Hacettepe University

Computer Engineering Department

BBM479/480 End of Project Report

Project Details

Title	Violence Detection From Videos
Supervisor	Assoc. Prof. Aydın Kaya

Group Members

	Full Name	Student ID
1	Murat Şehzade	21827828
2	Hamza Etcibaşı	21827407
3	Berke Bayraktar	21992847

Abstract of the Project (/ 10 Points)

Explain the whole project shortly including the introduction of the field, the problem statement, your proposed solution and the methods you applied, your results and their discussion, expected impact and possible future directions. The abstract should be between 250-500 words.

The exponential growth of video data, especially in security surveillance, presents a challenge in manual monitoring and a need for automation. In this context, our project focuses on detecting and classifying acts of violence from video data. Given the surge of interest in deep learning and computer vision for video monitoring tasks, we propose an approach grounded on both deep learning techniques and other proven methods for violence detection.

We utilize a model based on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) cells, specifically leveraging a pre-trained VGG19 as a spatial feature extractor. This model works by processing frames sequentially through the TimeWarp layer, which allows for the extraction of spatial features from each frame before the sequence is passed to the LSTM layer. The LSTM layer then learns the temporal relationships between frames. To optimize our model, we experimented with different hyperparameters such as learning rate, image size, number of frames, stride, and input types (RGB, flow, and a combination of RGB and flow). Results showed that an RGB input yielded 0.98 test accuracy being the highest, indicating its efficacy in our CNN-LSTM based model.

The application and effectiveness of the proposed CNN-LSTM model, augmented by the incorporation of an ensemble of existing methodologies, have marked a significant stride in the field of automated violence detection and classification from video data. The efficacy of this model holds promising implications for the future of automated security surveillance systems. The expected impact of this study is considerable. The model demonstrates potential for deployment in real-world scenarios to enhance security surveillance operations, thereby providing a layer of automation that could transform how security monitoring is currently handled. This not only mitigates the risk of human error but also improves the efficiency of monitoring, particularly in areas with an extensive number of surveillance cameras.

Going forward, there are several directions for future exploration. The developed model can be subjected to further fine-tuning and tested on a broader array of datasets for diversified violence scenarios. This could ensure a more versatile violence detection system. In addition, although the work focused on utilizing RGB input for optimal outcomes, future work can investigate other input modalities, such as infrared or thermal imaging, to improve model robustness in varied environments.

Introduction, Problem Definition & Literature Review (/ 20 Points)

Introduce the field of your project, define your problem (as clearly as possible), review the literature (cite the papers) by explaining the proposed solutions to this problem together with limitations of these problems, lastly write your hypothesis (or research question) and summarize your proposed solution in a paragraph. Please use a scientific language (you may assume the style from the studies you cited in your literature review). You may borrow parts from your previous reports but update them with the information you obtained during the course of the project. This section should be between 750-1500 words.

The field of action detection within video streams, particularly the recognition of potentially violent incidents, has gained prominence in recent times. As the global crime rate increases, the necessity for robust and accurate detection mechanisms to respond swiftly to such incidents has become apparent. The primary objective of this project is to contribute to the development of more precise and efficient computer vision systems that can recognize violent activities or suspicious events in surveillance footage.

Recent approaches for action recognition may be roughly divided into global, frame-based methods and local, interest-point-based methods. Spatio-temporal feature points [1] are detected to depict human activity in a video as a local method [2]. It is proposed to use an unsupervised strategy similar to the bag-of-words approach to learn the probability of distribution of these feature points in [3]. Then a video can be represented with bag-of-feature techniques [4]. However, if there are too few interest points or too much movement, they could not offer enough useful details. On the other hand, global features reflect the state of motion in the frame at a certain moment in time using global features such as optical flow [5].

The problem of human action recognition was addressed in [6] using optical flow histograms based on the horizontal and vertical directions as action descriptors. Optical flow is a useful feature for motion detection and tracking because it can explain the coherent motion of moving objects and this method is improved by Gao et al. [7] by incorporating the orientation of the violent flow features resulting in oriented violent flows (OViF) features; they used both SVM [8] and Adaboost [9] as a classifier. Keçeli et al. [10] computed the optical flow to input video frames using the Lucas-Kanade method [11, 12], then several 2D templates were constructed with overlapping optical flow magnitudes and orientations. Zhou et al.[13] proposed image acceleration modality to better extract the motion attributes. They used three kinds of input i.e. RGB images for spatial networks (FightNet), optical flow between consecutive frames and acceleration images for temporal networks LSTM [14]. Soliman et al. [15] proposed a model using a pre-trained network (VGG-16) as spatial feature extractor, LSTM as temporal feature extractor and sequence of fully connected layers for classification purpose. Deniz et al. [16] presented a novel method to detect violent sequences which uses extreme acceleration patterns as the main feature by applying the Radon transform to the power spectrum of consecutive frames. As a consequence, optical flow and convolutional neural networks based classification and detection are widely used for object tracking and motion representation. They also use different machine learning algorithms to improve the performance of the detection.

Hypothesis: By utilizing a transfer learning strategy and implementing a CNN-LSTM model with a TimeWarp layer, we can effectively extract spatial and temporal features from video data and accurately classify videos into different classes using different input modalities.

Proposed Solution: The proposed solution involves building a CNN-LSTM model on top of a pre-trained VGG19 convolutional neural network. The VGG19 model serves as a spatial feature extractor, while the LSTM layer processes the sequential features extracted by the base model to learn temporal relationships between frames. A TimeWarp layer is used to feed each frame of a video into the base model sequentially, allowing for spatial feature extraction. The final output of the LSTM layer represents the entire video and is fed into fully connected layers for classification. Two methods, "squeeze" and "loop," are devised for feeding individual frames into the pre-trained VGG19 model. The choice between the methods depends on factors such as memory availability and processing speed. The

proposedsolution aims to address the challenges of small datasets by leveraging transfer learning and effectively capturing both spatial and temporal information in videos for accurate classification.

Methodology (/ 25 Points)

Explain the methodology you followed throughout the project in technical terms including datasets, data pre-processing and featurization (if relevant), computational models/algorithms you used or developed, system training/testing (if relevant), principles of model evaluation (not the results). Using equations, flow charts, etc. are encouraged. Use sub-headings for each topic. Please use a scientific language. You may borrow parts from your previous reports but update them with the information you obtained during the course of the project. This section should be between 1000-1500 words (add pages if necessary).

DATASET

For these experiments we have employed the hockey fight dataset we talked about in our earlier reports. This dataset includes 1000 videos from hockey fight videos which we split up as 800 training vs 200 test videos. Splitting was done before frame sampling which we will talk about in a moment to prevent training frames leaking into test frames which would result in experiments yielding faulty values. After the splitting is done, frames are sampled from each video such that 41-50 frames are extracted per video with a stride of 1. In addition, for this dataset, frames are 500x500 in size.

