Gesture Recognition With Varying Frame Rates

096290 - Course Final Project Report

March 8, 2022

Roy Matza **Yael Kiselman** **Ariel Abramowitz**

1. Introduction

Surgical gesture recognition is a common task based on various data, mainly video and kinematics that are recorded in the operating room from surgeons or as part of simulations in order to obtain synthetic data for experiments. In this project we wanted to check the correlation between temporal resolution of the data and the network performance. We found that the network is quite indifferent to integration of several frame rates. Consequently, we had checked hybrid networks that involve both 1D convolutions and RNNs which showed no significant improvement.

1.1. Motivation

We hypothesize that a model can learn different patterns given different frame rates. For example, high resolution frame rate can capture the nuances of suturing action (like tying a knot), whereas low resolution rate can capture the movement of a hand between the different knots. SlowFast architecture by Feichtenhofer, Christoph, et al. [1] took advantage of the concept of different frame rates and incorporated two different trainable paths. SlowFast architecture was designed for action *classification*, which doesn't account for the temporal dimension. In gesture recognition, the sequential nature of the input has to be preserved, i.e., one has to perform action *segmentation*. MS-TCN++ by Li et al. [3] and its predecessors (MS-TCN, TCN…) are designed for the purpose of action segmentation and have been proven effective on surgical data, both kinematic and video. Our project aims at merging or altering the two approaches.

2. Materials and Methods

2.1. Dataset and metrics

We used a manually collected *kinematic* dataset by a research team at the Technion and common metrics which are introduced in [2].

2.2. Data preprocessing[1]

In the course of experimenting we had tried to alter the data in order to see if it remedies some inaccuracies during the measurements. A common practice of smoothing the data is interpolation, which has already been performed on surgical data in [4]. Other well known techniques are normalization and some feature engineering on the data. We experimented with the following methods of data preprocessing:

(1) Interpolation: Only one of n frames of the original remained the same, while the other frames were interpolated (using linear interpolations). This had no effect on the results and worsened as n>3. (2) Calculating gradients: motion could be better expressed by velocities. We used the gradients of position data and angular data and fed them the models. Interestingly, the newly processed data alone had the same effects as previous data, while the combined types of data had a negative effect on the models. (3) Normalization: We found that normalization had dramatically lowered the performance of the model.

2.3. Proposed architectures (see Fig. 1)

"SlowFast" RNN: We used two parallel multi-layer bidirectional RNNs, each with a different hidden dimension, and each handles data at different frame rates. One RNN is the "slow" path and the other RNN is the "fast" path. We made the fast path relatively lightweight as in *SlowFast*. First, a batch of kinematic data is sampled from which we sub-sample frames. The original batch is fed into the fast path and the sub-sampled batch is fed into the slow path. We upsample the slow path output and concatenate it with the fast path output. Finally, outputs are fed into a fully-connected layer which outputs a score for each class and each frame. We also tested the effect of adding connections between paths (Lateral connections) by summing the fast path into the slow between subsequent layers, as in SlowFast.

[1] Preprocessing scripts are added to the submission

Hybrid-TCN-RNN: We used two networks in parallel, one of them was a simple RNN, while the second one was a TCN-like network that used constant dilation. The idea is that the 1D convolution would learn local patterns of the data, whereas the RNN would be responsible for the long-term memory. Like the previous architecture the both outputs are concatenated and fed into a fully connected layer.

