An agitation detection system based on surveillance videos for people suffering from dementia

Muhammad Idrees
Faculty of Electrical and Computer Engineering
Univeristy of Ontariotech
Email: muhammad.idrees@ontariotechu.net

Hamed Akhoundi
Faculty of Electrical and Computer Engineering
Univeristy of Ontariotech
Email: hamed.akhoundi@ontariotechu.net

Mohammad Javad Rajaei
Faculty of Electrical and Computer Engineering
Univeristy of Ontariotech
Email: mohammadjavad.rajaei@ontariotechu.net

***Abstract*—Agitation is a common phenomenon in people suffering from dementia which is characterized by harmful behavior such as kicking, slapping, punching, hitting, and choking etc. To mitigate these harmful behaviors, an agitation detection system needs to be established. This paper, therefore, focuses on the development of one such system. At first, key features are engineered from the raw skeleton key points. The skeleton key points are generated by passing the videos through OpenPose and DeepSORT. OpenPose and DeepSORT work in parallel to generate skeleton key points while also keeping track of individual ID`s. These features include Euclidean norms, speeds, and accelerations of the link lengths. PCA analysis reveals non-linear separational boundaries in the feature space and therefore, non-linear supervised models such as Decision Trees, KNN, Random Forest classifier and DNN are considered for training and testing. Further, a semi-supervised spatiotemporal autoencoder is also trained and evaluated on raw videos rather than the engineered features. KNN achieves an accuracy of 98.58 % on test set among the supervised models, while the autoencoder achieves 93.7%. Additionally, a web-based dashboard is also developed which has the functionality to inform the concerned authority about the location and time of the agitated event if detected.**

Keywords—Dementia, skeleton, key points, PCA, supervised, sparse, autoencoder, agitation, violence, spatiotemporal

I. INTRODUCTION

Over the years, surveillance cameras and other surveillance equipment have been installed at different places to monitor public behavior and ensure safety. Public places include hospitals, subways, schools, banks, stores, highways [1]. In 2019, IHS Markit predicted that there would be 1 billion surveillance cameras worldwide by the end of 2021. Observation includes analyzing behaviors of people to find any questionable and unusual events. Since videos are recoded 24/7, detecting suspicious activity is a very difficult and time-consuming task. Moreover, finding such activities in real-time without the help of computer vision within the enormous volume of data are extremely challenging [2, 3]. Therefore, different approaches have been developed to recognize human activities in real time [4, 5, 6, 7, 8]. Surveillance videos can be analyzed using these methods to detect suspicious activity and enhance the security systems in smart locations.

Over the years, to develop a strong framework to conceive semantically significant scene behaviors, statistical-based methods have gradually replaced rule-based ones [9]. Rule-based systems utilize predetermined rules to categories activities as normal or abnormal [10, 11]. They work well but can only identify those kinds of specified anomalies. Additionally, they have very little scalability and reliability, especially for events that are not visible in the scene.

Most of the early research focuses on the use of supervised techniques to classify abnormal/violent behavior in videos. Supervised techniques require training video clips that have been properly labelled [12]. These techniques mainly concentrate on types of abnormal events. Since supervised approaches cannot identify previously undetected anomalous events, they are insufficient for generalization. It is necessary to have many abnormal events but collecting them and labeling them is complex and cost intensive. Moreover, it is challenging to find a training data set that includes every potential anomalous behavior that might appear in reality. Therefore, more focus has been put on unsupervised and semi-supervised approaches to address these problems.

Since normal video clips are all that are needed to train models in semi-supervised approaches instead of supervised ones, they are more widely applicable than supervised ones. Early works, in terms of feature extraction, typically used a variety of high-level features to represent the normal behavior [13, 14]. These features are easily misunderstood when confronted with intricate or crowded scenes that have many shadows and blurs. Xu et al. [15], proposed to learn motion/appearance feature representations using stacked denoising autoencoders. The networks used in their work are

relatively shallow, since training deep autoencoders on small abnormality datasets is prone to overfitting. Moreover, their networks are not end-to-end trained, and the learned representation needs externally trained classifiers (multiple one-class SVMs) which are not optimized for the learned features. They achieved an average accuracy of 92.01% based on their collected database. Similarly, a spatiotemporal autoencoder [16] is trained on normal videos and the reconstruction loss is used to separate normal videos from abnormal ones.

Sugan et al. [17] used cestrum features extracted from equivalent rectangular band width (ERB) triangular filter banks for speech emotion recognition. Two new triangular filter banks were proposed and used together with the traditional filter banks to extract four different cestrum features. The experimental results show that the maximum recognition accuracies of this method based on speaker-dependent (SD), and speaker-independent (SI) scenes were 77.08% and 55.83%, respectively.

Since the number of people facing dementia is rapidly increasing, our project is focused on detecting events of agitated behaviors in surveillance videos of hospitals. The most common symptoms of dementia are repetitive and agitated behaviors. More specifically, in this project, an end-to-end machine-learning pipeline on real-time camera data is trained and tested to detect symptoms of an agitated behavior. First, OpenPose [18] and DeepSORT [19] are used in parallel to extract the key points of each person's skeleton, while also keeping track of the person`s ID. Further, the raw key points are converted into rich features and are analyzed using PCA. PCA reveals the first 33 principal components of the feature space are sufficient to model the classification task. Moreover, a sparse autoencoder is also trained on normal videos that responds differently to a video with agitated events going on. In section 2, we describe the methods used to build the pipeline in both, supervised and semi-supervised settings. In section 3, the results are presented while section 4 discusses the performance, applicability, and the shortcomings of each approach. Additionally, a web-based interface for the real time execution of the model is also briefly discussed. The main contributions of this article are engineering features from the raw skeleton key points and the investigation of a spatiotemporal autoencoder to detect agitation in a raw grayscale video.

