Emergency Clinical Procedure Detection With Deep Learning

Lingfeng Li[1], Richard A. Paris[1], Conner Pinson[1], Yan Wang[1], Joseph Coco[2], Jamison Heard[3], Julie A. Adams[4], *Senior Member, IEEE*, Daniel V. Fabbri[2], and Bobby Bodenheimer[1], *Senior Member, IEEE*

***Abstract*— Information about a patient's state is critical for hospitals to provide timely care and treatment. Prior work on improving the information flow from emergency medical services (EMS) to hospitals demonstrated the potential of using automated algorithms to detect clinical procedures. However, prior work has not made effective use of video sources that might be available during patient care. In this paper we explore the use convolutional neural networks (CNNs) on raw video data to determine how well video data alone can automatically identify clinical procedures. We apply multiple deep learning models to this problem, with significant variation in results. Our findings indicate performance improvements compared to prior work, but also indicate a need for more training data to reach clinically deployable levels of success.**

I. INTRODUCTION

Possessing and communicating accurate patient information is critical to achieving optimal medical outcomes. Unfortunately, too often emergency medical services (EMS) does not communicate complete information to a treating hospital [1]. This lack of communication can lead to inferior patient outcomes as the initial triage of a patient's condition can be done incorrectly at the receiving hospital [2]. This paper discusses a component of a noninvasive system that could potentially detect automatically what clinical procedures have been performed on a patient. Ideally, this system would supplement current care procedures by providing improved information on patient care to receiving hospitals, with the goal of improving a patient's initial triage level.

Heard et al. [2] made initial forays into this area by presenting a system that employed information from multiple types of sensors, including video, to categorize clinical procedures. Building on work in human activity recognition from multiple sensors [3], this work used contextual information provided by video sources to locate a medic's hands. Based on this location and other sensor information, this work represented a first step at clinical procedure identification. Its highest accuracy was achieved when the algorithm algorithm knew the active body region to which the procedure was applied (18% accuracy without body region knowledge and 40% accuracy with perfect body region knowledge).

This paper represents a new effort into this problem and only uses video data. It is motivated by recent developments in video classification and recognition in deep learning [4-6]. We use convolutional neural networks (CNNs) on raw video data to detect different clinical procedures performed during EMS transport. We attempted this because we want to see how far video data alone can take us, and to assess how much video data is necessary to achieve adequate performance from these data sets alone. There is an operational advantage from video data in that would not require paramedics to wear sensors. On the other hand, video data suffers from occlusion problems and noise due to lighting changes, and thus may have other difficulties for learning algorithms. Nonetheless, understanding how well video data by itself can work for classification is an important step in the development of an automatic clinical procedure system.

II. RELATED WORK

Karpathy et al. [8] demonstrated the effectiveness of convolutional neural networks (CNN) on video classification tasks. They explored four different models for fusing information over temporal dimension through networks. All models exhibited strong capabilities for classifying video clips. Of their approaches, we primarily explored the single frame approach and the late fusion approach, as illustrated in Figure 1.

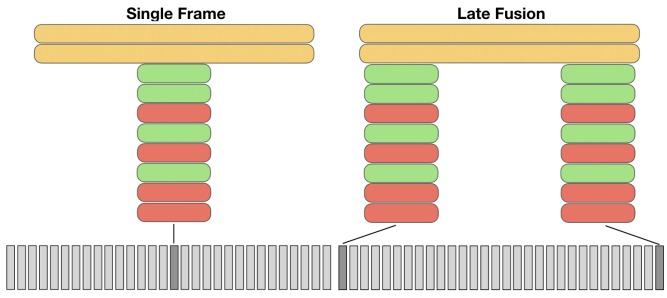

Fig. 1. Illustration of the Single Frame and Late Fusion Structures. The single frame model classifies based on one frame of the video stream (depicted at the bottom), whereas the late fusion model uses multiple temporally close frames. Red and green boxes together represent a mixture of convolutional layers and utility layers; yellow boxes represent fully connected layers.

Ng et al. [6] and Donahue et al. [5] demonstrated the use of Long Short Term Memory (LSTM) for video classification tasks. These two groups of researchers processed individual

This work was supported by the Department of Defense Contract Number W81XWH-17-C-0252 from the CDMRP Defense Medical Research and Development Program.

