



Article

An Overview of Deep Learning Based Methods for Unsupervised and Semi-Supervised Anomaly Detection in Videos

B. Ravi Kiran ^{1,2,*}, Dilip Mathew Thomas ² and Ranjith Parakkal ²

¹ Computer Science, Université de Lille 3, 59655 Villeneuve-d'Ascq, France

² Uncanny Vision Solutions, Bangalore, Karnataka 560008, India; dilip@uncannyvision.com (D.M.T.); ranjith@uncannyvision.com (R.P.)

* Correspondence: beedotkiran@gmail.com; Tel.: +33-783-149-467

† Current address: 79 Rue Brillat Savarin, Paris 75013, France.

Received: 20 November 2017; Accepted: 1 February 2018; Published: 7 February 2018

Abstract: Videos represent the primary source of information for surveillance applications. Video material is often available in large quantities but in most cases it contains little or no annotation for supervised learning. This article reviews the state-of-the-art deep learning based methods for video anomaly detection and categorizes them based on the type of model and criteria of detection. We also perform simple studies to understand the different approaches and provide the criteria of evaluation for spatio-temporal anomaly detection.

Keywords: unsupervised methods; anomaly detection; representation learning; autoencoders; LSTMs; generative adversarial networks; Variational Autoencoders; predictive models

1. Introduction

Unsupervised representation learning has become an important domain with the advent of deep generative models which include the variational autoencoder (VAE) [1], generative adversarial networks (GANs) [2], Long Short Term memory networks (LSTMs) [3], and others. Anomaly detection is a well-known sub-domain of unsupervised learning in the machine learning and data mining community. Anomaly detection for images and videos are challenging due to their high dimensional structure of the images, combined with the non-local temporal variations across frames.

We focus on reviewing firstly, deep convolution architectures for feature or representation learnt “end-to-end” and secondly, predictive and generative models specifically for the task of video anomaly detection (VAD). Anomaly detection is an unsupervised learning task where the goal is to identify abnormal patterns or motions in data that are by definition infrequent or rare events. Furthermore, anomalies are rarely annotated and labeled data rarely available to train a deep convolutional network to separate normal class from the anomalous class. This is a fairly complex task since the class of normal points includes frequently occurring objects and regular foreground movements while the anomalous class include various types of rare events and unseen objects that could be summarized as a consistent class. Long streams of videos containing no anomalies are made available using which one is required to build a representation for a moving window over the video stream that estimates the normal behavior class while detecting anomalous movements and appearance, such as unusual objects in the scene.

Given a set of training samples containing no anomalies, the goal of anomaly detection is to design or learn a feature representation, that captures “normal” motion and spatial appearance patterns. Any deviations from this normal can be identified by measuring the approximation error either geometrically in a vector space or the posterior probability of a given model which fits training

sample representation vectors or by modeling the conditional probability of future samples given their past values and measuring the prediction error of test samples by training a predictive model, thus accounting for temporal structure in videos.

1.1. Anomaly Detection

Anomaly detection is an unsupervised pattern recognition task that can be defined under different statistical models. In this study we will explore models that perform linear approximations by PCA, non-linear approximation by various types of autoencoders and finally deep generative models.

Intuitively, a complex system under the action of various transformations is observed, the normal behavior is described through a few samples and a statistical model is built using the said normal behavior samples that is capable of generalizing well on unseen samples. The normal class distribution \mathcal{D} is estimated using the training samples $\mathbf{x}_i \in X_{\text{train}}$, by building a representation $f_{\theta} : X_{\text{train}} \rightarrow \mathbb{R}$ which minimizes model prediction loss

$$\theta^* = \arg \min_{\theta} \sum_{\mathbf{x}_i \in X_{\text{train}}} L_{\mathcal{D}}(\theta; \mathbf{x}_i) = \arg \min_{\theta} \sum_{\mathbf{x}_i \in X_{\text{train}}} \|f_{\theta}(\mathbf{x}_i) - \mathbf{x}_i\|^2 \quad (1)$$

error over all the training samples, over all i , is evaluated. Now the deviation of the test samples $\mathbf{x}_j \in X_{\text{test}}$ under this representation f_{θ^*} is evaluated as the anomaly score, $a(\mathbf{x}_j) = \|f_{\theta^*}(\mathbf{x}_j) - \mathbf{x}_j\|_2$ is used as a measure of deviation. For said models, the anomalous points are samples that are poorly approximated by the estimated model f_{θ^*} . Detection is achieved by evaluating a threshold on the anomaly score $a_j > T_{\text{thresh}}$. The threshold is a parameter of the detection algorithm and the variation of the threshold w.r.t detection performance is discussed under the Area under ROC section. For probabilistic models, anomalous points can be defined as samples that lie in low density or concentration regions of the domain of an input training distribution $P(\mathbf{x}|\theta)$.

Representation learning automates feature extraction for video data for tasks such as action recognition, action similarity, scene classification, object recognition, semantic video segmentation [4], human pose estimation, human behavior recognition and various other tasks. Unsupervised learning tasks in video include anomaly detection [5,6], unsupervised representation learning [7], generative models for video [8], and video prediction [9].

1.2. Datasets

We now define the video anomaly detection problem setup. The videos considered come from a surveillance camera where the background remains static, while the foreground constitutes of moving objects such as pedestrians, traffic and so on. The anomalous events are the change in appearance and motion patterns that deviate from the normal patterns observed in the training set. We see a few examples demonstrated in Figure 1.

Here we list the frequently evaluated datasets, though this is not exhaustive. The UCSD dataset [5] consists of pedestrian videos where anomalous time instances correspond to the appearance of objects like a cyclist, a wheelchair, and a car in the scene that is usually populated with pedestrians walking along the roads. People walking in unusual locations are also considered anomalous. In CUHK Avenue Dataset [10] anomalies correspond to strange actions such as a person throwing papers or bag, moving in unusual directions, and appearance of unusual objects like bags and bicycle. In the Subway entry and exit datasets people moving in the wrong direction, loitering and so on are considered as anomalies. UMN dataset [11] consists of videos showing unusual crowd activity, and is a particular case of the video anomaly detection problem. The Train dataset [12] contains moving people in a train. The anomalous events are mainly due to unusual movements of people in the train. And finally the Queen Mary University of London U-turn dataset [13] contains normal traffic with anomalous events such as jaywalking and movement of a fire engine. More recently, a controlled environment based LV dataset has been introduced by [14], with challenging examples for the task of online video anomaly detection.

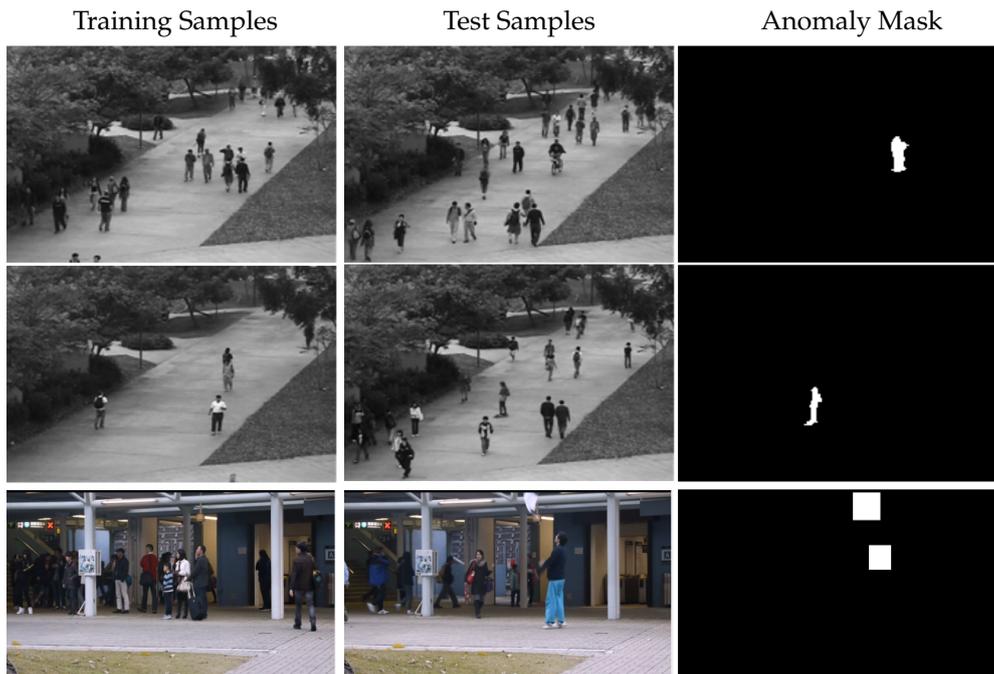


Figure 1. UCSD dataset (top two rows): Unlike normal training video streams, anomalies consist of a person on a bicycle and a skating board, ground truth detection shown in anomaly mask. Avenue dataset (bottom row): Unlike normal training video streams, anomalies consist of a person throwing papers. Other examples include walking with an abnormal object (bicycle), a person running (strange action), and a person walking in the wrong direction.

2. Representation Learning for Video Anomaly Detection (VAD)

Videos are high dimensional signals with both spatial-structure, as well as local temporal variations. An important problem of anomaly detection in videos is to learn a representation of input sample space $f_{\theta} : \mathcal{X} \rightarrow \mathbb{R}^d$, to d -dimensional vectors. The idea of feature learning is to automate the process of finding a good representation of the input space, that takes into account important prior information about the problem [15]. This follows from the No-Free-Lunch-Theorem which states that no universal learner exists for every training distribution \mathcal{D} . Following work already established for video anomaly detection, the task concretely consists in detecting deviations from models of static background, normal crowd appearance and motion from optical flow, change in trajectory and other priors. Representation learning consists of building a parameterized model $f_{\theta} : \mathcal{X} \rightarrow \mathcal{Z} \rightarrow \mathcal{X}$, and in this study we focus on representations that reconstruct the input, while the latent space \mathcal{Z} is constrained to be invariant to changes in the input, such as change in luminance, translations of objects in the scene that don't deviate normal movement patterns, and others. This provides a way to introduce prior information to reconstruct normal samples.

2.1. Taxonomy

The goal of this survey is to provide a compact review of the state of the art in video anomaly detection based on unsupervised and semi-supervised deep learning architectures. The survey characterizes the underlying video representation or model as one of the following:

1. **Representation learning for reconstruction:** Methods such as Principal component analysis (PCA), Autoencoders (AEs) are used to represent the different linear and non-linear transformations to the appearance (image) or motion (flow), that model the normal behavior in surveillance videos. Anomalies represent any deviations that are poorly reconstructed.