II. METHODS

A. Data Preparation

A model`s performance depends on the type of dataset used for training and validation. Since the study aims to detect agitation/violence in a video, two distinct classes are identified: violence and non-violence. More specifically, 1000 violence videos and 1000 non-violence videos are cropped to around 3 to 4 sec time intervals by carefully selecting the frames which have unambiguous violence and non-violence events respectively. After, each video is passed through the pipeline shown in figure 1 to generate the raw skeleton key points and consequently, store them in a csv file. The pipeline has two models running in parallel: OpenPose and DeepSORT. OpenPose computes 25 key points for each individual in a frame, while DeepSORT generates a bound box with ID tags. The key points and ID tags are correlated to the exact person by computing the approximate centroid of the key points from the OpenPose and comparing it the bounding box coordinates from DeepSORT. In this way, the ID of an individual is correctly tracked since the tags can change from frame to frame due to the limitation of DeepSORT. These raw key points are henceforth stored in a csv file with appropriate ID tags, frame numbers, frame width and height for scaling, and the corresponding class for each type of behavior (i.e., violence: 1, non-violence: 0).

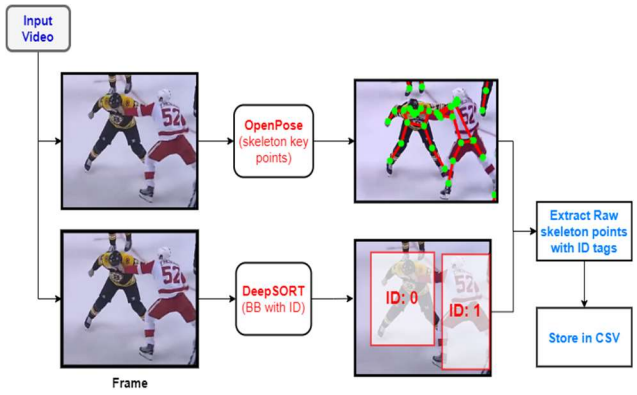

Fig. 1. *Pipeline for generating skeleton key points with appropriate ID tags*

B. Feature Engineering

Feature selection and engineering is an essential step in machine learning model development. After the raw skeleton key points are generated from the pipeline in figure 1, they are normalized on a scale of [0 1]. A key point in the center of frame is different than a key point in the corner of the frame, although it might belong to the same person who has just moved across the field of view of the camera. Using this intuition, it is not wise to use the raw key points as input features to the classifier. Thus, to generate more rich and meaningful features for the agitated activity, the raw key points are converted to distances (link lengths) by computing the L_2 norms. Multiple norms are computed to give the classifier more room to establish distinctive boundaries between the two classes of interest.

Moreover, since the agitated behavior is almost always associated with increased activity of the human joints, the speeds and accelerations of these links are also computed. This is done so by calculating the first and second order differences of the norms across two consecutive frames. The speeds and accelerations take into account the magnitude and intensity of changes in the link lengths which is a directed consequence of an agitated behavior. Nevertheless, to account for the orientation of the human body, angles between the links are also computed followed by the angular speeds and angular accelerations. In this way, a total of 58 features (table 1) are constructed from the raw key points data. The process of feature engineering is described in figure 2, while the OpenPose standard skeleton model is shown in figure 3

TABLE I. FEATURE DESCRIPTION

Feature	Description
d_{21}	Norm distance between key point 2 and 1
d_{31}	Norm distance between key point 3 and 1
d_{41}	Norm distance between key point 4 and 1
d_{51}	Norm distance between key point 5 and 1
d_{61}	Norm distance between key point 6 and 1
d_{71}	Norm distance between key point 7 and 1
d_{91}	Norm distance between key point 9 and 1
d_{101}	Norm distance between key point 10 and 1
d_{111}	Norm distance between key point 11 and 1
d_{121}	Norm distance between key point 12 and 1
d_{131}	Norm distance between key point 13 and 1
d_{141}	Norm distance between key point 14 and 1
s_{21}	Norm speed between key point 2 and 1
s_{31}	Norm speed between key point 3 and 1
s_{41}	Norm speed between key point 4 and 1
s_{51}	Norm speed between key point 5 and 1
s_{61}	Norm speed between key point 6 and 1
s_{71}	Norm speed between key point 7 and 1
s_{91}	Norm speed between key point 9 and 1
s_{101}	Norm speed between key point 10 and 1
s_{111}	Norm speed between key point 11 and 1
s_{121}	Norm speed between key point 12 and 1
s_{131}	Norm speed between key point 13 and 1
s_{141}	Norm speed between key point 14 and 1
a_{21}	Norm accel between key point 2 and 1
a_{31}	Norm accel between key point 3 and 1
a_{41}	Norm accl between key point 4 and 1
a_{51}	Norm accl between key point 5 and 1
a_{61}	Norm accl between key point 6 and 1
a_{71}	Norm accl between key point 7 and 1
a_{91}	Norm accl between key point 9 and 1
a_{101}	Norm accl between key point 10 and 1
a_{111}	Norm accl between key point 11 and 1
a_{121}	Norm accl between key point 12 and 1
a_{131}	Norm accl between key point 13 and 1
a_{141}	Norm accl between key point 14 and 1
θ_{81}	Angle of norm $d_{81}$with vertical axes
ω_{81}	First order difference of θ_{81} across two frames
d_{36}	Distance between key point 3 and 6
d_{47}	Distance between key point 4 and 7
d_{1013}	Distance between key point 10 and 13
d_{1114}	Distance between key point 11 and 14
θ_{31}	Angle of norm $d_{31}$with vertical axes
ω_{31}	First order difference of θ_{31} across two frames
θ_{61}	Angle of norm $d_{61}$with vertical axes
ω_{61}	First order difference of θ_{61} across two frames
θ_{41}	Angle of norm $d_{41}$with vertical axes
ω_{41}	First order difference of θ_{41} across two frames
θ_{71}	Angle of norm $d_{71}$with vertical axes
ω_{71}	First order difference of θ_{71} across two frames
θ_{101}	Angle of norm $d_{101}$with vertical axes
ω_{101}	First order difference of θ_{101} across two frames
θ_{131}	Angle of norm $d_{131}$with vertical axes
ω_{131}	First order difference of θ_{131} across two frames
θ_{111}	Angle of norm $d_{111}$with vertical axes
ω_{111}	First order difference of θ_{111} across two frames
θ_{141}	Angle of norm $d_{141}$with vertical axes
ω_{141}	First order difference of θ_{141} across two frames
$class\ (y)$	0: non-violence, 1: violence