READING DATASET

We have implemented a custom dataset class to read a large dataset of videos. Initially, we attempted to read all the videos directly, but we found that the process was slow due to the time-consuming nature of video reading. To address this issue, we devised a new approach wherein we converted each video into its constituent frames and saved them in a separate folder named after the video.

This approach allowed us to read the video data as a sequence of images, and we were able to choose individual samples based on specific parameters such as numFrames and stride. The numFrames parameter specifies the number of frames to be used for each video, while the stride parameter indicates the number of frames between two consecutive frames in the selected sequence.

By using this method, we were able to select a subset of the data more efficiently, reducing the time required to load the dataset. Additionally, we were able to fine-tune the dataset selection by adjusting the numFrames and stride parameters, which gave us greater flexibility in selecting the samples that best suited our needs. Overall, our custom dataset class was able to efficiently manage and load the large dataset of videos by converting them into their constituent frames and allowing us to choose specific samples based on our requirements.

We arranged images in the following way:

DATA AUGMENTATION

To enhance the diversity and generalizability of our dataset, we employed several commonly used data augmentation techniques. Specifically, we utilized scaling, horizontal flip, center crop, corner cropping, and normalization.

Scaling was applied to randomly resize the frames, while horizontal flipping was used to create mirror images of the original frames. Center cropping was employed to extract the central portion of the frames, and corner cropping was used to extract the four corners of the frames. Finally, normalization was applied to adjust the pixel values of the frames to have a mean of zero and standard deviation of one.

It should be noted that we evaluated the performance impact of each augmentation technique on our models, and found that some methods were actually detrimental to model performance. As such, we elected to discard these techniques in favor of those that provided a net benefit to our results.

NORMALIZATION

Normalization is an important preprocessing step in video classification because it can help to ensure that the video data is consistent and comparable across different samples. Video data can vary widely in terms of their pixel values, brightness, contrast, and color, which can make it difficult to compare videos accurately. So, we calculated mean and standard deviation values of the whole dataset for each rgb channel, saved it and used these values for every experiment to normalize our dataset.

CNN-LSTM MODEL

Because our dataset is small, we prefer to use transfer learning strategies. Our model was built on top of CNN (pre-trained vgg19) as a spatial feature extractor followed by LSTM cells. Each item in our model was with the shape of (20x100x100x3) which corresponds to (frame x H x W x RGB color channels) and since vgg19 works with the 3d shape of input we used (time Distribution techniques).

Our model consists of several layers:

1) A pre-trained VGG19 convolutional neural network with its weights frozen up to the 28th layer. This serves as the base feature extractor for the model.
2) A TimeWarp layer that feeds each frame of a video into the base model sequentially. This allows the model to extract spatial features from each frame before passing the sequence to the LSTM layer.
3) An LSTM layer with 80 numbers of hidden units and 1 number of RNN layers. This layer processes the sequence of features extracted by the base model, allowing the model to learn temporal relationships between the frames.
4) An ExtractLastCell layer that extracts the final output of the LSTM layer, which is the representation of the entire video.
5) A fully connected layer with 256 hidden units and a ReLU activation function.
6) A dropout layer with a rate of 0.2, which randomly sets a fraction of the activations to zero, preventing overfitting.
7) A final fully connected layer with 2 units, which outputs the predicted class probabilities for the input video.

TimeWarp Class

Most pre-trained (CNNs) are designed to process 3-dimensional inputs of shape (RGB, height, width), whereas video data is represented as a 4-dimensional tensor of shape (frames, RGB, height, width). To use a pre-trained CNN for video data, we need to perform spatial feature extraction and then feed the output to a temporal layer, such as LSTM. So, we wrote a TimeWarper class that applies the same operation for each group of tensors. In this case, each group of tensors represents one frame, and let's assume that numFrames=20. In this case, TimeWarper processes 20 consecutive frames, represented as (frames, RGB, height, width). The VGG19 model processes each frame individually, and the output is a 2-dimensional tensor that is fed into the LSTM.

We have devised two distinct methods for implementing the concept of feeding individual frames of a video into a pre-trained vgg19 model for spatial feature extraction.

The first method, referred to as "squeeze," involves converting the input video data from a 5-dimensional array (batch size, frames, RGB channels, height, width) to a 4-dimensional array (batch size multiplied by frames, RGB channels, height, width). This allows each frame to be processed as an independent item, which is sequentially fed into the pre-trained model. However, this method consumes a larger amount of GPU memory than the second approach.

The second approach, referred to as "loop," involves iterative looping over the frames of each video and feeding them one at a time into the pre-trained model. This method may be slower and potentially less effective in learning than the first approach, but it requires less GPU memory.

The selection of which approach to utilize will depend on various factors, including the available memory, processing speed, and the specific demands of the given task.

Loss Function & Optimizer

We used cross entropy loss for training our classification models and used Adam Optimizer to speed up the training process and improve the accuracy of the model.

Cross Entropy Loss function measures the difference between the predicted probability distribution and the actual probability distribution. In the case of violence detection, the predicted probability distribution could represent the model's confidence in whether a given video clip contains violence or not. The actual probability distribution could be a one-hot encoding of the video clip's true label (i.e., whether it contains violence or not). The cross-entropy loss encourages the model to assign higher probabilities to the correct class and lower probabilities to the incorrect class, ultimately improving its classification accuracy.

Results & Discussion (/ 30 Points)

