Deep Learning: A Statistical Perspective
Violent Crowds Video Classification - A Deep Learning Approach

Nils Gumaelius [* 1 2] **Lukas Dreier** [* 1 3]

Abstract

Security forces have to manually observe plenty of video material from CCTVs or other surveillance cameras whether there are any violent actions occurring or not. In this paper we describe the implementation of a video classifier to automate this task. For that, well known deep learning techniques like Convolutional Neural Network (CNN) and Long short-term memory (LSTM) are used and combined with hand crafted features like the Optical Flow to get the best working model. The main focus is to gain deeper understanding of the utilization of temporal features for deep learning models. The performance is benchmarked against proposals from literature. The performance of the best final model is comparable to these state-of-the-art implementations.

1. Introduction

Image classification is a well understood topic in deep learning. Since videos are rather similar to images, i.e. it is simply a sequence of images, deep learning is expected to be promising for videos as well. Over the last few years research experienced considerable progress regarding models for video classification. Nevertheless, there are several unresolved issues, i.e. the visualization of deep learning models or the automated generation of features. Furthermore, most of the benchmark data sets in video classification have a lot of diverse classes, which makes them more reliant on the feature generating model.

Classification on this data set is also very relevant for the modern society. A well performing implementation could help in detecting violence, riots and other dangers quickly. Video classification can be used for everything from quickly finding where to call the police, to finding scenes in movies and videos which, for example, should be censored in terms of child protection. Furthermore, the methodology can be transferred to any other data set which requires the temporal structure.

This project aims to get a deeper understanding on how temporal dependencies can be reflected in deep learning models. Time is an additional dimension in the data which contains information. It is possible that one image may look harmless if you look at it separated from the previous and the next image. Since the input is only a sequence of images we have to generate features which point out the dependencies regarding time. For this purpose we look for a data set containing videos which are hard to classify by only one single frame. The classification of one single frame is still the predominant way to classify videos. For completeness and benchmarking reasons we implement a single frame model as well. This paper gives an overview of different approaches and an understanding why some approaches are appropriate and others are not.

In this project we implement CNNs for prediction and feature generation, for the spatial features. To utilize the temporal features LSTMs are trained on sequences of the spatial features. To further improve the model the usage of transfer learning and Optical Flow is explored and analysed. We propose a two branch model with a pretrained CNN and the Optical Flow CNN as visualized in Figure 1 to obtain the best performance. We choose CNNs because of their remarkable performance in feature extraction of images and LSTMs since there architecture is predestined for the temporal structure of the data. The evaluation of the models is held on an open source data set containing violent and non-violent videos from a research group of the Open University Israel (Hassner et al., 2012).

[*]Equal contribution [1]Department of Statistics, Seoul National University, Seoul, Republic of Korea [2]Department of Engineering Sciences, Uppsala University, Uppsala, Sweden [3]Department of Mathematics, Technical University Munich, Munich, Germany. Correspondence to: Nils Gumaelius <nils.gumaelius@gmail.com>, Lukas Dreier <lukas.dreier@gmx.de>.

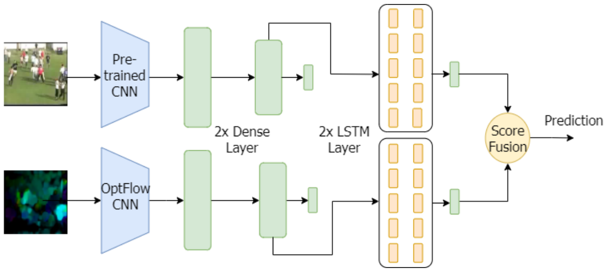


Figure 1. Proposed model

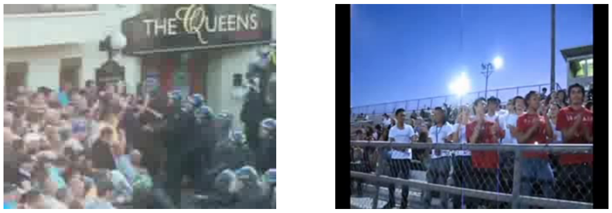
The remainder of the paper is organized as follows. Section 2 reviews the original research and the current state of the research in video classification. Section 3 explains our methodology for building various video classification models. Section 4 contains the performance evaluations of our model compared to the benchmark models. We finish the paper with a conclusion in Section 5 and give an outlook on further research topics.

