Dynamic Motion and Static Poses based Gesture Recognition via Depth Map

August 15, 2016

1 Team details

- Team name
  XJTUfx
- Team leader name
  Wei Shenghua[1]
- Team leader address, phone number and email
  Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
  (+86)-18710702749
  weishenghua@stu.xjtu.edu.cn
- Rest of the team members
  Zhang Zhi[2]
- Affiliation
  (1) School of Software Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
  (2) Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

2 Contribution details

- Title of the contribution
  Dynamic Motion and Static Poses based Gesture Recognition via Depth Map
- General method description
  We proposed a method to classify gestures only with depth videos. We first calcuate one motion history image and one static posture image from each depth video, which stacks the dynamic motion of a gesture and the static poses of a gesture respectively. Then we extract features from these two images with two different CNN networks. We concatenate the features

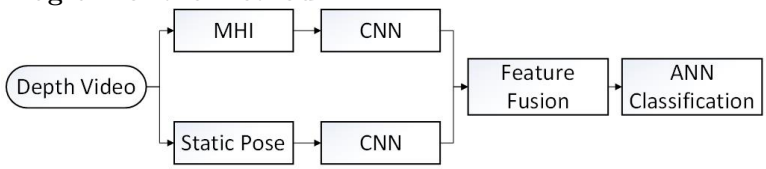
from the CNN networks as the final representation of a gesture. Finally, we train an one-hidden layer artificial neural network to classify the gesture using the concatenated feature.

- References
[1] Bobick A F, Davis J W. The Recognition of Human Movement Using Temporal Templates[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2001, 23(3):257-267.
[2] Chalearn gesture dataset (cgd2011). ChaLearn, California, 2011
[3] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in Neural Information Processing Systems, 2012, 25(2):2012.

- Representative image / diagram of the method
Representative image
motion history image

static posture image

The pictures above are motion history images and static posture images. We choose 3 differnt kind of gestures: class 1, class 124 and class 180. Each column is the motion history image and static posture image of each gesture from different classes.
Diagram of the method



- Describe data preprocessing techniques applied (if any)
We removed the background of the depth videos firstly.

3 Visual Analysis

3.1 Gesture Recognition Stage

3.1.1 Gesture representation

We only use the depth videos to classify gestures. We calculate the motion history image from the depth video to stack the motion of the gesture in one image. We also calculate the static posture image from the depth video to stack all the static poses of the gesture in one image. Here's the detail of motion history image and static posture image.

Dynamic motion

We use the depth video of the gesture to calculate the motion history image[1], in order to stack all the motion of a gesture in one single image. Fisrt we calculate the frame difference between two consecutive depth video frames. We subtract frame t-1 from frame t. Because of the poor quality of depth videos, the frame difference is quite noisy. We use some image processing techniques like medium filtering and open operation to eliminate the noises of frame difference.Then we stack all the image differences in one image by adding them all up. Finally we get a motion history image of a gesture to represent the dynamic motion in one image.

Static poses

We also use the depth video of the gesture to calculate the static posture image[2], in order to stack all the static poses of a gesture in one single image. For frame t, we first calculate the moving part at this moment. We subtract frame t from frame t-1 and frame t from frame t+1 respectively. Then we add these two image differences up as the moving part of frame t. We also subtract frame t from the first/original frame to calculate the difference to the original posture. Next we subtract moving part from the difference to the original posture to identify the pixels that are not moving but different from the original frame, which represents the static posture of frame t. Finally, we stack all the static postures of each frame by add them all up. We get a static posture image of the whole gesture to show all the static poses of a gesture in one image.

3.1.2 CNN-extracted features

We have already calculated two images to represent the dynamic motion and static posture of a gesture. Next we are going to extract features from these 2 images by 2 CNN networks. We train two CNN networks for motion history images and static posture images respectively. Then we use the trained CNN networks to extract features for motion history images and static posture images. We combine the features from 2 different networks as the final feature representation of a gesture instance.