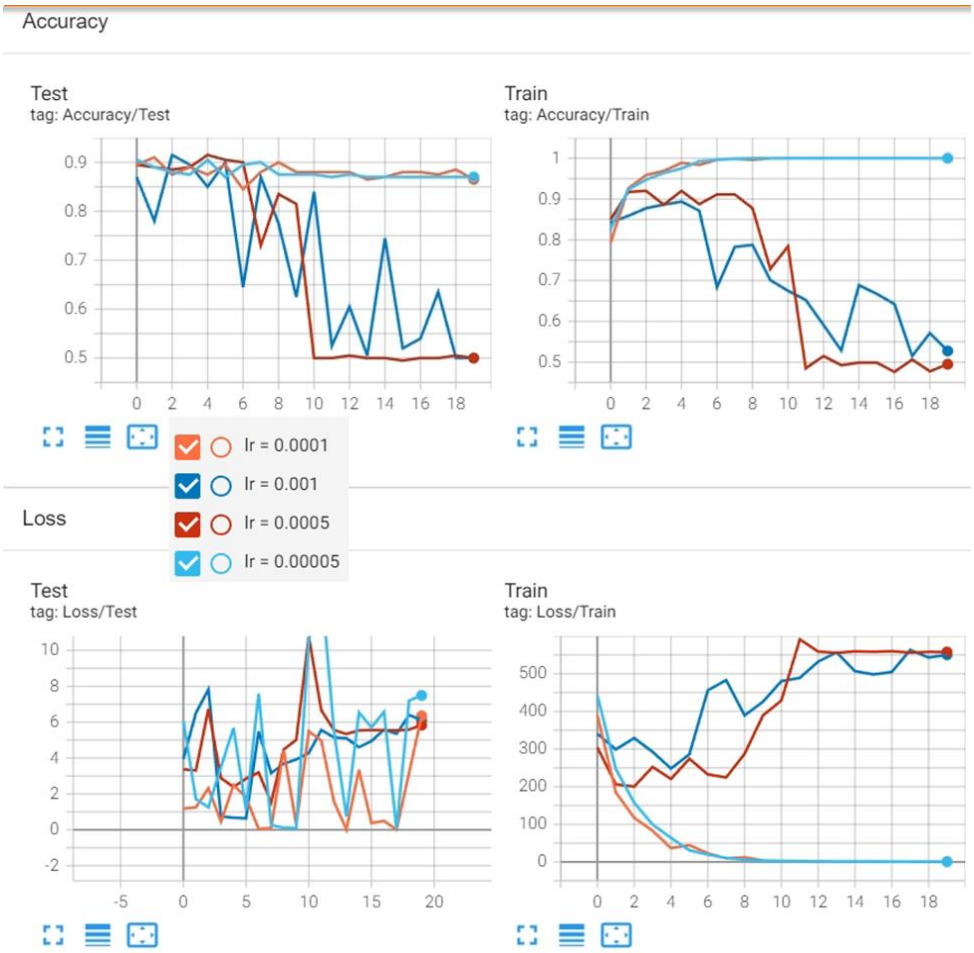
Explain your results in detail including system/model train/validation/optimization analysis, performance evaluation and comparison with the state-of-the-art (if relevant), ablation study (if relevant), a use-case analysis or the demo of the product (if relevant), and additional points related to your project. Also include the discussion of each piece of result (i.e., what would be the reason behind obtaining this outcome, what is the meaning of this result, etc.). Include figures and tables to summarize quantitative results. Use sub-headings for each topic. This section should be between 1000-2000 words (add pages if necessary).

We carried out our experiments starting with rgb inputs and chose many hyperparameters based on these experiments.

A) INPUT_TYPE = 'RGB'

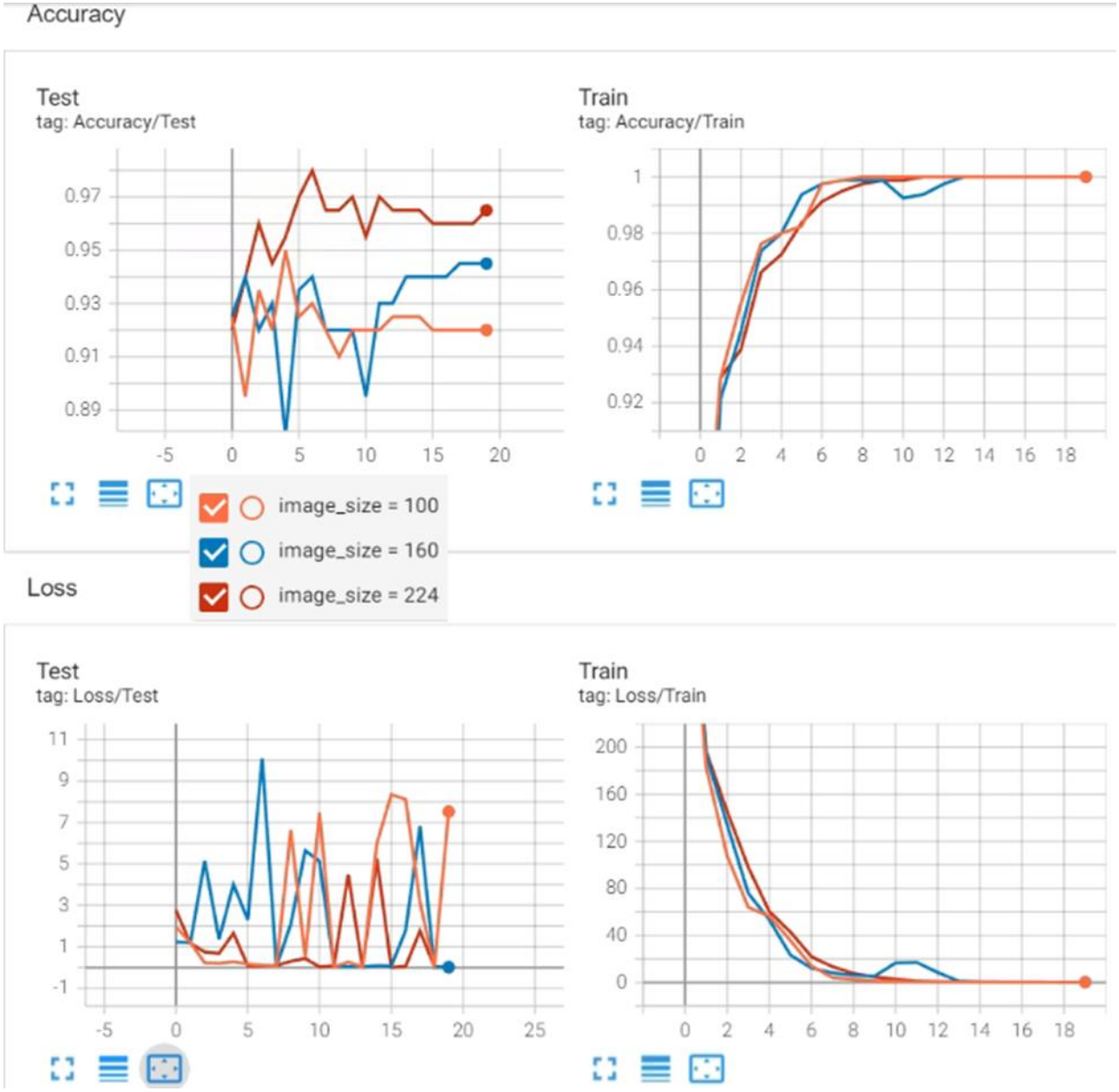
1) LEARNING RATE

In the course of our experimentation, we conducted multiple trials using different learning rates within the range of 1e-3 to 5*1e-5. Based on our findings, we were able to determine that a **learning rate of 1e-4** was most suitable for our particular model. This conclusion was reached after a thorough analysis of the results obtained from the various trials, and we believe it to be an optimal choice for achieving the desired outcomes.



