Query By Video for Surgical Activity

Computer Integrated Surgery, Spring, 2018
Felix Yu, Gianluca Croso
under the auspices of Tae Soo Kim, MsE, Haider Ali PhD, Gregory Hager PhD, Swaroop Vedula MD

Introduction

We developed a method of encoding surgical activity clips with tool annotations in order to effectively compare videos to each other in terms of the action being performed. To that end, we created a neural network pipeline to encode clips into features that capture spatio-temporal information, followed by nearest neighbor querying over an existing database of surgical clips to similar activities.

We do this as an intermediate step to providing automated feedback for surgeons. Currently, providing such feedback manually after an operation is both costly and difficult for various reasons.

Problem

There are multiple steps needed to automate feedback.

1. Segment videos into individual activity clips.
2. Develop database of expert commentary on these clips.
3. Create video encoding and similarity metric that allows querying for similar videos on this database.
4. Use existing commentary on these similar clips to generate skill related feedback on query clip.

We focus on the third problem, creating an encoding that can at least differentiate between different phases in a surgery. In particular, we focus on cataract surgeries. Although there are publications addressing phase classification and tool recognition in surgical videos, we are not aware of the existence of a competing method trying to solve this exact problem.

Solution/Methods

Our approach has three components. A convolutional neural network that captures spatial information (SCNN), a recurrent neural network that captures temporal information (TCNN), and a nearest neighbor querying of database for classification.

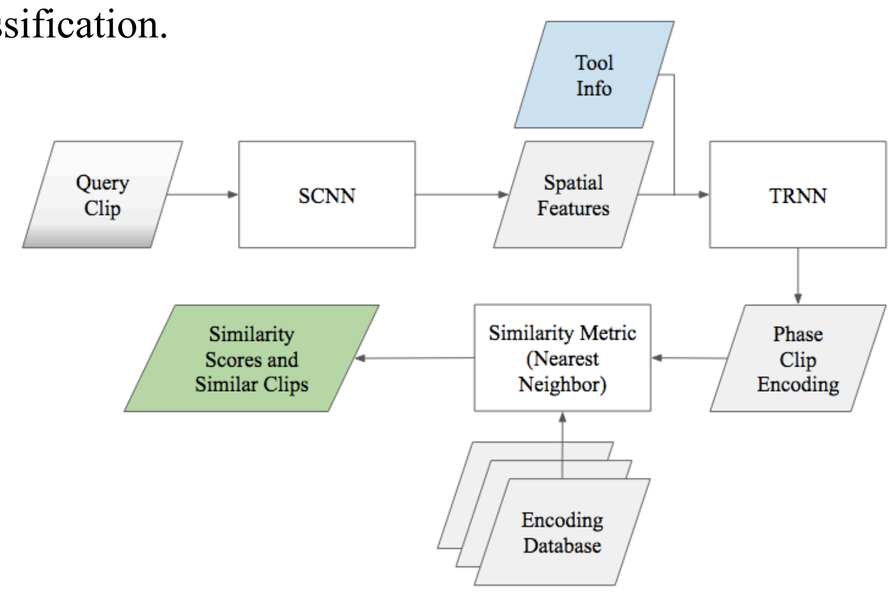

Fig 1: The flowchart for the pipeline. A query clip video will have the spatial and temporal features encoded through two neural networks, and then the database will be queried for similar clips.

The SCNN based on SqueezeNet [1] is trained for phase classification given a single frame of the video using cross entropy loss. The TRNN takes in features from SCNN across timesteps in a clip along with tool labeling to capture temporal information. The latter is trained using triplet loss, making encodings of different classes more separable.

$$L_{CE} = \sum_{i=1}^{10} b_i p(y_i) \qquad p(y_i) = \frac{1}{Z}\exp\{l_i\} \qquad L_T = \left[\|f_a - f_p\|^2 - \|f_a - f_n\|^2 + \alpha\right]_+$$

Eq 1 and 2: Formal definitions for cross entropy loss and triplet loss.

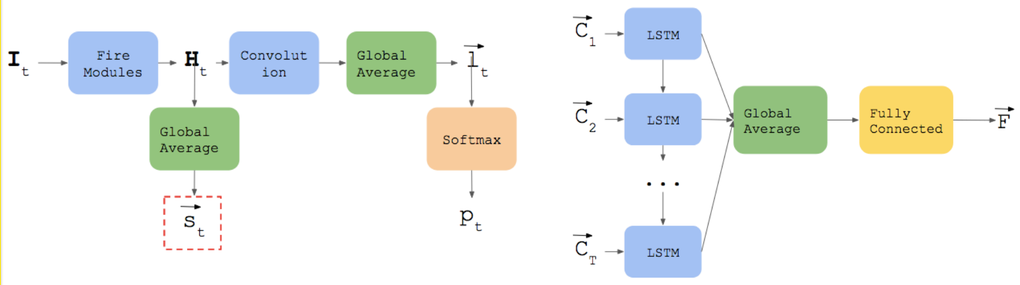

Fig 2: Pictoral representations of the SCNN and TRNN. Left is the SCNN, right is the TRNN.

