, which visualize two dimension outputs of t-SNE. The result is pretty good, frames with different contents are well separated.

For labelling, my first instinct is clustering. I try serveral clustering methods that were already implemented in scikit-learn. The DBSCAN turns out to be the best way to label the data. It's a density-based clustering method, can be any shape and is robust against noise. In this case, maybe the noise can be extract as useful information considering it is kind of an outlier from the clustered data.

2. Write a script to download data

I use the command “find . -type f -exec ffmpeg -i "{}" -r 1 {}%03d.jpg \; ” to sub-sample all the videos to image sequences at one frame per second rate, then I write a bash script to repeat the routine and compress the images. Due to the insufficient capacity of AWS EC2 instance, I can only run the script on one day amount of videos, and then manually download the compressed files.

3. Data sub-sampling

Due to the computational memory constraint, data sub-sampling is needed. I did some sampling methods research, such as tournament selection, but then I realize the most important part is how to calculate the data similarity. After some research I can't seem to find useful study about the similarity measurement about the feature vectors extracted from CNN, so I then use the simplest – Euclidean distance. My approach will be demonstrate below.

4. Regression

After the sub-sampling, it's important to find a way to map the unselected data back to where they belong. But t-SNE is a non-parametric mapping algorithm, which means it can't predict the incoming test points by the learned result. Author of t-SNE proposed a parametric t-SNE method using feed-forward neural network, because a deep neural network can parametrize arbitrarily complex non-linear functions, which maps the high dimensional data space X to low dimensional latent space Y. But here instead I use a SVM regressor, the result seems pretty well enough though.

5. Final data labelling

I try to map 45000 images which are the amount of one-day videos and cluster them for labelling, but the best result will only produce about 25 clusters, if we extract one image per cluster that seems to be way little amount of useful data, so I

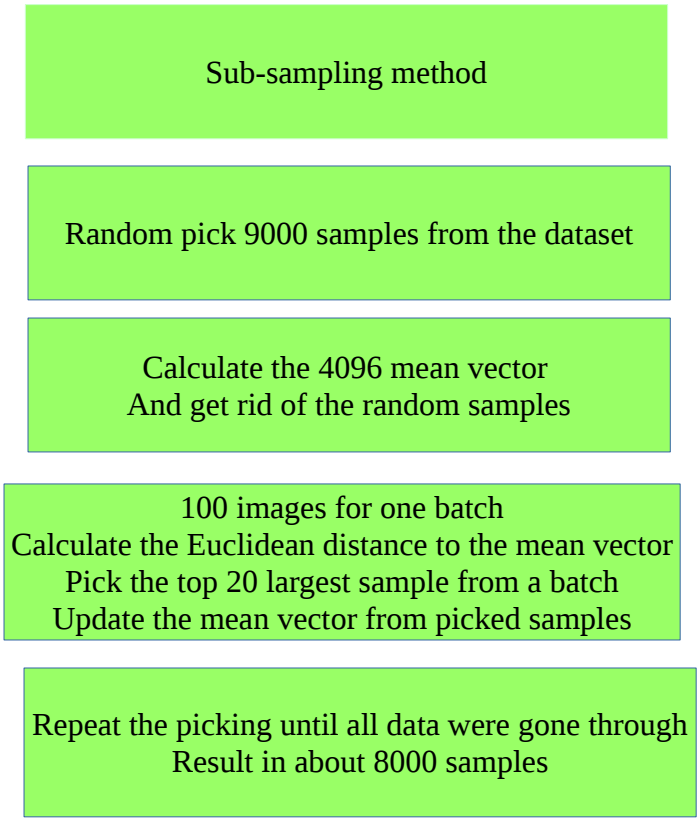
instead input 2000 images at one time and cluster them, I also store the different labelling result from DBSCAN and mean-shift clustering.