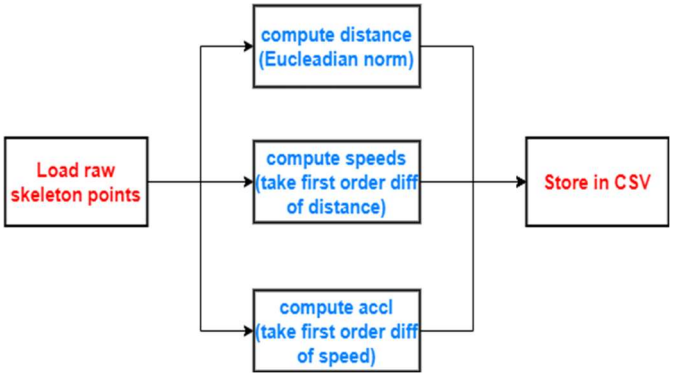

Fig. 2. *Key points conversion to engineered features*

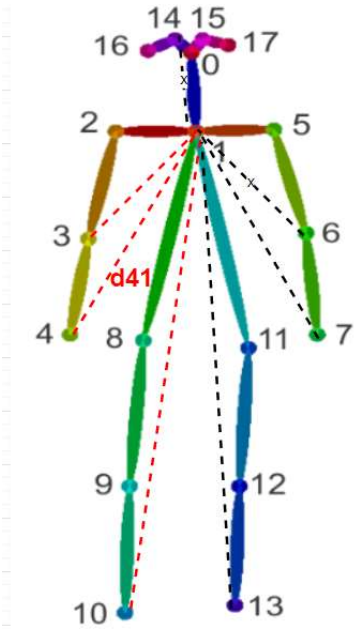

Fig. 3. *OpenPose standard skeleton model*

C. *Principal Component Analysis of features*

Principal component analysis (PCA) is applied to the engineered features to align the data along the principal directions of variance and investigate the redundancy in the feature space. The dataset obtained is of size (159875, 58). Where, the first dimension represents the number of data samples, and the second dimension represents the total number of features. First, the covariance matrix is computed to investigate the correlation between the features. This is shown in figure 4. It is observed that several features are highly correlated and thus can be removed to remove the redundancy and, later, speed up the model training and inference. The PCA is performed by applying singular value decomposition (SVD) to the covariance matrix. The variances along the principal directions (figure 5) indicate that the first 33 principal components have the highest significance and therefore, the rest of the components can be ignored with minimal loss of

information. If the first 33 principal components are considered, 99 % of variance in the data can still be retained. This is computed as:

$$Variance\ retained = \frac{\sum_{i=1}^{33} s_{ii}}{\sum_{i=1}^{58} s_{ii}} \cong 98.8\ \%$$

Further, visualization of the data along the principal directions (figure 6) indicates the two classes (i.e., violence and non-violence) can be separated by non-linear boundaries, although, the two classes do overlap occasionally. The less significant principal components do not contribute much to the boundary separation as can be seen in figure 6 where the samples are mostly merged with no feasible separation possible. The analysis thus concludes that a functional relationship between the input principal components and the output class can be learned if a non-linear model is used.

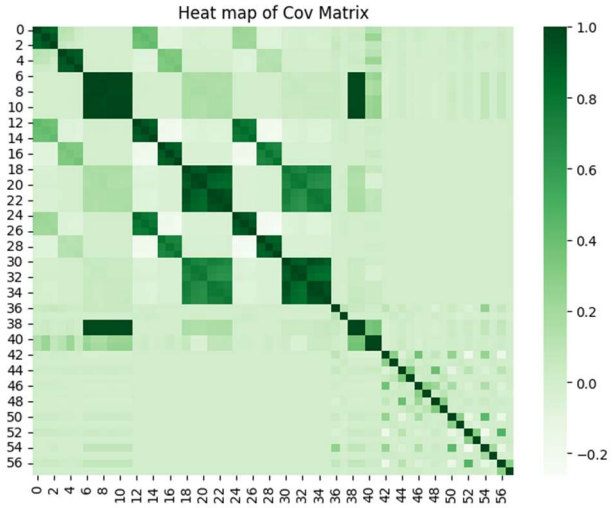

Fig. 4. *Heat map of covariance matrix computed from the raw 58 feature vectors*

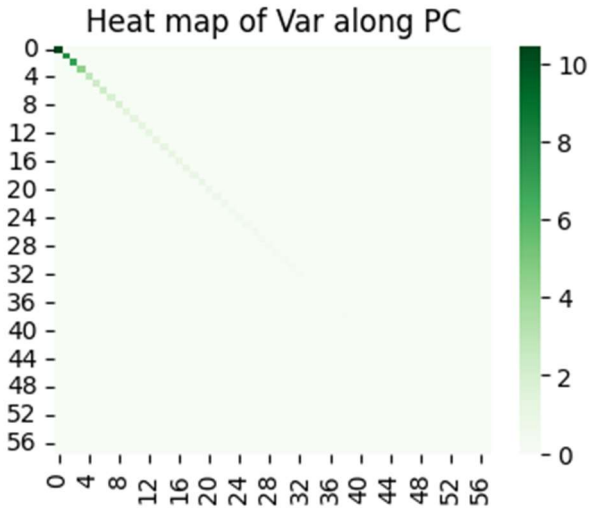

Fig. 5. *Heat map Variance along the Prinicipal components indicating the importance of the first 33 components*

