

AI Model for Detecting Abnormal Behaviour

System Architecture Design & Research Report

GROUP 1-C






Name	Position	Email	Phone No
Tran Quoc Dung	Software Development	dungtqsw00420@fpt.edu.vn	0943910306
Pham Hoang Duong	Data Research	duongphswh00843@fpt.edu.vn	0902162467
Nguyen Thai Son	Data Research	sonntsw00509@fpt.edu.vn	0902950403
Duong Quoc Trung	AI Researcher	trungdqswh00902@fpt.edu.vn	0962579876
Do Tuan Dat	Software Development	datdtswh00592@fpt.edu.vn	0865411803

COS40005, Computing Technology Project A, May-2024, 11/06/2024

DOCUMENT CHANGE CONTROL

Version	Date	Authors	Summary of Changes
1.00	01/07	All	Create an initial draft document.
1.10	16/07	Dũng, Đạt, Trung	Update the document with current project progress.
1.20	27/07	Dũng	Finalizing the document.

DOCUMENT SIGN OFF

Name	Position	Signature	Date
Tran Quoc Dung	Software Developer, Writer		16/07/2024
Pham Hoang Duong	Data Research, Software Dev		17/07/2024
Nguyen Thai Son	Data Researcher		17/07/2024
Duong Quoc Trung	AI Researcher		17/07/2024
Do Tuan Dat	Software Developer		17/07/2024

CLIENT SIGN OFF


Name	Position	Signature	Date
Le Van Khang	AI Engineer		27/07/2024
Organisation			
Quy Nhon AI Creative Alley			

Table of Contents

DOCUMENT CHANGE CONTROL	2
DOCUMENT SIGN OFF.....	2
CLIENT SIGN OFF	2
1. Introduction.....	4
1.1 Overview	4
1.2 Definitions, Acronyms and Abbreviations	4
2. Problem Analysis	5
2.1. System Goals and Objectives	5
2.2. Assumptions	5
2.3. Simplifications (if any).....	5
3. High-Level System Architecture and Alternatives.....	6
3.1. System Architecture	6
3.2. Other Alternative Architectures Explored	7
<i>Architecture 1</i>	7
<i>Architecture 2</i>	8
4. Research and Investigations.....	9
4.1. Research into Application Domain.....	9
4.2. Research into System Design	9
4.3. Other Research	9
a) The LSTM inner-working architecture.....	9
b) Training your model	10
5. References.....	11

1. Introduction

The software system being developed is an AI model designed to detect unusual behaviour in data sets. The system leverages advanced machine learning algorithms to analyse data and identify patterns that deviate from the norm, providing valuable insights for a variety of applications such as fraud detection, surveillance security and quality control.

The purpose of this report is to provide comprehensive guidance for AI model development and deployment. This document outlines the technical specifications, design considerations and research methods used in the project. It serves as a reference for project stakeholders, including team members, customers, and potential users, to understand project goals, progress, and outcomes

1.1 Overview

This report serves as a comprehensive guide to developing AI models designed to detect abnormal behaviour in data sets. The report provides detailed information about the project's context, objectives and methods. It covers the entire project lifecycle, from initial concept and design through to implementation, testing and final delivery.

1.2 Definitions, Acronyms and Abbreviations

- AI (Artificial Intelligence): Simulation of human intellectual processes using machines, especially computer systems.
- ML (Machine Learning): A subset of AI that involves the use of algorithms and statistical models to enable computers to perform specific tasks without explicit instructions.
- DMS (Data Management System): A software system designed to manage databases.
- SRS (Software Requirements Specification): A detailed description of the software system to be developed, including functional and non-functional requirements.
- PP (project plan): A document that outlines the project's goals, tasks, timelines, and milestones.
- SQAP (Software Quality Assurance Plan): Documents details of processes and standards to ensure software quality throughout the development process.