2. Related Work

The understanding of deep learning video classification is very limited and not fully explored compared to image classification. State-of-the-art models for video classification often use handcrafted features and no deep learning methods. In the following Section we give an overview over related methods for classifying videos, and the current research state regarding video classification with deep neural networks.

2.1. Original research

A study on classifying violent versus non-violent videos using vector flow was conducted by (Hassner et al., 2012). They labeled 245 videos ranging from 3 to 6 seconds. After their research they made the data set public for everyone to download. Their data set consists of publicly available videos fetched from YouTube, which show e.g. soccer fans in a stadium, hooligans or crowds. Examples are shown in Figure 2.



Violent

Non-violent

Figure 2. Two frames of the data set as images

Their approach considers statistics on how flow-vector magnitudes change over time represented by the Violent Flows (ViF) descriptor. Afterwards, they are classified as violent or non-violent using Support Vector Machines.

2.2. State-of-the-art model

(Zhou P, 2018) presents a model where they use established handcrafted feature extracting methods such as Histogram of Oriented Gradient (HOG) and Histograms of Optical Flow (HOF). Instead of using methods around found interest points they use it at the highest magnitude of movement in the frame. Then, they use Bag of Words methods for processing their generated features and Support Vector Machines for classifying their data. This model shall serve as benchmark for our evaluation because, to the best of our knowledge, this model achieved the best performance.

2.3. Overview of Deep learning video classification methods

In the beginning of the exploration of video classification, the predominant approach is the single frame model. In most cases it is considered sufficient to evaluate an image classification model on one frame of the video. (Karpathy et al., 2014) provide an overview of different video classification models. They propose various variations of CNNs. It ranges from the single frame model to models in which they use every frame and fuse them successively. The architecture in which they use every frame and fuse them seem to be quite promising, nonetheless the computational time would be immense.

As soon as one thinks of the reflection of time in deep neural networks LSTMs might come in your mind. LSTMs seem to be promising in capturing the time structure of the features and it has been done some research regarding their fit in video classification by (Ng et al., 2015).

We utilize both approaches and use a combination in the form of the score fusion. Further, we explain the respective models.

3. Methodology

In this section, we describe our applied methodology. Basically, our video classifier can be split in four parts. We start with the data acquisition and the preporcessing of the acquired data. After the preprocessing, either CNNs and/or Optical Flow in different variations are used for feature generation of the data. LSTMs or CNNs are then trained on the generated features, and are used for predicting a score expressing the probability of violence in the video. In the end some postprocessing is done to get the final reusults. The process of the methodology is visualized in Figure 3.

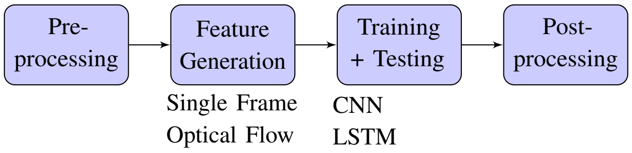


Figure 3. Visualization of the methodology

3.1. Preprocessing

For understanding the preprocessing of video data, it is essential to grasp their structure. Simply spoken, videos are temporally ordered images. For loading video data, several single frames are loaded and stored in a matrix reflecting the time, the height, the width of the frame and the color of the frame. Since the videos in the data set have different durations, the number of frames differ within the data set. To solve this problem we implement three different options to preprocess video data. All of the options have in common that we determin the minimum number of frames n, i.e. we do not perform any upsampling techniques to get more frames per video.

1. Take the first n frames
2. Distribute n points equidistantly and take the n frames which are closest to them
3. Distribute n points equidistantly and weigh the n frames with the bigger and smaller one accordingly

Throughout the project we use the third option since it captures some temporal information by itself.

3.2. Feature Generation

After the preprocessing it is essential to find the relevant features for the model. The goal is to find features which are able to reflect the temporal dependence of two or more frames. Two approaches are implemented, which can be combined easily. One approach is to use a pretrained CNN for feature generation of every single frame. The other method is to calculate and generate new frames with Optical Flow.