[1]Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, Tennessee

[2]Vanderbilt University Medical Center, Nashville, Tennessee

[3]Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY

[4]Collaborative Robotics and Intelligent Systems Institute, Oregon State University, Corvallis, Oregon

Corresponding author email: bobby.bodenheimer@vanderbilt.edu

frames with CNNs to aggregate information from frame data, and then the aggregated data is passed to the LSTM network for information summation. Results shown by Ng et al. and Donahue et al. suggest that LSTMs can achieve better video classification results than CNN methods if tuned well.

III. Data Collection and Dataset Creation

All experimental data were collected in the Center for Experiential Learning and Assessment (CELA) at the Vanderbilt University Medical Center [7]. Seven subjects with medical training performed 24 different procedures (Table 1) with varying amounts of repetition. Subjects performed the procedures on realistic medical mannequins that are commonly used for medical training. To collect video data four cameras were placed at different locations (Figure 2). They were positioned to capture as much of the procedure as possible and to ensure that the important parts of the procedure were always captured. Camera 2 (C2) was placed to emulate a ceiling mounted camera in an ambulance. Each camera collected video with a resolution of 3840 $\times$ 2160. This paper analyzed only C2 data as it was sufficient to ensure the mannequin, subject, and action were always visible.

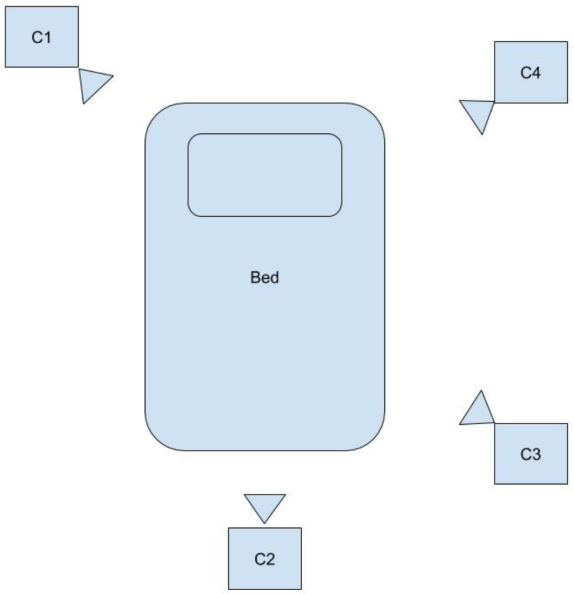

Fig. 2. Positioning of the four cameras used during data collection, as described in Paris et al. [7].

The collected video data were split into individual frames and each frame was assigned a category. Frames during which no procedure occurred were discarded. Each frame was cropped to reduce data size and eliminate extraneous information. Figure 3 shows the lines along which cropping occurred. The resulting frames were then resized to 256×256 pixels. In addition to the procedure name, each frame was labeled with the subject number and procedure occurrence as each procedure occurred multiple times.

Administer Medication	Bagging	Blood-Pressure Cuff
Chest-tube Prep	Chest-tube	Combat Tourniquet
CPR (Compression)	CPR (Breath)	Swab Area with Alcohol
Intubation	IO Line	IV Line
King Airway	Oral Airway	Pulse-OX
Draw Medication	ECG Leads	Vital Checking
Combat Gauze	Suturing	IM Administration
IV Tourniquet	Splinting	Wrap Head Wounds

TABLE I

TABLE 1: CATEGORIES OF CLINICAL PROCEDURES. ABBREVIATIONS: IM – INTRAMUSCULAR; ECG – ELECTROCARDIOGRAM; IV – INTRAVENOUS THERAPY; IO – INTRAOSSEOUS; OX – OXYGEN.

