A Class of Augmented Convolutional Networks Architectures for Efficient Visual Anomaly Detection

by

Ambareesh Ravi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2021

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see *Statement of Contributions* included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

In this dissertation work, the following related papers were published or submitted:

1. Ambareesh Ravi, Fakhri Karray, "AutoEncoder regularization using Support Vector Data Description for Anomaly Detection". *[Under Review in] IEEE International Conference on Systems, Man, and Cybernetics*, October 2021.

2. Ambareesh Ravi, Fakhri Karray, "Exploring Convolutional Recurrent Architectures for anomaly detection in videos". *[Under review in] Discover Artificial Intelligence, Springer*, June 2021.

3. Ambareesh Ravi, Qiang Zhao, Fakhri Karray, "Kernel Strengthening and Structural Similarity for improving unsupervised image Anomaly Detection". *[to be submitted]*

4. Ambareesh Ravi, Fakhri Karray, "Attentive AutoEncoders for improving Visual Anomaly Detection". *[Accepted in] IEEE International Conference on Autonomous Systems*, August 2021.

5. Ambareesh Ravi, Xiaozhuo Yu, Iara Santelices, Fakhri Karray, Baris Fidan, "General Frameworks for Anomaly Detection Explainability: A comparative study". *[Accepted in] IEEE International Conference on Autonomous Systems*, August 2021.

In papers 1,2,4, I was responsible for data processing, literature review, model architecture, model training, running all experiments and analysis.

In paper 3, I was responsible for data processing, literature review, model architecture, model training, analysis and parts of the experiments and co-author Qiang Zhao was partly responsible for the experiments and analysis pertaining to kernel strengthening.

In paper 5, I was responsible for data processing, model architecture, model training, running three out of four experiments and analysis. Co-authors Xiaozhuo Yu and Iara Santelices were partly responsible for literature review and the experiments. Dr. Baris Fidan was partly responsible for reviewing and guidance.

In all the above papers, Dr. Fakhri Karray was responsible for coordination, guidance, suggestions and review.

Few parts of sections 3 and 4 of this dissertation come from papers 1 to 5.

Abstract

Visual anomaly detection, the task of isolating visual data that do not conform to the defined notion of normality, is very crucial for the autonomous functioning of entities with exceptional potential in a spectrum of real-world applications. Prevalent methods of visual anomaly detection involve massive, complex, inefficient models whose performances are often restricted by the availability of data, the extent of hyper-parameter tuning and optimal model design. Moreover, popular deep learning approaches such as reconstruction-based methods that use a variant of AutoEncoders and generative methods like Generative Adversarial Network are not inherently designed for the task of anomaly detection. The above factors discussed raise the following severe problems:

1. The general model design may not be efficient without a dedicated anomaly detection objective hence lacking the ability to well distinguish anomalies from the normal data
2. The immense time and effort spent in the search of hyper-parameters and optimal model design restricts models to be immediately deployed for applications
3. The functioning of models involve a lot of human intervention and is data-centric preventing them to be used in automated, online detection tasks
4. The high performing, complex models are too huge to be used in edge applications with low computational capacity that require models with low memory footprint

To overcome these issues, several modular, model-agnostic, efficient and novel improvements to conventional architectures have been proposed and suggested in this work and they can potentially be employed in any AutoEncoder based anomaly detection task. The focus of this work is to develop models that are simple, efficient, require low memory usage and reduced effort expended on hyperparameter tuning and the proposed improvements can aid in readily augmenting the performance over baseline models by a significant margin by producing robust, discriminative and discernible representations to help better segregate anomalies from normal samples.

The overall generic framework proposed throughout this research consists of multiple, efficient architectures that can be used for immediate deployment of models for practical, real-world automated anomaly detection tasks with minimal human intervention and to impart capabilities like online learning and self-regularization for best performance on image and video tasks. The superiority and efficacy of the proposed solutions are enunciated through quantitative and qualitative performance evaluation on a variety of image and video datasets from diverse domains along with rich visualization and ablation studies. This work also focuses on the exploration of interpretability in AutoEncoder-based anomaly detection models with modifications to adapt popular classifier-centric explainability frameworks, to pave way for a better understanding of the function and decision of the models.

Acknowledgements

I thank my supervisor, Dr. Fakhri Karray for the constant support, encouragement and guidance he provided during my masters degree. I also thank him for several opportunities he provided me, to learn and grow.

I also thank the members of the thesis committee - my supervisor Dr. Fakhri Karray, ECE Professor, Dr. Ramadan El Shatshat and ECE Adjunct Professor, Dr. Shady Shehata for their in-depth review of my thesis and for providing prompt and helpful suggestions.

I thank my parents Mr. Ravi Viswanathan and Mrs. Anantha Ravi for their constant love, prayers, support and sacrifices that allowed me to pursue my Master's degree. My deepest appreciation also goes to my grandparents and my friends for their support.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AA CAE Attention Augmented Convolutional AutoEncoder 19, 51, 54

ACB Asymmetric Convolutional Blocks 14, 43, 45

AD Anomaly Detection 8, 23

ADB Asymmetric Deconvolutional Blocks 14, 43, 45

AE AutoEncoder 7, 10–13, 19, 20, 22, 23, 25, 33, 55, 57

BCAE1 Baseline CAE 1 43, 51, 54, 55, 57–59, 63, 64, 66, 72, 75

BCAE1 + SVDD Baseline 1 with SVDD encoder regularization 55, 57–59

BCAE2 Baseline CAE 2 xi, 43, 44, 47, 48, 50

BN Batch Normalization 43

CAE Convolutional AutoEncoder 1, 3, 8–15, 17, 19, 20, 22–25, 30, 33, 34, 36, 43, 45, 51, 53, 55, 57, 59, 70, 75, 77

CNN Convolutional Neural Networks 10, 14, 18, 20, 34, 43, 50, 75

ConvCells Convolutional Cells 26, 30

CRAE Convolutional Recurrent AutoEncoder 25

GAN Generative Adversarial Networks 7

KS CAE Kernel Strenghtened Convolutional AutoEncoder 14, 43–45, 48, 50

Leaky ReLU Leaky Rectified Linear Unit 43

MSE Mean Squared Error 12, 24, 30, 32, 44, 48, 50

PCA Principal Component Analysis 59

ReLU Rectified Linear Unit 43

SA CAE Soft-Attention Convolutional AutoEncoder 21, 51, 54

SE CAE Squeeze Excitation Convolutional AutoEncoder 19, 51, 54

Seq2Seq Sequence-to-Sequence 32, 62, 66, 67

SHAP SHapley Additive exPlanations 10

SOTA State of the Art 7, 18, 53, 57, 79

SSA CAE Softmax Soft-Attention Convolutional AutoEncoder 22, 51, 53, 54, 57

SSIM Structural Similarity Index viii, 15–17, 48, 50

SVDD Support Vector Data Description 3, 23, 24, 53, 55, 57

TSNE t-Distributed Stochastic Neighbor Embedding 59

VAE Variational AutoEncoder 7

XAI Explainable Artificial Intelligence xii, 3, 10, 33–35, 70, 75–77

Chapter 1

Introduction

This chapter introduces the task of visual anomaly detection in images and videos. The problems and shortcomings of the prevalent methods of anomaly detection based on CAE are discussed in detail. We also address the motivation behind the proposed approaches for augmenting visual anomaly detection performance and we also highlight the objectives for the ideal anomaly detection approach that can best suit a particular application.

1.1 Problem Definition

Anomaly detection is the task of identifying and isolating irregular, abnormal and aberrant samples among normal and mostly homogeneous data which are defined specific to the application at hand. Anomalies are also referred to as *outliers, novelties, irregularities or abnormalities* [3]. Anomaly detection is an important sub-field of machine learning that aids in the automatic identification of rare but significant events in applications across several domains [20, 21]. *Visual anomaly detection* is an area that exclusively deals with anomaly detection in visual data such as images, videos or visual stream of input. For the task of anomaly detection, deep learning methods are favoured over conventional machine learning algorithms. The fostered popularity and adoption of deep learning methods for anomaly detection can be attributed to the ability to learn and uncover patterns in complex high-dimensional data, better generalization and exceptional performance on a variety of learning tasks [19]. Due to the non-invasive nature of analysis materials and objects, simplicity due to unsupervised nature and reliability in usage, visual anomaly detection has been widely applied in many research fields and industrial applications.

Prevailing techniques for visual anomaly detection have several shortcomings and the most important one of all is that they are not specifically designed for the task [44, 69]. Since the nature and variety of novelties or anomalies can't be predetermined for any application, it is hard to define a dataset enveloping every type of possible anomaly. This naturally leads to a difficulty in equipping anomaly detection techniques with the innate capability to identify *any* potential anomalies that are not foreseen and taken into account. The current deep learning methods take a makeshift approach to use models designed for or trained on other tasks for anomaly detection which hatches a gap in efficiency of operation and the overall performance. An ideal anomaly detection algorithm should inherently learn the notion of normality from the most influential and discriminating features in the data, operate efficiently with capabilities of learning in a semi/unsupervised setup without relying on labelled examples (owing to the evident scarcity in labelled data) and it should also be capable of self-correction with minimal human intervention.

1.2 Motivation

Visual anomaly detection is employed in critical real-world applications in areas such as surveillance, medicine, autonomous driving, cyber-security, finance, quality assurance etc. most of which are in dire need of automation. Moreover, monotonous tasks like video surveillance where a person has to spend hours together in front of screens introduce fatigue and mental strain that could possibly lead to errors in judgement that can not be afforded. In cases like medical diagnosis, machine learning approaches have time and again proven their effectiveness in uncovering hidden patterns from data and in deriving actionable insights to a super-human level. These factors have made automation in these areas ineluctable and deep learning is the most favourable solution in complex cases. The practical applications for visual anomaly detection mandate a high degree of accuracy in predictions, reliable functioning and in some cases involve an online stream of temporal unlabelled visual data with scarce anomalous examples with little to no room for error in prediction at a rapid pace. But the prevalent deep learning methods require excessive time and human effort on design and hyper-parameter tuning for optimal operation, with multiple iterations of the design process for meagre improvements thereby restricting their use in areas of automation involving online learning. Such high demands and practical implications put anomaly detection research under keen scrutiny, warranting the need for the algorithms to be efficient in terms of computation and memory usage, to be robust, deliver reliable performance and with an end-to-end automated operation for instantaneous deployment.

1.3 Scope

This research work focuses on developing deep learning approaches involving convolutional architectures along with novel modifications and modular improvements that can help in reducing the time and effort expended for model design and tuning at the same time to guarantee realistic and definite improvements in performance without much extra burden on computational and memory requirements for image and frame-level video anomaly detection tasks. The proposed modifications are to help them function efficiently with better learning capabilities to understand the nuances of normal patterns in data with the inherent ability to distinguish rare anomalous samples from the vastly normal set of visual data. This work also focuses on the explainability of Convolutional AutoEncoder models in understanding their decisions and also to improve their functioning with additional support and expertise.

1.4 Objectives

The main objective of this work is to create performance and computation efficient convolutional models for detecting visual anomalies. The secondary focus is to conduct a feasibility study to apply XAI frameworks to interpret their decisions. To achieve the aforementioned objectives, the following contributions are made in this research:

1. Kernel level modification to improve learning in CAEs to performance.
2. Reconfiguration of the conventional loss function to accommodate a measure of visual similarity.
3. Imparting the ability to focus only on normal patterns of data by applying visual soft-attention mechanism to CAEs
4. Enforcing separation between normal and anomalous embeddings using SVDD regularization in CAEs for better isolation of anomalies.
5. Explore conventional and novel architectures for video anomaly detection to learn spatio-temporal features and compare their effectiveness.
6. Explore the feasibility of using popular explainability frameworks to CAEs without major structural modifications to the learnt models.

1.5 Thesis Organization

This thesis is organized into five chapters. The current chapter introduces the problem of anomaly detection, the shortcomings of current approaches, the main objectives and scope for the proposed solutions with the contributions of this work. Chapter 2 covers a detailed review of other literature works that pertain to the topic of anomaly detection. Chapter 3 deals with the proposed approaches in detail. Chapter 4 discusses the datasets for the experiments, metrics for evaluations and experiments on the proposed approaches with detailed results and discussions. The final chapter summarizes and concludes the major results of this research work.

Chapter 2

Related work

2.1 Introduction

This chapter discusses the progression of research in the field of visual anomaly detection in detail along with seminal and popular works that had breakthroughs in the field. This chapter also deals with possible open problems with the current approaches along with some of the attempts at solving them.

2.2 Conventional Methods

Early techniques for anomaly detection involved the application of statistical and conventional machine learning methods based on the nature and structure of the input data [21]. Many methods focus on data transformation and feature engineering to reduce data into compact representations and use them to compute the anomalous nature of new data points based on the physical proximity in multi-dimensional space. Methods like K-Nearest Neighbors, self-organizing maps [83], density-based clustering [47] etc. have been widely employed for anomaly detection tasks. Since most of such methods often use Euclidean distance to compute the proximity, the inherent manifold containing the representations are rarely accounted for, leading to poor performance with an increase in data complexity and dimension. Other decision tree based approaches binary tree-based classification and detection using isolation forest which is based on the number of splits [82] do not scale well for high dimensional data such as images.

2.3 Representational Learning Methods

As the field evolved with the adoption of representational learning methods. Initially, Principal Component Analysis (PCA) [38] was used to capture the correlation among data features and detect anomalies globally with respect to a reference set of data. Later, Support Vector Machine [76] was also a popular alternative to isolate outliers. In parallel, pure statistical methods like multi-variate analysis using Gaussian models [89] of data, z-score computation on transformed data to find outliers based on the standard deviation from the data distribution [82] were also employed to detect anomalies, along with other non-parametric techniques like histogram profiling [21], kernel-based methods for images. But these methods were later proven to be ineffective due to performance and scalability issues with large volume and complexity of the real-world images. One Class Support Vector Machine [73] was also a popular method to detect outlier that learns and controls a margin of separation enclosing the normal data. Similar to this is Support Vector Data Description [81], which forms a hyper-sphere of separation enclosing the normal data and detect outliers by separating them from the hyper-sphere. Both the methods require large memory and computational requirements, making them infeasible for very high dimensional data like images. Also, the above-mentioned methods do not possess the capability to analyse and learn the context in visual data like images that have a high correlation between the pixels in a local neighbourhood. As a result of which, many traditional image processing techniques with several modules like background estimation, optical flow estimation, object tracking for images and videos required data-specific design with the need for multiple modules running in tandem to produce desired results [74]. [14] and [87] extensively discuss the recent advances in representational learning in detail.

2.4 Deep learning Methods

The research community steadily moved on towards modern, learning-based, data-centric techniques like Restricted Boltzmann Machine (RBM) that consists of bidirectional connections to learn the latent representations and to model prior distribution from multi-dimensional data [79] and also to sample new data points as an aid to detect outliers [29]. An improvement over RBMs called Deep Belief Networks [35] with multiple RBM layers were employed for outlier detection showing overall better performance. This progression of works slowly paved the way for the application of deep learning techniques for visual anomaly detection and their effectiveness in discovering, learning and abstracting vital information from the complex and intricate structure of visual data and discerning different visual features were the pivoting reason. A fairly simple approach of using representations from a pre-trained model originally trained tasks such as classification, object localization, object detection or segmentation to compare against other samples proved to be effective over the traditional approaches for image anomaly detection [19]. CNN-based methods [41], [51] to detect anomalies from images using the learnt representations though happened to perform well in general, did not fare well when the number and diversity of anomalies were significantly high in data and they also required a large number of samples for training or tuning.

Reconstruction-based anomaly detection using AEs is the most popular method for images and will be discussed in detail in the following sections. Recently, a few other deep learning approaches with a dedicated anomaly detection objective that use deep classifier models such as a variant of SVDD in [69] and a one-class neural network that uses a modified OC-SVM based objective function as in [20] were specifically developed for visual anomaly detection. On the other hand several works on improving the learning efficiency of CNNs were also active like asymmetric convolutions [27], residual connections [34], inception modules [80], Squeeze Excitation networks [37] for SOTA results on object detection, Attention Augmented Convolutions [13] with a relative self-attention mechanism as a replacement for conventional convolutions etc. were also critical to the development of the field and improvements in model design [42].

2.5 Reconstruction and Prediction-based Deep Models

Modern anomaly detection techniques for visual data can be broadly grouped into three categories - reconstruction models, generative models and predictive models. *Reconstruction-based methods* learn to reconstruct normal data from encoded representations and their inability to reconstruct novel, anomalous samples are utilized to detect anomalies based on the reconstruction error. Generative modelling involves the use of generative models like VAE [6] or GAN [4], learn to generate new data samples with a generator-discriminator setup that can detect samples out of distribution and the discriminator is used for quantifying anomalies. Predictive models learn to encode data into compact representations from which they predict new data based on the previously learnt information from the past. This research predominantly focuses on reconstruction-based and prediction-based methods.

AutoEncoders [71], [46], [10], an important class of unsupervised neural networks, have been prevalent in anomaly detection tasks as they can operate well in the absence of labeled data. Reconstruction-based methods employ a variant of AE [6, 15, 22, 66, 68, 91, 95] architecture to learn the notion of normality from datasets containing only normal samples and their inability to reconstruct abnormal data is utilized. [19, 44, 66] discuss the popular methods for image and video anomaly detection extensively. The progression of development in reconstruction-based anomaly detection can be observed from the available literature. [72] used AutoEncoders for anomaly detection and compared the performance against traditional dimensionality-reduction methods like PCA and Kernel PCA. In [95], the authors put forward an extension of deep AutoEncoder based on Robust PCA to improve anomaly detection performance. [64] and [30] both propose variants of discriminative AutoEncoder setups that use supervised learning to ensure robust representations for various image tasks. [11] proposes a convolutional AutoEncoder model for segmenting the abnormalities in brain MR scan images. [5] uses an unsupervised variational AutoEncoder and models transfer learnt across multiple data sets to detect brain lesions from images. [33] uses patch-wise unsupervised generative image completion models to detect anomaly in surfaces for material inspection based on pixel-wise reconstruction error. The operation of these methods is dependent on the representational learning capacity with no separate learnt knowledge for distinguishing normal and abnormal samples. Hence, the representational capacity, compactness and reconstructional ability ultimately determine their performance on the task of AD. [26] exhibits explicit improvement in AutoEncoders by adopting Mahalanobis distance as a loss function to identify out-of-distribution samples from the latent encoding space. After which, [18] explained the shortcomings of pixel-based loss metrics such as L1 and L2 losses and how incorporating structural similarity can improve performance on datasets containing fibrous materials and fabrics. There are several other problems with reconstruction-based anomaly detection techniques like (1) inefficient learning as AutoEncoders are originally meant dimensionality detection and not anomaly detection, (2) this inefficiency also paves way for the problem of sub-par performance concerning the computational complexity of the models, (3) there is also a need to intuitively separate and concentrate normal samples together in the latent space to easily segregate anomalies and finally, (4) the occasional reconstruction of anomalies that hinder the performance of CAEs. To circumvent (4), [59] presents a method to explicitly limit the reconstruction capacity of CAE by introducing negative samples for discriminative learning. But there is a need for a method of better discriminative learning without the need for manual data segregation to enable complete automation in applications. This research tries to address all these open problems with the convolutional network and related techniques that are proposed.

2.6 Deep Learning Methods for Anomaly Detection in Videos

Videos are dynamic data that contain patterns of motion of objects in subsequent, coherent, temporally arranged frames in a time series. Temporal information is critical in understanding the context behind motion patterns in videos. 2D CAE models [44, 50, 60] although perform considerably well for videos, will not be able to identify the temporal behavioural patterns and change in motion since they operate frame-wise. [96] provides a comprehensive discussion on deep learning methods for anomaly detection in surveillance videos and discusses the open problems and presents a detailed analysis of supervised and unsupervised methods. Hybrid models and spatio-temporal AutoEncoders that operate on a set of frames together have been proven to perform well for video anomaly detection [32]. In addition to that, 3D CAE were also popular in the early stages but quickly went out of favour since they accompany a huge number of parameters and they were proven to be ineffective in learning representation of videos compared to other architectures [40, 78, 85]. Later, [28] first proposed the idea of using visual features from models trained on ImageNet [25] using transfer learning in LSTM networks to effect learning spatio-temporal features for video-related tasks like action recognition. Then, ConvLSTM which replaces fully convolutional layers with convolutional layers to operate on images was first introduced in [75] for predicting rainfall intensity patterns from the past images over a local region which was originally inspired from [65]. Many works that use a variant of ConvLSTM AutoEncoders [24, 56, 58, 77] came into existence showing significant improvements in performance on video-related tasks owing to their ability to learn and memorize past events that help in the reconstruction of the present or in predicting the future frames. It is established from all the previously mentioned works that convolutional recurrent networks are effective for learning video representations along with the fact that only ConvLSTMs are predominantly used. This marks a lack of studies on the comparison of various convolutional recurrent architectures for learning representations in videos as well as for anomaly detection. Hence, an important component of this research is to analyse different variants of architectures in order to find the most effective solution in terms of the trade-off between performance and computational requirement since there are many use cases like surveillance, security, autonomous driving that involve temporal, dynamic data that require highly accurate and efficient models that can distinguish normal and anomalous inputs.