2. **Predictive modeling:** Video frames are viewed as temporal patterns or time series, and the goal is to model the conditional distribution $P(\mathbf{x}_t | (\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \dots, \mathbf{x}_{t-p}))$. In contrast to reconstruction, where the goal is to learn a generative model that can successfully reconstruct frames of a video, the goal here is to predict the current frame or its encoded representation using the past frames. Examples include autoregressive models and convolutional Long-Short-Term-Memory models.
3. **Generative models:** Variational Autoencoders (VAE), Generative Adversarial Networks (GAN) and Adversarially trained AutoEncoders (AAE), are used for the purpose of modeling the likelihood of normal video samples in an end-to-end deep learning framework.

An important common aspect in all these models is the problem of representation learning, which refers to the feature extraction or transformation of input training data for the task of anomaly detection. We shall also remark the other secondary feature transformations performed in each of these different models and their purposes.

2.2. Context of the Review

A short review on the subject of video anomaly detection is provided here [16]. To the best of our knowledge, there has not been a systematic study of deep architectures for video anomaly detection, which is characterized by abnormal appearance and motion features, that occur rarely. We cite below the other domains which do not fall under this study.

- A detailed review of abnormal human behavior and crowd motion analysis is provided in [17] and [18]. This includes deep architectures such as Social-LSTM [19] based on the social force model [20] where the goal is to predict pedestrian motion taking into account the movement of neighboring pedestrians.
- Action recognition is an important domain in computer vision which requires supervised learning of efficient representations of appearance and motion for the purpose of classification [21]. Convolutional networks were employed to classify various actions in video quite early [22]. Recent work involves fusing feature maps evaluated on different frames (over time) of video [23] yielding state of the art results. Finally, convolutional networks using 3-D filters (C3D) have become a recent base-line for action recognition [24].
- Unsupervised representation learning is a well-established domain and the readers are directed towards a complete review of the topic in [25], as well as the deep learning book [26].

In this review, we shall mainly focus on the taxonomy provided and restrict our review to deep convolutional networks and deep generative models that enable end-to-end spatio-temporal representation learning for the task of anomaly detection in videos. We also aim to provide an understanding of what aspects of detection do these different models target. Apart from the taxonomy being addressed in this study, there have been many other approaches. One could cite work on anomaly detection based on K-Nearest Neighbors [27], unsupervised clustering [28], and object speed and size [29].

We briefly review the set of hand-engineered features used for the task of video anomaly detection, though our focus still remains deep learning based architectures. Mixture of dynamic textures (MDT) is a generative mixture model defined for each spatio-temporal windows or cubes of the raw training video [5,6]. It models appearance and motion features and thus detects both spatial and temporal anomalies. Histogram of oriented optical flow and oriented gradients [30] is a baseline used in anomaly detection and crowd analysis, [31–33]. Tracklets are representation of movements in videos and have been applied to abnormal crowd motion analysis [34,35]. More recently, there has been work on developing optical flow acceleration features for motion description [36].

Problem setup: Given a training sequence of images from a video, $X_{\text{train}} \in \mathbb{R}^{N_{\text{train}} \times r \times c}$, which contains only “normal motion patterns” and no anomalies, and given a test sequence $X_{\text{test}} \in \mathbb{R}^{N_{\text{test}} \times r \times c}$, which is susceptible to contain anomalies, the task consists in associating each

frame with an anomaly score for the temporal variation, as well as a spatial score to localize the anomaly in space. This is demonstrated in Figure 2.

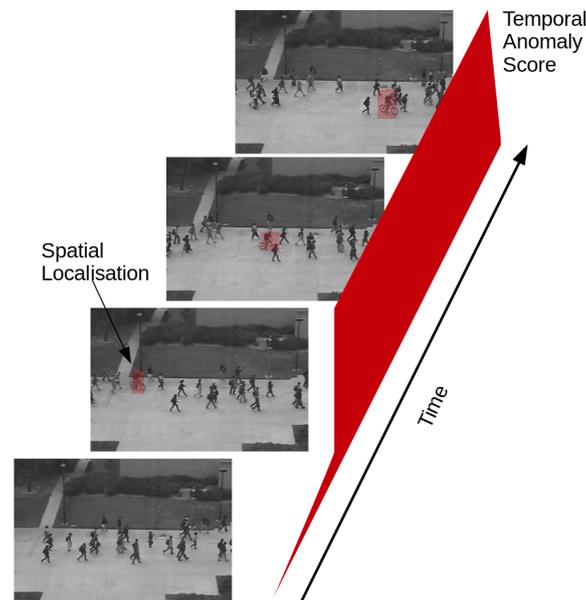


Figure 2. Visualizing anomalous regions and temporal anomaly score.

The anomaly detection task is usually considered unsupervised when there is no direct information or labels available about the positive rare class. However the samples in the study with no anomalies are available, and thus is a semi-supervised learning problem.

For $Z = \{x_i, y_i\}, i \in [1, N]$ we have samples only with $y_i = 0$. The goal thus of anomaly detection is two-fold: first, find the representation of the input feature $f_\theta(x_i)$, for example, using convolutional neural networks (CNNs), and then the decision rule $s(f_\theta(x_i)) \in \{0, 1\}$ that detects anomalies, whose detection rate can be parameterized as per the application.

3. Reconstruction Models

We begin with an input training video $X_{\text{train}} \in \mathbb{R}^{N \times d}$, with N frames and $d = r \times c$ pixels per frame, which represents the dimensionality of each vector. In this section, we shall focus on reducing the expected reconstruction error by different methods. We shall describe the Principal Component Analysis (PCA), Convolutional AutoEncoder (ConvAE), and Contractive AutoEncoders (CtractAE), and their setup for dimensionality reduction and reconstruction.

3.1. Principal Component Analysis

PCA finds the directions of maximal variance in the training data. In the case of videos, we are aiming to model the spatial correlation between pixel values which are components of the vector representing a frame at a particular time instant.

With input training matrix X , which has zero mean, we are looking for a set of orthogonal projections that whiten/de-correlate the features in the training set:

$$\min_{W^T W = I} \|X - (XW)W^T\|_F^2 = \|X - \hat{X}\|_F^2 \quad (2)$$

where, $W \in \mathbb{R}^{d \times k}$, with the constraint $W^T W = I$ representing an orthonormal reconstruction of the input X . The projection XW is a vector in a lower dimensional subspace, with fewer components than the vectors from X . This reduction in dimensionality is used to capture the anomalous behavior,

as samples that are not well reconstructed. The anomaly score is given by the Mahalanobis distance between the input and the reconstruction, or the variance scaled reconstruction error:

$$A = (X - \hat{X})\Sigma^{-1}(X - \hat{X})^T \tag{3}$$

We associate each frame with continual optical flow magnitude, and learn atomic motion patterns with standard PCA on the training set, and evaluate reconstruction error on the test optical flow magnitude. This serves a baseline for our study. A refined version was implemented and evaluated in [37] which evaluated the atomic movement patterns using a probabilistic PCA [38], over rectangular regions over the image domain.

Optical flow estimation is a costly step of this algorithm, and there has been large progress in the improving its evaluating speed. Authors [39], propose to trade of accuracy of for fast approximation of optical flow using PCA to interpolate flow fields.

3.2. Autoencoders

An Autoencoder is a neural network trained by back-propagation and provides an alternative to PCA to perform dimensionality reduction by reducing the reconstruction error on the training set, shown in Figure 3. It takes an input $x \in \mathbb{R}^d$ and maps it to the latent space representation $z \in \mathbb{R}^k$, by a deterministic application, $z = \sigma(Wx + b)$.

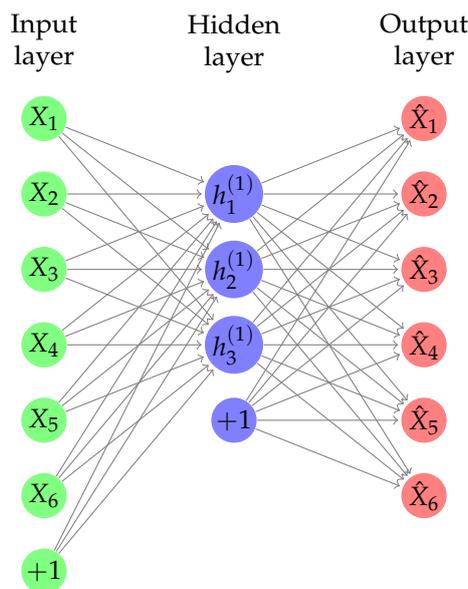


Figure 3. Autoencoder with single hidden layer.

Unlike the PCA the autoencoder (AE) performs a non-linear point-wise transform of the input $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, which is required to be a differentiable function. It is usually a rectified linear unit (ReLU) ($\sigma(x) = \max(0, x)$) or Sigmoid ($\sigma(x) = (1 + e^{-x})^{-1}$). Thus we can write a similar reconstruction of the input matrix given by:

$$\min_{U,V} \|X - \sigma(XU)V\|_F^2 \tag{4}$$

The low-dimensional representation is given by $\sigma(XU^*)$, where U^* represents the optimal linear encoding that minimizes the reconstruction loss above. There are multiple ways of regularizing the parameters U, V . One of the constraints is the average value of the activations in the hidden layer, this enforces sparsity.

3.3. Convolutional AutoEncoders (CAEs)

Autoencoders in their original form do view the input as a signal decomposed as the sum of other signals. Convolutional AutoEncoders (CAEs) [40], makes this decomposition explicit by weighting the result of the convolution operator. For a single channel input \mathbf{x} (for example a gray-scale image), the latent representation of the k th filter would be:

$$h^k = \sigma(\mathbf{x} * W^k) + b^k \tag{5}$$

The reconstruction is obtained by mapping back to the original image domain with the latent maps H and the decoding convolutional filter \widetilde{W}^k :

$$\hat{\mathbf{x}} = \sigma\left(\sum_{k \in H} h^k * \widetilde{W}^k + c\right) \tag{6}$$

where $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is a point-wise non-linearity like the sigmoid or hyperbolic tangent function. A single bias value is broadcast to each component of a latent map. These k -output maps can be used as an input to the next layer of the CAE. Several CAEs can be stacked into a deep hierarchy, which we again refer as a CAE to simplify the naming convention. We represent the stack of such operations as a single function $f_W : \mathbb{R}^{r \times c \times p} \rightarrow \mathbb{R}^{r \times c \times p}$ where the convolutional weights and biases are together represented by the weights W .

In retrospect, the PCA, and traditional AE, ignore the spatial structure and location of pixels in the image. This is also termed as being *permutation invariant*. It is important to note that when working with image frames of few 100×100 pixels, these methods introduce large redundancy in network parameters W , and furthermore span the entire visual receptive field. CAEs have fewer parameters on account of their weights being shared across many input locations/pixels.

