Efficient Multi-Stream Fusion for Violence Detection: Integrating RGB, Optical Flow, and Joint Features on Low-Memory Devices

Robin Li
Stanford University
lirobin@stanford.edu

Nathan Tran
Stanford University
natetran@stanford.edu

Jermaine Zhao
Stanford University
zx2004@stanford.edu

Abstract

Acts of violence that go unnoticed are an ever-increasing problem despite possessing the technological means to address it. In this project, we propose utilizing computer vision to detect and classify any acts of violence that might appear on security camera footage in order to alert the police. Existing literature shows that action recognition from video inputs is usually done using a CNN or LSTM based model. Building off of these works, we want to try integrating the concept of optical flow into a multi-stream model, while simultaneously applying our own approach to optical flow. The dataset for this project was obtained from Kaggle and contains short clips depicting either violent or non-violent activities. For this project, we proposed two different models utilizing a three-stream action recognition network (TARN) and compared them against two baseline models. After rounds of training and hyper-parameter tuning, we found that the End-to-End model utilizing TARN (TSE) outperformed all the other models, achieving the highest accuracy (0.752), precision (0.89), and F1 score (0.77). Going forward, we would like to see the implementation of such a model onto a lightweight device (e.g. RaspberryPi). Although we tried to accomplish this during the project, we unfortunately did not have the time to successfully implement it. By being able to push our model onto something as small as a RaspberryPi, we would be able to integrate it locally within security cameras and create a streamlined processes to alerting authorities.

1. Introduction

Incidents of violence occurring in less-surveyed urban areas, alleys, and streets are a significant public safety issue. Oftentimes, such crimes go unnoticed and unreported due to a lack in active patrolling. The ability to automatically detect such incidents can provide valuable insights and timely responses, potentially preventing escalation and facilitating immediate intervention. This project proposes using computer vision technology to automatically detect and classify incidents of violence in video clips. By analyzing visual data, the system aims to identify aggressive and involuntary behaviors that are often precursors or direct indicators of such distressing events.

1.1. Problem Statement

Our objective is to develop a system that uses computer vision to detect and classify violence incidents in real-time by feeding our system videos from surveillance cameras. Then, our system will process the video input of continuous video streams from surveillance cameras through a multi-stream process consisting of joint-feature extraction through an LSTM, RGB-feature extraction through a 2D ConvNet, and optical flow using a 3D ConvNet. The output will be a binary classification of either violence and non-violent behaviors. If our system detects violence, it will alert local security for immediate intervention. In all the methods that we are testing in this project, the model takes in video clips and then extracts the joints of humans through a pre-processing step, and finally inputs the processed data through the model which will classify the action in the video.

2. Related Works

The topic of video-based human action recognition is a subset of computer vision that has become increasingly prominent. There have been many advancements in deep learning in the past few years that have significantly enhanced the ability to classify and predict human actions from video data, such as 3D CNNs and LSTMs.

Pham *et al.* [6] provided a review of over 200 papers and found that convolutional networks (CNNs) are currently the most capable of being able to classify the actions of a human from a video. This is due to the architecture's ability to capture spatial hierarchies in visual data. Tran *et al.* [2] discussed that CNNs, especially 3D CNNs, are good at incorporating temporal information, which is very important for understanding video sequences over standard 2D CNNs.

One challenge that Pham *et al.* [6] identified within training various models is the quality of the datasets. Datasets such as KTH and Weizman were created in a simulated environment, meaning that the videos utilize human actors in order to create the actions. Performance on these artificial datasets was significantly better compared to datasets that sourced videos that contained authentic interactions.

Beyond CNNs alone, RNNs utilizing LSTM were shown to be effective, since RNNs were shown to recognize data effectively, given their ability to handle time series data when paired with LSTMs. RNNs process sequences well, which makes them suitable for scenarios where understanding the context and progression of actions is necessary. However, according to Tran *et al.* [2], one main downside of RNNs is that oftentimes, the data needs to be preprocessed, where CNNs can handle raw data due to the convolution process. In addition, according to Bütepage *et al.* [7], LSTMs struggle with complex movements.