2.7 Explainability in AutoEncoders

Explaining deep, black-box models have become critical in understanding and interpreting their decisions and it is important to explain decisions when the application have direct impact on people. Although there are several works [92], [84] pertaining to the XAI of CNN for various visual tasks, the number of works on XAI of CAE for the task of anomaly detection is very sparse. The completeness of any machine learning algorithm is to explain its decisions, especially when deciding on a part of an image as anomalous. [31] uses a model collaborative filtering in recommender systems to predict and extrapolate missing ratings and uses an additional explanation matrix for explaining the recommendations from the model.

[16] is the only work of XAI on CAE that proposes explanations based on the computation of feature-wise distances between image samples and employs a greedy approach to select contributing features with the highest standard deviation in reconstruction error. [7] uses kernel SHAP for explanations in AE by focusing on contributions of each neurons in the network. Many XAI works have been model specific and non-transferable across other models. This warrants the need for a generalized, model-agnostic setup to explain CAE since the conventional way of using the residual reconstruction maps is ineffective as it is impossible to spatially locate anomalies from the missing or deformed reconstruction of an object in the image.

2.8 Summary

This chapter presented an overall view and the current state of the anomaly detection research along with some popular works. The open problems with the current approaches were identified and appropriate solutions to those problems are presented in the next chapter with some methods to improve the overall efficiency and performance of the current deep learning architectures.

Chapter 3

Proposed Solution

3.1 Introduction

The prevalent methods that use vanilla CAEs suffer from the serious disadvantage of inefficient and impaired learning methodology due to the lack of a dedicated anomaly detection based learning objective. To overcome this disadvantage, several improvements to CAE are suggested in this section that is universally applicable to any reconstruction-based image or video anomaly detection tasks. The shortcomings of existing methods are discussed in detail along with the efficacy and advantages of the proposed solutions in this chapter. Along with said improvements, different convolutional recurrent architectures for prediction-based video anomaly detection are explored and compared and finally, the chapter ends with the adoption of different explainability frameworks for visual anomaly detection task to explain the nature and location of anomalies.

3.2 Review of AutoEncoders

An AE [15, 66] is a class of neural networks that can efficiently represent the input data. AutoEncoders were initially introduced for dimensionality reduction as they compress the input data into a latent representation and reconstruct the data back to the input dimension from the latent representation. An AutoEncoder can be formulated in terms of an encoder and a decoder. The encoder f_e is a neural network that compresses the input data $x \epsilon \mathbb{R}^{1\times d_i}$ of dimension d_i into a learnt multi-dimensional latent space $z \epsilon \mathbb{R}^{1\times d_l}$ of dimension d_l where z is the latent encoding given by equation (3.1) where $W_e and b_e$ are the weights and bias of the encoder, jointly represented by trainable encoder parameters θ_e and ϕ is any non-linear activation function. The decoder f_d is another neural network with the ability to reconstruct the data $x' \epsilon \mathbb{R}^{1\times d_i}$ back to the input dimension d_i from the latent representation d_l as given by equation (3.2) where W_d and b_d are the weights and bias of decoder jointly represented by θ_d. The AE is characterized by the parameters $\theta = [\theta_e, \theta_d]$. AutoEncoders can be reduced to PCA when the weights of the encoder and decoder are transpose of each other i.e. $\boldsymbol{W_e} = \boldsymbol{W_d}^T$ and the layers have identity activation function where $\phi(x) = x$ [46].

$$z = f_e(x) = \phi(\boldsymbol{W_e}x + \boldsymbol{b_e}) \tag{3.1}$$

$$x' = f_d(f_e(x)) = f_d(z) = \phi(\boldsymbol{W_d}z + \boldsymbol{b_d}) \tag{3.2}$$

3.2.1 AutoEncoders for Visual Anomaly Detection

Typically for anomaly detection, the dimension of the latent representation [1] is less than the dimension of the original data and such type of AutoEncoders are called *under-complete*. Since AutoEncoders are devoid of the need for labels for training, they belong to the class of *unsupervised* learning algorithms. Usually, MSE is used as the objective function to be minimized for training the AEs using any gradient-based algorithm for the reconstructions to be as close as the inputs for efficient representation of the input data. MSE is calculated between the input and the reconstruction given by the equation (3.3).

$$\boldsymbol{L}_{MSE}(\theta) = \frac{1}{n}\sum_{i=1}^{n} ||x_i - f_d(f_e(x_i;\theta_e);\theta_d)||^2 \tag{3.3}$$

The simplest possible form of an AE can be constructed using feed-forward neural networks as encoder and decoder. For images, to facilitate spatial learning, a special variant of AE using convolutional and transpose convolutional layers is widely employed which is generally referred to as CAE. In a CAE, convolutional layers are employed in the encoder for the abstraction of spatial information into scalar values in the receptive fields from the input layers and similarly transpose convolutional layers are used in the decoder to upsample/populate spatial information by increasing the dimensions from values as opposed to the convolutional layers. The multiple convolutional layers help in capturing high-level features from the local neighbourhood of pixels in the images in addition to

[1] latent representation is also known as the embedded representations or just embeddings

the low-level features and the AE learns the inherent distribution of the normal data in the high dimensional manifold through the encoding process. There are other possible variants of AE that are prominent in visual anomaly detection research differing in terms of configuration, construction and objective functions.

3.2.2 Current State of Reconstruction-based Anomaly Detection

AutoEncoders have been popularly employed in anomaly detection tasks [44] owing to their inherent ability to effectively learn the representation of normality from the normal input samples using the encoder through patterns that are essential for optimal reconstruction. The error or loss between the inputs and the reconstructions are in turn utilized as a metric to measure or score anomalies. This is under the critical assumption that the AE trained only on normal data will struggle to reconstruct anomalous inputs and so a high reconstruction error implies the presence of anomaly and vice-versa. The vital role of learning in the encoder of the AE for the task of anomaly detection is apparent. Albeit, it should be noted that the objective function for training the AE is not designed for the task of anomaly detection but only for optimal reconstruction and embedded representation. Hence, the natural disposition of AE may not be optimal for the task of anomaly detection although its performance has been proven to be compelling in most cases. It is imperative that imparting desirable properties like pronounced learning methodology and anomaly detection centred learning objective can help in the aggregation of normal embeddings together and far apart from the anomalous embeddings in the multi-dimensional latent space which can control the accidental reconstruction of anomalies which is a common issue with CAE. These modifications can help in augmenting the anomaly detection performance of CAE and make them more effective for real-world applications. The modifications that are proposed in this work come at low computational and memory requirement without any extra burden and are model-agnostic with provision to be used in any CAE for the task of anomaly detection.

3.3 Kernel Strengthened Convolutional AutoEncoders

The anomaly detection performance of CAEs can be directly correlated with the architectural complexity and the available volume of normal data for training. A significant portion of time and effort is expended in the quest for optimal architectural design and hyper-parameters and there is little guarantee for the reliability of model performance in relation to its complexity. In addition, it is hard to obtain a large dataset with perfectly normal samples and requires a lot of human effort for inspection to filter out accidental anomalies from the training dataset. To alleviate such practical limitations and ensure efficient operation of CAE for anomaly detection, a generic approach to improve kernel-level learning in CAE, there by ensuring robust representations is proposed in this section and is inspired from [27]. The proposed KS CAE consists of convolutional and transpose convolutional layers replaced with ACB and ADB layers respectively that show significant performance improvements over the baseline model. The exact configuration of the KS CAE is discussed in the next chapter in detail.

3.3.1 Asymmetric Convolution Blocks

An Asymmetric Convolution Block (ACB) [27] consists of horizontal and vertical kernels in addition to the conventionally used square kernels whose outputs are summed up and passed to the next layer. The output from the previous layer out_{l-1} will be the input to the a layer l and the operation of ACB can be expressed as given in the equation (3.4) where $H_{l(a,b)}(x)$ is the convolution operation with n kernels of size $a \times b$ at layer l and ϕ is the activation function at the layer (commonly ReLU) after a batch normalization layer. After a model is trained, the output and size compatible asymmetric kernels can be *fused*[2] together by channel-wise addition at each layer as proposed by [27] making this structural change capable of improving performance without the additional computational expense and it is pluggable into any CNN based architecture. The kernel-fusing to reduce computations requires careful design of the model architecture. The horizontal and vertical kernels learn the linear spatial information from the input images and since they operate in a linear fashion, it makes them robust to spatial neighbourhood distortions like rotational and positional shifts of pixels in the input images.

$$out_l = \phi\{H_{l(k,k)}(out_{l-1}) + H_{l(k,1)}(out_{l-1}) + H_{l(1,k)}(out_{l-1})\} \tag{3.4}$$

3.3.2 Asymmetric Deconvolutional Blocks

One of the important contributions in this work is the Asymmetric Deconvolutional Blocks (ADB) that was not proposed in [27]. ADBs are similar to ACBs where the convolutional layers are replaced by transpose convolutional layers that upsample from low dimensional

[2] To match the dimensions from the square kernel-layers, layers with vertical and horizontal kernels have strides, output padding and dilation appropriately adjusted to enable kernel-fusing.

representations. Transposed convolutions are used instead of deconvolutions where nearest neighbours interpolation or bi-cubic interpolation is employed to generate rich activation maps.

3.4 Structural Similarity Convolutional AutoEncoder

In anomaly detection, the prevalent deep learning based models for images like the CAEs operate on loss functions like L1 or L2 that have an unrealistic assumption that the pixels are independent of each other and don't take the local neighbourhood structure in images into account [93]. The proposed method in this section can help to alleviate this problem by using a modified, task-specific compound loss function that incorporates **visual similarity** metrics to better segregate anomalous samples. This modification has no effect on inference and hence provides an improvement in performance without any addition in computation similar to the previous approach.

3.4.1 Structural Similarity Index

SSIM [90] is a perceptual metric that quantifies the similarity between two image patches based on their properties such as luminescence, contrast and structure. It measures the change or degradation between images by considering the visual features in terms of structure by taking into account the interdependence between pixels in a local neighbourhood. SSIM is calculated as a weighted product of three components between two image patches p_1 and p_2 - structure (s), luminescence (l) and contrast (c).

$$s(p_1, p_2) = \frac{\sigma_{p_1 p_2} + c_s}{\sigma_{p_1}\sigma_{p_2} + c_s} \tag{3.5}$$

$$l(p_1, p_2) = \frac{2\mu_{p_1}\mu_{p_2} + c_l}{\mu_{p_1}^2 + \mu_{p_2}^2 + c_l} \tag{3.6}$$

$$c(p_1, p_2) = \frac{2\sigma_{p_1}\sigma_{p_2} + c_c}{\sigma_{p_1}^2 + \sigma_{p_2}^2 + c_c} \tag{3.7}$$

where c_s, c_l and c_c are factors for numerical stability during computation and $c_s = c_c/2$ with $c_l = (k_1 L)^2$ and $c_c = (k_2 L)^2$ where k1 = 0.01 and k2 = 0.03 as in the paper. So SSIM in terms of the above equations (3.5), (3.6) and (3.7) can be represented as follows:

$$SSIM(p_1, p_2) = [l(p_1, p_2)^{\alpha}.c(p_1, p_2)^{\beta}.s(p_1, p_2)^{\gamma}] \tag{3.8}$$

and to simplify the above equations, the weights α, β and γ can be set to 1. Doing so, the equation is reduced to

$$SSIM(p_1, p_2) = \frac{(2\mu_{p1}\mu_{p2} + c_l)(2\sigma_{p_1p_2} + c_c)}{(\mu^2_{p_1} + \mu^2_{p_2} + c_l s)(\sigma^2_{p_1} + \sigma^2_{p_2} + c_c)} \tag{3.9}$$

The value of SSIM ϵ [-1,1] where 1 denotes that the image patches are identical and -1 denotes that they are completely different [90]. The overall similarity score between two images can be calculated as the mean score of the SSIM value between the patches of two images by sliding a kernel with size $K \times K$ over the images uniformly (by default, we use 11x11 as suggested in the paper) [90]. The window size can be adjusted for more accurate results based on the data at hand.

3.4.2 SSIM based Compound Loss Function

In comparison to commonly used losses to train AutoEncoders such as L1 loss and L2 Loss / squared error which operate on a pixel by pixel basis, SSIM takes into account the inherent structure in the image by considering the local neighbourhood in the images that preserve shape, texture and patterns which is important for the model to learn the context in the images. The false assumption in conventional loss functions about the independence of pixels in the local neighbourhood [93] while in reality, the pixels in images have a strong correlation with groups of other pixels or the local neighbourhoods that form objects or entities in the images. This flaw is noticeable as pixel-based losses produce blurry, deformed or partial reconstructions whereas training with SSIM helps in producing slightly better reconstructions with sharper edges.

To illustrate the appealing properties of SSIM over mean squared error, an example is presented in Figure 3.1 which shows 4 images from HAM10000 data set which are very similar in terms of colour schemes but are different in terms of the texture which is apparent to a human eye. As seen from the figure, images 1 and 2 are similar to each other and the SSIM index (normalized between 0 and 1) is 0.68 and the MSE score 0.019 whereas images 1 and 3 are very different and the SSIM index is 0.61 as expected but MSE is 0.017 which is lower than the value between 1 and 2. Since SSIM provides a clear and observable degree of visual similarity between two images under comparison, it is preferred over MSE for visual tasks. The differentiable nature of SSIM from [18] is shown in equation 3.9 and

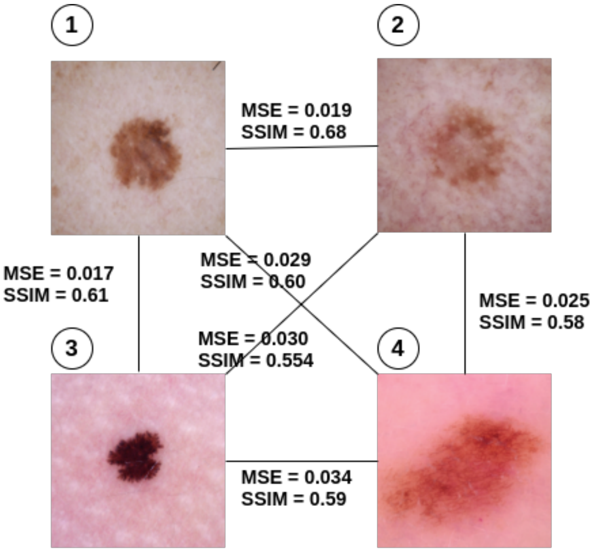


Figure 3.1: An example illustrating the preference of SSIM over MSE

hence SSIM can be used as an objective (loss) function for learning, minimization and to backpropagate the gradients to the input layers in a convolutional neural network such as a CAE. Thus, a *compound loss function* consisting of a weighted sum of mean squared error with SSIM index as shown in equation (3.10) is proposed.

$$\boldsymbol{L}_{comp}(\theta) = \eta \boldsymbol{L}_{MSE}(\theta) + (1 - \eta)(1 - \boldsymbol{SSIM}(x_i, x')) \tag{3.10}$$

In equation (3.10), the term $(1 - SSIM(x_i, x'))$ is an alteration to support minimization for all the values of SSIM between the two images. For simplicity, the value of the weight η is taken as 0.5, giving equal weightage to both the components. It was found through experimentation that the values of $\eta = 1 - \eta = 0.5$ worked consistently across all the models and datasets, although there were slight improvements with other values for few datasets. Since both the terms of the compound loss function in equation (3.10) can be differentiated, this loss function can directly be incorporated into any existing convolutional neural network architecture. The tuning of model parameters due to the SSIM term helps in implicitly minimizing the squared error between pixels as the training progress yielding reconstructions that are sharper, similar to the inputs and thereby helping in better representation of normality.

3.5 Attentive AutoEncoders

Attention was first introduced for machine translation [9] and it inspired from the ability of human beings to focus on important parts of complex scenarios to make insights out of them. Humans tend to efficiently locate regions of interest from the information presented and attention mechanisms were designed to replicate this mannerism. Attention can expand the abilities of neural network models [12] by helping them focus on vital parts of the input. They have shown SOTA performance in many natural language tasks as well as visual tasks. Attention has also become a hot topic in visual understanding and has mostly been employed for object recognition, segmentation, image captioning and visual question-answering leading to better performance of many CNN models on these tasks [52].

3.5.1 Visual Attention Mechanisms

In CNNs, as the input propagates through multiple convolutional layers, only the disparate local neighbourhoods at each layer are covered by the kernels and through subsequent layers, the spatial information learnt is abstracted and reduced to scalar values as a result of which the global information representing the complete context of the image is lost. Though larger kernels tend to capture information in larger neighbourhoods and an increment in the number of layers improves the abstraction process, they can constitute tremendous raise in the computational requirement. Hence, attention can come in handy by guiding CNNs to focus on important sections of the inputs relating to the context, thereby improving the performance readily, without much increase in computational complexity.

Attention in computer vision operates by augmenting the important parts of the image while attenuating the other parts to emphasize the relative importance of the essential input features over the others. Visual attention mechanisms fall under two main categories - soft and hard attention. Hard attention is a method in which one or more parts of the image is cropped and taken for processing. In soft attention, parts of the inputs which are not essential are faded out but not completely discarded. In this section, different soft attention mechanisms and their role in improving the performance of convolutional models are primarily focused on. The most important properties of an effective attention mechanism are differentiability, the ability to learn both local, global contexts and good approximation capabilities that can help in abstraction from lower to higher granularity of context. In this section, three attention models are discussed - two of which were adopted from literature and one is a novel architecture that is proposed.

3.5.2 Adopted Mechanisms to Convolutional AutoEncoders

AA CAE

Attention Augmented Convolution, introduced in [13] is a convolutional self-attention mechanism to learn information from images in a global context since the conventional convolutions work in local neighbourhoods which has shown consistent performance improvements on image classification and object detection tasks [13]. The input images are fed to a convolutional layer and a multi-head self-attention layer simultaneously. The final outputs are produced by concatenating their respective outputs while maintaining translation equivariance to retain the positional information in images [13]. As the mechanism was primarily introduced to learn information from images for discriminative visual tasks, it could also be incorporated into convolutional AutoEncoders. The mechanism contains two important parameters to be tuned that determine the overall performance, d_k, d_v which are key depth and attention channels respectively. The convolutional layers of the CAE are replaced with attention augmented convolutional layers and come at a parameter increase of *15%* with lowest settings of d_k, d_v.

SE CAE

Squeeze Excitation Network [37] introduces channel-wise attention in convolutional architectures by learning channel-wise inter-dependencies through adaptive weighing of feature map in each channel. The adaptive weighting mechanism helps the network to analyse the importance of each feature map at any layer. The 'squeeze' part of the network outputs a vector of length equivalent to the number of channels by compressing each feature map into a scalar value and the 'excitation' part of the network is made of two fully connected layers that process the squeezed vector into a *Sigmoid* activated vector of the same size denoting the importance of each channel after which the channels are scaled accordingly. The squeeze-excitation blocks are added after each convolution and transpose convolution operations and the modification to the CAE model comes at a computation cost less than *2.5%*.

3.5.3 Proposed Novel Attention Mechanisms for Convolutional AutoEncoders

For reconstruction-based anomaly detection tasks, the CAEs often suffer from performance deterioration due to the accidental reconstruction of anomalies. An ideal AE mechanism is expected to learn the important features of the normal input required for reconstruction at a global scale so that it doesn't reconstruct a local patch of anomaly accidentally. The second important property of AE for anomaly detection is to learn discriminative embeddings in the latent space to differentiate well between normal and anomalous samples so that the embeddings precisely affect the reconstruction of the input sample. As attention inherently is a mechanism that can help the network focus on important features of the input $x \epsilon \mathbb{R}^{WxH}$ (W, H are the width and height of the input image), it can help in reducing the *almost-perfect* reconstruction capacity of the AE models. CNNs have appealing properties of visual learning and universal approximation capabilities and in this section, few simplistic alterations to the CAE using soft-attention mechanisms are made to augment the overall performance of CAEs. In summary, two variants of the soft-attention mechanisms - one trainable through a modified loss function and another self-contained softmax attention mechanism are proposed.

Convolutional AutoEncoder with Learnable Input Soft-attention

The proposed convolutional soft attention block consists of two convolution layers with different kernel sizes, a smaller kernel k_1 to learn the local, elementary features and project the learnt representations over multiple channels in the first layer f_{conv1} which is followed by a larger kernel k_2 to learn a larger neighbourhood and compress back to the original number of channels in the second layer f_{conv2}. The convolutional layers are followed by batch normalization and *Sigmoid* activation (σ) to produce the output *attention map* x_{AM} as probability scores of importance for each pixel. These probability scores act as an adaptive weighting mechanism denoting the importance of any spatial region of an input image. The attention map is weighted with the original input and sent to the actual AutoEncoder architecture as input, the description of which is shown in Figure 3.2.