3.4. CAEs for Video Anomaly Detection

In the recent work by [41], a deep convolutional autoencoder was trained to reconstruct an input sequence of frames from a training video set. We call this a Spatio-Temporal Stacked frame AutoEncoder (STSAE), to avoid confusion with similar names in the rest of the article. The STSAE in [41] stacks p frames $\mathbf{x}_i = [X_i, X_{i-1}, \dots, X_{i-p+1}]$ with each time slice treated as a different channel in the input tensor to a convolutional autoencoder. The model is regularized by augmenting the loss function with L2-norm of the model weights:

$$\mathcal{L}(W) = \frac{1}{2N} \sum_i \|\mathbf{x}_i - f_W(\mathbf{x}_i)\|_2^2 + \nu \|W\|_2^2 \tag{7}$$

where the tensor $\mathbf{x}_i \in \mathbb{R}^{r \times c \times p}$ is a cuboid with spatial dimensions r, c are the spatial dimensions and p is the number of frames temporally back into the past, with hyper-parameter ν which balances the reconstruction error and norm of the parameters, and N is the mini-batch size. The architecture of the convolutional autoencoder is reproduced in Figure 4.

The image or tensor $\hat{\mathbf{x}}_i$ reconstructed by the autoencoder enforces temporal regularity since the convolutional (weights) representation along with the bottleneck architecture of the autoencoder compresses information. The spatio-temporal autoencoder in [42] is shown in the right panel of Figure 4. The reconstruction error map at frame t is given by $E_t = |X_t - \hat{X}_t|$, while the temporal regularity score is given by the inverted, normalized reconstruction error:

$$s(t) = 1 - \frac{\sum_{(x,y)} E_t - \min_{(x,y)}(E_t)}{\max_{(x,y)}(E_t)} \tag{8}$$

where the \sum , \min and \max operators are across the spatial indices's (x, y) . In other models, the normalized reconstruction error is directly used as the anomaly score. One could envisage the

use of Mahalanobis distance here since the task is to evaluate the distance between test points from the points from the normal ones. This is evaluated as the error between the original tensor and the reconstruction from the autoencoder.

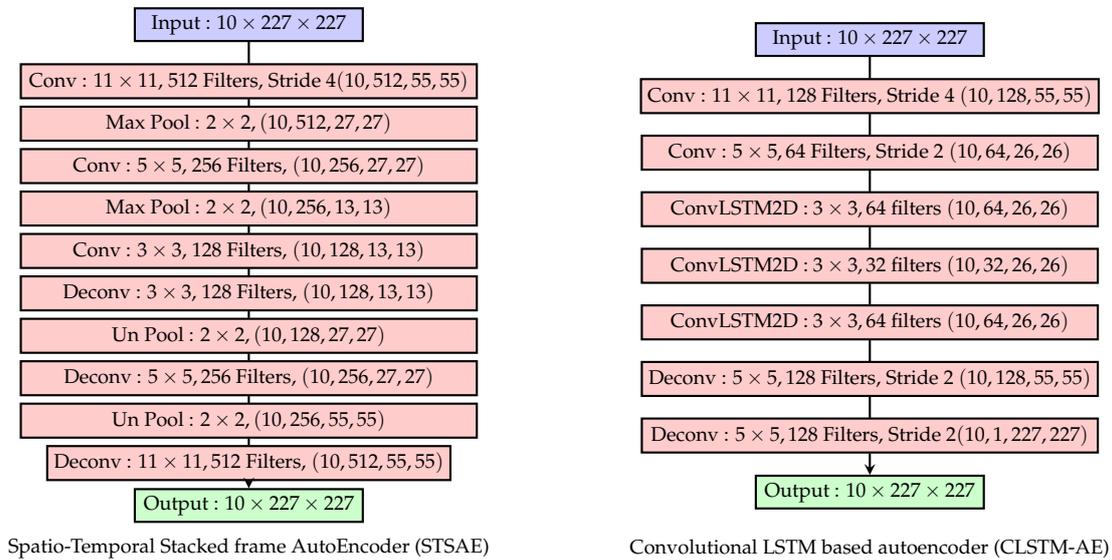


Figure 4. Spatio-Temporal Stacked frame AutoEncoder (left): A sequence of 10 frames are being reconstructed by a convolutional autoencoder, image reproduced from [41]. Convolutional LSTM based autoencoder (right): A sequence of 10 frames are being reconstructed by a spatio-temporal autoencoder, [42]. The Convolutional LSTM layers are predictive models that model the spatio-temporal correlation of pixels in the video. This is described in the predictive model section.

Robust versions of Convolutional AutoEncoders (RCAE) are studied in [43], where the goal is to evaluate anomalies in images by imposing L2-constraints on parameters W as well as adding a bias term. A video-patch (spatio-temporal) based autoencoder was employed by [44] to reconstruct patches, with a sparse autoencoder whose average activations were set to parameter ρ , enforcing sparseness, following the work in [45].

3.5. Contractive Autoencoders

Contractive autoencoders explicitly create invariance by adding the Jacobian of the latent space representation w.r.t the input of the autoencoder, to the reconstruction loss $L(\mathbf{x}, r(\mathbf{x}))$. This forces the latent space representation to remain the same for small changes in the input [46]. Let us consider the autoencoder with the encoder mapping the input image to the latent space $\mathbf{z} = f(\mathbf{x})$ and the decoder mapping back to the input image space $r(\mathbf{x}) = g(f(\mathbf{x}))$. The regularized loss function is:

$$\mathcal{L}(W) = \mathbb{E}_{\mathbf{x} \sim X_{\text{train}}} \left[L(\mathbf{x}, r(\mathbf{x})) + \lambda \left\| \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right\| \right] \tag{9}$$

Authors in [47] describe what regularized autoencoders learn from the data generating the density function, and show for contractive and denoising encoders that this corresponds to the direction in which density is increasing the most. Regularization forces the autoencoders to become less sensitive to input variation, though enforcing minimal reconstruction error keeps it sensitive to variations along the manifold having high density. Contractive autoencoders capture variations on the manifold, while mostly ignoring variations orthogonal to it. Contractive autoencoder estimates the tangent plane of the data manifold [26].

3.6. Other Deep Models

De-noising AutoEncoders (DAE) and Stacked DAEs (SDAEs) are well-known robust feature extraction methods in the domain of unsupervised learning [48], where the reconstruction error minimization criteria is augmented with that of reconstructing from corrupted inputs. SDAEs are used to learn representations from a video using both appearance, i.e. raw values, and motion information, i.e. optical flow between consecutive frames [49]. Correlations between optical flow and raw image values are modeled by coupling these two SDAE pipelines to learn a joint representation.

Deep belief networks (DBNs) are generative models, created by stacking multiple hidden layer units, which are usually trained greedily to perform unsupervised feature learning. They are generative models in the sense that they can reconstruct the original inputs. They have been discriminatively trained using back-propagation [50] to achieve improved accuracies for supervised learning tasks. The DBNs have been used to perform a raw image value based representation learning in [51].

An early application of autoencoders to anomaly detection was performed in [52], on non-visual data. The Replicating neural network [52], constitutes of a feed-forward multi-layer perceptron with three hidden layers, trained to map the training dataset to itself and anomalies correspond to large reconstruction error over test datasets. This is an autoencoder setup with a staircase like non-linearity applied at the middle hidden layer. The activation levels of this hidden units are thus quantized into N discrete values, $0, \frac{1}{N-1}, \frac{1}{N-2}, \dots, 1$. The step-wise activation function used for the middle hidden layer divides the continuously distributed data points into a number of discrete-valued vectors. The staircase non-linearity quantizes data points into clusters. This approach identifies cluster labels for each sample, and this often helps interpret resulting outliers.

A rejection cascade over spatio-temporal cubes was generated to improve the performance speed of Deep-CNN based video anomaly detection framework by authors in [53].

Videos can be viewed as a special case of spatio-temporal processes. A direct approach to video anomaly detection can be estimating the spatio-temporal mean and covariance. A major issue is estimating the spatio-temporal covariance matrix due to its large size n^2 (where $n = N \text{ pixels} \times p \text{ frames}$). In [54], space-time pixel covariance for crowd videos were represented as a sum of Kronecker products using only a few Kronecker factors, $\Sigma_{n \times n} \approx \sum_{i=1}^r \mathbf{T}_i \otimes \mathbf{S}_i$. To evaluate the anomaly score, the Mahalanobis distance for clips longer than the learned covariance needs to be evaluated. The inverse of the larger covariance matrix needs to be inferred from the estimated one, by block Toeplitz extension [55]. It is to be noted that this study [54] only evaluates performance on the UMN dataset.

4. Predictive Modeling

Predictive models aim to model the current output frame X_t as a function of the past p frames $[X_{t-1}, X_{t-2}, \dots, X_{t-p+1}]$. This is well-known in time series analysis under auto-regressive models, where the function over the past is linear. Recurrent neural networks (RNN) model this function as a recurrence relationship, frequently involving a non-linearity such as a sigmoid function. LSTM is the standard model for sequence prediction. It learns a gating function over the classical RNN architecture to prevent the vanishing gradient problem during backpropagation through time (BPTT) [3]. Recently there have also been attempts to perform efficient video prediction using feed-forward convolutional networks for video prediction by minimizing the mean-squared error (MSE) between predicted and future frames [9]. Similar efforts were performed in [56] using a CNN-LSTM-deCNN framework while combining MSE and an adversarial loss.

4.1. Composite Model: Reconstruction and Prediction

This composite LSTM model in [7], combines an autoencoder model and predictive LSTM model, see Figure 5. Autoencoders suffer learning trivial representations of input, by memorization, while memorization is not useful for predicting future frames. On the other hand, the future predictor's

role requires memory of temporally past few frames, though this would not be compatible with the autoencoder loss which is more global. The composite model was used to extract features from video data for the tasks of action recognition. The composite LSTM model is defined using a fully connected LSTM (FC-LSTM) layer.

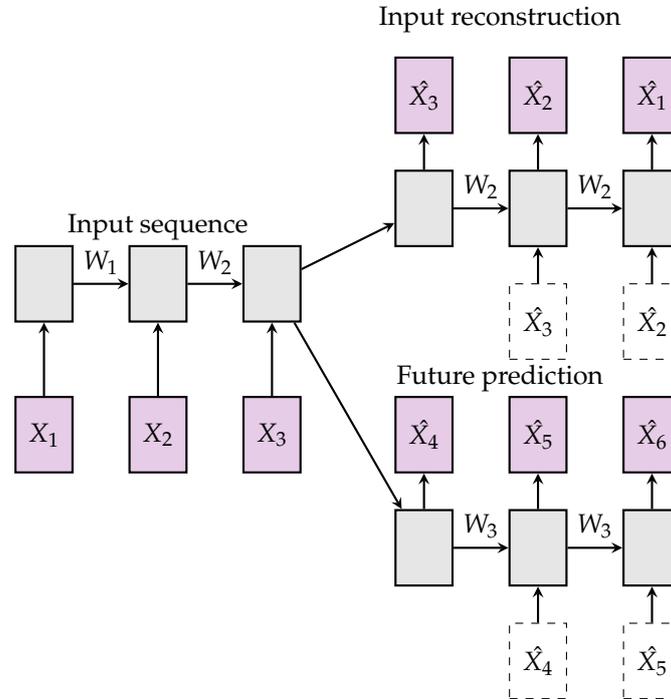


Figure 5. Composite LSTM module [7]. The LSTM model weights W represent fully connected layers.

