Neural Network Recognition of Medical Procedures in Recorded Video using TensorFlow

Conner Pinson
Electrical Engr. & Computer Science
Vanderbilt University
conner.pinson@vanderbilt.edu

Richard Paris
Electrical Engr. & Computer Science
Vanderbilt University
richard.a.paris@vanderbilt.edu

Bobby Bodenheimer
Electrical Engr. & Computer Science
Vanderbilt University
robert.e.bodenheimer@vanderbilt.edu

Problem

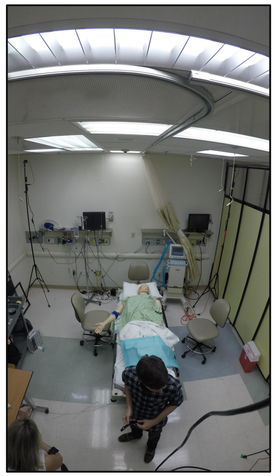
When an ambulance responds to an emergency, the EMTs on board perform numerous procedures to keep the patient stable on the way to the medical center. This situation is high pressure and decisions on what procedure to perform can come and go fast. When the patient is passed off to the emergency room doctor, the EMTs verbally recount all of the procedures they performed en route. These accounts can be flawed due to the amount of procedures performed or the EMT's inability to recall exact details from a high stress situation.

Our goal is to automate this process. Using new machine learning techniques, we are exploring the possibility of a camera and computer combination being able to recognize a wide breadth of medical procedures. Based on prior experience with machine learning techniques and the TensorFlow library, we predict that we will be able to train a model with reasonable accuracy given the appropriate amount of training data.

This project seeks to prove the efficacy of this approach by training a TensorFlow model on the amount of data (Paris et al. 2019) we currently have. We also wish to compare the approach described in this project and another study using xy-coordinates for major body parts generated by OpenPose (open source software that assigns coordinates to primary locations on each body in the scene).

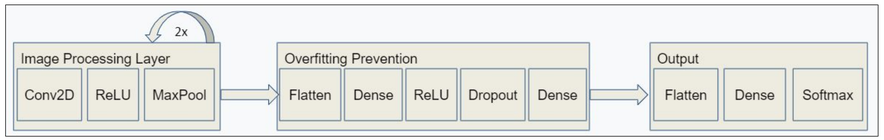
Data Set and Preparation

The training data that acts as the TensorFlow input consists of around 21 hours of footage recorded from 4 different angles at 30 frames per second. We focused on training from one camera angle since, if practical, it would be a more deployable option. These videos were split into their constituent frames using FFMPEG and distributed into a folder that corresponds to the procedure that is being performed in the frame. The procedure labels were scraped from a corresponding csv file.



Example of a still frame from the training data. This is while an EMT is demonstrating wrapping a head wound.

Another example showing a demonstration of a CPR compression in progress.



VIsual representation of the structure of the neural network. Note that the first block is run a total of 3 times.

References: Paris, R.A., Sullivan, P., Heard, J., Scully, D., McNaughton, C., Ehrenfeld, J.M., Adams, J.A., Coco, J., Fabbri, D. and Bodenheimer, B., 2019, March. Heatmap generation for emergency medical procedure identification. In Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling (Vol. 10951, p. 1095130). International Society for Optics and Photonics.
TensorFlow, the core open source library to help you develop and train ML models: https://www.tensorflow.org/

Acknowledgements: Funding for this project received from the Department of Defense Contract Number W81XWH-17-C-0252 from the CDMRP Defense Medical Research and Development Program. We would also like to thank Dan Fabbri for his help.

Neural Network Structure

We used TensorFlow and Python for the construction of the neural network. The Keras API provides the Sequential model which provided the basis for our structure. The data was thrice passed through a combination of a convolutional layer (to extract high level features), a rectified linear unit (ReLU) activation function (to add the needed element of nonlinearity to the neural network), and a max pooling layer (for reducing the spatial size of the convolved figure. This decreases computational requirements and helps extract rotational and positional invariant features). To prevent overfitting of the data, we flatten the 3D features maps to 1D feature vectors. Then we densely connect the layers to make the flatten output more manageable. Finally, we pass the data through another ReLU activation function, randomly dropout 50% of the data, and densely connect the outputs into one neuron. To output the data, we flatten the output back into a 1D vector. Then we densely connect the layers again to reduce the number of neurons presented by the flatten before inputting them into the output layer. The final output layer uses the softmax "most likely candidate" output to present the label the neural network has predicted.

Results and Future Research

Solely using frames from the collected data, we were able to get an accuracy of about 18.67%. This means that 18.67% of a new set of data is correctly labeled by the neural network. Considering the relatively small data set we trained on, this was an expected result. We believe we may be able to up this accuracy considerably if we were to consider the temporality of the data. It would also be worth exploring the effectiveness of using a combination of video data and the xy-coordinate data generated by the OpenPose skeletal recognition software.