3.2.1. CNN

CNN are very commonly used in image problems. It takes an image as input and usually consists of groups of convolutional and pooling layers stacked on each other topped with fully-connected layers in the end. In the process of classifying or visualizing images the CNN is trained to generate features for its prediction. The output of the CNN comes from an activation layer in the end which assigns weights to a value between 0 and 1 resulting in the prediction. Instead of taking the output from the last layer, taking the output of the second last layer will result in features generated for the model to make the prediction. To generate the features of the regular frames the pretrained model InceptionV3, trained on ImageNets data, is used. The last two layers are then retrained for the violent crowd data in order to fine-tune the model. The same architecture is used for feature extraction of the Optical Flow data, but the whole model is retrained, due to the lack of correlation between OpticalFlow images and the ImageNet images.

3.2.2. Optical Flow

Optical Flow is a common technique in working with videos. In general, it provides information on how the pixels move throughout time in a video. It is a more sophisticated version of taking differences between frames. Depending on calculating the Optical Flow only for a bunch of pixel or all pixel the procedure is called sparse or dense Optical Flow. We provide an explanation of the two techniques and visualize the Optical Flow based on the video in Figure 4. Afterwards, we discuss the advantages and drawbacks.

Figure 4. Frames of violent video

Firstly, we outline the mathematical foundations of the Optical Flow in general. The Optical Flow is also known as the Lucas-Kanade method, named after the two scientists (Lucas & Kanade, 1981). Since traditional image registration methods, i.e. methods to transform several images into one, tend to be very costly and not robust enough against rotation they proposed this new method. The underlying assumption of the method is that two frames are in approximate registration. Mathematically spoken, there exist two functions $F(x)$ and $G(x)$ which return the respective pixel value at the location x for different time points. The method should then find a vector h which minimizes an arbitrary norm of

$F(x+h)$ and $G(x)$. Commonly used is the $\mathcal{L}_1$-norm or the $\mathcal{L}_2$-norm. The vector h indicates the movement of a pixel. We will shortly explain how this h is derived. $F(x+h)$ can be rewritten as

$$F(x+h) \approx F(x) + hF'(x)$$

which can be derived through linear approximation. Since we want to minimize the difference between $E = \sum_x \big[F(x+h) - G(x)\big]^2$ we set the derivative of E to 0 and get the equation

$$\begin{aligned}0 &= \frac{\partial E}{\partial h} \\ &\approx \frac{\partial}{\partial h} \sum_x \big[F(x) + hF'(x) - G(x)\big]^2 \\ &= \sum_x 2F'(x)\big[F(x) + hF'(x) - G(x)\big].\end{aligned}$$

Rearranging the equation yields

$$h \approx \frac{\sum_x F(x)\big[G(x) - F(X)\big]}{\sum_x F(x)^2}$$

which iteratively can be used to obtain the best h.

Sparse

The foundations for the Lucas-Kanade filter are built such that the remaining question is how the set of pixels is chosen. In this project a method based on a corner detection technique proposed by (Jianbo Shi & Tomasi, 1994) is chosen. They describe a feature selection technique which quantifies the dissimilarity of features in other frames. They aim to determine windows which are defined by large gradients in both directions. Each window gets a score R and, depending on that value, every window is either classified as flat, edge or corner. The score R is calculated by taking the minimum of the two eigenvalues λ_1 and λ_2 of the matrix Z. Z can be seen as a matrix measuring the magnitude of the gradient in both directions. The threshold guarantees that on the one hand the feature is above the noise level and on the other hand that it is well conditioned.

Combining these two techniques allows us to initially find relevant features to track and then calculate the sparse Optical Flow to observe the behavior of them. The implementation of these two methods is explained and implemented by David Stavens of the Stanford AI Lab (Stavens, 2007). We use the sparse Optical Flow in two ways. Firstly, we use only the last frame of the Optical Flow as input to a single frame classifier. Secondly, we use the whole sequence of Optical Flow frames as input to a multi frame classifier.

