BULAT Mathilde
MARTYR Louis
NOUADJE Enge

ICPBL Project

Hanyang University - Fall Semester 2024 Computer Vision

Abstract

This project aims to develop a system for detecting violence in surveillance videos. It uses optical flow techniques, focusing on the Farneback algorithm, to track motion in video frames. Combined with a customized machine learning model, the system analyzes spatial and temporal features to identify patterns associated with violent behavior. The goal is to create a practical tool for real-time use in security settings, offering a reliable way to enhance public safety in monitored areas.

1. How does Farneback algorithm work ?

The Farneback algorithm is an optical flow estimation method that can be considered an extension of the Lucas-Kanade algorithm. Like the Lucas-Kanade approach, it computes the motion between two successive images by comparing the intensity of each pixel. This allows us to track objects in videos and estimate their direction and speed.

The basic principle is similar to the Lucas-Kanade algorithm : it relies on the use of a quadratic polynomial to approximate the intensity of an image in a neighbourhood, based on the gradient of the local pixel. The method represents the pixel data as a vector 2D coordinate, allowing comparison of two consecutive images at the same pixel using their respective quadratic models.

As a remainder, let a pixel I(x, y, t). Since the intensity of pixels do not vary, we can introduce the notion of offset, then estimate the motion thanks to Taylor development.

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x+y+t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + c$$

We obtain Vx and Vy that we want to resolve :

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0$$

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0$$

With Vx = dx/dt and Vy = dy/dt optical flow of I(x,y,t).

To resolve Vx and V by using Farneback algorithm, we have use the method of polynomial expansion where each neighbor of a pixel is approximated by a polynomial :

$$I(x, y) \approx x^\top A x + b^\top x + c$$

x is a vector of 1x2 containing x and y.
A is a symmetric matrix of unknowns 2x2 which captures the pair part of the signal
b is vector of 2x1 unknown which captures the unpair part of the signal
c is an unknown scalar
Since we have for 2 successives images :

$$I_1(x, y) \approx I_2(x + d_x, y + d_y)$$

with d = (dx, dy) the motion vector that we are looking for, we can transform the expression in a quadratic form to resolve d :

$$I_2(x + d) \approx (x - d)^\top A_2 (x - d) + b_2^\top (x - d) + c_2$$

By using the difference between both images :

$$\Delta I = I_1(x, y) - I_2(x + d)$$

We can say :

$$\Delta I \approx (A_1 - A_2) x^\top x + (b_1 - b_2)^\top x + c_1 - c_2 + 2d^\top A_2 x - d^\top A_2 d + b_2^\top d$$

To find the motion d, we just have to minimize this expression :

$$\nabla_\mathrm{d} \Delta I = 0$$

As many others algorithm, we use image pyramid to get several resolution and detect more precisely the movements (whether the speed is faster or slower).

The main difference with the Lucas-Kanade algorithm is that we use linear gradient approximations instead of quadratic polynomial approximation in the case of the Farneback algorithm.

SOURCE : https://espace.etsmtl.ca/id/eprint/2940/1/BELKACEMI_Imene.pdf Page 33
https://en.wikipedia.org/wiki/Optical_flow

N.B. Due to insufficient storage space on our drives, we couldn't store the whole dataset with the Farneback visualizations for each data. Therefore, we added the visualization function in the code, but we couldn't run it and had to skip it to be able to train our model.

2. Output video of Lucas-Kanade optical flow implementations & evaluation

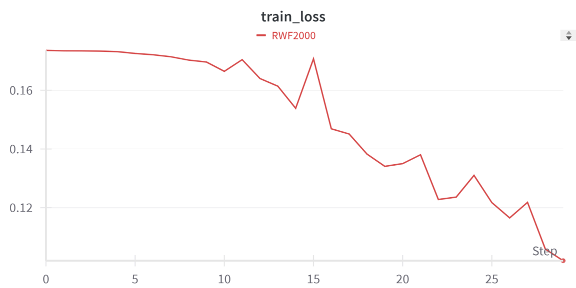
Link to the video: output.mp4

3. Output video of anomaly video detection

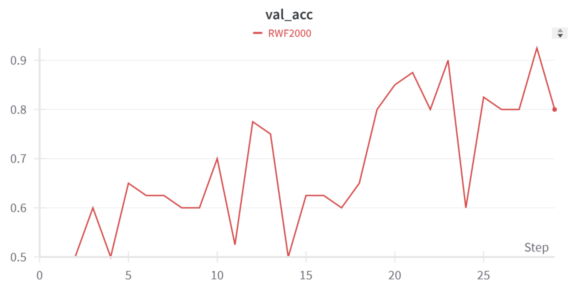
Link to the long video: video_detection_longvideo.mp4
Link to the short video: video_detection_shortvideo.mp4

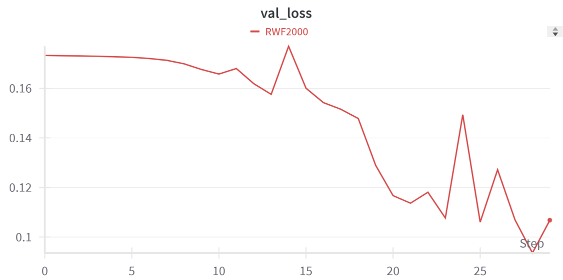
We were surprised by the results, as both of the visualizations ended up not working as expected even though our model had satisfying results after the training. Unfortunately, we couldn't figure out why this has happened.