2. Problem Analysis

2.1. System Goals and Objectives

The primary goal of our system is to create an AI model capable of detecting abnormal behaviours. To achieve this, we aim to:

- The system should be able to analyse input data (video frames, dataset source) and identify instances of abnormal behaviour with high precision. (SRS: At least 80% accuracy)
- Real-time Detection: The system operates in real time, providing timely alerts when abnormal behaviour is detected. This is crucial for applications such as security surveillance, healthcare monitoring, and safety systems.
- Scalability: The solution should be scalable to handle large volumes of data and multiple concurrent streams. It should be able to process data from various sources efficiently.
- Minimal False Positives: While detecting abnormal behaviours, the system should minimize false positives to avoid unnecessary alarms or alerts. (SRS: Great stability)
- The system should provide an intuitive interface for users to interact with, configure, and monitor the AI model. This includes setting thresholds, adjusting sensitivity, and reviewing detected events (SRS: UI Development Focus)

2.2. Assumptions

In developing the system design, we consider these assumptions:

- Data Availability: Relevant data will be available for training and testing the AI model. In this project, input videos (data) will be converted into other formats with a specific number of frames (16 frames). The quality and diversity of this data will significantly impact the model's performance.
- Training Data: Labelled training data contains examples of both normal and abnormal behaviours. This annotated dataset will be crucial for training the model, making it able to discriminate these behaviours.
- Hardware Infrastructure: We assume access to suitable hardware (e.g., GPUs, CPUs) for training and deploying the AI model. The system's performance will depend on the computational resources available.
- Stable Environment: Without sudden changes in lighting, camera angles, or other external factors that could affect model performance during deployment.

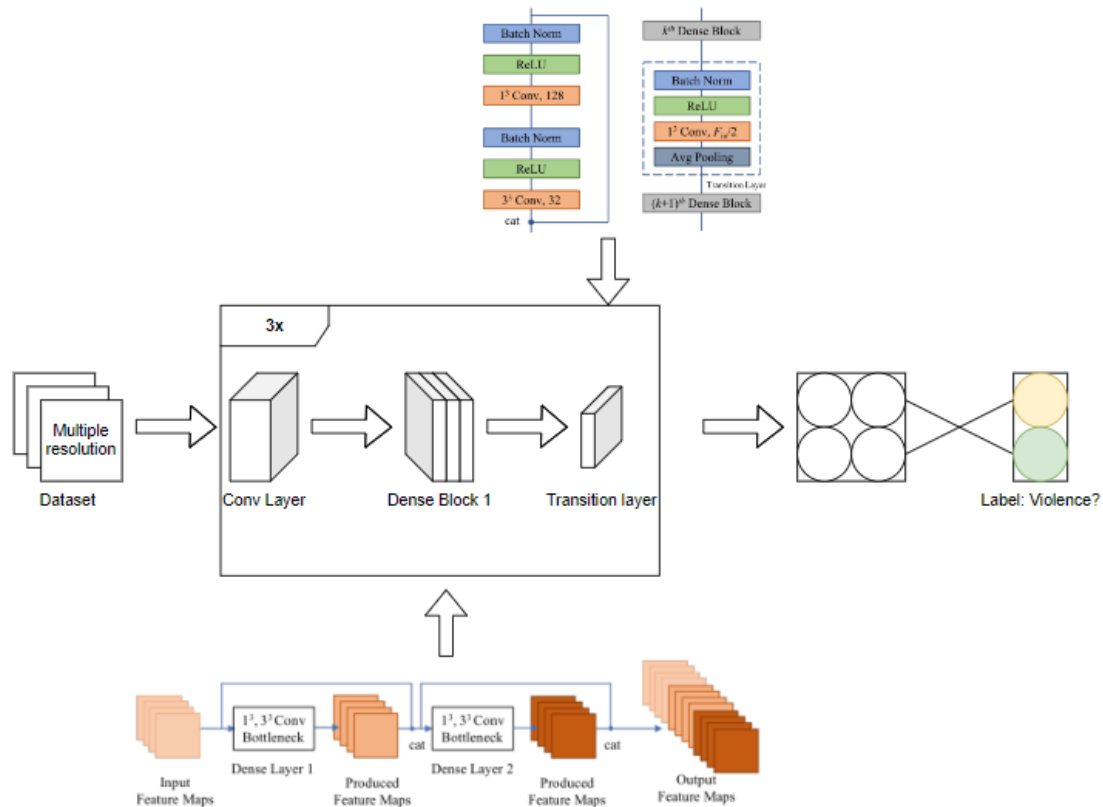
2.3. Simplifications (if any)

These aspects are surely simplified:

- Single Behaviour Type: Initially, we focus on detecting a specific type of abnormal behaviour, which are violent behaviours. Expanding to multiple behaviour types may be considered in future iterations.
- Binary Classification: We simplify the problem to binary classification (normal vs abnormal) rather than fine-grained categorization of abnormal behaviours. This simplification allows us to build a foundational model.
- Fixed Camera Perspective: We assume fixed camera positions and consistent perspectives for simplicity. Handling dynamic camera angles and movements would require additional complexity.