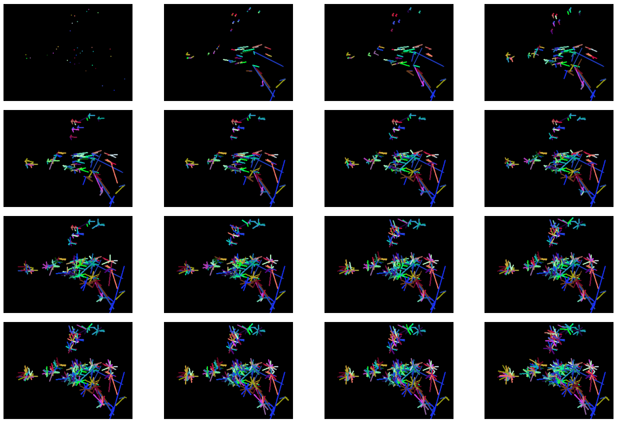


Figure 5. Visualization of sparse Optical Flow for a violent video

In Figure 5 the sparse Optical Flow is visualized. In the first picture most of the relevant pictures are people in the crowd. During the video these pixel move up and down and to the right and to the left which is typical for a violent video. In non-violent videos the pixel movement looks more controlled and it is reduced to upward and downward movements as you can see in Figure 6.

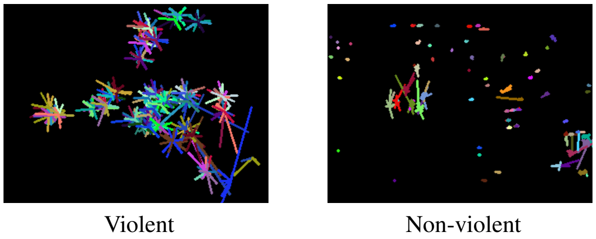


Figure 6. Comparison between the last frames of the sparse Optical Flow for violent (left) and non-violent (right) videos

Dense

The dense Optical Flow works in the same way as the sparse Optical Flow. Instead of computing the Optical Flow for a determined pixel set it computes the flow for pixels in the frame. In this project an algorithm based on (Farnebäck, 2003)s proposed algorithm is used. The algorithm outputs a 2-channel array with the Optical Flow vectors (u, v) which describes the motion in x and y direction between the frames. These vectors are shown with colour-coding (direction corresponding to Hue value, and magnitude corresponding to intensity) in Figure 7.

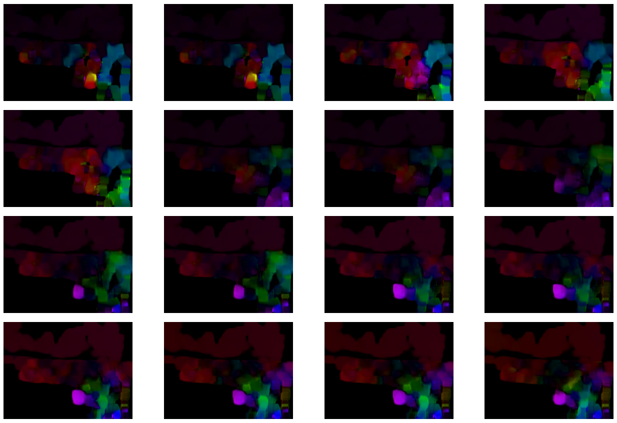


Figure 7. Visualization of dense Optical Flow for a violent video

Advantages of Optical Flow
Our goal is to express the temporal structures through features. Both Optical Flow implementations can do that. Since they monitor the changes throughout frames it is beneficial for the model to understand the temporal dependence. Further, parts of the image which do not move significantly are just black. The advantage of that representation is on the one hand that the matrix is sparse and on the other hand that these parts can be neglected by the model.

3.3. Training and Testing

The generated features are then used to train or test the models. First, the proposed LSTM network is discussed and afterwards the CNN single frame model.

3.3.1. LSTM

LSTMs were initially proposed by (Schmidhuber, 1997). The idea of LSTMs is to avoid that a deep network looses information and that the gradient deviates for early layers. For this purpose an input gate i_t, an output gate o_t and a forget gate f_t are introduced.

$$\begin{aligned}
f_t &= \sigma_t(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_t(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= \sigma_h(c_t)
\end{aligned}$$

W and U are weight matrices and b is a vector trained during the process. σ_g is the sigmoid activation function and σ_c the hyperbolic tangent function. $\circ$ is the elementwise product.

This architecture allows the network to process sequential data and through the different gates the information flow is regulated. The processing of sequential data is a major advantage for our purposes since it is possible to reflect temporal features in the model architecture. LSTMs are used both for the raw data as well as for the Optical Flow processed data.