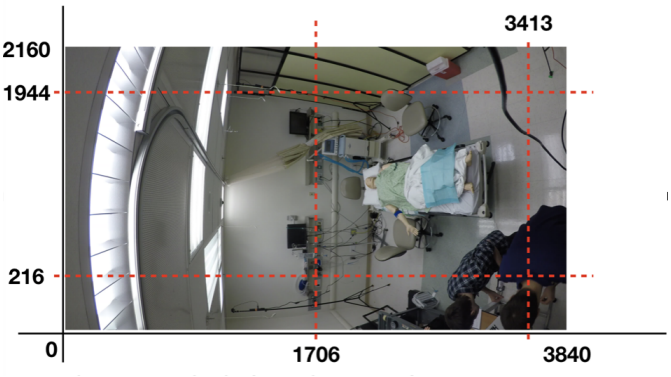

Fig. 3. Image cropping plan for frame data — only the central region is kept.

Of the data set composed of seven subjects, data from five subjects were used for training, one for validation, and one for training. We performed 5-fold cross validation by rotating the subject used for validation and for testing. We chose to fold on subjects as we want to ensure the model generalizes to new and different medics performing the procedures.

Subjects in this experiment completed each procedure a randomized number of times and the time to complete each procedure varied leading to imbalanced classes. Class sizes ranged from 3,000 frames to 60,000 frames. To ensure the model does not overfit to one category and every class is equally represented the data was either downsampled or upsampled to approximately 8,000 frames. To downsample, approximately 8,000 random frames were selected and others discarded. To upsample, we duplicated each frame; then, if the upsampled category contained more than 8,000, it was downsampled as in the other categories. For example, in the fold 2 training set, the category *Pulse-OX* contained 3544 frames, and all frames were duplicated to reach 7088 frames (approximately 8000). Category *Chest-tube Prep* contained 6480 frames. We duplicated these frames and selected 8000 random frames, discarding the rest.

Validation and testing sets were balanced similarly with only the target number of frames changing. Our target was chosen so that only 15% of the classes would require duplication. The other 85% would then require downsampling. For each fold a different value was chosen based on the size of the categories in that validation or testing fold.

The previous balancing plan applies to two of the CNN models we test in this paper, the so-called "main model" (Section IV-A) and "variant 1" (Section IV-B). A third model, "variant 2" (Section IV-C), makes use of temporal relationships and has additional data inputs. For this model, we reduce our data size by resampling each session at 6 fps. In other words we choose frames 0, 5, 10, etc. To combat class balance issues we need to include more incidences of categories with lower representation. To do this we divide our categories into large, medium, and small. Large categories only resample once, medium categories resample an additional two times, and small categories an additional four times. These additional resamplings are offset from 1 to 4 frames from the first frame which gives us slight differences in our samples.

Our analysis system is built on Keras, Python's deep learning library.[1] Keras' built-in image augmentation framework helped increase the variability in the data. The brightness, rotation, zoom were all randomly modified. Images were randomly shifted vertically or horizontally, and could randomly be flipped vertically or horizontally. For models using temporal relationships, the same augmentations were applied to each related frame.

IV. Methods

The models in this section are largely based on InceptionV3 due to its success in image recognition tasks [10]. All training is done starting with the pretrained imageNet weights for low-level feature detection. For each fold the model was trained until a baseline validation accuracy was reached and the testing accuracy measured. This occurred three times per fold with the highest testing accuracy being recorded.

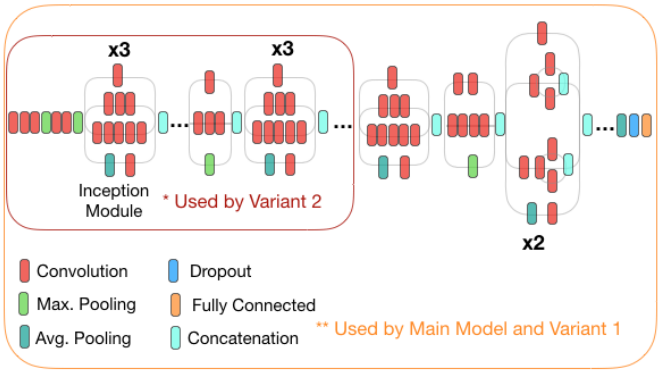

Fig. 4. A Compressed View of InceptionV3 [10]. The portion used by the main model and variant 2 are depicted with the orange and red lines, respectively.

A. Main Model: Full Inception Model with Single Frame

Karpathy et al. [8] demonstrated that a single frame was sufficient to achieve a high accuracy (40%) in large datasets such as the UCF-101 and the Sport1M datasets.