Mathew *et al.* [9] explored two deep learning-based approaches, specifically single-frame CNNs and ConvLSTM networks, for human activity recognition from videos. They found that the single-frame CNN model achieved a higher accuracy (99.8%) on the UCF50 dataset compared to the ConvLSTM model, which shows how effect CNNs are in capturing spatial features from individual frames, while ConvLSTMs combine the strengths of CNNs and LSTMs to handle spatiotemporal data effectively. Simonyan and Zisserman [12] explored ConvNets for video recognition by using both spatial and temporal information. They proposed a two-stream architecture, where one stream processes spatial data from still frames, and the other processes temporal data from optical flow between frames. They showed that this method significantly outperforms previous deep learning methods, showing that combining spatial and temporal networks through late fusion enhances the recognition accuracy significantly [14].

There are other approaches proposed by different papers. Tran *et al.* [2] propose a two-stream CNN approach: one identifying the actions from one single frame and the second identifying actions using multi-frame optical flow for motion information. This method is particularly advantageous when there is limited training data.

Surek *et al.* [13] focus on evaluating hybrid deep learning models that combine supervised and semi-supervised learning approaches for human activity recognition in RGB videos. The authors tested a label smoothing technique with a 3D ResNet-50 and evaluated the performance of a model based on a semi-supervised learning methodology. They showed the feasibility of applying these deep learning architectures in real-life scenarios, where the algorithm processing can follow the real rate of image capture.

In our project, we aim to detect violence in real-time, which means we need the system to be able to handle all kinds of human behaviors and interactions that might not have been in the training data. We also need a model that could work well for real-time processing in surveillance systems to detect assaults. Being able to process data and make decisions quickly is really important for this kind of application so the assaults can be intervened in time. Bütepage *et al.* [7] suggested models that use temporal encoding-decoding networks to predict the future 3D poses based on past data. Their models use a fully-connected network with a bottleneck structure so that they can predict future motion capture frames from a sequence of previous frames, creating a temporal encoder for human motion. The authors tested their framework on a big mocap dataset and evaluated the learned features on tasks like action classification and predicting future motion. They found that their method did better than existing techniques, especially at predicting skeletal motion and classifying actions using the learned representations. Two advantages from their model are that it generalizes well to unseen data and is computationally efficient once trained.

2.1. Optical Flow

Optical flow is traditionally a technique that is used to estimate the motions of objects, surfaces, and edges within a scene based on a sequence of images. Optical flow is usually represented as a velocity field, where each pixel in an image is has a velocity vector indicating its movement across frames.

It has seen uses in Computer Vision through tasks such as motion detecting and tracking, 3D reconstruction, and autonomous navigation. For the task of implementing optical flow, recent methods have utilized CNNs. Dosovitskiy *et al.* [1] have proposed optical flow networks that consist of nine sets of alternating convolution layers and ReLU non-linearity. The filter-sizes decreases as the network gets deeper, starting with 7x7 and going down to 3x3. One important detail is that there are no fully-connected layers so that the network can process different sized inputs.

Optical flow provides valuable motion information that complements spatial appearance features from individual frames. Top-performing action recognition methods often use optical flow as an additional input stream alongside RGB frames. Integrating optical flow with deep learning models like two-stream CNNs has significantly improved recognition accuracy on benchmark datasets. We will try to take a similar but different approach [11] [4].

3. Dataset

3.1. Data Description and Collection

The dataset we are using for our project is from Kaggle.com, and it is called the "Real Life Violence Situations Dataset," [3] obtained by Mohamed Elesaway, Mohamad

Hussein, and Mina Abd El Massih. This dataset is composed of 2,000 short videos sourced from Youtube: 1,000 short videos depicting street fights and 1,000 short videos that don't depict fighting. The labels for the dataset are quite weak, being only violence versus non-violence. There is no further classification about the specific actions that are occurring in the video.

For training and evaluation, we split our dataset into 1,600 videos (800 violent and 800 non-violent) for training, 200 videos (100 violent, 100 non-violent) for validation, and 200 videos (100 violent, 100 non-violent) for testing. Each subset of videos contains the same number of videos per class (violent versus non-violent) in order to ensure that the model learns equally from both classes.

3.2. Data Pre-processing

Given that our dataset contains a variety of videos with different lighting, camera angles, number of people, resolutions, and other confounding variables, we wanted to create a more uniform way in order to represent the data. In doing so, we decided to use OpenPose [10], a real-time multi-person 2D pose estimation tool. However, the source code for OpenPose was not compatible with newer versions of OpenCV, one of the required libraries. Instead, we opted to use Lightweight OpenPose, a more optimized version of the original OpenPose with negligible accuracy drop. By feeding each data point into OpenPose, we were able to extract the skeletons of the humans in the video in order to keep track of the movement of their joints, as seen in Figure 1. This would be beneficial for identifying if there was violence.