After the features are extracted from the frames in the CNNs, they are stored into sequences, representing all frames in each video. For both the OpticalFlow branch and the regular image branch two layers of LSTM networks are used, with the first layer having 1,024 hidden units and the second one 512. In between each layer a dropout of 0.4 is used. The LSTM part of the network is trained with batch size 10 for the Optical Flow and 20 for the regularly generated features. The models are trained for 175 respective 300 epochs. Adam is used as the optimizer with a learning rate of 10^{-3}. For the prediction an dense layer with 256 neurons with ReLu as a function are connected to an activation layer with a sigmoid function mapping the weights to the probability between 0 and 1.

3.3.2. Single Frame Model

In single frame models we do not use a sequence to classify a video. Instead, we use just one single frame which turns the problem into an image classification problem which is a well understood deep learning domain.

For our classification problem we use CNNs. Instead of extracting the second last layer and using the generated features as in 3.2.1 we predict based on that model. The model is again trained only on the last two layers due to computational constraints. Further, we use the activation layer with sigmoid function for the prediction. This model is a very simple one since it uses only the information from one specific frame and ignores all the temporal information. To include the temporal information in the model this model was also tested with the last frame of the processed Optical Flow videos.

3.4. Postprocessing

The prediction of the model returns a score between 0 and 1. The convential way is to take a 0.5 threshold to predict either 0s or 1s. In our case we realize that it might be beneficial to adjust this threshold to other values. Further, we notice that ensembling approaches might be useful.

3.4.1. Score fusion and threshold adjustment

To combine the results of the two methods the mean of the two predicted results is calculated before being rounded. An analysis on the validation data revealed that often when the model is uncertain about the result, which results in a prediction of around 0.5, the actual event is not violent. To fix this problem the threshold of rounding is raised to 0.6. This results in both higher accuracy and higher recall with a sacrifice of lower precision. For a real life application this would be optimal regarding the importance of not missing any predictions of violent data compared to have some false negative.

3.5. Discarded Approaches

During the project several other approaches were thought of or implemented but had to be discarded due to various reasons. In the following chapter, we want to introduce some discarded approaches as they may fit other purposes.

3.5.1. Removal of Background noise

Firstly, when the videos are analysed an obvious characteristic is that there is a lot of noise in the background which might influence the prediction of the model significantly. The consequence is removing the noise. We discard this approach since the Optical Flow goes in a similar direction monitoring background noise as no relevant pixels to track. Through the selection of good pixels to track in the sparse Optical Flow and it only focuses on the main part of the video which makes it easier for the model to focus on the main actions instead of some background noise.

3.5.2. Person Detection

We implement a person detection to focus on one, i.e. the main person in the video. We hope that this focus will allow us to reflect the activities of this person in the data and to remove irrelevant information. Unfortunately, the videos are varying a lot in their quality and the position of the point of interest that it is not possible to clearly detect one person in every video. Nevertheless, we are confident that this focusing technique could work well, for example, for permanently installed CCTVs.

4. Results

All aforementioned approaches are implemented in a modular way such that the user can easily choose which method should be used. Based on that flexibility we evaluate the performance of the different methods. In this Section we present the evaluated performances.

4.1. Set-up of tests

The official benchmarks on the tests require a 5-fold cross-validation on the data set. To save time a single test with the test data containing one fifth of the data is conducted on the proposed models. Then the best model is chosen and tested with the official benchmark requirement.

4.2. Evaluation Methods

For performance evaluation we use accuracy, precision, recall and area-under-curve receiver-operating-characteristic (AUCROC). Probably the most unknown measure is the AUCROC. It basically plots the true positive rate against the false positive rate and takes the integral. The resulting value lies between 0 and 1 and gives an indication on how well the classifier performs with 0 interpreted as lack of performance and 1 as perfect classification.

4.3. Results

Table 8 shows the performances of the different models created in this paper.

Method	Accuracy	Precision	Recall	AUC [**]
SF [1*]	59.18	100	9.09	0.88
SOF [2*]	70.24	62.90	95.12	0.66
OptFlow [3]	78.37	76.92	81.30	0.84
LSTM1 [4]	86.12	83.46	90.24	0.94
LSTM2 [5]	86.94	88.24	85.37	0.93

[1] Single Frame
[2] Sparse Optical Flow
[3] CNN + Dense Optical Flow
[4] CNN + LSTM
[5] CNN + LSTM (Normal & Dense)
[*] no cross validation testing
[**] Plots can be found in Appendix A

Figure 8. Evaluations of the implemented models

The results show that more complex model were able to model the temporal information of the video data properly. The lack of performance of the single frame model is expected and clearly indicates the necessity for additional information in this case. One could argue that for a real life application the regular CNN + LSTM model is the best suitable model. Except for the precision and slightly the accuracy it outperforms the combined Normal and dense model in recall (how many of the violent videos where classified as violent) which would be really important in a real life application.