While training, the objective is to retain the importance of essential pixels at high magnitudes and reduce the rest. This can be achieved by either reducing the norm of the weights of the attention block or the attention map itself so that the convolutional output under the constraint will automatically learn the region of pixels to concentrate on. Since both methods yielded almost the same results, reducing the norm of the final attention map with a penalty factor λ was chosen as the cost-effective approach. The equations of the convolutional input soft-attention block are given in equation (3.11). The block has two adjustable parameters namely the number of channels for intermediate projection C_p and the penalty factor λ. The penalty factor should be chosen in such a way that it doesn't lead to loss of vital information from the input image. The proposed loss-based soft-attention block is added before the input layer of CAE while the rest of the model is retained as such. This model is referred to as SA CAE

$$
\begin{aligned}
x_{BN} &= \text{BatchNorm}(f_{conv2}(f_{conv1}(x))) \\
x_{AM} &= \sigma(x_{BN}) \\
\hat{x} &= x \odot x_{AM} \\
L_{ConvAttention} &= L_{MSE} + \lambda||x_{AM}||
\end{aligned} \tag{3.11}
$$

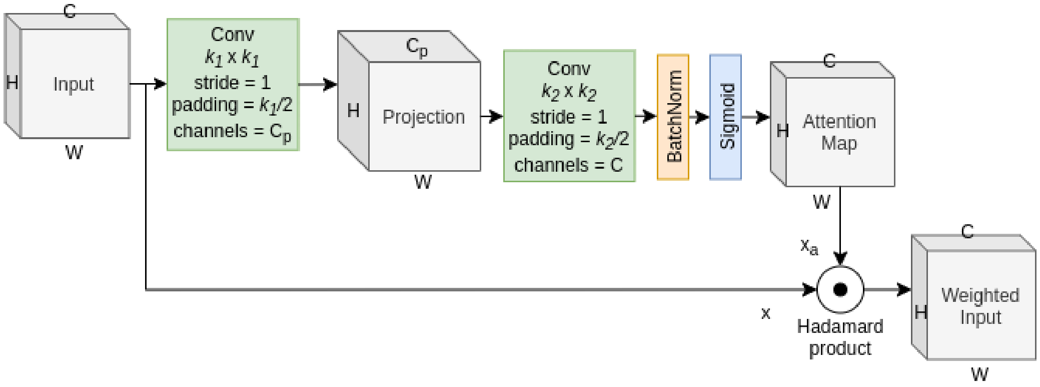


Figure 3.2: Proposed convolution input soft-attention block

Convolutional AutoEncoder with Learnable Softmax Input Soft-attention

The previously proposed approach consists of two parameters λ and C_p that have to be carefully and simultaneously tuned for maximum performance. Previous approach was presented to show the effective nature of our primary approach to input soft-attention which is the *Softmax Attention mechanism* and how the changes affect the performance. To reduce the effort in terms of hyper-parameters search and tuning, the structure of the attention block was modified further such that the output attention map can determine the importance of pixels without any parameters that require tuning.

The components till the batch normalization layer were retained and further improvements were annexed. The outputs at the batch normalization layer are summed along the width and height of the tensors producing two vectors of size $1 \times H \times C$ and $W \times 1 \times C$ respectively for each sample in the batch. These vectors represent the overall context value of the pixels width-wise and height-wise and applying *softmax* on them results in a probability distribution of importance along their respective axes. The two tensors are then multiplied to get $W \times H \times C$ again and this *softmax attention map*, x_{SAM} results in bands of probabilities for neighbourhoods with important features in the input image. This results in all values retained as probability scores and is weighed with the input x to get the final input $\hat{x}$ to the AE. This alleviates the need to modify the loss function with a penalty term. Scaling or normalizing x_{SAM} can help in better visualization of the outputs at intermediate layers and considerable improvements were observed by scaling the values between 0.3 and 1 in intermediate layers' activations. This attention mechanism is incorporated similar to the previous approach by prefixing it to CAE. This model is referred to as SSA CAE and the mechanism can be mathematically expressed as in equation (3.12)

$$\begin{aligned}
&S_w, S_h = \text{AxisWiseSummation}(x_{BN}) \\
&x_{SAM} = \text{softmax}(S_w) \cdot \text{softmax}(S_h) \\
&\hat{x} = x \odot x_a, \;\; \text{where } x_a \epsilon \{x_{CAM}, x_{SAM}\}
\end{aligned} \tag{3.12}$$

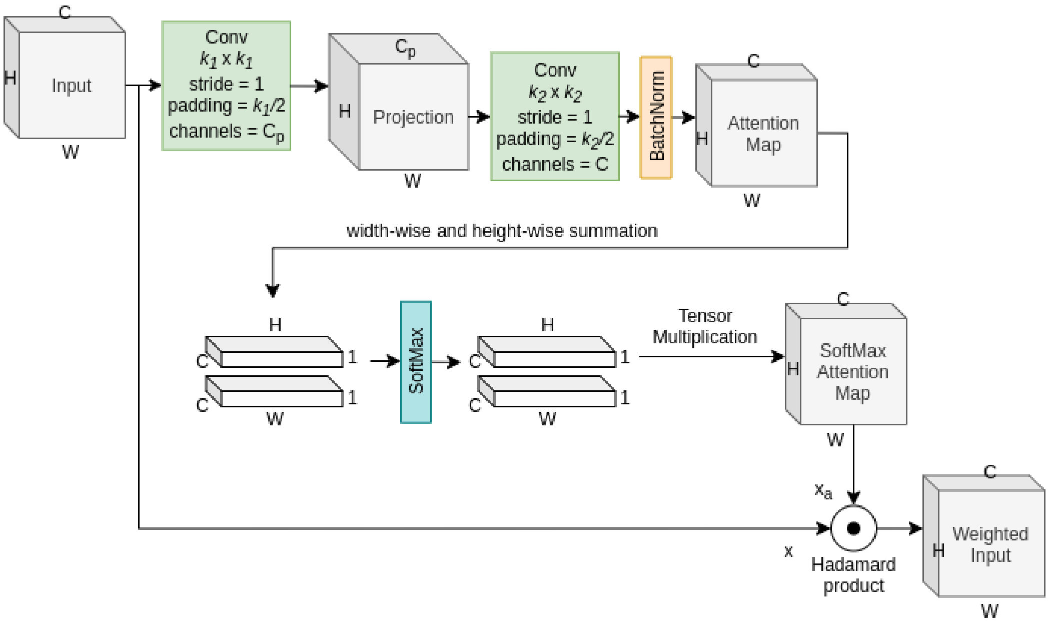


Figure 3.3: Proposed convolution input softmax soft-attention block

The attention map $x_a \epsilon \{x_{AM}, x_{SAM}\}$ is weighted with the image inputs x using Hadamard (element-wise) product and sent to the CAE as inputs. One of the attention blocks is added before the input layer of the baseline model while the rest of the architecture is retained as such.

3.6 Descriptive AutoEncoders

AutoEncoders in general are originally designed with the objective of data compression and representation. Utilizing the AE for the task of anomaly detection, though works well, is not primarily equipped for the task. So, the learnt latent representations from input images in the form of encodings from CAE are not separately clustered for normal and anomalous samples and are randomly distributed in the manifold space depending on the learning in the model. The most desirable property of an AD algorithm is to well distinguish between the representations of normality and anomaly. The absence of this property in CAE as anomaly detector leads to unfavourable outcome of accidental *perfect* reconstruction of anomalies degrading their performance as the anomaly scoring is directly related to reconstruction of data. To improve the discriminative nature of latent embeddings of CAE between normal and abnormal samples, the encoder should learn to identify features from an image that establish normality and concentrate the normal embeddings together in the embedding space. Hence, the usage of SVDD [69, 81] as a regularizer for the encoder of the AutoEncoder post-training, to help the model better segregate normal and abnormal samples is proposed as SVDD helps in concentrating the representations of normal samples together in the embedding space inside a multi-dimensional hyper-sphere and in producing embeddings that are farther from them for anomalous samples and this modification can also be imparted into any AE architecture with no extra computation cost.

3.6.1 Support Vector Data Description

Support Vector Data Description (SVDD) was first proposed as an alternative to the One-Class Support Vector Machine (OC-SVM) in [81]. Originally, OC-SVM was proposed to create a hyper-plane of separation between data and the origin to differentiate between normal samples and outliers. In SVDD, a hyper-sphere concentrates all the normal samples creating a spherical boundary of the normal feature space and anything outside the sphere can be deemed as an outlier. The spherical boundary is characterized by two parameters - the centre of the feature space C_s and the radius R which is the distance from the centre to any feature or data point on the boundary of the sphere. The strictness of the boundary can be relaxed using a slack parameter ξ and penalty factor C. The objective for SVDD on any input vector $x_i \epsilon \mathbb{R}^d$ can be represented by equation (3.13)

$$\begin{aligned}
&\min_{R,\ C_s} R^2 + C\sum_{i=1}^{n}\xi_i \\
&\text{subject to:} \\
&||x_i - C_s||^2 \leq R^2 + \xi_i \\
&\xi_i \geq 0 \\
&\forall i = 1, 2, 3...n
\end{aligned} \tag{3.13}$$

3.6.2 Encoder Regularization using SVDD

To improve the discriminative ability of CAEs towards anomalies and to establish the notion of normality through the learnt embeddings, SVDD is used as an objective function to train CAEs by being employed as a regularizer to the encoder in order to concentrate the *normal* embeddings together in a hyper-sphere and far apart from the embeddings of the outliers. The SVDD objective is applied on the *normal* embeddings ($z_i = f_e(x_i)$) from the CAE. Equation (3.13) can be solved using Lagrangian multipliers and the final objective function can be reformulated as a loss function to be used with the encoder part of the AutoEncoder as in equation (3.14). The *soft-boundary* variant as implemented by [69] is used for experiments.

$$\min_{\theta_e} \frac{1}{n}\sum_{i=1}^{n} ||f_e(x_i;\theta_e) - C_s||^2 + \lambda||\theta_e||^2 \tag{3.14}$$

This loss function can be directly combined with a primary objective function like MSE (equation 3.3). Through thorough experimentation, it was found that instead of training an AutoEncoder model from scratch, mere fine-tuning the encoder of the trained baseline CAE on the training data with a very low learning rate yielded better results and proved to be computationally efficient. Hence concerning fine-tuning, instead of taking all of the encoder parameters θ_e into consideration, it is sufficient to take only the parameters of the last layer of the encoder for regularization. Fine-tuning at a very low learning rate is imperative in achieving the right balance between making the CAE's embeddings of normal data concentrated together in the latent multi-dimensional space and preventing adverse effects on the reconstruction performance of the CAE. The centre of the hyper-sphere can be initialized as the mean of all the embeddings of the training set before fine-tuning.

3.7 Convolutional Recurrent Architectures

Videos are dynamic data with change in spatial information and motion of entities over time. Normal events in videos often exhibit regular temporal patterns when compared to portions with anomalies that exhibit contorted patterns and learning to identify them will give additional robustness for applications involving temporal coherence in inputs like video surveillance. Convolutional Recurrent architectures have been popular in video anomaly detection and are advantageous with the ability of learning spatio-temporal aspect of videos in a cogent way when compared to 2D CAE that can only learn spatial information. The usage of fully connected layers in LSTM for learning videos facilitates complete connections between inputs and the state transitions and, as a result, spatial information is not learnt effectively [75] and convolutional recurrent networks alleviate this problem by using convolutions that are inherently superior for encoding and propagating spatial information and the recurrent nature of the network makes it possible to coherently learn motion dynamic along the temporal dimension in the input videos. Similar to previous methods, this section focuses on learning regular or normal patterns in videos but using convolutional recurrent AutoEncoder architectures are more *natural fit* and they comprise of multiple layers of convolutional cells that are temporally *unrolled* to jointly learn spatial features and temporal coherence i.e. motion patterns in data towards reconstructing the normal inputs without supervision (labels). Hence, the primary hypothesis of the proposed solution is that the ability of convolutional recurrent AE to identify anomaly should be superior owing to the ability to jointly learn spatio-temporal correlation and features from data.

3.7.1 Possible Variants and Configurations

Before diving deep into the detailed workings and configurations of the networks under consideration, it is important to note and address that several configurations are possible and the best working ones are heuristically chosen for experimental analysis and the design configurations are universally maintained across the different flavours of recurrent convolutional networks in this study.

1. A variety of topologies using the convolutional recurrent units is possible. For example, one could engineer a CRAE with multiple convolutional layers in each cell or multiple convolutional recurrent layers with one convolutional operation in each or hybrid models that can predict and reconstruct frames jointly. This research focuses only fundamental convolutional recurrent architectures for video anomaly detection.

2. Several activation functions are possible at each state and we use the ones listed in the equations according to heuristics.

3. The bias at each layer is optional and the performance does not differ much with ignoring the bias and help to reduce the number of learnable parameters.

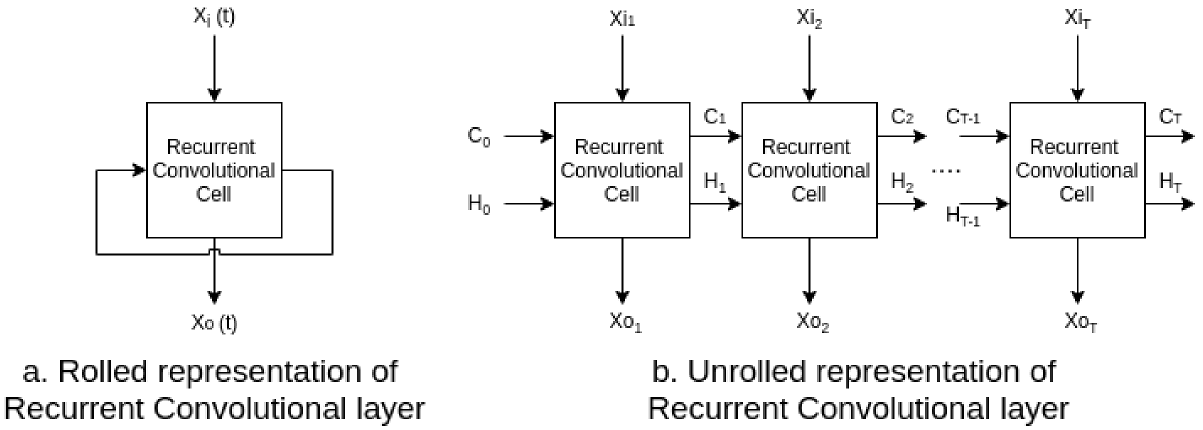
4. There is possible variation in terms of state design where some or all convolutional operations can be replaced by Hadamard (element-wise $\odot$) product to accommodate consistency in output shape.

5. Depending on the type of deep learning framework, the input and output shapes can be interchanged and for representational purposes the input shape of $B \times T \times W \times H \times C$ is used for all the experiments in this work (explained in detail in the later sections).

3.7.2 Convolutional Recurrent Cells

ConvLSTM [75] have become popular over the past few years for learning dynamic visual inputs. They integrate visual learning from convolutional layers with temporal learning in LSTM layers with the fully connected layers in LSTM substituted by convolutional operation and hence capturing better spatio-temporal features. Although ConvLSTM has been the most popular choice for video tasks, there are other possible variants such as ConvRNN and ConvGRU based on the different recurrent networks employed. For all the convolutional recurrent architectures for video-related tasks, ConvCells are the building blocks to develop various architectures of different complexities. In ConvCells, the internal states for a dynamic, directed, acyclic graph to learn and model sequences over several time-steps of unrolling. In this section, different types of ConvCells are discussed in detail in order to explore various convolutional recurrent architectures for video anomaly detection. The pictorial representation of a generic ConvCells is presented in figure 3.4 where X_i, X_o denote the inputs and outputs, C_t, H_t represent the cell state and hidden state of the ConvCells[3] with a suffix to denote the time step among a total of T time steps of unrolling. Similar to recurrent neural networks, ConvCells utilize back-propagation through time (BPTT) to propagate gradients to early time steps and different parts of the cell to facilitate sequential learning. Similar mechanism is used in decoder but with transpose convolutions instead of normal convolutions to upsample input data or activation maps.Moreover, blocks with *2D Conv* marked in representations in the following sections are made up of a convolutional (or

[3] The states are initialized randomly


Xi (t)
Recurrent Convolutional Cell
Xo (t)
Xi1
Xi2
XiT
C0
H0
C1
H1
C2
H2
CT-1
HT-1
....
CT
HT
Recurrent Convolutional Cell
Recurrent Convolutional Cell
Recurrent Convolutional Cell
Xo1
Xo2
XoT

a. Rolled representation of Recurrent Convolutional layer

b. Unrolled representation of Recurrent Convolutional layer

Figure 3.4: Generic representation of recurrent convolutional layer operation

transpose convolutional layer), a batch normalization layer and a ReLU layer in tandem. For efficient computation, multiple *2D Conv* blocks on the same input can be aggregated, processed and then separated since only the number of kernels will increase and other parts of the network will operate normally.

ConvRNN cell

A ConvRNN cell uses vanilla *Recurrent Neural Networks* [70] with fully connected layers replaced by convolution operation to facilitate spatio-temporal learning. ConvRNN cell is simple in terms of internal structure and consists of a hidden state and an output state with each state containing associated weights. The current hidden state of a time step is a function of the previous hidden state and the current input, and is passed to the next time step. The current output state on the other hand, is a function of the current hidden state with an activation function (sigmoid σ). The input to the ConvRNN cell at each time step t is of the shape $B \times W \times H \times C$ where B is the batch size, $W \times H$ the resolution of the input frame and C the number of channels. This enables reusability of the states through the dynamic unrolling process to persist the learnt information over time in the *memory.* The equations governing the operation of ConvRNN cell are shown in equation (3.15) where $*$ represents the convolution operation, W, b the weights and biases, X, H, O the input, hidden and output states respectively at the time step t.

$$
\begin{aligned}
H_t &= tanh(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \\
O_t &= \sigma(W_{ho} * H_t + b_o)
\end{aligned}
\tag{3.15}
$$

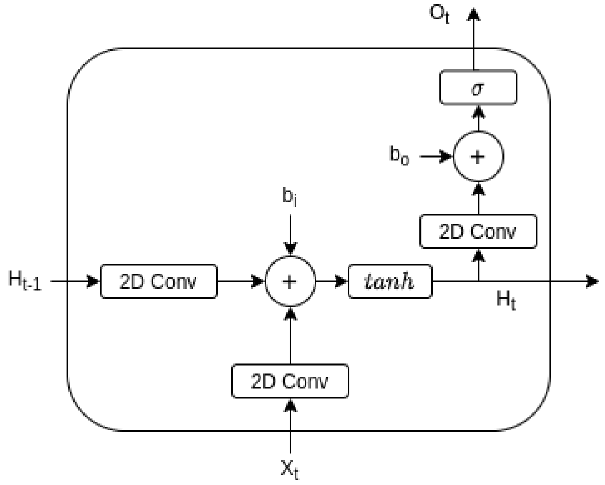


Figure 3.5: Internal structure of a ConvRNN cell

ConvLSTM cell

LSTM [36] has proven to be better than RNNs on sequence modelling tasks with tremendous performance improvements due to the ability of LSTMs to avoid vanishing and exploding gradients and maintaining a persistent *cell state* to learn and retain long term dependencies better. The ConvLSTM cell consists of input, output and forget gates along with a cell state. The three gates regulate the flow of information in and out of the cell whereas the cell state retains memory over long period of time. The input shape and replacement of fully connect layers with convolution operation are same as in ConvRNN cell. The equation (3.16) shows the operation of ConvLSTM cell for inputs as discussed in the previous section.

$$\begin{aligned}
&i = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i) \\
&f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f) \\
&C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
&o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o) \\
&H_t = o_t \odot \tanh(C_t)
\end{aligned} \tag{3.16}$$

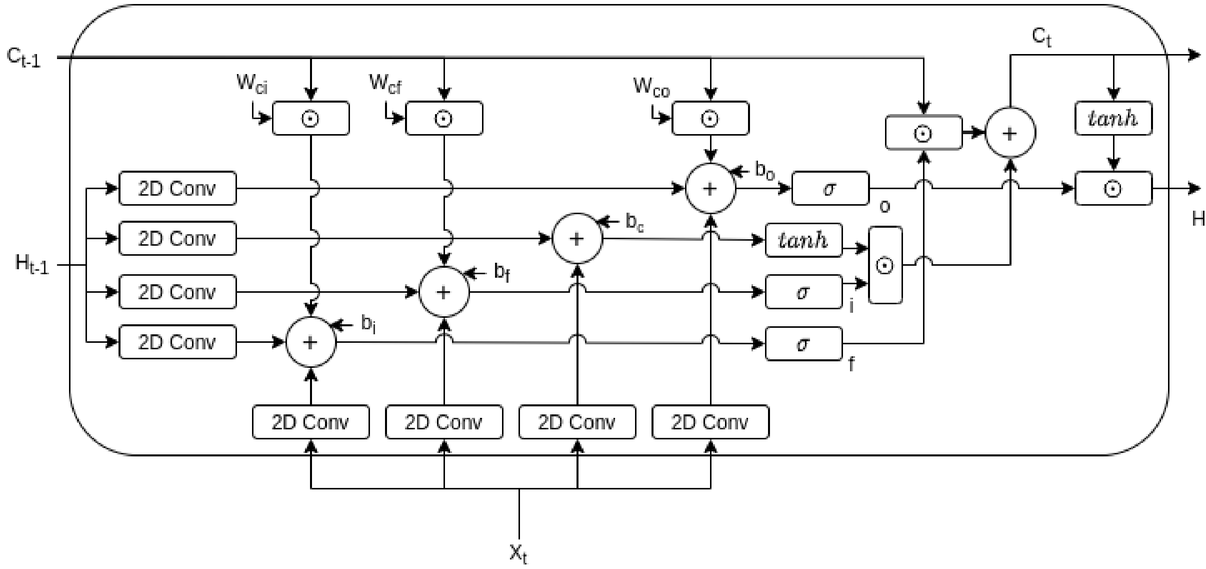


Figure 3.6: Internal structure of a ConvLSTM cell

ConvGRU cell

Gated Recurrent Units (GRU) [23] brought further improvements to sequential learning over LSTMs (and RNNs) with fewer parameters and simpler internal structure with clear improvements observed in speech, language and music modelling. GRU consists of a reset gate r and an update gate u to regulate information inside the cell through an activation gate a which is a function of the previous hidden state and the current updated input. The hidden state is the transformed activated state that is passed to the next time step of unrolling.