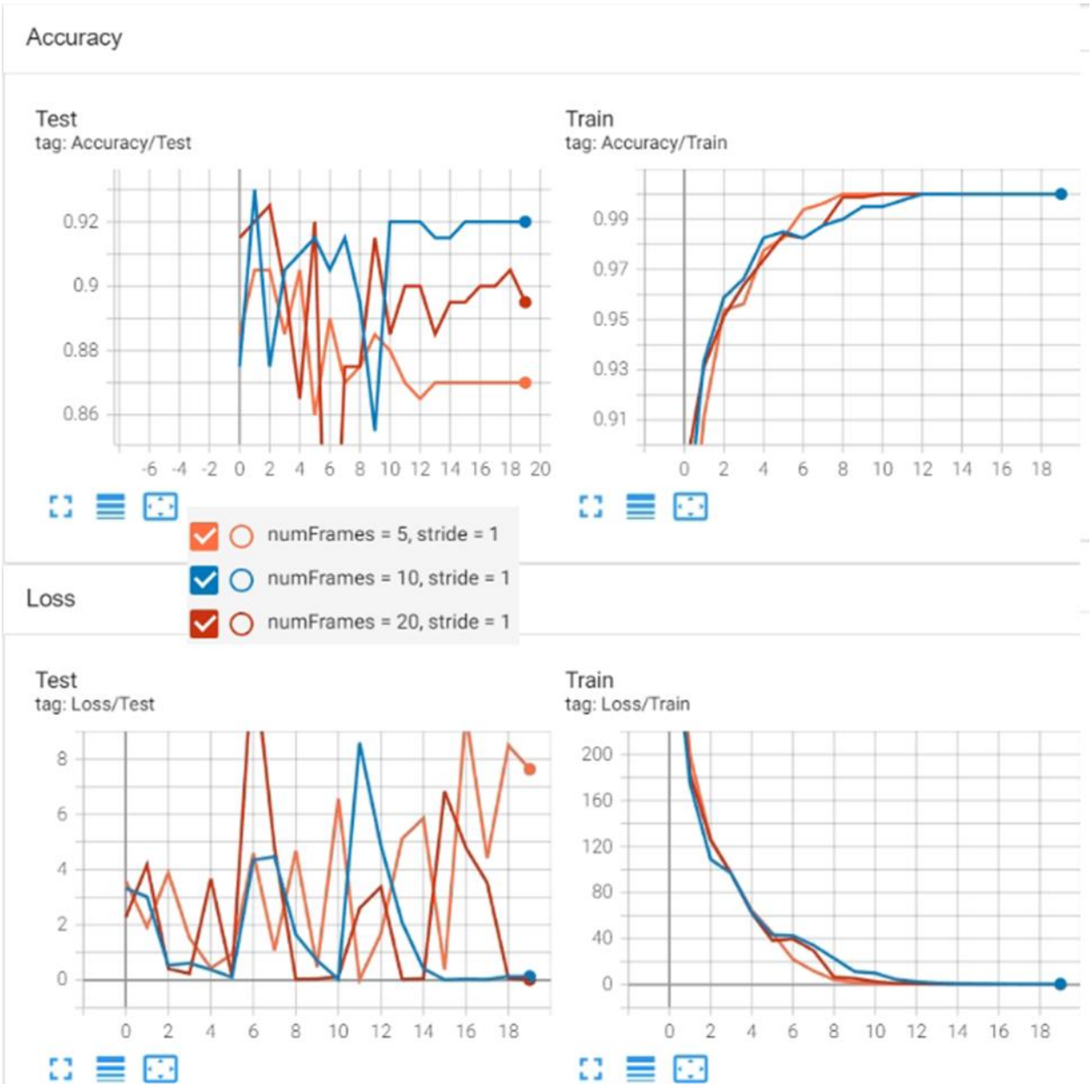
2) IMAGE SIZE

Our experimental process involved testing various image sizes as a critical hyperparameter that significantly affects both training and test time. After conducting a series of experiments using image sizes of 100, 160, and 224, we found that larger image sizes were more effective in understanding video characteristics. Specifically, we observed that an image size of 224 yielded the highest accuracy and lowest test loss. However, it was also four times slower than the smallest image size of 100, while an image size of 160 was twice as slow as 100. Given these factors, we ultimately decided to adopt an image size of 100 as the default setting to prioritize overall speed in our processes.