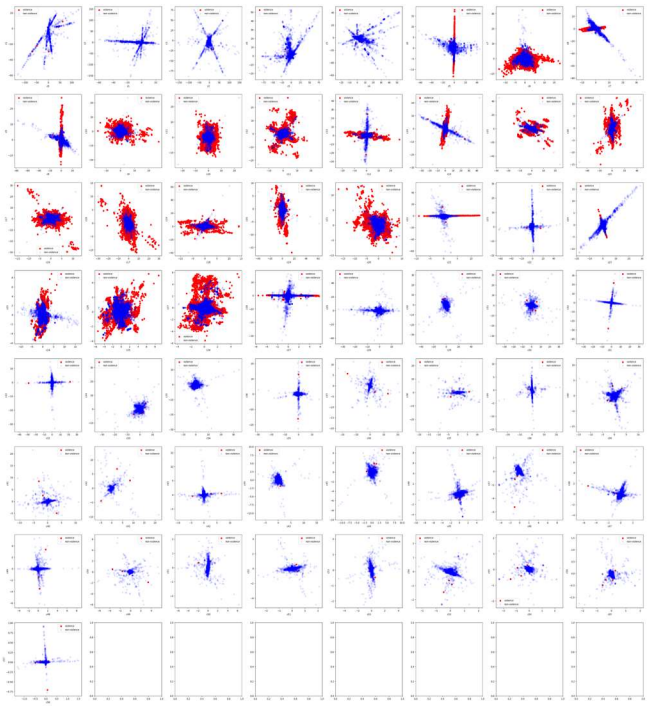

Fig. 6. *Visualization of prinicpal components (red: violence, blue: non-violence)*

D. Model selection (supervised)

PCA analysis reveals 33 prime components are enough to model the data while still retaining 99% integrity of the data. This is also validated by the covariance matrix, which indicates a certain extent of redundancy in the feature space and thus, some features can be dropped to speed up the model training and inference. Further, the visualization of principal features indicates there exist non-linear boundaries between the two classes and therefore, a non-linear model can be used to learn the functional relationship between the features and the class. Multiple non-linear models are considered. For Decision trees, the hyperparameter of interest is the criterion which can be $gini$, log_loss, and $entropy$. The other hyperparameters are kept on default settings. Similar is the case with random forest classifier. For the KNN (K-Nearest neighbor), only the number of neighbors is tuned while the other hyperparameters are set to default. Finally, for the DNN (Deep Neural Networks), three configurations are considered. The first configuration has two hidden layers with 20 neurons each, the second configuration has three hidden layers with 20, 20 and 10 neurons respectively, while the third configuration has a total of 4 hidden layers with 25, 25, 20, and 20 neurons respectively. The activation function for all the hidden layers is set to $tanh$, since it is better at separating non-linear boundaries, while the output activation is sigmoid. The penalty loss is set to $Hinge\ Loss$ instead of binary entropy to avoid overfitting. The batch size is kept equal to the training size to minimize the loss on the entire training set at once and speed up the training process. The summary of the model is shown in table 2.

TABLE II. SUMMARY OF SUPERVISED NON-LINEAR MODELS

Model	Hyperparameter configuration
Decision Trees	$criterion$: ['$gini$','log_loss','$entropy$']
Random Forests	$criterion$: ['$gini$','log_loss','$entropy$']
KNN	$n\_neighbors$: [3, 10, 20]
DNN	$Layers$: [33, 20, 20, 1] $Layers$: [33, 20, 20,10, 1] $Layers$: [33, 25, 25,20,20, 1]

E. Model selection (spatiotemporal autoencoder)

In addition to the supervised models, a semi supervised spatiotemporal autoencoder is also implemented to classify agitation. Autoencoders are easy to train since they do not require any labelled data and simply reconstruct the input. The key rationale behind the autoencoder is to feed it the raw input and classify the behavior into a normal or abnormal. A deep spatiotemporal autoencoder is trained on normal videos such that it learns the representation of normal videos. When an abnormal video is passed with sufficient agitation events going on, the encoded outputs diverge from normal and can be used to classify an agitated event in the video. The autoencoder, however cannot locate the location of agitation behavior since it processes the entire video at once. After a lot of experimentation, the model in figure 5 was implemented. The sparse autoencoder takes input volume of dimension (64,128,128,1). Where, 64 represents the number of frames, (128,128) the spatial window size, while the image is in gray scale. In the encoder part, the channels increase while the spatial and temporal windows decrease. At the bottleneck, the code layer is of dimension (4,8,8,256). Before the code layer, a dropout layer is inserted to impose the sparsity constraint and thus avoid learning the identity function. The dropout layer is set to a dropping rate of 10%. In the encoder side, Conv3D filters are used followed by RELU activation. The decoder side is the exact mirror image of the encoder. The autoencoder is trained on normal videos and once it learns the representation of normal videos, it is then used to encode the normal and agitated videos. The encoded values are then used as features to a feedforward network to train a classifier for agitation. During training the autoencoder uses only the reconstruction loss as given by equation 1. No sparsity constraint or image sharpening terms are used to penalize the loss function since they deprecate the performance of the encoder [16].