4.2. Convolutional LSTM

Convolutional long short-term memory (ConvLSTM) model [57] is a composite LSTM based encoder-decoder model. FC-LSTM does not take spatial correlation into consideration and is permutation invariant to pixels, while a ConvLSTM has convolutional layers instead of fully connected layers, thus modeling spatio-temporal correlations. The ConvLSTM as described in Equation (10) evaluates future states of cells in a spatial grid as a function of the inputs and past states of its local neighbors.

Authors in [57], consider a spatial grid, with each grid cell containing multiple spatial measurements, which they aim to forecast for the next K future frames, given J observations in the past. The spatio-temporal correlations are used as input to a recurrent model, the convolutional LSTM. The equations for input, gating and the output are presented below.

$$\begin{aligned}
 i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
 C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} + X_t + W_{hc} * H_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
 H_t &= o_t \circ \tanh(C_t)
 \end{aligned}
 \tag{10}$$

Here, $*$ refers to the convolution operation while \circ refers to the Hadamard product, the element-wise product of matrices. Encoding network compresses the input sequence into a hidden state tensor while the forecasting network unfolds the hidden state tensor to make a prediction.

The hidden representations can be used to represent moving objects in the scene, a larger transitional kernel captures faster motions compared to smaller kernels [57].

The ConvLSTM model was used as a unit within the composite LSTM model [7] following an encoder-decoder, with a branch for reconstruction and another for prediction. This architecture, shown in Figure 6, was applied to the VAD task by [58,59], with promising results.

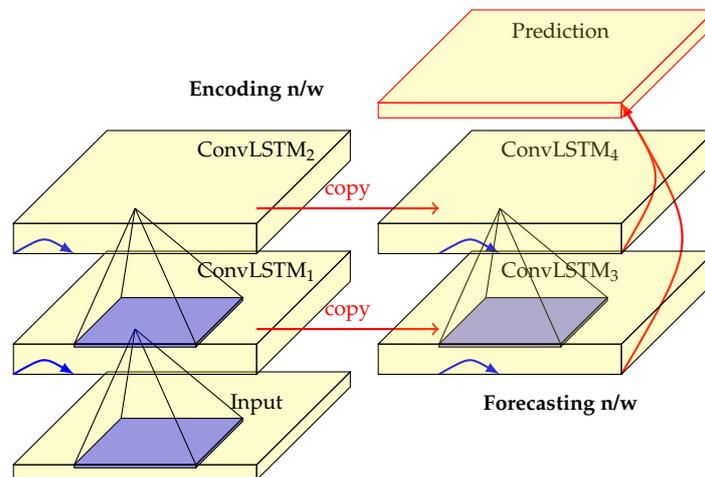


Figure 6. A convolutional LSTM architecture for spatio-temporal prediction.

In [60], a convolutional representation of the input video is used as input to the convolutional LSTM and a de-convolution to reconstruct the ConvLSTM output to the original resolution. The authors call this a ConvLSTM Autoencoder, though fundamentally it is not very different from a ConvLSTM.

4.3. 3D-Autoencoder and Predictor

As remarked by authors in [61], while 2D-ConvNets are appropriate representations learnt for image recognition and detection tasks, they are incapable of capturing the temporal information encoded in consecutive frames for video analysis problems. 3-D convolutional architectures are known to perform well for action recognition [24], and are used in the form of an autoencoder. Such a 3D autoencoder learns representations that are invariant to spatio-temporal changes (movement) encoded by the 3-D convolutional feature maps. Authors in [61] propose to use a 3D kernel by stacking T-frames together as in [41], shown in Figure 7. The output feature map of each kernel is a 3D tensor including the temporal dimension and are aimed to summarize motion information.

The reconstruction branch follows an autoencoder loss:

$$\mathcal{L}_{\text{rec}}(W) = \frac{1}{N} \sum_{i=1}^N \|X_i - f_{\text{rec}}^W(X_i)\|_2^2 \quad (11)$$

The prediction branch loss is inversely weighted by moving window's length that falls off symmetrically w.r.t the current frame, to reduce the effect of past frames on the predicted frame:

$$\mathcal{L}_{\text{pred}}(W) = \frac{1}{N} \sum_{i=1}^N \frac{1}{T^2} \sum_{t=1}^T (T-t) \|X_{i+T} - f_{\text{pred}}^W(X_i)\|_2^2 \quad (12)$$

Thus the final optimization objective minimized is:

$$\mathcal{L}(W) = \mathcal{L}_{\text{rec}}(W) + \mathcal{L}_{\text{pred}}(W) + \lambda \|W\|_2^2 \quad (13)$$

Anomalous regions where spatio-temporal blocks, that even when poorly reconstructed by the autoencoder branch, would be well predicted by the prediction branch. The prediction loss was

designed to enforce local temporal coherence by tracking spatio-temporal correlation, and not for the prediction of the appearance of new objects in the relatively long-term future.

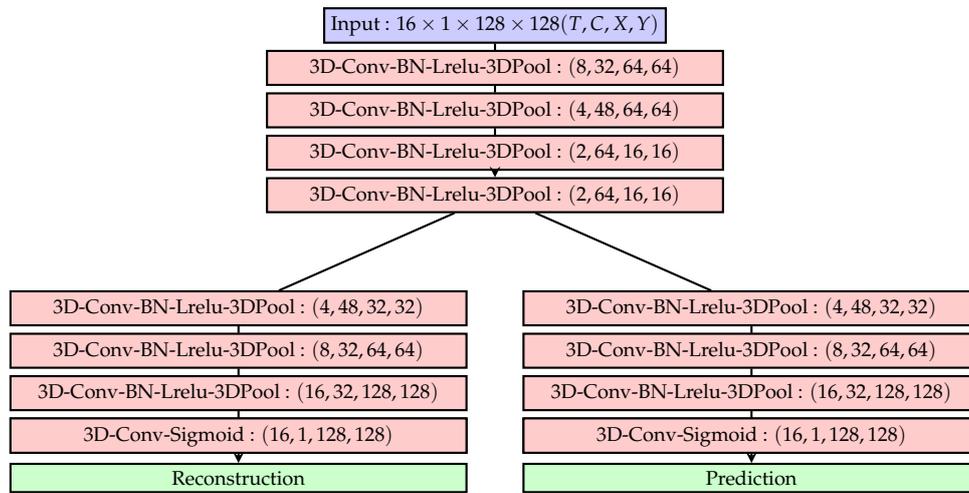


Figure 7. Spatio-temporal autoencoder architecture from [61] with reconstruction and prediction branches, following the composite model in [7]. Batch Normalization(BN) is applied at each layer, following which a leaky Relu non-linearity is applied, finally followed by a 3D max-pooling operation.

4.4. Slow Feature Analysis (SFA)

Slow feature analysis [62] is an unsupervised representation learning method which aims at extracting slowly varying representations of rapidly varying high dimensional input. The SFA is based on the *slowness principle*, which states that the responses of individual receptors or pixel variations are highly sensitive to local variations in the environment, and thus vary much faster, while the higher order internal visual representations vary on a slow timescale. From a predictive modeling perspective SFA extracts a representation $\mathbf{y}(t)$ of the high dimensional input \mathbf{x}_t that maximizes information on the next time sample \mathbf{x}_{t+1} . Given a high dimensional input varying over time, $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T], t \in [t_0, t_1]$ SFA extracts a representation $\mathbf{y} = f_\theta(\mathbf{x})$ which is a solution to the following optimization problem [63]:

$$\arg \min_{\mathbf{E}_t[\mathbf{y}_t]=0, \mathbf{E}_t[\mathbf{y}_t^2]=1, \mathbf{E}_t[\mathbf{y}_t \mathbf{y}_{t'}]=0} \mathbf{E}_t[\mathbf{y}_{t+1} - \mathbf{y}_t] \quad (14)$$

As seen in the constraints, the representation is enforced to have, zero mean to ensure a unique solution, while unit covariance to avoid trivial zero solution. Feature de-correlation removes redundancy across the features.

SFA has been well known in pattern recognition and has been applied to the problem of activity recognition [64,65]. Authors in [66] propose an incremental application of SFA that updates slow features incrementally. The SFA is calculated using batch PCA, iterated twice. The first PCA to whiten the inputs. The second PCA is applied on the derivative of the normalized input to evaluate the flow features. To achieve a computationally tractable solution, a two-layer localized SFA architecture is proposed by authors [67] for the task of online slow feature extraction and consequent anomaly detection.

Other Predictive Models: A convolutional feature representation was fed into an LSTM model to predict the latent space representation and its prediction error was used to evaluate anomalies in a robotics application [68]. A recurrent autoencoder using an LSTM that models temporal dependence between patches from a sequence of input frames is used to detect video forgery [69].

5. Deep Generative Models

5.1. Generative vs. Discriminative

Let us consider a supervised learning setup $(X_i, y_i) \in \mathbb{R}^d \times \{C_j\}_{j=1}^K$, where i indexes the number of samples $i = 1 : N$ in the dataset. Generative models estimate class conditional posterior distribution $P(X|y)$, which can be difficult if the input data are high dimensional images or spatio-temporal tensors. A discriminative model evaluates the class probability $P(y|X)$ directly from the data to classify the samples X into different classes $\{C_j\}_{j=1}^K$.

Deep generative models that can learn via the principle of maximum likelihood differ with respect to how they represent or approximate the likelihood. The explicit models are ones where the density $p_{\text{model}}(\mathbf{x}, \theta)$ is evaluated explicitly and the likelihood maximized.

In this section, we will review the stochastic autoencoders; the variational autoencoder and the adversarial autoencoder, and their applications to the problem of anomaly detection. And finally the generative adversarial networks to anomaly detection in images and videos.

5.2. Variational Autoencoders (VAEs)

Variational Autoencoders [1] are generative models that approximate the data distribution $P(X)$ of a high dimensional input X , an image or video. The graphical mode of the VAE is reproduced from [1] in Figure 8. A variational approximation of the latent space \mathbf{z} is achieved using an autoencoder architecture, with a probabilistic encoder $q_\phi(\mathbf{x}|\mathbf{z})$ that produces Gaussian distribution in the latent space, and a probabilistic decoder $p_\theta(\mathbf{z}|\mathbf{x})$, which given a code produces distribution over the input space. The motivation behind variational methods is to pick a family of distributions over the latent variables with its own variational parameters $q_\phi(\mathbf{z})$, and estimate the parameters for this family so that it approaches q_ϕ .