Image resize

The size of depth video is 320x240. Thus the motion history image and the static posture image are 320x240, too. We resize all the images to 64x64.

CNN network structure

We adopt the Alexnet[3] for the motion history image and the static posture image both. We modified the input size and the output category number of the original network. The input size of the network is changed from 224x224x3 to 64x64x1. The output category number of the network is changed from 1000 to 249, which is the category number of the IsoGD dataset.

CNN network training

The architecture of our model consists of two CNNs, one for extracting features from motion history image and one for extracting features from static posture image. These 2 CNNs use the same structure mentioned before and they are trained separately.

CNN feature extraction and feature fusion

For each gesture sample, we put its motion history image into the first CNN to get the CNN-learned feature. We get the outcome of the 7th-layer of the network, which is a 4096-d vector. We also get another 4096-d vector of static posture from another CNN in the same way. These two 4096-d vectors are concatenated as the final representation of a gesture.

3.1.3 Classification method

After the CNN-feature extraction and feature fusion, each gesture is represented as a 8192-d vector(concatenated from two 4096-d vector). We classify gestures with an one-hidden-layer artificial network. The input of the ann is the 8192-d vector. The hidden layer has 1024 neurons. The output layer has 249 neurons for 249 categories.

3.1.4 Data augmentation

The original IsoGD dataset is quite unbalanced. Besides, in order to train CNN networks, we need more samples. We extended the original motion history image and static posture image. We use some image processing techniques like image rotation, image zooming, image translation, image contrast adjustment to generate more samples. We extend both motion history images and static posture images to 2000 samples per class, which also solved the unbalanced dataset problem.

3.1.5 Method complexity

We trained 2 CNNs to extract features from motion history image and static posture image. We also trained a simple one-hidden-layer ann to classify the CNN-extracted features. The most time consuming part is the CNN training.

3.2 Data Fusion Strategies

As mentioned before, we stack a depth video into 2 different images, and concatenate the features from 2 different CNNs.

3.3 Global Method Description

- Which pre-trained or external methods have been used (for any stage, if any)

  We used some depth video processing code in CGD 2011 sample code. We also used matconvnent(a CNN tool in matlab) to train our modified CNNs.

- Qualitative advantages of the proposed solution

  After we got the validation data labels, we tested on the validation data and get the highest accuracy of 36.69% with our model.

- Novelty degree of the solution and if it has been previously published

  We used motion history image and static posture image to represent gesture, which are published before. We used some image processing techniques to get better result which overcomes the noise of the depth video. We modified the Alexnet to extract features from images. We fuse the features of 2 different images from 2 CNNs. The fused CNN feature contains both motion information and static poses of a gesture.Fianlly we use an ann to classify gestures, which produce better results than SVM according to our experiment. This work hasn't been published yet.

4 Other details

- Language and implementation details (including platform, memory, parallelization requirements)

  Language: Matlab
  Platform: Matlab 2014a
  Memory: 32 GB
  GPU: Tesla K40
  Cuda Support: Yes(for matconvnet faster implementation)
  OS: Windows 7

- Training/testing expended time?
  We spent about 12 hours to calculate all motion history images. It took almost the same time to calculate the static posture images. We spent 3 hours for data augmentation. We spent 15 hours to train each CNN for 20 epochs. We spent about 3 hours to extract features from CNNs and fuse them. We spent 20 minutes to train the final ann because of the simplicity of the network.

- General comments and impressions of the challenge? what do you expect from a new challenge in face and looking at people analysis?

The IsoGD competition is quite challenging. Because of the large size of data, it takes us a long time to try any new method. We wanted to make good use of RGB video but we didn't have enough time to implement our ideas. It's not easy to complete the mission in such little time. But it's also a good experience,too. We hope the future looking at people analysis challenge can provide different data sources and extend the time a little bit. Maybe with longer time, the participants can make full use of all the data.