[1]https://keras.io/

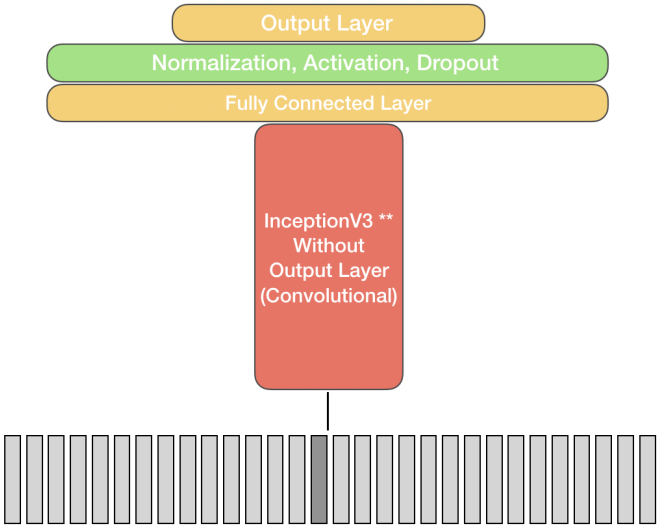

Fig. 5. An Illustration of main model, a direct adaptation of InceptionV3. A red box represents the part of model taken from InceptionV3 (portion of the model marked with ** in Figure 4); a yellow box represents dense layers; a green box represents utility layers.

Those datasets are more complex than the clinical procedure dataset. The InveptionV3 architecture is a popular choice for deep learning image recognition tasks and we choose to use it as the basis of our network. Given the similarity of domains (action recognition), we expect similarly strong results. The clinical procedure dataset is smaller than similar datasets so the pretrained ImageNet weights were used to reduce training time and increase performance. Figure 5 is an illustration of this architecture, which we call our main model. Our model is made of the InveptionV3 architecture without the final output layer. Instead we place a fully connected layer, normalization, activation, dropout, and a second fully connected layer to adapt the model to our problem.

B. Variant 1: Full Inception Model with Combined Categories

In the application domain for which we are training our recognition models, many of the procedures, while different actions, have temporal or physical associations with each other. For example, *Chest-tube Prep* must occur before *Chest-tube* and *CPR (Breath)* alternates with *CPR (Compression)*. In this variant we explore combining the *CPR (Breath)* and *CPR (Compression)* categories, as it is possible that we might get improved performance by grouping these associations manually rather than letting the algorithm determine them automatically. In particular, the main model commonly confused these two procedures and the goal is to remove that confusion and look for features common to both procedures.

C. Variant 2: Partial Inception Model with Late Fusion

Popular datasets, such as UCF101, Sport1M, or Youtube8M, cover a wide variety of topics, fields, and activities. Most of classes can be distinguished with one single frame. One problem we face is that some of the

categories in the clinical procedure dataset, due to the similarity in context and equipment, cannot be easily distinguished from each other with a still image.

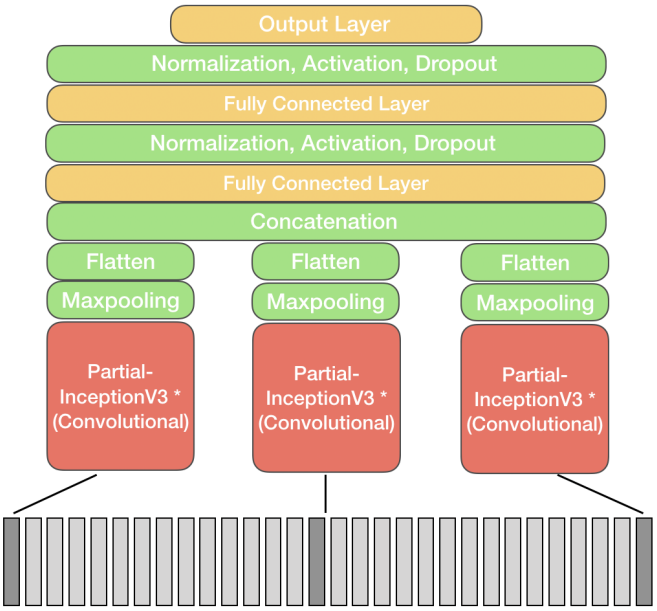

Fig. 6. The variant 2 model, a CNN model using the concept of Late Fusion. Red boxes represent the part of model taken from InceptionV3 (portion marked with * in Figure 4); yellow boxes represent dense layers; green boxes represent utility layers; light and dark grey boxes represent available and selected frame data.