The loss function constitutes of the KL-Divergence regularization term, and the expected negative reconstruction error with an additional KL-divergence term between the latent space vector and the representation with a mean vector and a standard deviation vector, that optimizes the variational lower bound on the marginal log-likelihood of each observation.

$$\mathcal{L}(\theta, \phi, \mathbf{x}^{(i)}) = \sum_{i=1}^{N_{\text{train}}} -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) + \frac{1}{L} \sum_{l=1}^L (\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})) \quad (15)$$

$$\text{where } \mathbf{z}^{(i,l)} = g_\phi(\epsilon^{(i,l)}, \mathbf{x}^{(i)}) \text{ and } \epsilon^{(i,l)} \sim p(\epsilon)$$

The function g_ϕ maps sample $\mathbf{x}^{(i)}$ and noise vector $\epsilon^{(i)}$ to a sample from the approximate posterior for that data-point $\mathbf{z}^{(i,l)} = g_\phi(\epsilon^{(i)}, \mathbf{x}^{(i)})$ where, $\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$. To solve this sampling problem authors [1] propose the reparameterization trick. The random variable $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ is expressed as function of a deterministic variable $\mathbf{z} = g_\phi(\epsilon, \mathbf{x})$ where ϵ is an auxiliary variable with independent marginal $p(\epsilon)$. This reparameterization rewrites an expectation w.r.t $q_\phi(\mathbf{z}|\mathbf{x})$ such that the Monte Carlo estimate of the expectation is differentiable w.r.t. ϕ . A valid reparameterization was the unit-Gaussian case $\mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i,l)} + \sigma^{(i)} \odot \epsilon^{(i,l)}$ where $\epsilon^{(i,l)} \sim \mathcal{N}(0, I)$.

Specifically for a VAE, the goal is to learn a low dimensional representation \mathbf{z} by modeling $p_\theta(\mathbf{x}|\mathbf{z})$ with a simpler distribution, a centered isotropic multivariate Gaussian, i.e. $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}, \mathbf{0}, I)$. In this model both the prior $p_\theta(\mathbf{z})$, and $q_\phi(\mathbf{z}|\mathbf{x})$ are Gaussian; and the resulting loss function was described in Equation (15).

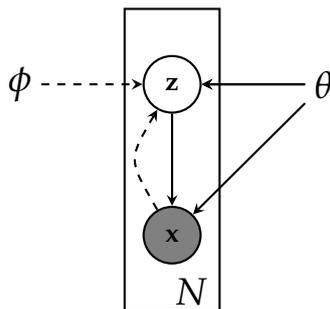


Figure 8. Graphical model for the VAE: Solid lines denote the generative model $p_{\theta}(\mathbf{z})p_{\theta}(\mathbf{x}|\mathbf{z})$, dashed lines denote the variational approximation $q_{\phi}(\mathbf{z}|\mathbf{x})$ to the intractable posterior $p_{\theta}(\mathbf{z}|\mathbf{x})$ [1].

5.3. Anomaly Detection Using VAE

Anomaly detection using the VAE framework has been studied in [70]. Authors define the reconstruction probability as $\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})}[\log p_{\theta}(\mathbf{x}|\mathbf{z})]$. Once the VAE is trained, for a new test sample $\mathbf{x}^{(i)}$, one first evaluates the mean and standard deviation vectors with the probabilistic encoder, $(\boldsymbol{\mu}_{\mathbf{z}^{(i)}}, \boldsymbol{\sigma}_{\mathbf{z}^{(i)}}) = f_{\theta}(\mathbf{z}|\mathbf{x}^{(i)})$. Then samples L latent space vectors, $\mathbf{z}^{(i,l)} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}^{(i)}}, \boldsymbol{\sigma}_{\mathbf{z}^{(i)}})$. The parameters of the input distribution are reconstructed using these L samples, $\boldsymbol{\mu}_{\hat{\mathbf{x}}^{(i,l)}}, \boldsymbol{\sigma}_{\hat{\mathbf{x}}^{(i,l)}} = g_{\phi}(\mathbf{x}|\mathbf{z}^{(i,l)})$ then the reconstruction probability for test sample $\mathbf{x}^{(i)}$ is given by:

$$P_{\text{recon}}(\mathbf{x}^{(i)}) = \frac{1}{L} \sum_{l=1}^L p_{\theta}(\mathbf{x}^{(i)}|\boldsymbol{\mu}_{\hat{\mathbf{x}}^{(i,l)}}, \boldsymbol{\sigma}_{\hat{\mathbf{x}}^{(i,l)}}) \tag{16}$$

Multiple samples drawn from the latent variable distribution, lets $P_{\text{recon}}(\mathbf{x}^{(i)})$ take into account the variability of the latent variable space, which is one of the essential distinctions between the stochastic variational autoencoder and a standard autoencoder, where latent variables are defined by deterministic mappings.

5.4. Generative Adversarial Networks (GANs)

A GAN [2] consists of a generator G , usually a decoder, and a discriminator D , usually a binary classifier that assigns a probability of an image being generated (fake), or sampled from the training data (real). The generator G in fact learns a distribution p_g over data \mathbf{x} via a mapping $G(\mathbf{z})$ of samples \mathbf{z} , 1D vectors of uniformly distributed input noise sampled from latent space \mathcal{Z} , to 2D images in the image space manifold \mathcal{X} , which is populated by normal examples. In this setting, the network architecture of the generator G is equivalent to a convolutional decoder that utilizes a stack of strided convolutions. The discriminator D is a standard CNN that maps a 2D image to a single scalar value $D(\cdot)$. The discriminator output $D(\cdot)$ can be interpreted as the probability that the given input to the discriminator D was a real image \mathbf{x} sampled from training data X or a generated image using $G(\mathbf{z})$ by the generator G . D and G are simultaneously optimized through the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))] \tag{17}$$

The discriminator is trained to maximize the probability of assigning real training examples the “real” and samples from p_g the “fake” label. The generator G is simultaneously trained to fool D via minimizing $V(G) = \log(1 - D(G(\mathbf{z})))$, which is equivalent to maximizing $V(G) = D(G(\mathbf{z}))$. During adversarial training, the generator improves in generating realistic images and the discriminator progresses in correctly identifying real and generated images. GANs are implicit models [71], that sample directly from the distribution represented by the model.

5.5. GANs for Anomaly Detection in Images

This section reviews work done by authors [72] who apply a GAN model for the task of anomaly detection in medical images. GANs are generative models that best produce a set of training data points $\mathbf{x} \sim P_{\text{data}}(\mathbf{x})$ where P_{data} represents the probability density of the training data points. The basic idea in anomaly detection is to be able to evaluate the density function of the normal vectors in the training set containing no anomalies while for the test set we evaluate a negative log-likelihood score which serves as the final anomaly score. The score corresponds to the test sample's posterior probability of being generated from the same generative model representing the training data points. GANs provide a generative model that minimizes the distance between the training data distribution and the generative model samples without explicitly defining a parametric function, which is why it is called an implicit generative model [71]. Thus to be successfully used in an anomaly detection framework the authors [72] evaluate the mapping $\mathbf{x} \rightarrow \mathbf{z}$, i.e., Image domain \rightarrow latent representation. This was done by choosing the closest point \mathbf{z}_γ using back-propagation. Once done the residual loss in the image space was defined as $\mathcal{L}_R(\mathbf{z}_\gamma) = \sum |\mathbf{x} - G(\mathbf{z}_\gamma)|$.

GANs are generative models and to evaluate a likelihood one requires a mapping from the image domain to the latent space. This is achieved by authors in [72], which we shall shortly describe here. Given a query image $\mathbf{x} \sim p_{\text{test}}$, the authors aim to find a point \mathbf{z} in the latent space that corresponds to an image $G(\mathbf{z})$ that is visually most similar to the query image \mathbf{x} and that is located on the manifold \mathcal{X} . The degree of similarity of \mathbf{x} and $G(\mathbf{z})$ depends on to which extent the query image follows the data distribution p_g that was used for training of the generator.

To find the best \mathbf{z} , one starts randomly sampling \mathbf{z}_1 from the latent space distribution \mathcal{Z} and feeds it into the trained generator which yields the generated image $G(\mathbf{z}_1)$. Based on the generated image $G(\mathbf{z}_1)$ we can define a loss function, which provides gradients for the update of the coefficients of \mathbf{z}_1 resulting in an updated position in the latent space, \mathbf{z}_2 . In order to find the most similar image $G(\mathbf{z}_\Gamma)$, the location of \mathbf{z} in the latent space \mathcal{Z} is optimized in an iterative process via $\gamma = 1, 2, \dots, \Gamma$ back-propagation steps.

5.6. Adversarial Discriminators Using Cross-Channel Prediction

Here we shall review the work done in [73] applied to anomaly detection in videos. The anomaly detection problem in this paper is formulated as a cross-channel prediction task, where the two channels are the raw-image values F_t and the optical flow vectors O_t for frames F_t, F_{t-1} in the videos. This work combines two architectures, the pixel-GAN architecture by [74] to model the normal/training data distribution, and the Split-Brain Autoencoders [75]. The Split-Brain architectures aims at predicting a multi-channel output by building cross-channel autoencoders. That is, given training examples $X \in \mathbb{R}^{H \times W \times C}$, we split data into $X_1 \in \mathbb{R}^{H \times W \times C_1}$ and $X_2 \in \mathbb{R}^{H \times W \times C_2}$, where $C_1, C_2 \subset C$, and the authors train multiple deep representations $\tilde{X}_2 = \mathcal{F}_1(X_1)$ and $\tilde{X}_1 = \mathcal{F}_2(X_2)$, which when concatenated provided a reconstruction of the input tensor X , just like an autoencoder. Various manners of aggregating these predictors have been explored in [75]. In the same spirit as the cross-channel autoencoders [75], Conditional GANs were developed [74] to learn a generative model that learns a mapping from one input domain to the other.

The authors [73] train two networks much in the spirit of the conditional GAN [74] where: $\mathcal{N}^{O \rightarrow F}$ which generates the raw image frames from the optical flow and $\mathcal{N}^{F \rightarrow O}$ which generates the optical flow from the raw images. F_t are image frames with RGB channels and O_t are vertical and horizontal optical flow vector arrays. The input to discriminator D is thus a 6-D tensor. We now describe the adaption of the cross-channel autoencoders for the task of anomaly detection.

- $\mathcal{N}^{F \rightarrow O}$: Training set is $X = \{(F_t, O_t)\}_{t=1}^N$. The L1 loss function with $x = F_t, y = O_t$:

$$\mathcal{L}_{L1}(x, y) = \|y - G(x, z)\|_1 \tag{18}$$

with the conditional adversarial loss being

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{(x,y) \in X} [\log D(x, y)] + \mathbb{E}_{x \in \{F_t\}, z \in Z} \log(1 - D(x, G(x, z))) \quad (19)$$

- Conversely in $\mathcal{N}^{O \rightarrow F}$ the training set changes to $X = \{(O_t, F_t)\}_{t=1}^N$.