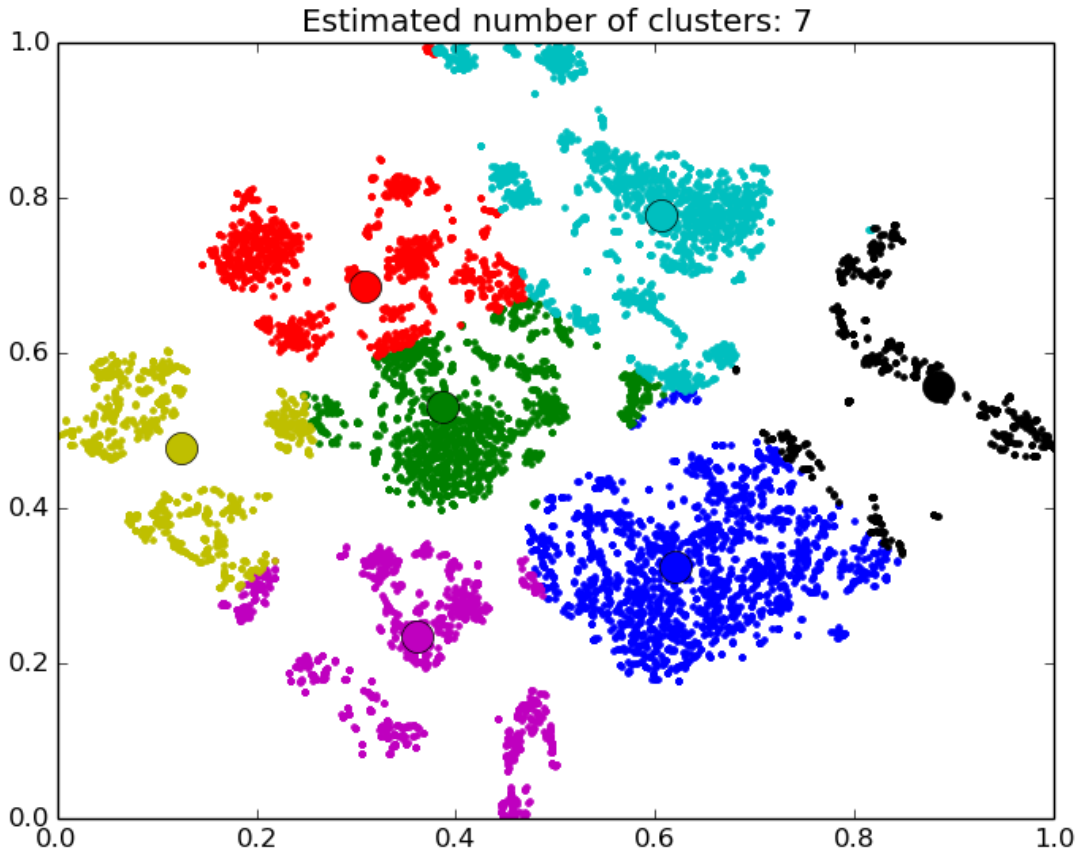
6. Discussions

The sub-sampling and labelling method have a lot of space to be improved, such as using a better similarity measurement and a better selecting method, or use a better clustering algorithm to separate data points.

I am not sure how the data is going to be used, but if it's going to be used for RNN to classify human behavior, isn't the strategy of picking one image per pool won't be enough for training? Because RNN requires a sequence of temporal related data right? Just one image from one kind of behavior sequence seems not sufficient for RNN to learn I guess. This leads to another question, should I use motion feature? Since it requires a pre-trained CNN to extract features, it's reasonable to need a pre-trained RNN or CNN which is able to understand motion in order to extract features, which is kind of a paradox that we need a trained model that understand motion to extract data to train a motion model? That is pretty much my questions about this project.

I've been doing this project in my free time beside works, plus I don't have qualified hardware to do this at my home, so it's all after work time during weekdays. Sorry for taking such a long time! I won't be very upset if I failed the test because I actually learned so much from this project. Before this I haven't write any Python program or bash script before. So is sampling method, I've always known statistics is a very important area in machine learning, actually machine learning came from statistics? Though I haven't spend time to take a class on statistics, but at least now it's on my to-do list! And t-SNE, it appears in so many deep learning paper, it's such a cool visualization technic, I really learned a lot from it. Also I've been studying deconvolutional neural network in order to know my trained model more, I wonder what are your suggestions on this?





Sub-sampled 8000 data and clusterd by mean-shift

My pedestrian detector result using t-SNE