UiA University of Agder

Master

Deep Learning for Crowd Anomaly Detection

The impact of deep optical flow methods as a hand-crafted feature

KENNETH KROGSTAD AASTVEIT

SUPERVISOR
Christian Walter Peter Omlin

University of Agder, 2022
Faculty of Engineering and Science
Department of Engineering and Sciences

Obligatorisk gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Vi erklærer herved at vår besvarelse er vårt eget arbeid, og at vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	Ja
2.	**Vi erklærer videre at denne besvarelsen:** • Ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • Ikke refererer til andres arbeid uten at det er oppgitt. • Ikke refererer til eget tidligere arbeid uten at det er oppgitt. • Har alle referansene oppgitt i litteraturlisten. • Ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	Ja
3.	Vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	Ja
4.	Vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	Ja
5.	Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	Ja
6.	Vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	Ja
7.	Vi har i flertall blitt enige om at innsatsen innad i gruppen er merkbart forskjellig og ønsker dermed å vurderes individuelt. Ordinært vurderes alle deltakere i prosjektet samlet.	Nei

Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).
Oppgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	Ja
Er oppgaven båndlagt (konfidensiell)?	Nei
Er oppgaven unntatt offentlighet?	Nei

Abstract

Today, public areas across the globe are monitored by an increasing amount of surveillance cameras. This widespread usage has presented an ever-growing volume of data that cannot realistically be examined in real-time. Therefore, efforts to understand crowd dynamics have brought light to automatic systems for the detection of anomalies in crowds. This thesis explores the methods used across literature for this purpose, with a focus on those fusing dense optical flow in a feature extraction stage to the crowd anomaly detection problem. To this extent, five different deep learning architectures are trained using optical flow maps estimated by three deep learning-based techniques. More specifically, a 2D convolutional network, a 3D convolutional network, and LSTM-based convolutional recurrent network, a pre-trained variant of the latter, and a ConvLSTM-based autoencoder is trained using both regular frames and optical flow maps estimated by LiteFlowNet3, RAFT, and GMA on the UCSD Pedestrian 1 dataset. The experimental results have shown that while prone to overfitting, the use of optical flow maps may improve the performance of supervised spatio-temporal architectures.

Contents

Abstract ii

List of Figures vi

List of Tables vii

1 Introduction 1

2 Theoretical Background 3
- 2.1 Artificial Neural Networks . . . 3
  - 2.1.1 The Perceptron . . . 3
  - 2.1.2 Deep Neural Networks . . . 4
  - 2.1.3 Convolutional Neural Networks . . . 5
  - 2.1.4 3D Convolutional Neural Networks . . . 7
  - 2.1.5 Recurrent Neural Networks . . . 8
  - 2.1.6 Autoencoders . . . 9
- 2.2 Optical Flow . . . 10
  - 2.2.1 Traditional Methods . . . 10
  - 2.2.2 Feature Based Methods . . . 13
  - 2.2.3 Deep Learning Based Methods . . . 13
- 2.3 Crowd Anomaly Detection . . . 14
  - 2.3.1 Crowd Features . . . 14
  - 2.3.2 Traditional Methods . . . 15
  - 2.3.3 Deep Learning Based Methods . . . 15

3 State of the Art 17
- 3.1 Deep Learning for Optical Flow . . . 17
  - 3.1.1 FlowNet . . . 17
  - 3.1.2 Recurrent All-Pairs Field Transforms . . . 19
  - 3.1.3 Global Motion Aggregation . . . 20
- 3.2 Datasets for Optical Flow . . . 20
  - 3.2.1 KITTI . . . 20
  - 3.2.2 MPI-Sintel . . . 21
  - 3.2.3 Flying Chairs . . . 21
- 3.3 Deep Learning for Crowd Anomaly Detection . . . 22
  - 3.3.1 Methods Fusing Optical Flow . . . 22
  - 3.3.2 Performance Metrics . . . 22
- 3.4 Datasets for Crowd Anomaly Detection . . . 23
  - 3.4.1 UCSD . . . 24
  - 3.4.2 CUHK Avenue . . . 24
  - 3.4.3 ShanghaiTech Campus . . . 24
  - 3.4.4 Subway . . . 25
  - 3.4.5 UCF-Crime . . . 25

4 Method **26**
4.1 PyTorch . . . 26
4.2 Dataset . . . 26
4.3 Convolutional Neural Networks . . . 26
4.3.1 2D CNN . . . 27
4.3.2 3D CNN . . . 28
4.4 Convolutional Recurrent Neural Network . . . 29
4.5 Autoencoder . . . 30
4.6 Data Preprocessing . . . 31
4.6.1 Optical Flow Estimation . . . 31
4.6.2 PyTorch Datasets & DataLoaders . . . 33
4.6.3 Data Augmentation . . . 33

5 Experiments & Results **35**
5.1 2D CNN . . . 35
5.2 3D CNN . . . 36
5.3 CRNN . . . 37
5.4 CRNN-ResNet152 . . . 38
5.5 Autoencoder . . . 39
5.6 Summary . . . 40

6 Discussion and Future Work **41**

7 Conclusion **42**

A 2D CNN Results **43**

B 3D CNN Results **45**

C CRNN Results **47**

D CRNN-ResNet152 Results **49**

E Autoencoder Results **51**

Bibliography **54**

List of Figures

2.1 McCulloch-Pitts computational model of a neuron . . . 4
2.2 A deep neural network in the shape of a multi-layer perceptron . . . 4
2.3 Schematic diagram illustrating the interconnections between layers in the neocognitron [35] . . . 5
2.4 The architecture of a convolutional neural network including multiple convolutional and pooling layers [38] . . . 6
2.5 The sliding window technique of a kernel applied to an input in order to produce a feature map . . . 6
2.6 Max pool operation of size 2×2 and stride 2×2 . . . 7
2.7 Comparison of 2D and 3D convolution operations . . . 7
2.8 Representation of a recurrent neural network both folded and unfolded . . . 8
2.9 Comparison of RNN cell (left) and LSTM cell (right) . . . 9
2.10 A general representation of an autoencoder architecture . . . 10
2.11 The optical flow problem [80] . . . 11
2.12 Sparse vs. dense optical flow [92] . . . 12
2.13 Optical flow representation [93] . . . 12
2.14 The four main stages of the crowd behavior analysis pipeline [6] . . . 15

3.1 The two network architectures: FlowNetSimple (top) and FlowNetCorr (bottom) [115] . . . 18
3.2 The network structure of LiteFlowNet [217] . . . 18
3.3 Simplified network architecture of LiteFlowNet3 [220] . . . 19
3.4 The network structure of RAFT [217] . . . 19
3.5 The self-contained GMA module added to the RAFT architecture [228] . . . 20
3.6 Frame and the corresponding ground truth from the KITTI-2015 dataset [233] 21
3.7 Frame and the corresponding ground truth from the MPI-Sintel dataset [235] 21
3.8 Frames and their corresponding ground truths from the Flying Chairs dataset [115] . . . 21
3.9 Anomalous examples from the UCSD datasets [246] . . . 24
3.10 Anomalous example from the CUHK Avenue dataset (wrong direction) [159] 24
3.11 Anomalous example from the ShanghaiTech Campus dataset . . . 25
3.12 Anomalous examples from the Subway dataset . . . 25
3.13 Anomalous examples from the UCF-Crime dataset [1] . . . 25

4.1 2D CNN architecture for crowd anomaly detection . . . 27
4.2 3D CNN architecture for crowd anomaly detection . . . 28
4.3 CRNN encoder-decoder overview . . . 29
4.4 CRNN architecture for crowd anomaly detection . . . 30
4.5 AE architecture for crowd anomaly detection . . . 31
4.6 An example anomalous frame from the UCSD dataset . . . 32
4.7 Optical flow pre-trained on the Flying Chairs dataset . . . 32
4.8 Optical flow pre-trained on the KITTI dataset . . . 32
4.9 Optical flow pre-trained on the MPI-Sintel dataset . . . 32

4.10 Visualized optical flow map of insignificant motion at different normalization values 32

A.1 2D CNN results using regular frames 43
A.2 2D CNN results using LiteFlowNet3 frames 43
A.3 2D CNN results using RAFT frames 44
A.4 2D CNN results using GMA frames 44

B.1 3D CNN results using regular frames 45
B.2 3D CNN results using LiteFlowNet3 frames 45
B.3 3D CNN results using RAFT frames 46
B.4 3D CNN results using GMA frames 46

C.1 CRNN results using regular frames 47
C.2 CRNN results using LiteFlowNet3 frames 47
C.3 CRNN results using RAFT frames 48
C.4 CRNN results using GMA frames 48

D.1 CRNN-ResNet152 results using regular frames 49
D.2 CRNN-ResNet152 results using LiteFlowNet3 frames 49
D.3 CRNN-ResNet152 results using RAFT frames 50
D.4 CRNN-ResNet152 results using GMA frames 50

E.1 Autoencoder results using regular frames 51
E.2 Autoencoder results using LiteFlowNet3 frames 52
E.3 Autoencoder results using RAFT frames 52
E.4 Autoencoder results using GMA frames 53

List of Tables

5.1 Summarization of the 2D CNN results on the test set 36
5.2 Summarization of the 2D CNN results on the validation set 36
5.3 Summarization of the 3D CNN results on the test set 37
5.4 Summarization of the 3D CNN results on the validation set 37
5.5 Summarization of the CRNN results on the test set 38
5.6 Summarization of the CRNN results on the validation set 38
5.7 Summarization of the CRNN-ResNet152 results on the test set 38
5.8 Summarization of the CRNN-ResNet152 results on the validation set 38
5.9 Summarization of the AE results on the test set 39
5.10 Summarization of the AE results on the validation set 39
5.11 Summarization of each model's performance on the test set 40
5.12 Summarization of each model's performance on the validation set 40

Chapter 1

Introduction

Surveillance cameras are increasingly being used in public spaces across the globe for the purpose of preserving public safety and social order. By the end of 2021, the number of closed-circuit television (CCTV) surveillance cameras were estimated to grow beyond one billion globally [1], with an ever-increasing demand expected as smart city infrastructure and artificial intelligence (AI) based capabilities expand [2]. In this context, the monitoring capability of law enforcement agencies are unable to keep up with the growing volume of data, thus facing a pressing need for the development of intelligent and automatic surveillance systems in the reality of growing threats to social security. To this extent, areas such as object detection [3] or facial recognition [4] have been broadly studied in the last decades.

Subject of active research from different disciplines, especially within social studies and computer vision, crowd behavior analysis is increasingly attracting attention as a way to understand underlying crowd mechanisms that could be paramount for the safety of public spaces [5]. With an increase of population and diversity of human activities, this area aims to extract meaningful information of how individuals behave when they are part of a larger group. However, there is a lack of consensus on what crowd behavior analysis constitutes, due to the ambiguous nature of crowded scenes. For example, a group of ten individuals may be considered a crowd in a park, but hardly so in other environments such as the Shibuya Crossing in Japan. This issue is exacerbated when tackling sub-topics of the field; recent studies have attempted to adopt a common taxonomy with crowd anomaly detection at the forefront of the state-of-the-art [5, 6]. Notably, this topic aims to identify changing crowd behaviors in a scene that differ from the conventional or expected behavior, based on a selection of predictive crowd features that determine the relation between individuals and the crowd they belong to [7, 8]. In this case, anomalous behavior is intrinsically ambiguous and strongly dependent on the norms defined in the considered environment [9, 8]. For instance, a crowded scene of running individuals may be considered normal in one environment such as a marathon, contrary to an environment such as a pilgrimage site. Moreover, the wide range of changing circumstances between environments, such as illumination, angles, occlusion, seasons, and weather, bring forth additional complexity. As a result, detecting anomalous—or abnormal—events in real-world CCTV footage pose an important, yet challenging task, particularly considering their spatio-temporal nature.

An emerging trend for crowd behavior analysis is the use of deep learning-based models, which recently have achieved remarkable advances in computer vision tasks. These models are suitable for automatically analyzing video sources such as those from CCTVs, and thereby have seen widespread use for the monitoring of crowded scenes [4]. In turn, numerous deep learning methods have been employed in an attempt to solve the challenges of the crowd anomaly detection problem. In general, these challenges are two-fold: the lack of specificity, and insufficiency of data. In terms of supervised learning models requiring labeled data, human activities and interactions are difficult to represent and abnormal events may

be difficult to obtain or simply not have happened yet in a considered environment. Furthermore, this approach lacks practicality due to the time consumption required to manually label video footage. As a consequence, these models may be inadequate in understanding the nature of an anomaly, and thus the task is often considered an unsupervised rather than a supervised learning problem. To this end, datasets are commonly separated between training videos that contain only normal activities, and testing videos that consist of both normal and abnormal activities. An unsupervised model will be able to catch the normality existing in the training dataset, and during testing provide a prediction of whether any activity that deviates from the learned normality is an anomaly. In essence, the crowd anomaly detection problem can be seen as a binary classification task—each frame is predicted to either belong to a normal class or an abnormal class. Nevertheless, supervised deep learning-based methods have seen increased usage as of late due to the richness of features Convolutional Neural Networks (CNN) are able to extract [10, 11].

The extraction of proper features play a vital role in how crowd anomaly detection methods are able to capture the wide variety of anomalies present in crowded scenes [12]. While CNNs are able to automatically extract key features without human effort, hand-crafted feature extraction plays a larger role in the effectiveness of unsupervised detection methods [9, 13]. A common metric is the estimation of motion patterns that capture spatial and temporal behaviors within a crowded scene, such as optical flow [14], histogram of oriented gradients [15], and trajectory-based methods [16]. With the advancement of deep learning, optical flow has emerged as a powerful technique to model spatiotemporal correspondence in videos, and has been wildly applied in various computer vision tasks such as autonomous driving, action recognition, scene understanding, and robotics [17]. In short, these methods aim to represent the pixel motion between two consecutive frames in terms of velocity and direction. Subsequently, some studies have applied this in the feature extraction stage of deep learning methods for the crowd anomaly detection problem, with impressive results [9, 18, 19]. Based on these studies, this thesis aims to evaluate how different optical flow methods may improve the classification of both unsupervised and supervised deep learning models for the crowd anomaly detection problem, without the need for extensive network modifications.

Furthermore, there is a pressing need for fair and interpretable deep crowd anomaly detection methods [20, 21]. By and large, most studies have mainly focused on the detection accuracy aspect without concern for preserving the privacy of individuals. From the perspective of both research ethics and policing, the mitigation of any potential bias such as gender or race, or risk toward identifying features is of paramount importance [20, 22]. Accordingly, some studies have explored the anomaly explanation issues [23, 24] and data anonymization [25, 26, 27].

The rest of this thesis is organized as follows: Chapter 2 provides preliminary background theory and introduces the fundamental elements of this thesis. Chapter 3 reviews related work in terms of optical flow estimation and the crowd anomaly detection methods that incorporate this in the feature extraction stage. Chapter 4 describes the methodical approach taken to attain the goals of this thesis, including discussions of the data and different model architectures. Chapter 5 provides details of the experiments conducted and the evaluation of each approach. Chapter 6 discusses both remaining challenges and the potential for future work. Finally, Chapter 7 concludes the paper.

Chapter 2

Theoretical Background

This chapter aims to provide the fundamental theory of elements that lay the foundation for the topics and work covered in following chapters. First, an introduction to artificial neural networks is outlined from their conception to the multiple architectures used in this thesis. Second, the essence of optical flow estimation is described and looks at the evolution of both traditional and deep learning-based methods. Lastly, the various features and methods used in literature across the last decade for the crowd anomaly detection problem is summarized.

2.1 Artificial Neural Networks

Artificial neural networks (ANN), or simply neural networks (NN), reflect the behavior of the human brain by loosely modeling the interconnectedness between biological neurons. These artificial neurons aim to mimic how biological neurons signal one another based on how strong their connection is, and allow computing systems to learn complex tasks without human intervention. As such, ANNs have seen widespread usage in a wide variety of disciplines as computational power has improved, such as object recognition, face identification, autonomous vehicle technology, cancer detection, and more. This section follows the development of ANNs from their conception to the introduction of architectures utilized in the methodical approach of this thesis.

2.1.1 The Perceptron

With the aim of creating a model that would mimic the functionality of a biological neuron, McCulloch and Pitts proposed in 1943 the first computational model of a neuron [28]. By means of propositional logic, given a set of Boolean inputs $x_1 \dots x_n$, and weights $w_1 \dots w_n$ representing a connections strength, each neuron forms a weighted sum z of its inputs and passes it through an activation function deciding if the neuron should fire or not. Forming the Threshold Logic Unit (TLU), or the McCulloch-Pitts model as shown in Figure 2.1, a threshold θ was proposed as the activation to determine the output $y = f(z)$ as shown in Equation 2.1. Yet, the TLU could not explicitly *learn* weights, and thus could not emulate how the biological brain learns over time. It was not before over a decade later an extension of this model was proposed by Rosenblatt in 1958 called the perceptron [29]. His key contribution was the introduction of a learning rule, able to modify the weights of each input as described by [30, 31]. Essentially, the perceptron is a supervised, binary classifier. Meaning it can be provided with a set of examples, i.e. a training set, and separate two distinct classes by a linear decision boundary learned from comparing the output to the desired target result, and using the learning rule to adjust the weights in order to move the outputs closer to the targets. Thereby, the perceptron can be considered the foundational building block of neural network architectures, and a single-layered network as it contains only one computational layer.

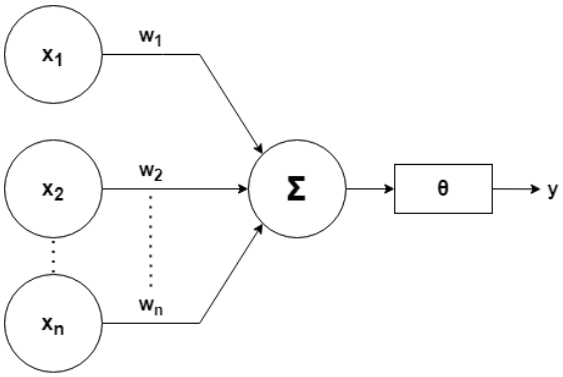


Figure 2.1: McCulloch-Pitts computational model of a neuron

$$f(z) = \begin{cases} 1 & \text{if } z > \theta \\ 0 & \text{if } z \leq \theta \end{cases} \tag{2.1}$$

Equation 2.1 illustrates the threshold function. Given a threshold of $\theta = 2$, the artificial neuron would output $y = 1$ if the weighted sum passed through the activation function is more than 2, otherwise $y = 0$.

2.1.2 Deep Neural Networks

In the pursuit of a more general framework to understand cognition, a pioneering neurocomputing group published a compendium called Parallel Distributed Processing (PDP) in 1986 [32]. In an interest to to train a neural network with multiple layers in a nonlinear fashion in contrast to the perceptron, the idea of learning internal representations by backpropagation was proposed by Rumelhart et al. [32]. By introducing hidden layers of neurons stacked together (i.e. fully-connected, or dense), it would come to be known as a Multi-Layer Perceptron (MLP), or a feed-forward network, as each output of a layer would be propagated to the next layer as seen in Figure 2.2. Divided into two parts, the forward propagation phase would feed each layer to the next and compute predictions and a measure of how well the network performed. In their work, [32] measured the sum of squared errors, or the Mean Squared Error (MSE). The backpropagation phase would propagate the error backward and compute the gradient of the MSE with respect to the weight of each input and output pair, and make adjustments to weights accordingly to reduce the error. Often used interchangeably in literature, the MLP can be considered the foundational architecture of deep neural networks (DNN) [33]. The amount of hidden layers required to be considered deep is not clearly defined [34], but the general consensus among researchers is two.

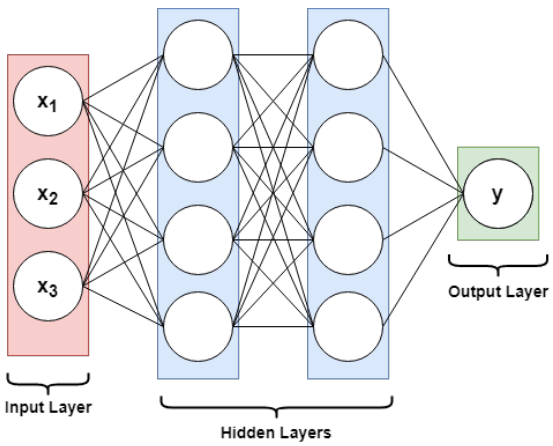


Figure 2.2: A deep neural network in the shape of a multi-layer perceptron

2.1.3 Convolutional Neural Networks

The neocognitron, an NN proposed by Fukushima in 1980 [35], was inspired by studies of the visual cortex of mammals and acquired the ability to recognize visual patterns irrespective of translations or local distortions of the input through learning. This multi-layered network introduced the concept of hierarchical layers connecting each neuron only to the neurons of a small patch in the previous layer, or in the context of the visual cortex, their receptive fields. By gradually reducing the spatial dimension of deeper layers through subsampling layers, this architecture allows the network to concentrate on extracting local features of the input space in shallow layers (e.g. lines, curves, etc.), and more global features (e.g. faces, objects, etc.) in deeper layers as each respective receptive field expands [34, 36]. As a result, the final layer of the neocognitron indirectly has a receptive field covering the entire input space as illustrated by Fukushima in Figure 2.3. The benefit of this approach was that it avoided pitfalls discovered in early DNNs—while in principle able to identify patterns with respect to variation, such a network would lead to multiple neurons requiring similar weight patterns positioned at various locations in the input, so as to detect distinctive patterns wherever they appear on the input [37]. This would lead to an inefficient network architecture requiring thousands, if not millions of parameters, and would require larger datasets with samples of every possible variation. Over time, it was the combination of layers to extract features and subsampling layers to reduce spatial resolution that gradually evolved into what we now call convolutional neural networks (CNN) [37].

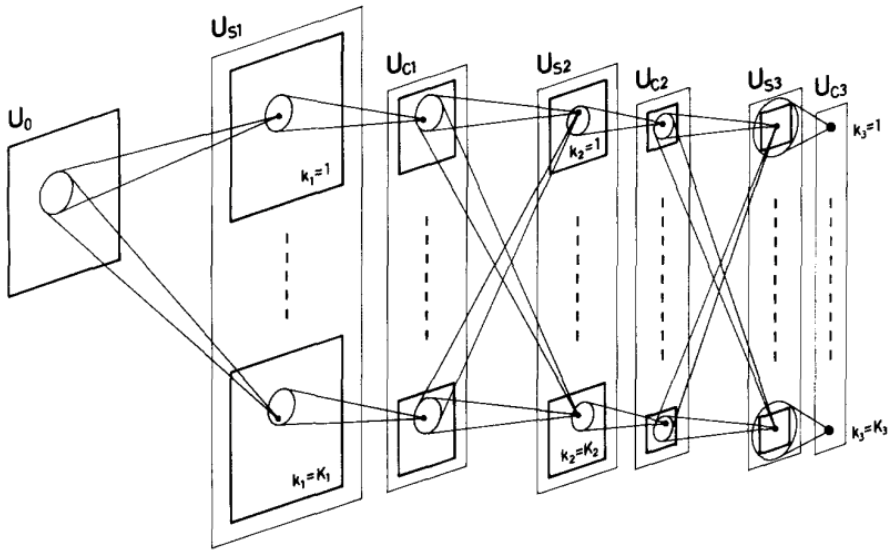


Figure 2.3: Schematic diagram illustrating the interconnections between layers in the neocognitron [35]

Widely used in the field of computer vision, the CNNs of today share the same basic architecture to that of the neocognitron, and are typically fed a three-dimensional matrix of pixels $w \times h \times c$ as input. Comprised of one or more convolutional layers followed by pooling layers (i.e. subsampling) in-between, subsequent fully-connected layers perform classification as in a feed-forward neural network as illustrated in Figure 2.4. In order to learn spatial patterns, convolutional layers consist of learnable sets of feature extraction filters, or kernels. Often used interchangeably in literature, kernels are defined as a two-dimensional matrix of weights with size $m \times n$ applied to an input in a sliding fashion, whereas filters are a three-dimensional $m \times n \times c$ structure of multiple kernels stacked together. Each channel of a filter compose a kernel that will be convolved with the respective channel of an image.

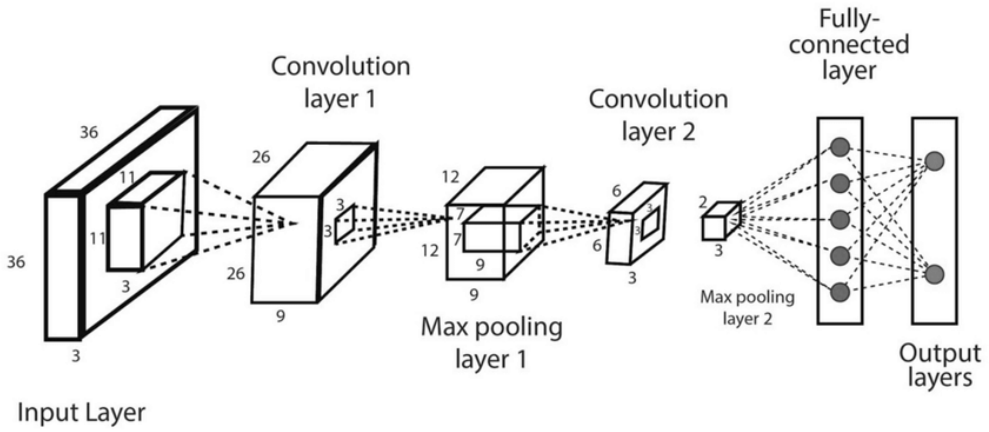


Figure 2.4: The architecture of a convolutional neural network including multiple convolutional and pooling layers [38]

This convolution operation, which the convolutional layer derives its name from, computes the output of neurons that are connected to local regions in the input and produces a feature map from the dot product between the kernel weights and the neuron's input [39]. In short, given a kernel size of 3×3, a feature map is produced by looking at a 3×3 grid of pixels in a sliding window from the output of the previous layer, extracting features as described in Subsection 2.1.3 and illustrated in Figure 2.5. There are two common parameters when applying filters—stride determines the number of pixels the filter moves for each step, and padding which extends the area of an image such that the resulting feature map of the convolutional operation preserves the dimensions of the input. The outcome of these layers are shared weights across the input image, meaning the learned filters at each layer can be applied across the whole of the image to account for variance. This property of CNNs is essential, as it drastically reduces the number of parameters in contrast to regular DNNs.

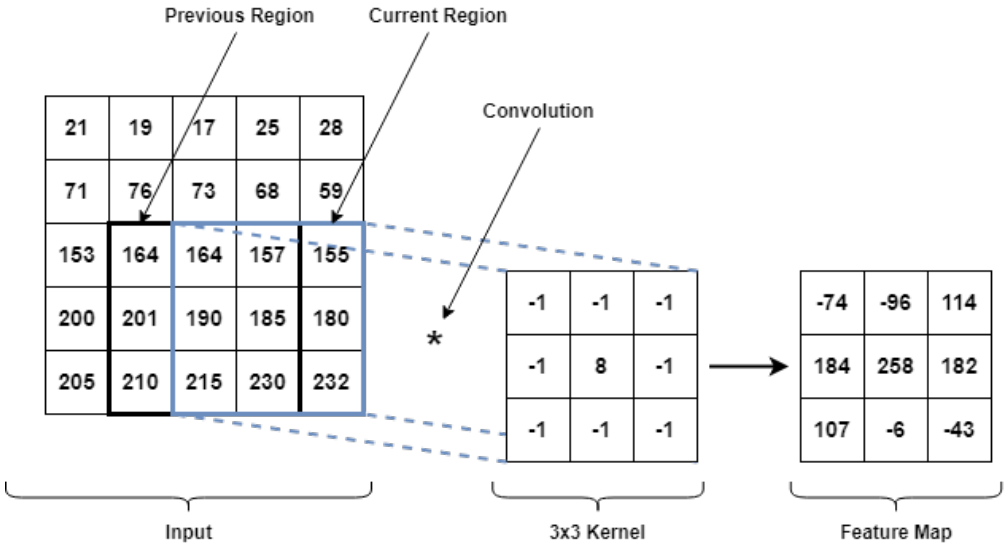


Figure 2.5: The sliding window technique of a kernel applied to an input in order to produce a feature map

Furthermore, pooling layers similarly use a sliding window to downsample feature maps, further causing a reduction of parameters. The most common pooling operation, max pooling, applies a filter of a given stride and size to calculate the maximum value that appears in each region. A max pool layer of size 2×2 with a stride of 2 will reduce the number of features by a factor of 4, as illustrated in Figure 2.6.