The Optical Flow models are unfortunately not performing as well as our hypothesis suggested. Both models only including Optical Flow images perform under 80 percent. Two reasons may cause this bad performance. One being a bad method for choosing labeled features. A high percentage of the videos contains text, irrelevant for the information in the videos, for example, score keeping in a football game or translation of a news videos. Our method for generating labeled features is not able to handle the text and many of the labeled features are text instead of humans in the video. Either a more advanced method or a solution for our proposed method would be needed. The other obvious improvement would be using a more advanced Optical Flow method. In this project a very simple model (Farnebäck, 2003) is used because of the lack of hardware. There exists a lot of proven better models (Ilg et al., 2016) which could be used for projects with internal GPUs.

4.4. Comparison to benchmark models with cross-validation test

Method	Accuracy	AUC
HOG [1]	57.43	0.94
HOF [1]	58.83	0.94
ViF [1]	81.30	0.85
LSTM1 [3]	86.12	0.94
LHOF+BoW [2]	86.57	0.90
LSTM2 [4]	86.94	0.93
LHOG+LHOF+BoW[2]	94.31	0.97

[1] (Hassner et al., 2012)
[2] (Zhou P, 2018)
[3] CNN + LSTM
[4] CNN + LSTM (Normal & Dense)

Figure 9. Results of implemented models versus benchmark tests

Figure 9 shows the comparison between the top models constructed in this paper, the original research on this data (Hassner et al., 2012) and one of the state-of-the-art models (Zhou P, 2018). The deep learning approach achieves a test accuracy in the middle of the (Hassner et al., 2012) model and the current best model. This shows that even though it performs much better than the state-of-the-art back in 2014 it still needs to be improved a lot to compare to the best models right now. Given the small data set, on which deep learning models regularly performs bad at and the Optical Flow limitations the score is still impressive and the model is viable for similar problems.

5. Conclusion

To conclude this paper we want to state the relevance of our work regarding the society and the future research in video classification. Further, we outline the limitations during the project which made it hard to further improve our results. Associated with our limitations we want to propose ideas for further research to extend the understanding and the possibilities of video classification.

5.1. Relevance

With limited resources we manage to build a working video classifier which has a high recall and allows the society to trust in such technology. Comparing to the handcrafted feature generating methods, our deep learning model can much easier be used and faster tuned for different tasks. In reference to future work we also state the possible improvements of the classifier.

5.2. Limitations

As the time for this project was limited by three weeks we were not able to implement everything we had in mind at the beginning of this topic. Besides the time constraint, we were highly constrained by our computational resources. Since we had no access to local GPUs, Google Colab was used for training and testing the models. This restricted us to only 30 GB of RAM, their GPU and limitations regarding the installation dockerfiles and github repos.

5.3. Future Work

Besides the discarded approaches in 3.5 there are several other options to improve the quality of the prediction. One possible extension could be the use of more sophisticated Optical Flow methods. The described methods for Optical Flow in 3.2.2 are basic in their implementation. For more sophisticated implementations more computational resources, i.e. the use of internal GPU, is required. Another possible way to improve the model would be a structured study of hyperparameters, we have used conservative set-ups for our models, so there is a lot of room for improvement. The extension of the data set can help to give the model the option to get a deeper understanding of the data.

In general, the video classification at all provides several interesting research fields, i.e. a better visualization of the generated features or the automated feature generation in general. (Le et al., 2011) proposes feature generation through unsupervised learning and achieves good performances and savings regarding the computational costs. This approach seems to be promising for this set-up as well since the videos differ a lot in the way they are recorded.

References

Farnebäck, G. Two-frame motion estimation based on polynomial expansion. 2003. doi: https://doi.org/10.1007/3-540-45103-X_50.

Hassner, T., Itcher, Y., and Kliper-Gross, O. Violent flows: Real-time detection of violent crowd behavior. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, June 2012. doi: 10.1109/cvprw.2012.6239348. URL https://doi.org/10.1109/cvprw.2012.6239348.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016.