3. High-Level System Architecture and Alternatives

3.1. System Architecture



The finalized version of the architecture

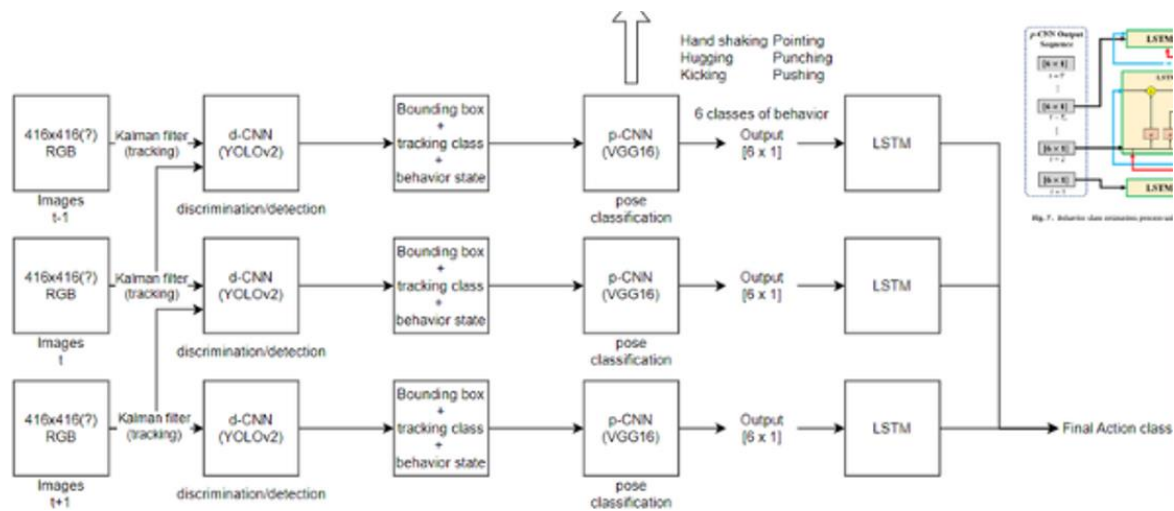
This is the 3D-Convolutional Neural Network structure by Xinghao Jiang et al. The model consists of these main elements:

- The dataset can be any form of videos. The pre-processing phase will resize the data before working on the later steps
- The initial Convolution Layer produces intermediate feature maps from the frames fed from the dataset
- The Dense block is a 3x3x3 Convolutional layer that produces the output feature maps.
- The transition layers between the 3 Dense blocks are used to simplify the feature extractions by down sampling them and match the number of output and input feature maps between adjacent blocks
- The Global Average Pooling layer provides the field of features before deciding the final classification of the video

This architecture provides some significant strength compared to others, such as:

- This entire architecture only uses 1 model (3D-CNN). This makes the workflow easy to follow, faster inference speed with a moderate amount of computing power
- With some optimization, this system can run at a near-real time speed, which means a possible way to do online detection
- The entire system is lightweight, so it could be implemented on AI boards such as the Nvidia Jetson series