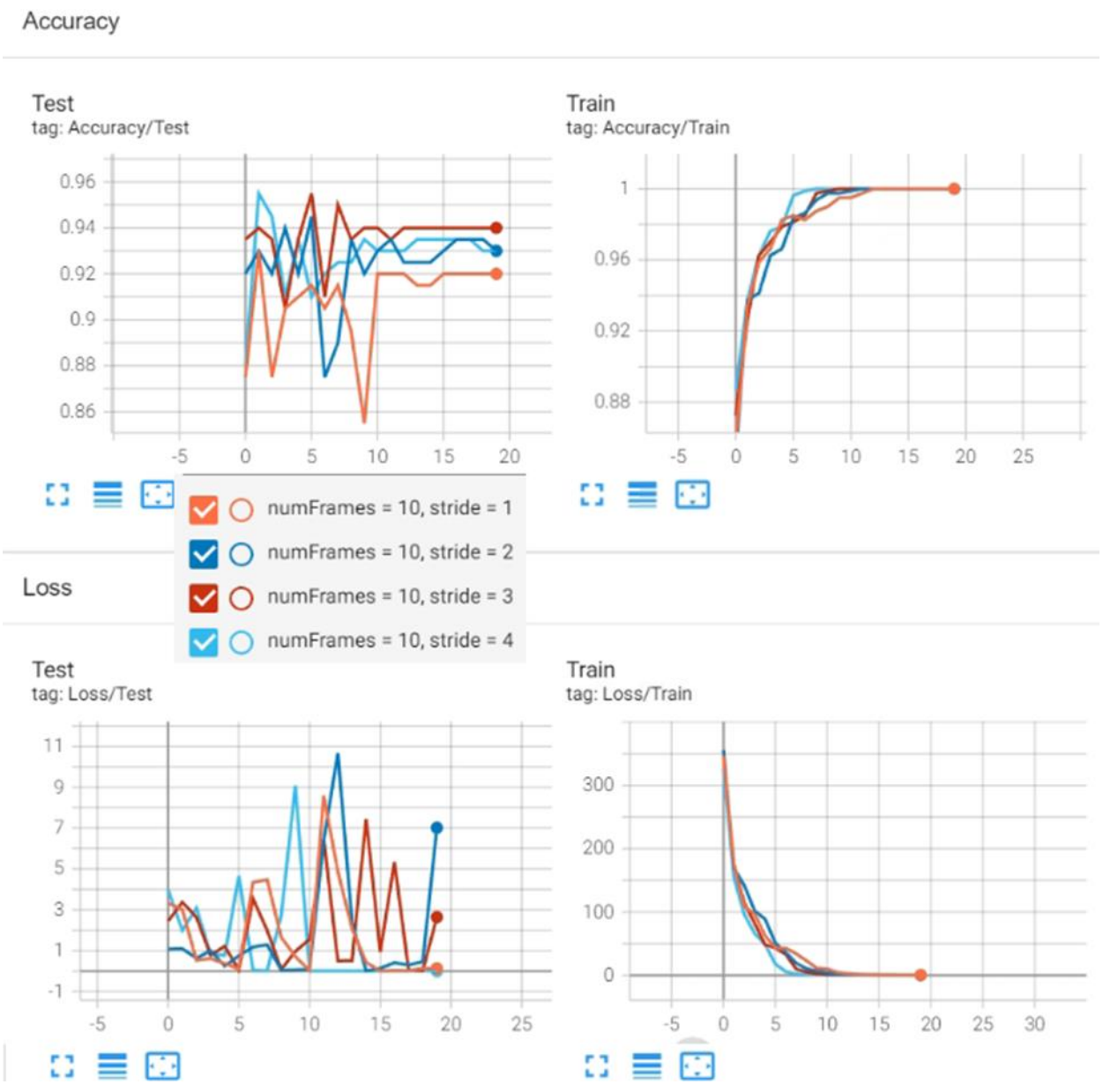
3) NUMBER OF FRAMES

Throughout the course of our experimentation, we conducted numerous trials, utilizing varying numbers of frames ranging from 5 to 20. After analyzing the results, we were able to determine that 10 frames were the most suitable for our particular model. Our conclusion was reached after a thorough analysis of the data obtained from the various trials, and we strongly believe that this is the optimal choice for achieving the desired outcomes.



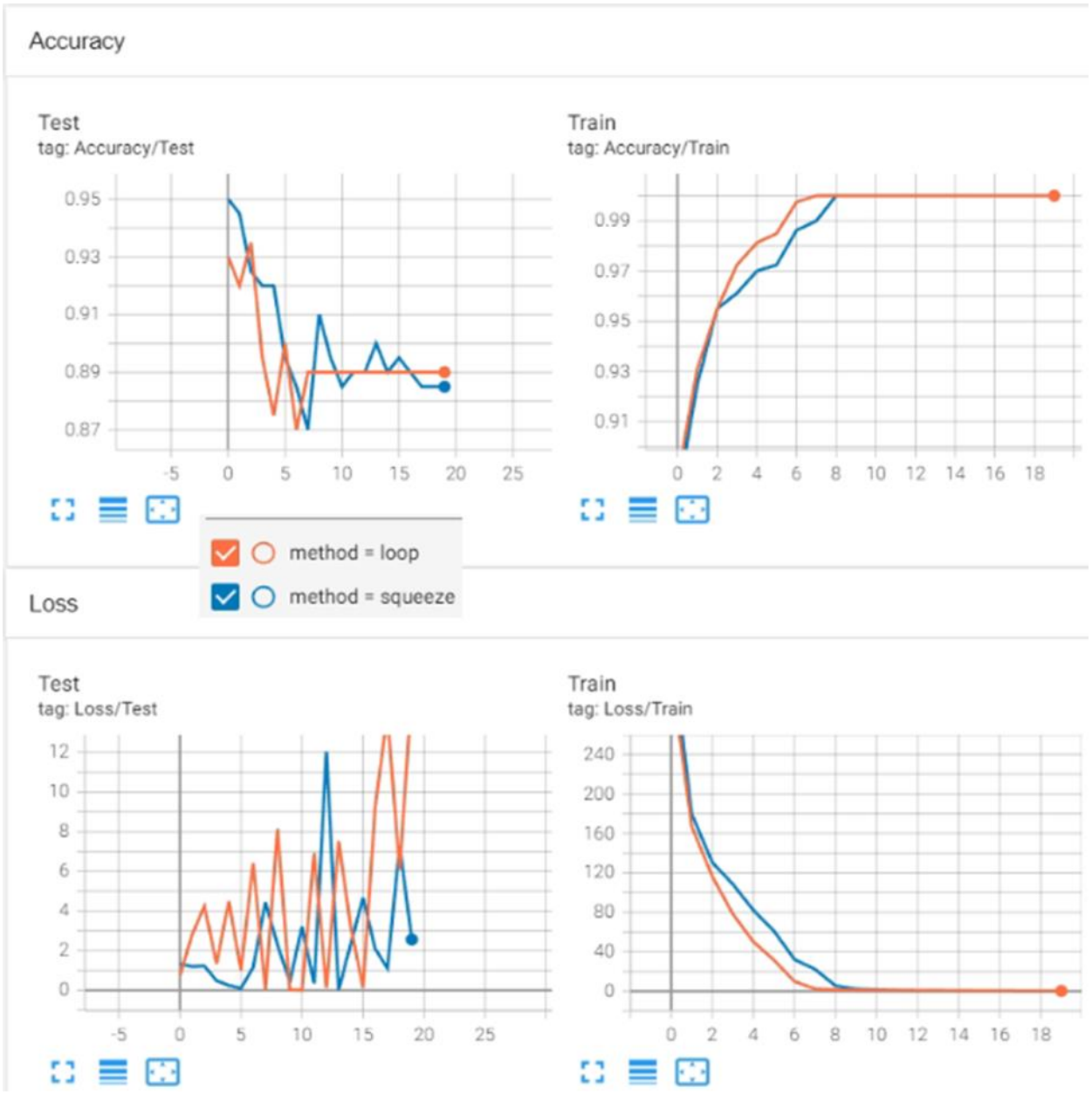
4) STRIDE (using same numFrames)

Another parameter we put into test was stride to consider when extracting images from videos. We have tested 4 different parameters from stride=1 to stride=4. Based on results obtained, which can be seen on the graph below, we can see that stride of 3 performs best with a score of 0.94 by a small margin where as stride of 1 performs lowest with 0.92. This can be attributed to the fact that, algorithm is able to generalize better when looking at frames with stride of 3 instead of 1. Nonetheless, bigger stride doesnt' always mean better results. As it can be seen on the graph, stride of 4 reduces score to 0.93. Although a marginal difference, it could be exacerbated when working on a larger dataset where it would make sense to try different stride values with more difference.