We sampled 240 frames per video by taking 8 seconds from each video at 30 fps. If the video had less frames than that, we repeated the last frame so that the input dimensions were uniform for the model. This makes sure that all the input sequences have a consistent length, simplifying model training and evaluation. We used data augmentation to resize the frames to a resolution of 224x224 pixels and normalized using the mean and standard deviation of the ImageNet dataset. We extracted RGB frames and optical flow frames using a pre-trained ResNet50 model, and we also extracted pose estimations.

4. Methods

We propose four methods to detect violent and non-violent actions in videos where the first two are our baseline and the last two are the models we are proposing. The models are C3D [2], PoseLSTM (joint feature only), Three-Stream Action Recognition Network with Separation of Feature Extraction and Training (Spatio-Temporal-Pose), and Three-stream Action Recognition Network with End-to-End Training (Spatio-Temporal-Pose). These four methods analyze and classify from different feature dimensions,

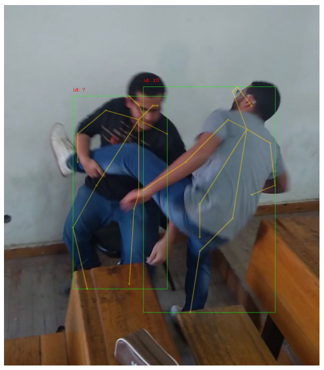


Figure 1. Example of using Lightweight OpenPose in order to extract the joint features of human subjects within a frame of a video. The number of people are also kept track of in the processing.

providing diverse solutions for detecting violent actions in videos.

4.1. Model 1: C3D

The C3D model captures spatiotemporal features in videos through 3D convolutions, making it suitable for video classification tasks. The detailed steps are as follows: First, video frames are processed using the C3D model to extract spatiotemporal features through 3D convolutions. Then, the extracted spatiotemporal features are classified through fully connected layers to output the final classification result (violent and non-violent actions).

4.2. Model 2: PoseLSTM (joint feature only)

We used Lightweight OpenPose, a CPU based human-pose estimation library, to extract joint features of individuals from videos and store them as JSON files. Each JSON file is organized by time (frame number) and person ID, along with their joint positions which are stored as a list of coordinate pairs. These JSON files are then fed into an LSTM to obtain binary classifications (Violence/Non-Violence) for training the model.

4.3. Model 3 (TSP): Three-stream Action Recognition Network with Separation of Feature Extraction and Training (Spatio-Temporal-Pose)

For our model, we propose integrating RGB, optical flow, and joint features, where the RGB stream processes raw video frames using a spatial pre-trained ResNet, and the optical flow stream processes raw video frames using a temporal pre-trained ResNet, and the pre-computed data are stored as numpy files for efficient processing.

We first extracted human pose keypoints from each frame by using a pose estimation algorithm, and we represented each frame's keypoints as an array of coordinates (x, y) for 18 keypoints, which resulted in a 36-dimensional vector per frame. Then, for each frame, we extracted features using a pre-trained ResNet 101 model [5] to extract spatial features from the RGB frames and optical flow frames, which resulted in a final feature vector with each frame being 2048-dimensional.

Let X be the input video sequence, where $X = [x_1, x_2, ..., x_T]$ and T is the number of frames. Each frame x_t is represented as $x_t = [f_{rgb}, f_{flow}, f_{pose}]$, where f_{rgb}, f_{flow}, and f_{pose} are the feature vectors extracted from the RGB frames, optical flow frames, and pose keypoints, respectively. The combined feature vector x_t is fed into the LSTM network, which outputs a sequence of hidden states $h = [h_1, h_2, ..., h_T]$.

For each video, we combined the RGB features, optical flow features, and the pose keypoints into a single feature matrix by concatenating the features along the last dimension. This resulted in a comprehensive feature representation for each frame.