Once these models are trained, we create a database of encodings using our training data. Given a new query clip, we can find the closest video (based on Euclidean distance) and classify the query clip to be the same class.

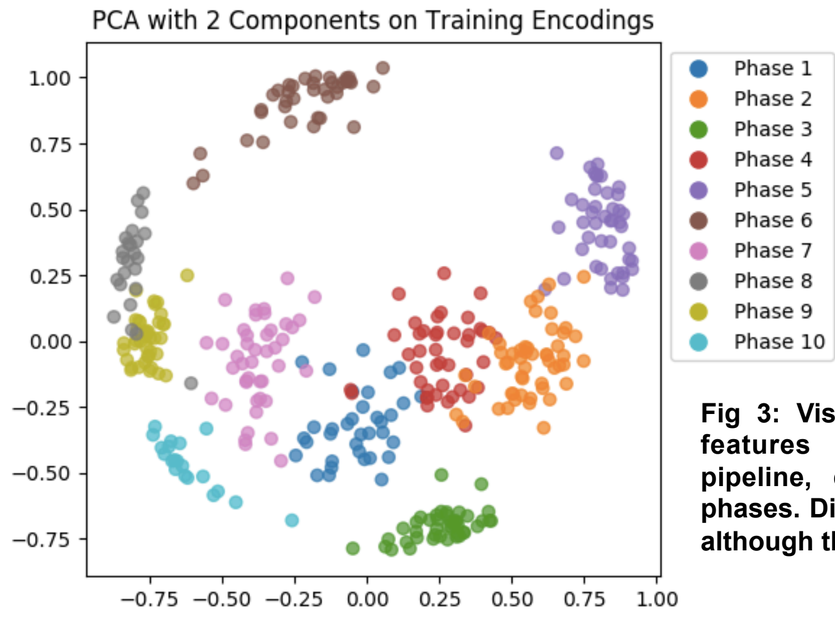

Fig 3: Visualization of our database features generated through the pipeline, colored according to the phases. Distinct clusters can be seen, although there are regions of overlap.

Outcome/Results

In order to quantify the usefulness of including tool information, we trained two versions of our pipeline, one without using tool information and one with. We achieved classification accuracy of 35.6% and 71.1% respectively. Below is also the precision/recall for each phase, as well as the confusion matrix for the latter model.

	Precision Video	Recall Video	Precision Video + Tool	Recall Video + Tool
Phase 1	0.000	0.000	1.000	0.364
Phase 2	0.727	0.615	1.000	0.769
Phase 3	0.538	0.538	0.733	0.846
Phase 4	0.444	0.154	0.917	0.846
Phase 5	0.417	0.769	0.706	0.923
Phase 6	0.229	0.786	0.387	0.857
Phase 7	0.154	0.133	0.789	1.000
Phase 8	0.083	0.067	0.667	0.133
Phase 9	1.00	0.278	0.688	0.611
Phase 10	0.667	0.200	1.000	0.800

Miscellaneous

Future Work

- Writing manuscript to submit for publication either in PlosOne or JAMA Open.
- Work will be continued by cataract group with sparse involvement by us.
- Future work includes training model on larger dataset, investigating skill related encodings, and obtaining more fine-grain tool annotations.

Lessons Learned

- Start simple and don't unnecessarily over-complicate models.
- Things may not work first try.

Credits

The SCNN based on SqueezeNet was provided by our mentor Tae Soo Kim. All other aspects were implemented equally and together by Felix Yu and Gianluca Croso. Documentation of code was mostly done by Gianluca, while Felix focused more on writing the report.

Citations

[1] F. Iandola, S. Song, M. Moskewicz, K. Ashraf, W. Dally J., K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size, ICLR Conference, 2017.

Acknowledgements

We would like to thank those involved in the Cataract Project for mentoring us throughout this semester, as well as Dr. Shameema Sikder for providing support with the Wilmer Eye Institute Pooled Professor's fund.We would also like to thank those in Gregory Hager's lab for providing us with feedback, and finally Dr. Taylor and Ehsan for facilitating this class.