5) Loop vs Squeeze

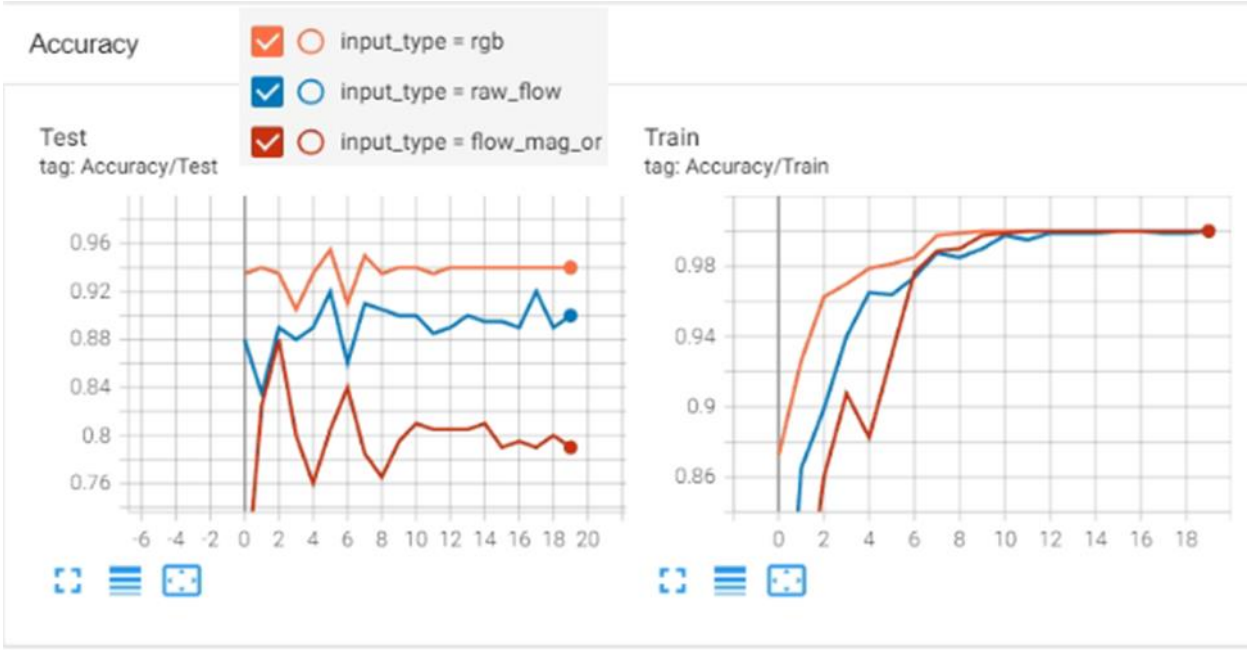
During the course of our experimentation, we conducted several trials to determine the most appropriate method for our model - loop or squeeze. Our findings revealed that the squeeze method was more suitable for our specific model. This conclusion was reached after a thorough analysis of the results obtained from the multiple trials. We firmly believe that this is the optimal choice for achieving the desired outcomes based on our rigorous evaluation.



B) INPUT_TYPE = ‘FLOW’

In our study, we evaluated the efficacy of optical flow in analyzing video data. Specifically, we utilized the Lucas Canade algorithm to calculate optical flow between every two consecutive frames in our dataset, which yielded raw flow images. We then computed the magnitude and orientation between these two images and tested the effectiveness of using a combination of flow, magnitude, and orientation data in our analysis.

To compare the results obtained from these methods, we generated accuracy graphs and compared them with the accuracy of using RGB input modality. This evaluation process was conducted with a rigorous and systematic approach, utilizing multiple trials to ensure the validity of our findings.

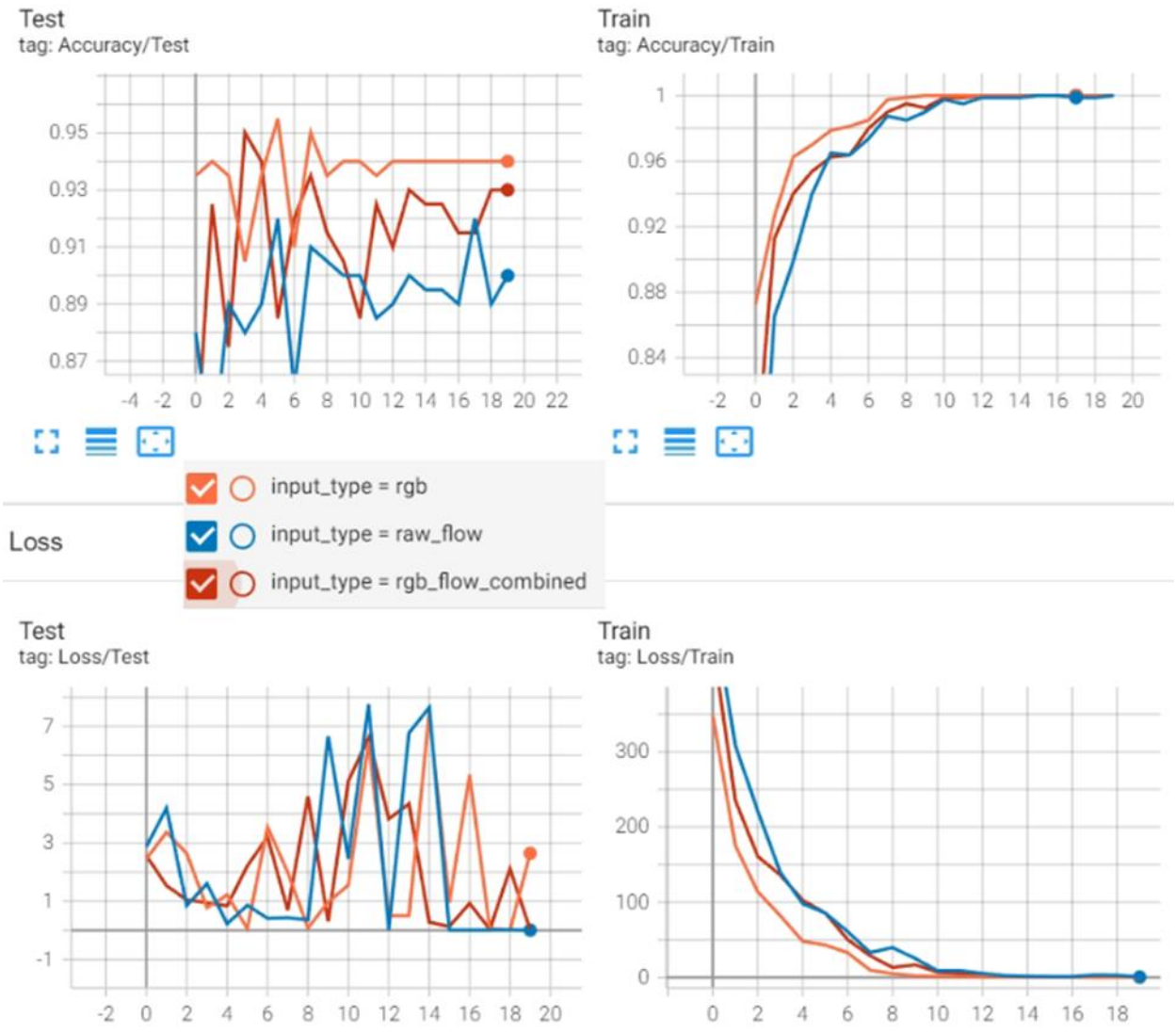


We have experimented with 3 different input types as explained in the earlier section. When using RGB only mode, the model achieved a test accuracy of around 0.94 being the highest. Whereas opting for raw flow without rgb had a significant performance hit, dropping test accuracy to around 0.90. When flow was combined with magnitude orientation. It had huge detrimental effects on the test accuracy as well, dropping it to around 0.79. However, we were anticipating the opposite, where magnitude orientation would have performance benefits.