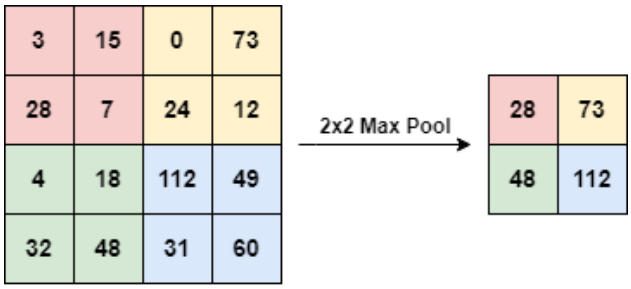
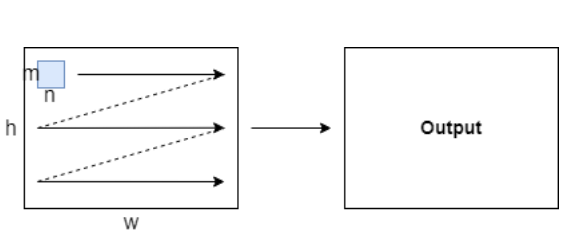


Figure 2.6: Max pool operation of size 2×2 and stride 2×2

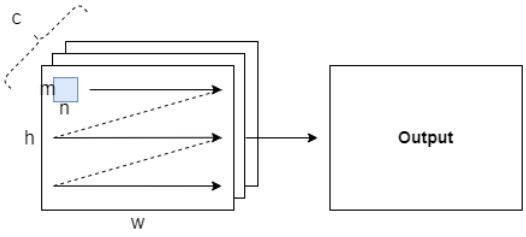
2.1.4 3D Convolutional Neural Networks

When considering filters with multiple channels, such in the case of an RGB image, each kernel is applied to each channel separately, and added together as the final output. This is considered a two-dimensional convolution (2D CNN), as each kernel only moves across the input along the height and width of each channel. On one hand, this approach is favorable for solving image classification problems as image data contain spatial information only. On the other hand, data that come in sequences, such as videos, include temporal information vital for accurate classification. As a consequence of this, CNN-based approaches to video classification problems have seen a multitude of developments in recent years to incorporate spatio-temporal feature extraction. Among these methods, a more recent one is the use of three-dimensional CNNs (3D CNNs) proposed by Ji et al. in 2013 to perform human-action recognition in video sequences [40], and have since seen application to various domains such as medical imaging [41] and crowd anomaly detection [42].

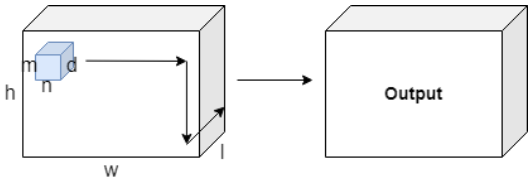
In contrast to 2D CNNs, both the temporal and spatial dimensions are captured by applying convolutions in 3D space as illustrated in Figure 2.7c. Given an input sequence of size $w \times h \times l \times c$ where l denotes the number of frames, the weights are defined by a filter of $m \times n \times c$ kernels with size $m \times n \times d$ where d is the kernel temporal depth, and m and d denote the kernel spatial size. As a result, a 3D CNN is able to extract spatio-temporal features by stacking multiple contiguous frames together and applying convolutions across the resultant volume in the shape of cubes. A comparison of 2D and 3D convolutions are shown in Figure 2.7.



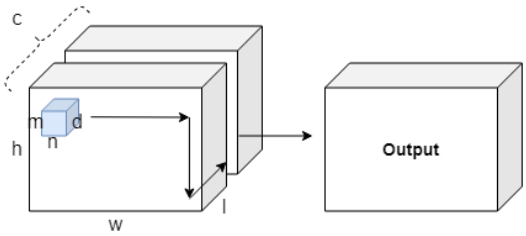
(a) 2D convolution on an image with one channel



(b) 2D convolution on multiple channels



(c) 3D convolution on series of images



(d) 3D convolution on multiple channels

Figure 2.7: Comparison of 2D and 3D convolution operations

2.1.5 Recurrent Neural Networks

While 3D CNNs are a more recent development, the idea of NNs exhibiting temporal behavior can be traced back a lot further. Building upon the works of McCulloch and Pitt's theories of networks with circles [28], Rumelhart et al. first introduced the concept of recurrent neural networks (RNN) in 1985 [32], with Rumelhart's student Jordan developing the concept further only a year later [43]. Following this, numerous works using RNNs have been published since, across a wide variety of application domains where data is sequential [44].

Illustrated in Figure 2.8, wherein the unfolded representation is simply a way conceptualize the network, RNNs cyclic nature is apparent and conveys their central idea. For an input sequence x, a hidden state vector s represents past knowledge at any discrete time step t. As such, for any given t, the hidden state from the previous time step $t-1$ along with with the current input x_t is used to derive the current hidden state s_t, and so on. The hidden state vector is illustrated as an RNN cell in Figure 2.9, and is simply a fully-connected layer with a hyperbolic tangent (tanh) activation function. Essentially, RNNs are distinguished by having working memory—hidden state from previous inputs influence the current input and output, and thus is able to extract temporal dependencies. Moreover, every time step of the sequence is processed using the same weight parameters denoted W, and leverage backpropagation through time (BPTT) to determine gradients by summing the error at each time step as opposed to feed-forward networks. By reusing weights, RNNs are able to significantly reduce the number of neurons required to process long sequences of data.

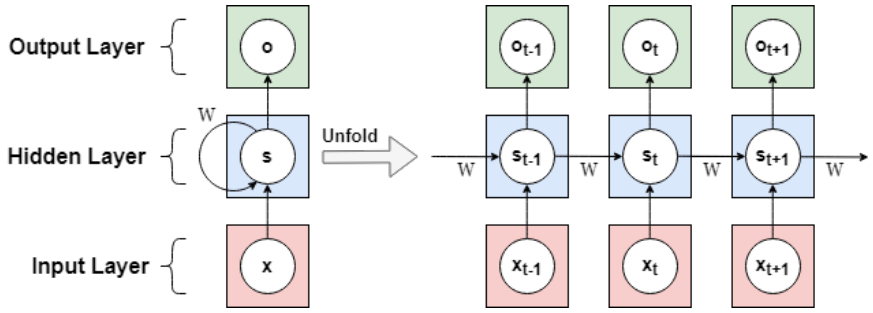


Figure 2.8: Representation of a recurrent neural network both folded and unfolded

A known limitation of RNNs is that of the vanishing or exploding gradient [45, 46]. As the backpropagated gradients either grow or shrink at each time step, very deep networks intrinsically cause an unstable gradient problem. To solve this, Hochreiter and Schmidhuber presented the Long Short-Term Memory (LSTM) cell as illustrated in Figure 2.9 in 1997 [47]. As the name implies, an LSTM cell is capable of remembering short term dependencies over long periods of time. An LSTM layer itself consists of a set of recurrently connected cells—or memory blocks—each divided into three distinct gates: the input gate, the output gate, and the forget gate. These gates are on their own a distinct NN, each with their respective sigmoid activation function that output in an interval of $[0, 1]$, and a pointwise multiplication operation comparable to that of a filter (by the definition of the word) to determine what to remember or what to forget. This is possible as the network is trained such that the sigmoid activation outputs close to zero when an input is deemed irrelevant, and closer to one when relevant. Thus each pointwise multiplication will be less or more influenced by a lower or higher activation respectively. As opposed to an RNN cell with working memory through hidden state, LSTM cells also maintain a cell state vector c that each gate connects with, representing the current long-term memory of the network. Hence, at any distinct time step t, the current input from a sequence x_t, the previous hidden state s_{t-1}, and the previous cell state c_{t-1}, each gate can be described as follows. The forget gate decides which parts of c_{t-1} should be forgotten given x_t and s_{t-1}. The input gate determines both what

information should be added to c_{t-1}, and whether is it actually worth remembering. To do this, it consists of two discrete NNs—sometimes referred to as the candidate gate [48], a tanh activated NN outputs a value in the interval $[-1, 1]$, and a regular sigmoid activated NN. The former generates a new update vector based on x_t given the context of s_{t-1} with an impact determined by the tanh activation, whereas the latter determines the relevance of the update vector. The combined vector of the pointwise multiplication is then added to c_{t-1}, resulting in the long-term memory of the network being updated. The output gate decides the new s_t and c_t by applying a pointwise tanh of c_t and passing it onto the resultant sigmoid activation of x_t and s_{t-1} by pointwise multiplication, thus passing only the current relevant information as output. Circling back to the gradient problem of RNNs, [49] details a thorough explanation of how LSTMs prevent this when backpropagating. A more recent variant of LSTM called the Gated Recurrent Unit (GRU) designed to be computationally less expensive and easier to implement was proposed by Cho et al. [50] and has since been used extensively [51].

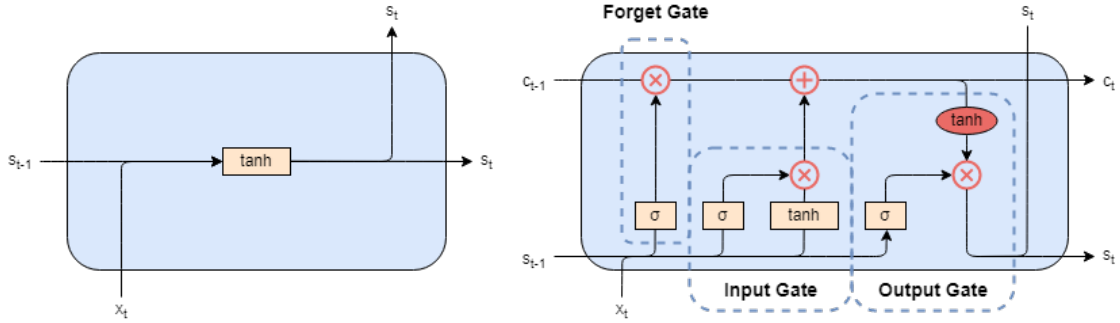


Figure 2.9: Comparison of RNN cell (left) and LSTM cell (right)

Although there is no common architecture or abbreviation, some works have also combined both CNNs and RNNs to develop what are essentially convolutional recurrent neural networks (CRNN), or recurrent convolutional neural networks (RCNN) [52, 53, 54]. To this extent, the extracted features of a CNN can be utilized as the sequential input to an RNN. Similarly, hybrid models of LSTM and CNNs (ConvLSTM) have been proposed for a variety of problems [55, 56].

2.1.6 Autoencoders

As with many other concepts presented in this chapter, the idea of autoencoders (AE) were first introduced in 1986 by the PDP group to address the problem of "backpropagation without a teacher", by using the input data as the teacher [32, 57]. The NNs discussed previously have all been supervised—i.e. they have required the use of labeled datasets in order to predict or classify outcomes accurately. AEs, on the other hand, require no labels and instead are capable of discovering hidden structure within data to learn a compressed representation of an input. Essentially, they are unsupervised models whose output is to approximate the input [58]. Traditionally used for dimensionality reduction or feature learning, AEs now span a variety of domains, such as anomaly detection [59] and image processing [60].

As illustrated in Figure 2.10, AEs consist of two phases: encoding and decoding. Given an input image x (although AEs are commonly used for images, this is not a limitation and used only for demonstrative purposes), the encoder ideally learns and describes the latent attributes of the image and maps this to a low-dimensional latent space H, often called a bottleneck, representing the compressed features. The decoder is then used to reconstruct the initial input as output $\hat{x}$ from the compressed latent space representation. To effectively learn a meaningful and generalizable H, AEs train by minimizing the reconstruction error, $\mathcal{L}(x, \hat{x})$, which measures the difference between x and $\hat{x}$.

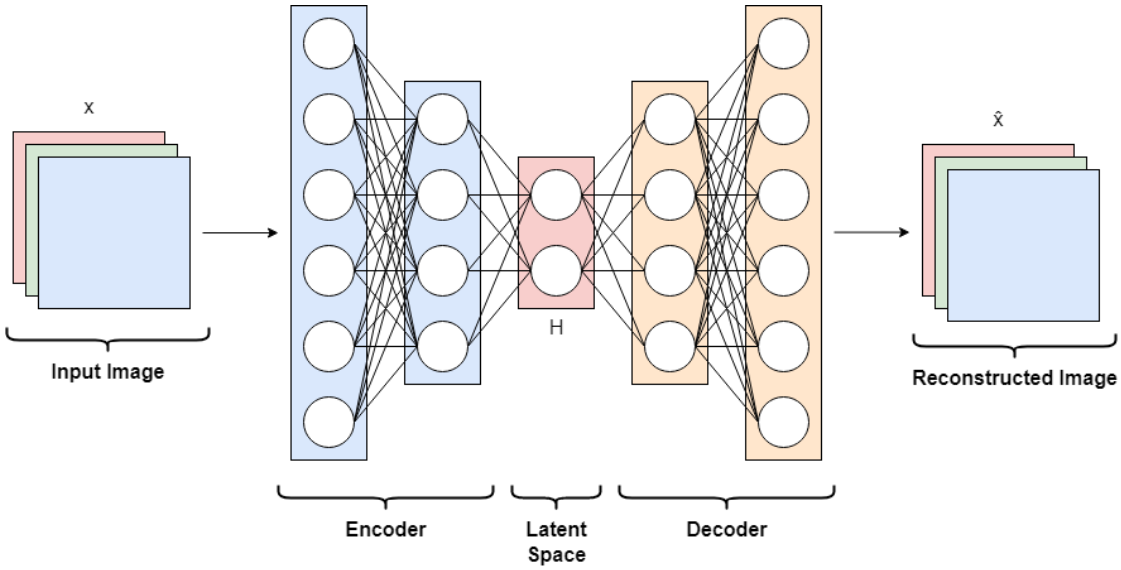


Figure 2.10: A general representation of an autoencoder architecture

Similarly to RNNs, combining CNNs with autoencoders (CAE) have shown state-of-the-art performance in image processing tasks [61, 62, 63, 64]. Additionally, some works incorporate LSTM into the AE architecture (LSTM-AE) to learn compressed representations of temporal (and spatial) dependencies [65, 66, 67]. Finally, accounting for both CNN's feature extraction ability and LSTM's ability to capture temporal patterns, hybrids of ConvLSTMs and AEs have been proposed [68, 69, 70].

It is worth noting that the concept of encoder-decoder architectures are not limited to AEs, and have seen widespread usage in both CNNs (e.g. U-Net [71] or SegNet [72]) and RNNs [73]. As encoders are able to provide the network with a rich representation of low resolution features at latent space, decoders can map these lower resolution features to an output, leading to a lighter network structure.

2.2 Optical Flow

Optical flow is the pattern of apparent motion of image objects between two consecutive frames of a sequence, caused by the relative movement between an object and the camera [74, 75]. In practice, the goal of optical flow estimation is to compute an approximation of a 2D vector (or motion) field, where each vector is defined by the displacement between aforementioned objects from one frame to the next. This problem has long been studied as a part of computer vision since the work of Horn and Schunck in 1981 [76], and with the concise description of both motion and velocity it provides, has seen application to areas such as, but not limited to, robotics [77], autonomous driving [78], and action recognition [79].

2.2.1 Traditional Methods

Traditionally, optical flow estimation requires two prerequisite assumptions—brightness constancy, and smooth motion (i.e. neighboring pixels have similar or small motion) [76, 80]. Methods taking these assumptions into account have long been the predominant and most successful ways to estimate optical flow, and are referred to as variational methods [81, 82]. On the basis of these assumptions, assuming two frames as illustrated in Figure 2.11, we can express the intensity of a pixel I as a function of its spatial position (x, y) and time t, that is $I(x, y, t)$, which moves a distance of $(\delta x, \delta y)$ over δt time to obtain the new image $I(x + \delta x, y + \delta y, t + \delta t)$.

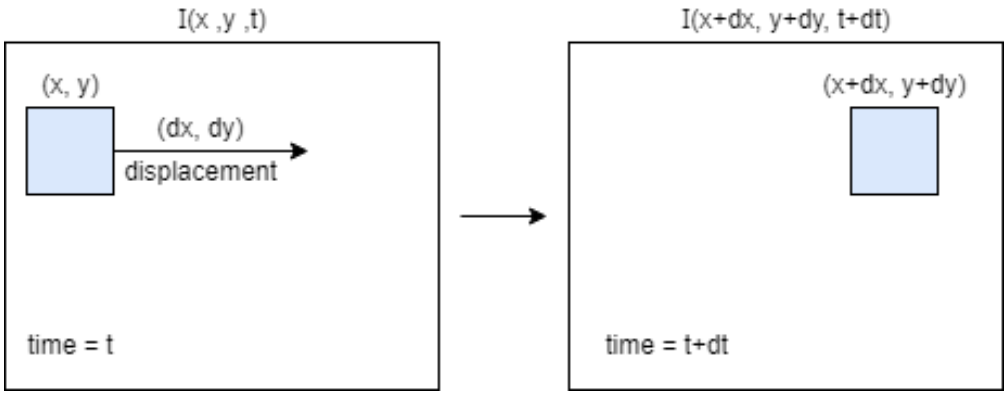


Figure 2.11: The optical flow problem [80]

Based on the assumption that the pixel intensity is constant between consecutive frames, Equation 2.2 can be established.

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{2.2}$$

To remove the common terms, a first-order Taylor expansion of Equation 2.2 is performed assuming small motion between consecutive frames:

$$\begin{aligned} I(x + \delta x, y + \delta y, t + \delta t) &= I(x, y, t) + \tfrac{\delta I}{\delta x}\delta x + \tfrac{\delta I}{\delta y}\delta y \ + \tfrac{\delta I}{\delta t}\delta t + \dots \\ \Rightarrow \tfrac{\delta I}{\delta x}\delta x + \tfrac{\delta I}{\delta y}\delta y \ &+ \tfrac{\delta I}{\delta t}\delta t = 0 \end{aligned} \tag{2.3}$$

Finally, Equation 2.3 is divided by δt using the chain rule for differentiation to derive the brightness constancy equation:

$$\frac{\delta I}{\delta x}\delta u + \frac{\delta I}{\delta y}\delta v + \frac{\delta I}{\delta t} = 0 \qquad \text{or} \qquad I_x u + I_y v + I_t = 0 \tag{2.4}$$

where $u = \frac{\delta x}{\delta t}$ and $v = \frac{\delta y}{\delta t}$ are two unknown variables of a single linear equation, characteristically known as the optical flow vectors or the displacements respectively along the x and y axes.

Essentially, solving for u and v is the definition of the optical flow problem, and forms the basis a variety of algorithms based on the assumption of brightness constancy and smooth motion to address this issue. These can broadly be classified as two separate strategies as illustrated in Figure 2.12: sparse optical flow such as the Kanade–Lucas–Tomasi (KLT) method [83], and dense optical flow such as the Farneback method [84]. The former does not compute displacements per-pixel but instead tracks a smaller number of feature points such as edges or corners to represent overall object motion (using Shi-Tomasi's Good Features to Track [85]). At the cost of being more computationally expensive, the latter will track per-pixel displacements to attain higher accuracy for matching moving objects. Important to note is that there is a lack of explicit separation between what is classified as one method or the other in existing literature, and sparse methods are commonly referred to as local methods, while dense methods are referred to as global methods. For example, what constitutes one or the other can be difficult to follow, as some works refer to variational methods as global only [86, 87, 88, 89], others refer to them as both [82, 90], and some define no clear distinction [80, 17, 91, 86]. This issue surrounds a variety of terms. For this thesis, a generalization is made that follows the aforementioned broad classification of sparse and dense optical flow.

(a) Sparse optical flow

(b) Dense optical flow

Figure 2.12: Sparse vs. dense optical flow [92]

A common evaluation technique of optical flow estimation is to visualize the result as shown in Figure 2.12b. A more concise example is illustrated in Figure 2.13, wherein the optical flow vectors are encoded by color corresponding to their direction, and the magnitude expressed by the color intensity.

Figure 2.13: Optical flow representation [93]

2.2.2 Feature Based Methods

Despite being among the most popular techniques, variational methods are susceptible to a variety of challenges as their assumptions are often violated. In practice, there is no guarantee realistic scenes uphold the given assumptions, and may consist of large displacements, strong changes in illumination, image noise, or occlusion [87]. While image processing techniques have been proposed to remedy some these problems [94], large displacements remained a bigger issue [95, 81]. This gave birth to a new variety of optical flow methods known as feature-based methods [82], wherein feature matching of images complementary serve variational methods.

Feature matching is a technique that can be divided into two parts. First, a *feature* extraction or detection step which aims to return a set of feature points. These points are located at salient image structures and are generally as a consequence very sparse, but some works have proposed feature matching for dense optical flow [96, 97, 98]. The most common feature extraction techniques are based on either edge-like structures when using algorithms such as Harris corners [99], the Canny edge detector [100], and Shi-Tomasi's Good Features to Track [85], or blob-like structures when using algorithms such as Scale Invariant Feature Transform (SIFT) [101] or Speeded Up Robust Features (SURF) [102]. The evaluation results between these algorithms have shown to depend on the lighting conditions of a scene [103]. Ideally, these feature extraction techniques should be robust to image transformations such as rotation, scale, illumination, noise and affine transformations [104]. The second step is *matching*. Here, the goal is to find feature correspondence between neighboring frames, i.e. an association of the feature points extracted from each frame. These matches are found based on visual descriptors gathered from image patches [105, 106], such as histogram of gradients [107, 108, 109] or binary patterns [110, 111] extracted around the feature points. As a result, approaches based on feature matching gain an advantage as they can compute large displacements. While they can be applied to image sequences under such circumstances for the estimation of optical flow, another common use case of feature matching is to align image pairs [112, 113].

2.2.3 Deep Learning Based Methods

The advances of optical flow estimation the past decades have largely solved the case of small displacements [86], yet the challenges presented in the previous sections have remained. However, as with other image processing tasks mentioned in this chapter, deep learning has had a massive impact on this field of research. While not an NN directly, a feature-based method called DeepFlow was proposed in 2013 by Weinzaepfel et al., based on a convolutional structure similar to that of CNNs but with no learned parameters [114]. Possibly, this sparked the exploration of deep learning for optical flow estimation as FlowNet, arguably the first deep learning optical flow architecture for realistic scenes introduced by Fischer et al. in 2015, closely relate their work [115]. Since then, over 352 deep learning-based methods have been proposed based solely on the results of the MPI-Sintel benchmark [116] (further discussed in Section 3.2).

In general, these methods learn dense optical flow by computing per-pixel predictions. What separates most deep learning methods for optical flow with that of other DNNs, is that they require networks to learn per-pixel features for two separate input frames before combining them at a higher level. This roughly resembles the matching approach described in the previous section [115]. Furthermore, said methods have long relied on synthetic datasets such as Flying Chairs [115] or the previously mentioned MPI-Sintel dataset [117] which do not reflect genuine scenes specific to that of particular domains. As such, their performance may depend on the content and the application for which they are used [118]. While both supervised and unsupervised learning methods are present in current literature, the best performing are those of the supervised nature based on current benchmarks.

2.3 Crowd Anomaly Detection

Crowd anomaly detection has long stood at the forefront of problems in the field of computer vision. Although a unanimous definition of an anomaly has yet to be met, the general consensus is based on that of observed deviations from normal behavior in crowded scenes. What constitutes said behavior is derived from spatial and temporal dynamics of a given scene, and as such it is difficult or impossible to form generalized patterns upholding different environments and conditions. To this end, the crowd anomaly detection problem encompasses a wide range of difficulties in which research has accomplished significant strides in recent years. Overall, there is a great diversity of approaches considering the ambiguous nature of different crowds, and access to relevant datasets pose one of the key challenges for the detection of anomalous events. In particular, methods have been proposed to detect anomalous motion patterns of pedestrians [119, 120, 121], unexpected presences within crowds [122, 123, 124], escape panics [125, 126, 127], violent behaviors [128, 129], traffic accidents [128], and more. These methods have conventionally relied on the hand-crafted extraction of crowd features to effectively model a scene, but recent advances in deep learning has contributed to automatic feature extraction, or a mixture of both. The intent of this section is to outline the various features and methods used in literature the last decade for the detection of anomalies in crowds.

2.3.1 Crowd Features

The extraction of crowd features play a crucial role in the crowd anomaly detection problem. Depending on the approach, these features exist as a set of metrics to be evaluated over time for a given individual (a microscopic approach) or crowd as a whole (a macroscopic approach) from a video sequence [130]. In their work, Sánchez et al. [6] propose a taxonomy as shown in Figure 2.14 which identifies the following relevant crowd features for the understanding of crowd behavior:

- **Velocity.** Measures the average speed at which individuals at a microscopic level, or crowds at a macroscopic level, are moving. This feature can be equated to the magnitude of an optical flow vector as described in Subsection 2.2.1 [131, 132, 133].
- **Direction.** At a microscopic level determines the main directions of movement followed by each individual, or the crowd as a whole at macroscopic level [134].
- **Density.** Determines the density of a crowd given the proximity of individuals. At macroscopic level where dense crowds make it difficult to separate individuals at microscopic level, density estimation is performed instead [134].
- **Collectiveness.** Measures the degree of individuals acting as a union in collective motions [135, 130].
- **Valence.** Measures the positive and negative affect of a crowd given a combination of velocity and density at macroscopic level [136].
- **Arousal.** Aimed at monitoring how calm or excited a crowd is based on variance of motion magnitude [136].

Some works have further quantified other crowd features, such as speed or merging probabilities [137], uniformity and conflict [138, 139, 140], stability [138, 139, 141, 142, 140], or trajectories representing both direction and velocity extracted from methods such as sparse optical flow [143, 144, 124].

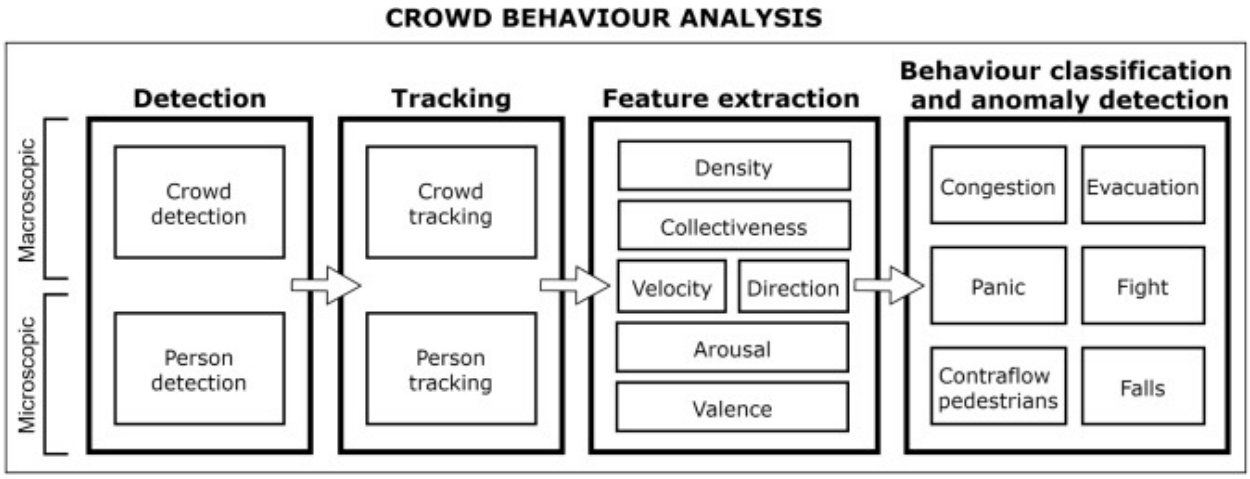


Figure 2.14: The four main stages of the crowd behavior analysis pipeline [6]

2.3.2 Traditional Methods

At large, crowd features such as density, motion, and trajectory are analyzed from surveillance videos to detect abnormalities in crowded scenes [6]. The extracted features are further used to model the interactions of individuals and the activities present in the scene [145], followed by a detection stage. In the conventional sense, [6] classifies the many research works analyzing such features and detection means for crowded scenes into four methods: Gaussian Mixture Model (GMM) [146, 147, 148] or Hidden Markov model (HMM) [149, 150, 151] techniques modeling normal behavior patterns to detect abnormal patterns, and optical flow [152, 153, 154] or spatio-temporal techniques [155, 156, 157] analyzing motion over time. Important to note is that these techniques are commonly used in conjunction with one another in a hybrid approach, and are not strictly separate. Besides the aforementioned classification, some works have proposed sparse coding or dictionary learning based methods [158, 159, 160]. However, these conventional techniques have shown to be ineffective at identifying complex patterns, and difficult to implement in real-time applications [161].