$$\begin{aligned}
u &= \sigma(W_{xu} * X_t + W_{hu} * H_{t-1} + b_u) \\
r &= \sigma(W_{xr} * X_t + W_{hr} * H_{t-1} + b_r) \\
a &= tanh(r \odot (W_{ha} * H_{t-1}) + W_{xa} * X_t) \\
H_t &= (a \odot (1 - u)) + (u \odot H_{t-1})
\end{aligned} \tag{3.17}$$

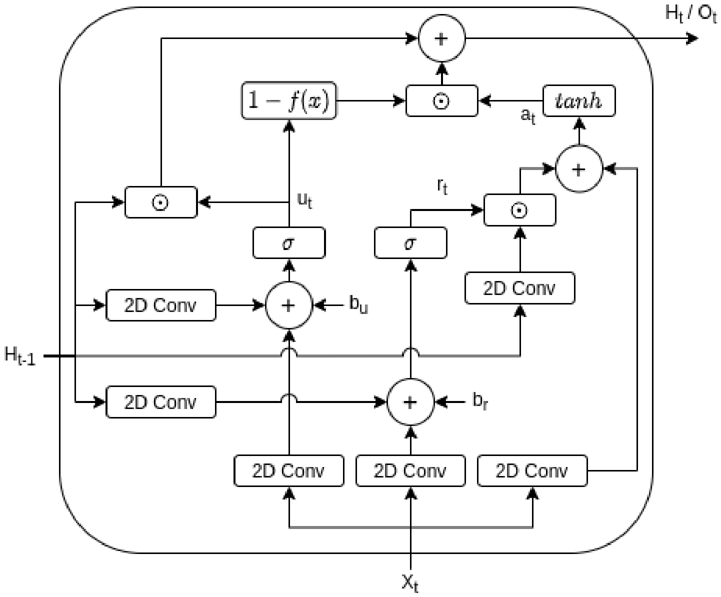
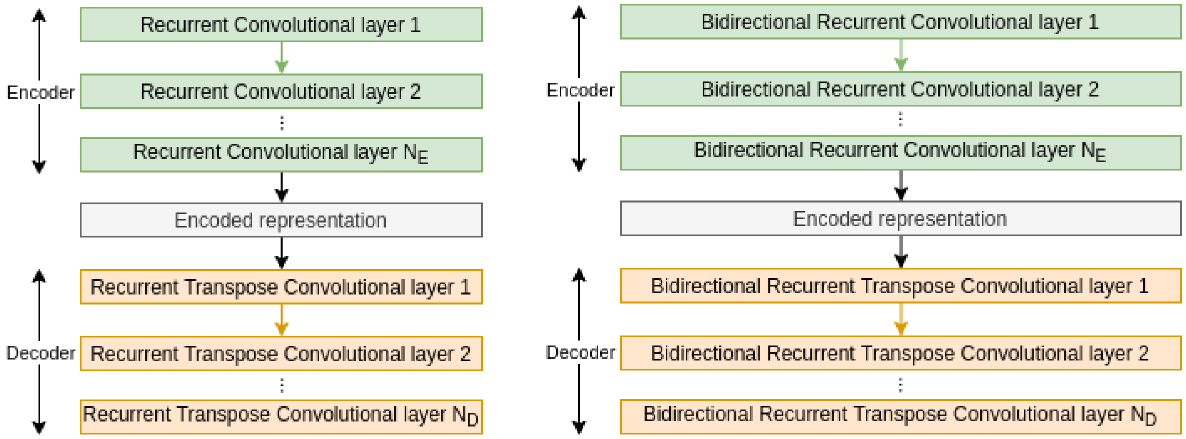


Figure 3.7: Internal structure of a ConvGRU cell

3.7.3 Convolutional Recurrent AutoEncoders

Convolutional Recurrent AutoEncoders use ConvCells as building blocks in each layer to abstract spatio-temporal information over multiple layers in the encoder into a compact representation, from which transpose ConvCells in the decoder are used in layers to reconstruct data into the input dimension. As the input video clip is compressed to a latent representation for eventual reconstruction with normal motion of objects in the frame, it forces the model to learn the essential attributes that represent normality from the inputs. Three variants of convolutional recurrent AutoEncoders are presented in this work which are *ConvRNN CAE*, *ConvGRU CAE*, *ConvLSTM CAE* which share a common structure except for their respective variant of convolutional recurrent cells. MSE is used as the objective function for minimization and all other operation is similar to that of CAE except the input having an extra, temporal dimension $B \times T \times W \times H \times C$ where at each time step, the network processes a batch of frames of the time step and learns the complete video as the network is unrolled for T time steps.


Encoder
Recurrent Convolutional layer 1
Recurrent Convolutional layer 2
Recurrent Convolutional layer N_E
Encoded representation
Decoder
Recurrent Transpose Convolutional layer 1
Recurrent Transpose Convolutional layer 2
Recurrent Transpose Convolutional layer N_D
Encoder
Bidirectional Recurrent Convolutional layer 1
Bidirectional Recurrent Convolutional layer 2
Bidirectional Recurrent Convolutional layer N_E
Encoded representation
Decoder
Bidirectional Recurrent Transpose Convolutional layer 1
Bidirectional Recurrent Transpose Convolutional layer 2
Bidirectional Recurrent Transpose Convolutional layer N_D

(a) Structure of recurrent convolutional AutoEncoder

(b) Structure of bidirectional recurrent convolutional AutoEncoder

3.7.4 Bidirectional Convolutional Recurrent AutoEncoders

Similar to the convolutional recurrent AutoEncoder discussed in the previous section, a bidirectional convolutional recurrent AutoEncoder is similar in terms of the architecture except that the convolutional recurrent layers are bidirectional that can learn from the temporal and reversed inputs under the intuition that the AutoEncoder can learn both from past and future input time steps and have proven advantages and performance enhancement in tasks involving understanding context from data, especially predictive tasks. The bidirectional convolutional recurrent cell consists of two modules, a *forward* and *backward* model each equivalent to a vanilla convolutional recurrent cell. The forward module operates normally as stated in the previous section and the backward module operates by learning information from the temporally-reversed input data batch as shown in figure 3.8. Finally, the output from the two modules is combined to produce the final five-dimensional activation maps and passed on to the next layer. Although there are several methods to combine the outputs from the forward and backward modules like addition, concatenation, multiplication, dot product etc. combination by calculating the *mean* is used for the experiments. The overall structure of a bidirectional convolutional recurrent AutoEncoder is shown in figure 3.8b. Different variants can be created depending upon the type of recurrent convolutional cell used such as the *bidirectional ConvRNN AutoEncoder*, *bidirectional ConvGRU AutoEncoder* and *bidirectional ConvLSTM AutoEncoder*.

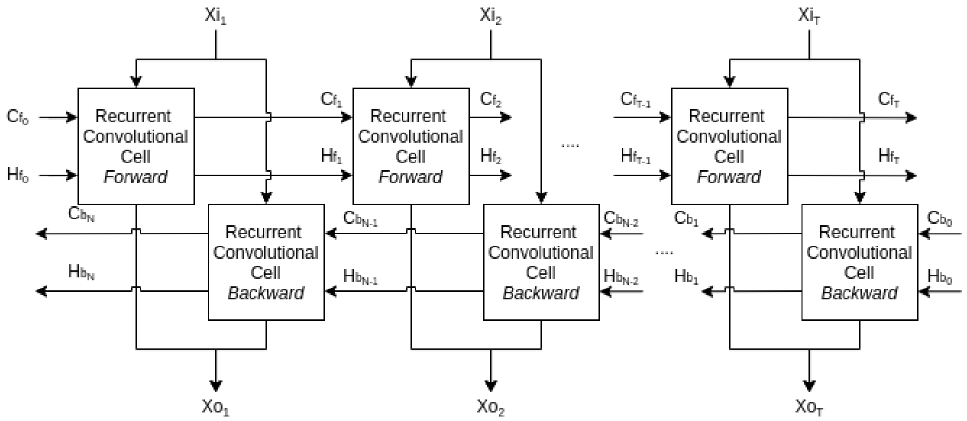

Xi1
Xi2
XiT
Cf0
Hf0
Recurrent Convolutional Cell Forward
Cf1
Hf1
Recurrent Convolutional Cell Forward
Cf2
Hf2
....
CfT-1
HfT-1
Recurrent Convolutional Cell Forward
CfT
HfT
CbN
HbN
Recurrent Convolutional Cell Backward
CbN-1
HbN-1
Recurrent Convolutional Cell Backward
CbN-2
HbN-2
....
Cb1
Hb1
Recurrent Convolutional Cell Backward
Cb0
Hb0
Xo1
Xo2
XoT

Figure 3.8: Representation of a generic bidirectional recurrent convolutional layer

3.7.5 Sequence to Sequence Convolutional Recurrent models

Seq2Seq models are a special blend of AutoEncoders and recurrent architectures belonging to the category of *auto-regressive* architectures that are used for modelling time-series data with the goal to learn sequence from domain and to transform the learnt knowledge into prediction in a different domain and are widely used in NLP tasks. The goal of Seq2Seq models for anomaly detection is to learn normalcy and predict the future frames from a set of seed frames that are provided as the input to the model as opposed to mere reconstruction as in the previously discussed models. The hypothesis is that the normal patterns of motion in videos learnt while training can be easily predicted similar to *cause and effect* phenomenon and the model will be able to predict the future of normal events with a high degree of certainty almost matching the rest of the normal input video clip. This model is trained with sets of input seed frames with the objective of predicting the next n frames and, the error between the actual rest of the frames and the predicted set is minimized using MSE as the objective function. Eventually, a well-trained model on normal data will fail to predict the future of an initiated anomalous event. This comparison between an actual and predicted set of frames helps in the quantification of anomalies. A convolutional recurrent AutoEncoder with an encoder and a decoder can be re-purposed into a Seq2Seq model where the major differences are in the inputs and the overall learning mechanisms. For the experiments, three variants of Seq2Seq architecture *Seq2Seq ConvRNN CAE*, *Seq2Seq ConvGRU CAE* and *Seq2Seq ConvLSTM CAE* are used. The structure of the architecture is represented in figure 3.9.

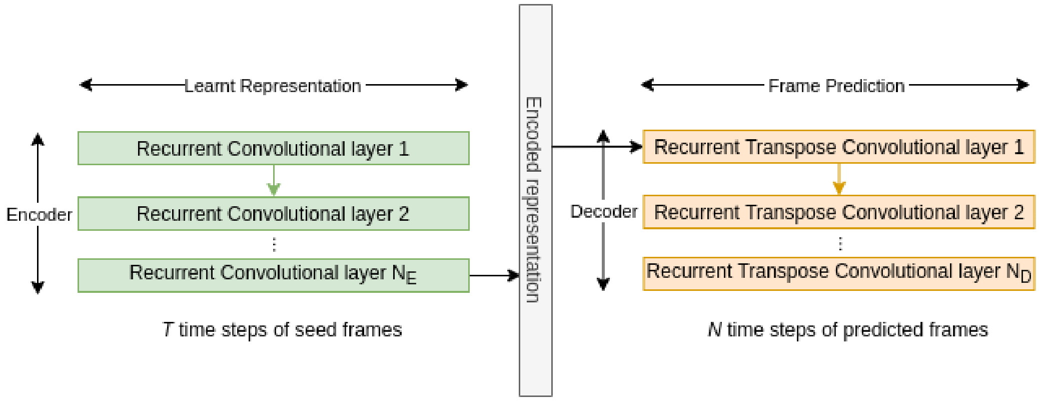

Learnt Representation
Frame Prediction
Encoder
Recurrent Convolutional layer 1
Recurrent Convolutional layer 2
Recurrent Convolutional layer N_E
Encoded representation
Decoder
Recurrent Transpose Convolutional layer 1
Recurrent Transpose Convolutional layer 2
Recurrent Transpose Convolutional layer N_D
T time steps of seed frames
N time steps of predicted frames

Figure 3.9: Structure of Seq2Seq recurrent convolutional architecture

3.8 Explainability in Convolutional AutoEncoders

Most of the prevalent methods use the residual reconstruction error between the inputs and the reconstructions to explain the anomalies to a certain extent. But the partial reconstruction or absence of an anomaly does not exactly explain the nature of anomalies. This warrants the need for explaining the nature of anomalies along with their presence. The reconstruction based approach using AEs for detecting anomalies doesn't accompany class predictions and since most XAI frameworks operated on class predictions and employing a classifier to AE completely changes the paradigm of the solution, methods that allow the use of CAE without any major modifications are explored. Four major XAI methods are explored in this section along with the modifications that are required to enable them to be used with CAEs. Most XAI methods work on the same fundamental concept of quantifying the input contribution towards the output.

3.8.1 Layer Relevance Propagation (LRP)

Layer Relevance Propagation (LRP) [8] is an XAI approach based on feature relevance, distribution and conservation as the output prediction score is propagated layer by layer towards the input by a backward pass based on a redistribution rule to find out the relevance or the contribution of each input feature towards the outcome of the prediction. The trained weights that are learnt by the model are utilized for the distribution of relevance in each layer with a condition that the total relevance value R at any given layer is conserved for a particular class c whose decision is explained. Equation (3.18) without ϵ shows the formula to calculate relevance between neurons in two consecutive layers j and k connected by weight w_{jk} whose activation is given by a. This is the proportion of influence/contribution of neuron in j towards the neuron in layer k. This is the simplest rule of LRP though more complex rules exist. A variant of ϵ-LRP as in equation (3.18) is used in this work, where a small constant ϵ is added to the denominator that compensates the weakness of relevance propagated to k and it can alleviate the effect of noise-producing sparse explanations.

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} \tag{3.18}$$

3.8.2 Local Interpretable Model-agnostic Explanation (LIME)

Local Interpretable Model-agnostic Explanation (LIME) [67] is a model-agnostic XAI framework that uses a local, *surrogate* models to explain the class predictions or regression of any machine learning model. The surrogate model imitates the behaviour of the parent model to ensure local fidelity. LIME analyses the deviation in class predictions with respect to multiple variations of reference input that are generated with added noise or change in input features. For example, the input images of a CNN classifier are altered with variation in terms of small patches of contiguous pixels known as *interpretable components*. The set of perturbed samples of the reference image is utilized to generate prediction scores with respect to change in local neighbourhoods. The surrogate model (mostly a simple linear model) is then trained on this set to learn to weight the perturbed regions of pixels and the large weights among the learnt parameters denote the explanations for the reference image to indicate the essential attributes that vastly contribute towards the prediction or decision of the particular class of the reference image under consideration.

$$exp(x) = argmin_{f_s \epsilon F_s} \mathcal{L}(f_o, f_s, \pi_x) + \Omega(f_s) \tag{3.19}$$

The explanation of data sample x using LIME can be given by a surrogate model $f_s \epsilon F_s$ (where F_s is the family of surrogate models) such that it has minimum loss between its prediction and the prediction of the original model f_o with the least possible complexity $\Omega(f_s)$ with the neighbourhood around x given by the proximity measure $\pi(x)$ as denoted in equation (3.19). This interpretable surrogate model f_s after training on the perturbed data X_p created based on the original sample x can help in explaining the prediction of the original model f_o (a CAE in this case).

3.8.3 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) [55] is a popular feature attribution based XAI method to measure the importance of input features towards predictions. SHAP is a additive attribution technique that employs *Shapely values* which is a concept from coalitional game theory that describes the fair distribution of agents towards a desirable outcome which in this case is the fair contribution of each pixel in the input image towards the final prediction. SHAP provides consistent method for the global interpretation of each data sample among the dataset owing to the distinct properties of Shapely values. SHAP operates by replacing input features with random variable to determine their contributions from the relative difference between original and modified predictions. SHAP incorporates the idea of several other techniques such as Gradient-based explanations, saliency, LIME, DeepLIFT etc. KernelSHAP is such a combination of ideas that was proposed as the better alternative to LIME and the weights of kernelSHAP can be determined by the equation (3.20) where $|z'|$ is the number of features considered for the coalition and M is the maximum coalition among features.

$$\pi_z(z') = \frac{(M-1)}{(\frac{M}{|z'|})|z'|(M-|z'|)} \tag{3.20}$$

3.8.4 Counterfactuals

The counterfactual [88] method of explanations for neural networks is another model-agnostic approach to find the minimal change in feature values that can cause a drastic alteration to the model's class prediction. It is important to identify and understand the smallest relative change by which a normal sample is altered to make it an anomalous sample. Since Counterfactual is a generalized idea and the algorithm is usually data and model-specific, a simple algorithm that creates rectangular region-wise alterations that cause the highest reconstruction errors is applied to understand the image region. Counterfactuals and adversarial examples can help in determining the perturbations or changes in input that drive the model towards other extremes of predictions.

3.8.5 Modification to AutoEncoders for Applying Generic XAI Frameworks

Most of the above algorithms are built around predictions from a classifier and anomaly detection can be considered as a binary classification problem. But reconstruction-based anomaly detection using CAEs do not provide probability scores directly and simple adjustments are made to CAE architecture for this case. For any dataset, the maximum value of reconstruction error from the anomalous test set (anomaly type) is found and used to normalize the reconstruction losses between 0 and 1. The normalized reconstruction loss can be considered as an anomaly score where a lower score denotes the normality of the input sample. The optimal threshold (δ_o) of anomaly detection (binary classification) is a threshold δ_i that can be manually fixed or found using from the set of all n thresholds represented by δ in the Receiver Operator Characteristics curve that gives the maximum value of geometric mean of *sensitivity* (true positive rate *TPR*) and *specificity* (1 – False Positive Rate *FPR*) as in Equation (3.21). This optimal threshold (δ_o) can help in finding the required balance between precision and recall according to the nature of the application. After the threshold is fixed, the *residual error normalization* can be implemented as a parameter-less (lambda) layer in most of the deep learning frameworks and can be added at the end of CAE.

$$\begin{aligned} &\delta_o = \underset{\delta_i \epsilon \delta}{\operatorname{argmax}} \sqrt{TPR(\delta_i) \times (1 - FPR(\delta_i))} \\ &\text{where } \delta = \{\delta_1, \delta_2, \dots \delta_n\}, \text{ index } i \epsilon \{1, 2, \dots n\},\ 0 \leq \delta_i \leq 1 \end{aligned} \tag{3.21}$$

3.9 Summary

In this chapter, *four* novel models that can help to augment the anomaly detection performance of reconstruction based convolutional AutoEncoders were discussed which can be used for both image and video tasks. The advantages of the proposed methods are that they are model-agnostic with the potential to be used for any CAE and are computationally effective as the increase in the number of parameters is little to nothing. The rationale, advantages, design and intuition behind each of the models were discussed and the experimental results showing their effectiveness are discussed in Chapter 4 to validate the proposition. Moreover, the feasibility of using explainability frameworks to CAEs was discussed in detail along with challenges and possible solutions to overcome them.

Chapter 4

Experiments and Analysis

4.1 Introduction

The previous chapter covered the proposed solutions for improving visual anomaly detection. This chapter deals with all the experiments involved in achieving the desired objective with the proposed solutions.

4.2 Datasets

To show the versatility of each of the proposed solutions, the experiments were conducted on a variety of datasets from different domains. For the experiments, 3 image anomaly detection datasets were used. The detailed descriptions of the datasets are provided in the following sections.

4.2.1 HAM10000

HAM10000 [86] is a medical image dataset consisting of dermatoscopic images with pigment skin lesions from multiple populations, acquired and stored by different modalities with 7 categories - Actinic keratoses and intraepithelial carcinoma / Bowen's disease (AKIEC), basal cell carcinoma (BCC), benign keratosis-like lesions (BKL), dermatofibroma (DF), melanoma (MEL) and vascular lesions (VASC) and melanocytic nevi (NV) which is considered as the normal class. For the experiments, 6600 normal samples are randomly sampled for training & validation and the rest of the normal samples along with anomalous samples are used for testing purposes. The statistical overview of the dataset is presented in Table 4.1

Type	NV	AKIEC	BCC	BKL	DF	MEL	VASC
Samples	6705	327	514	1099	115	1113	142

Table 4.1: Statistical overview of HAM10000 data set

4.2.2 Daytime Driver Distraction Dataset

The Daytime Driving Distraction dataset (DDDS) proposed in [63] is an anomaly detection dataset that depicts an autonomous and assisted driving scenario. It consists of images capturing upper body movements indicating the behaviour of 25 drivers in a simulated environment. The data set is equally distributed with four classes involving different scenarios while driving - Normal, Talking, Texting and Operating GPS. The statistics of the driver distraction data are shown in Table 4.2. The samples containing the behaviour of 16 randomly selected drivers normal images are used for training the proposed models in all the experiments and the rest of the data are used for testing.