$$L_{mse}\,(I,O) = \frac{1}{N}\sum_{t=0}^{t=T} ||I_t - O_t||^2 \qquad (1)$$

Where I is the input volume, O is the output volume, $T = 64$ is the total number of frames in the input/output volume. N represents the total number of pixels in the input and output volume. N: 64 (128)(128) = 1,048,576

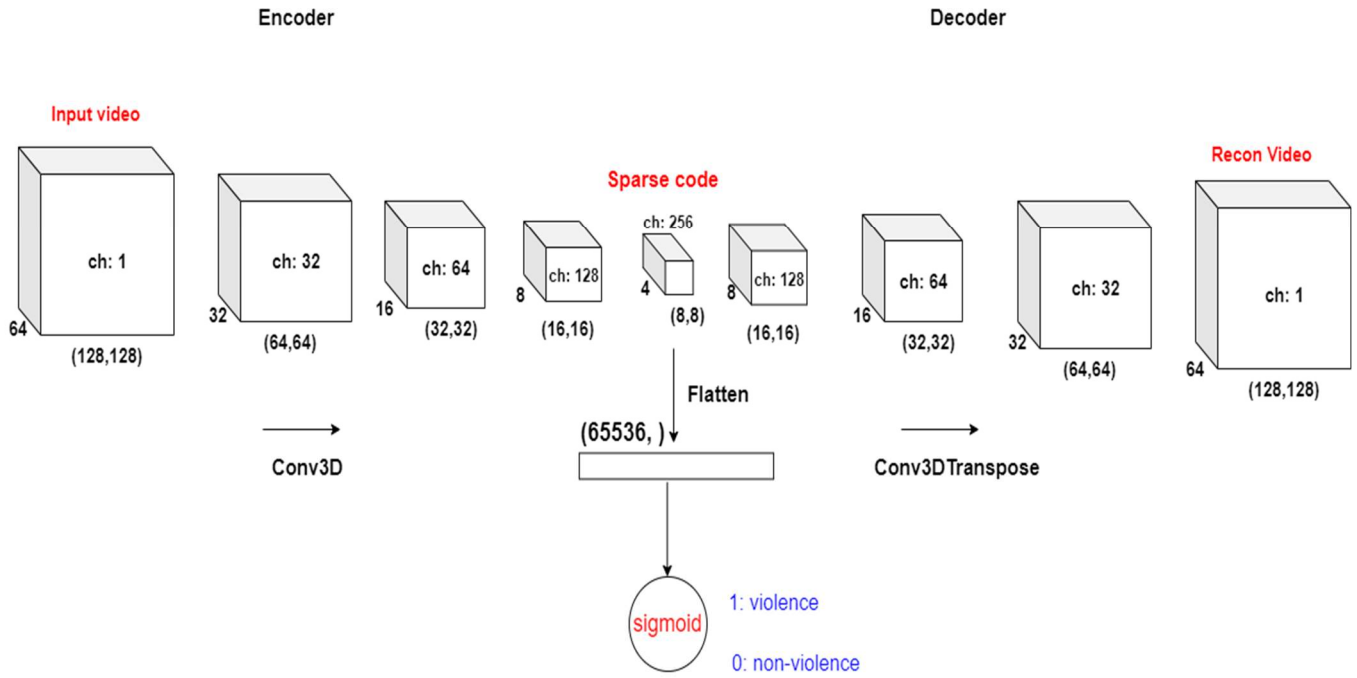

Fig. 7. *Sparse Autoenocder with a feed forward network for agitaiton detection*

III. EXPERIMENTS AND REULTS

Experiments were performed by training and testing multiple versions of the models identified in the model selection sections. The best configurations of these models are discussed below.

A. Results for supervised models

The dataset contained a total of 159875 samples, out of which approximately 44 % belonged to the non-violence class while the rest belonged to the violence class. Only the first 33 principal components were chosen as input features. This dataset was further divided into a train and test set of 70 % and 30% respectively. It was observed that the best accuracy on the test set is achieved when a KNN (K-Nearest Neighbors) classifier is used. However, the performance difference among the models is quite marginal and any model can be used during the inference stage. The best results are summarized in table 3, while the Loss curves for the DNN`s (Deep Neural Networks) are presented in figure 8-11.

TABLE III. RESULTS FOR SUPERVISED MODELS

Model	Best hyperparameters	Test Accuracy	Test Precision	Test Recall
Decision Trees	*criterion*: *entropy*	95.55%	94.99%	96.06%
Random Forests	*criterion*: *gini*	98.21%	97.39%	98.87%
KNN	$n_{neighbors} = 3$	**98.58%**	**98.30%**	**98.81%**
DNN	l: [33, 25, 25,20,20, 1]	97.42%	97.50%	97.84%