2.3.3 Deep Learning Based Methods

Extensive surveys have been conducted on models that employ deep learning to solve the crowd anomaly detection problem [6, 162, 163, 164]. Given the vast history of literature within this topic, a clear timeline is nigh impossible to cover—nevertheless, it stands to reason CNNs and AEs are among the most popular of choices for the detection of anomalies within crowds. The former is applied in a supervised fashion, oftentimes using pre-trained networks to avoid constructing a model from scratch. The latter is an unsupervised approach focused on minimizing a reconstruction error. Thus, this subsection brings attention to their use in literature over the last decade, whereas Chapter 3 will detail the current state-of-the-art within this field fusing optical flow methods with deep crowd anomaly detection.

Supervised CNN-based methods have been proposed to detect anomalous events or behavior for specific domains. As they require annotated datasets, general-purpose anomaly detection is largely seen as not possible. Nevertheless, given specific requirements for a particular environment, they have proved their effectiveness [161]. That being the case, research work in this area can generally be divided into (i) 2D CNNs; (ii) 3D CNNs; (iii) pre-trained CNNs; and (iv) LSTM-based models. Furthermore, in an unsupervised manner, a diverse range of AE-based methods have been proposed (v).

(i) **2D CNNs.** While most existing techniques have relied on pre-trained CNN models, some works applied rigorous pre-processing steps based on hand-crafted features to form hybrid models. In this sense, most works have relied on spatio-temporal CNNs using extracted optical flow [165, 128, 166], wherein classification was commonly performed by CNNs themselves [167, 168, 169] or Support Vector Machines (SVM) [170, 171]. Other approaches include using a CNN to classify extracted spatio-temporal features such as collectiveness, stability, conflict, and density using a combination of KLT and GMM [172]—or simply regular end-to-end CNNs [161].

(ii) **3D CNNs.** Few works have attempted to use 3D CNNs that do not rely on pre-trained networks. Nonetheless, given their ability to capture spatio-temporal features, interest has grown over recent years. In their work, Ullah et al. [173] used a 2D CNN to identify frames with individuals from surveillance videos, and pass those frames to a 3D CNN to capture spatio-temporal features and perform classification of violence. Similarly, Song et al. [174] proposed a 3D CNN using a novel sampling method to extract a number of key frames in a pre-processing step. Sabokrou et al. [175] explored the use of an AE to detect cubic patches of frames to find regions of interest, of which interesting patches are fed into a deeper 3D CNN.

(iii) **Pre-trained CNNs.** To train CNNs, there are generally two options—training the domain specific problem from scratch, or using a pre-trained model, often referred to as transfer learning [176]. To this extent, most of literature that explore CNNs for crowd anomaly detection rely on pre-trained networks. For this purpose, a variety of both pre-trained 2D CNNs and 3D CNNs have already been built and trained on images. For the detection of video anomalies, Gutoski et al. [177] performed a comparative study of transfer learning approaches using twelve different pre-trained 2D CNN models on seven datasets. For example, AlexNet [178] was among the first networks used for transfer learning, of which [179, 180] used to extract high-level features for abnormal events in crowded scenes. VGGNet [181] improved upon AlexNet by introducing a deeper architecture of up to 19 layers, and was used by several works to extract spatial features of input videos [182, 183, 184, 185, 186]. An exceedingly deeper network of up to 152 layers, dubbed ResNet (Residual Network) [187], was later proposed and used to train several crowd anomaly detection methods [188, 189, 190, 191, 192]. To additionally capture behavior in the temporal domain as well, modifications of the aforementioned models have been proposed to perform 3D convolutions. As such, Zheng-ping et al. [193] used a pre-trained 3D VGGNet model for anomaly detection, and a 3D version of ResNet [194] was applied by some authors for video anomaly detection [129, 195, 196].

(iv) **LSTM-based.** The use of 3D convolutions to capture spatio-temporal features can result in time-consuming training [197]. As a result, LSTMs have been used for temporal modeling to reduce computation time. A CNN used to extract features and an LSTM used as a mechanism for memory was proposed by [198]. In their work, Zhou et al. [199] used a ConvLSTM-based unit to learn spatio-temporal features for the detection of abnormal events. Additional works have also studied the impact of ConvLSTMs for anomaly detection [200, 201, 202]. Furthermore, Alahi et al. [203] was inspired by RNNs and introduced Social-LSTM for human trajectory prediction in crowded scenes.

(v) **Autoencoders.** Among this work, [204, 205, 206, 207, 208] proposed Variational Autoencoders (VAE) [209] for abnormal event detection. VAEs describe the potential latent state in a probabilistic way, and as such provide a reliable way of detecting anomalies given their small probability of appearing [204]. Furthermore, 2D CAEs have been employed in combination with GMMs [210] or LSTMs [211]. Similarly to that of CNNs, 3D CAEs overcome the problem of losing temporal information and has seen use for video anomaly detection [212, 213, 214].

Chapter 3

State of the Art

This chapter presents the current state-of-the-art in the fields of optical flow estimation and crowd anomaly detection using deep learning fused with optical flow.

3.1 Deep Learning for Optical Flow

At present, optical flow methods based on deep learning have completely outperformed the traditional methods in both accuracy and run time [82]. Among these, the current state-of-the-art is largely supervised methods that build upon novel concepts already adopted by traditional methods, or extending previous works. This section aims to describe a few of the most popular ones.

3.1.1 FlowNet

In the earliest FlowNet, the authors propose and compare two CNN-based architectures: FlowNetS and FlowNetC illustrated in Figure 3.1. In FlowNetS, the authors simply stack both input images together and feed them through a convolutional architecture, allowing the network itself to decide how to extract motion information from the image pair. FlowNetC on the other hand is based on two separate, yet identical processing streams for each input image later combined by a "correlation" layer. This layer performs multiplicative patch comparisons between two feature maps. More specifically, given two multi-channel feature maps, the correlation of two patches are computed similarly to that of a convolutional step, but instead of convolving data with a filter, it convolves data with other data and thus has no trainable weights. The result is an end-to-end architecture with matching capabilities. Moreover, each of the two proposed architectures include a refinement step of "upconvolutional" layers. As pooling results in reduced resolutions, this step is required in order to provide dense per-pixel predictions. As such, the feature maps are extended by unpooling and a convolution, then concatenated with their corresponding feature maps. This way, the authors preserve both high-level information and fine local information. They were the first to show that deep learning could directly predict optical flow, and both architectures outperformed traditional methods such as DeepFlow [114] and EpicFlow [215].

Building upon the works of [115], Ilg et al. proposed FlowNet2 less than a year later in 2016 [216]. Their work had four main contributions. First, the authors were able to improve the results of both FlowNet architectures just by modifying the datasets, as they observed that the schedule of presenting data with different properties mattered during the training process. Second, they proposed stacking multiple FlowNetS and FlowNetC architectures. To this extent, the authors experiment with both the order of networks, and an image warping technique of the second input image towards the first image using the computed flow estimate of the previous network and bilinear interpolation. Third, they created a new dataset with small displacements to further fine-tune the best performing stacked architectures. Fourth,

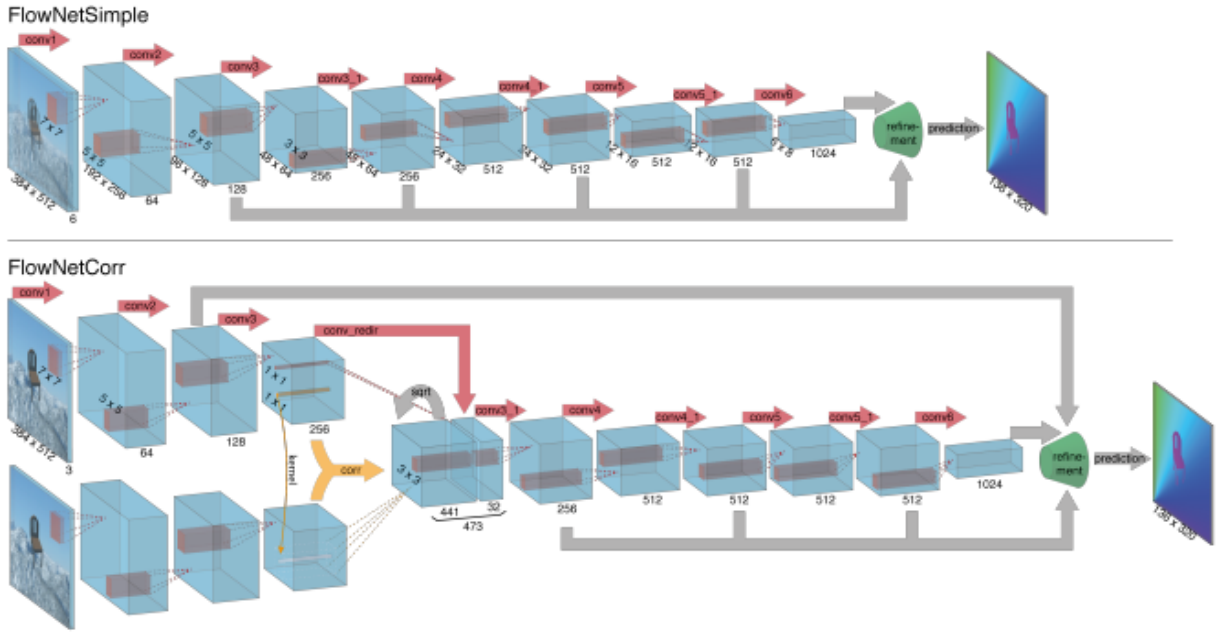


Figure 3.1: The two network architectures: FlowNetSimple (top) and FlowNetCorr (bottom) [115]

a fusion network of a stacked architecture consisting of one FlowNetC and two FlowNetS architectures and a modified FlowNetS denoted FlowNet2 was proposed. The final result was an architecture that runs orders of magnitude faster than the previous state-of-the-art, with higher accuracy and without compromise for small and large displacements or noise.

The drawback of FlowNet2 is that it comprises over 160 million parameters to achieve accurate flow estimation. To combat this issue, Hui et al. proposed the refined CNN-based encoder-decoder network LiteFlowNet in 2018 [217], as illustrated in Figure 3.2. The authors achieved this by introducing two general concepts, namely pyramidal feature extraction to infer flow fields, and feature warping. The former is a common feature extraction technique that takes an input and outputs proportionally sized feature maps at multiple levels in a convolutional fashion [218], which leads to a lighter network compared to that of FlowNet2. The latter improves upon FlowNet2's image warping technique by directly warping the feature maps of the second image. This way, the authors achieved performance on par with that of FlowNet2 while faster in run-time, and being 30 times smaller in model size.

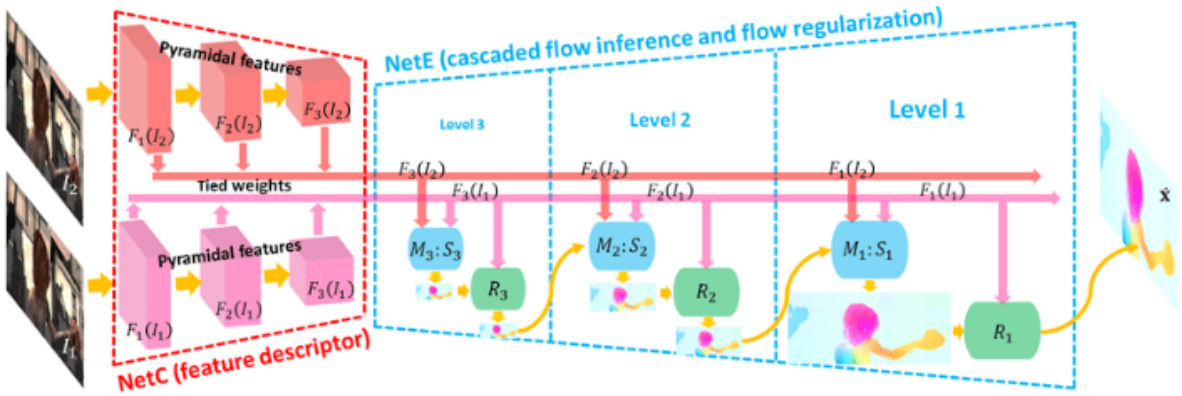


Figure 3.2: The network structure of LiteFlowNet [217]

The authors of LiteFlowNet further improved their work a year later and introduced LiteFlowNet2 [219]. By optimizing their previous network architecture, they attained better flow accuracy than their previous work while being 2.2 times faster in run-time. As such, they outperformed the previous state-of-the-art FlowNet2. Another year later in 2020, two of the previous authors of LiteFlowNet2 Hui and Loy proposed LiteFlowNet3 as illustrated in Figure 3.3, aimed at solving ambiguous correspondences caused by challenges such as occlusion or illumination changes [220]. To address these issues, they introduced specialized CNN modules to improve feature matching, while remaining a fast, lightweight network.

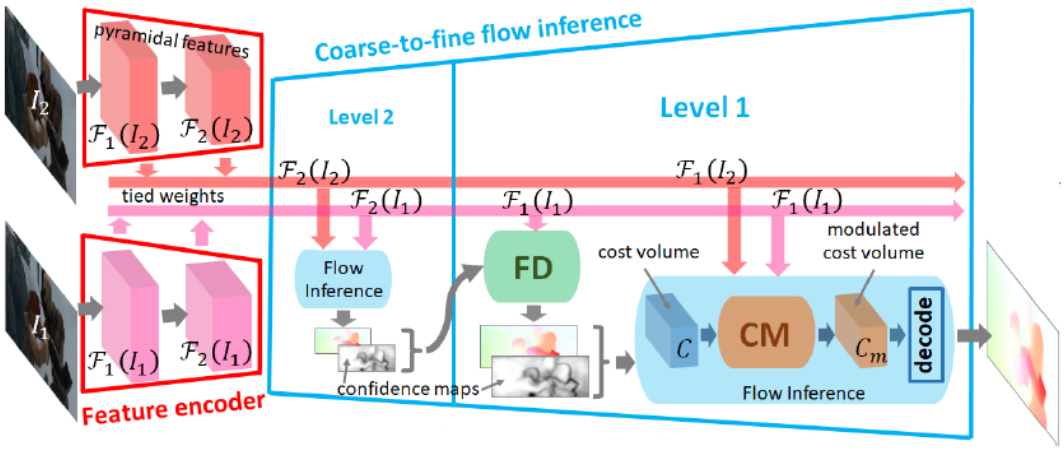


Figure 3.3: Simplified network architecture of LiteFlowNet3 [220]

3.1.2 Recurrent All-Pairs Field Transforms

Aiming to design an effective architecture for optical flow estimation, Teed and Deng introduced Recurrent All-Pairs Field Transforms (RAFT) in 2020 [221]. While previous works have often relied on end-to-end CNNs, the authors proposed a convolutional recurrent architecture to iteratively update a flow field as illustrated in Figure 3.4. Comprised of three components, a feature encoder extracts per-pixel feature vectors from both input images, along with a context encoder extracting features only from the first frame. Similar to that of FlowNet, a correlation layer produces a correlation volume for all pairs of pixels in a pyramid-fashion to compute visual similarity. Finally, a GRU-based iterative approach estimates a sequence of flow estimates based on the features retrieved from the correlation pyramid and the context network. By maintaining and updating a single fixed flow field at high resolution, RAFT is both lightweight and robust against challenges such as small fast-moving objects. Given its state-of-the-art performance, several authors have based their optical flow work on RAFT [222, 223, 224, 225, 226, 227].

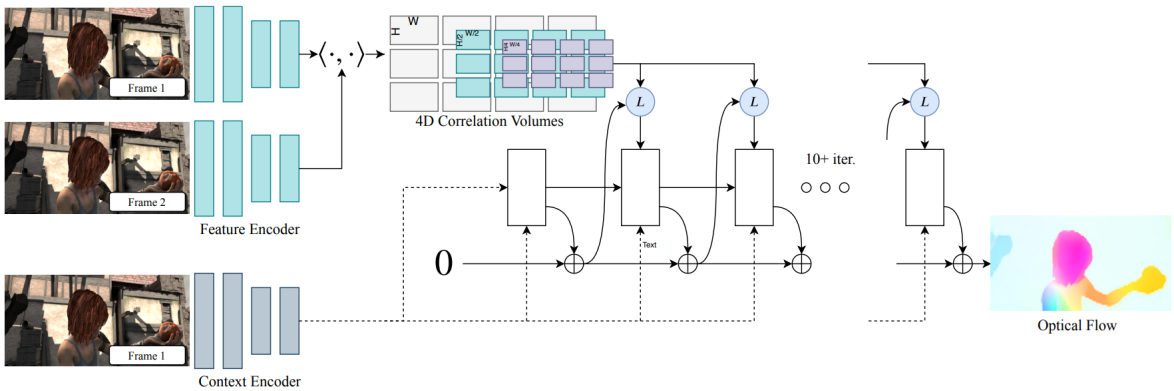


Figure 3.4: The network structure of RAFT [217]

3.1.3 Global Motion Aggregation

Jiang et al. proposed Global Motion Aggregation (GMA) in 2021 directed at solving the occlusion problem [228]. In contrast to other state-of-the-art methods, GMA is inspired by the recent success of transformer-based NNs [229], in which the idea of attention is the key contribution to the optical flow problem. Existing as a self-contained addition to the RAFT architecture as illustrated in Figure 3.5, the network has added flexibility towards choosing between or combining the local and global motion features depending on the needs of specific pixel locations. To achieve this, the GMA module globally aggregates motion features based on appearance self-similarity of the first input image, and concatenates them with the local motion features and the context network to be decoded by the GRU. As such, local image regions such as those caused by occlusion could preference the global motion features.

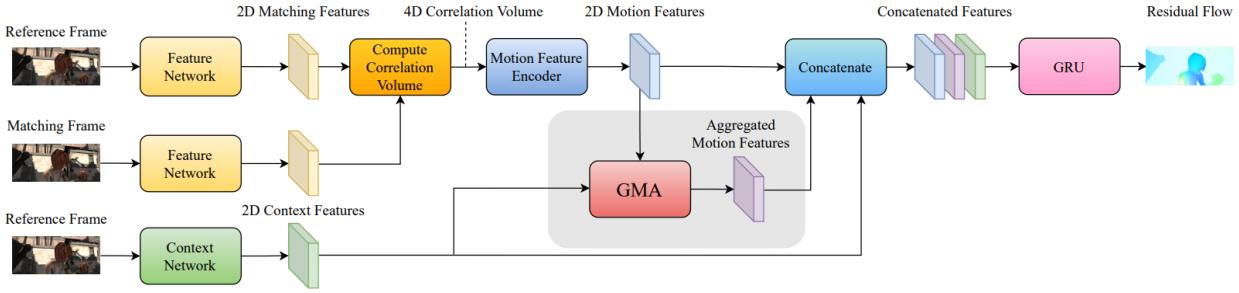


Figure 3.5: The self-contained GMA module added to the RAFT architecture [228]

3.2 Datasets for Optical Flow

Well-defined datasets play a crucial role in computer vision tasks, and optical flow is no outlier. In fact, the field of optical flow was among the first to introduce standard datasets for quantitative comparisons in 1994 [230]—four years earlier than the classic MNIST handwritten digits database [231] commonly used for supervised learning tasks. As optical flow estimation techniques have improved over the last decade, the demand for more challenging datasets grew, and the first dense ground truth dataset Middlebury was introduced by Baker et. al [232] in 2011. With new evaluation standards, this work was followed by the current state-of-the-art benchmarking datasets KITTI [233, 234], MPI-Sintel [235], and Flying Chairs [115] in following years. These present their own sets of challenges, and as such have an impact on the scenes one wishes to estimate the optical flow of.

3.2.1 KITTI

The KITTI dataset was first created by Geiger et al. in 2012 [233] (thus oftentimes referred to as KITTI-2012), comprised of 194 training and 195 test image pairs. All images are grayscale and include complex lighting conditions and large displacements, and were gathered from stereo videos of realistic road scenes. Later in 2015, Menze and Geiger introduced the extended KITTI-2015 dataset comprised of 200 training and 200 test scenes, obtained by annotating 400 dynamic scenes from the KITTI-2012 raw data collection using detailed 3D CAD models for all vehicles in motion [234]. In general, the ground truth was obtained by combining recordings from calibrated cameras and a 3D laser scanner. For KITTI-2015, the 3D CAD models are fitted to the point clouds obtained by the laser. However, given occlusion or distant objects, the ground truth remains an approximation which is taken into account for evaluation metrics.

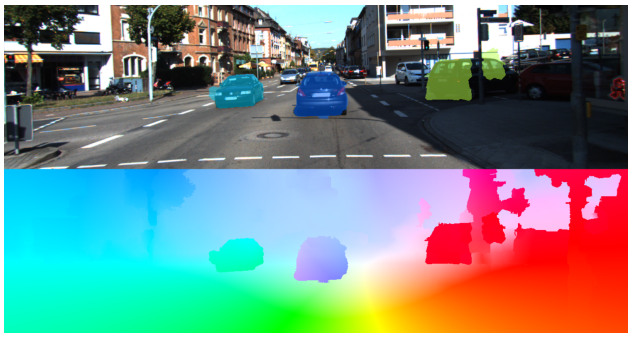


Figure 3.6: Frame and the corresponding ground truth from the KITTI-2015 dataset [233]

3.2.2 MPI-Sintel

Before the introduction of KITTI-2015, MPI-Sintel was the largest optical flow dataset created by Butler et al. in 2012. Derived from the open source 3D animated short film Sintel, this dataset is completely synthetic and includes scenes under conditions of varying complexity to pose new challenges for estimation methods. To this end, the dataset is comprised of three versions given different render passes: albedo, clean, and final. In the albedo version, frames consist of flat, unshaded surfaces exhibiting constant albedo over time. The clean version introduces illumination and specular reflections, and the final version is a fully realized render with intricate features such as depth of field and atmospheric effects. In total, each version contains 1064 training and 564 test frames. For research purposes, the clean and final versions are the most commonly used, as they provide the most realistic image sequences.

Figure 3.7: Frame and the corresponding ground truth from the MPI-Sintel dataset [235]

3.2.3 Flying Chairs

Designed specifically for training CNN-based optical flow estimation methods, Flying Chairs was introduced as a synthetic dataset by the authors of FlowNet [115]. As previous datasets proved too small to train CNNs, a larger dataset was required. As such, the authors applied affine transformations to real images and synthetically rendered chairs based on randomly sampled parameters, resulting in 22,872 image pairs and flow fields.

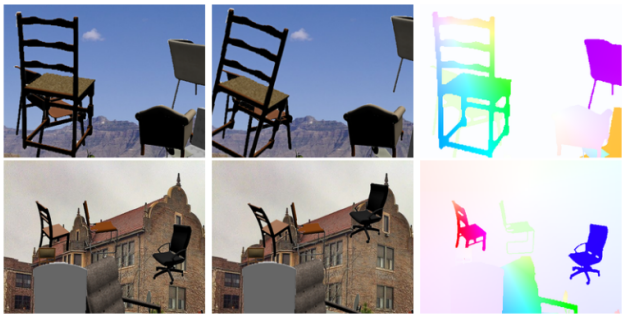


Figure 3.8: Frames and their corresponding ground truths from the Flying Chairs dataset [115]

3.3 Deep Learning for Crowd Anomaly Detection

The ability of dense optical flow to capture motion patterns in videos has shown effective results in the detection of anomalies in crowds. This section outlines both the state-of-the-art deep learning methods for crowd anomaly detection fusing optical flow between 2019-2022, and commonly used performance metrics.

3.3.1 Methods Fusing Optical Flow

In their work, Duman and Erdem [9] proposed an LSTM-based CAE, using dense optical flow to obtain velocity and direction information to detect anomalies in an unsupervised manner. The authors used the Farneback algorithm to obtain optical flow maps of eight consecutive video frames, to be used as input to the deep learning model. Nguyen and Meunier [236] use a pre-trained FlowNet2 model to estimate optical flow and propose an end-to-end CNN-based architecture for anomaly detection in video sequences. For this purpose, their model incorporates two processing streams. The first is a CAE used to learn spatial structures, while the second is a U-Net based structure used to predict instant motion given an input video frame. A shared encoder forces the model to learn correspondence. To this end, their approach exploits the correspondence between pattern appearances and their motions. Direkoğlu [18] used optical flow vectors to propose motion information images. The author observed that the optical flow angles were different when comparing abnormal situations to normal situations. As such, the author estimated the optical flow at each frame using the Lucas-Kanade algorithm, and calculated the angle difference between optical flow vectors in consecutive frames. To this end, there is a significant difference between abnormal and normal motion information images. The proposed method was evaluated using a simple CNN and popular pre-trained CNNs. Lucas-Kanade was likewise used by Sabih and Vishwakarma [237] to estimate optical flow, and used a supervised bidirectional ConvLSTM to classify normal and abnormal frames.

3.3.2 Performance Metrics

While a large number of performance evaluation metrics have been used across literature depending on both the dataset and detection method [5], a few are more prevalent than others. Further used for the evaluation of methods in this thesis, these are defined as follows:

- **True Positive (TP).** The number of anomalous frames correctly predicted.
- **True Negative (TN).** The number of normal frames correctly predicted.
- **False Positive (FP).** The number of normal frames predicted as anomalous.
- **False Negative (FN).** The number of anomalous frames detected as normal.
- **Accuracy.** The ratio of correct predictions to the total number of predictions, computed as:

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN} \tag{3.1}$$

- **Precision.** The ratio of correctly predicted positive observations to the total predicted positive observations. Precision is intuitively the ability of the classifier not to label as positive a sample that is negative [238]:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

- **Recall.** The ratio of correctly predicted positive observations to all observations for a given label. Recall is intuitively the ability of the classifier to find all the positive samples [239]:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

- **F1 Score.** The harmonic mean of precision and recall, often used for uneven class distributions:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3.4}$$

- **Receiver Operating Characteristic (ROC).** Defined as a curve showing the performance of a classification model by plotting the true positive rate against the false positive rate at different thresholds [240].

- **Area Under the ROC Curve (AUC).** The AUC is one of the most widely used metrics for crowd anomaly detection. It is an aggregate measure of performance across all possible classification thresholds by calculating the area under the ROC curve [240]. As such, an AUC of 0 indicates that all predictions are wrong, whereas an AUC of 1 indicates all predictions are correct.

- **Regularity Score (RGS).** Commonly used for reconstruction tasks. Defined as the reconstruction error of a pixel's intensity value I at location (x, y) in frame t of a video sequence [241]:

$$e(x, y, t) = \|I(x, y, t) - fw(I(x, y, t)\|_2 \tag{3.5}$$

  where I is the input frame and fw is the reconstructed frame. Given a frame t, the sum of all pixel-wise errors in a frame is subsequently derived as:

$$e(t) = \sum_{(x,y)} e(x, y, t) \tag{3.6}$$

  and a following regularity score $s(t)$ of a frame t is computed as:

$$s(t) = 1 - \frac{e(t) - min_t e(t)}{max_t e(t)} \tag{3.7}$$

  To this end, a lower value signifies a higher chance of an anomaly occurring for a given frame, as a result of significant error between an input frame and the reconstructed output frame.

3.4 Datasets for Crowd Anomaly Detection

Real world datasets of crowded scenes are comprised of observational videos commonly collected from mounted cameras or CCTVs. Every area of crowd analysis has their own respective datasets, each with their own set of distinctive crowd features to take account of. This section looks at the most widely used state-of-the-art datasets that include abnormal events, namely UCSD [242], CUHK Avenue [159], ShanghaiTech Campus [243], Subway [244], and UCF-Crime [245].

3.4.1 UCSD

The UCSD (University of California San Diego) anomaly detection datasets are the most widely used datasets in current literature. Split into two subsets named Pedestrian 1 (Ped1) and Pedestrian 2 (Ped2), both training frames and test frames were acquired with a stationary camera mounted at an elevation, overlooking pedestrian walkways. Specifically, Ped1 contains 34 training video sequences and 36 testing video sequences of pedestrians walking towards or away from the camera. Ped2 contains 16 training video sequences and 12 testing video sequences of pedestrian movement parallel to the camera plane. In both datasets, the training frames are regarded as normal with varying crowd density. In addition to normal frames, the testing sets contain abnormal events that include non-pedestrian entities such as bikers, skaters, carts, or people walking on the grass, and anomalous pedestrian motions. Both datasets include ground truth annotations indicating whether an anomaly is present at a given frame.