Type	Normal	Talking	Texting	Operating GPS
Samples	4993	4921	4991	4926

Table 4.2: Statistical overview of Daytime Distraction Driving data set

4.2.3 MVTec

MV Tec dataset [17] is a widely used image anomaly detection dataset collected under the context of industrial inspection and quality control during manufacturing. It consists of 5 categories of textures - Carpet, Grid, Leather, Tile, Wood and 10 categories of objects - Bottle, Cable, Capsule, Hazelnut, Metal nut, Pill, Screw, Toothbrush, Transistor, Zipper. The dataset contains over 70 varieties of defects. For each category, the train set of a normal set and the test set with separate normal and anomalous sets are available and they are utilized as such for all the experiments. Table 4.3 shows the overview of the dataset.

Category	Train samples	Normal test samples	Anomalies
Carpet	280	28	89
Grid	264	21	57
Leather	245	25	92
Tile	230	33	84
Wood	247	19	60
Bottle	209	20	63
Cable	224	58	92
Capsule	219	23	109
Hazelnut	391	40	70
Metal Nut	220	22	93
Pill	267	26	141
Screw	320	41	119
Toothbrush	60	12	30
Transistor	213	60	40
Zipper	240	32	119

Table 4.3: Statistical overview of MV Tec dataset

4.2.4 CUHK Avenue

CUHK Avenue dataset [53] is a surveillance video anomaly detection dataset that consists of 16 train and 21 test videos where each video is about 2 minutes long with frame-level ground truth. The anomalies consist of running, jogging, jumping, unattended objects, throwing objects in the air, walking off-direction, moving closer and away from the camera, loitering etc. occurring both in the background and foreground. The training set consists of a few un-recorded anomalies too. The dataset also consists of frame-level and binary ground-truth annotations denoting the presence of anomalies.

4.2.5 UCSD Pedestrian 1 & 2

UCSD Ped 1 and 2 [57] are video anomaly detection datasets that were collected from a stationary camera overlooking walkways capturing pedestrian motion in two different scenes. UCSD Ped 1 contains 34 train videos and 36 test videos whereas Ped 2 contains 16 train videos and 12 test videos with fewer anomalies but both datasets have frame-wise temporal annotations. The common anomalies are naturally occurring events that include bikers, trucks, carts, people on skateboards and people walking off-direction or on the grass.

The frame-level ground-truth labels are provided with the dataset.

4.2.6 Subway Entrance and Exit

The subway dataset [1] is a video anomaly detection dataset that depicts a surveillance scenario with two cameras in a subway station. The Entrance set consists of a 1 hour 35 minutes long video where the first 15 minutes is used for training and the rest for testing with 66 anomalies. The Exit set consists of a 45 minutes long video with the first 5 minutes of the video for training and the remainder for testing with 19 anomalies though there are more when inspected. The anomalies consist of jumping turnstiles, skipping a payment, walking in the wrong direction, running, loitering etc. The dataset consists of event-level ground truth and hence a window of 15 frames is used on either side of the temporal label to replicate the labels although some events seem to last longer up to 50 frames.

Dataset	# Train videos	# Test Videos	Average frames/ video	# Anomalies
Subway Entrance	1	N/A	121,749	66
Subway Exit	1	N/A	64,901	19
UCSD Ped 1	34	36	200	40
UCSD Ped 2	16	12	164	12
CUHK Avenue	16	21	835	47

Table 4.4: Statistical overview of video datasets

4.3 Evaluation Metrics

The evaluation metrics determine the quality of results and after rigorous review, a set of evaluation metrics were consolidated for use in the results of all the experiments. Anomaly detection can be considered as an imbalanced binary classification problem and metrics that apply to binary classification can hence be used. The presence of anomaly can be regarded as the positive class and normal samples belonging to the negative class. The basis for all the metrics can be formulated using the following elementary metrics as shown in Table 4.5

Metric	Definition
True Positives (TP)	Number of correctly predicted positive class samples by a model
True Negatives (TN)	Number of correctly predicted negative class samples by a model
False Positives (TP)	Number of incorrectly predicted positive class samples by a model
False Negatives (TN)	Number of incorrectly predicted negative class samples by a model
Precision	$TP/(TP + FP)$
True Positive Rate (TPR)/ Recall/ Sensitivity	$TP/(TP + FN)$
Specificity	$TN/(TN + FP)$
False Positive Rate (FPR)	$FP/(TN + FP)$

Table 4.5: Definition of elementary metrics

4.3.1 AUC-ROC score

The area under the Receiver Operator Characteristics Curve (AUC-ROC) score is a classification performance measure at different thresholds that quantifies the degree of separability. The Receiver Operator Characteristics (ROC) curve is a plot of True Positive Rate (TPR) on the y-axis vs False Positive Rate in the x-axis (FPR) and the AUC-ROC score is given by the area under it. A high AUC-ROC score (ideally 1.0) denotes that the model is better at distinguishing between positive and negative classes.

4.3.2 AUC-PR score

The area under the Precision-Recall curve (AUC-PR) is a classification performance that combines precision and recall into a single metric. It is given by the area under the curve in the plot between precision and recall. It can also be defined as the average of the precision values for each recall.

4.3.3 F1-Score

F1-Score combines precision and recall values using harmonic mean between them and measure the balance between both. AUC-PR and F1-Score can both be used to select the optimal threshold to achieve a balance between the two.

4.3.4 Equal Error Rate

Equal Error Rate (EER) is the point on the ROC curve where the probability of miss-classifying positive and negative samples are equal i.e. the lower the EER, the better performing a model is.

4.4 Experimental Setup

The general experimental setup for the models, common to all the experiments is discussed in this section. Special configurations and modifications to model design and architectures will be mentioned in the corresponding sections. All the testing experiments were conducted using PyTorch based on Python on a computer with Intel Core i7-6700K, 32 GB RAM and NVIDIA GeForce GTX 1070 8GB VRAM. The codebase for the experiments is available at https://github.com/ambareeshravi/Thesis_VideoAnomalyDetection/

4.4.1 General Configurations

All the models in our experiments are trained using Adam optimizer [43] for 300 epochs and a starting *learning rate* of 1×10^{-3} equipped with learning rate reduction on validation loss plateau by a factor of 0.75 and patience of 4 epochs. The training process ends at 300 epochs for all the experiments irrespective of convergence but it is important to note that almost all the models converge within 250 epochs. The training procedure is also bolstered with early stopping with patience of 8 epochs. Unless stated otherwise, *Mean Squared Error* is used as the objective function for minimization. The results of the experiments are reported as the average of 3 individual runs.

4.4.2 Baseline Convolutional AutoEncoder

All the experiments pertaining to the proposed modifications were conducted as an extension to a baseline Convolutional AutoEncoder and against which their effectiveness on multiple datasets are demonstrated. The baseline CAE consists of 5 strided convolutional layers in the encoder and 5 transpose convolutional layers in the decoder. Strided convolutions are used to avoid explicit feature aggregation by non-learnable layers like pooling. The encoder and decoder are mirror replica of each other with the addition of padding to adjust the output size in the decoder. The number of kernels (filters) in the encoder are 64,64,64,96,96 in order and the same in the decoder but in the reverse order, with the last one equivalent to the number of channels. Each encoder layer consists of kernels of size 3×3 with stride 2 followed by BN [39] layer which is known to improve the performance, speed of learning and to alleviate vanishing and exploding gradients ensuring smooth propagation of the gradients to the early layers in CNNs and a ReLU [2] activation layer that introduces non-linearity by retaining the positive values while clipping the negative values to zero. The last layer has kernels with stride 1 and the embedding size is 1×1536. The configuration is the same for the decoder with the exception that transpose convolution layers instead of convolutional layers and Leaky ReLU layers with 0.2 leakiness which alleviate the *dying ReLU* problem by returning low-weighted, negative values instead of clipping negative values to zero instead of ReLU being used. As the input images are normalized to values between 0 and 1, *Sigmoid* activation is used in the last layer of the decoder. This model configuration was carefully chosen to show the effectiveness of the proposed approaches that a significant boost in performance can be achieved even with such a small model. This model is used in the experiments in sections 4.7, 4.8, 4.9, 4.10 and this model is referred to as *BCAE1*.

4.5 Kernel Strengthening AutoEncoder

First, a baseline convolutional AutoEncoder, BCAE2 is chosen with 5 layers in the encoder and 5 layers in the decoder. The number of kernels in the encoder are $64, 128, 256, 512, 300$. The encoder of BCAE2[1] consists of strided convolutions and strided transpose convolutions in the decoder, the details of which is shown in Figure 4.1. Each layer is followed by BN and activation layer - ReLU in encoder and Leaky ReLU in decoder. Sigmoid activation is used in last layer of the decoder since the inputs are normalized in the range of $[0, 1]$. The embedding size of the CAE is 1×300.

Another CAE is configured with strengthened kernels by using ACB and ADB layers instead of conventional convolutional and transpose convolution layers in the first 4 and last 4 layers of the encoder and decoder of the BCAE2 respectively. This model is referred to as KS CAE Other layers such as the BN and the ReLU are retained. The BN and ReLU

[1] different from the previously mentioned general baseline model BCAE1

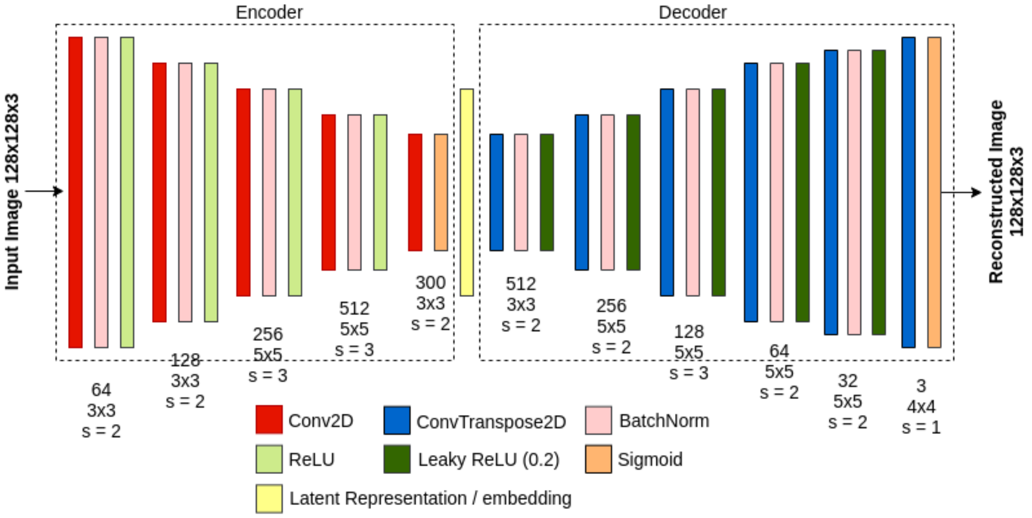


Figure 4.1: The architecture or BCAE2 model

layers also help in adjusting the magnitude of the activations since the activations from the horizontal, vertical and square convolutional kernels are summed up and passed to the consecutive layers. *To show the efficiency of kernel strengthening, only half the number of kernels* are used in the middle layers of KS CAE. The proposed modified architecture is shown in Figure 4.2

4.5.1 Experiments

The experiments were conducted on two image datasets - *HAM10000* and *DDDS*. The comparison was made between Baseline CAE and KS CAE models on the datasets i.e. without and with the proposed modification to evaluate its effectiveness. MSE was used as a loss function and as the anomaly score for the experiments. The models were trained in parallel under the same parametric conditions such as epochs, batch size, input resolution, learning rate, and even on the same batches of shuffled data every epoch to ensure fair benchmarking of results and were tested on the same randomized test sets.

4.5.2 Results and Analysis

The results of experiments on *HAM10000* and *DDDS* datasets are shown in tables 4.6 and 4.7 respectively along with the experiments from the next sections. It is apparent from the

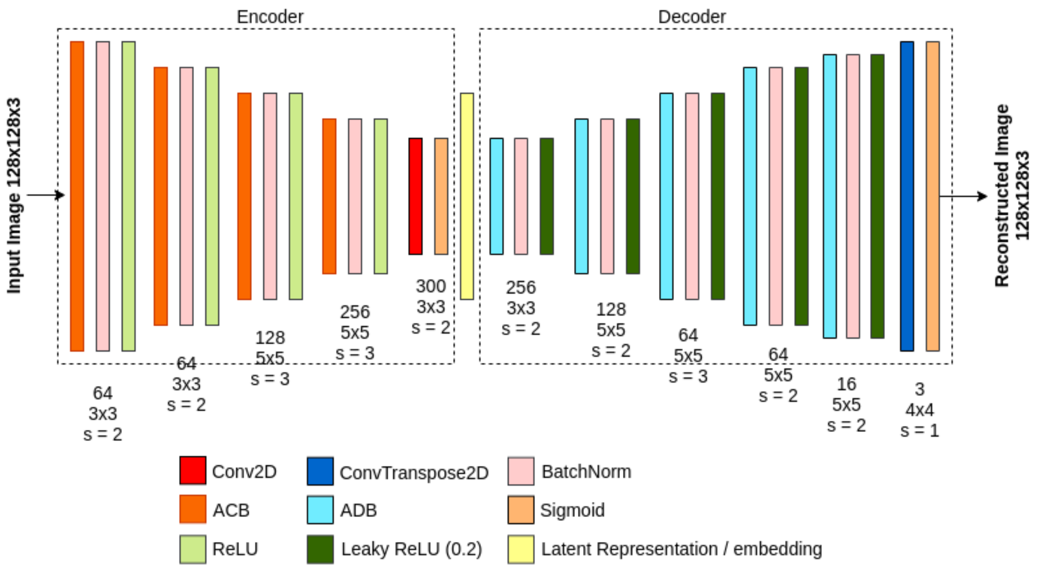


Figure 4.2: The proposed Kernel Strengthened Convolutional AutoEncoder architecture

tables that the proposed KS CAE outperforms the baseline model with almost double the number of filters per layer by a good margin on both the loss functions. For example, DF is the most complicated anomaly type in *HAM10000* which is virtually indistinguishable from normal data and the proposed method increases the AUC-ROC score by 4% from 0.61 to 0.65 along with improvements with respect to other anomalies types. The improvements can also be seen on *DDDS* data set. This shows the general ability of the method irrespective of the CAE configuration. The results on the above data sets are clear, substantial improvements over [94] that uses sparse representation Convolutional AutoEncoders with larger sizes and number of kernels. [49] proposes to use Deep Isolation Forests trained on data from all classes except for each anomaly class and the proposed model that is trained only on the normal class still outperforms their performance on HAM10000. The proposed method also performs better than [48] which uses a huge ResNet 154 as a teacher model for a vision transformer to detect Melanoma (MEL) in HAM10000 data set by training on all other classes. For DDDS dataset, there is no other benchmark than [94] since it is a new dataset and the proposed method out-performs the results of [94] by a minimum of 3.5% AUC-ROC score on both the datasets.

The reconstruction ability of the models on HAM10000 data set is shown in Figure 4.3 and on Daytime Driver Distraction set is show in Figure 4.4. From the figures, it can be observed that the kernel strengthening of models using ACB and ADB help in

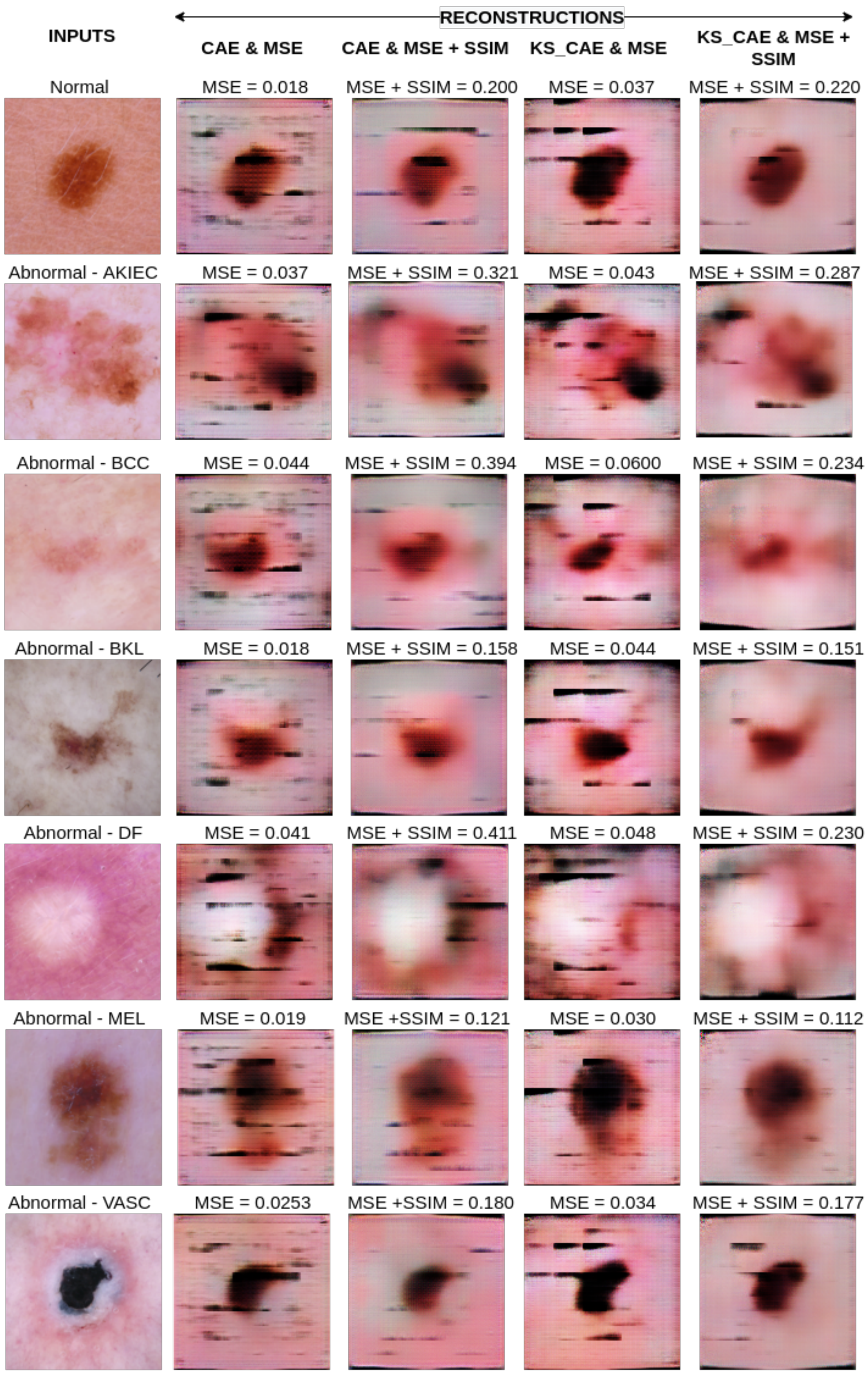


Figure 4.3: Reconstructions of models on HAM10000 set

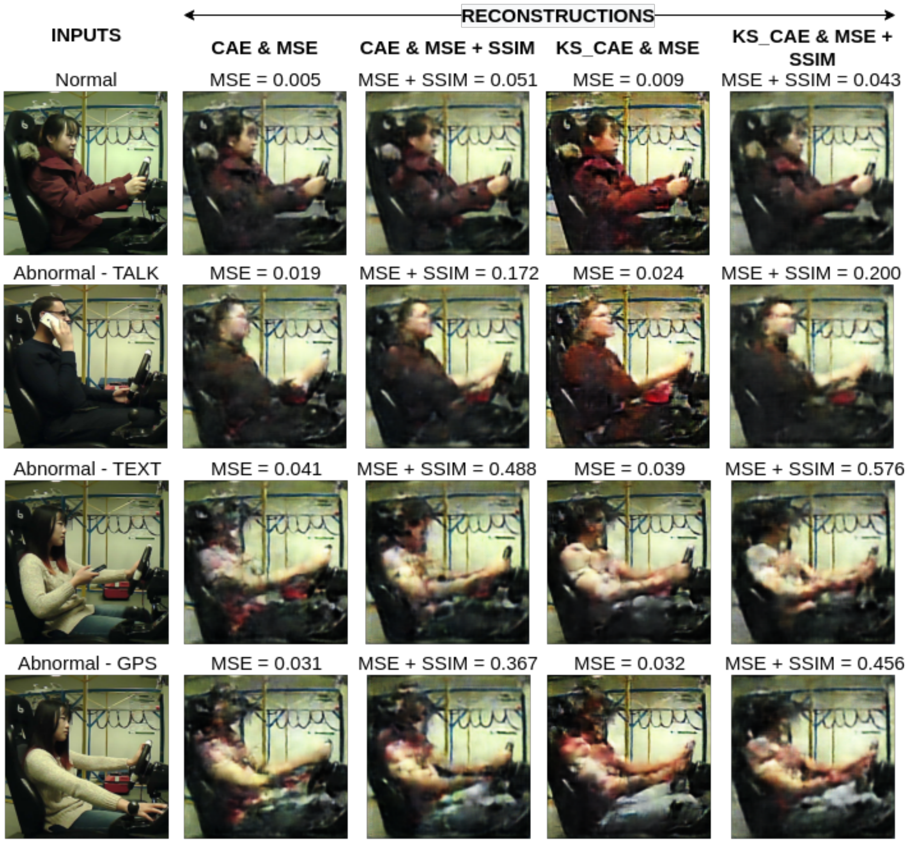


Figure 4.4: Reconstructions of models on Daytime Driver Distraction set

producing sharper images with similar shape, texture and visual features to that of the input images when compared to the BCAE2 models trained and it is visually observable from Figure 4.4 that the reconstructions are better with kernel strengthened models. Naturally, the models with strengthened kernels have loss values that are attenuated for normal samples and increased for abnormal samples in comparison to their baseline counterparts which is a desirable property that indicates the ability of the model to separate the anomalous samples from the normal ones. The above-mentioned phenomenon can also be observed from the first layer activations of the model as seen from Figure 4.5 where the kernel strengthened models trained on compound loss preserve the shape and texture of the normal samples while the texture is dissociated and smudged when it comes to abnormal samples. In terms of computational efficiency, since the disparately learnt kernels in kernel-strengthened models are combined into single units channel-wise after training, the computational performance of the model is the same as that of the BCAE2 thereby

