# Using Generic Deep Volume Features for Large Scale Continuous Gesture Recognition

August 17, 2016

## 1 Team details

- Team name: TARDIS
- Team leader name: Necati Cihan Camgoz
- Team leader address, phone number and email: CVSSP, University of Surrey, 10BA00; n.camgoz@surrey.ac.uk
- Rest of the team members: Simon Hadfield, Oscar Koller, Richard Bowden
- Team website URL (if any): -
- Affiliation: University of Surrey, RWTH Aachen University

## 2 Contribution details

- Title of the contribution: Using Generic Deep Volume Features for Large Scale Continuous Gesture Recognition
- Final score: Validation Jaccard Index Score: 0.280859681433
- General method description: We have trained an end-to-end deep network for gesture recognition (jointly learning both the feature representation and the classifier). The network performs three-dimensional (i.e. space-time) convolutions to extract features related to both the appearance and motion of the data. Space-time invariance is encoded via pooling layers. The earlier stages of the network are partially initialised using the work of Tran et al. before being adapted to the task of gesture recognition.
- References: Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, Learning Spatiotemporal Features with 3D Convolutional Networks, IEEE International Conference on Computer Vision (ICCV) 2015, Santiago, Chile.

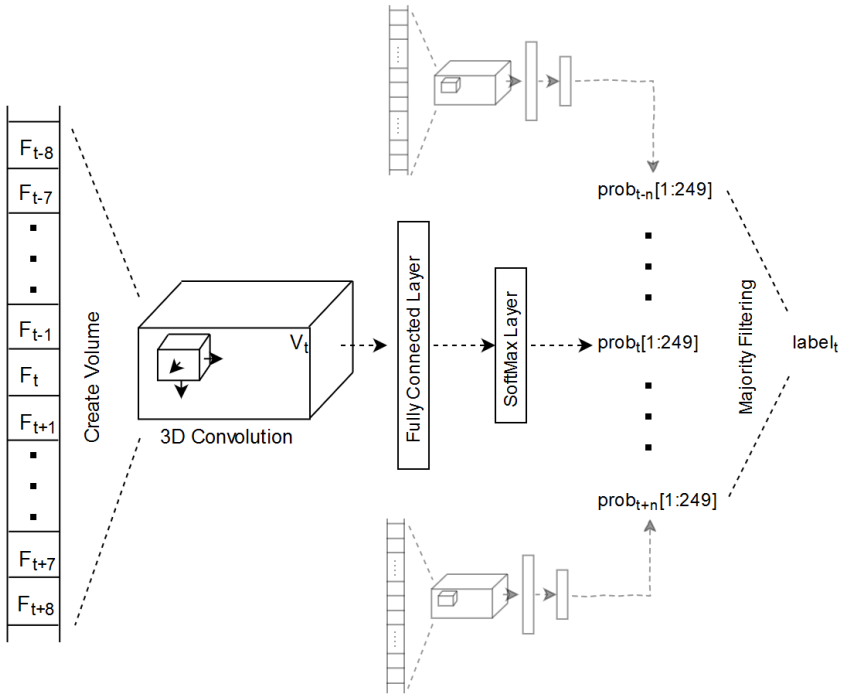


Figure 1: Flow chart of the system

- Representative image / diagram of the method: See Figure 1.
- Describe data preprocessing techniques applied (if any): -

# 3 Visual Analysis

## 3.1 Gesture Recognition (or/and Spotting) Stage

### 3.1.1 Features / Data representation

Describe features used or data representation model FOR GESTURE RECOGNITION (OR/AND SPOTTING) STAGE (if any): A learned representation based on the repeated application of space-time convolutions, nonlinearities and poolings. These features encode both appearance and motion.

### 3.1.2 Dimensionality reduction

Dimensionality reduction technique applied FOR GESTURE RECOGNITION (OR/AND SPOTTING) STAGE (if any): None

#### 3.1.3 Compositional model

Compositional model used, i.e. pictorial structure FOR GESTURE RECOGNITION (OR/AND SPOTTING) STAGE (if any): None

#### 3.1.4 Learning strategy

Learning strategy applied FOR GESTURE RECOGNITION (OR/AND SPOTTING) STAGE (if any): Classification is performed by a series of fully-connected layers and a softmax cost function. The parameters of the classifier are learned via stochastic gradient descent.

#### 3.1.5 Other techniques

Other technique/strategy used not included in previous items FOR GESTURE RECOGNITION (OR/AND SPOTTING) STAGE (if any): To perform spotting, the deep-volume features are computed on a sliding spatio-temporal volume. The output predictions are then refined via 2 stage majority filtering to remove noise.

#### 3.1.6 Method complexity

Method complexity FOR GESTURE RECOGNITION (OR/AND SPOTTING) STAGE: The computational complexity complexity is cubic with respect to the number of classes.

### 3.2 Data Fusion Strategies

List data fusion strategies (how different feature descriptions are combined) for learning the model / network: Single frame, early, slow, late. (if any): The fusion of different feature types (e.g. appearance and motion features) is handled internally as part of the learned representation for the network. As such it happens "early" (before classification), but it is more complicated than "single frame" fusion because the space-time convolution and pooling combine features across time.

### 3.3 Global Method Description

- Which pre-trained or external methods have been used (for any stage, if any): The C3D network of Tran et al. was partially used as the starting point for learning out network.
- Which additional data has been used in addition to the provided ChaLearn training and validation data (at any stage, if any): None
- Qualitative advantages of the proposed solution: As usual with deep learning solutions, one of the largest advantages is that the learned features will

be specifically adapted to the task. In addition, the space-time convolutions in this approach allow both appearance and motion features to be encoded by the system.

- Results of the comparison to other approaches (if any): None
- Novelty degree of the solution and if is has been previously published: As far as we are aware, this is the first time that convolutional volume features have been adapted to the task of continuous gesture recognition.

# 4 Other details

- Language and implementation details (including platform, memory, parallelization requirements): All code was compiled and run on Ubuntu 14.04. The code uses matlab (tested on 2016a) and the caffe command line interface. The code requires an nvidia GPU with 12GB of VRAM (although less is needed if the batch size is reduced). No other special hardware is required.
- Human effort required for implementation, training and validation?: The process was fully automated, there was no human-in-the-loop input.
- Training/testing expended time?: 27 hours to train each model. 5 hours to extract features on the test set.
- General comments and impressions of the challenge? what do you expect from a new challenge in face and looking at people analysis?: The deadlines were extremely tight given the huge amount of data. This was compounded by the validation data being held back for such a long time. There was little possibility for innovation because by the time we saw our results, there was no time to try anything else.