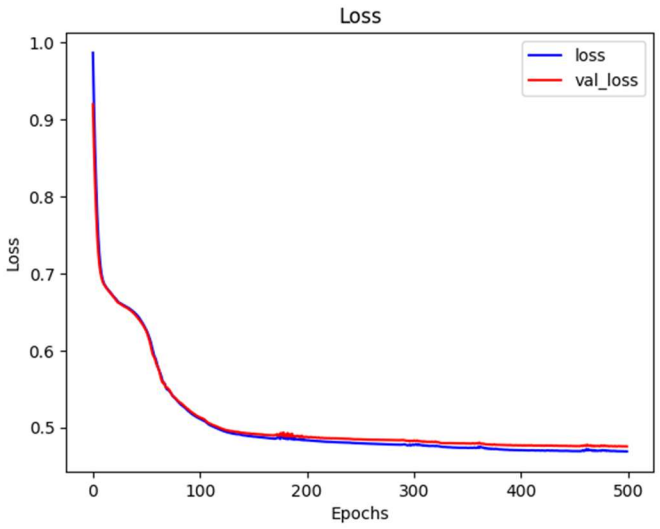

Fig. 8. *Loss curve for DNN*

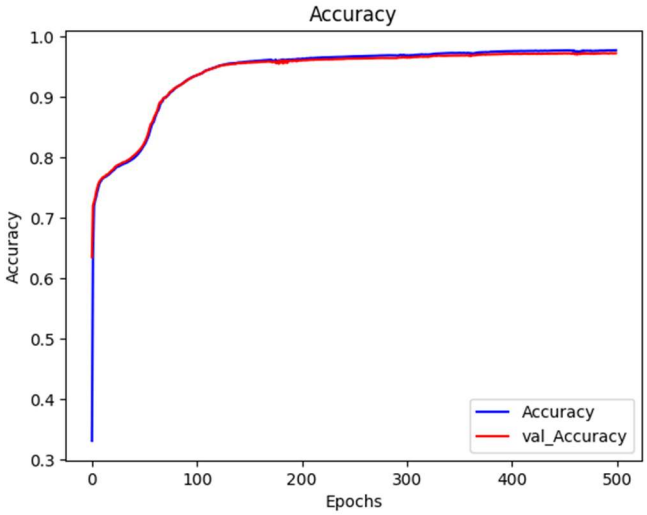

Fig. 9. *Accuracy curve for DNN*

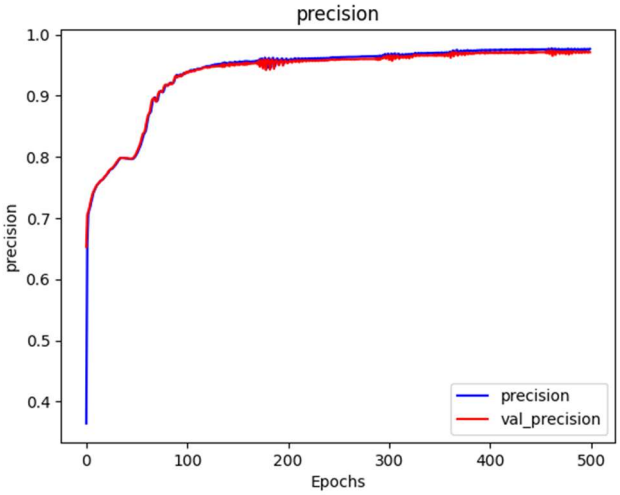

Fig. 10. *Precision curve for DNN*

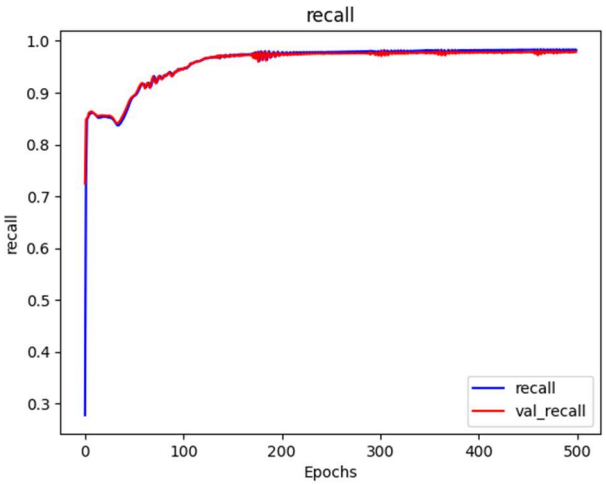

Fig. 11. *Recall curve for DNN*

B. *Results for spatiotemporal autoencoder*

The spatiotemporal autoencoder is trained on agitated videos to learn the key features of the violence activity across the spatial and temporal dimensions. The model is trained on 890 agitated videos which are cropped to 64 frames each. The training is stopped at around 300 epochs when the loss curve flattens out. The code layer is a (4,8,8,256) dimensional tensor which is flattened to a (65536,) dimensional feature vector. The encoder is then used to encode the agitated and non-agitated videos into a feature vector of size (65536,). A binary classifier is then trained on the encoded data which achieves an accuracy of 93.7 % on test set. The loss and accuracy curves are shown in figure 12 and 13 respectively. The encoded layer responds differently to a normal and agitated video as shown in figure 14 and 15 respectively.