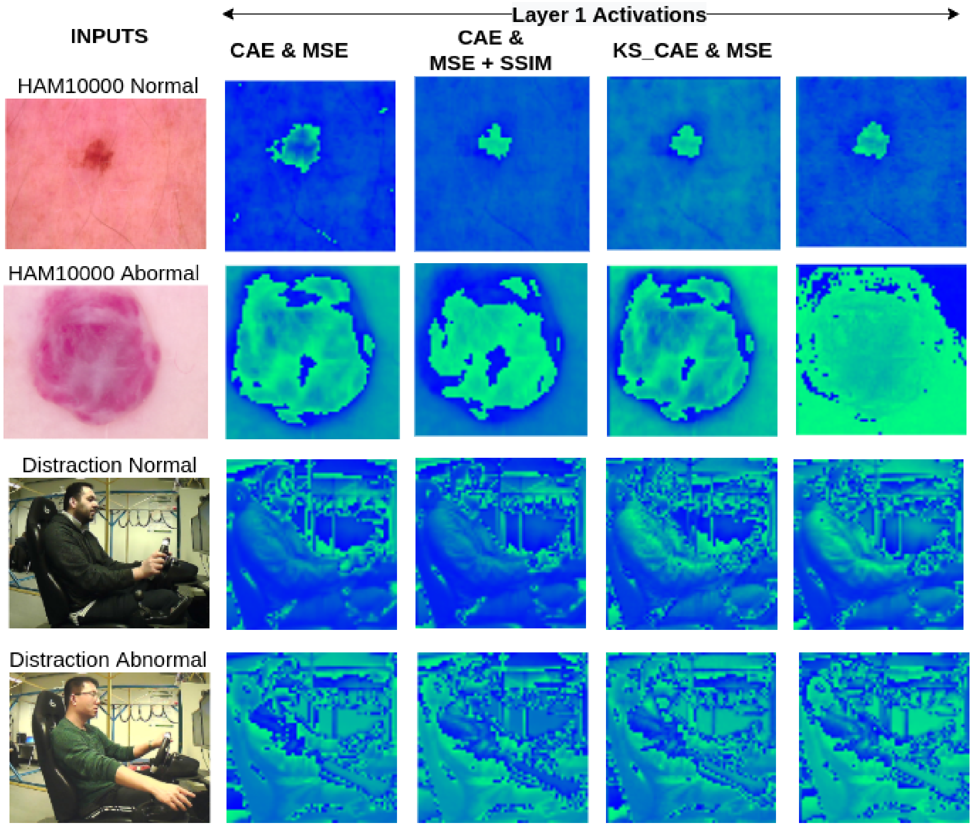


Figure 4.5: Examples of layer activations of the models on normal and abnormal data

rendering better performance at no extra computational cost.

4.6 Structural Similarity AutoEncoder

4.6.1 Experiments

The experiments from section 4.5 were extended with an addition of the proposed SSIM based compound loss function instead of MSE on BCAE2 and KS CAE architectures to show the universal nature of the proposed improvement and its model-agnostic nature. The compound loss function is directly used to score the anomalies.

Model	**Loss function**	**AUC-ROC Score**						
		AKIEC	BCC	BKL	DF	MEL	VASC	**MEAN**
CAE + Sparse coding [94]	MSE	0.78	0.74	0.70	0.65	0.79	0.66	0.72
Deep IF [49]	N/A	0.79	0.82	0.74	0.69	0.69	0.75	0.74
ResNet 154 [48]	N/A	N/A	N/A	N/A	N/A	0.62	N/A	N/A
CAE	MSE	0.82	0.70	0.72	0.61	0.81	0.66	0.72
	MSE + SSIM	0.83	0.72	0.71	0.63	0.82	0.72	0.74
KS-CAE	MSE	0.84	0.72	0.72	0.65	0.80	0.65	0.74
	MSE+SSIM [proposed]	**0.85**	**0.74**	**0.73**	**0.66**	**0.83**	**0.72**	**0.75**

Table 4.6: Performance on HAM10000 set

Model	**Loss function**	**AUC-ROC Score**			
		TALKING	TEXTING	USING GPS	**MEAN**
CAE + Sparse coding [94]	MSE	0.64	0.70	0.83	0.72
CAE	MSE	0.65	0.68	0.93	0.75
	MSE + SSIM	0.69	0.68	0.95	0.78
KS-CAE	MSE	0.66	0.71	0.93	0.77
	MSE+SSIM [proposed]	**0.66**	**0.70**	**0.97**	**0.78**

Table 4.7: Performance on Daytime Driver Distraction set

4.6.2 Results and Analysis

The improvement in performance on both BCAE2 and KS CAE can be seen in comparison to employing MSE from tables 4.6 and 4.7. The average improvement with respective the proposed compound loss function is 2% with a minor change. There is steady improvement across different types of anomalies in *HAM10000* and an improvement by 4% on *DDDS* is conspicuous from anomaly type *talking* where the person has the phone closer to the body and is hard to distinguish from the normal posture. From the figures 4.3 and 4.4, it can be observed that the usage of SSIM helps to preserve the intensity and colour of the input images to help maintain the overall structure when compared to the models trained with MSE. This is very critical as the pixel intensity directly affects the anomaly score. It is visually observable from Figure 4.4 that the reconstructions are better with the Kernel strengthened model trained on the compound loss function containing MSE and SSIM with normal parts of the image perfectly reconstructed and abnormal parts disfigured. The progression of improvement can be seen from left to right attributed to the usage of the compound loss function and in turn, augmented by kernel strengthening. Moreover, experimenting with different values of η can further improve the performance as η is specific to each dataset although we have fixed it for the purpose of experiments and consistency of results.

4.7 Attention-based AutoEncoders

In CNNs, the input propagation through multiple convolutional layers leads to learning by kernels at a local neighbourhood level and the spatial information learnt is abstracted into singular values in subsequent layers leading to the loss of a global context. Larger kernels and deeper networks can improve the abstraction process but only at a high computational cost. Hence, attention can come in handy by guiding CNNs to focus on sections of the inputs that are crucial in making decisions relating to the context in the input thereby improving the performance readily without much increase in computational complexity. Attention in computer vision operates by augmenting the important parts of the image while attenuating the other parts, to emphasize the relative importance of the essential input features over the others. In this section, few popular attention mechanisms along with the proposed soft-attention mechanisms are explored for the purpose of visual anomaly detection.

4.7.1 Experiments

The baseline convolutional AutoEncoder model BCAE1 will be used all the experiments henceforth. The proposed convolutional soft-attention model (SA CAE) with its improved version, the softmax convolutional soft-attention model (SSA CAE) from section 3.4.3 were evaluated on *HAM10000* for image anomaly detection and 5 video datasets - *UCSD Ped 1 & 2*, *Avenue* and *Subway Entrance & Exit* for frame-level video anomaly detection and compared with relevant work from literature to emphasize the efficacy of our approach over both the baseline architecture and the popular CAE based approaches. It was found that the increase in C_p more than 256 leads to saturation in performance and that $C_p = 64$ was most effective in achieving a balance between performance and computation. Though the value of λ depends on the nature of the dataset, it was found through experimentation that $\lambda = 1 \text{ x } 10^{-6}$ worked well for almost all the datasets. The kernels sizes were also chosen to be $k_1 = 3$ and $k_2 = 5$ for all the experiments though larger values of k_2 can aid in performance but simultaneously rise computations. λ should be chosen such that the model doesn't lose vital information due to heavy penalty leading to diminished weights. Other hyper-parameters optimization like the number of kernels, layers, kernel sizes were purposefully avoided to show that the performance improvements were due to the proposed approaches rather than optimal model design. Along with said models, AA CAE and SE CAE are also used for experiments to explore the general capability of attention mechanisms in CAEs for anomaly detection.

4.7.2 Results and Analysis

It is apparent from Table 4.8 showing the performance of all models, that our proposed SSA CAE consistently outperforms other models in all our experiments and that the margin of improvement in performance from that of the baseline model is significant for each dataset. Figure 4.6 shows the inputs and reconstructions of different models on a variety of datasets that are required for the analysis.

On *UCSD1* and *UCSD2* datasets, SSA CAE reconstructed anomalies like carts (white trucks) into two separate persons walking as it can be seen from Figure 4.6 and was able to detect cycles as anomalies producing deformed reconstructions whereas other models reconstructed carts by 80% and cycles almost perfectly. This shows the ability of SSA CAE model to distinguish anomalous object using the learning knowledge through focus, owing to the attention mechanism thereby making sure that unseen objects are not *perfectly* reconstructed. The overall improvements in AUC-ROC score were 3% and 5% on *UCSD1* and *UCSD2* respectively.

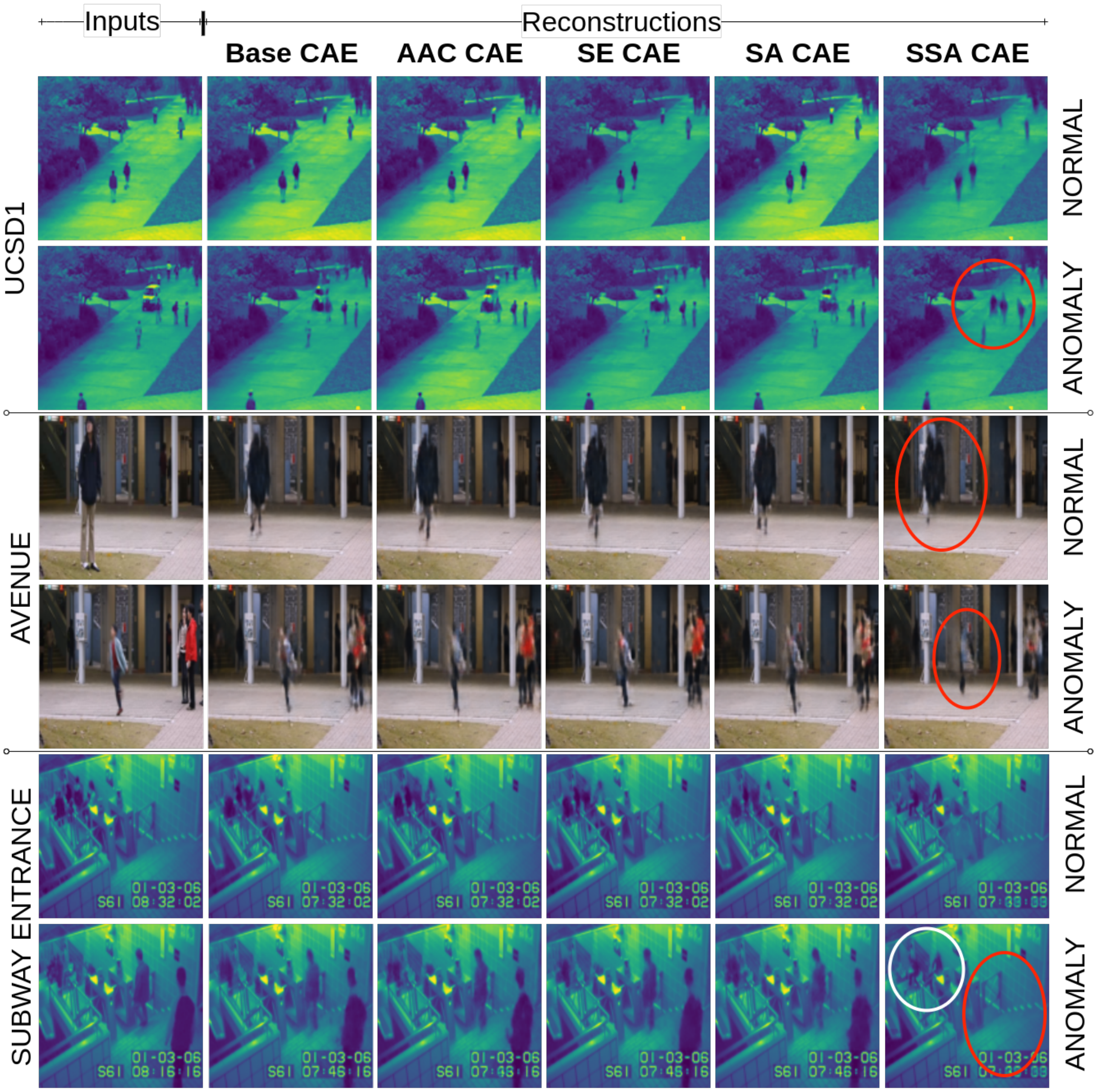


Figure 4.6: Comparing reconstructions of attention based models

Anomalies are *Cart, kid jumping, static person* in UCSD1, AVENUE and SUBWAY ENTRANCE respectively

On *Avenue* dataset, SSA CAE was able to identify and remove parts of humans when they appeared too close to the mounted stationary camera as it can be observed from Figure 4.6 and interestingly, SSA CAE identified static objects like a bag in the left-bottom corner of the frame by not constructing them while the baseline model failed at both the cases. This again showcases the ability of SSA CAE to handle unseen objects better resulting in the improvement of 8% AUC-ROC.

An intriguing phenomenon was observed with *Subway Entrance* and *Exit* datasets (Figure 4.6); SSA CAE model removed people close to the camera completely while preserving the people in the background and near the turnstile intact although there were a few instances of people in the foreground in the train set. The baseline model reconstructed the input almost perfectly. The difference in such results is that SSA CAE was able to learn normality well under context using the attention mechanism and was able to distinguish people close to the camera as an anomaly while understanding that people appearing in the background to be normal. The improvement in terms of AUC-ROC score was around 2%.

HAM10000 is inherently complicated and tangible reasoning without domain knowledge is hard and hence the performance metrics are solely relied upon for comparison as the visual analysis is difficult. The performance of the proposed model is compared with that of the other SOTA works such as [32] which uses CAE and [54] that used variational CAE with class-wise mean AUC-ROC, though their architectures are slightly more complex than ours. Though the reconstructions are slightly blurry, SSA CAE can identify anomalies better than the other models due to the reduction in the overall reconstruction capability of the CAE, alleviating the problem of partially or fully reconstructing anomalies and it is highly sensitive to the variations inputs and in a few cases, SSA CAE was even able to successfully detect people walking in the opposite direction which is an anomaly category in *UCSD* and *Subway* datasets though the temporal aspect was not taken into account while training.

4.8 Encoder Regularization in AutoEncoder

As established in the previous chapters, one of the methods to improve the anomaly detection performance of CAEs is to increase the discriminative ability between normal and anomalous embeddings so that it is reflected in reconstructions leading to ideal reconstructions for normal data and substantially poor reconstructions for abnormal data. The solution proposed using SVDD in section 3.5 is evaluated in this section. SVDD is an

Dataset	Model	AUC-ROC % ↑	EER % ↓	Precision % ↑	Recall % ↑	F1-Score % ↑
AVENUE	BCAE1	81.60	25.97	75.28	74.38	74.47
	AA CAE	78.08	27.84	73.31	72.44	72.54
	SE CAE	80.52	24.86	76.50	75.80	75.89
	SA CAE	79.75	25.93	76.03	74.55	74.61
	SSA CAE	89.67	15.76	85.44	85.46	85.43
	Conv-AE [32]	70.2	25.1	N/A	N/A	N/A
	Conv3D AE [44]	86.0	N/A	N/A	N/A	N/A
	ConvLSTM AE [44]	84.0	N/A	N/A	N/A	N/A
	Hybrid AE [62]	82.8	N/A	N/A	N/A	N/A
	U-Net AE [61]	86.9	N/A	N/A	N/A	N/A
SUBWAY ENTRANCE	BCAE1	74.28	33.38	96.38	61.15	73.47
	AA CAE	74.61	33.28	96.28	62.65	74.65
	SE CAE	72.08	32.13	96.21	66.63	77.62
	SA CAE	73.85	33.10	96.18	65.70	76.95
	SSA CAE	74.88	31.41	96.53	64.46	75.99
	Conv-AE [32]	94.3	26.0	N/A	N/A	N/A
SUBWAY EXIT	BCAE1	95.68	11.11	99.91	92.98	96.28
	AA CAE	95.72	11.11	99.91	92.59	96.08
	SE CAE	95.53	11.11	99.91	93.12	96.36
	SA CAE	95.60	11.11	99.91	93.03	96.31
	SSA CAE	97.75	5.92	99.92	94.08	96.87
	Conv-AE [32]	80.7	9.9	N/A	N/A	N/A
UCSD1	BCAE1	68.32	37.59	67.20	67.22	66.27
	AA CAE	66.06	38.54	68.16	67.10	65.06
	SE CAE	66.49	38.45	67.02	67.08	66.17
	SA CAE	67.54	38.00	66.31	66.29	65.10
	SSA CAE	71.09	33.28	67.97	68.11	68.00
	Conv-AE [32]	81.0	27.9	N/A	N/A	N/A
	Conv3D AE [44]	70.0	N/A	N/A	N/A	N/A
	ConvLSTM AE [44]	67.0	N/A	N/A	N/A	N/A
UCSD2	BCAE1	83.85	26.52	84.95	66.92	70.74
	AA CAE	83.28	26.38	83.96	70.15	73.54
	SE CAE	82.71	27.18	83.87	69.15	72.69
	SA CAE	86.89	19.36	86.56	80.35	82.11
	SSA CAE	88.06	18.78	86.96	79.85	81.76
	Conv-AE [32]	90.0	21.7	N/A	N/A	N/A
	Conv3D AE [44]	64.0	N/A	N/A	N/A	N/A
	ConvLSTM AE [44]	77.0	N/A	N/A	N/A	N/A
	Hybrid AE [62]	84.3	N/A	N/A	N/A	N/A
	U-Net AE [61]	96.2	N/A	N/A	N/A	N/A
HAM10000	BCAE1	68.60	35.50	72.03	68.85	70.01
	AA CAE	68.78	35.19	72.60	70.94	71.63
	SE CAE	69.36	34.74	72.53	69.19	70.37
	SA CAE	69.10	35.08	72.58	70.50	71.32
	SSA CAE	70.15 [**76.69**]	34.45	73.30	69.56	70.82
	Variational AE [54]	Mean 77.9	N/A	N/A	N/A	N/A

Table 4.8: Evaluation results of the attention based models including the proposed ones

established technique for feature separation in the multi-dimensional space and the proposed approach is to incorporate it as an encoder regularizer for CAEs that can be used as a fine-tuning method after normal training. The intuition behind the proposed approach can be explained on the basis of the vital role played by embeddings of AEs towards reconstructions and in turn scoring anomalies. It is imperative that imparting desirable properties for anomaly detection such as aggregation of normal embeddings together and isolation of anomalous embeddings from them can help to augment the performance and the proposed method aids in the same. The SVDD objective in equation 3.13 is used for fine-tuning of the encoder of CAE on normal data as inputs and normal embeddings as outputs ($Z_{normal} = f_e(X_{normal})$). The hypothesis is that SVDD fine-tuning could help the CAE learn the notion of normality better and in concentrating the normal embeddings together in the multi-dimensional latent space as a cluster and any unseen anomalous input through the CAE will produce embeddings that do not conform to the cluster, thus making the model more descriptive in distinguishing between normal and anomalous data.

4.8.1 Experiments

As discussed in the previous chapter, only the parameters of the encoder (θ_e) are taken into consideration for fine-tuning. Moreover, since this is a minor transformation in the manifold, only the last layer of the encoder weights are changed in reality. The centre of the hyper-sphere in equation 3.14 can be initialized as the mean of all the embeddings of the training set before fine-tuning. It is important to emphasize that the fine-tuning has to take place at a steady pace on a low learning rate for a few epochs as high learning might disrupt the learnt knowledge of the CAE paving way to adverse effects in the reconstruction ability. The model for the experiments is a copy of the trained BCAE1 and is referred to as *BCAE1 + SVDD*.