(a) A cart in UCSDPed1 (b) A cart in UCSDPed2

Figure 3.9: Anomalous examples from the UCSD datasets [246]

3.4.2 CUHK Avenue

The videos of this dataset were captured at the CUHK (Chinese University of Hong Kong) campus avenue, and contains 16 training video sequences and 21 testing video sequences. The training videos capture normal situations, whereas testing videos include both normal and abnormal events such as strange actions, wrong directions, or abnormal objects. In addition, the dataset contains challenges such as a slight camera shake, a few outliers in the training data, and some normal patterns that seldom appear in the training set [159]. The ground truth dataset includes abnormal events marked in rectangles.

Figure 3.10: Anomalous example from the CUHK Avenue dataset (wrong direction) [159]

3.4.3 ShanghaiTech Campus

Collected at the ShanghaiTech University campus, this dataset is comprised of 13 scenes with complex light conditions and camera angles. Comprised of 330 training and 107 test videos, of which there are 130 abnormal events and over 270,000 training frames, it is considered one of the largest and most challenging datasets available for anomaly detection in videos [247]. Abnormal events are produced by strange objects in the scene, pedestrians moving at anomalous speed (e.g. running or loitering), moving in unexpected directions, sudden motion, chasing, or brawling [6, 248].

Figure 3.11: Anomalous example from the ShanghaiTech Campus dataset

3.4.4 Subway

The Subway dataset was collected from the recordings of an entrance platform and an exit platform in an underground train station. The videos are 2 hours long in total, and contain 209,150 frames of which anomalies are represented by wrong directions, loitering, or avoiding payment [159]. Both videos are annotated at frame-level.

(a) A person tries to exit through the entrance gate

(b) A person tries to enter through the exit gate

Figure 3.12: Anomalous examples from the Subway dataset

3.4.5 UCF-Crime

The UCF (University of Central Florida) crime dataset consists in total of 1900 untrimmed real-world surveillance videos that cover 13 abnormal events: abuse, arrest, arson, assault, road accident, burglary, explosion, fighting, robbery, shooting, stealing, shoplifting, and vandalism. These were chosen due to their impact on public safety. It is divided into a training set consisting of 800 normal and 810 anomalous videos, and a testing set which includes the remaining 150 normal and 140 anomalous videos.

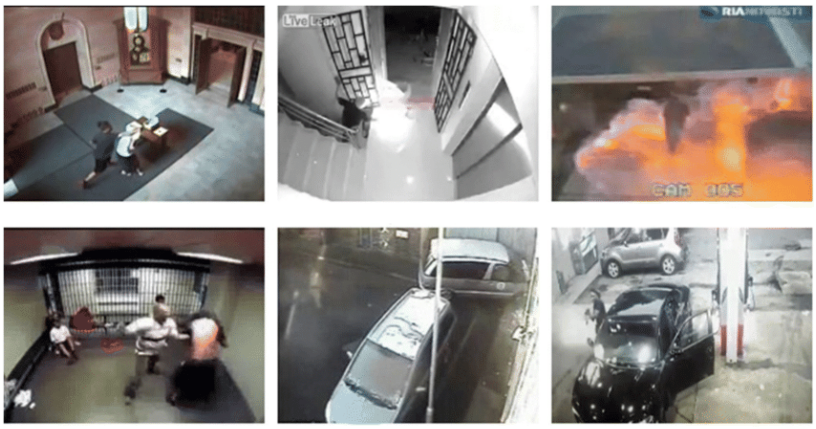


Figure 3.13: Anomalous examples from the UCF-Crime dataset [1]

Chapter 4

Method

Besides the authors of [236] who estimated optical flow using FlowNet2, there seems to be a lack in literature experimenting with state-of-the-art deep optical flow estimation methods. While this may simply be due to computational costs for real-time systems, it is an intriguing area to explore nonetheless. As a result, the goal of this thesis is to evaluate how different deep learning approaches to optical flow estimation may have an effect on the performance of different crowd anomaly detection methods, represented by popular image or video classification architectures. This chapter will discuss the methods used to attain this goal.

4.1 PyTorch

PyTorch is an open source machine learning framework for Python [249], standing as an alternative to popular frameworks such as Caffe [250] or TensorFlow [251]. It strives to make writing models, data loaders, and optimizers as easy and productive as possible, while remaining performant. As such, it provides an imperative and Pythonic programming style that supports code as a model, while remaining efficient and supporting hardware accelerators (e.g. GPUs).

4.2 Dataset

The UCSD Ped1 dataset provides a simple set of videos where anomalies are easily recognizable. While it may not reflect that of real-world surveillance footage given its videos were recorded at the same time of day during similar weather conditions, it is easily approachable. All videos are recorded in grayscale at a resolution of 238×158, and adds up to 6800 individual training frames and 7200 test frames. 4005 frames of the test set are deemed abnormal. This is in stark contrast to, for example, the ShanghaiTech Campus dataset which contains 274,515 training frames of size 856×480. In consideration of that, the UCSD Ped1 dataset allows for both faster estimation of optical flow and model training given its size. Moreover, it includes a binary flag per frame indicating whether an anomaly is present at that frame, allowing development of supervised models with ease.

4.3 Convolutional Neural Networks

Two different CNN architectures are used for the detection of anomalies in crowds—a simple 2D CNN with no regard to the temporal dimension, and a 3D CNN to capture spatio-temporal features. The former treats the issue as a regular binary image classification task. The latter views each video as a 3D image and classifies each frame given the learned 3D filters. For both architectures and the rest discussed in this chapter, each model receives an input of grayscale images (one channel) in the case of regular frames, or RGB images (three channels) for optical flow. For demonstrative purposes, all illustrations are based on an input with three channels.

4.3.1 2D CNN

The 2D CNN architecture is very simple, and takes only a single image as an input and outputs the probability that the image is either normal or abnormal. This model relies on the feature extraction ability of 2D convolutions to capture spatial features representing an anomaly. The 2D CNN receives an image of size $227 \times 227 \times 3$, and feeds it through an end-to-end architecture comprised of three convolutional blocks. Each block represents a 2D convolutional with the following layers:

1. A 2D convolutional layer with a kernel size of 5×5, 3×3 and 3×3 respectively, each with a stride of 1.
2. A batch normalization layer to make training faster and more stable by standardizing the outputs of hidden units across an entire batch [252].
3. The non-linear Rectified Linear Unit (ReLU) [253] activation function, defined as follows:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \tag{4.1}$$

4. Dropout is a technique where randomly selected neurons are ignored during training. While the authors of [252, 254] suggest batch normalization eliminates the need for dropout, other studies disagree [255]. Furthermore, the use of dropout has proven effective for datasets of limited size to avoid overfitting [256, 257]. As a result, each convolutional block consists of a dropout layer with a probability $p = 0.25$.

The last convolutional block is followed by a max pool of size 2×2 with a stride of 2, fed into a final convolutional layer with a kernel size of 56×56 instead of a flattened feature map fed to a dense layer as illustrated in Figure 4.1. The cross entropy loss is computed between the input and the target and returns two values representing the probability of the output being either normal or abnormal.

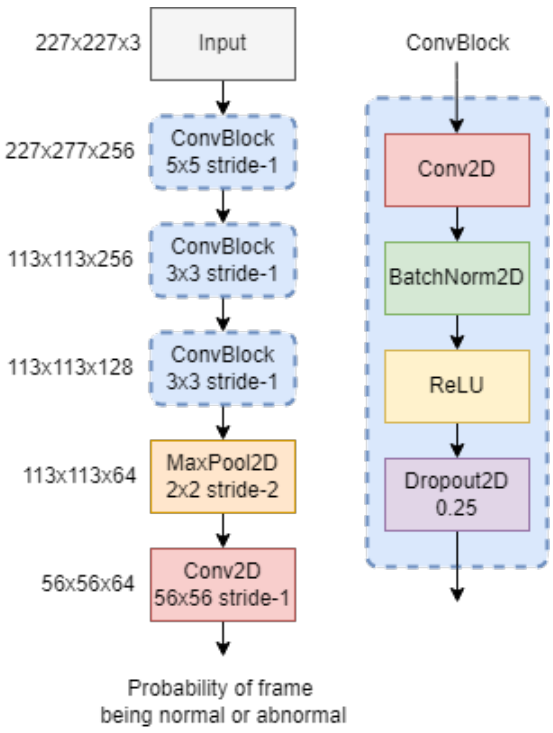


Figure 4.1: 2D CNN architecture for crowd anomaly detection

4.3.2 3D CNN

While the 2D CNN takes a single image as an input, the 3D CNN takes an entire video of frames as input. As a consequence of this, it requires a significant memory footprint and thus both the model size and image dimensions are reduced. To this end, with the UCSD dataset in mind where each video consists of 200 frames, the 3D CNN receives an input of size $164 \times 164 \times 3 \times 200$.

Similarly to that of the 2D CNN, it is comprised of two blocks of 3D convolutions with kernel sizes of $5 \times 5 \times 5$ and $3 \times 3 \times 3$ with a stride of 2 respectively. However, this model has a higher dropout probability of $p = 0.7$ and flattens the output of the second block, feeding the result into two dense layers of 32 neurons with a final dense layer of 200 neurons (the amount of frames for each UCSD video) at the end as illustrated in Figure 4.2. Typically for action recognition and other video classification tasks, there is a wide variety of labels each frame has a given probability of belonging to. The most commonly occurring label is then used as classification for the video as a whole. In this instance, given only two labels, each individual frame of a video has a probability between 0 and 1 to be abnormal. Thus, for the video as a whole, the Sigmoid activation function shown in Equation 4.2 is used together with binary cross entropy loss to determine the probability of each frame being abnormal.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.2}$$

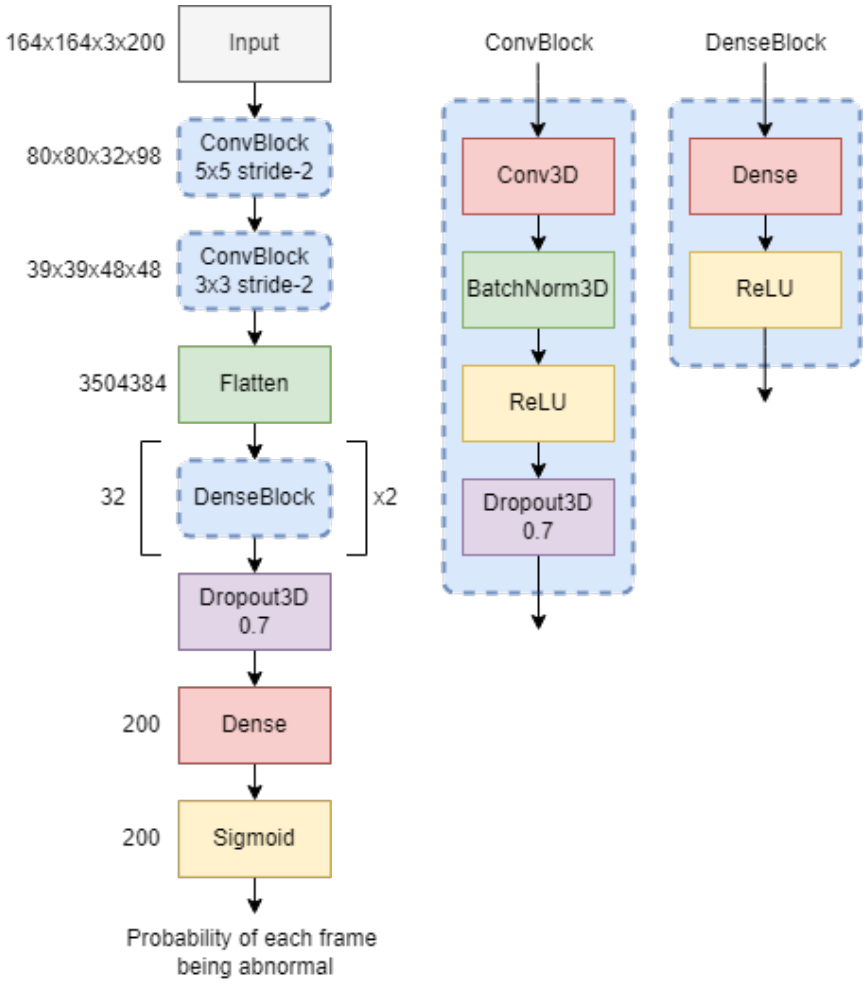


Figure 4.2: 3D CNN architecture for crowd anomaly detection

4.4 Convolutional Recurrent Neural Network

The CRNN model is comprised of a 2D CNN-based encoder and an RNN-based decoder as illustrated in Figure 4.3. In short, a sequence of images are passed to the CNN such that every 2D image $x(t)$ is compressed into a 1D vector $z(t)$. The RNN receives a sequence of input vectors $z(t)$ from the CNN encoder and outputs a final sequence $h(t)$ passed to a fully-connected network. To this end, the CNN extracts spatial features and the RNN extracts temporal features, forming a model able to capture the spatio-temporal nature of crowd anomalies. Moreover, The CRNN requires a smaller memory footprint when compared to that of the 3D CNN due to its ability to compress data.

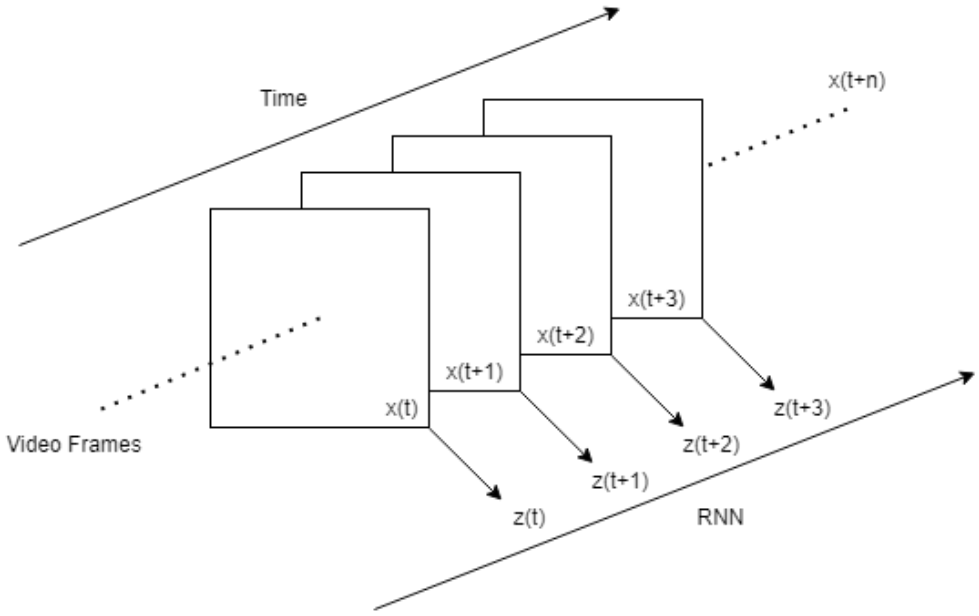


Figure 4.3: CRNN encoder-decoder overview

The CNN-based encoder is comprised of four convolutional blocks similar to that of the previous models. The output of the final convolutional block is flattened and passed to two dense layers. These are followed by a dropout with a probability of $p = 0.6$, and a final dense layer representing the latent space extracted by the 2D CNN. The RNN-based decoder is a simple LSTM-based architecture with three recurrent layers, i.e. three LSTMs are stacked wherein each takes the output of the previous LSTM. The hidden state vector is represented by a size of 512 neurons. The output of the final recurrent layer is followed by a dense layer with a ReLU activation and an additional dropout. A final dense layer with a Sigmoid activation function is used together with binary cross entropy loss as with the 3D CNN as illustrated in Figure 4.4.

A second CRNN model employs a pre-trained ResNet-152 [258] model using the ImageNet [259] dataset. To achieve this, the convolutional layers of the encoder are replaced by the learned weights of the ResNet-152 model.

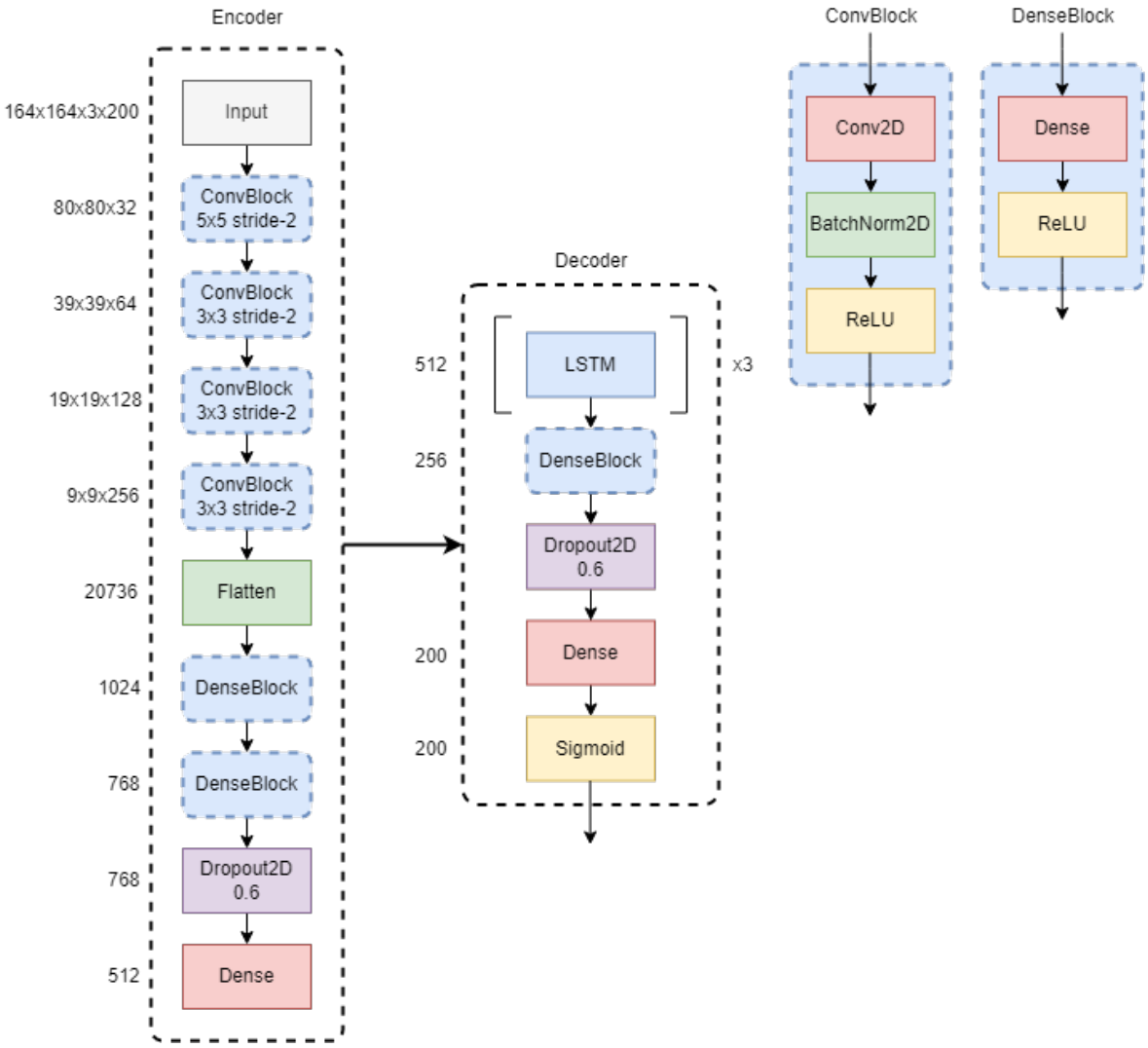


Figure 4.4: CRNN architecture for crowd anomaly detection

4.5 Autoencoder

The AE architecture follows the work by [260]. For a sequence of 8 consecutive images as input, an encoder-decoder architecture is comprised as follows:

1. A spatial encoder of two convolutional layers with the non-linear tanh activation function as shown in Equation 4.3.

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{4.3}$$

2. The encoded features of 8 consecutive frames are fed into a ConvLSTM-based temporal encoder-decoder with three recurrent layers. Each layer has a hidden state vector of 64, 32, and 64 neurons respectively, with an additional dropout of $p = 0.5$. In a convolution-based LSTM, the internal matrix operations are replaced by convolution operations.

3. The encoded representation by the final recurrent layer represents the compressed spatio-temporal latent space. As such, the temporal decoder mirrors the encoder to reconstruct the video volume, followed by a spatial decoder of two deconvolutional layers [261] using the tanh activation function. The final result as illustrated in Figure 4.5 is a spatio-temporal autoencoder that outputs a reconstruction of an input video sequence.

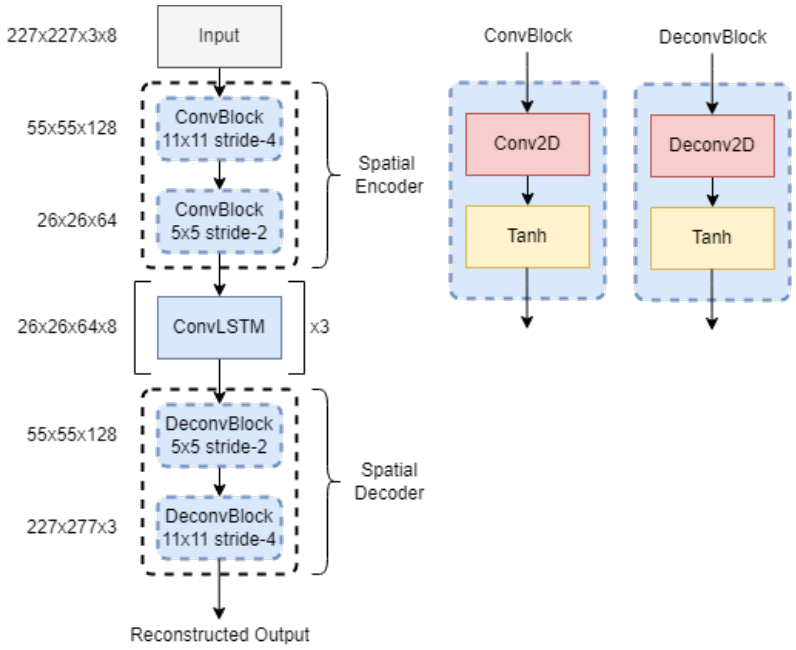


Figure 4.5: AE architecture for crowd anomaly detection

4.6 Data Preprocessing

The data pre-processing stage was among the largest undertakings for this thesis. During early stages of the project, several datasets comprised of either video or image files were used to evaluate optical flow performance. With that in mind, extensive and reusable tools were written such that different optical flow methods could be used to estimate either frame-level or video-level optical flow depending on the dataset. This section will detail the process and some of the challenges that came with it.

4.6.1 Optical Flow Estimation

The consideration of which deep optical flow estimation methods to use relied on one aspect: their placing on the MPI-Sintel benchmark. To perform an adequate assessment of the impact various optical flow methods cause, it was desirable to use the state-of-the-art of varying quality. GMA, RAFT, and LiteFlowNet3 are placed at 8th, 29th, and 84th respectively. The degree to which their quality differ is in terms of the estimated per-pixel features for matching moving objects, given the challenges posed in Subsection 2.2.2.

In practice, each of the aforementioned methods take two images as input, and output a respective optical flow map. As such, for a given directory of images, the optical flow of each two consecutive frames can broadly be estimated in a similar fashion $y = f(x_1, x_2)$ where x_1 is the first image, x_2 is the second image, and y is the output. With this in mind, a command line tool was made such that one can specify a path of images or videos to run optical flow inference of with any pre-trained model. Listing 1 shows an example of how to estimate the optical flow of a directory of images using GMA pre-trained on the MPI-Sintel dataset. Moreover, the save argument decides whether to save the output as images, a video, or a custom .flo format [262].

```
py process_video.py --path ./Datasets/UCSD/Test/Test024
--flow GMA --model sintel --save images
```

Listing 1: Command line tool for optical flow estimation

This tool is able to extract optical flow of either images or video files, and will automatically accustom to whether the path points to a single directory of files, or a directory with sub-directories of files. For the example above, each flow output of the model is saved as an image using an optical flow visualization technique based on [263]. Important to note is that LiteFlowNet3 only comes with a pre-trained model on the MPI-Sintel dataset, whereas RAFT and GMA include Flying Chairs and KITTI models as well. The examples below illustrate optical flow visualizations of an anomalous frame using each aforementioned method.

(a) RAFT (b) GMA

Figure 4.7: Optical flow pre-trained on the Flying Chairs dataset

Figure 4.6: An example anomalous frame from the UCSD dataset

(a) RAFT (b) GMA

Figure 4.8: Optical flow pre-trained on the KITTI dataset

(a) LiteFlowNet3 (b) RAFT (c) GMA

Figure 4.9: Optical flow pre-trained on the MPI-Sintel dataset

The downside of the UCSD dataset is that there are often very small motions, i.e. the low resolution of images and slow-walking pedestrians cause little to none flow vectors to be estimated. This problem is worsened by the flow visualization technique if motion is barely apparent for two given frames. As illustrated in Figure 4.10, insignificant motion causes the optical flow map to produce unreliable results, which if used for classification can cause major discrepancies in anomaly detection performance. To remedy this, [264] suggested to normalize the visualized flow using a fixed value denoted below as n instead of the maximum flow magnitude.

(a) Default (b) $n = 0.5$ (c) $n = 1.0$ (d) $n = 2.0$

Figure 4.10: Visualized optical flow map of insignificant motion at different normalization values

While this issue largely relies on the context and may cause deviation of magnitude (similar to example the example shown in Figure 4.10b where the intensity of the green color is high) in regular optical flow maps, a normalization value of $n = 2.0$ yielded the best results overall for further experiments. Based on the previous discoveries and results, LiteFlowNet3, RAFT, and GMA pre-trained on MPI-Sintel is the method in which optical flow is estimated for the remainder of this thesis.

4.6.2 PyTorch Datasets & DataLoaders

PyTorch provides two data primitives in the form of Datasets and DataLoaders. While a Dataset stores samples of data and their corresponding labels, a DataLoader wraps an iterable around the Dataset to enable easy access to the samples [265]. These are at the core of PyTorch's deep learning pipeline, and provide utility functions for ease of use. A Dataset represents a map from indices or keys to data samples with optional augmentations, whereas DataLoaders provide utility functions in the form of batching or shuffling the data, as well as enabling multiprocessing or hardware acceleration. Therefore, a model's performance heavily relies on the data scheme it is presented.

For the supervised methods discussed in the previous section, there are two approaches. First, the 2D CNN uses PyTorch's generic Dataset named ImageFolder [266]. Simply put, an ImageFolder Dataset is comprised of a number of directories acting as the label to their corresponding contents. To this end, the UCSD images are split in the pre-processing stage into an anomaly directory and a normal directory. Second, both the 3D CNN and CRNN models use the UCSD dataset as it is presented—each index represents a folder of images stacked together, with each image's corresponding label. The Dataset used for the unsupervised AE is derived from the work of [260]: the input to the model is in the shape of stacked image volumes, where each volume consists of 8 consecutive frames with various skipping strides.

4.6.3 Data Augmentation

Data augmentation is a technique used to either modify existing data, or artificially generate additional data samples from existing data. A problem with the UCSD dataset is that it is comprised of very few samples with a large class imbalance. Roughly one third of the images are classified as anomalies, and thus given its size has a large tendency to overfit. As a result, both dropout as discussed earlier, together with data augmentation, is key to help resolve class imbalance and reduce overfitting.

In PyTorch, this is a two-step process. Image transformations, called transforms in PyTorch [267], apply randomized transformations to all images of a given batch. In other words, for each transform provided, each image in a batch will receive the same transformations, while the next batch is applied new random transformations, and so on. As such, transforms do not artificially generate additional samples, and requires a manual process to introduce new samples. For each of the architectures described in the previous sections, the following data augmentation techniques are used.