Fig. 12. *Loss vs Epochs for the encoder classifier*

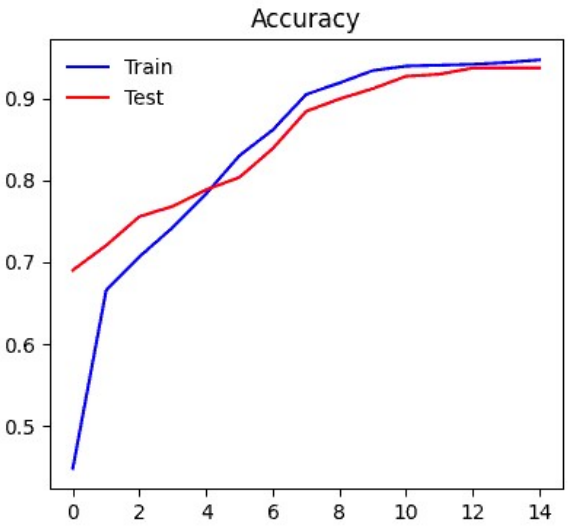

Fig. 13. *Accuracy vs Epochs for the encoder classifier*

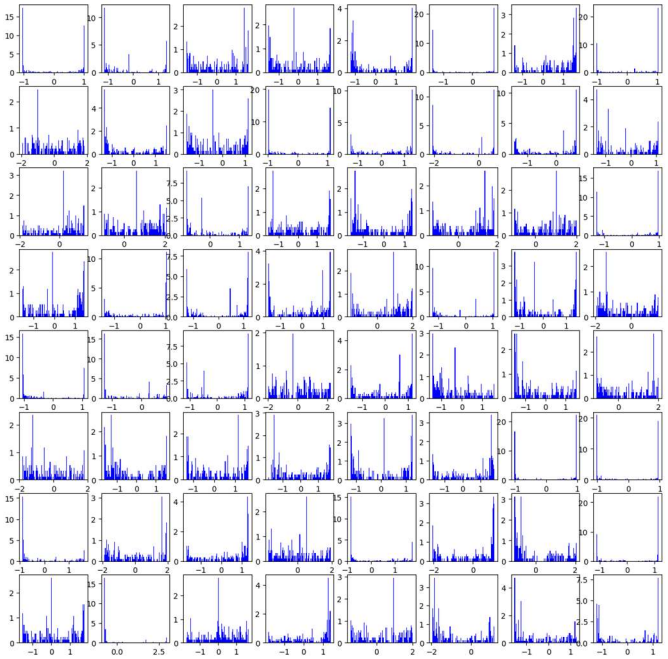

Fig. 14. *First 64 channels of the Encoded layer for a normal video*

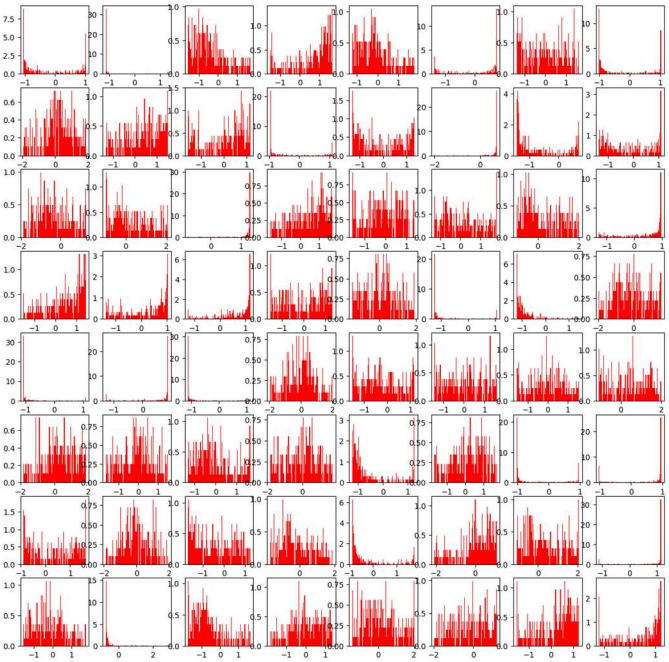

Fig. 15. *First 64 channels of the Encoded layer for an agitated video*

IV. DISCUSSION

A. Disscusion on supervised models

The results from supervised models reveal excellent performance on detecting the agitated behavior. As revealed by the principal component analysis, the separation boundaries among the components are non-linear and are captured adequately by all the supervised models. In our analysis, the KNN outperformed all the other models, however, the difference in performance is marginal and all the models are equally suitable for the inference stage. For the KNN, a performance decrease is observed when the number of neighbors is increased. The best accuracy, precision and recall are realized when 3 neighbors are used. For the decision trees and random forest classifier, the best criterion is $entropy$ and $gini$ respectively. However, the performance difference on the other criterion is marginal and any setting is deemed suitable. The DNN performs almost like the other non-linear models with the best results obtained when four hidden layers (25, 25, 20 and 20 neurons each) are used. The model however starts to overfit gradually at large epoch numbers as is seen by the leaky divergence of the validation loss.

During the inference stage, the raw key points are extracted from the surveillance videos using OpenPose and DeepSORT and then processed into the engineered features. These features are then transformed along the principal direction using the eigen matrix of the training data. The first 33 components are then passed through the KNN model to get a decision on each frame of the video. As such, the model output on 10 frames is obtained and the class which has the maximum frequency of occurrence in these 10 frames is chosen as the output of the model. In addition to the model development, a web based graphical user interface is also implemented which displays the result of the pipeline and can be connected to any camera source. The bounding box shade is changed to red color when an agitated event is detected by the pipeline while the back end also sends an SMS notification. A snapshot of the graphical user interface is shown in figure 16.