Jianbo Shi and Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994. doi: 10.1109/CVPR.1994.323794.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR 2011*. IEEE, June 2011. doi: 10.1109/cvpr.2011.5995496. URL https://doi.org/10.1109/cvpr.2011.5995496.

Lucas, B. D. and Kanade, T. An iterative image registration technique with an application to stereo vision. In *In IJCAI81*, pp. 674–679, 1981.

Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. Beyond short snippets: Deep networks for video classification. pp. 4694–4702, 06 2015. doi: 10.1109/CVPR.2015.7299101.

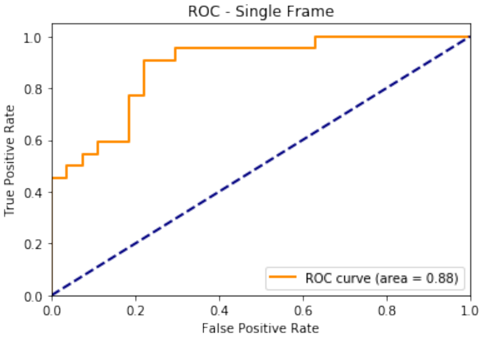
Schmidhuber, S. H. J. Long short-term memory. 1997. doi: 10.1162/neco.1997.9.8.1735.

Stavens, D. The opencv library: computing optical flow, 2007.

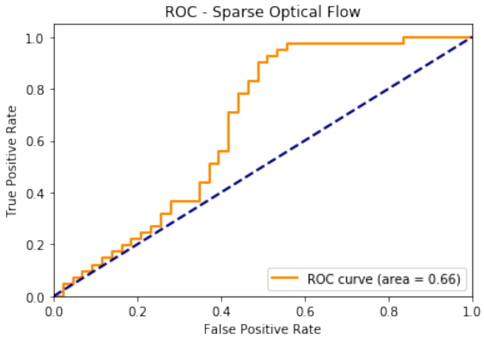
Zhou P, Ding Q, L. H. H. X. Violence detection in surveillance video using low-level features. 10 2018. doi: 10.1371/journal.pone.0203668.

A. AUCROC Plots

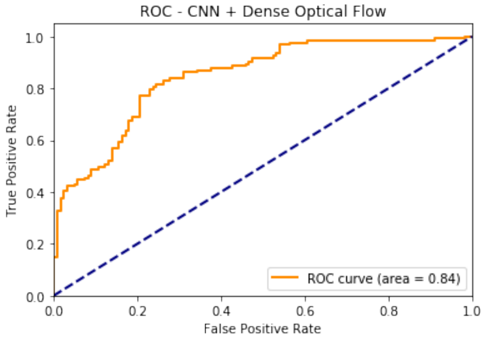
A.1. Single Frame



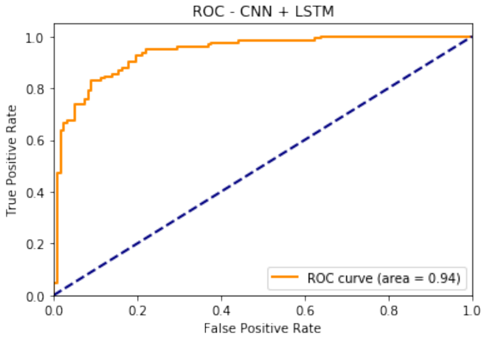
A.2. Sparse Optical Flow



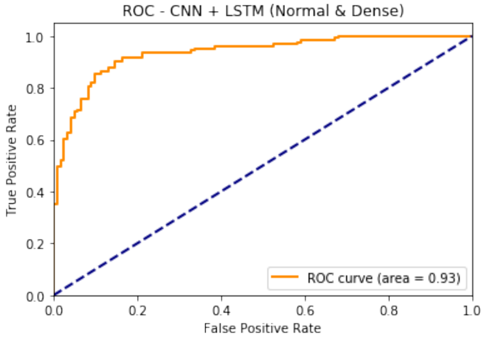
A.3. CNN + Dense Optical Flow



A.4. CNN + LSTM



A.5. CNN + LSTM (Normal Dense)


ROC - CNN + LSTM (Normal & Dense)
True Positive Rate
False Positive Rate
ROC curve (area = 0.93)