First, all RGB images (i.e. optical flow) are normalized such that each channel has the same distribution by subtracting the mean from each pixel and dividing the result by a standard deviation. In most cases, pixels are defined by 8-bit integers such that their value is between 0 and 255 for all three channels. To make networks learn faster, it is desired to bound the values of each pixel such that they are in the same range as a network's activation functions. While results seem to vary across literature, it is common practice to normalize images using the ImageNet $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$ standards [268] for

each channel respectively that are based on millions of images. While optical flow maps are a particular case of special images, this normalization technique performed well. All grayscale images are normalized with a mean of 0.375 and standard deviation of 0.2.

Second, each supervised model introduces additional data samples in the form of augmented images of anomalies to counteract the class imbalance. More specifically, instead of experimenting with the result of different transformations by hand, an AutoAugment [269] policy is used on an additional subset of the UCSD dataset comprised of abnormal samples only. For the 2D CNN, these samples are additional, abnormal images. For the 3D CNN and RNN architectures based on a sequence of inputs, the introduced samples are additional test videos, i.e. videos that contain both normal and abnormal images. These subsets are concatenated with the original PyTorch Dataset through the use of PyTorch's ConcatDataset [270]. The transformations applied by the AutoAugment policy include, but are not limited to, rotation, sharpness, color variations, affine, and more.

Third, a data augmentation technique to generate additional samples for the unsupervised AE is derived from [260] following the practice of [241]. As previously mentioned, skipping strides are used for each video volume. As such, data augmentation is performed in the temporal dimension to increase the size of the training dataset. For example, the first stride sequence is made up of frame $\{1, 2, 3, 4, 5, 6, 7, 8\}$, the second stride sequence is made up of frame $\{1, 3, 5, 7, 9, 11, 13, 15\}$, the third stride would contain frame $\{1, 4, 7, 10, 13, 16, 19, 22\}$, and so on. No additional transformations are applied.

Chapter 5

Experiments & Results

Due to how each architecture described in the previous chapter are different, the experiments conducted vary in terms of hyperparameters and evaluation techniques. In total, 20 different models have been evaluated—each model trained using GMA, RAFT, and LiteFlowNet3 optical flow maps, in addition to regular images. As a general rule, each model used a standard split ratio of training and testing samples, except for the AE. Moreover, two videos in their entirety are excluding from the training and testing stage and used solely for validation of each final model, using samples the models have yet to observe. Finding the best performing split ratios have mostly relied on trial-and-error due to the nature of training with very few video samples in their entirety. Additionally, each model used the Adam [271] optimizer, a stochastic gradient descent-based method to update weights.

Each model regardless of whether it was trained and tested on optical flow maps or regular frames used the same hyperparameters, with the exception of epochs. As such, the following sections generalize each model of the same type unless stated otherwise. Furthermore, the results presented are based on weighted averages to account for class imbalance. Altogether, this chapter presents the experiments conducted and the results obtained by each model. A further discussion of the results overall is presented in the next chapter.

5.1 2D CNN

The 2D CNN architecture is an outlier as it relies only on single frames as input, in contrast to all the other architectures. Instead of splitting by videos, an 80:20 split ratio of all frames were used, i.e. 80% of frames are used for training, while 20% are used for testing. A learning rate of 0.05 was used, with a batch size of 16. Experiments showed that regular frames converged almost twice as fast than that of optical flow maps, i.e. 40 epochs and 100 epochs respectively. Likely, this is a result of training on grayscale images versus RGB images. An overview of the loss and accuracy across epochs, and a visualization of the results on each validation video, is shown in Appendix A. A noteworthy observation is that the model trained on regular frames has a test loss that is less than that of the training loss, whereas the models trained on optical flow maps show an indication of overfitting as the test loss grows larger than the training loss. As summarized in Table 5.1, the regular model far outperforms the models using optical flow maps. However, based on the results on the validation set as summarized in Table 5.2, it generalized poorly for unseen frames and showed a large tendency to classify normal images as abnormal. Surprisingly, even with overfitting in mind, LiteFlowNet3 performed best on both the test set and validation set in terms of optical flow methods, regardless of the noisy images generated. Overall, each method likely overfits and generalizes poorly, but optical flow performs better on the validation set.

	Test Set							
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
Regular	1941	763	9	7	0.99	0.99	0.99	0.99
LiteFlowNet3	1875	610	78	157	0.91	0.91	0.91	0.91
RAFT	1826	607	127	160	0.89	0.89	0.89	0.89
GMA	1820	520	133	247	0.86	0.86	0.86	0.86

Table 5.1: Summarization of the 2D CNN results on the test set

	Validation Set							
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
Regular	151	30	139	80	0.45	0.52	0.45	0.48
LiteFlowNet3	226	13	64	97	0.59	0.55	0.60	0.57
RAFT	196	14	94	96	0.52	0.52	0.53	0.52
GMA	221	10	69	100	0.57	0.53	0.58	0.55

Table 5.2: Summarization of the 2D CNN results on the validation set

5.2 3D CNN

The dataset used for the 3D CNN performed best when using a 90:10 split of the video samples, compared to 80:20. This model was trained using a learning rate of 0.005, with a batch size of 4 due to the memory constraints imposed by storing full video samples in memory. While higher epoch counts were experimented with, the model converged extremely fast compared to any other method, usually between 3 to 6 epochs. Beyond this point, the accuracy plateaued as the model started to overfit. To evaluate the performance on the test set, an ROC curve was calculated for each video, with the best performing threshold used to determine the final classifications. Shown in Table 5.3, each method performed similarly, with RAFT and GMA slightly ahead. Given that GMA is an extension of RAFT, similar results were to be expected. Interestingly, of optical flow methods, GMA performs best at classifying abnormal frames and worst at classifying regular frames. The opposite is true for LiteFlowNet3. The model may be picking up on distinct features in the GMA videos representing anomalies, whereas the LiteFlowNet3 videos include less distinct features due to the noisy frames such that anomalies are hard to pick up. In terms of the validation set summarized in Table 5.4, regardless of the worse performance on the test set, the model performs better than the 2D CNN. Clearly shown in the examples provided in Appendix B, the temporal feature extraction gives rise to temporal windows deemed anomalous, causing a significant decrease in false positives. Seemingly, the largest problem of each method used is the ability to recognize anomalous objects entering or leaving the frame. When considering the videos as a whole, the upper and lower regions are generally comprised of individuals leaving or entering the frame, and thus anomalous objects may be deemed indistinguishable from said individuals. This is less of an issue when the object in mind is clearly featured in the center. For the most part, each method performs similarly on the validation set.

	Test Set								
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score	AUC
Regular	356	580	511	153	0.58	0.62	0.60	0.57	0.62
LiteFlowNet3	396	540	553	111	0.58	0.66	0.58	0.57	0.67
RAFT	476	490	473	161	0.60	0.65	0.60	0.60	0.66
GMA	514	452	435	199	0.60	0.63	0.60	0.61	0.67

Table 5.3: Summarization of the 3D CNN results on the test set

	Validation Set							
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
Regular	94	195	16	95	0.72	0.81	0.72	0.74
LiteFlowNet3	92	173	18	117	0.66	0.78	0.66	0.68
RAFT	76	212	34	78	0.72	0.76	0.72	0.73
GMA	95	178	15	112	0.68	0.79	0.68	0.70

Table 5.4: Summarization of the 3D CNN results on the validation set

5.3 CRNN

Among the various methods explored in this thesis, the CRNN models proved most difficult to tune. Several variations of splits, augmentations, and network depths were attempted to overcome sporadic and hard to interpret results. Each of the CRNN models were trained across 200 epochs, and while the results showed an increase in test accuracy, a dramatic increase in loss as the model started to overfit was observed no matter the steps performed to alleviate said problem. Furthermore, there were seemingly random dips and rises in both loss and accuracy on the test set. To this end, the model proved counter-intuitive and thus the results presented in this section are not necessarily representative of other CRNN approaches, and more a product of uncertainty. The final results are based on an 80:20 split trained with a learning rate of 0.0001, and a batch size of 30.

Regardless of the problems stated above, the CRNN models show signs of temporal windows similar to that of the 3D CNN models, as visualized in Appendix C. While the results on the validation set are more scattered than their 3D CNN counterparts, this may be the attributing factor that causes the accuracy and AUC score of each CRNN model on the test set as summarized in Table 5.5 to be higher than the 3D CNN models. I.e., while the 3D CNN is able to capture small, but consistent temporal windows, the CRNN casts a wider net and thus classifies a few additional frames as abnormal when considering the test set as a whole. Furthermore, albeit minor, optical flow performs better on both the test set and the validation set. The latter, summarized in Table 5.6, shows that while able to predict abnormal frames well, a significant amount of abnormal frames are still classified as normal.

Test Set									
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score	AUC
Regular	1577	384	425	414	0.70	0.70	0.70	0.70	0.68
LiteFlowNet3	1592	376	410	422	0.70	0.70	0.70	0.70	0.67
RAFT	1881	245	121	553	0.75	0.74	0.76	0.73	0.67
GMA	1844	220	158	578	0.73	0.71	0.74	0.70	0.57

Table 5.5: Summarization of the CRNN results on the test set

Validation Set								
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
Regular	101	143	9	147	0.61	0.79	0.61	0.62
LiteFlowNet3	75	195	35	95	0.67	0.74	0.68	0.69
RAFT	84	170	26	120	0.63	0.74	0.63	0.65
GMA	100	144	10	146	0.61	0.79	0.61	0.62

Table 5.6: Summarization of the CRNN results on the validation set

5.4 CRNN-ResNet152

The CRNN model pre-trained on ResNet152 showed similar faults to the regular CRNN model as visualized in Appendix D. Equal hyperparameters to the regular CRNN model were used, and the best performing results were observed between epochs 40 and 50, approximately half the amount of the previous model. Regardless of overfitting, the results from the test set summarized in Table 5.7 show a minor increase in accuracy for each model, with LiteFlowNet3 classifying significantly less false positives than any other model. Furthermore, the model trained on regular frames has both the highest AUC on the test set and the highest accuracy on the validation set summarized in Table 5.8. Given that ResNet152 is trained on a large variety of normal images, it is reasonable to assume that this transfers better to regular frames in contrast to optical flow maps.

Test Set									
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score	AUC
Regular	1484	547	518	251	0.72	0.76	0.73	0.74	0.78
LiteFlowNet3	1924	384	78	414	0.82	0.83	0.82	0.81	0.61
RAFT	1874	272	128	526	0.76	0.75	0.77	0.74	0.65
GMA	1700	364	302	434	0.73	0.73	0.74	0.73	0.65

Table 5.7: Summarization of the CRNN-ResNet152 results on the test set

Validation Set								
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
Regular	97	168	13	122	0.66	0.79	0.66	0.68
LiteFlowNet3	35	207	75	83	0.60	0.61	0.60	0.61
RAFT	86	171	24	119	0.64	0.75	0.64	0.66
GMA	105	155	5	135	0.65	0.82	0.65	0.66

Table 5.8: Summarization of the CRNN-ResNet152 results on the validation set

5.5 Autoencoder

The AE models are different in that they are trained on normal videos, and tested on abnormal videos. Thus the results presented in this section are based on computing a RGS between an input and the reconstructed output, and based on a threshold, classify frames as normal or abnormal. First, each AE model is trained across 50 epochs with a learning rate of 0.0001, and a batch size of 64. Second, the RGS for each frame of a video is calculated. Based on the final score for each video, a threshold to determine what classifies an anomaly is computed by testing against a list of thresholds, and choosing the best performing one. As such, the result of each video is not determined by a static threshold, but instead video-dependent. However, this presents an interesting challenge as observed in Appendix E—if the RGS is unable to capture spatiotemporal features, and a video overall is comprised largely of anomalies, said threshold classifies the entire video as abnormal. An algorithm to smooth local minima or maxima to improve performance, Persistence 1D [272], has been applied by some works [273]. Experiments were conducted to test a large variation of smoothing parameters, but in this instance, none increased the overall accuracy. Nevertheless, the result of all test videos as summarized in Table 5.9 show impressive results, especially for normal frames. While this thesis started with a belief that the spatial features of optical flow maps would prove useful for reconstruction techniques, another likelihood is that not enough features are captured. For example, the previous chapter visualized an anomaly using each optical flow method. When considering regular frames, 11 pedestrians are visible and encoded by the AE, whereas optical flow maps featured at best 4 or less. As a result, there may not be enough information for a reconstructed output to contain significant errors signifying an anomaly. Lastly, while the results summarized in Table 5.10 are not based on unseen samples as the previous sections, they provide a view into the results on the same two videos for comparison's sake. Clearly, the AE outperforms every other method classifying a large quantity of abnormal frames, while barely classifying any false positives. For the most part, what separates the regular frames from optical flow maps is the amount of true negatives classified.

	Test Set								
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score	AUC
Regular	3156	2452	517	750	0.81	0.82	0.82	0.82	0.81
LiteFlowNet3	3173	2078	892	733	0.76	0.76	0.76	0.76	0.75
RAFT	3251	2096	874	655	0.77	0.78	0.78	0.78	0.77
GMA	3135	2312	658	771	0.79	0.79	0.79	0.79	0.79

Table 5.9: Summarization of the AE results on the test set

	Validation Set							
Frames	TP	TN	FP	FN	Accuracy	Precision	Recall	F1 Score
Regular	277	81	20	4	0.94	0.94	0.94	0.94
LiteFlowNet3	270	26	75	11	0.77	0.76	0.77	0.73
RAFT	271	33	68	10	0.80	0.79	0.80	0.76
GMA	273	40	61	8	0.81	0.82	0.82	0.80

Table 5.10: Summarization of the AE results on the validation set

5.6 Summary

While accuracy alone does not constitute the absolute effectiveness of the models detailed in the previous sections, the results summarized below provide a general comparison between each method. In summary, optical flow maps may improve upon regular frames by up to 10% during the testing stage, but is strongly dependent on method used. The largest difference was observed on the validation set of the 2D CNN, showing an increase of up to 14%. On average, both RAFT and GMA provide the best results, with RAFT surprisingly ahead of its successor. Based on the visualizations of each optical flow method provided in Subsection 4.6.1, it is very hard to interpret why the performance of each method fluctuates from different models, often from worst to best. One would imagine the optical flow maps with the most clear and distinct features would consistently be ahead, but this is not the case. However, it does show that when using deep optical flow methods, their performance compared to each other is insignificant enough that for certain tasks one may favor a method with a faster computation time with minimal loss to overall performance.

Additionally, the results are likely strongly tied to the UCSD dataset in particular. As observed in the aforementioned visualizations, the dimensions of the images and the degree of perceived motion drastically change the end result. For instance, pedestrians are often not discernible due to small motions. Essentially, this is a product of the frame rate at which the videos are recorded. Moreover, for datasets with larger images, the perceived motions are likely more discernible as the quantity of per-pixel movement increases.

	Test Set (Accuracy)				
Method	2D CNN	3D CNN	CRNN	CRNN-ResNet152	Autoencoder
Regular	**0.99**	0.58	0.70	0.72	**0.81**
LiteFlowNet3	0.91	0.58	0.70	**0.82**	0.76
RAFT	0.89	**0.60**	**0.75**	0.76	0.77
GMA	0.86	**0.60**	0.73	0.73	0.79

Table 5.11: Summarization of each model's performance on the test set

	Validation Set (Accuracy)				
Method	2D CNN	3D CNN	CRNN	CRNN-ResNet152	Autoencoder
Regular	0.45	**0.72**	0.61	**0.66**	**0.94**
LiteFlowNet3	**0.59**	0.66	**0.67**	0.60	0.76
RAFT	0.52	**0.72**	0.63	0.64	0.79
GMA	0.57	0.68	0.61	0.65	0.82

Table 5.12: Summarization of each model's performance on the validation set

Chapter 6

Discussion and Future Work

This thesis has explored several deep learning methods for the the crowd anomaly detection problem using a variety of normal frames and optical flow maps. Yet, the methods presented suffer a variety of drawbacks. While the focus was to conduct experiments to see whether optical flow maps would improve the performance of spatio-temporal networks without additional tuning, overfitting proved an insurmountable challenge. With a total of 20 models to train, requiring significant time investment, the vast array of variables involved for each model to improve generalization turned out of scope. For this reason, the results presented in this thesis may be improved by a number of techniques. First, k-fold cross validation as used by other works [274, 275, 276] may be essential in splitting the UCSD dataset to improve generalization. The largest issue observed during the course of this thesis was that data augmentation techniques for videos in particular is a challenging problem. When one subset of videos consists of one class, while a second subset consists of two classes in an unbalanced manner, image transformations alone are not viable. Introducing additional samples improved the results slightly, but not to the extent of eliminating overfitting. Early experiments looked at video frame sampling as presented in [277], but said method required extensive manual annotation and hindered the ability to capture spatio-temporal features by providing only small video segments based on random frames. Second, neither the CRNN model or the AE model were tested using GRU cells instead of LSTM cells. While not necessarily a factor that would have improved performance, it would have been interesting to experiment with to gauge their difference on videos. Third, as stated previously, there is a lack in literature exploring deep optical flow methods for the crowd anomaly detection problem—commonly, earlier works have relied on feature-based methods tracking sparse optical flow. Due to the already large amount of models to train, this was deemed out-of-scope, but it would have been intriguing to compare the results of such methods, e.g. Lucas-Kanade, to deep optical flow methods.

Another interesting area of current research is the use of vision transformers for image classification [278, 279]. Slowly emerging as a competitor to conventional CNNs, they show strong preservation of spatial information [280], which could be a promising aspect when considering the spatio-temporal nature of crowd anomalies. Interpretability and anonymization are additional aspects of utmost importance. The former is largely based on the ability to explain *what* an anomaly is or *why* something is deemed an anomaly, and is crucial for fair and ethical decision-making [281]. The latter seeks to preserve the privacy of individuals as regulations regarding artificial intelligence expands across countries. On one hand, the optical flow methods discussed in this thesis may arguably be considered an anonymization technique as the risk of identifying features of individuals are significantly low. On the other hand, the optical flow maps themselves are generated by deep learning methods—and thus may violate potential future regulations. Altogether, there is a wide diversity of areas that warrant further exploration as potential future work.

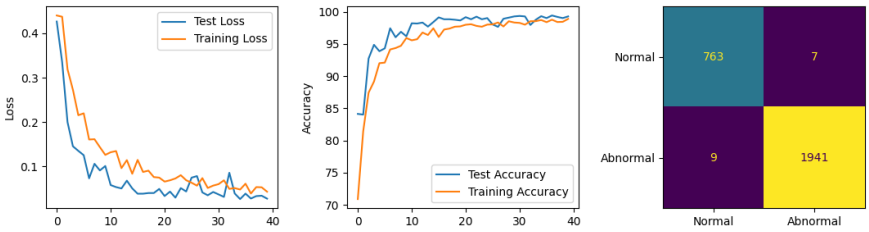
Chapter 7

Conclusion

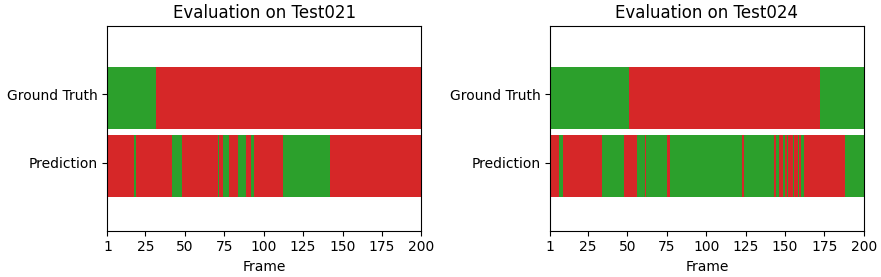
In conclusion, this thesis explores the use of dense optical flow maps estimated by deep learning-based techniques for the crowd anomaly detection problem. The theoretical foundations of both areas are outlined, and recent efforts toward fusing crowd anomaly detection with optical flow is reviewed. As a wide variety of models have been proposed for anomaly detection across literature, a total of five different architectures are investigated using both regular frames and optical flow maps. More specifically, a 2D convolutional network, a 3D convolutional network, an LSTM-based convolutional recurrent network, a pre-trained variant of the latter using ResNet152 trained on the ImageNet dataset, and a ConvLSTM-based autoencoder. Each architecture is trained using regular frames, and optical flow maps estimated by LiteFlowNet3, RAFT, and GMA. In total, 20 models are trained on the UCSD Pedestrian 1 dataset. While the results were prone to overfitting, they showed that optical flow maps may provide an increase in accuracy of up to 10% using spatio-temporal models. However, it is likely the results are heavily context-dependent on both the dataset and the architecture used, and thus there is no one-method-fits-all. Furthermore, based on overall performance across a test set and a validation set to account for generalization, an autoencoder approach proved both superior and easier to train, although the best performance was the result of regular frames. The 2D convolutional network showed exceptional results on the test set, but generalized very poorly when introduced to unseen data. To this end, spatio-temporal models fusing optical flow maps may prove more effective than conventional approaches using regular frames, but further experiments are required for more conclusive results.

Appendix A

2D CNN Results

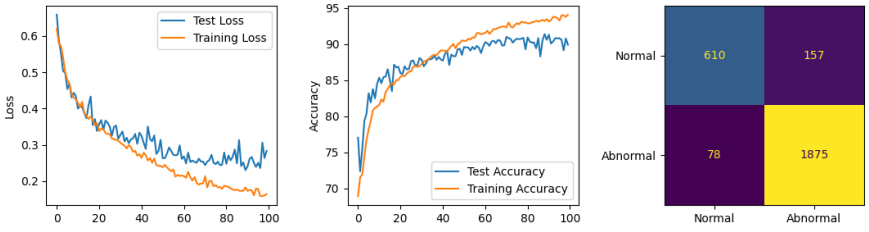


(a) Loss, accuracy, and confusion matrix

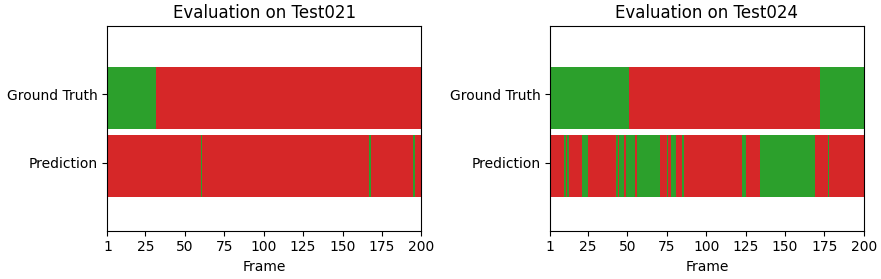


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure A.1: 2D CNN results using regular frames

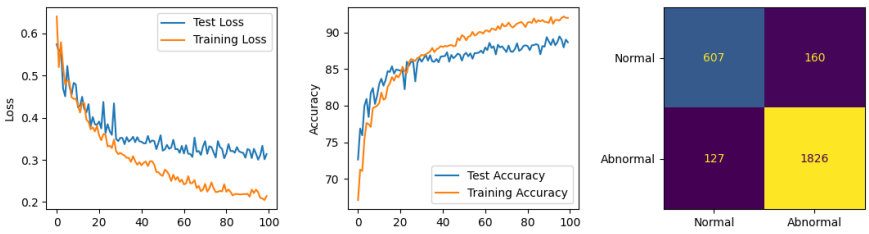


(a) Loss, accuracy, and confusion matrix

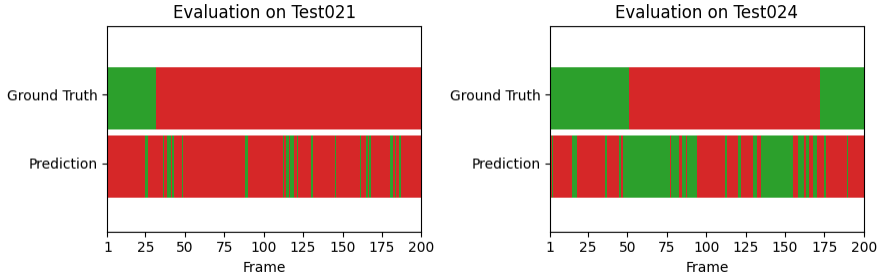


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure A.2: 2D CNN results using LiteFlowNet3 frames

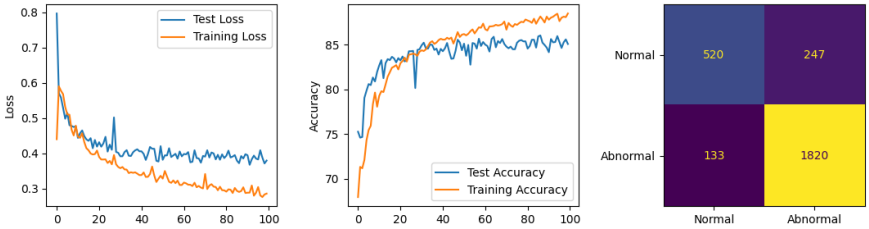


(a) Loss, accuracy, and confusion matrix

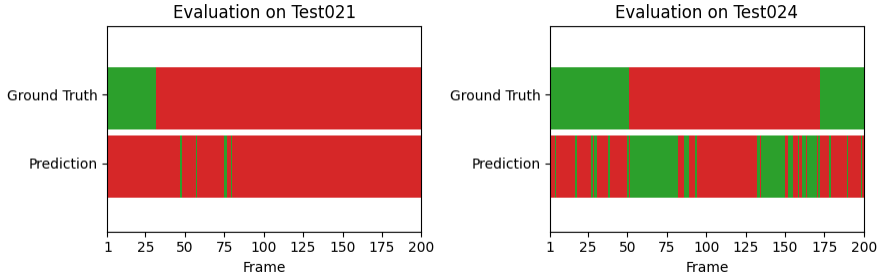


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure A.3: 2D CNN results using RAFT frames



(a) Loss, accuracy, and confusion matrix

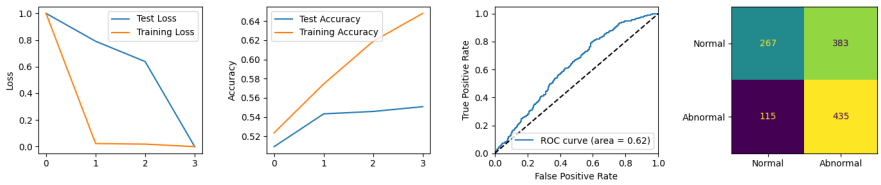


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

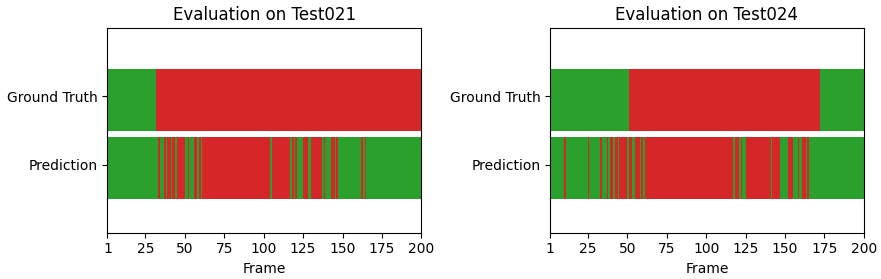
Figure A.4: 2D CNN results using GMA frames

Appendix B

3D CNN Results

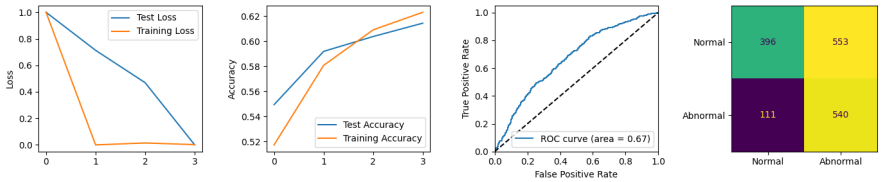


(a) Loss, accuracy, and confusion matrix

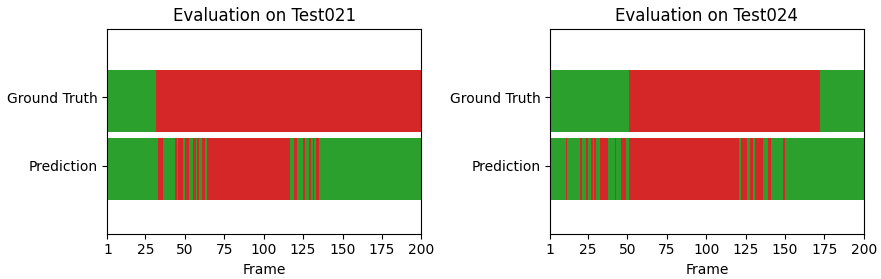


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure B.1: 3D CNN results using regular frames