C) INPUT_TYPE = 'RGB + FLOW'

Lastly, our experimental process involved conducting three distinct trials to compare the efficacy of different input types - RGB, flow, and a combination of RGB and flow - in our CNN-LSTM based model. Our analysis of the results revealed that the RGB input was the most effective modality for our model. Specifically, we observed that the combination of RGB and flow was slightly less effective than the RGB input alone, as illustrated in the plotted data below.

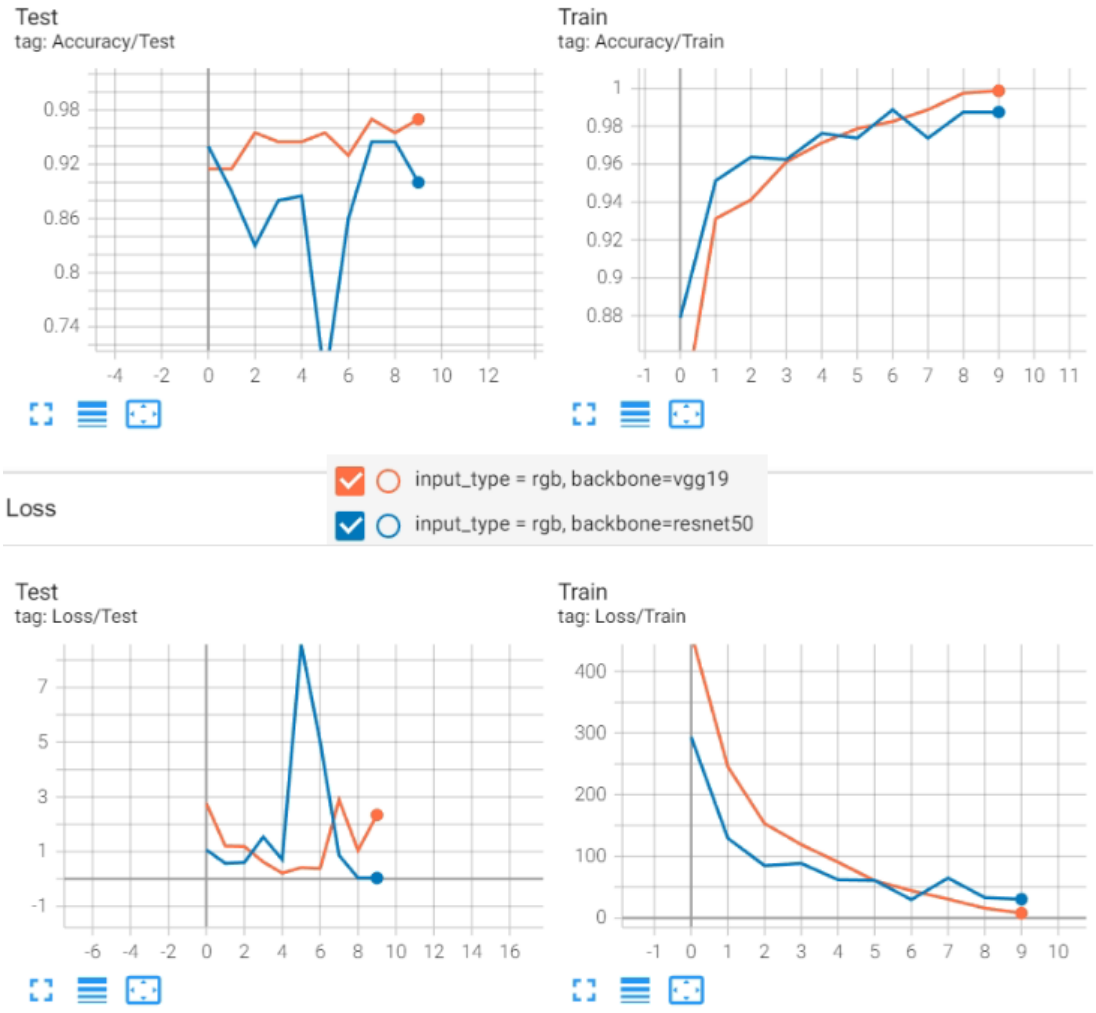
Our study utilized a rigorous and systematic approach to testing the effectiveness of different input types, employing multiple trials to ensure the validity of our findings. Based on our results, we believe that the RGB input is the optimal choice for achieving the desired outcomes in our CNN-LSTM based model.



2) BACKBONE: VGG19 vs RESNET50

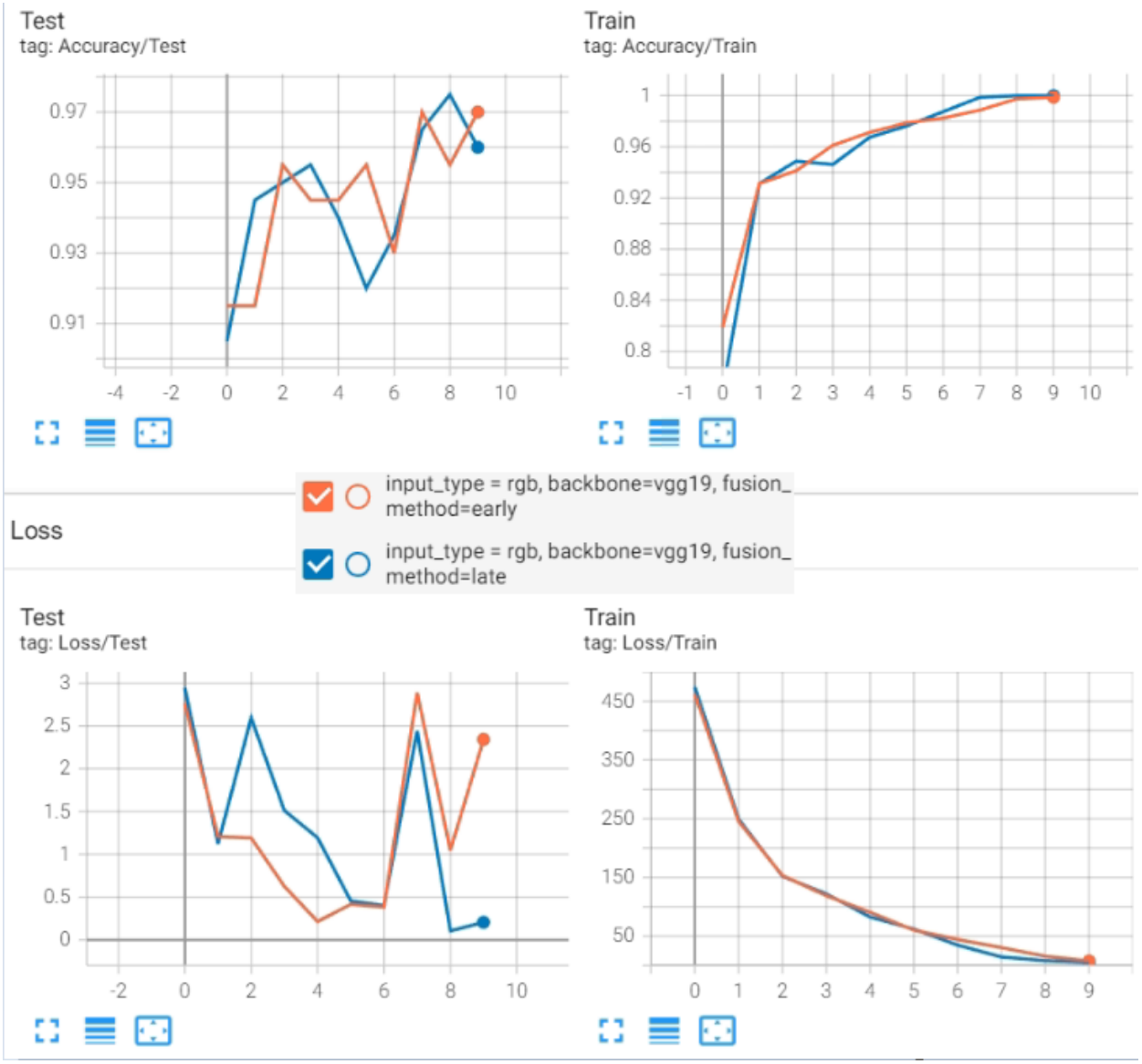
In our comprehensive evaluation of the VGG19 and ResNet50 backbones for a specific task, we have found that the VGG19 architecture demonstrates superior performance in terms of accuracy and convergence to 1.00 on the training set. Furthermore, the VGG19 model exhibits a more consistent loss and accuracy plot throughout the training process.