The generators/discriminators follow a U-net architecture as in [74] with skip connections. The two generators $G^{F \rightarrow O}, G^{O \rightarrow F}$ are trained to map training frames and their optical flow to their cross-channel counterparts. The goal is to force a poor cross-channel prediction on test video frames containing an anomaly so that the trained discriminators shall provide a low probability score.

The trained discriminators $D^{F \rightarrow O}, D^{O \rightarrow F}$ are patch-discriminators that produce scores S^O, S^F on a grid with resolution smaller than the image. These scores do not require the reconstruction of the different channels to be evaluated. The final score is $S = S^O + S^F$ which is normalized between $[0, 1]$ based on the maximum value of individual scores for each frame. The U-net uses the Markovian structure present spatially by the skip connections shown between the input and the output of the generators in Figure 9. Cross-channel prediction aims at modeling the spatio-temporal correlation present across channels in the context of video anomaly detection.

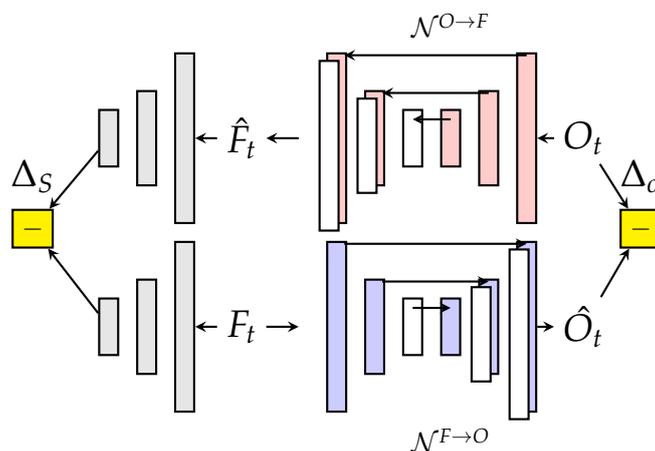


Figure 9. The cross-channel prediction conditional GAN architecture in [73]. There are two GAN models: flow→RGB predictor ($\mathcal{N}^{O \rightarrow F}$) and RGB→Flow predictor ($\mathcal{N}^{F \rightarrow O}$). Each of the generators shown has a U-net architecture which uses the common underlying structure in the image RGB channels and optical flow between two frames.

5.7. Adversarial Autoencoders (AAEs)

Adversarial Autoencoders are probabilistic autoencoders that use GANs to perform variational approximation of the aggregated posterior of the latent space representation [76] using an arbitrary prior.

AAEs were applied to the problem of anomalous event detection over images by authors [77]. In Figure 10, \mathbf{x} denotes input vectors from training distribution, $q(\mathbf{z}|\mathbf{x})$ the encoder’s posterior distribution, $p(\mathbf{z})$ the prior that the user wants to impose on the latent space vectors \mathbf{z} . The latent space distribution is given by

$$q(\mathbf{z}) = \int_{\mathbf{x} \in X_{\text{train}}} q(\mathbf{z}|\mathbf{x}) p_d(\mathbf{x}) d\mathbf{x} \quad (20)$$

where $p_d(\mathbf{x})$ represents the training data distribution. In an AAE, the encoder acts like the generator of the adversarial network, and it tries to fool the discriminator into believing that $q(\mathbf{z})$ comes from the actual data distribution $p(\mathbf{z})$. During the joint training, the encoder is updated to improve the reconstruction error in the autoencoder path, while it is updated by the discriminator of the adversarial

network to make the latent space distribution approach the imposed prior. As prior distribution for the generator of the network, authors [77] use the Gaussian distribution of 256 dimensions, with the dropout set to 0.5 probability. The method achieves close to state of the art performance. As the authors remark themselves, the AAE does not take into account the temporal structure in the video sequences.

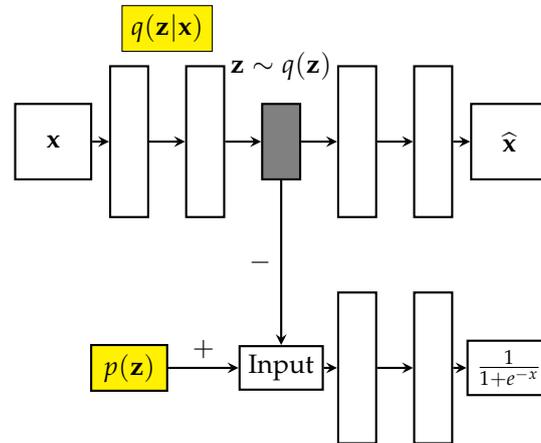


Figure 10. Two paths in the Adversarial Autoencoder : Top path refers to the standard autoencoder configuration that minimizes reconstruction error. The bottom path constitutes of an adversarial network that ensures an approximation of the user input defined samples from distribution $p(z)$, and the latent or code vector distribution, provided by $q(z|x)$.

5.8. Controlling Reconstruction for Anomaly Detection

One of the common problems using deep autoencoders is their capability to produce low reconstruction errors for test samples, even over anomalous events. This is due to the way autoencoders are trained in a semi-supervised way on videos with no anomalies, but with sufficient training samples, they are able to approximate most test samples well.

In [78], the authors propose to limit the reconstruction capability of the generative adversarial networks by learning conflicting objectives for the normal and anomalous data. They use negative examples to enforce explicit poor reconstruction. Thus this setup is weakly supervised, not requiring labels. Given two random variables X, Y with samples $\{x\}_{i=1}^K, \{y\}_{j=1}^J$, we want the network to reconstruct the input distribution X while poorly reconstruct Y . This was achieved by maximizing the following objective function:

$$p_{\theta}(\hat{X}|X) - p_{\theta}(\hat{Y}|Y) = \sum_{i=1}^K \log p_{\theta}(\hat{x}_i|x_i) + \sum_{j=1}^J \log p_{\theta}(\hat{y}_j|y_j) \tag{21}$$

where θ refers to the autoencoders parameters. This setup assumes strong class imbalance, i.e.very few samples of the anomalous class Y are available compared to the normal class X . The motivation for negative learning using anomalous examples is to consistently provide poor reconstruction of anomalous samples. During the training phase, authors [68] reconstruct positive samples by minimizing the reconstruction error between samples, while negative samples are forced to have a bad reconstruction by maximizing the error. This last step was termed as negative learning. The datasets evaluated were the reconstruction of the images from MNIST and Japanese highway video patches [79].

In similar work by [80], discriminative autoencoders aim at learning low-dimensional discriminative representations for positive (X^+) and negative (X^-) classes of data. The discriminative autoencoders build a latent space representation under the constraint that the positive data should be

better reconstructed than the negative data. This is done by minimizing the reconstruction error for positive examples while ensuring that those of the negative class are pushed away from the manifold.

$$\mathcal{L}_d(X^+ \cup X^-) = \sum_{\mathbf{x} \in X^+ \cup X^-} \max(0, t(\mathbf{x}) \cdot (\|\hat{\mathbf{x}} - \mathbf{x}\| - 1)) \tag{22}$$

In the above loss function, $t(\mathbf{x}) = \{-1, +1\}$ denotes as the label of the sample, and $e(\mathbf{x}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|$ the distance of that example to the manifold. Minimizing the hinge loss in Equation (22) achieves reconstruction such that the discriminative autoencoders build a latent space representation of data that better reconstructs positive data compared to the negative data

6. Experiments

There are two large classes of experiments : first, reconstructing the input video on a single frame basis $X_t \rightarrow \tilde{X}_t$, second the reconstruction of a stack of frames $X_{t-p:t} \rightarrow \tilde{X}_{t-p:t}$. These reconstruction schemes are performed either on raw frame values, or on the optical flow between consequent frame pairs. Reconstructing raw image values modeled the back-ground image, since minimizing the reconstruction error was in fact evaluating the background. Convolutional autoencoders reconstructing a sequence of frames captured temporal appearance changes as described by [41]. When learning feature representations on optical flow we indirectly operate on two frames, since each optical flow map evaluates the relative motion between two consequent frame pairs. In the case of predictive models the current frame X_t was predicted after observing the past p frames. This provides a different temporal structure as compared to a simple reconstruction of a sequence of frames $X_{t-p:t} \rightarrow \tilde{X}_{t-p:t}$, where the temporal coherence results from enforcing a bottleneck in the autoencoder architectures. The goal of these experiments were not evaluate the best performing model, and were intended as a tool to understand how background estimation and temporal appearance were approximated by the different models. A complete detailed study is beyond the scope of this review.

In this section, we evaluate the performance of the following classes of models on the UCSD and CUHK-Avenue datasets. As a baseline, we use the reconstruction of a dense optical flow calculated using the Farneback method in OpenCV 3, by principal component analysis, with around 150 components. For predictive models, as a baseline we use a vector autoregressive model (VAR), referred to as *LinPred*. The coefficients of the model are estimated on a lower dimensional, random projection of the raw image or optical flow maps from the input training video stream. The random projection avoids badly conditioned and expensive matrix inversion. We compare the performance of Contractive autoencoders, simple 3D autoencoders based on C3D [24] CNNs (C3D-AE), the ConvLSTM and ConvLSTM autoencoder from the predictive model family and finally the VAE from the generative models family. The VAE’s loss function consists of the binary cross-entropy (similar to a reconstruction error) between the model prediction and the input image, and the KL-divergence $D_{KL}[Q(z|X)||P(z)]$, between the encoded latent space vectors and $P(z) = \mathcal{N}(0, I)$ multivariate unit-Gaussian. These models were built in Keras [81] with Tensorflow back-end and executed on a K-80 GPU.

6.1. Architectures

Our Contractive and Variational AE (VAE) constitutes of a random projection to reduce the dimensionality to 2500 from an input frame of size 200×200 . The Contractive AE constitutes of one fully connected hidden layer of size 1250 which map back to the reconstruction of the randomly projected vector of size 2500. While the VAE contains two hidden layers (dimensions: 1024, 32), which maps back to the output of 2500 dimensions. We use the latent space representation of the VAE to fit a multivariate Gaussian on the training dataset and evaluate the negative-log probability for the test samples.

6.2. Observations and Issues

The results are summarized in the Tables 1 and 2. The performance measures reported are the Area Under Receiver-Output-Characteristics plot (AU-ROC), Area Under Precision-Recall plot. These scores are calculated when the input channels correspond to the raw image (raw) and the optical flow (flow), each of which has been normalized by the maximum value. The final temporal anomaly score is given by Equation (8). These measures are described in the next section. We also describe the utility of these measures under different frequencies of occurrences of the anomalous positive class.

Table 1. Area Under-ROC (AUROC).