Next, we used an LSTM network to model the temporal dependencies in the video sequences. We started with two LSTM layers, one with 256 units and one with 128 units, and each layer is followed by a dropout layer with a dropout rate of 0.5 to prevent overfitting. Each of these return sequences to maintain the temporal information across frames. Then, we employed two fully connected layers with 64 and 32 units, and then used ReLU activation functions $f(x) = \max(0, x)$ and followed by dropout layers with a dropout rate of 0.5. Our output layer had a single neuron with a sigmoid activation function to predict the probability of violence in the video. We used a binary cross-entropy loss function and optimized the model by using Adam [8].

4.4. Model 4 (TSE): Three-stream Action Recognition Network with End-to-End Training (Spatio-Temporal-Pose)

This model is similar to the TSP model, except we processed the data differently, as seen in Figure 2. We integrate RGB, optical flow, and joint features, where the RGB and optical flow streams utilize ResNet, and the joint features are processed using LSTM. Instead of focusing on modular approach where we separated the feature extraction adn training, we employed an integrated approach. We computed the optical flow between consecutive frames and converted each optical flow map to a 3-channel format by adding a zero channel.

The model architecture is like the one used in the TSP model, where we extracted human pose keypoints. Then, we combined the RGB features, optical flow features, and pose keypoints like our previous model. We used the same model architecture as the previous model.

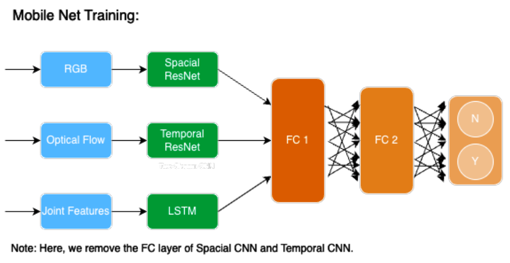


Figure 2. Possible pipeline for the method proposed in section 4.4, highlighting how the different processing techniques converge into the FC layers.

5. Experiment and Results

We used various evaluation metrics, including accuracy, precision, recall, F1 Score, and mean average precision (mAP).

5.1. Baseline Models

In the performance metrics comparison, the PoseLSTM model consistently outperforms the C3D model across the different evaluation metrics. The PoseLSTM model shows significantly higher accuracy, precision, recall, and F1 score than the C3D model, showing that it performs better in classifying violence overall. In addition, the PoseLSTM model has considerably fewer parameters than the C3D model, making it more efficient in terms of computational resources and storage requirements. Therefore, from both performance and efficiency perspectives, the PoseLSTM model holds a distinct advantage in handling the related tasks.

Table 1. Comparison of Metrics for C3D and PoseLSTM

	C3D	PoseLSTM
Loss	0.693	0.645
Accuracy	0.482	0.613
Precision	0.238	0.702
Recall	0.482	0.613
F1 Score	0.314	0.571
Total params	226965506 (865.80 MB)	116098 (453.51 KB)

5.2. Our Proposed Models

5.2.1 Three-stream Action Recognition Network with Separation of Feature Extraction and Training (TSP)

We evaluated our Three-stream Action Recognition Network with Separation of Feature Extraction and Training (Spatio-Temporal-Pose) for violence detection. The model uses pre-extracted RGB image features (stored in a numpy file) extracted from a spatial ResNet101, pre-extracted optical flow features(stored in a numpy file) from a temporal ResNet101, and pose features from OpenPose as input. These features were then fed into a sequential model with LSTM layers and dense layers with dropout to prevent overfitting. After fine-tuning, the model showed significant performance improvements.

Before fine-tuning, the model's accuracy was around 55%, indicating moderate performance. Fine-tuning involved several adjustments. The ResNet101 layers for RGB and optical flow features were fine-tuned on our dataset, while the pose features remained unchanged. In particular, we changed the dropout rate from 0.5 to 0.9. This is because of our challenge on training on a relatively small dataset. Using such aggressive dropout rate will help to prevent overfitting on a small dataset. The LSTM layers were optimized for the violence detection task, with dropout rates maintained the same to prevent overfitting. Hyperparameters like learning rate, batch size, and epochs were adjusted for better convergence. In particular, we changed learning rate from 0.001 to 0.0001, batch size from 8 to 32, epoches from 10 to 20.

After fine-tuning, the model's accuracy improved significantly from 55% to 67%. Fine-tuning allowed the ResNet101 layers to adjust their weights, leading to more relevant and accurate feature representations, improving the model's ability to distinguish between violent and non-violent actions.