To overcome this difficulty, we took inspiration from Late Fusion, and sample the video three times across a one second span: each data sample consists of the first frame of the second, the middle frame of the second, and the last frame of the second (1st frame, 15th frame, and 30th frame). Figure 6 is an illustration of our variant 2 (late fusion) model that takes advantage of both frame information and temporal information between frames. We first use convolutional layers and Inception Modules [10], which concatenate the results of different sizes of convolutional filters, to summarize information from each frame sample. Frame information is then fused to temporal relations in the top fully connected layers. Lastly, the output layer classifies clinical procedure based on given frame information and temporal information. The standard InceptionV3 model is further pared down to use only six inception modules as a feature extractor. Three equivalent copies with shared weights of those six modules were created. These three were each fed into a separate maxpooling and flatten layer before being concatenated and fed through a series of layers as shown in Figure 6.

V. Results

The results of the 5-fold cross validation can be found in Table 2 and the categorical accuracy of models across 5-fold cross validation can be found in Table 3. Table 3 is computed by calculating the accuracy for each category for each fold and then averaging the categorical data across 5 folds, so that each experiment is given same weight in the result.

Fold	Main Model	Variant 1	Variant 2
Fold 1	53.593%	57.798%	47.402%
Fold 2	42.398%	45.842%	50.506%
Fold 3	46.813%	44.975%	51.053%
Fold 4	44.525%	48.197%	61.770%
Fold 5	52.673%	53.504%	55.114%
Avg. Acc	48.000%	50.063%	53.169%

TABLE II

TABLE 2: THE TESTING ACCURACY OF THE THREE MODELS ACROSS VARIOUS FOLDS AND THE RESULTING ACCURACY OVER ALL FOLDS.

	Main Model	Variant 1	Variant 2
Administer Medication	36.743%	31.083%	12.222%
Bagging	66.476%	62.873%	90.371%
Blood-Pressure Cuff	39.218%	44.975%	24.572%
Chest-Tube	58.013%	59.649%	72.113%
Chest-Tube Prep	0%	0.422%	0%
Combat Gauze	45.668%	35.321%	35.576%
Combat Tourniquet	98.661%	94.474%	94.177%
CPR (Breath)	55.593%	72.120%	31.827%
CPR (Compression)	43.424%	N/A	32.649%
Draw Medication	7.099%	7.272%	4.382%
ECG Leads	56.050%	84.432%	69.910%
IM Administration	25.591%	18.931%	8.593%
Intubation	32.750%	51.789%	13.106%
IO Line	63.121%	67.852%	68.993%
IV Line	60.799%	58.810%	80.794%
IV Tourniquet	39.007%	35.527%	23.977%
King Airway	31.208%	47.505%	13.705%
Oral Airway	33.761%	41.290%	56.685%
Pulse-OX	35.269%	16.118%	10.000%
Splinting	79.053%	95.100%	61.038%
Suturing	54.862%	49.358%	40.258%
Swab Area W/ Alcohol	6.189%	19.863%	2.920%
Vital Checking	46.600%	24.816%	37.195%
Wrap Head Wound	92.089%	93.817%	81.202%
Total	47.474%	49.113%	41.731%

TABLE III

TABLE 3: THE TESTING ACCURACY BY CATEGORY FROM THE 5-FOLD CROSS VALIDATION.

Entry *CPR (Breath)* for variant 1 gives the accuracy for the combined CPR category. Table 2 is the traditional way of calculating accuracy for a cross-validation; however, Table 3 gives a better idea of the performance of the classifiers across each category, and since it balances each category equally, it gives a better overall indicator of performance.