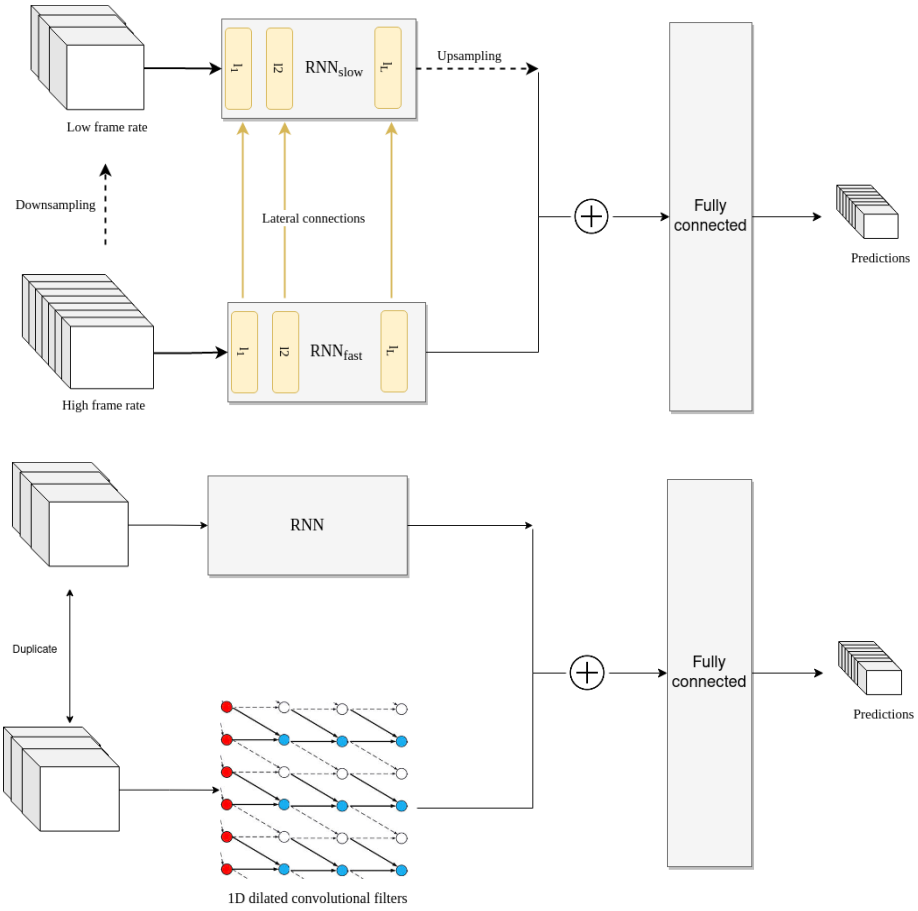

Figure 1. *Proposed architectures: 'SlowFast' RNN (above) and Hybrid-TCN-RNN (below)*

3. Experiments and Results

3.1. *Choosing Model and Tuning Hyperparameters*

We have tried over 200 configurations of the methods written above. As a base line, the models which provided the final results were trained for 40 epochs with Adam optimizer with learning rate of 0.0021 and a batch size of 5. Sample (frame) rate was 2 fps (15 Hz) . RNN was a 4-layer GRU. For the SlowFast configuration, we used a hidden dimension of 33 for the slow path and 8≈33/4 for the fast path, and the frame rate ratio between the paths was 2. Dropout of 0.35 was added. We didn't conduct any data preprocessing or added lateral connections since they got low performance relative to the original state. We tried to train using the tool data but found no empirical benefit in doing so. After some explorations on one fifth of the kinematic dataset, we performed cross-validation with 5 folds in order to determine which model is the best.

3.2. *Results and Discussion*

Model (Slow/Fast [Hz])	Accuracy	Edit	F1-macro	F1@10	F1@25	F1@50
"SlowFast" RNN (15/15)	**82.39**	**84.31**	**78.01**	**87.8**	**83.89**	**68.23**
"SlowFast" RNN (15/7.5)	82.16	83.56	77.77	87.28	83.22	67.93
Hybrid-TCN-RNN (dilation = 3)	81.6	78.49	77.28	82.1	77.91	62.95
Hybrid-TCN-RNN (dilation = 2)	80.88	77.13	76.63	80.37	76.28	60.99
Hybrid-TCN-RNN (dilation = 1)	81.68	80.51	77.47	83.49	79.2	64.34
Base RNN	81.76	83.74	77.42	86.98	82.53	67.4

Table 1. *Results of our model averaged over 5 folds*

Table 1 shows the best results averaged over 5 folds. The baseline was a simple RNN (GRU) initiated with the same (relevant) hyperparameters. For the hidden dimension ratio, we took the sum of the hidden dimensions as the hidden dimension of the RNN. The results imply that a dual path network with different frame rates has a low impact when using RNNs (as opposed to convolutions

in *SlowFast*) or TCNs. However, for some parameters our network did show better results. Surprisingly, in the SlowFast RNN the same ratio between frame rates got the best results.

References

[1] *Feichtenhofer, Christoph, et al. "Slowfast networks for video recognition." Proceedings of the IEEE/CVF international conference on computer vision. 2019.*

[2] *Basiev, Kristina, et al. "Open surgery tool classification and hand utilization using a multi-camera system." arXiv preprint arXiv:2111.06098 (2021).*

[3] *Li, et al. "MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation". IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

[4] *Todd Edward Murphy. "Towards Objective Surgical Skill Evaluation with Hidden Markov Model-based Motion Recognition". Phd Thesis.*