The improvements can be attributed to several factors. Firstly, fine-tuning allowed the model to learn task-specific features critical for violence detection. Secondly, adjusting the pre-trained weights to our dataset ensured more relevant and precise features, leading to better performance metrics. Thirdly, fine-tuning reduced overfitting to pre-trained tasks and improved generalization to new, unseen data. Lastly, the LSTM layers could leverage more accurate temporal features, enhancing the model's ability to capture the dynamics of violent actions over time.

Challenges like limited training data were mitigated through aggressive dropout rates and data augmentation techniques. Future improvements could include leveraging larger datasets or synthetic data generation, optimizing computational resources, and exploring the trade-off between temporal resolution and performance for real-time applications.

Fine-tuning the Three-stream Action Recognition Network (with Separation of Feature Extraction and Training) significantly enhanced its performance by adapting the pre-trained feature extractors to the specific characteristics of our violence detection dataset, resulting in improved accuracy, precision, recall, and F1 score.

5.2.2 Three-stream Action Recognition Network with End-to-End Training (TSE)

Three-stream Action Recognition Network with End-to-End Training(Spatio-Temporal-Pose) uses an end-to-end training approach to integrate feature extraction into the model, using the ResNet101 layer to dynamically extract RGB and optical flow features for the violence detection task. Previous models use pre-extracted and stored generic RGB, optical flow, and pose features, and feature extraction is separated from model training. End-to-end training allows for continuous feedback and optimization between the feature extraction layer and the temporal modeling layer.

The model achieved an accuracy of 75.16% in correctly identifying violent actions in the test data set. This is significantly better than the initial model's accuracy of around 55%, demonstrating the effectiveness of the end-to-end training approach.

The precision of 0.89 means that when the model identifies an instance as violent, it is very likely to be correct, minimizing false alarms. This high precision is crucial in applications like surveillance, where false alarms can be costly and disruptive.

However, the recall of 0.67 suggests that the model misses around one-third of the actual violent instances. While it accurately identifies most violent cases, it still fails to catch some true positives.

The F1 score of 0.77 provides a balanced measure between precision and recall, indicating a strong overall performance, but there is still room for improvement, particularly in increasing recall without compromising precision.

With a total of 4,701,825 parameters, the model's complexity is manageable and aligns well with its performance metrics. With such amount of parameters, it will be possible to run on an embedded device locally (e.g. surveillance camera), without harming the users' privacy.

The improved performance can be attributed to several key factors. Firstly, by using dynamic feature extraction to intergrate feature extraction directly into the model using fine-tuned ResNet101 layers, the model can extract features specifically tailored to violence detection, rather than using pre-extracted generic visual features. This leads to more relevant and accurate feature representations. In this model, the RGB and optical flow features are extracted within the model itself using ResNet101 layers, allowing the feature extraction process to be dynamic and task-specific, leading to better performance. Secondly, by using end-to-end training to allow continuous feedback and optimization between the feature extraction and temporal modeling layers ensures that the features extracted are directly relevant to the subsequent layers, leading to better overall performance. Lastly, by using data augmentation like rotation, shifts, and flipping introduce variability in the training data, helping the model learn to recognize violent actions under different conditions and perspectives. This improves the model's robustness and generalization capabilities.

The end-to-end training model demonstrates substantial improvements in violence detection performance, with a balanced trade-off between accuracy, precision, and recall. The ability to dynamically extract task-specific features directly within the model, rather than relying on pre-extracted generic features, contributes significantly to the improved performance. Future improvements could focus on increasing recall and optimizing computational resources by fine-tuning the hyperparameters, ensuring the model's robustness and reliability in practical applications.

5.2.3 Reasoning for Correct and Incorrect Classification

This confusion matrix shows examples of both correct and incorrect predictions of violence and non-violence in video frames for our model. The top-left quadrant shows frames where actual violence was correctly predicted as violence. The top-right quadrant displays frames where actual violence was incorrectly predicted as non-violence. The bottom-left quadrant contains frames where actual non-violence was incorrectly predicted as violence. The bottom-right quadrant shows frames where actual non-violence was correctly predicted as non-violence.