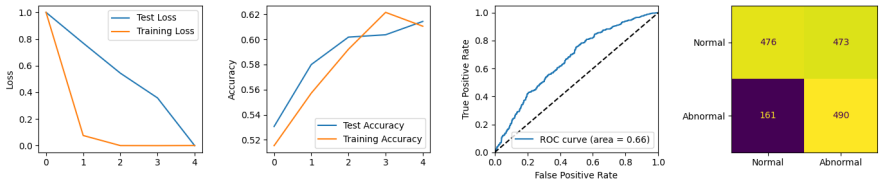


(a) Loss, accuracy, and confusion matrix

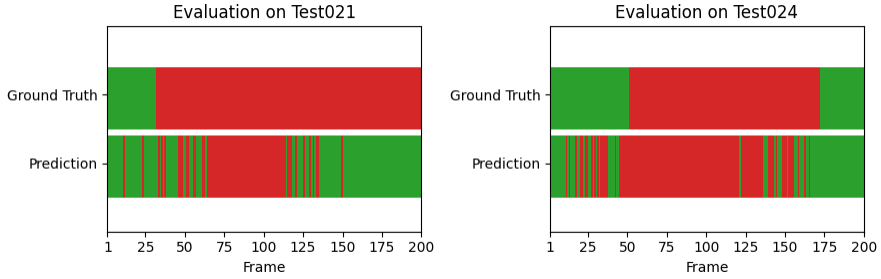


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure B.2: 3D CNN results using LiteFlowNet3 frames

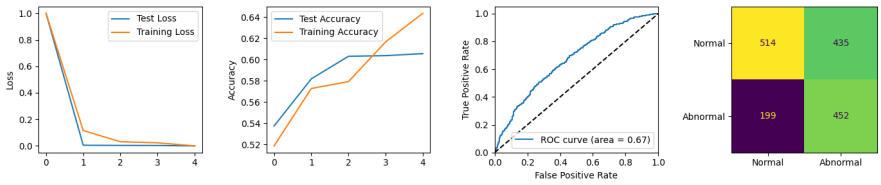


(a) Loss, accuracy, and confusion matrix

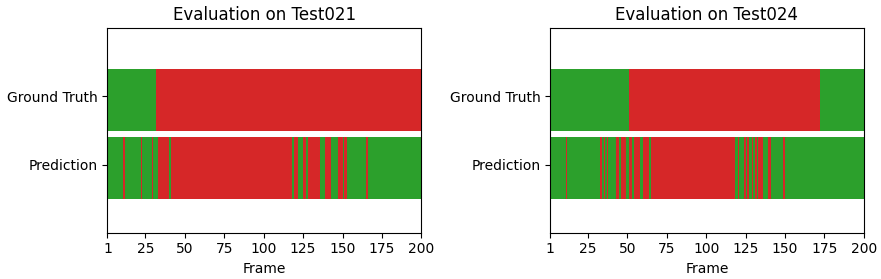


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure B.3: 3D CNN results using RAFT frames



(a) Loss, accuracy, and confusion matrix

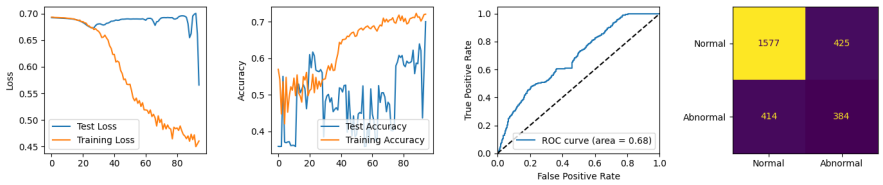


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

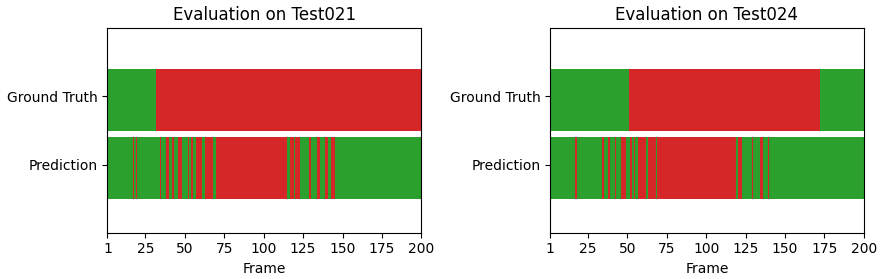
Figure B.4: 3D CNN results using GMA frames

Appendix C

CRNN Results

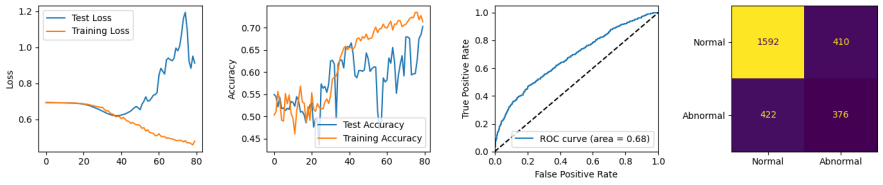


(a) Loss, accuracy, and confusion matrix

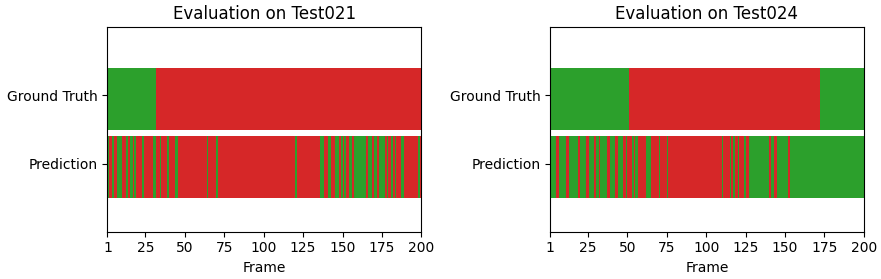


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure C.1: CRNN results using regular frames

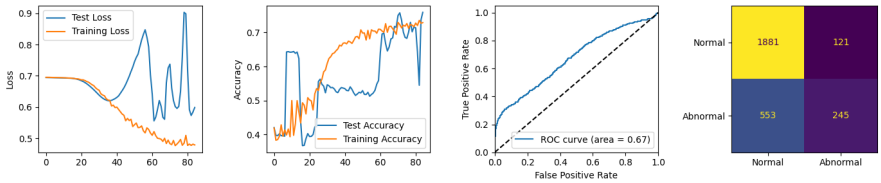


(a) Loss, accuracy, and confusion matrix

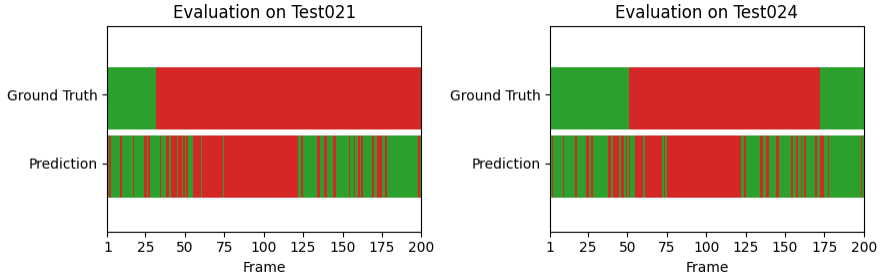


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure C.2: CRNN results using LiteFlowNet3 frames

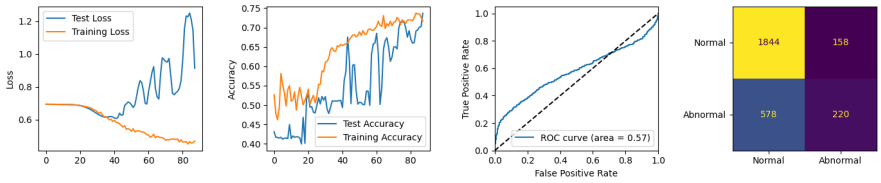


(a) Loss, accuracy, and confusion matrix

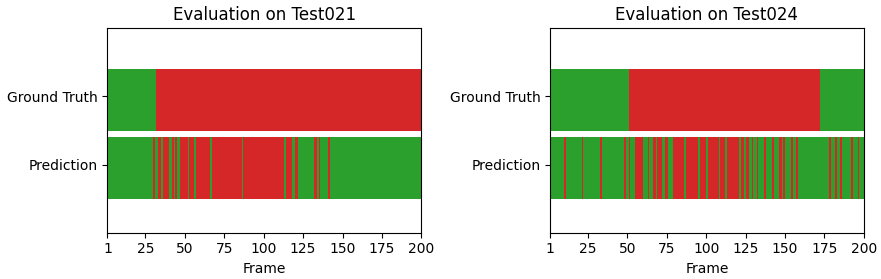


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure C.3: CRNN results using RAFT frames



(a) Loss, accuracy, and confusion matrix

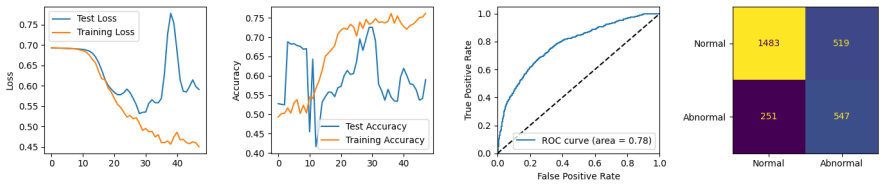


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

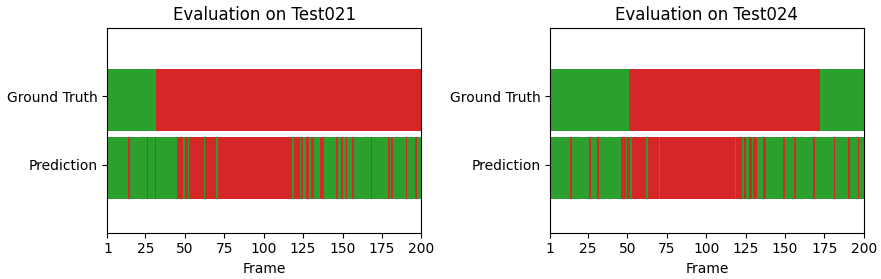
Figure C.4: CRNN results using GMA frames

Appendix D

CRNN-ResNet152 Results

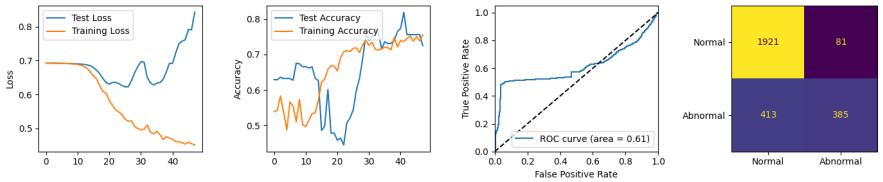


(a) Loss, accuracy, and confusion matrix

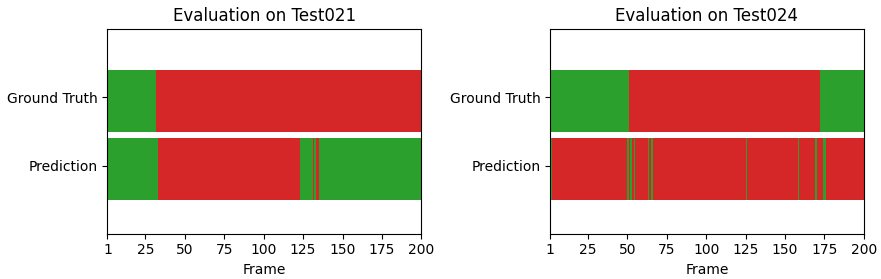


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure D.1: CRNN-ResNet152 results using regular frames

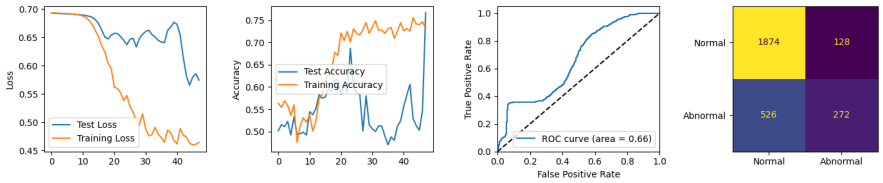


(a) Loss, accuracy, and confusion matrix

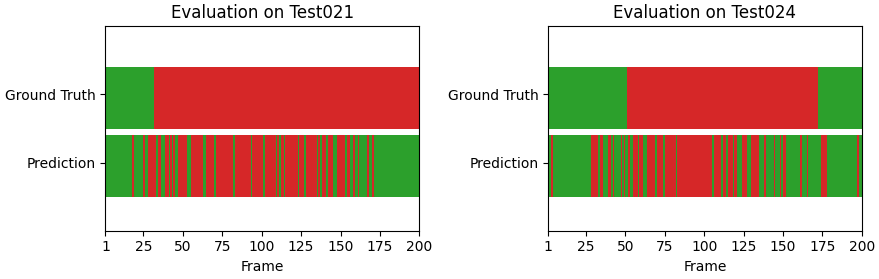


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure D.2: CRNN-ResNet152 results using LiteFlowNet3 frames

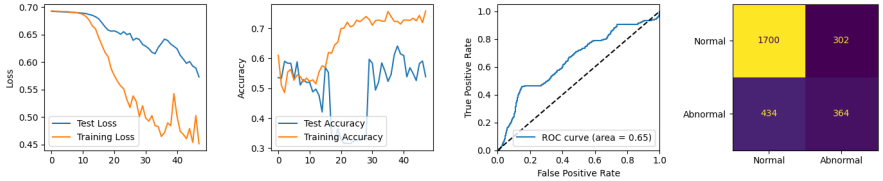


(a) Loss, accuracy, and confusion matrix

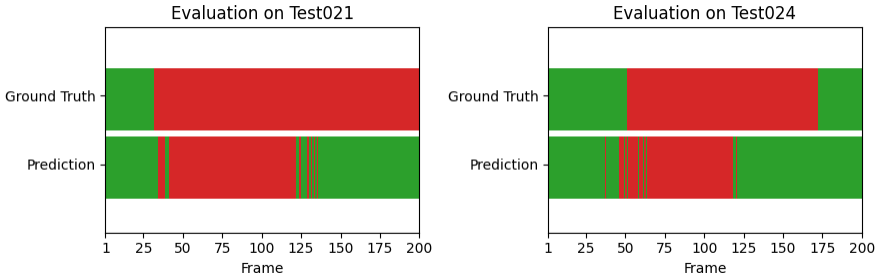


(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure D.3: CRNN-ResNet152 results using RAFT frames



(a) Loss, accuracy, and confusion matrix



(b) Validation on two unseen videos. Green indicates regular frames, red indicates abnormal frames.

Figure D.4: CRNN-ResNet152 results using GMA frames

Appendix E

Autoencoder Results

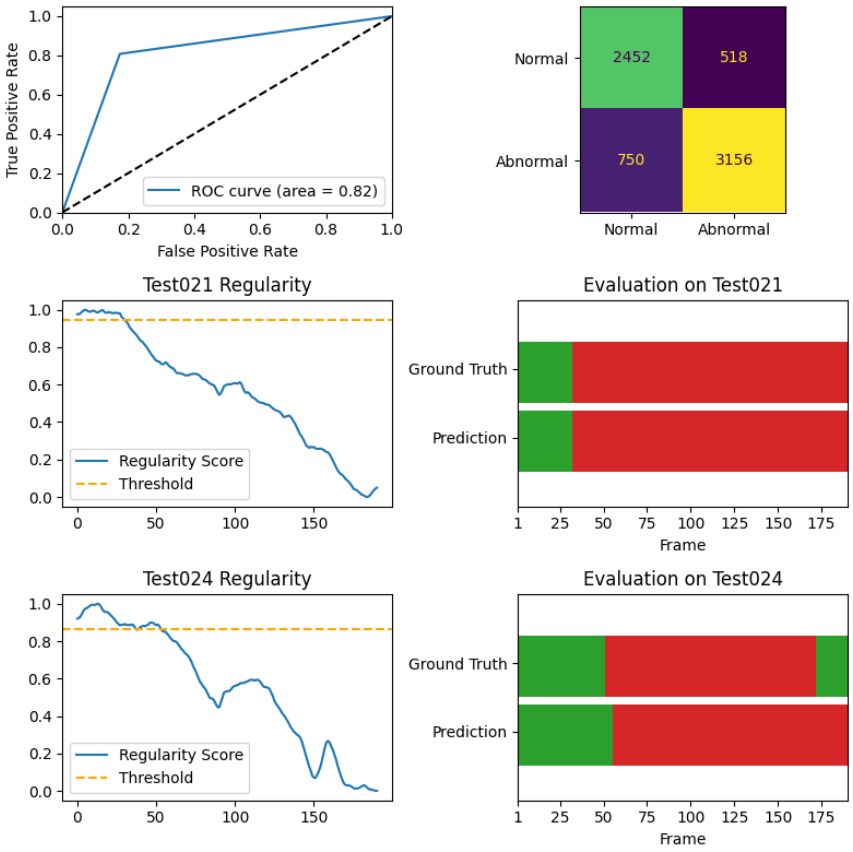


Figure E.1: Autoencoder results using regular frames

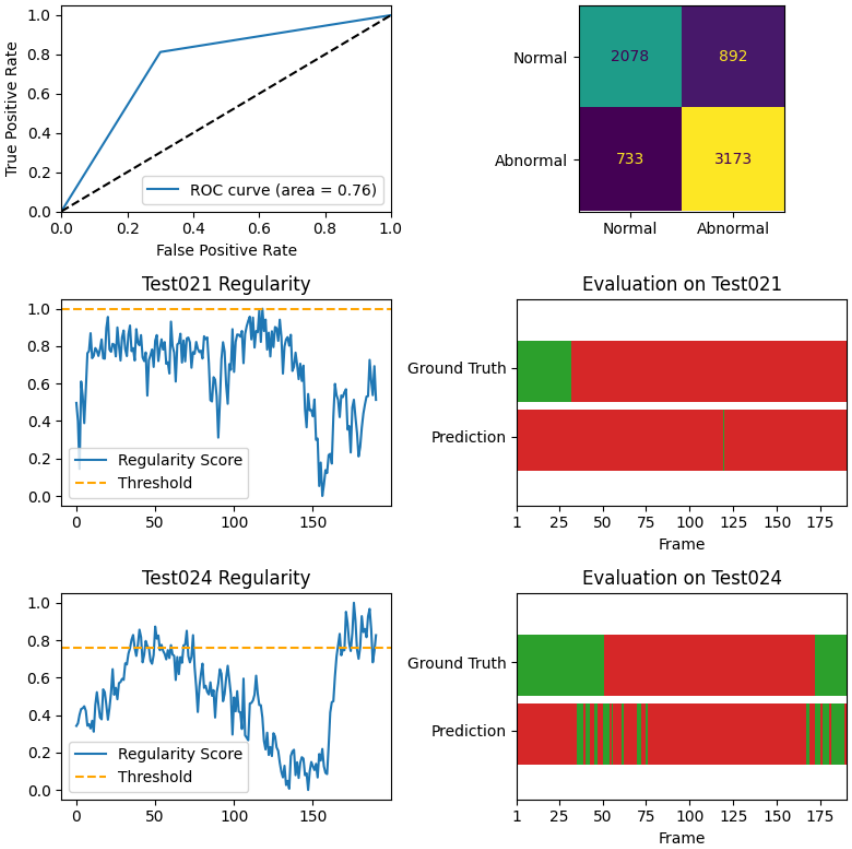


Figure E.2: Autoencoder results using LiteFlowNet3 frames

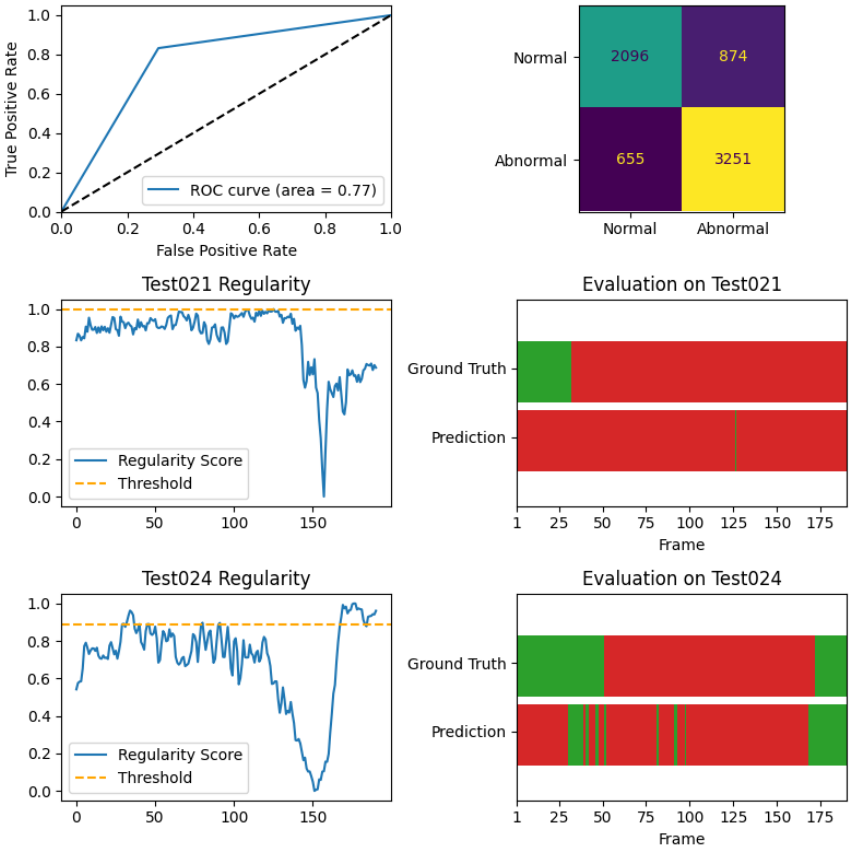


Figure E.3: Autoencoder results using RAFT frames

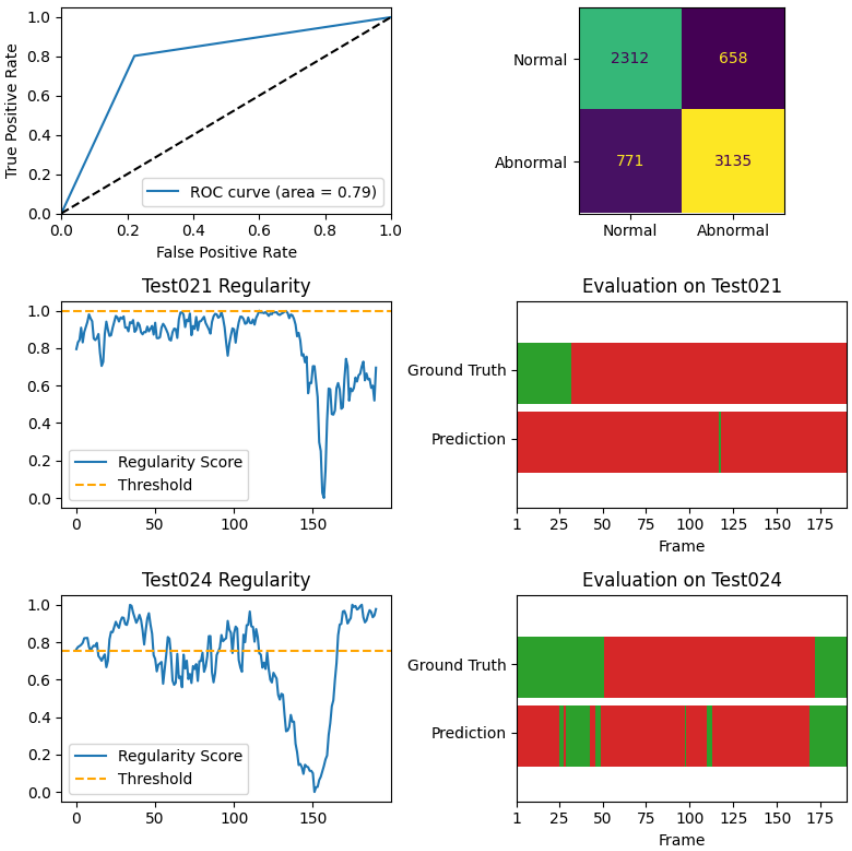


Figure E.4: Autoencoder results using GMA frames

Bibliography

[1] Liza Lin and Newley Purnell. *A World With a Billion Cameras Watching You Is Just Around the Corner*. 2019. URL: https://www.wsj.com/articles/a-billion-surveillance-cameras-forecast-to-be-watching-within-two-years-11575565402 (visited on 01/05/2021).

[2] Persistence Market Research. *CCTV Camera Market Outlook (2022 – 2032)*. 2022. URL: https://www.persistencemarketresearch.com/market-research/cctv-cameras-market.asp (visited on 01/05/2021).

[3] Roberto Olmos, Siham Tabik and Francisco Herrera. 'Automatic handgun detection alarm in videos using deep learning'. In: *Neurocomputing* 275 (2018), pp. 66–72. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.05.012. URL: https://www.sciencedirect.com/science/article/pii/S0925231217308196.

[4] Gaurav Tripathi, Kuldeep Singh and Dinesh Vishwakarma. 'Convolutional neural networks for crowd behaviour analysis: a survey'. In: *The Visual Computer* 35 (May 2019), pp. 1–24. DOI: 10.1007/s00371-018-1499-5.

[5] Mounir Bendali-Braham et al. 'Recent trends in crowd analysis: A review'. In: *Machine Learning with Applications* 4 (2021), p. 100023. ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2021.100023. URL: https://www.sciencedirect.com/science/article/pii/S2666827021000049.

[6] Francisco Luque Sánchez et al. 'Revisiting crowd behaviour analysis through deep learning: Taxonomy, anomaly detection, crowd emotions, datasets, opportunities and prospects'. In: *Information Fusion* 64 (2020), pp. 318–335. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2020.07.008. URL: https://www.sciencedirect.com/science/article/pii/S1566253520303201.

[7] Myo Thida et al. 'A Literature Review on Video Analytics of Crowded Scenes'. In: Nov. 2013, pp. 17–36. ISBN: 978-3-642-41511-1. DOI: 10.1007/978-3-642-41512-8_2.

[8] Kenneth Krogstad Aastveit and Odd Flakk. 'Crowd Anomaly Detection'. In: (2021).

[9] Elvan Duman and Osman Ayhan Erdem. 'Anomaly Detection in Videos Using Optical Flow and Convolutional Autoencoder'. In: *IEEE Access* 7 (2019), pp. 183914–183923. DOI: 10.1109/ACCESS.2019.2960654.

[10] Ryota Hinami, Tao Mei and Shin'ichi Satoh. *Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge*. 2017. DOI: 10.48550/ARXIV.1709.09121. URL: https://arxiv.org/abs/1709.09121.

[11] Abid Mehmood. 'Efficient Anomaly Detection in Crowd Videos Using Pre-Trained 2D Convolutional Neural Networks'. In: *IEEE Access* PP (Oct. 2021), pp. 1–1. DOI: 10.1109/ACCESS.2021.3118009.

[12] Meixue Yuan et al. 'A Systematic Survey on Human Behavior Recognition Methods'. In: *SN Computer Science* 3 (Jan. 2022). DOI: 10.1007/s42979-021-00932-x.

[13] Tiago Santana Nazaré, Rodrigo Fernandes de Mello and Moacir Antonelli Ponti. 'Are pre-trained CNNs good feature extractors for anomaly detection in surveillance videos?' In: *ArXiv* abs/1811.08495 (2018).

[14] Neeta Nemade and Vinaya Gohokar. 'Comparative Performance Analysis of Optical Flow Algorithms for Anomaly Detection'. In: *SSRN Electronic Journal* (Jan. 2019). DOI: 10.2139/ssrn.3419775.

[15] Limin Xia and Zhenmin Li. 'An abnormal event detection method based on the Riemannian manifold and LSTM network'. In: *Neurocomputing* 463 (2021), pp. 144–154. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.08.017. URL: https://www.sciencedirect.com/science/article/pii/S0925231221012017.

[16] Aniket Bera, Sujeong Kim and Dinesh Manocha. 'Realtime Anomaly Detection Using Trajectory-Level Crowd Behavior Learning'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2016, pp. 1289–1296. DOI: 10.1109/CVPRW.2016.163.