Although ResNet50 has demonstrated better accuracy in widely recognized image classification tasks such as ImageNet. We think that in our specific task, the dataset used in our evaluation had a simpler structure or contained features that were better captured by the shallower VGG19 network. In such cases, the additional depth of ResNet50 may not provide a significant advantage and could even lead to overfitting or increased complexity without substantial performance gains.



3) FUSION METHOD: EARLY vs LATE FUSION

For the best configuration, where we use RGB input type, vgg19 as a backbone, numFrames=10 and stride=3, squeeze method and image_size=224; we didn't see any significant change between utilizing early and late fusion methods.



The Impact and Future Directions (/ 15 Points)

Explain the potential (or current if exist) impacts of your outcome in terms of how the methods and results will be used in real life, how it will change an existing process, or where it will be published, etc. Also, explain what would be the next step if the project is continued in the future, what kind of qualitative and/or quantitative updates can be made, shorty, where this project can go from here? This section should be between 250-500 words.

For our project we focused more on development of the model and its performance. Therefore, in this stage of our project there aren't any current real life uses to it. Nevertheless, the potential for future software applications should not be dismissed outright. Particularly, the results of this study could contribute significantly to the development of a state-of-the-art software platform aimed at enhancing security surveillance operations. In practical terms, this software could be utilized to automate the process of security surveillance in real-world scenarios. By integrating our research findings with the capabilities of real-time security camera footage analysis, the developed software would possess the potential to recognize and respond to a myriad of security incidents autonomously. This, in turn, could contribute to improvements in overall security management efficacy and efficiency. The further development and maturation of such software applications, once fully realized, may eventually result in a marketable product. This product could be of interest to a wide array of potential users, spanning from private businesses to public sector organizations, each seeking to support their security infrastructure.

The findings from our developed model's performance in violence detection indicate potential avenues for further enhancement and exploration. One avenue for improvement involves subjecting the model to additional fine-tuning and evaluation on a more diverse range of datasets that encompass a broader array of violence scenarios. By doing so, we can ensure that the violence detection system becomes more versatile and capable of accurately identifying instances of violence across various contexts.

Moreover, while our current work primarily focuses on utilizing RGB input for optimal outcomes, future research can delve into investigating alternative input modalities to improve the model's robustness in different environments. For instance, exploring the integration of infrared or thermal imaging as input sources could potentially provide valuable insights and enhance the model's effectiveness in detecting violence. Such modalities can capture additional visual cues or temperature variations that might be indicative of aggressive or violent behavior, augmenting the system's overall performance and adaptability. By conducting further fine-tuning and evaluation on a diverse range of datasets, and exploring alternative input modalities, we can enhance the versatility and robustness of the violence detection system. This would enable its applicability in various real-world scenarios, improving its efficacy in detecting and preventing violent incidents.

References

[1] Laptev, I. (2005). On space-time interest points. International journal of computer vision, 64(2), 107-123.

[2] Niebles, J. C., & Fei-Fei, L. (2007, June). A hierarchical model of shape and appearance for human action classification. In 2007 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[3] Niebles, J. C., Wang, H., & Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. International journal of computer vision, 79(3), 299-318.

[4] Liu, J., Luo, J., & Shah, M. (2009, June). Recognizing realistic actions from videos “in the wild”. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1996-2003). IEEE.

[5] Danafar, S., & Gheissari, N. (2007, November). Action recognition for surveillance applications using optic flow and SVM. In Asian Conference on Computer Vision (pp. 457-466). Springer, Berlin, Heidelberg.

[6] Wang, H., Klaser, A., Schmid, C., & Cheng-Lin, L. (2011, June). Action recognition by dense trajectories. Computer Vision and Pattern Recognition (CVPR). In 2011 IEEE Conference on (pp. 3169-3176).

[7] Gao, Y., Liu, H., Sun, X., Wang, C., & Liu, Y. (2016). Violence detection using oriented violent flows. Image and vision computing, 48, 37-41.

[8] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[9] Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 28(2), 337-407.

[10] Keçeli, A. S., & Kaya, A. Y. D. I. N. (2017). Violent activity detection with transfer learning method. Electronics Letters, 53(15), 1047-1048.

[11] Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision (Vol. 81, pp. 674-679).

[12] Lucas, B. D. (1985). Generalized image matching by the method of differences. Carnegie Mellon University.

[13] Zhou, P., Ding, Q., Luo, H., & Hou, X. (2017, June). Violent interaction detection in video based on deep learning. In Journal of physics: conference series (Vol. 844, No. 1, p. 012044). IOP Publishing.

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[15] Soliman, M. M., Kamal, M. H., Nashed, M. A. E. M., Mostafa, Y. M., Chawky, B. S., & Khattab, D. (2019, December). Violence recognition from videos using deep learning techniques. In 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS) (pp. 80-85). IEEE.

[16] Deniz, O., Serrano, I., Bueno, G., & Kim, T. K. (2014, January). Fast violence detection in video. In 2014 international conference on computer vision theory and applications (VISAPP) (Vol. 2, pp. 478-485).