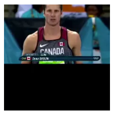
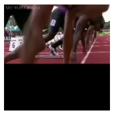
	Predicted Violence	Predicted Non-Violence
Actual Violence		
Actual Non-Violence		

Table 2. Confusion matrix of frames from videos

We think there are many reasons for correct classifications. Firstly, violent actions often involve specific movements, postures, or interactions that are visually distinctive. The model can learn these features from the RGB frames, optical flow, and pose data. Furthermore, by using a pre-trained ResNet101 to extract features, we ensure that high-level features are effectively captured, which helps better discriminate between violent and non-violent samples. In addition, the LSTM layers in your model can capture temporal dependencies, which are crucial for recognizing actions that unfold over several frames.

We think there are many reasons for incorrect classifications. Firstly, some frames might not clearly indicate violence or non-violence, making it challenging for the model to classify them accurately. For example, a frame of someone running could be part of a violent chase or a non-violent athletic event. Secondly, objects or people blocking the view (occlusions) or complex backgrounds can confuse the model. Thirdly, inconsistent or unusual poses might not be well-represented in the training data, leading to misclassification. Lastly, poor lighting or low-quality video frames can obscure important features needed for correct classification.

5.3. Discussion

5.3.1 Contribution of different features in models

In out two baseline models (C3D and PoseLSTM), the contributions of the three streams are similar to our expectations. Among them, the C3D model only uses RGB and Optical Flow, among which RGB contributes more because it can provide more detailed and dynamic scene and object information. In the second model, we only use pose feature as input, and its contribution is 100%.

In the TSP model, each feature (RGB, Optical Flow, Pose) is pre-extracted and independently input into the model for training. This independence allows the pose feature to focus on capturing the details and posture changes of the human body, which makes an important contribution to the violence detection task. Therefore, in TSP model, the contribution of pose features is higher, accounting for 25.4%. RGB features provide rich scene and object information, and their contribution ratio is 41.7%. Optical flow features can capture motion information and contribute 32.9%.

In the TSE model, an end-to-end training method closely integrates feature extraction and model training processes. This method allows the model to dynamically adjust and optimize the feature extraction process, so that each feature can serve the violence detection task more targetedly. Since RGB features can capture more details and dynamic information in end-to-end training, their importance increases significantly, accounting for 46.5%. Optical flow features make a significant contribution in capturing motion information, accounting for 35.0%. Although posture features are still important, their relative contribution decreases in end-to-end training, accounting for 18.5%. This difference is because during end-to-end training, RGB features can more fully utilize the dynamic adjustment advantages of the model, thus accounting for a larger proportion of the overall performance improvement.

We can see the comparisons between the four models in Figure 3.

5.3.2 Comparing Results

As we can see from Table 3, the TSE model outperforms the others with the lowest loss (0.513), highest accuracy (0.752), highest precision (0.89), and a strong F1 score (0.77), and has the lowest parameter count (17.9 MB). The

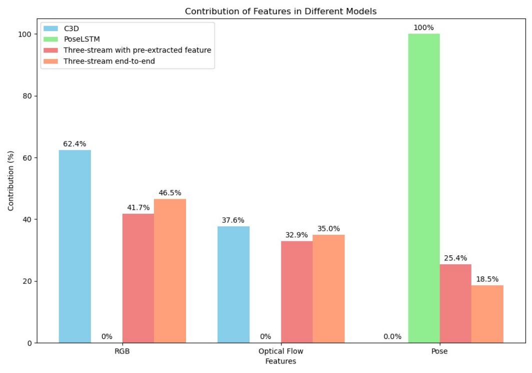


Figure 3. Description of the contribution from the four models

TSP model also performs well, with an accuracy of 0.667. PoseLSTM has good precision (0.701) and a moderate F1 score (0.571). However, it its accuracy (0.613) compared to TSE and TSP is lower. The C3D model is outperformed by all the other models, with the highest loss (0.693), lowest accuracy (0.482), and lowest precision (0.238), and weakest F1 score (0.314). Our results show that the TSE model outperforms all the other models while having fewwer parameters, making it our best model.