Methods Feature	PCA Flow	LinPred (Raw, Flow)	C3D-AE (Raw, Flow)	ConvLSTM (Raw, Flow)	ConvLSTM-AE (Raw, Flow)	CtractAE (Raw, Flow)	VAE (Raw, Flow)
UCSDped1	0.75	(0.71, 0.71)	(0.70, 0.70)	(0.67, 0.71)	(0.43, 0.74)	(0.66, 0.75)	(0.63, 0.72)
UCSDped2	0.80	(0.73, 0.78)	(0.64, 0.81)	(0.77, 0.80)	(0.25, 0.81)	(0.65, 0.79)	(0.72, 0.86)
CUHK-avenue	0.82	(0.74, 0.84)	(0.86, 0.18)	(0.84, 0.18)	(0.50, 0.84)	(0.83, 0.84)	(0.78, 0.80)

Table 2. Area Under-Precision Recall (AUPR).

Methods Feature	PCA Flow	LinPred (Raw, Flow)	C3D-AE (Raw, Flow)	ConvLSTM (Raw, Flow)	ConvLSTM-AE (Raw, Flow)	CtractAE (Raw, Flow)	VAE (Raw, Flow)
UCSDped1	0.78	(0.71, 0.71)	(0.75, 0.77)	(0.67, 0.71)	(0.52, 0.81)	(0.70, 0.81)	(0.66, 0.76)
UCSDped2	0.95	(0.92, 0.94)	(0.88, 0.95)	(0.94, 0.95)	(0.74, 0.95)	(0.88, 0.95)	(0.92, 0.97)
CUHK-avenue	0.66	(0.49, 0.65)	(0.65, 0.15)	(0.66, 0.15)	(0.34, 0.70)	(0.69, 0.70)	(0.54, 0.68)

Reconstruction model issues: Deep autoencoders identify anomalies by poor reconstruction of objects that have never appeared in the training set, when raw image pixels are used as input. It is difficult to achieve this in practice due to a stable reconstruction of new objects by deep autoencoders. This pertains to the high capacity of autoencoders, and their tendency to well approximate even the anomalous objects. Controlling reconstruction using negative examples could be a possible solution. This holds true, but to a lower extent, when reconstructing a sequence of frames (spatio-temporal block).

AUC-ROC vs. AUC-PR: The anomalies in the UCSD pedestrian dataset have a duration of several hundred frames on average, compared to the anomalies in the CUHK avenue dataset which occur only for a few tens of frames. This makes the anomalies statistically less probable. This can be seen by looking at the AU-PR Table 2, where the average scores for CUHK-avenue are much lower than for UCSD pedestrian datasets. It is important to note that this does not mean the performance over the CUHK-Avenue dataset is lower, but just the fact that the positive anomalous class is rarer in occurrence.

Rescaling image size: The models used across different experiments in the articles that were reviewed, varied in the input image size. In some cases the images were resized to sizes (128, 128), (224, 224), (227, 227). We have tried to fix this to be uniformly (200, 200). Though it is essential to note that there is a substantial change in performance when this image is resized to certain sizes.

Generating augmented video clips: Training the convolutional LSTM for video anomaly detection takes a large number of epochs. Furthermore, the training video data required is much higher and data augmentation for video anomaly detection requires careful thinking. Translations and rotations may be transformations to which the anomaly detection algorithm requires to be sensitive to, based on the surveillance application.

Performance of models: VAEs perform consistently as well as or better than PCA on optical flow. It is still left as a future study to understand clearly, why the performance of a stochastic autoencoder such as VAE is better. Convolutional LSTM on raw image values follow closely behind as the first predictive model performing as good as PCA but sometimes poorer. Convolutional LSTM-AE is a similar architecture with similar performance. Finally, the 3D convolutional autoencoder, based on the work by [24], performs as well as PCA on optical flow, while modeling local motion patterns.

To evaluate the specific advantages of each of these models, a larger number of real world, video surveillance examples are required demonstrating the representation or feature that is most discriminant. In experiments, we have also observed that application of PCA on the random projection of individual frames performed well in avenue dataset, indicating that very few frames were sufficient to identify the anomaly; while the PCA performed poorly on UCSD pedestrian datasets, where motion patterns were key to detect the anomalies.

For single frame input based models, optical flow served as a good input since it already encoded part of the predictive information in the training videos. On the other hand, convolutional LSTMs and linear predictive models required $p = [2, 10]$ input raw image values, in the training videos, to predict the current frame raw image values.

6.3. Evaluation Measures

The anomaly detection task is a single class estimation task, where 0 is assigned to samples with likelihood (or reconstruction error) above (below) a certain threshold, and 1 assigned to detected anomalies with low likelihood (high reconstruction error). Statistically, the anomalies are a rare class and occurs less frequently compared to the normal class. The most common characterization of this behavior is the expected frequency of occurrence of anomalies. We briefly review the anomaly detection evaluation procedure as well as the performance measures that were used across different studies. For a complete treatment of the subject, the reader is referred to [82].

The final anomaly score is a value that is treated as a probability which lies in $s(t) \in [0, 1], \forall t, t \in [1, T], T$ being the maximum time index. For various level sets or thresholds of the anomaly score $a_{1:T}$, one can evaluate the True Positives (TP, the samples which are truly anomalous and detected as anomalous), True Negatives (TN, the samples that are truly normal and detected as normal), False Positives (FP, the samples which are truly normal samples but detected as anomalous) and finally False Negatives (FN, the samples which are truly anomalous but detected as normal).

$$\begin{aligned} \text{True positive rate (TPR) or Recall} &= \frac{\sum TP}{TP+FN} \\ \text{False positive rate (FPR)} &= \frac{\sum FP}{TP+FN} \\ \text{Precision} &= \frac{\sum TP}{TP+FP} \end{aligned} \quad (23)$$

We require these measures to evaluate the ROC curve, which measures the performance of the detection at various False positive rates. That is ROC plots TPR vs FPR, while the PR plots precision vs. recall.

The performance of anomaly detection task is evaluated based on an important criterion, the probability of occurrence of the anomalous positive class. Based on this value, different performance curves are useful. We define the two commonly used performance curves: *Precision-Recall* (PR) curves and *Receiver-Operator-Characteristics* (ROC) curves. The area under the PR curve (AU-PR) is useful when true negatives are much more common than true positives (i.e., $TN \gg TP$). The precision recall curve only focuses on predictions around the positive (rare) class. This is good for anomaly detection because predicting true negatives (TN) is easy in anomaly detection. The difficulty is in predicting the rare true positive events. Precision is directly influenced by class (im)balance since FP is affected, whereas TPR only depends on positives. This is why ROC curves do not capture such effects. Precision-recall curves are better to highlight differences between models for highly imbalanced data sets. For this reason, if one would like to evaluate models under imbalanced class settings, AU-PR scores would exhibit larger differences than the area under the ROC curve.

7. Conclusions

In this review paper, we have focused on categorizing the different unsupervised learning models for the task of anomaly detection in videos into three classes based on the prior information used to build the representations to characterize anomalies.

They are reconstruction based, spatio-temporal predictive models, and generative models. Reconstruction based models build representations that minimize the reconstruction error of training samples from the normal distribution. Spatio-temporal predictive models take into account the spatio-temporal correlation by viewing videos as a spatio-temporal time series. Such models are trained to minimize the prediction error on spatio-temporal sequences from the training series, where the length of the time window is a parameter. Finally, the generative models learn to generate samples from the training distribution, while minimizing the reconstruction error as well as distance between generated and training distribution, where the focus is on modeling the distance between sample and distributions.

Each of these methods focuses on learning certain prior information that is useful for constructing the representation for the video anomaly detection task. One key concept which occurs in various architectures for video anomaly detection is how temporal coherence is implemented. Spatio-temporal autoencoders and Convolutional LSTM learn a reconstruction based or spatio-temporal predictive model that both use some form of (not explicitly defined) spatio-temporal regularity assumptions. We can conclude from our study that evaluating how sensitive the learned representation is to certain transformations such as time warping, viewpoint, applied to the input training video stream, is an important modeling criterion. Certain invariances are as well defined by the choice of the representation (translation, rotation) either due to reusing convolutional architectures or imposing a predictive structure. A final component in the design of the video anomaly detection system is the choice of thresholds for the anomaly score, which was not covered in this review. The performance of the detection systems were evaluated using ROC plots which evaluated performance across all thresholds. Defining a spatially variant threshold is an important but non-trivial problem.

Finally, as more data is acquired and annotated in a video-surveillance setup, the assumption of having no labeled anomalies progressively turns false, partly discussed in the section on controlling reconstruction for anomaly detection. Certain anomalous points with well defined spatio-temporal regularities become a second class that can be estimated well; and methods to include the positive anomalous class information into detection algorithms becomes essential. Handling class imbalance becomes essential in such a case.

Another problem of interest in videos is the variation in temporal scale of motion patterns across different surveillance videos, sharing a similar background and foreground. Learning a representation that is invariant to such time warping would be of practical interest.

There are various additional components of the stochastic gradient descent algorithm that were not covered in this review. The Batch Normalization [83] and drop-out based regularization [84] play an important role in the regularization of deep learning architectures, and a systematic study is important to be successful in using them for video anomaly detection.

Acknowledgments: The authors would like to thank Benjamin Crouzier for his help in proof reading the manuscript, and Y. Senthil Kumar (Valeo) for helpful suggestions. The authors would also like to thank their employer to perform fundamental research.

Author Contributions: All authors conceived and designed the experiments, and the redaction of the article. Experiments were performed by B Ravi Kiran.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Diederik, P.K.; Max, W. Stochastic Gradient VB and the Variational Auto-Encoder. In Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
2. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 8–13 December 2014; pp. 2672–2680.

3. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
4. Fayyaz, M.; Saffar, M.H.; Sabokrou, M.; Fathy, M.; Huang, F.; Klette, R. STFCN: Spatio-Temporal Fully Convolutional Neural Network for Semantic Segmentation of Street Scenes. In Proceedings of the ACCV 2016 International Workshops, Taipei, Taiwan, 20–24 November 2016; pp. 493–509.
5. Mahadevan, V.; Li, W.X.; Bhalodia, V.; Vasconcelos, N. Anomaly Detection in Crowded Scenes. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1975–1981.
6. Li, W.; Mahadevan, V.; Vasconcelos, N. Anomaly Detection and Localization in Crowded Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 18–32.
7. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using lstms. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 843–852.
8. Vondrick, C.; Pirsivash, H.; Torralba, A. Generating videos with scene dynamics. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 613–621.
9. Mathieu, M.; Couprie, C.; LeCun, Y. Deep multi-scale video prediction beyond mean square error. *arXiv* **2015**, arXiv:1511.05440.
10. Lu, C.; Shi, J.; Jia, J. Abnormal Event Detection at 150 FPS in Matlab. In Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, Australia, 1–8 December 2013.
11. UMN. Unusual Crowd Activity Dataset. Available online: <http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi> (accessed on 1 October 2017).
12. Zaharescu, A.; Wildes, R. Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 563–576.
13. Benezeth, Y.; Jodoin, P.M.; Saligrama, V.; Rosenberger, C. Abnormal events detection based on spatio-temporal co-occurrences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2458–2465.
14. Leyva, R.; Sanchez, V.; Li, C.T. The LV dataset: A realistic surveillance video dataset for abnormal event detection. In Proceedings of the 2017 5th International Workshop on Biometrics and Forensics (IWBF), Coventry, UK, 4–5 April 2017; pp. 1–6.
15. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*; Cambridge University Press: Kwun Tong, UK, 2014.
16. Chong, Y.S.; Tay, Y.H. Modeling Representation of Videos for Anomaly Detection using Deep Learning: A Review. *arXiv* **2015**, arXiv:1505.00523.
17. Popoola, O.P.; Wang, K. Video-based abnormal human behavior recognition: A review. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 865–878.
18. Li, T.; Chang, H.; Wang, M.; Ni, B.; Hong, R.; Yan, S. Crowded scene analysis: A survey. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 367–386.
19. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Li, F.F.; Savarese, S. Social lstm: Human trajectory prediction in crowded spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 961–971.
20. Helbing, D.; Molnar, P. Social force model for pedestrian dynamics. *Phys. Rev. E* **1995**, *51*, 4282–4286.
21. Kantorov, V.; Laptev, I. Efficient feature extraction, encoding and classification for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2593–2600.
22. Baccouche, M.; Mamalet, F.; Wolf, C.; Garcia, C.; Baskurt, A. Sequential deep learning for human action recognition. In Proceedings of the Second International Workshop, HBU 2011, Amsterdam, The Netherlands, 16 November 2011; pp. 29–39.
23. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Li, F. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
24. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision, Cambridge, UK, 22–24 July 2015; pp. 4489–4497.

25. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828.
26. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: <http://www.deeplearningbook.org> (accessed on 1 October 2017).
27. Saligrama, V.; Chen, Z. Video anomaly detection based on local statistical aggregates. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2112–2119.
28. Abuolaim, A.A.; Leow, W.K.; Varadarajan, J.; Ahuja, N. On the Essence of Unsupervised Detection of Anomalous Motion in Surveillance Videos. In Proceedings of the 17th International Conference, CAIP 2017, Ystad, Sweden, 22–24 August 2017; pp. 160–171.
29. Basharat, A.; Gritai, A.; Shah, M. Learning object motion patterns for anomaly detection and improved object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
30. Zhao, B.; Li, F.F.; Xing, E.P. Online Detection of Unusual Events in Videos via Dynamic Sparse Coding. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011.
31. Adam, A.; Rivlin, E.; Shimshoni, I.; Reinitz, D. Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 555–560.
32. Wang, T.; Snoussi, H. Histograms of optical flow orientation for abnormal events detection. In Proceedings of the 2013 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), Clearwater, FL, USA, 15–17 January 2013.
33. Colque, R.V.H.M.; Júnior, C.A.C.; Schwartz, W.R. Histograms of Optical Flow Orientation and Magnitude to Detect Anomalous Events in Videos. In Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images, Salvador, Brazil, 26–29 August 2015.
34. Mousavi, H.; Mohammadi, S.; Perina, A.; Chellali, R.; Murino, V. Analyzing tracklets for the detection of abnormal crowd behavior. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2015; pp. 148–155.
35. Mousavi, H.; Nabi, M.; Galoogahi, H.K.; Perina, A.; Murino, V. Abnormality detection with improved histogram of oriented tracklets. In Proceedings of the 18th International Conference, Genoa, Italy, 7–11 September 2015; pp. 722–732.
36. Edison, A.; V., J.C. Optical Acceleration for Motion Description in Videos. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1642–1650.
37. Kim, J.; Grauman, K. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2921–2928.
38. Tipping, M.E.; Bishop, C.M. Mixtures of probabilistic principal component analyzers. *Neural Comput.* **1999**, *11*, 443–482.
39. Wulff, J.; Black, M.J. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 120–130.
40. Masci, J.; Meier, U.; Cireşan, D.; Schmidhuber, J. Stacked convolutional auto-encoders for hierarchical feature extraction. In Proceedings of the 21st International Conference on Artificial Neural Networks, Espoo, Finland, 14–17 June 2011; pp. 52–59.
41. Hasan, M.; Choi, J.; Neumann, J.; Roy-Chowdhury, A.K.; Davis, L.S. Learning temporal regularity in video sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 27–30 June 2016; pp. 733–742.
42. Chong, Y.S.; Tay, Y.H. Abnormal event detection in videos using spatiotemporal autoencoder. In Proceedings of the 14th International Symposium, ISNN 2017, Sapporo, Hakodate, and Muroran, Hokkaido, Japan, 21–26 June 2017; pp. 189–196.
43. Chalapathy, R.; Menon, A.K.; Chawla, S. Robust, Deep and Inductive Anomaly Detection. In Proceedings of the ECML PKDD 2017: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, Skopje, Macedonia, 18–22 September 2017.

44. Sabokrou, M.; Fathy, M.; Hoseini, M. Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder. *Electron. Lett.* **2016**, *52*, 1122–1124.
45. Ng, A. Sparse autoencoder. *CS294A Lect. Notes* **2011**, *72*, 1–19.
46. Rifai, S.; Vincent, P.; Muller, X.; Glorot, X.; Bengio, Y. Contractive auto-encoders: Explicit invariance during feature extraction. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 833–840.
47. Alain, G.; Bengio, Y. What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.* **2014**, *15*, 3563–3593.
48. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
49. Xu, D.; Yan, Y.; Ricci, E.; Sebe, N. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Comput. Vis. Image Underst.* **2017**, *156*, 117–127.
50. Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.A.; Vincent, P.; Bengio, S. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* **2010**, *11*, 625–660.
51. Vu, H.; Nguyen, T.D.; Travers, A.; Venkatesh, S.; Phung, D. Energy-Based Localized Anomaly Detection in Video Surveillance. In *Advances in Knowledge Discovery and Data Mining, Part I, Proceedings of the 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, 23–26 May 2017*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 641–653.
52. Hawkins, S.; He, H.; Williams, G.; Baxter, R. Outlier detection using replicator neural networks. In *DaWaK*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2454, pp. 170–180.
53. Sabokrou, M.; Fayyaz, M.; Fathy, M.; Klette, R. Deep-Cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes. *IEEE Trans. Image Process.* **2017**, *26*, 1992–2004.
54. Greenewald, K.; Hero, A. Detection of Anomalous Crowd Behavior Using Spatio-Temporal Multiresolution Model and Kronecker Sum Decompositions. *arXiv* **2014**, arXiv:1401.3291.
55. Wiesel, A.; Bibi, O.; Globerson, A. Time Varying Autoregressive Moving Average Models for Covariance Estimation. *IEEE Trans. Signal Process.* **2013**, *61*, 2791–2801.
56. Lotter, W.; Kreiman, G.; Cox, D. Unsupervised learning of visual structure using predictive generative networks. *arXiv* **2015**, arXiv:1511.06380.
57. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; pp. 802–810.
58. Medel, J.R. *Anomaly Detection Using Predictive Convolutional Long Short-Term Memory Units*; Rochester Institute of Technology: Rochester, NY, USA, 2016.
59. Medel, J.R.; Savakis, A. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv* **2016**, arXiv:1612.00390.
60. Luo, W.; Liu, W.; Gao, S. Remembering history with convolutional LSTM for anomaly detection. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 439–444.
61. Zhao, Y.; Deng, B.; Shen, C.; Liu, Y.; Lu, H.; Hua, X.S. Spatio-Temporal AutoEncoder for Video Anomaly Detection. In Proceedings of the 2017 ACM on Multimedia Conference, Mountain View, CA, USA, 23–27 October 2017; pp. 1933–1941.
62. Wiskott, L.; Sejnowski, T.J. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.* **2002**, *14*, 715–770.
63. Creutzig, F.; Sprekeler, H. Predictive coding and the slowness principle: An information-theoretic approach. *Neural Comput.* **2008**, *20*, 1026–1041.
64. Sun, L.; Jia, K.; Chan, T.H.; Fang, Y.; Wang, G.; Yan, S. DL-SFA: Deeply-learned slow feature analysis for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2625–2632.
65. Zhang, Z.; Tao, D. Slow feature analysis for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 436–450.
66. Kompella, V.R.; Luciw, M.; Schmidhuber, J. Incremental slow feature analysis: Adaptive low-complexity slow feature updating from high-dimensional input streams. *Neural Comput.* **2012**, *24*, 2994–3024.

67. Hu, X.; Hu, S.; Huang, Y.; Zhang, H.; Wu, H. Video anomaly detection using deep incremental slow feature analysis network. *IET Comput. Vis.* **2016**, *10*, 258–265.
68. Munawar, A.; Vinayavekhin, P.; De Magistris, G. Spatio-Temporal Anomaly Detection for Industrial Robots through Prediction in Unsupervised Feature Space. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 1017–1025.
69. D’Avino, D.; Cozzolino, D.; Poggi, G.; Verdoliva, L. Autoencoder with recurrent neural networks for video forgery detection. *Electronic Imaging* **2017**, *7*, 92–99.
70. An, J.; Cho, S. *Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability*; Technical Report; SNU Data Mining Center: Seoul, Korea, 2015.
71. Goodfellow, I.J. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv* **2017**, arXiv:1701.00160.
72. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In Proceedings of the International Conference on Information Processing in Medical Imaging, Boone, NC, USA, 25–30 June 2017; pp. 146–157.
73. Ravanbakhsh, M.; Sangineto, E.; Nabi, M.; Sebe, N. Abnormal Event Detection in Videos using Generative Adversarial Nets. In Proceedings of the IEEE International Conference on Image Processing (ICIP) 2017, Beijing, China, 17–20 September 2017.
74. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017*.
75. Zhang, R.; Isola, P.; Efros, A.A. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; Volume 1, No. 2.
76. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I. Adversarial Autoencoders. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
77. Dimokranitou, A. Adversarial Autoencoders for Anomalous Event Detection in Images. Master’s Thesis, Purdue University, Indianapolis, IN, USA, 2017.
78. Munawar, A.; Vinayavekhin, P.; De Magistris, G. Limiting the Reconstruction Capability of Generative Neural Network using Negative Learning. In Proceedings of the 27th IEEE International Workshop on Machine Learning for Signal Processing, MLSP, Tokyo, Japan, 25–28 September 2017.
79. Wataken777. Youtube. Tokyo Express Way. Available online: <https://www.youtube.com/watch?v=UQgj3zkh8zk> (accessed on 1 October 2017).
80. Razakarivony, S.; Jurie, F. Discriminative autoencoders for small targets detection. In Proceedings of the 2014 22nd International Conference on Pattern Recognition (ICPR), Stockholm, Sweden, 24–28 August 2014; pp. 3528–3533.
81. Chollet, F. Keras. 2015. Available online: <https://github.com/keras-team/keras> (accessed on 1 August 2017).
82. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874.
83. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
84. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