[17] Mingliang Zhai et al. 'Optical flow and scene flow estimation: A survey'. In: *Pattern Recognition* 114 (2021), p. 107861. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2021.107861. URL: https://www.sciencedirect.com/science/article/pii/S0031320321000480.

[18] Cem Direkoglu. 'Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks'. In: *IEEE Access* PP (Apr. 2020), pp. 1–1. DOI: 10.1109/ACCESS.2020.2990355.

[19] Hongyong Wang et al. 'Video Anomaly Detection By The Duality Of Normality-Granted Optical Flow'. In: *CoRR* abs/2105.04302 (2021). arXiv: 2105.04302. URL: https://arxiv.org/abs/2105.04302.

[20] Hongjing Zhang and Ian Davidson. 'Towards Fair Deep Anomaly Detection'. In: (Dec. 2020).

[21] Guansong Pang et al. 'Deep Learning for Anomaly Detection: A Review'. In: (July 2020).

[22] Utkarsh Singh et al. 'Crowd Monitoring: State-of-the-Art and Future Directions'. In: *IETE Technical Review* (Aug. 2020). DOI: 10.1080/02564602.2020.1803152.

[23] Kun Liu and Huadong Ma. 'Exploring Background-Bias for Anomaly Detection in Surveillance Videos'. In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM '19. Nice, France: Association for Computing Machinery, 2019, pp. 1490–1499. ISBN: 9781450368896. DOI: 10.1145/3343031.3350998. URL: https://doi.org/10.1145/3343031.3350998.

[24] Md Amran Siddiqui et al. *Sequential Feature Explanations for Anomaly Detection*. 2015. DOI: 10.48550/ARXIV.1503.00038. URL: https://arxiv.org/abs/1503.00038.

[25] Katsuhiro Honda et al. 'A study on fuzzy clustering-based k-anonymization for privacy preserving crowd movement analysis with face recognition'. In: *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*. 2015, pp. 37–41. DOI: 10.1109/SOCPAR.2015.7492779.

[26] Hidefumi Nishiyama. 'Crowd surveillance: The (in)securitization of the urban body'. In: *Security Dialogue* 49.3 (2018), pp. 200–216. DOI: 10.1177/0967010617741436.

[27] Romero Morais et al. *Learning Regularity in Skeleton Trajectories for Anomaly Detection in Videos*. 2019. DOI: 10.48550/ARXIV.1903.03295. URL: https://arxiv.org/abs/1903.03295.

[28] Warren S. McCulloch and Walter Pitts. 'A logical calculus of the ideas immanent in nervous activity'. In: *The bulletin of mathematical biophysics* (1943). DOI: 10.1007/BF02478259.

[29] Frank Rosenblatt. 'The perceptron: a probabilistic model for information storage and organization in the brain.' In: *Psychological review* 65 6 (1958), pp. 386–408.

[30] Martin T Hagan et al. *Neural Network Design (2nd Edition)*. English. Paperback. Martin Hagan, 1st Sept. 2014, p. 800. ISBN: 978-0971732117.

[31] Jaswinder Singh and Rajdeep Banerjee. 'A Study on Single and Multi-layer Perceptron Neural Network'. In: *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. 2019, pp. 35–40. DOI: 10.1109/ICCMC.2019.8819775.

[32] David E. Rumelhart, James L. McClelland and PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations (Volume 1)*. English. Hardcover. A Bradford Book, 17th July 1986, p. 567. ISBN: 978-0262181204.

[33] Iqbal Sarker. 'Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions'. In: *SN Computer Science* 2 (Nov. 2021). DOI: 10.1007/s42979-021-00815-1.

[34] Géron Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. English. Paperback. O'Reilly Media, 15th Oct. 2019, p. 856. ISBN: 978-1492032649.

[35] Kunihiko Fukushima. 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position'. In: *Biological Cybernetics* 36 (2004), pp. 193–202.

[36] Kunihiko Fukushima. 'Recent advances in the deep CNN neocognitron'. In: *Nonlinear Theory and Its Applications, IEICE* 10 (Jan. 2019), pp. 304–321. DOI: 10.1587/nolta.10.304.

[37] Yann Lecun et al. 'Gradient-Based Learning Applied to Document Recognition'. In: *Proceedings of the IEEE* 86 (Dec. 1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[38] Iqbal Sarker. 'Deep Cybersecurity: A Comprehensive Overview from Neural Network and Deep Learning Perspective'. In: *SN Computer Science* 2 (May 2021). DOI: 10.1007/s42979-021-00535-6.

[39] Stanford. *CS231n Convolutional Neural Networks for Visual Recognition*. 2022. URL: https://cs231n.github.io/convolutional-networks/ (visited on 04/05/2021).

[40] Shuiwang Ji et al. '3D Convolutional Neural Networks for Human Action Recognition'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 221–231. DOI: 10.1109/TPAMI.2012.59.

[41] Satya Singh et al. '3D Deep Learning on Medical Images: A Review'. In: (Mar. 2020).

[42] Luo Sha et al. 'An improved two-stream CNN method for abnormal behavior detection'. In: *Journal of Physics: Conference Series* 1617 (Aug. 2020), p. 012064. DOI: 10.1088/1742-6596/1617/1/012064.

[43] Michael I. Jordan. 'Chapter 25 - Serial Order: A Parallel Distributed Processing Approach'. In: *Neural-Network Models of Cognition*. Ed. by John W. Donahoe and Vivian Packard Dorsel. Vol. 121. Advances in Psychology. North-Holland, 1997, pp. 471–495. DOI: https://doi.org/10.1016/S0166-4115(97)80111-2. URL: https://www.sciencedirect.com/science/article/pii/S0166411597801112.

[44] Alex Sherstinsky. 'Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network'. In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2019.132306. URL: https://www.sciencedirect.com/science/article/pii/S0167278919305974.

[45] Sepp Hochreiter. 'The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions'. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6 (Apr. 1998), pp. 107–116. DOI: 10.1142/S0218488598000094.

[46] Y. Bengio, P. Simard and P. Frasconi. 'Learning long-term dependencies with gradient descent is difficult'. In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[47] Sepp Hochreiter and Jürgen Schmidhuber. 'Long Short-term Memory'. In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[48] Apple Inc. *Using Long Short-Term Memory Layers (LSTM)*. 2022. URL: https://developer.apple.com/documentation/accelerate/bnns/using_long_short-term_memory_layers_lstm (visited on 05/05/2021).

[49] Noah Weber. *Why LSTMs Stop Your Gradients From Vanishing: A View from the Backwards Pass*. 2017. URL: https://weberna.github.io/blog/2017/11/15/LSTM-Vanishing-Gradients.html (visited on 05/05/2021).

[50] Kyunghyun Cho et al. 'Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation'. In: *EMNLP*. 2014.

[51] Rahul Dey and Fathi Salem. 'Gate-variants of Gated Recurrent Unit (GRU) neural networks'. In: Aug. 2017, pp. 1597–1600. DOI: 10.1109/MWSCAS.2017.8053243.

[52] Chao Sun et al. 'A convolutional recurrent neural network with attention framework for speech separation in monaural recordings'. In: *Scientific Reports* 11 (Jan. 2021). DOI: 10.1038/s41598-020-80713-3.

[53] Mengjia Qiao et al. 'Crop yield prediction from multi-spectral, multi-temporal remotely sensed imagery using recurrent 3D convolutional neural networks'. In: *International Journal of Applied Earth Observation and Geoinformation* 102 (2021), p. 102436. ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2021.102436. URL: https://www.sciencedirect.com/science/article/pii/S0303243421001434.

[54] Naoki Asatani et al. 'Classification of respiratory sounds using improved convolutional recurrent neural network'. In: *Computers & Electrical Engineering* 94 (2021), p. 107367. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2021.107367. URL: https://www.sciencedirect.com/science/article/pii/S0045790621003372.

[55] Md. Zabirul Islam, Md. Milon Islam and Amanullah Asraf. 'A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images'. In: *Informatics in Medicine Unlocked* 20 (2020), p. 100412. ISSN: 2352-9148. DOI: https://doi.org/10.1016/j.imu.2020.100412. URL: https://www.sciencedirect.com/science/article/pii/S2352914820305621.

[56] Jiarui Zhang et al. 'LSTM-CNN Hybrid Model for Text Classification'. In: *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2018, pp. 1675–1680. DOI: 10.1109/IAEAC.2018.8577620.

[57] Pierre Baldi. 'Autoencoders, Unsupervised Learning, and Deep Architectures'. In: *ICML Unsupervised and Transfer Learning*. 2012.

[58] Meina Qiao et al. 'Abnormal event detection based on deep autoencoder fusing optical flow'. In: *2017 36th Chinese Control Conference (CCC)* (2017), pp. 11098–11103.

[59] Raghavendra Chalapathy and Sanjay Chawla. 'Deep Learning for Anomaly Detection: A Survey'. In: (Jan. 2019).

[60] Licheng Jiao and Jin Zhao. 'A Survey on the New Generation of Deep Learning in Image Processing'. In: *IEEE Access* PP (Nov. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2956508.

[61] Xiao-Jiao Mao, Chunhua Shen and Yu-Bin Yang. 'Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections'. In: (June 2016).

[62] Pavan A et al. 'LCA-Net: Light Convolutional Autoencoder for Image Dehazing'. In: *CoRR* abs/2008.10325 (2020). arXiv: 2008.10325. URL: https://arxiv.org/abs/2008.10325.

[63] Mengjia Yan et al. 'Detecting spatiotemporal irregularities in videos via a 3D convolutional autoencoder'. In: *Journal of Visual Communication and Image Representation* 67 (2020), p. 102747. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2019.102747. URL: https://www.sciencedirect.com/science/article/pii/S1047320319303682.

[64] Arnab Banerjee et al. 'Carp-DCAE: Deep convolutional autoencoder for carp fish classification'. In: *Computers and Electronics in Agriculture* 196 (2022), p. 106810. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2022.106810. URL: https://www.sciencedirect.com/science/article/pii/S0168169922001272.

[65] Nitish Srivastava, Elman Mansimov and Ruslan Salakhutdinov. 'Unsupervised Learning of Video Representations using LSTMs'. In: *CoRR* abs/1502.04681 (2015). arXiv: 1502.04681. URL: http://arxiv.org/abs/1502.04681.

[66] Sepehr Maleki, Sasan Maleki and Nicholas R. Jennings. 'Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering'. In: *Applied Soft Computing* 108 (2021), p. 107443. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2021.107443. URL: https://www.sciencedirect.com/science/article/pii/S1568494621003665.

[67] Evangelin Dasan and Ithayarani Panneerselvam. 'A novel dimensionality reduction approach for ECG signal via convolutional denoising autoencoder with LSTM'. In: *Biomedical Signal Processing and Control* 63 (2021), p. 102225. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2020.102225. URL: https://www.sciencedirect.com/science/article/pii/S1746809420303554.

[68] Tareq Tayeh et al. 'An Attention-based ConvLSTM Autoencoder with Dynamic Thresholding for Unsupervised Anomaly Detection in Multivariate Time Series'. In: *CoRR* abs/2201.09172 (2022). arXiv: 2201.09172. URL: https://arxiv.org/abs/2201.09172.

[69] Anitha Ramchandran and Arun Kumar. 'Unsupervised deep learning system for local anomaly event detection in crowded scenes'. In: *Multimedia Tools and Applications* 79 (Dec. 2020). DOI: 10.1007/s11042-019-7702-5.

[70] Rashmiranjan Nayak, Umesh Chandra Pati and Santos Kumar Das. 'Video Anomaly Detection using Convolutional Spatiotemporal Autoencoder'. In: *2020 International Conference on Contemporary Computing and Applications (IC3A)*. 2020, pp. 175–180. DOI: 10.1109/IC3A48958.2020.233292.

[71] Olaf Ronneberger, Philipp Fischer and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: https://arxiv.org/abs/1505.04597.

[72] Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1511.00561. URL: https://arxiv.org/abs/1511.00561.

[73] Hana Yousuf et al. 'A Systematic Review on Sequence to Sequence Neural Network and its Models'. In: *International Journal of Electrical and Computer Engineering* 11 (Oct. 2020). DOI: 10.11591/ijece.v11i3.pp2315-2326.

[74] OpenCV. *Optical Flow*. 2022. URL: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html (visited on 08/05/2021).

[75] David J. Fleet and Y. Weiss. 'Optical Flow Estimation'. In: *Handbook of Mathematical Models in Computer Vision*. 2006.

[76] Berthold K.P. Horn and Brian G. Schunck. 'Determining optical flow'. In: *Artificial Intelligence* 17.1 (1981), pp. 185–203. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(81)90024-2. URL: https://www.sciencedirect.com/science/article/pii/0004370281900242.

[77] Nixon Adu-Boahen, Joseph Panford and Joseph Panfordn. 'Optical Flow for Robot Navigation'. In: *International Journal of Computer Science and Information Security* 15 (Jan. 2018), pp. 229–237.

[78] Ka Man Lo. *Optical Flow Based Motion Detection for Autonomous Driving*. 2022. DOI: 10.48550/ARXIV.2203.11693. URL: https://arxiv.org/abs/2203.11693.

[79] Aytakin Nabisoy and Saber Malekzadeh. 'Video Action Recognition Using spatio-temporal optical flow video frames'. In: (Feb. 2021). DOI: 10.13140/RG.2.2.11802.36807.

[80] Lei Su et al. 'Transformer Vibration Detection Based on YOLOv4 and Optical Flow in Background of High Proportion of Renewable Energy Access'. In: *Frontiers in Energy Research* 10 (2022). ISSN: 2296-598X. DOI: 10.3389/fenrg.2022.764903. URL: https://www.frontiersin.org/article/10.3389/fenrg.2022.764903.

[81] Thomas Brox and Jitendra Malik. 'Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3 (2011), pp. 500–513. DOI: 10.1109/TPAMI.2010.143.

[82] Syed Shah and Xiang Xuezhi. 'Traditional and modern strategies for optical flow: an investigation'. In: *SN Applied Sciences* 3 (Mar. 2021). DOI: 10.1007/s42452-021-04227-x.

[83] Carlo Tomasi and Takeo Kanade. *Detection and Tracking of Point Features*. Tech. rep. International Journal of Computer Vision, 1991.

[84] Gunnar Farnebäck. 'Two-Frame Motion Estimation Based on Polynomial Expansion'. In: *Image Analysis*. Ed. by Josef Bigun and Tomas Gustavsson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370. ISBN: 978-3-540-45103-7.

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

[86] Denis Fortun, Patrick Bouthemy and Charles Kervrann. 'Optical flow modeling and computation: A survey'. In: *Computer Vision and Image Understanding* 134 (2015). Image Understanding for Real-world Distributed Video Networks, pp. 1–21. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2015.02.008. URL: https://www.sciencedirect.com/science/article/pii/S1077314215000429.

[87] Zhigang Tu et al. 'Variational method for joint optical flow estimation and edge-aware image restoration'. In: *Pattern Recognition* 65 (2017), pp. 11–25. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2016.10.027. URL: https://www.sciencedirect.com/science/article/pii/S0031320316303430.

[88] Andres Bruhn, Joachim Weickert and Christoph Schnörr. 'Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods'. In: *International Journal of Computer Vision* 61 (Feb. 2005), pp. 211–231. DOI: 10.1023/B:VISI.0000045324.43199.43.

[89] Andres Bruhn et al. 'Variational Optic Flow Computation in Real-Time'. In: (Jan. 2012).

[90] Andres Bruhn et al. 'Real-Time Optic Flow Computation with Variational Methods'. In: vol. 2756. Aug. 2003, pp. 222–229. ISBN: 978-3-540-40730-0. DOI: 10.1007/978-3-540-45179-2_28.

[91] G. Aubert, R. Deriche and Pierre Kornprobst. 'Computing Optical Flow Via Variational Techniques'. In: *SIAM Journal on Applied Mathematics* 60 (May 1999). DOI: 10.1137/S0036139998340170.

[92] Abhijit Patait. *An Introduction to the NVIDIA Optical Flow SDK*. 2019. URL: https://developer.nvidia.com/blog/an-introduction-to-the-nvidia-optical-flow-sdk/ (visited on 08/05/2021).

[93] Maxim Kuklin. *Optical Flow in OpenCV (C++/Python)*. 2021. URL: https://learnopencv.com/optical-flow-in-opencv/?ck_subscriber_id=371373457 (visited on 09/05/2021).

[94] Zhigang Tu et al. 'A combined post-filtering method to improve accuracy of variational optical flow estimation'. In: *Pattern Recognition* 47.5 (2014), pp. 1926–1940. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2013.11.026. URL: https://www.sciencedirect.com/science/article/pii/S0031320313005128.

[95] Zhuoyuan Chen et al. 'Large Displacement Optical Flow from Nearest Neighbor Fields'. In: June 2013, pp. 2443–2450. DOI: 10.1109/CVPR.2013.316.

[96] Wenzhe Chen et al. 'A Dense Optical Flow-Based Feature Matching Approach in Visual Odometry'. In: Aug. 2017, pp. 343–348. DOI: 10.1109/IHMSC.2017.189.

[97] Wenzhe Chen et al. 'A Dense Optical Flow-Based Feature Matching Approach in Visual Odometry'. In: *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. Vol. 2. 2017, pp. 343–348. DOI: 10.1109/IHMSC.2017.189.

[98] Gabriel Eilertsen, Per-Erik Forssén and Jonas Unger. 'BriefMatch: Dense Binary Feature Matching for Real-Time Optical Flow Estimation'. In: May 2017, pp. 221–233. ISBN: 978-3-319-59125-4. DOI: 10.1007/978-3-319-59126-1_19.

[99] Christopher G. Harris and M. J. Stephens. 'A Combined Corner and Edge Detector'. In: *Alvey Vision Conference*. 1988.

[100] John Canny. 'A Computational Approach to Edge Detection'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.

[101] G LoweDavid. 'Distinctive Image Features from Scale-Invariant Keypoints'. In: *International Journal of Computer Vision* (2004).

[102] Herbert Bay et al. 'Speeded-Up Robust Features (SURF)'. In: *Computer Vision and Image Understanding* 110.3 (2008). Similarity Matching in Computer Vision and Multimedia, pp. 346–359. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2007.09.014. URL: https://www.sciencedirect.com/science/article/pii/S1077314207001555.

[103] Navid Nourani-Vatani, Paulo Borges and Jonathan Roberts. 'A study of feature extraction algorithms for optical flow tracking'. In: Jan. 2012.

[104] Ebrahim Karami, Siva Prasad and Mohamed Shehata. 'Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images'. In: Nov. 2015.

[105] Pengju Zhang, Yihong Wu and Bingxi Liu. *Leveraging Local and Global Descriptors in Parallel to Search Correspondences for Visual Localization*. 2020. DOI: 10.48550/ARXIV.2009.10891. URL: https://arxiv.org/abs/2009.10891.

[106] Connelly Barnes et al. 'The Generalized PatchMatch Correspondence Algorithm'. In: Sept. 2010, pp. 29–43. ISBN: 978-3-642-15557-4. DOI: 10.1007/978-3-642-15558-1_3.

[107] Tian Wang and Hichem Snoussi. 'Histograms of Optical Flow Orientation for Visual Abnormal Events Detection'. In: *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*. 2012, pp. 13–18. DOI: 10.1109/AVSS.2012.39.

[108] Mahmoud Mohamed. 'Illumination Robust Optical Flow Model Based on Histogram of Oriented Gradients'. In: Sept. 2013.

[109] Virgínia Mota. 'A tensor motion descriptor based on histogram of gradients and optical flow'. In: *Pattern Recognition Letters* (Aug. 2013).

[110] Tian Chang, Fei Long and Jianming Huang. 'Micro-Expression Recognition Using Optical Flow and Local Binary Patterns on Three Orthogonal Planes'. In: *Proceedings of the Seventh International Symposium of Chinese CHI*. Chinese CHI '19. Xiamen, China: Association for Computing Machinery, 2019, pp. 44–48. ISBN: 9781450372473. DOI: 10.1145/3332169.3333575. URL: https://doi.org/10.1145/3332169.3333575.

[111] Fawad Hussain, Farrah Aslam and Muhammad Haroon Yousaf. 'HUMAN ACTIVITY BASED VIDEO RETRIEVAL USING OPTICAL FLOW AND LOCAL BINARY PATTERNS'. In: (Oct. 2018).

[112] Paul-Edouard Sarlin et al. 'From Coarse to Fine: Robust Hierarchical Localization at Large Scale'. In: *CVPR*. 2019.

[113] Paul-Edouard Sarlin et al. 'SuperGlue: Learning Feature Matching with Graph Neural Networks'. In: *CVPR*. 2020.

[114] Philippe Weinzaepfel et al. 'DeepFlow: Large Displacement Optical Flow with Deep Matching'. In: Dec. 2013, pp. 1385–1392. DOI: 10.1109/ICCV.2013.175.

[115] Philipp Fischer et al. 'FlowNet: Learning Optical Flow with Convolutional Networks'. In: (Apr. 2015).

[116] Max Planck Institute for Intelligent Systems. *Results and Rankings*. 2022. URL: http://sintel.is.tue.mpg.de/results (visited on 10/05/2021).

[117] D. J. Butler et al. 'A naturalistic open source movie for optical flow evaluation'. In: *European Conf. on Computer Vision (ECCV)*. Ed. by A. Fitzgibbon et al. (Eds.) Part IV, LNCS 7577. Springer-Verlag, Oct. 2012, pp. 611–625.

[118] Gregory Schröder et al. *Optical Flow Dataset and Benchmark for Visual Crowd Analysis*. 2018. DOI: 10.48550/ARXIV.1811.07170. URL: https://arxiv.org/abs/1811.07170.

[119] Maria Andersson et al. 'Recognition of Anomalous Motion Patterns in Urban Surveillance'. In: *IEEE Journal of Selected Topics in Signal Processing* 7.1 (2013), pp. 102–110. DOI: 10.1109/JSTSP.2013.2237882.

[120] Yutao Han, Rina Tse and Mark Campbell. 'Pedestrian Motion Model Using Non-Parametric Trajectory Clustering and Discrete Transition Points'. In: *IEEE Robotics and Automation Letters* 4.3 (July 2019), pp. 2614–2621. DOI: 10.1109/lra.2019.2898464. URL: https://doi.org/10.1109%5C%2Flra.2019.2898464.

[121] Li Li and Christopher Leckie. 'Trajectory Pattern Identification and Anomaly Detection of Pedestrian Flows Based on Visual Clustering'. In: Nov. 2016, pp. 121–131. ISBN: 978-3-319-48389-4. DOI: 10.1007/978-3-319-48390-0_13.

[122] Habib Ullah et al. 'Anomalous entities detection and localization in pedestrian flows'. In: *Neurocomputing* 290 (2018), pp. 74–86. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.02.045. URL: https://www.sciencedirect.com/science/article/pii/S0925231218301802.

[123] Radu Tudor Ionescu et al. 'Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video'. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 7834–7843.

[124] Soma Biswas and Vikas Gupta. 'Abnormality Detection in Crowd Videos by Tracking Sparse Components'. In: *Mach. Vision Appl.* 28.1–2 (Feb. 2017), pp. 35–48. ISSN: 0932-8092. DOI: 10.1007/s00138-016-0800-8. URL: https://doi.org/10.1007/s00138-016-0800-8.

[125] Muhammad Umar Karim Khan, Hyun-Sang Park and Chong-Min Kyung. 'Rejecting Motion Outliers for Efficient Crowd Anomaly Detection'. In: *IEEE Transactions on Information Forensics and Security* 14 (July 2018), pp. 1–1. DOI: 10.1109/TIFS.2018.2856189.

[126] Joey Tianyi Zhou et al. 'AnomalyNet: An Anomaly Detection Network for Video Surveillance'. In: *IEEE Transactions on Information Forensics and Security* 14.10 (2019), pp. 2537–2550. DOI: 10.1109/TIFS.2019.2900907.

[127] Kuldeep Singh et al. 'Crowd anomaly detection using Aggregation of Ensembles of fine-tuned ConvNets'. In: *Neurocomputing* 371 (2020), pp. 188–198. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.08.059. URL: https://www.sciencedirect.com/science/article/pii/S092523121931197X.

[128] Yanhua Shao et al. 'A Multitask Cascading CNN with MultiScale Infrared Optical Flow Feature Fusion-Based Abnormal Crowd Behavior Monitoring UAV'. In: *Sensors* 20.19 (2020). ISSN: 1424-8220. DOI: 10.3390/s20195550. URL: https://www.mdpi.com/1424-8220/20/19/5550.

[129] Wei Lin et al. 'Learning to detect anomaly events in crowd scenes from synthetic data'. In: *Neurocomputing* 436 (2021), pp. 248–259. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.01.031. URL: https://www.sciencedirect.com/science/article/pii/S0925231221000527.

[130] Bolei Zhou, Xiaoou Tang and Xiaogang Wang. 'Measuring Crowd Collectiveness'. In: *CVPR*. 2013.

[131] Xuguang Zhang et al. 'Energy Level-Based Abnormal Crowd Behavior Detection'. In: *Sensors* 18 (Feb. 2018), p. 423. DOI: 10.3390/s18020423.

[132] Vagia Kaltsa et al. 'Swarm - based Motion Features for Anomaly Detection in Crowds'. In: Oct. 2014. DOI: 10.1109/ICIP.2014.7025477.

[133] Habib Ullah, Mohib Ullah and Nicola Conci. 'Real-time anomaly detection in dense crowded scenes'. In: *Proceedings of SPIE - The International Society for Optical Engineering* 9026 (Feb. 2014). DOI: 10.1117/12.2040521.

[134] Ayse Elvan Gunduz et al. 'Density Aware Anomaly Detection in Crowded Scenes'. In: *IET Computer Vision* 10 (Feb. 2016). DOI: 10.1049/iet-cvi.2015.0345.

[135] Cong Zhang et al. 'Data-Driven Crowd Understanding: A Baseline for a Large-Scale Crowd Dataset'. In: *IEEE Transactions on Multimedia* 18.6 (2016), pp. 1048–1061. DOI: 10.1109/TMM.2016.2542585.

[136] Xuguang Zhang et al. 'Crowd emotion evaluation based on fuzzy inference of arousal and valence'. In: *Neurocomputing* 445 (2021), pp. 194–205. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.02.047. URL: https://www.sciencedirect.com/science/article/pii/S0925231221003015.

[137] Rakshenda Javed et al. 'Direction, Velocity, Merging Probabilities and Shape Descriptors for Crowd Behavior Analysis'. In: *IEEE Access* PP (July 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2929242.

[138] Jing Shao, Chen Change Loy and Xiaogang Wang. 'Scene-Independent Group Profiling in Crowd'. In: June 2014. DOI: 10.1109/CVPR.2014.285.

[139] Jing Shao, Chen Change Loy and Xiaogang Wang. 'Learning Scene-Independent Group Descriptors for Crowd Understanding'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27 (Jan. 2016), pp. 1–1. DOI: 10.1109/TCSVT.2016.2539878.

[140] Jing Shao, Chen Change Loy and Xiaogang Wang. 'Learning Scene-Independent Group Descriptors for Crowd Understanding'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.6 (2017), pp. 1290–1303. DOI: 10.1109/TCSVT.2016.2539878.

[141] Yuanping Xu et al. 'Towards Intelligent Crowd Behavior Understanding Through the STFD Descriptor Exploration'. In: *Sensing and Imaging* 19 (Apr. 2018). DOI: 10.1007/s11220-018-0201-3.

[142] Saad Ali and Mubarak Shah. 'A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis'. In: July 2007, pp. 1–6. ISBN: 1-4244-1180-7. DOI: 10.1109/CVPR.2007.382977.

[143] Teng Li et al. 'Crowded Scene Analysis: A Survey'. In: *IEEE Transactions on Circuits and Systems for Video Technology* X (Feb. 2015). DOI: 10.1109/TCSVT.2014.2358029.

[144] Allam S. Hassanein et al. 'Identifying motion pathways in highly crowded scenes: A non-parametric tracklet clustering approach'. In: *Computer Vision and Image Understanding* 191 (2020), p. 102710. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2018.08.004. URL: https://www.sciencedirect.com/science/article/pii/S1077314218301887.

[145] Simon Denman et al. 'Automatic surveillance in transportation hubs: No longer just about catching the bad guy'. In: *Expert Systems with Applications* 42.24 (2015), pp. 9449–9467. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0957417415005370.

[146] Mohammad Sabokrou et al. 'Real-Time Anomaly Detection and Localization in Crowded Scenes'. In: June 2015.