Metric	C3D	PoseLSTM	TSP	TSE
Loss	0.693	0.645	0.563	0.513
Accuracy	0.482	0.613	0.667	0.752
Precision	0.238	0.701	-	0.89
F1 Score	0.314	0.571	-	0.77
Params	865.8 MB	453.5 KB	493.8 KB	17.9 MB

Table 3. Comparison of Metrics for C3D, Joint Feature + LSTM, Three-Stream Pre-extracted (TSP), and Three-Stream end-to-end (TSE)

6. Conclusion & Future Work

After evaluating four models for detecting violent actions in videos, we found that the TSE model outperformed all the other models, achieving the highest accuracy (0.752), precision (0.89), and F1 score (0.77). This is because of its end-to-end training approach, which allowed for continuous optimization between feature extraction and temporal modeling layers. Our results showed that by integrating multiple streams of information, the model can detect violent actions better. The TSE model's dynamic feature extraction and continuous feedback mechanism was beneficial in capturing task-specific features, which helped it perform better than traditional methods.

There are several limitations in our study. We used a relatively small dataset that only contained binary classifications of violence versus non-violence, which limited the model's ability to generalize to more diverse and complex scenarios. Additionally, the real-time implementation on a Raspberry Pi, although initiated, was not completed due to time constraints. This can make it accessible to security cameras, which can be seen in Figure 4.

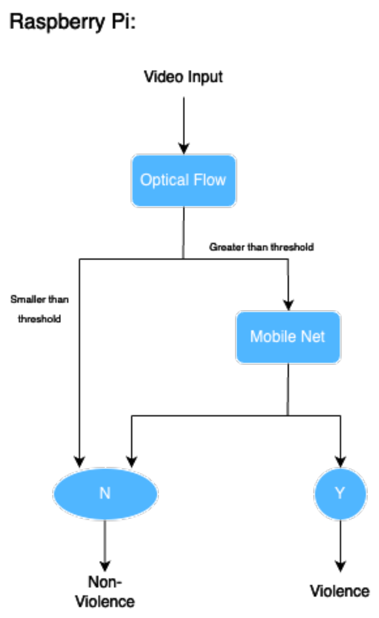


Figure 4. Flow chart for the implementation of a violence detector within a raspberry pi, focusing on computational efficiency in order to make it possible to install within security cameras.

7. Contributions & Acknowledgements

Our team consisted of three people who contributed to the project: Jermaine Zhao, Robin Li, and Nathan Tran. Jermaine worked on the methods, conducted the experiments, analyzed and presented the results in that section of the paper, and implemented the training and coding aspects of the project. Robin conducted literature reviews and created the diagrams for the project. Robin also worked on the introduction, dataset, methods, results, and conclusion sections of the paper. Nathan conducted literature reviews and worked on the Raspberry Pi integration, although we ran out of time to finish. Nathan also worked on the abstract, introduction, and the dataset section of the paper.

References

[1] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazırbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. pages 2758–2766, 2015.

[2] R. F. L. T. Du Tran, Lubomir Bourdev and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *IEEE International Conference on Computer Vision (ICCV)*, 2015(1):4489–4497, 2015.

[3] M. Elesaway, M. Hussein, and M. A. E. Massih. Real life violence situations dataset. https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset.

[4] G. Farneback. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA)*, pages 363–370, Halmstad, Sweden, 2003.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv*, 1512(03385):1–12, 2015.

[6] A. C. P. Z. Hieu H. Pham, Louahdi Khoudour and S. A. Velastin. Video-based human action recognition using deep learning: A review. *ArXiv*, 2208(03775):1–43, 2023.

[7] D. K. Judith Bütepage, Michael J. Black and H. Kjellström. Deep representation learning for human motion prediction and classification. *ArXiv*, 1702(07486):1–10, 2017.

[8] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *arXiv*, 1412(6980):1–15, 2015.

[9] S. Mathew, A. Subramanian, S. Pooja, and B. MS. Human activity recognition using deep learning approaches: Single frame cnn and convolutional lstm. *International Journal of Emerging Trends in Engineering Research*, 8(9):1–10, 2020.

[10] D. Osokin. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. *arXiv*, 1811.12004, 2018.

[11] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black. On the integration of optical flow and action recognition. *arXiv*, 1712(08416):1–10, 2017.

[12] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv*, 1406(2199):1–12, 2014.

[13] G. A. S. Surek, L. O. Seman, S. F. Stefenon, V. C. Mariani, and L. dos Santos Coelho. Video-based human activity recognition using deep learning approaches. *Sensors*, 23(6384):1–15, 2023.

[14] Y. Xie. Deep learning approaches for human action recognition in video data. *International Journal of Advanced Research in Computer Science and Software Engineering*, 13(4):1–12, 2023.