3.2. Other Alternative Architectures Explored



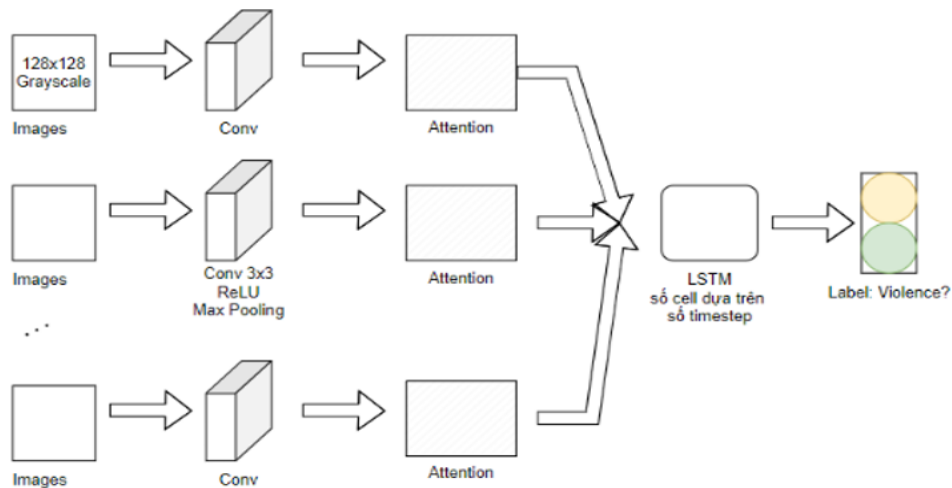
Architecture 1

This is the Deep Learning convolutional structure design from Ko and Sim⁽¹⁾, the model is understood as below:

- d-CNN: d-CNN read the image data containing people doing actions (coloured) and separate the images of people from the environment using YOLOv2 to know the object and select it within a bounding box
- p-CNN: p-CNN has a predefined number of abnormal poses due to VGG16. Once it received the cropped-out image of objects (people involved), it returns a one-hot vector as output that represents the likelihood of abnormal poses defined above
- LSTM: LSTM receives the vector and analyses it within a sequence of time (chain of vectors representing sequence of actions) and will recognize certain patterns if repeated or occur enough and flag it as abnormal, this will be memorized as each time a subsequent sequence analyses to helps identify the behaviour better

Disadvantage:

- Using multiple sequence like so can cause latency issues when ran in a long period of time
- Images and videos are compressed in a more intensive format (coloured) might affect performance of the system
- LSTM is more memory intensive



Architecture 2

This is the CNN-LSTM model that use attention mechanism ⁽²⁾ that works like so:

- The input video is processed into many individual frames and scale down (with grey colour) at the first layer that is the input layer
- Each image is passed to the convolution layers (2) that have filters to create a feature map of the object (person involved) and each layer has more filters that support deeper learning for the model. The feature maps capture the most important feature and is then applied with ReLU and Max Pooling in a separate level that respectively helps to remove negative values of elements from the map, so the image isn't too saturated which affect the learning of model and to decrease the size of the map to optimize and reduce the number of unimportant data
- The attention mechanism is applied that will make the output from the convolution layers, which is a feature map, to be even more emphasized on. Depending on the type of detection the model needs to do, the attention will focus more on that specific feature – like hand/leg movement in violence detection. This is done using weight
- The last layer is LSTM which contains cells that help for the learning process of the model by recognizing patterns and trends from data. At this level, all the features extracted from the layer before are evaluated in a sequence over a period of time and will decide whether or not the behaviour is violence. Using this knowledge, when an action that is similar happens, it will recognize it as violence.

Disadvantages:

- More complexity in sequential modelling with both CNN and LSTM being used
- Performance is heavier due to the especially in CPU and RAM usage due to LSTM

4. Research and Investigations

4.1. Research into Application Domain

In various environments, the supervision of activities is required to detect the appearance of anomalous behaviours/appearances that could pose operational and safety threats to many entities. This adds a crucial layer of security into the operations of numeric businesses, public safety administration, and industrial operations.

Amongst these environments, a common pattern exists where there are only a small number of occurrences of anomalous behaviours and appearances compared to the humongous volume of normal behaviours and appearances. This fact makes it difficult to capture the notable features of anomalies. Hence, it is a much more efficient and effective approach to capture the significant features of normal data, instead, and use those captured characteristics to determine whether a new piece of data is anomalous or not.

4.2. Research into System Design

Our objective for this project is to develop a system that can detect anomalous behaviours/appearances, given unlabelled data of videos of normal entities and activities.

The first task to accomplish this objective is to segment our videos into a fixed number of frames every second to get the most meaningful spatial representation of movements in the videos.

Then, we need to extract the most valuable features from those images (frames) as the data for the system to analyse. Convolutional 3d layers and max pooling layers will be used for this task.