4.8.2 Results and analysis

This section presents the results of experiments pertaining to SVDD encoder regularization to BCAE1 on 2 image - *HAM10000*, *MVTec* and 3 video datasets - *UCSD*, *Subway* and *Avenue*. The evaluation results on the datasets are provided in 4.9 along with the visualization containing the reconstructions from different models in Figure 4.7. The overall performance improvement of the models due to encoder regularization using SVDD is significant and consistent on almost all the datasets exhibiting the effectiveness of the proposed solution. Moreover, the performance of BCAE1 + SVDD is comparable with

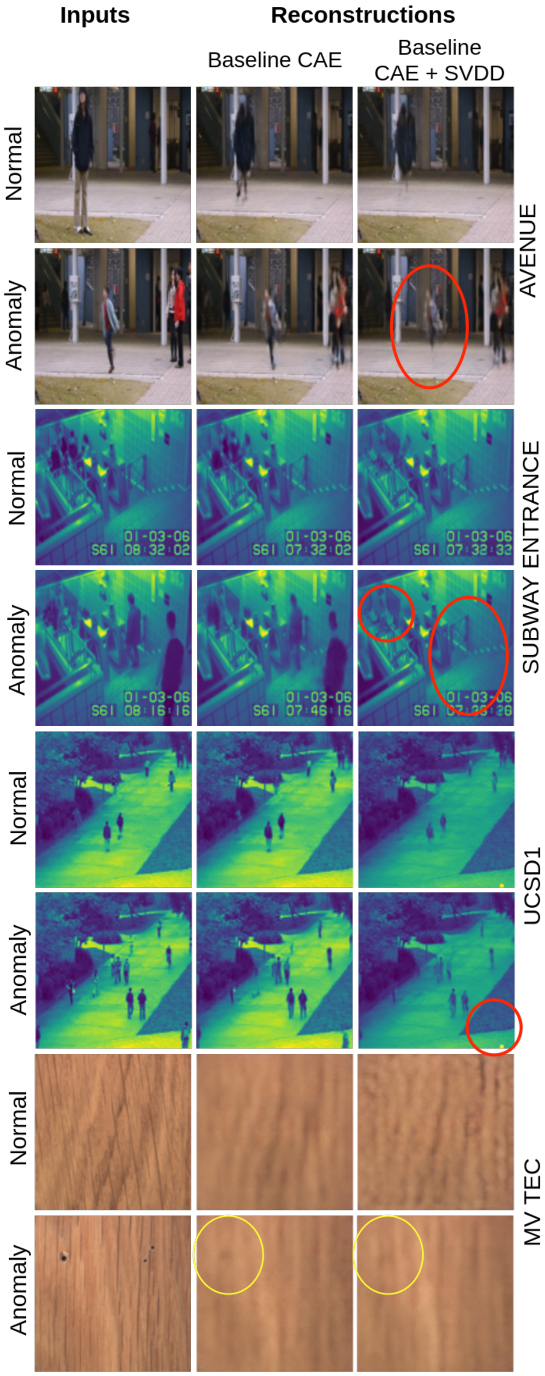


Figure 4.7: Comparison of effect of SVDD fine-tuning on reconstructions

that of several other works despite the compact nature of the architecture of the baseline model under consideration.

In particular, few instances showed the capability of the BCAE1 + SVDD to better separate anomalies. On *UCSD1* and *UCSD2*, BCAE1 + SVDD deliberately avoided reconstruction of people on grass whereas BCAE1 did although it is an anomaly and such instances lead to a better performance by 1.6% and 5% AUC-ROC on *UCSD1* and *UCSD2* respectively, surpassing that of the performance of complex architectures such as ConvLSTM and Conv3D AEs. On Avenue, it was seen that BCAE1 + SVDD partially erased anomalies more while leaving normal parts of the image intact as can be seen from Figure 4.7 and the performance was better than BCAE1 by 3%, outperforming all the methods except Conv3D AE which could be due to its ability to learn spatio-temporal features comprehensively. Similarly, on Subway *Entrance* and *Exit* datasets, there were noticeable changes similar to SSA CAE of erasing people too close to the camera and the improvements are 5% and 3% respectively. Consistent performance improvements were also seen on the image datasets - *HAM10000* and *MVTec*. All these above instances reflect the better separability between normal and anomalous data due to SVDD regularization that helps in concentrating normal embeddings together. This proposed mechanism can also be potentially employed as a *self-correction mechanism* in applications that involve online learning.

Although the aim of this work is not to create SOTA models and to show the effectiveness of the proposed solution on simple CAE architectures, the performance of BCAE1 + SVDD is compared with complex SOTA methods from other works that employ 2D CAEs (separated by horizontal lines in Table 4.9). It is worth mentioning that the selection of such simple architecture was to show the effectiveness of the proposed solution and which has a maximum of 96 kernels of size 3×3 at any layer is being compared with other works that typically use large kernels of size 5×5, 11×11 and number of filters in the range of 128 to 1024. [32] employs a fully-connected AE with HOG and HOF hand-crafted features as inputs along with a stacked CAE with convolutional and pooling layers taking 10 frames of size 227×227 and the number of kernels are between 128 and 512 with embedding size of $128x13 \times 13$. It is clearly an unfair comparison yet the proposed models outperform [32] on *Avenue*, *Subway Exit* and with comparable performance on *UCSD2* despite [32] using persistence-1D algorithm over a 50 frame window to smoothen regularity to boost performance. Similarly, [44] evaluate the model on *UCSD* and *Avenue* datasets, analyse performance of ConvLSTM and 3D CAE which are inherently complex architectures that take temporal aspects into account while learning. Little information is provided about the specifications of the architectures except for input size being 128×128. The BCAE1 + SVDD outperformed both the models on all the datasets. Likewise, [62] proposes a hy-

Dataset	Model	AUC-ROC (in %) ↑	EER (in %) ↓	Precision (in %) ↑	Recall (in %) ↑	F1-Score (in %) ↑
AVENUE	BCAE1	81.60	25.97	75.28	74.38	74.47
	BCAE1 + SVDD	84.56	23.54	77.75	76.91	76.99
	Conv-AE [32]	70.2	25.1	N/A	N/A	N/A
	Conv3D AE [44]	86.0	N/A	N/A	N/A	N/A
	ConvLSTM AE [44]	84.0	N/A	N/A	N/A	N/A
	Hybrid AE [62]	82.8	N/A	N/A	N/A	N/A
SUBWAY ENTRANCE	BCAE1	74.28	33.38	96.38	61.15	73.47
	BCAE1 + SVDD	79.03	26.20	96.65	71.55	81.08
	Conv-AE [32]	94.3	26.0	N/A	N/A	N/A
SUBWAY EXIT	BCAE1	95.68	11.11	99.91	92.98	96.28
	BCAE1 + SVDD	97.45	10.87	99.92	89.14	94.17
	Conv-AE [32]	80.7	9.9	N/A	N/A	N/A
UCSD1	BCAE1	68.32	37.59	67.20	67.22	66.27
	BCAE1 + SVDD	69.92	35.5	67.66	67.32	65.95
	Conv-AE [32]	81.0	27.9	N/A	N/A	N/A
	Conv3D AE [44]	70.0	N/A	N/A	N/A	N/A
	ConvLSTM AE [44]	67.0	N/A	N/A	N/A	N/A
UCSD2	BCAE1	83.85	26.52	84.95	66.92	70.74
	BCAE1 + SVDD	88.85	19.42	86.87	77.91	80.19
	Conv-AE [32]	90.0	21.7	N/A	N/A	N/A
	Conv3D AE [44]	64.0	N/A	N/A	N/A	N/A
	ConvLSTM AE [44]	77.0	N/A	N/A	N/A	N/A
	Hybrid AE [62]	84.3	N/A	N/A	N/A	N/A
HAM10000	BCAE1	68.60	35.50	72.03	68.85	70.01
	BCAE1 + SVDD	68.79 (**76.82***)	35.51	72.16	70.67	71.3
	Variational AE* [54]	77.9	N/A	N/A	N/A	N/A
MVTec	BCAE1	61.34	42.83	69.13	69.80	69.44
	BCAE1 + SVDD	66.36	38.33	70.25	67.58	68.62

Table 4.9: Results of SVDD encoder regularization in comparison to other models.
(* represents class-wise mean)

brid CAE combining supervised and unsupervised paradigms of learning from input video cuboids of size 10 × 10 × 3 to learn the spatio-temporal information jointly with an adversarial discriminator and the results are reported on *UCSD2* and *Avenue* datasets while BCAE1 + SVDD out-bests the hybrid CAE on both datasets. Finally, for *HAM10000* [54] uses a variational CAE and very less information regarding the exact architecture is shared apart from the fact that the network is similar to *DCGAN* with 5 convolutional and 5 deconvolutional layers and the mean of class-wise AUC-ROC scores are reported instead of the overall value and BCAE1 + SVDD performance is comparable to theirs.

Ablation studies

It is critical to perform ablation studies to learn the influence of parameters that could potentially impact the overall performance. Fine-tuning BCAE1 for encoder regularization using SVDD involves only the encoder $f_e(x)$ and the process aids in shifting the normal embeddings and concentrating them together into a cluster in a multi-dimensional space as it happens at a low learning rate for a few epochs. The effect of *learning rate* and the number of *epochs* of fine-tuning is shown in Figure 4.8 and the significant performance gain with the right choice of them is discernible. The black dot in each plot represents the performance of the original BCAE1 models before fine-tuning (encoder regularization). From the figure, typical values around 5×10^{-7} for learning rate and 10 epochs consistently provide substantial performance enhancements. It is very important to set the initial value of the centre equal to the mean of normal embeddings in the train set instead of zeros as setting to zeros will disrupt the learnt knowledge by diminishing the weights to lower values near zero.

Visual analysis of embeddings separation

A prominent yet obvious change that was observed in encoder activations was that many dormant kernels produced more informative, high-contrast activations on encoder regularization. To show the effect of encoder regularization on embeddings and its impact on separability, the embeddings of BCAE1 and BCAE1 + SVDD are visualized in tandem for comparison in a low dimensional space on both normal and anomalous data. The BCAE1 + SVDD models picked are the best ones from the ablation studies. The visualization is performed on *UCSD2* and *Avenue* datasets in which BCAE1 + SVDD achieved significant performance increase. PCA is used instead of TSNE for consistency and reproducibility of results. The First 3 test videos of UCSD2 and Avenue with mostly anomalous samples and few normal samples are used for visualization as can be seen in Figures 4.9 and

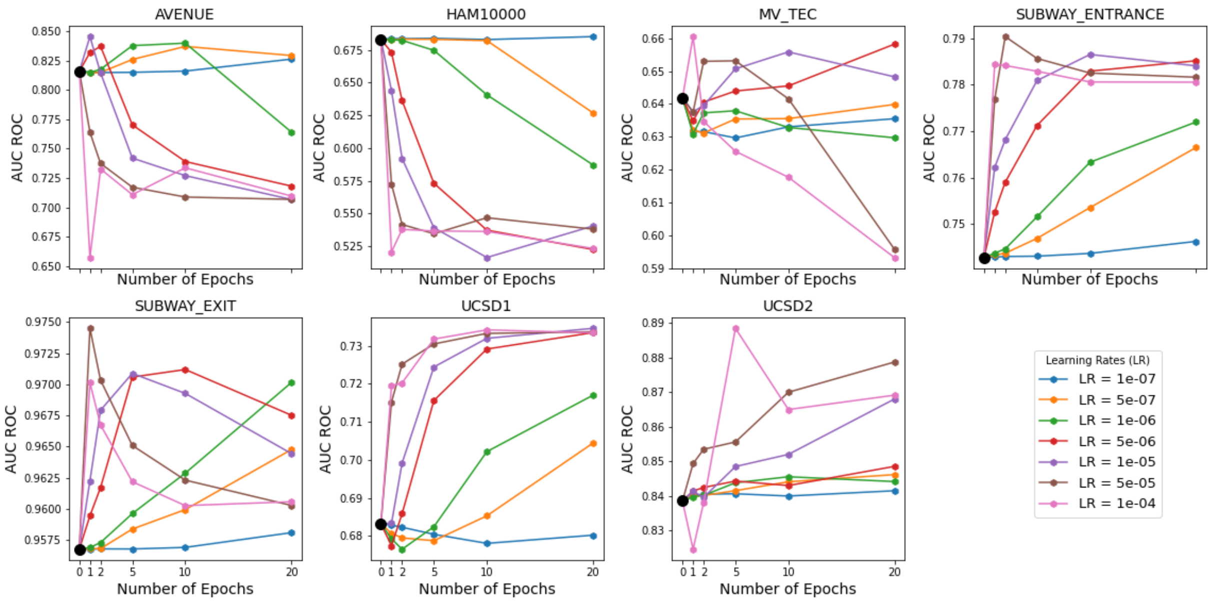


Figure 4.8: AUC-ROC scores with respect to learning rates and number of epochs of fine-tuning with SVDD encoder regularization (Black dot represent the performance of baseline model)

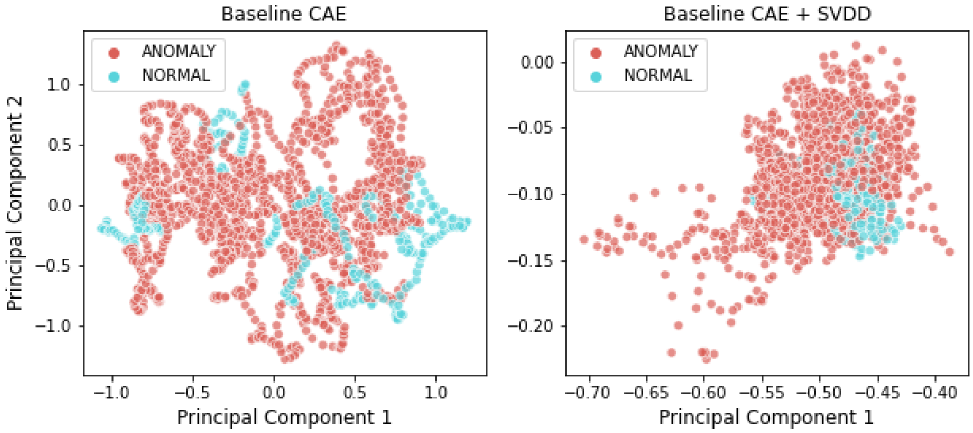


Figure 4.9: Effect of SVDD regularization on encodings on UCSD2 dataset

4.10. The improvement in separation between normal and anomalous embeddings can be well observed showing that encoder regularization using SVDD not only separates concentrates normal embeddings together but also isolates anomalies successfully in the latent

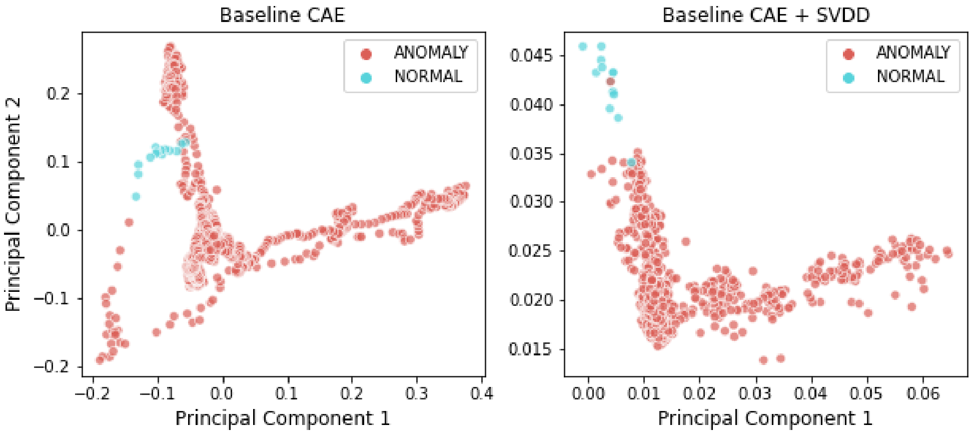


Figure 4.10: Effect of SVDD regularization on encodings on Avenue dataset

embedding space.

4.9 Convolutional Recurrent Architectures

Temporal correlation and learning are vital in understanding the context of events in videos. For example, consider an event of a car driving down a highway which separates a farm. The separate events of the car staying on the highway and car parked on the farm are considered to be normal. A good anomaly detection model should distinguish the aforementioned events from a mishap of the car losing control and driven off the highway into the farm which is anomalous event. Such patterns can only be understood and learnt by considering the temporal aspect in videos which construct the events, motion patterns and behaviour of objects and spatial models [2] will not be able to identify the behavioural pattern and change in motion as they operate frame-wise. There are several potential uses cases and applications in which motion of the objects could play a crucial role like surveillance, security, autonomous driving etc. The models that could handle spatio-temporal learning are of highly complex nature and require huge computing resources to function. Hence, it is important to explore and analyse the different possible architectures to choose the efficient one with the correct balance between performance and computational requirement according to the application at hand. The main purposes of the experiments are summarized as follows:

[2] 2D convolutional models without temporal learning

1. Explore different variants of convolutional recurrent neural networks for the task of spatio-temporal learning and correlation

2. Assessing the need for complex architectures such as convolutional recurrent neural networks and the performance difference achieved over the 2D convolutional AutoEncoders discussed in the previous sections. Evaluate the trade-off between computational complexity and performance improvement.

3. Understand the working of different types of architectures such as the convolutional recurrent AutoEncoders, BiDirectional recurrent convolutional AutoEncoders and the Seq2Seq frame prediction models

4. Selecting the best convolutional recurrent configuration for the task of video anomaly detection based on the trade-off between performance and computational complexity.

4.9.1 Experiments

To explore the efficacy of a variety of convolutional architectures on video anomaly detection that were discussed in the last chapter, we experiment with change in two important parameters - number of recurrent layer used L_R in the network and type of layers used for reconstruction or prediction of frames DUT where $DUT \epsilon$ {Recurrent transpose convolutional layers, Time distributed 2D transpose convolutional layers}. The former is to study the effectiveness and role of recurrent convolutional layers in learning from the early layers of the network before severe abstraction and the latter to understand the performance improvement owing to convolutional recurrent layers over spatial convolutional layers in learning for reconstructing or predicting motion in videos from the latent embeddings. For all the architectural variants, the learning in the model is conditioned in such a way that the important information is contained out of the ultimate encoder layer which is vital for the reconstruction of existing or predicting the future frames.

The experiments are conducted on 9 different architectures - *ConvRNN AutoEncoder (CRNN AE), ConvLSTM AutoEncoder (CLSTM AE), ConvGRU AutoEncoder (CGRU AE), BiDirectional ConvRNN AutoEncoder (BiCRNN AE), BiDirectional ConvLSTM AutoEncoder (BiCLSTM AE), BiDirectional ConvGRU AutoEncoder (BiCGRU AE), Seq2Seq ConvRNN network (Seq2Seq CRNN NN), Seq2Seq ConvLSTM network (Seq2Seq CLSTM NN), Seq2Seq ConvGRU network (Seq2Seq CGRU NN)* with variation in two important parameters - L_R and DUT on 5 different video datasets that contain spatial and temporal anomalies. The inputs frames are resized to 128 × 128 and are arranged as tensors of shape $T \times W \times H \times C$. The normal frames are labelled as 1 and anomalies as 0. The models are trained only on normal data for 300 epochs using MSE as objective function and Adam optimizer with a starting learning rate of 1×10^{-03} equipped with learning rate decay and early stopping with a batch size of 32 in a computing cluster. The dataset is augmented with varying strides of frames such as $1, 2, 4, 8, 16$ prior to training and the test set is retained as such without any change. The error/loss are calculated between every pair of input and predicted frames and are used for performance evaluation. For each of the models, the frame-wise losses are calculated using MSE and temporally aggregated. The aggregated loss $e(t)$ at time t are used to calculate the regularity $s(t)$ which denotes the probability of a frame being normal [3]. The temporal regularity $s(t)$ is calculated using the equation (4.1) where $I(x, y, t)$ is the pixel intensity at position x, y at time step t. *Sav-Gol filter* is applied on the regularity for a window of 15 frames instead of the Persistence 1D [45] algorithm on a window of 50 frames used by many works. This process helps to smooth local minima or maxima.

$$
\begin{aligned}
e(t) &= ||I_{(x,y,t)} - f_d(f_e(I_{(x,y,t)}))||_2 \\
s(t) &= 1 - [\frac{(e(t) - min_{e(t)})}{(max_{e(t)} - min_{e(t)})}]
\end{aligned}
\tag{4.1}
$$

4.9.2 Results and Analysis

The experimental results[5] on 9 architectures are tabulated in Table 4.13 with the findings and results of the research under various factors and contexts discussed in this section.

General comparison with 2D convolutional models and models from other works

To show the better performance of convolutional recurrent architecture over 2D convolutional AutoEncoders due to temporal learning, the results are compared on video datasets. The baseline model BCAE1 has the same structure as the convolutional recurrent models except the type of layers are all spatial layers that operate on individual frames of the video. The performance increment due to the consideration of temporal correlation from data is observable from table 4.11 containing the comparison of results. Additionally, the table also shows the results of other works [6] that employ similar architectures to the ones

[3] Regularity is 1.0 for a perfectly normal frame and lower for anomalous frame

[5] A comprehensive tabulation of evaluation metrics and extra results of test samples are available in the supplementary material pertaining to Paper [2] in the statement of contribution

[6] separated by horizontal lines in the table

Parameter	Description	Effect studied?
Image size $W \times H$	The performance of models increase with increase in resolution and saturate after a maximum resolution $W_{max} \times H_{max}$. Directly proportional to computational requirement	Fixed to 128×128
Channels C	Grayscale images have 1 channel and colored images have 3 channels	Depends on the input
Time steps T	Time steps or number of frames per unrolling / video clip, as rightly discussed by [32] has little to no effect on performance	Fixed to 8
Number of layers L	Total number of layers in encoder or decoder signifies how deep an architecture is	Fixed to 5
Number of recurrent layers L_R	The remaining $L - L_R$ layers are time distributed 2D convolutional layers	Varied from 1 to 3
Decoder Upsampling type DUT	Decoder Upsampling type represents the nature of the decoder whether recurrent transpose convolutional layers are used or only time distributed 2D transpose convolutions are used instead	Both variants studied [4]

Table 4.10: Configurable hyper-parameters in convolutional recurrent models

that are considered in this study including more complex and heavier models operating at a higher input resolution of 224 than ones in this study that use 128. It can be seen that the proposed convolutional recurrent variants outperform the baseline 2D BCAE1 models on all the datasets and few other convolutional models from other works on most datasets despite the smaller architectural size.