The main model and variant 1 use single frame data for classification, while variant 2 uses 3 frames evenly spaced from 1 second of data for procedure detection. While they are using different types of data for classification task, they share the same goal of accurately detecting different clinical procedures. As a result, even though they use different types of testing datasets, their testing accuracies are generally comparable.

When comparing the overall results of Table 2 and Table 3, the averaged categorical accuracy data of main model and variant 1 are within 1% difference with their testing accuracy data. This is because number of testing data for each category is approximately the same, and the 5-fold cross validation

accuracy is a good estimator of the overall performance at making accurate classification for each model tested. However, there is a difference between variant 2's overall cross-validation performance and its averaged categorical accuracy data. This difference is caused by the low sample count of several categories. For example while the accuracy of category *Chest-tube Prep* and *Pulse-OX* is zero for Fold 4, category *Chest-tube Prep* only contains 9 samples and *Pulse-OX* only contains 19 samples, while other categories normally contains around 120 samples. Thus, this discrepancy supports the conclusion that our data set is too sparse to support high classification accuracy for some categories, and indicates which categories have sparse data sets.

VI. Discussion

The averaged categorical accuracy data of all three models are higher than the averaged accuracy achieved in the previous work where perfect knowledge of the body was assumed [2]. It therefor suggests that it is viable to perform the clinical procedure detection task without paramedics wearing sensors on their arms, although a combined method may yield higher performance that could reach standards high enough for the medical domain. Our results strongly suggest that richer video data would be helpful, and indicate where such data could be productively collected.

	CPR-B	CPR-C	Other	Accuracy
CPR-B Truth	**7064**	1825	3617	56.485%
CPR-C Truth	4084	**5029**	3416	40.139%

TABLE IV

TABLE 4: CONFUSION MATRIX BETWEEN CPR (BREATH) AND CPR (COMPRESSION)

Table 4 is the confusion matrix between CPR (Breath) and CPR (Compression) for the main model. A confusion matrix summarizes the prediction of data according to their predicted labels. It is an accurate classification when the predicted label matches the true label (shown as bold figures in Table 4); otherwise, it is an inaccurate classification.

The table records all data from *CPR (Breath)* and *CPR (Compression)* categories (the first column) in the 5-fold and their predicted labels (the first row of the table). Predicted labels other than *CPR (Breath)* and *CPR (Compression)* are all recorded in *Other* category, and the accuracy for each category is calculated based on the given data. A close examination reveals that while each category achieves respectable categorical accuracy, a relatively large portion of the error is due to the model's inability to distinguish between *CPR (Breath)* and *CPR (Compression)*.

One cause of such inability is the close temporal proximity and the repetitiveness of the two categories. Compression and breath regularly happen one after another multiple times, and one major part of the CPR sequence is the transition time from one to the other. During data labeling, it is logical to randomly select a point in the transition and mark the division between two categories; however, during training, such

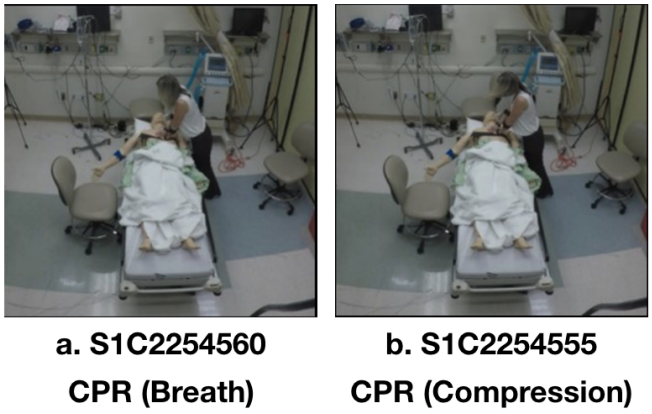

Fig. 7. Similarity between CPR (Breath) (left) and CPR (Compression) (right).

an indistinct boundary between the two categories will cause confusion for the model. Figure 7 is an example of such problem. While both Figure 7a and Figure 7b come from the same transition period and they share significant similarity, Figure 7a is labeled as *CPR (Breath)*, but Figure 7b is labeled as *CPR (Compression)*.

Such confusion is avoidable. The hospital may not benefit from knowing how many times compression is applied versus that of breath, so it makes sense to combine the two categories and report CPR as one category, and the similar logic also applies to *Chest-tube* and *Chest-tube Prep*, discussed next.