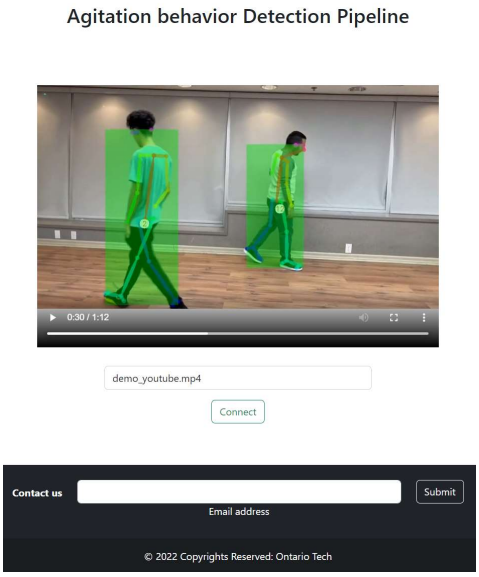

Fig. 16. *Snapshot of the web based graphical user interface*

B. Disscusion on spatiotemporal autoencoder

The autoencoder is trained by minimizing the reconstruction loss. After around 300 epochs, the loss curve flattens out and the training is stopped. Later, the encoder part of the network is used to encode the agitated and normal videos and is stored in an array. A classifier is then trained on the two classes which achieves an accuracy of 93.7 % on test set. The autoencoder is then tested on real time video stream and for the most part, it performs well at detecting agitation. The inference time of the autoencoder is very fast and it takes around 45 ms to process a 64-frame video and get a decision on the type of behavior. However, it is very sensitive to changes in pixel intensities and background changes. For future work, it is recommended to remove the background using pre-processing techniques and convert the frames to binary format. This will remedy the problem of changing pixel intensities. Moreover, optical flow fields can also be integrated into the pre-processing step to improve the performance of autoencoder. Additionally, the distribution of the code layer can be further studied by performing principal components analysis. Investigation of the separation boundaries between the two classes in the code space can lead to better selection of models for the classifier.

V. CONCLUSION

An agitation detection pipeline was developed and implemented on real time videos. Two models were investigated for the task. One model requires the detection of human pose as a skeleton key point, while the other uses a sparse autoencoder to generate key features of the type of behavior in a video frame. The raw skeleton features are converted into rich features by computing Euclidean norms, speed, and accelerations of the link lengths between two key points. PCA reveals non-linear separation boundaries in the feature subspace indicating the two classes (i.e., violence, non-violence) can be separated if a non-linear model is used. Multiple supervised models are trained and tested, and it is observed the performance difference among the models is marginal. Further, a sparse autoencoder is also trained to encode the feature of the video frames and classify them into agitated or normal videos. A classifier is then trained in a supervised fashion to distinguish agitated behavior in videos from a normal one.

The cost of generating skeleton key points is large since it requires computationally intensive models like OpenPose and DeepSORT. OpenPose runs with a frame rate of 0.3 fps and can hinder the performance in real time. Further, any error in the key point detection will deprecate the performance at the classifier stage. On the other hand, the sparse autoencoder runs very fast and takes around 45 ms to process a 64-frame video. It can detect anomalies in the entire video frames and therefore, is well suited for complex and crowded scenes where OpenPose struggles to generate correct key points. The performance of the autoencoder can be improved further by using a binary image and subtraction of backgrounds. For future studies, the models investigated in this paper can be tested on more datasets and compared to previous state of the art.

ACKNOWLEDGMENT

This project is aimed to develop an agitation detection system for a dementia unit in the hospitals.

REFERENCES

[1] W. Sultani, C. Chen and M. Shah, "Real-world anomaly detection in surveillance videos., 18–23 June 2018; pp. 6479–6488.," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, Salty Lake City, Utah, June 2018.

[2] I. S. Gracia, O. D. Suarez, G. B. Garcia and T.-K. Kim, "Fast fight detection," *PLoS ONE,* vol. 10, no. 4, April 2015.

[3] O. Deniz, I. Serrano, G. Bueno and T.-K. Kim, "Fast violence detection in video," *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP),* p. 478–485, Jan. 2014.

[4] K. Muhammad, S. Khan, M. Elhoseny, S. H. Ahmed and S. W. Baik, "Efficient Fire Detection for Uncertain Surveillance Environment," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 5, pp. 3113 - 3122, May 2019.

[5] A. Ullah, K. Muhammad, I. U. Haq and S. W. Baik, "Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments," *Future Generation Computer Systems,* vol. 96, pp. 386-397, July 2019.

[6] G. Batchuluun, Y. G. Kim, J. H. Kim, H. G. Hong and K. R. Park, "Robust Behavior Recognition in Intelligent Surveillance Environments," *Sensors,* vol. 16, no. 7, 2016.

[7] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang and S. W. Baik, "Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 7, pp. 1419 - 1434, July 2019.

[8] A. Ullah, K. Muhammad, J. D. Ser and S. w. baik, "Activity Recognition Using Temporal Optical Flow Convolutional Features and Multilayer LSTM," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 12, pp. 9692 - 9702, Dec 2019.

[9] O. P. Popoola and K. Wang, "Video-Based Abnormal Human Behavior Recognition—A Review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 42, no. 6, pp. 865 - 878, Nov 2012.

[10] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz and M. Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic," *Computer Vision and Image Understanding,* vol. 113, no. 1, pp. 80-89, Jan 2009.

[11] Y. A. Ivanov and A. F. Bobick, "Recognition of multi-agent interaction in video surveillance," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 06 August 2002.

[12] J. Hu, E. Zhu, S. Wang, X. Liu, X. Guo and J. Yin, "An Efficient and Robust Unsupervised Anomaly Detection Method Using Ensemble Random Projection in Surveillance Videos," *Sensors,* vol. 19, no. 19, 2019.

[13] A. Basharat, A. Gritai and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 05 August 2008.

[14] Z. Fu, W. Hu and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *IEEE International Conference on Image Processing 2005*, Genova, Italy, 14 November 2005.

[15] D. Xu, Y. Yan, E. Ricii and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Computer Vision and Image Understanding,* vol. 156, pp. 117-127, March 2017.

[16] S. S. Khan, P. K. Mishra, N. Javed, B. Ye, K. Newman, A. Mihailidis and A. Laboni, "Unsupervised Deep Learning to Detect Agitation From Videos in People With Dementia," *IEEE Access,* pp. 10349 - 10358, Jan 2022.

[17] W. Sultani, C. Chen and M. Shah, "Real-world Anomaly Detection in Surveillance Videos," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 6479-6488, 2018.

[18] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 43, no. 1, pp. 172 - 186, 17 July 2019.

[19] N. Wojke, A. Bewley and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *Computer Vision and Pattern Recognition*, 2017.