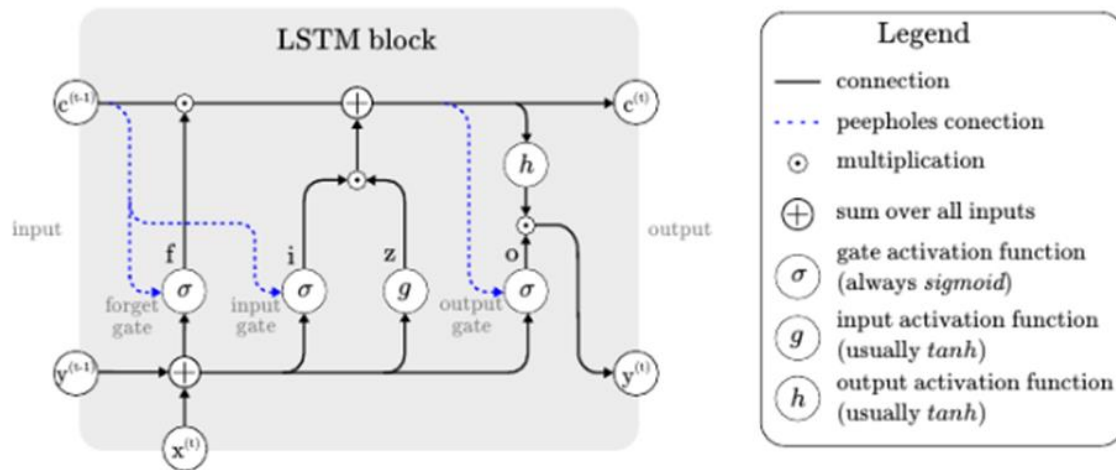
To address this objective, especially with data that are not labelled, different deep learning models have been considered (for instance, isolation forest, which detects outliers by how easily they could be separated from a group of data). Amongst these models, One-Class Support Vector Machine has been selected, as it works by encompassing features extracted video frames to determine a boundary that will be used to rule out anomalous data, which fits the most to our objective.

4.3. Other Research

A review on the Long Short-Term memory model⁽³⁾: The research given helps to understand more about the basic architectural functioning of a vanilla LSTM model and how to apply it for training

a) The LSTM inner-working architecture

The basic architecture of an LSTM comprises 4 main elements: a cell, input gate, output gate and forget gate. These work together in cohesion with the cell capable of memorizing information and its memory being regulated by the different gates. Here's how it works



A block of memory within a time-step

- Block input: This combines the last iteration output $y^{(t-1)}$ and the current input $x^{(t)}$
- Input gate: This combines the last iteration output $y^{(t-1)}$ and the current input $x^{(t)}$ with the last cell value $c^{(t-1)}$ helps to control information for the block input to determine which information to keep from the last cell state
- Forget gate: This combines the last iteration output $y^{(t-1)}$ and the current input $x^{(t)}$ with the last cell value $c^{(t-1)}$ to determine which information to remove from the last cell state
- Cell: This combines last cell value $c^{(t-1)}$ and the input gate $i^{(t)}$ and the block input to determine which value to memorize for the next iteration
- Output gate: This combines the last iteration output $y^{(t-1)}$ and the current input $x^{(t)}$ with the last cell value $c^{(t-1)}$
- Block output: This combines the current cell value $c^{(t)}$ and the output gate $o^{(t)}$

b) Training your model

- Using the “Backpropagation Through Time” technique, the current cell value $c^{(t)}$ is compared with the next cell value $c^{(t+1)}$ and the current output $y^{(t)}$ using gradients. The gradients help to adjust the weights for each component of a memory block that is generated when an output receives an error.
- Now the block will be updated to change its weights parameters of the cell and gates by adjusting them proportionally based on the gradients generated before. This will help reduce the error and is corresponding to the learning of the model. This process goes on for multiple time before everything is optimize for the best result

5. References

- (1) Ko, K., Sim, K., 2018, 'Deep convolutional framework for abnormal behavior detection in a smart surveillance system', Engineering Applications of Artificial Intelligence, vol. 67, pp. 226 - 234, viewed 6 July 2024, <https://www.sciencedirect.com/science/article/pii/S0952197617302579>
- (2) Tay, N., Tee, C., Ong, T., Teh, P., 2019, Abnormal Behaviour Recognition using CNN-LSTM with Attention Mechanism, IEEE, viewed 5 July 2024, <https://ieeexplore.ieee.org/document/8974824>
- (3) Houdt, G., Mosquera, C., Napoles, G., 2020, A Review on the Long Short-Term Memory Model, ResearchGate, viewed 9 July 2024, https://www.researchgate.net/publication/340493274_A_Review_on_the_Long_Short-Term_Memory_Model