4. Plots & analysis



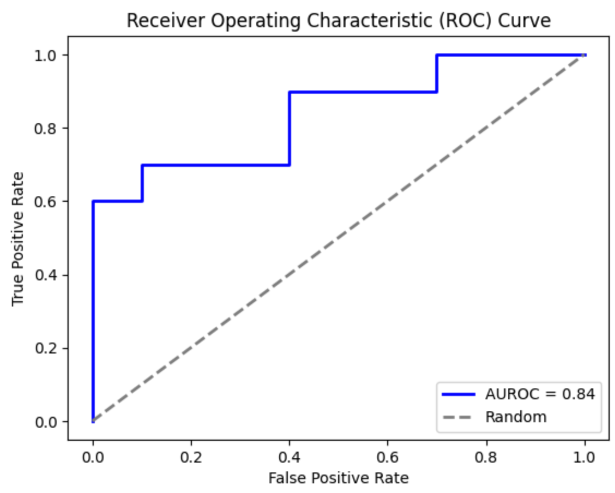
Based on these graphs we can see that our model is correctly learning and improving as the training accuracy keeps on increasing and the training loss keeps on decreasing over the epochs.





Again we can see that our model is learning and improving logically throughout the epochs, as the validation loss generally decreases, and the validation generally increases, with some variations and drops in accuracy that could be explained by some overfitting. However, with only 30 epochs the model could adapt itself to get back on track for generalization and did not fall into deep overfitting.

From these results, we can clearly see that the best model was achieved on epoch 28 with a validation loss near 0.09, and a validation accuracy of around 0.925.



The Receiver Operating Characteristic (ROC) curve for our fusion model highlights solid classification performance, achieving an AUROC score of 0.84. This score reflects the model's strong capability to differentiate between classes, performing significantly better than random guessing.

5. Does optical flow lead to better performance?

Optical flow is based on the differential approach to compute the differences between two successive images. The advantage of this approach is that it is relatively simple to implement, analyze, and understand due to its moderate computational complexity. For example, the Lucas-Kanade algorithm, which solves a system of linear equations, has a low computational cost because it computes motion locally over small neighborhoods. In contrast, the Farneback algorithm is more computationally intensive, as it involves quadratic polynomial interpolation to model pixel intensities.

More generally, optical flow is not limited to analyzing static images. It relies on temporal information to estimate the motion of objects between frames, providing insights into dynamic scenes. This is fundamentally different from CNN-based approaches like R-CNN, Fast R-CNN, and Faster R-CNN, which focus on detecting and classifying objects in single images. YOLO, on the other hand, detects objects in real time but does not fundamentally capture motion information. While the primary goal of optical flow is not to detect or classify objects, its ability to track motion allows us to identify specific behaviors or interactions between multiple objects. This is precisely what we aim to achieve in these scenarios involving dynamic scenes.

In terms of performance, optical flow leads to significant advantages in scenarios where motion is key, and that is why we use optical flow in the case of violence detection. While static methods treat each frame independently, optical flow captures the continuity of motion, reducing irregularities caused by rapid movement. Additionally, it works with object detection models by focusing on areas with motion, which makes the process faster and more accurate.

6. Does customized model lead to better performance?

The advantage of using a customized optical flow model is that it allows us to optimize it for specific use cases by adjusting key variables and parameters to better suit particular motion patterns or environmental conditions, making it more flexible.

In contrast, pre-built optical flow models come with fixed parameters, which makes them easier to use, especially when time and computational resources are limited. These pre-built models, such as those available in OpenCV (e.g., Farneback, Lucas-Kanade), are certainly optimized and efficient, which is an advantage for quick implementation and general use cases.

On the other hand, while customized models can provide better accuracy by allowing fine-tuned parameter adjustments, they come with higher implementation complexity. Developing a custom model demands additional time, expertise, and computational resources, particularly when training and testing (indeed, we experimented by requiring more GPU resources during model training !). Moreover, this increased complexity implies higher computational costs when compared to pre-built models.

Finally, the choice between a customized optical flow model and a pre-built one depends on the specific application. If general performance suffices, pre-built models are advantageous due to their simplicity and lower computational requirements. However, in situations that demand handling edge cases, unique motion patterns, or environmental variability, a customized model offers significant advantages in accuracy and adaptability.

7. Does it work well for different dataset?

After training our model, we achieved satisfying accuracy, meaning that the model seems to perform well on the training dataset and may generalize to other datasets with similar characteristics. However, we observed inconsistencies in predicting violent activity, where the model's performance was not entirely reliable.

This highlights a key limitation. While optical flow-based models can achieve good overall accuracy, their performance may vary depending on the dataset's specific features. For instance, the detection of violence often involves complex movements such as speed that may not be well captured by optical flow alone. This suggests that the model might require additional fine-tuning, preprocessing steps, or the integration of complementary methods, such as deep learning-based activity recognition (CNN), to improve its robustness across diverse datasets.