Category	Percentage
Chest-tube	59.678%
IM Administration	24.552%
Suturing	10.866%
Others	4.904%

TABLE V

TABLE 5: SORTED MISCLASSIFICATION CATEGORIES FOR CHEST-TUBE PREP.

Category *Chest-tube Prep* stands out among other categories, because it consistently has accuracy near zero for all models. Table 5 shows the sorted categories to which models wrongly classify data of *Chest-tube Prep*. The data suggests that for the majority of the time, models classify data from *Chest-tube Prep* as *Chest-tube* (59.678%). It suggests that *Chest-tube* and *Chest-tube Prep* also suffer from the ambiguity problem during category transitions, as these two categories' occurrences are highly correlated. In addition, other factors such as the imbalance of data across subjects also potentially contribute to the low accuracy of *Chest-tube Prep*. There are 10,262 frames of data available for *Chest-tube Prep*; however, two out of seven subjects do not contain any data, while two subjects contains more than 6,000 frames of data.

From the model size's stand point, due to the fact that variant 2 (the late fusion model) only uses a partial InceptionV3

model up to "mixed6" layer, and the parameters are shared across all branches, the number of the parameters used by the this model ($\approx$ 14M parameters) is significantly less than that used by the main model ($\approx$ 24M parameters). The reduced number of parameters, however, does not suggest that variant 2 requires less computation power than main model.

Both variant 1 and variant 2 explore ideas to increase the performance of CNN based video classification methods for clinical procedures. The results and confusion matrices generated by main model, variant 1, and variant 2 give us suggestions for more practical ways of categorizing data and directions to create more powerful models.

VII. Future Work

In this work, we show that video classification methods from deep learning provide a tool for clinical procedure detection, potentially enhancing the communication between EMS and receiving hospitals. Future work includes training with improved data sets suggested by this work, incorporating data from multiple sensors into this framework as in Heard et al. [2], and trying improved deep learning models. The ultimate goal is to have models that work on video gathered in the field that can automatically detect clinical procedures to medically useful standards in real time.

New deep learning models for video classification are continually emerging. As part of this work, we tried a CNN-LSTM model, similar to work by Ng et al. [6], but were not able to get satisfactory results. Also of note, Feichtenhofer et al. [9] have just made available their SlowFast network, which recently had high performance on a video benchmark. As deep learning models improve, we expect our own identification results to improve.

ACKNOWLEDGMENTS

The authors thank Kaiwen Xu for helpful suggestions.

References

[1] M. K. James, L. A. Clarke, R. M. Simpson, A. J. Noto, J. R. Sclair, G. K. Doughlin, and S.-W. Lee, "Accuracy of pre-hospital trauma notification calls," in *The American Journal of Emergency Medicine*, 2018.

[2] J. Heard, R. A. Paris, D. Scully, C. McNaughton, J. M. Ehrenfeld, J. Coco, D. Fabbri, B. Bodenheimer, "Automatic Clinical Procedure Detection for Emergency Services," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, 2019, pp. 337-340. doi: 10.1109/EMBC.2019.8856281

[3] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," in *IEEE Communications on Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[4] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal. 2015. "Recurrent neural networks for emotion recognition in video," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. 467-474. ACM

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. 2015. "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2625-2634

[6] Y.H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. "Beyond Short Snippets: Deep Networks for Video Classification," in *Proceedings of the Computer Vision & Pattern Recognition, Boston*, MA, USA, 7–12 June 2015.

[7] R. A. Paris, P. Sullivan , J. Heard, D. Scully, C. McNaughton, J. M. Ehrenfeld, J. A. Adams, J. Coco, D. Fabbri, and B. Bodenheimer (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.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li. (2014). "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* pp. 1725-1732.

[9] C. Feichtenhofer , H. Fan, J. Malik, and K. He (2019). "Slowfast networks for video recognition," in *Proceedings of the IEEE International Conference on Computer Vision* pp. 6202-6211

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). "Rethinking the inception architecture for computer vision" in *Proceedings of the IEEE conference on computer vision and pattern recognition* pp. 2818-2826.