[147] Roberto Leyva, Victor Sanchez and Chang-Tsun Li. 'Video Anomaly Detection With Compact Feature Sets for Online Performance'. In: *IEEE Transactions on Image Processing* PP (Apr. 2017), pp. 1–1. DOI: 10.1109/TIP.2017.2695105.

[148] Jongmin Yu, Jeonghwan Gwak and Moongu Jeon. 'Gaussian-Poisson mixture model for anomaly detection of crowd behaviour'. In: Oct. 2016, pp. 106–111. DOI: 10.1109/ICCAIS.2016.7822444.

[149] Ayse Elvan Gunduz et al. 'Density aware anomaly detection in crowded scenes'. In: *IET Comput. Vis.* 10 (2016), pp. 374–381.

[150] Yibin Wang et al. 'A GM-HMM based abnormal pedestrian behavior detection method'. In: *2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. 2015, pp. 1–6. DOI: 10.1109/ICSPCC.2015.7338935.

[151] Shifu Zhou et al. 'Unusual event detection in crowded scenes by trajectory analysis'. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 1300–1304. DOI: 10.1109/ICASSP.2015.7178180.

[152] Yuan Yuan, Yachuang Feng and Xiaoqiang Lu. 'Statistical Hypothesis Detector for Abnormal Event Detection in Crowded Scenes'. In: *IEEE Transactions on Cybernetics* 47.11 (2017), pp. 3597–3608. DOI: 10.1109/TCYB.2016.2572609.

[153] Rensso Victor Hugo Mora Colque et al. 'Histograms of Optical Flow Orientation and Magnitude and Entropy to Detect Anomalous Events in Videos'. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.3 (2017), pp. 673–682. DOI: 10.1109/TCSVT.2016.2637778.

[154] Ang Li et al. 'Histogram of Maximal Optical Flow Projection for Abnormal Events Detection in Crowded Scenes'. In: *International Journal of Distributed Sensor Networks* 11 (2015).

[155] Hossein Mousavi et al. 'Analyzing Tracklets for the Detection of Abnormal Crowd Behavior'. In: *2015 IEEE Winter Conference on Applications of Computer Vision*. 2015, pp. 148–155. DOI: 10.1109/WACV.2015.27.

[156] Dinesh Singh and Krishna Mohan Chalavadi. 'Graph formulation of video activities for abnormal activity recognition'. In: *Pattern Recognition* 65 (May 2017), pp. 265–272. DOI: 10.1016/j.patcog.2017.01.001.

[157] Rima Chaker, Zaher Al Aghbari and Imran Junejo. 'Social Network Model for Crowd Anomaly Detection and Localization'. In: *Pattern Recognition* 61 (July 2016). DOI: 10.1016/j.patcog.2016.06.016.

[158] Bosi Yu, Yazhou Liu and Quansen Sun. 'A Content-Adaptively Sparse Reconstruction Method for Abnormal Events Detection With Low-Rank Property'. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.4 (2017), pp. 704–716. DOI: 10.1109/TSMC.2016.2638048.

[159] Cewu Lu, Jianping Shi and Jiaya Jia. 'Abnormal Event Detection at 150 FPS in MATLAB'. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2720–2727. DOI: 10.1109/ICCV.2013.338.

[160] Yang Cong, Junsong Yuan and Ji Liu. 'Sparse reconstruction cost for abnormal event detection'. In: *CVPR 2011*. 2011, pp. 3449–3456. DOI: 10.1109/CVPR.2011.5995434.

[161] Kinjal Vishnuprasad Joshi and Narendra M. Patel. 'A CNN Based Approach for Crowd Anomaly Detection'. In: *Int. J. Next Gener. Comput.* 12 (2021).

[162] Khosro Rezaee et al. 'A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance'. In: *Personal and Ubiquitous Computing* (June 2021). DOI: 10.1007/s00779-021-01586-5.

[163] John Gatara Munyua, Geoffrey Mariga Wambugu and Stephen Thiiru Njenga. 'A Survey of Deep Learning Solutions for Anomaly Detection in Surveillance Videos'. In: *International Journal of Computer and Information Technology(2279-0764)* 10.5 (Oct. 2021). DOI: 10.24203/ijcit.v10i5.166. URL: https://www.ijcit.com/index.php/ijcit/article/view/166.

[164] Abid Mehmood. 'Efficient Anomaly Detection in Crowd Videos Using Pre-Trained 2D Convolutional Neural Networks'. In: *IEEE Access* 9 (2021), pp. 138283–138295. DOI: 10.1109/ACCESS.2021.3118009.

[165] Muhammad Farooq, Mohamad Saad and Sultan Khan. 'Motion-shape-based deep learning approach for divergence behavior detection in high-density crowd'. In: *The Visual Computer* 38 (May 2022). DOI: 10.1007/s00371-021-02088-4.

[166] Mahdyar Ravanbakhsh et al. *Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection*. 2016. DOI: 10.48550/ARXIV.1610.00307. URL: https://arxiv.org/abs/1610.00307.

[167] Shifu Zhou et al. 'Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes'. In: *Signal Processing: Image Communication* 47 (2016), pp. 358–368. ISSN: 0923-5965. DOI: https://doi.org/10.1016/j.image.2016.06.007. URL: https://www.sciencedirect.com/science/article/pii/S0923596516300935.

[168] Yang Cong, Junsong Yuan and Ji Liu. 'Abnormal event detection in crowded scenes using sparse representation'. In: *Pattern Recognition* 46.7 (2013), pp. 1851–1864. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2012.11.021. URL: https://www.sciencedirect.com/science/article/pii/S0031320312005055.

[169] Cem Direkoglu. 'Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks'. In: *IEEE Access* 8 (2020), pp. 80408–80416. DOI: 10.1109/ACCESS.2020.2990355.

[170] Jiayu Sun, Jie Shao and Chengkun He. 'Abnormal event detection for video surveillance using deep one-class learning'. In: *Multimedia Tools and Applications* 78 (2017), pp. 3633–3647.

[171] Xing Hu et al. 'A weakly supervised framework for abnormal behavior detection and localization in crowded scenes'. In: *Neurocomputing* 383 (2020), pp. 270–281. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.11.087. URL: https://www.sciencedirect.com/science/article/pii/S0925231219316911.

[172] Yuanping Xu et al. 'Dual-channel CNN for efficient abnormal behavior identification through crowd feature engineering'. In: *Machine Vision and Applications* 30 (July 2019). DOI: 10.1007/s00138-018-0971-6.

[173] Fath U Min Ullah et al. 'Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network'. In: *Sensors* 19 (May 2019), p. 2472. DOI: 10.3390/s19112472.

[174] Wei Song et al. 'A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks'. In: *IEEE Access* 7 (2019), pp. 39172–39179. DOI: 10.1109/ACCESS.2019.2906275.

[175] Mohammad Sabokrou et al. 'Deep-Cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes'. In: *IEEE Transactions on Image Processing* 26.4 (2017), pp. 1992–2004. DOI: 10.1109/TIP.2017.2670780.

[176] Gaurav Tripathi, Kuldeep Singh and Dinesh Kumar Vishwakarma. 'Crowd Emotion Analysis Using 2D ConvNets'. In: *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. 2020, pp. 969–974. DOI: 10.1109/ICSSIT48917.2020.9214208.

[177] Matheus Gutoski et al. 'A Comparative Study of Transfer Learning Approaches for Video Anomaly Detection'. In: *International Journal of Pattern Recognition and Artificial Intelligence* 35.05 (2021), p. 2152003. DOI: 10.1142/S0218001421520030.

[178] Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton. 'ImageNet Classification with Deep Convolutional Neural Networks'. In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.

[179] Alaa Almazroey and Salma Kammoun jarraya. 'Abnormal Events and Behavior Detection in Crowd Scenes Based on Deep Learning and Neighborhood Component Analysis Feature Selection'. In: Mar. 2020, pp. 258–267. ISBN: 978-3-030-44288-0. DOI: 10.1007/978-3-030-44289-7_25.

[180] Valentina Franzoni, Giulio Biondi and Alfredo Milani. 'Emotional sounds of crowds: spectrogram-based analysis using deep learning'. In: *Multimedia Tools and Applications* 79 (Dec. 2020). DOI: 10.1007/s11042-020-09428-x.

[181] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556.

[182] Aman Ahmed et al. 'Crowd Detection and Analysis for Surveillance Videos using Deep Learning'. In: *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*. 2021, pp. 1–7. DOI: 10.1109/ICESC51422.2021.9532683.

[183] Ou Ye et al. 'Abnormal Event Detection via Feature Expectation Subgraph Calibrating Classification in Video Surveillance Scenes'. In: *IEEE Access* 8 (2020), pp. 97564–97575. DOI: 10.1109/ACCESS.2020.2997357.

[184] Ahlam Al-Dhamari, Rubita Sudirman and Nasrul Humaimi Mahmood. 'Transfer Deep Learning Along With Binary Support Vector Machine for Abnormal Behavior Detection'. In: *IEEE Access* 8 (2020), pp. 61085–61095. DOI: 10.1109/ACCESS.2020.2982906.

[185] Fariba Rezaei and Mehran Yazdi. 'A New Semantic and Statistical Distance-Based Anomaly Detection in Crowd Video Surveillance'. In: *Wireless Communications and Mobile Computing* 2021 (May 2021), pp. 1–9. DOI: 10.1155/2021/5513582.

[186] Chin-Chia Tsai, Tsung-Hsuan Wu and Shang-Hong Lai. 'Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation'. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 3065–3073. DOI: 10.1109/WACV51458.2022.00312.

[187] Kaiming He et al. *Identity Mappings in Deep Residual Networks*. 2016. DOI: 10.48550/ARXIV.1603.05027. URL: https://arxiv.org/abs/1603.05027.

[188] Guansong Pang et al. *Self-trained Deep Ordinal Regression for End-to-End Video Anomaly Detection*. 2020. DOI: 10.48550/ARXIV.2003.06780. URL: https://arxiv.org/abs/2003.06780.

[189] Zirgham Ilyas et al. 'A Hybrid Deep Network Based Approach for Crowd Anomaly Detection'. In: 80.16 (July 2021), pp. 24053–24067. ISSN: 1380-7501. DOI: 10.1007/s11042-021-10785-4. URL: https://doi.org/10.1007/s11042-021-10785-4.

[190] Chongke Wu et al. *Video Anomaly Detection Using Pre-Trained Deep Convolutional Neural Nets and Context Mining*. 2020. DOI: 10.48550/ARXIV.2010.02406. URL: https://arxiv.org/abs/2010.02406.

[191] Romany F. Mansour et al. 'Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model'. In: *Image and Vision Computing* 112 (2021), p. 104229. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2021.104229. URL: https://www.sciencedirect.com/science/article/pii/S0262885621001347.

[192] Peipeng Chen, Yuan Gao and Andy J. Ma. 'Multi-level Attentive Adversarial Learning with Temporal Dilation for Unsupervised Video Domain Adaptation'. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 776–785. DOI: 10.1109/WACV51458.2022.00085.

[193] Zheng-ping Hu et al. 'Parallel spatial-temporal convolutional neural networks for anomaly detection and location in crowded scenes'. In: *Journal of Visual Communication and Image Representation* 67 (2020), p. 102765. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2020.102765. URL: https://www.sciencedirect.com/science/article/pii/S1047320320300158.

[194] Kensho Hara, Hirokatsu Kataoka and Yutaka Satoh. *Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition*. 2017. DOI: 10.48550/ARXIV.1708.07632. URL: https://arxiv.org/abs/1708.07632.

[195] Junyu Gao, Maoguo Gong and Xuelong Li. *Audio-visual Representation Learning for Anomaly Events Detection in Crowds*. 2021. DOI: 10.48550/ARXIV.2110.14862. URL: https://arxiv.org/abs/2110.14862.

[196] Yi Hao et al. 'Spatiotemporal consistency-enhanced network for video anomaly detection'. In: *Pattern Recognition* 121 (2022), p. 108232. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2021.108232. URL: https://www.sciencedirect.com/science/article/pii/S0031320321004131.

[197] Yiheng Cai et al. 'Video anomaly detection with multi-scale feature and temporal information fusion'. In: *Neurocomputing* 423 (2021), pp. 264–273. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2020.10.044. URL: https://www.sciencedirect.com/science/article/pii/S0925231220315976.

[198] Dorcas Oladayo Esan, Pius A. Owolawi and Chuling Tu. 'Anomalous Detection System in Crowded Environment using Deep Learning'. In: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2020, pp. 29–35. DOI: 10.1109/CSCI51800.2020.00012.

[199] Fuqiang Zhou et al. 'Unsupervised Learning Approach for Abnormal Event Detection in Surveillance Video by Hybrid Autoencoder'. In: *Neural Processing Letters* 52 (Oct. 2020). DOI: 10.1007/s11063-019-10113-w.

[200] Rashmika Nawaratne et al. 'Spatiotemporal Anomaly Detection Using Deep Learning for Real-Time Video Surveillance'. In: *IEEE Transactions on Industrial Informatics* 16.1 (2020), pp. 393–402. DOI: 10.1109/TII.2019.2938527.

[201] Bo Li, Sam Leroux and Pieter Simoens. 'Decoupled appearance and motion learning for efficient anomaly detection in surveillance video'. In: *Computer Vision and Image Understanding* 210 (July 2021), p. 103249. DOI: 10.1016/j.cviu.2021.103249.

[202] Elizabeth Varghese, Sabu M Thampi and Stefano Berretti. 'A Psychologically Inspired Fuzzy Cognitive Deep Learning Framework to Predict Crowd Behavior'. In: *IEEE Transactions on Affective Computing* (2020), pp. 1–1. DOI: 10.1109/TAFFC.2020.2987021.

[203] Alexandre Alahi et al. 'Social LSTM: Human Trajectory Prediction in Crowded Spaces'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 961–971. DOI: 10.1109/CVPR.2016.110.

[204] Qinmin Ma and Bai Yuan Ding. 'Abnormal Event Detection in Videos Based on Deep Neural Networks'. In: *Sci. Program.* 2021 (Jan. 2021). ISSN: 1058-9244. DOI: 10.1155/2021/6412608. URL: https://doi.org/10.1155/2021/6412608.

[205] Renuka Sharma, Satvik Mashkaria and Suyash P. Awate. 'A Semi-supervised Generalized VAE Framework for Abnormality Detection using One-Class Classification'. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 1302–1310. DOI: 10.1109/WACV51458.2022.00137.

[206] Ming Xu et al. 'An Efficient Anomaly Detection System for Crowded Scenes Using Variational Autoencoders'. In: *Applied Sciences* 9 (Aug. 2019), p. 3337. DOI: 10.3390/app9163337.

[207] Faraz Waseem, Rafael Perez Martinez and Chris Wu. *Visual anomaly detection in video by variational autoencoder*. 2022. DOI: 10.48550/ARXIV.2203.03872. URL: https://arxiv.org/abs/2203.03872.

[208] Shiyang Yan et al. 'Abnormal Event Detection From Videos Using a Two-Stream Recurrent Variational Autoencoder'. In: *IEEE Transactions on Cognitive and Developmental Systems* 12.1 (2020), pp. 30–42. DOI: 10.1109/TCDS.2018.2883368.

[209] Diederik P. Kingma and Max Welling. 'An Introduction to Variational Autoencoders'. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. DOI: 10.1561/2200000056.

[210] Yaxiang Fan et al. *Video Anomaly Detection and Localization via Gaussian Mixture Fully Convolutional Variational Autoencoder*. 2018. DOI: 10.48550/ARXIV.1805.11223. URL: https://arxiv.org/abs/1805.11223.

[211] Karishma Pawar and Vahida Attar. 'Assessment of Autoencoder Architectures for Data Representation'. In: Oct. 2019, pp. 101–132. ISBN: 978-3-030-31755-3. DOI: 10.1007/978-3-030-31756-0_4.

[212] K.V. Deepak et al. 'Deep Multi-view Representation Learning for Video Anomaly Detection Using Spatiotemporal Autoencoders'. In: *Circuits, Systems, and Signal Processing* 40 (Mar. 2021). DOI: 10.1007/s00034-020-01522-7.

[213] Mujtaba Asad et al. 'Anomaly3D: Video anomaly detection based on 3D-normality clusters'. In: *Journal of Visual Communication and Image Representation* 75 (2021), p. 103047. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2021.103047. URL: https://www.sciencedirect.com/science/article/pii/S1047320321000201.

[214] K.V. Deepak, S. Chandrakala and C. Mohan. 'Residual spatiotemporal autoencoder for unsupervised video anomaly detection'. In: *Signal, Image and Video Processing* 15 (Feb. 2021). DOI: 10.1007/s11760-020-01740-1.

[215] Jerome Revaud et al. *EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow*. 2015. DOI: 10.48550/ARXIV.1501.02565. URL: https://arxiv.org/abs/1501.02565.

[216] Eddy Ilg et al. *FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks*. 2016. DOI: 10.48550/ARXIV.1612.01925. URL: https://arxiv.org/abs/1612.01925.

[217] Tak-Wai Hui, Xiaoou Tang and Chen Change Loy. *LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation*. 2018. DOI: 10.48550/ARXIV.1805.07036. URL: https://arxiv.org/abs/1805.07036.

[218] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. 2016. DOI: 10.48550/ARXIV.1612.03144. URL: https://arxiv.org/abs/1612.03144.

[219] Tak-Wai Hui, Xiaoou Tang and Chen Change Loy. *A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization*. 2019. DOI: 10.48550/ARXIV.1903.07414. URL: https://arxiv.org/abs/1903.07414.

[220] Tak-Wai Hui and Chen Change Loy. *LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation*. 2020. DOI: 10.48550/ARXIV.2007.09319. URL: https://arxiv.org/abs/2007.09319.

[221] Zachary Teed and Jia Deng. *RAFT: Recurrent All-Pairs Field Transforms for Optical Flow*. 2020. DOI: 10.48550/ARXIV.2003.12039. URL: https://arxiv.org/abs/2003.12039.

[222] Abdelrahman Eldesokey and Michael Felsberg. *Normalized Convolution Upsampling for Refined Optical Flow Estimation*. 2021. DOI: 10.48550/ARXIV.2102.06979. URL: https://arxiv.org/abs/2102.06979.

[223] Deqing Sun et al. *AutoFlow: Learning a Better Training Set for Optical Flow*. 2021. DOI: 10.48550/ARXIV.2104.14544. URL: https://arxiv.org/abs/2104.14544.

[224] Austin Stone et al. *SMURF: Self-Teaching Multi-Frame Unsupervised RAFT with Full-Image Warping*. 2021. DOI: 10.48550/ARXIV.2105.07014. URL: https://arxiv.org/abs/2105.07014.

[225] Taihong Xiao et al. *Learnable Cost Volume Using the Cayley Representation*. 2020. DOI: 10.48550/ARXIV.2007.11431. URL: https://arxiv.org/abs/2007.11431.

[226] Haofei Xu et al. *GMFlow: Learning Optical Flow via Global Matching*. 2021. DOI: 10.48550/ARXIV.2111.13680. URL: https://arxiv.org/abs/2111.13680.

[227] Zachary Teed and Jia Deng. *RAFT-3D: Scene Flow using Rigid-Motion Embeddings*. 2020. DOI: 10.48550/ARXIV.2012.00726. URL: https://arxiv.org/abs/2012.00726.

[228] Shihao Jiang et al. *Learning to Estimate Hidden Motions with Global Motion Aggregation*. 2021. DOI: 10.48550/ARXIV.2104.02409. URL: https://arxiv.org/abs/2104.02409.

[229] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[230] John Barron, David Fleet and S. Beauchemin. 'Performance Of Optical Flow Techniques'. In: *International Journal of Computer Vision* 12 (Feb. 1994), pp. 43–77. DOI: 10.1007/BF01420984.

[231] Y. Lecun et al. 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[232] Simon Baker et al. 'A Database and Evaluation Methodology for Optical Flow'. In: *2007 IEEE 11th International Conference on Computer Vision*. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4408903.

[233] Andreas Geiger, Philip Lenz and Raquel Urtasun. 'Are we ready for autonomous driving? The KITTI vision benchmark suite'. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[234] Moritz Menze and Andreas Geiger. 'Object scene flow for autonomous vehicles'. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3061–3070. DOI: 10.1109/CVPR.2015.7298925.

[235] Daniel J. Butler et al. 'A Naturalistic Open Source Movie for Optical Flow Evaluation'. In: *ECCV*. 2012.

[236] Trong Nguyen Nguyen and Jean Meunier. *Anomaly Detection in Video Sequence with Appearance-Motion Correspondence*. 2019. DOI: 10.48550/ARXIV.1908.06351. URL: https://arxiv.org/abs/1908.06351.

[237] Mohammad Sabih and Dinesh Vishwakarma. 'Crowd anomaly detection with LSTMs using optical features and domain knowledge for improved inferring'. In: *The Visual Computer* 38 (May 2022). DOI: 10.1007/s00371-021-02100-x.

[238] scikit-learn developers. *sklearn.metrics.precision_score*. 2022. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html (visited on 21/05/2021).

[239] scikit-learn developers. *sklearn.metrics.recall_score*. 2022. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html (visited on 21/05/2021).

[240] Google Developers. *Classification: ROC Curve and AUC*. 2020. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (visited on 21/05/2021).

[241] Mahmudul Hasan et al. *Learning Temporal Regularity in Video Sequences*. 2016. DOI: 10.48550/ARXIV.1604.04574. URL: https://arxiv.org/abs/1604.04574.

[242] Antoni B. Chan, Zhang-Sheng John Liang and Nuno Vasconcelos. 'Privacy preserving crowd monitoring: Counting people without people models or tracking'. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–7. DOI: 10.1109/CVPR.2008.4587569.

[243] Weixin Luo, Wen Liu and Shenghua Gao. 'A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework'. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 341–349. DOI: 10.1109/ICCV.2017.45.

[244] Amit Adam et al. 'Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.3 (2008), pp. 555–560. DOI: 10.1109/TPAMI.2007.70825.

[245] Waqas Sultani, Chen Chen and Mubarak Shah. *Real-world Anomaly Detection in Surveillance Videos*. 2018. DOI: 10.48550/ARXIV.1801.04264. URL: https://arxiv.org/abs/1801.04264.

[246] Antoni B. Chan, Zhang-Sheng John Liang and Nuno Vasconcelos. *UCSD Anomaly Detection Dataset*. 2008. URL: http://www.svcl.ucsd.edu/projects/anomaly/dataset.html (visited on 16/05/2021).

[247] Keval Doshi and Yasin Yilmaz. 'Rethinking Video Anomaly Detection - A Continual Learning Approach'. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022, pp. 3036–3045. DOI: 10.1109/WACV51458.2022.00309.

[248] W. Liu, D. Lian W. Luo and S. Gao. 'Future Frame Prediction for Anomaly Detection – A New Baseline'. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[249] Adam Paszke et al. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[250] Yangqing Jia et al. 'Caffe: Convolutional Architecture for Fast Feature Embedding'. In: *arXiv preprint arXiv:1408.5093* (2014).

[251] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[252] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. DOI: 10.48550/ARXIV.1502.03167. URL: https://arxiv.org/abs/1502.03167.

[253] Xavier Glorot, Antoine Bordes and Y. Bengio. 'Deep Sparse Rectifier Neural Networks'. In: vol. 15. Jan. 2010.

[254] Christian Garbin, Xingquan Zhu and Oge Marques. 'Dropout vs. batch normalization: an empirical study of their impact to deep learning'. In: *Multimedia Tools and Applications* 79 (May 2020), pp. 1–39. DOI: 10.1007/s11042-019-08453-9.

[255] Guangyong Chen et al. *Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks*. 2019. DOI: 10.48550/ARXIV.1905.05928. URL: https://arxiv.org/abs/1905.05928.

[256] Nitish Srivastava et al. 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting'. In: *Journal of Machine Learning Research* 15 (June 2014), pp. 1929–1958.

[257] L. Brigato and L. Iocchi. *A Close Look at Deep Learning with Small Data*. 2020. DOI: 10.48550/ARXIV.2003.12843. URL: https://arxiv.org/abs/2003.12843.

[258] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[259] Olga Russakovsky et al. 'ImageNet Large Scale Visual Recognition Challenge'. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[260] Yong Shean Chong and Yong Haur Tay. *Abnormal Event Detection in Videos using Spatiotemporal Autoencoder*. 2017. DOI: 10.48550/ARXIV.1701.01546. URL: https://arxiv.org/abs/1701.01546.

[261] Matthew D. Zeiler et al. 'Deconvolutional networks'. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 2528–2535. DOI: 10.1109/CVPR.2010.5539957.

[262] Daniel Scharstein. *Some utilities for reading, writing, and color-coding .flo images*. 2007. URL: https://vision.middlebury.edu/flow/code/flow-code/README.txt (visited on 23/05/2021).

[263] Tom Runia and Dmitry Frumkin. *Optical Flow Visualization*. 2020. URL: https://github.com/tomrunia/OpticalFlow_Visualization (visited on 23/05/2021).

[264] Kedar Tatwawadi. *add an option to normalize the flow using a fixed value*. 2020. URL: https://github.com/tomrunia/OpticalFlow_Visualization/pull/7 (visited on 23/05/2021).

[265] PyTorch. *DATASETS & DATALOADERS*. 2022. URL: https://pytorch.org/tutorials/beginner/basics/data_tutorial.html (visited on 23/05/2021).

[266] PyTorch. *IMAGEFOLDER*. 2017. URL: https://pytorch.org/vision/stable/generated/torchvision.datasets.ImageFolder.html (visited on 23/05/2021).

[267] PyTorch. *TRANSFORMING AND AUGMENTING IMAGES*. 2017. URL: https://pytorch.org/vision/stable/transforms.html (visited on 24/05/2021).

[268] PyTorch. *MODELS AND PRE-TRAINED WEIGHTS*. 2017. URL: https://pytorch.org/vision/stable/models.html (visited on 24/05/2021).

[269] Ekin D. Cubuk et al. *AutoAugment: Learning Augmentation Policies from Data*. 2018. DOI: 10.48550/ARXIV.1805.09501. URL: https://arxiv.org/abs/1805.09501.

[270] PyTorch. *TORCH.UTILS.DATA*. 2019. URL: https://pytorch.org/docs/stable/data.html#torch.utils.data.ConcatDataset (visited on 24/05/2021).

[271] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[272] Tino Weinkauf. *Extracting and Filtering Minima and Maxima of 1D Functions*. 2022. URL: https://www.csc.kth.se/~weinkauf/notes/persistence1d.html (visited on 02/06/2021).

[273] Ambareesh Ravi and Fakhri Karray. 'Exploring Convolutional Recurrent architectures for anomaly detection in videos: a comparative study'. In: *Discover Artificial Intelligence* 1 (Dec. 2021). DOI: 10.1007/s44163-021-00004-2.

[274] Jingtao Hu et al. 'An Efficient and Robust Unsupervised Anomaly Detection Method Using Ensemble Random Projection in Surveillance Videos'. In: *Sensors* 19.19 (2019). ISSN: 1424-8220. DOI: 10.3390/s19194145. URL: https://www.mdpi.com/1424-8220/19/19/4145.

[275] Tian Wang et al. 'AED-Net: An Abnormal Event Detection Network'. In: (2019). DOI: 10.48550/ARXIV.1903.11891. URL: https://arxiv.org/abs/1903.11891.

[276] Sheri Reddy, V Kakulapati and Prince Appiah. 'Advance Security: Anomaly Detection in Mobile Crowd Sensing Using Machine Learning Techniques'. In: *SSRN Electronic Journal* (Mar. 2021).

[277] Limin Wang et al. 'Temporal Segment Networks: Towards Good Practices for Deep Action Recognition'. In: *The European Conference on Computer Vision (ECCV)*. 2016.

[278] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929.

[279] Haiping Wu et al. *CvT: Introducing Convolutions to Vision Transformers*. 2021. DOI: 10.48550/ARXIV.2103.15808. URL: https://arxiv.org/abs/2103.15808.

[280] Maithra Raghu et al. *Do Vision Transformers See Like Convolutional Neural Networks?* 2021. DOI: 10.48550/ARXIV.2108.08810. URL: https://arxiv.org/abs/2108.08810.

[281] Jonas Herskind Sejr and Anna Schneider-Kamp. 'Explainable outlier detection: What, for Whom and Why?' In: *Machine Learning with Applications* 6 (2021), p. 100172. ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2021.100172. URL: https://www.sciencedirect.com/science/article/pii/S2666827021000864.