Performance variation due to L_R and DUT

The importance of using recurrent layer instead of time distributed spatial layers can be observed by varying L_R. This experiment can also shed some light on the extent to which number of layers can help in performance improvement and in achieving the optimal number of recurrent layers for temporal learning. From table 4.13, it can be observed that there is steady performance increase with respect to increase in L_R in architecture with

Dataset	Model	AUC-ROC	EER	Precision	Recall	F1-Score
Avenue	2D CAE	0.816	0.259	0.752	0.743	0.744
	Hand Crafted + spatio-temporal CAE [32]	0.702	0.251	N/A	N/A	N/A
	ConvLSTM [44]	0.840	N/A	N/A	N/A	N/A
	Predictive ConvLSTM 224 [58]	N/A	N/A	0.952	1.00	N/A
	CGRU AE (1, Y) [***ours***]	0.853	0.249	0.804	0.796	0.792
UCSD Ped 1	2D CAE	0.683	0.375	0.672	0.672	0.662
	Hand Crafted + spatio-temporal CAE [32]	0.810	0.279	N/A	N/A	N/A
	ConvLSTM [44]	0.670	N/A	N/A	N/A	N/A
	Predictive ConvLSTM 224 [58]	N/A	N/A	0.864	0.923	N/A
	Seq2Seq CLSTM NN (3,N) [***ours***]	0.737	0.310	0.696	0.694	0.694
UCSD Ped 2	2D CAE	0.838	0.265	0.849	0.669	0.7074
	Hand Crafted + spatio-temporal CAE [32]	0.900	0.217	N/A	N/A	N/A
	ConvLSTM [44]	0.770	N/A	N/A	N/A	N/A
	Predictive ConvLSTM 224 [58]	N/A	N/A	0.923	1.00	N/A
	Seq2Seq CGRU NN (2,N) [***ours***]	0.862	0.192	0.875	0.790	0.813
Subway Entrance	2D CAE	0.7428	0.333	0.963	0.622	0.734
	Hand Crafted + spatio-temporal CAE [32]	0.940	0.260	N/A	N/A	N/A
	Predictive ConvLSTM 224 [58]	N/A	N/A	0.816	0.0.939	N/A
	Seq2Seq CGRU NN (3,N) [***ours***]	0.854	0.226	0.973	0.701	0.800
Subway Exit	2D CAE	0.954	0.111	0.991	0.928	0.962
	Hand Crafted + spatio-temporal CAE [32]	0.807	0.099	N/A	N/A	N/A
	Predictive ConvLSTM 224 [58]	N/A	N/A	0.659	0.967	N/A
	Seq2Seq CLSTM NN (2,N) [***ours***]	0.978	0.060	0.999	0.940	0.969

Table 4.11: Comparing convolutional recurrent models with 2D convolutional AutoEncoders and models from other works

a mix of recurrent and time distributed spatial layers in encoder and decoder i.e. when $DUT = N$. This shows the effectiveness of using *convolutional and transpose convolutional recurrent cells* to abstract and upsample data with embedded motion patterns in them. But when the decoder contains only time distributed recurrent layers $DUT = Y$, the performance increases till $L_R = 2$ and then saturates (at times even reduces) for $L_R > 2$. This also shows that using a mix of recurrent and time distributed spatial layers can help in augmenting the performance and hence performance of architectures with $DUT = N$ are consistently better denoting the ability of transpose convolutional recurrent layers (or cells) towards reconstructions or predictions.

Comparison of the convolutional recurrent cells - CRNN vs CLSTM vs CGRU

The CRNN variants of models constantly exhibit patterns of severely memorizing the background on *Avenue* and *UCSD1* datasets when $L_R > 1$. The effect can be attributed to the simple internal mechanism of ConvRNN cells that are devoid of the ability to learn motion patterns as well as its counterparts, over consecutive layers resulting in the absence of learnt notion of normality from input video clips. CGRU model variants consistently perform best on all the datasets although their performance is slightly sub-par on *Avenue* in comparison to CLSTM variants and this scenario is interesting since Avenue is the only dataset with coloured input frames among the lot. The performance of CRNN architectures with $L_R = 1$ is considerably good, immaterial of the type of the upsampling layer (DUT).

Comparison of the architectural variant - Normal vs BiDirectional vs Seq2Seq

Although the BiDirectional recurrent layers have shown tremendous performance improvements on language tasks, the performance of BiDirectional model variants on video anomaly detection is sub-par, accompanying a large number of parameters without definite performance improvements. The reconstructions and recall of the BiDirectional variants are better compared to other models confirming the initial hypotheses that BiDirectional convolutional recurrent layers have better learning capabilities owing to the ability to learn from the past and future sequences. But this characteristic may not be constructive for video anomaly detection since the models were able to almost perfectly reconstruct even the anomalous sequences. On the other hand, the Seq2Seq variants perform the best and this can be attributed to the mechanism of learning where the models are conditioned to predict the future set of normal frame from a more intensively abstracted compact representation. This nature is also capitalized for anomaly detection since even a little deviation in normality in the seed frames results in drastic variation between the actual rest of the frames and predicted frames thereby yielding a higher loss and anomaly scores. The normal variants perform consistently well and considerably better in comparison to the baseline spatial BCAE1 models. This improvement in anomaly detection performance demonstrates the efficacy of using convolutional recurrent layers for representational learning of motion patterns in videos.

Trade-off between performance and computational complexity

It is important to factor in computation complexity as models in many real-time applications run on devices without huge computing resources like GPU or TPU and it is crucial that the models are efficient to achieve the desired outcome. The number of trainable parameters in each architectural configuration for coloured inputs are tabulated in table 4.12. The use of Seq2Seq models accompany an increase in parameters by 16% on an average with a significant boost in the overall performance. This can be attributed to the paradigm of learning from predicting future frames from the past and current frames as opposed to mere compression and reconstruction. The right configuration of Seq2Seq model with little effort on hyper-parameter tuning can provide significant boost in performance without much increment in computation. Considering the overall performance of different models, ***Seq2Seq GRU*** models have the best trade-off between anomaly detection performance and computational efficiency and hence are the *overall best performing architecture* with clear results on almost all the video datasets. Moreover, ConvGRU models consistently perform better than ConvLSTM models with lower computation requirement and should be the natural choice for video-related tasks which is in complete contradiction to what is seen in the literature where ConvLSTM models are predominantly used for video-related tasks. Finally, based on the performance metrics on the evaluated datasets from table 4.13 and in comparison to other models, *ConvGRU cell* is the most effective learning configuration.

Visual analysis of reconstructions and predictions

The reconstructions of AutoEncoder models and predictions of Seq2Seq models are analysed visually in this section to compare the quality of output which is representative of the anomaly detection performance. For prediction of frames in Seq2Seq models, the number of seed input frames and predicted frames during training and testing were both set to 4. But for testing purposes, for seed frames of time steps T_{t-4} to T_{t-1}, *eight* frames from T_t to T_{t+7} are analysed and presented in this section. Naturally, it is important to note that the predictions after 6^{th} frame are of poor quality with contents blurred and deformed as seen from figure 4.12 although the recurrent models are expected to perform better for a longer periods of unrolling. As discussed in the earlier sections, for $L_R > 1$, CRNN model variants seem to memorize the background without any useful reconstructions for both normal and abnormal data as seen in figure 4.11. The outputs from CLSTM AE, CGRU AE are slightly better with an increasing value of L_R. As hypothesised, BiDirectional variants exhibited reconstructions with better motion-patterns but with the ability to reconstruct even the anomalous objects in the frame as seen in figure 4.13, showing better ability in learning

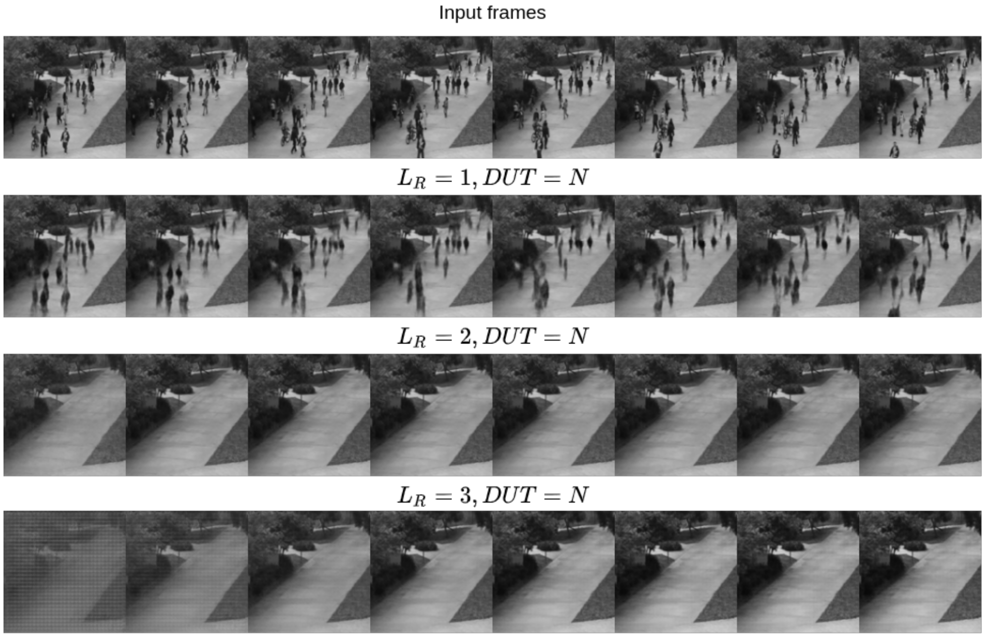


Figure 4.11: CRNN variants learning the background with increasing L_R

videos although this might not be suitable for anomaly detection. Moreover, the outputs of all the model variants are better when recurrent transpose convolutional layers are used in the decoder for upsampling instead of time distributed 2D transpose convolutions.

Table 4.13: Performance comparison of models on different datasets (variants represented by (L_R, DUT))

Model	Avenue		Subway Entrance		Subway Exit		UCSD1		UCSD2	
	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER
CRNN(1,N)	0.83	0.24	0.78	0.31	0.96	0.11	0.69	0.35	0.82	0.24
CRNN(1,Y)	0.83	0.25	0.76	0.33	0.96	0.11	0.69	0.36	0.83	0.26
CRNN(2,N)	0.71	0.35	0.79	0.27	0.96	0.11	0.73	0.32	0.83	0.22
CRNN(2,Y)	0.83	0.23	0.76	0.31	0.96	0.11	0.68	0.36	0.79	0.28
CRNN(3,N)	0.68	0.36	0.78	0.28	0.96	0.11	0.67	0.36	0.70	0.31
CRNN(3,Y)	0.71	0.35	0.79	0.27	0.96	0.11	0.73	0.33	0.85	0.23

Table continued on the next page

Table 4.13 – *continued from the previous page*

Model	Avenue		Subway Entrance		Subway Exit		UCSD1		UCSD2	
	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER
CLSTM(1,N)	0.82	0.23	0.74	0.34	0.96	0.11	0.68	0.36	0.77	0.27
CLSTM(1,Y)	0.78	0.27	0.74	0.34	0.96	0.11	0.65	0.39	0.86	0.22
CLSTM(2,N)	0.83	0.24	0.68	0.37	0.95	0.11	0.70	0.35	0.85	0.22
CLSTM(2,Y)	0.78	0.26	0.74	0.32	0.96	0.11	0.68	0.35	0.80	0.30
CLSTM(3,N)	0.82	0.25	0.75	0.34	0.96	0.11	0.69	0.34	0.84	0.23
CLSTM(3,Y)	0.83	0.25	0.71	0.35	0.95	0.11	0.69	0.35	0.81	0.28
CGRU(1,N)	0.80	0.24	0.74	0.33	0.95	0.11	0.67	0.36	0.81	0.25
CGRU(1,Y)	0.85	0.25	0.75	0.33	0.96	0.11	0.67	0.37	0.86	0.21
CGRU(2,N)	0.78	0.25	0.69	0.38	0.96	0.11	0.70	0.35	0.86	0.22
CGRU(2,Y)	0.78	0.27	0.71	0.34	0.96	0.11	0.67	0.36	0.83	0.23
CGRU(3,N)	0.82	0.27	0.75	0.32	0.96	0.11	0.69	0.35	0.85	0.23
CGRU(3,Y)	0.81	0.24	0.70	0.36	0.96	0.11	0.66	0.37	0.58	0.44
BiCRNN(1,N)	0.78	0.26	0.77	0.31	0.96	0.11	0.68	0.36	0.80	0.29
BiCRNN(1,Y)	0.84	0.26	0.76	0.31	0.96	0.11	0.67	0.38	0.81	0.27
BiCRNN(2,N)	0.70	0.36	0.79	0.27	0.96	0.11	0.73	0.33	0.78	0.32
BiCRNN(2,Y)	0.83	0.23	0.76	0.31	0.96	0.11	0.68	0.37	0.84	0.23
BiCRNN(3,N)	0.65	0.41	0.79	0.27	0.96	0.11	0.67	0.36	0.75	0.29
BiCRNN(3,Y)	0.71	0.35	0.79	0.27	0.96	0.11	0.73	0.33	0.85	0.23
BiCLSTM(1,N)	0.78	0.26	0.73	0.33	0.96	0.11	0.68	0.36	0.85	0.24
BiCLSTM(1,Y)	0.79	0.27	0.74	0.33	0.96	0.11	0.68	0.38	0.82	0.26
BiCLSTM(2,N)	0.76	0.29	0.62	0.42	0.96	0.11	0.69	0.36	0.70	0.31
BiCLSTM(2,Y)	0.78	0.26	0.77	0.32	0.96	0.11	0.68	0.36	0.85	0.23
BiCLSTM(3,N)	0.79	0.27	0.76	0.33	0.97	0.11	0.70	0.34	0.85	0.22
BiCLSTM(3,Y)	0.78	0.28	0.74	0.33	0.95	0.11	0.70	0.35	0.85	0.23
BiCGRU(1,N)	0.76	0.29	0.72	0.33	0.96	0.11	0.68	0.36	0.81	0.27
BiCGRU(1,Y)	0.78	0.26	0.74	0.34	0.96	0.11	0.68	0.37	0.84	0.25
BiCGRU(2,N)	0.75	0.29	0.59	0.45	0.96	0.11	0.69	0.35	0.85	0.21
BiCGRU(2,Y)	0.73	0.30	0.75	0.32	0.96	0.11	0.66	0.38	0.79	0.24
BiCGRU(3,N)	0.79	0.27	0.74	0.32	0.97	0.11	0.69	0.34	0.76	0.28
BiCGRU(3,Y)	0.75	0.27	0.69	0.35	0.96	0.11	0.68	0.36	0.85	0.23
Seq2Seq CRNN(1,N)	0.71	0.35	0.85	0.23	0.97	0.11	0.72	0.33	0.78	0.29

Table continued on the next page

Table 4.13 – *continued from the previous page*

Model	Avenue		Subway Entrance		Subway Exit		UCSD1		UCSD2	
	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER	AUC ROC	EER
Seq2Seq CRNN(2,N)	0.70	0.36	0.79	0.27	0.96	0.11	0.73	0.33	0.82	0.27
Seq2Seq CRNN(3,N)	0.67	0.36	0.76	0.30	0.96	0.11	0.72	0.34	0.57	0.45
Seq2Seq CLSTM(1,N)	0.68	0.33	0.85	0.23	0.97	0.07	0.71	0.34	0.76	0.33
Seq2Seq CLSTM(2,N)	0.69	0.33	0.85	0.22	0.98	0.06	0.74	0.32	0.83	0.25
Seq2Seq CLSTM(3,N)	0.67	0.35	0.86	0.22	0.98	0.07	0.74	0.31	0.82	0.24
Seq2Seq CGRU(1,N)	0.68	0.34	0.85	0.22	0.97	0.07	0.73	0.33	0.84	0.24
Seq2Seq CGRU(2,N)	0.69	0.34	0.85	0.22	0.97	0.07	0.74	0.32	0.86	0.20
Seq2Seq CGRU(3,N)	0.70	0.34	0.85	0.23	0.97	0.08	0.74	0.32	0.80	0.31

4.10 Explainability in AutoEncoders

As discussed in the previous chapter, often with reconstruction-based anomaly detection using CAEs, the reconstruction error map between the input and reconstruction can be obtained, normalized and used as a visual indicator of the anomalies. But the *residual reconstruction error maps* show only the absence or a deformed or partial reconstruction of a visual entity and not localization of the precise anomalous regions of the input image. The error maps result in similar pattern as the input image rendering them useless in the understanding of decisions of the model in terms of the learnt notion of normality that is reflected in the reconstructions. As discussed in the previous chapter, XAI methods can prove beneficial in understanding the decisions of reconstruction based anomaly detection

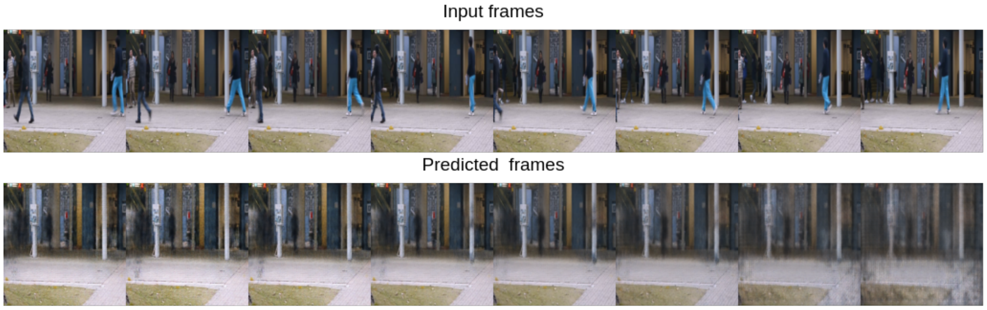


Figure 4.12: Poor quality of predicted frames in Seq2Seq CLSTM NN with $L_R = 2, DUT = N$

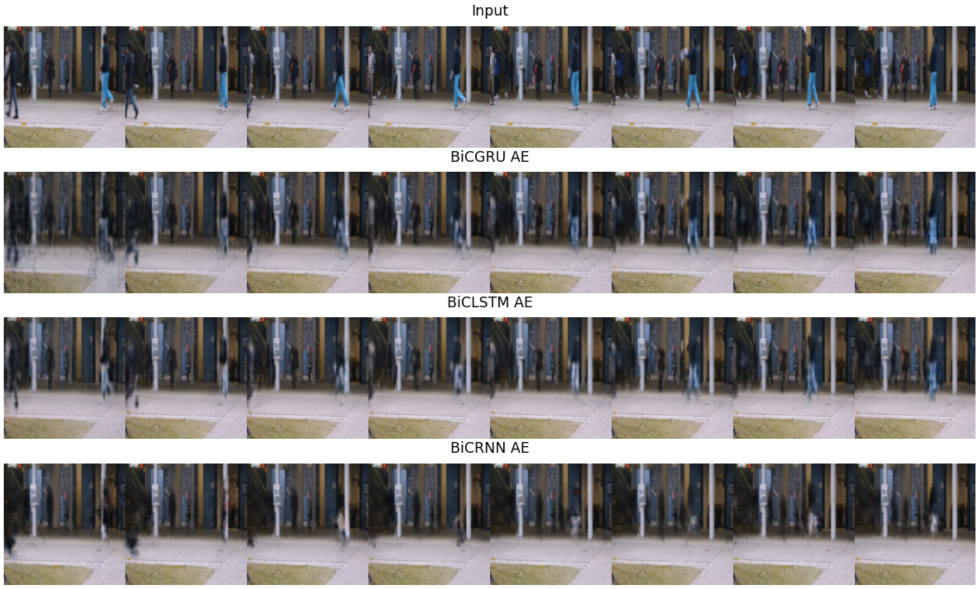


Figure 4.13: BiDirectional exhibiting better learning of motion patterns

Model	Number of trainable parameters					
	DUT = 2D TransposeConv Upsampling			DUT = Recurrent Upsampling		
	$L_R = 1$	$L_R = 2$	$L_R = 3$	$L_R = 1$	$L_R = 2$	$L_R = 3$
CRNN AE	0.60 M	0.76 M	0.84 M	0.76 M	1.00 M	1.15 M
CLSTM AE	1.01 M	1.51 M	1.77 M	1.59 M	2.40 M	2.92 M
CGRU AE	0.85 M	1.20 M	1.39 M	1.34 M	1.96 M	2.36 M
BiCRNN AE	0.85 M	1.23 M	1.42 M	1.26 M	1.85 M	2.22 M
BiCLSTM AE	1.67 M	2.72 M	3.28 M	2.92 M	4.65 M	5.76 M
BiCGRU AE	1.34 M	2.12 M	2.52 M	2.42 M	3.77 M	4.65 M
Seq2Seq CRNN NN	0.74 M	0.91 M	0.98 M	0.91 M	1.15 M	1.29 M
Seq2Seq CLSTM NN	1.16 M	1.65 M	1.91 M	1.74 M	2.55 M	3.06 M
Seq2Seq CGRU NN	0.99 M	1.35 M	1.53 M	1.49 M	2.11 M	2.51 M

Table 4.12: Complexity of the various model configurations

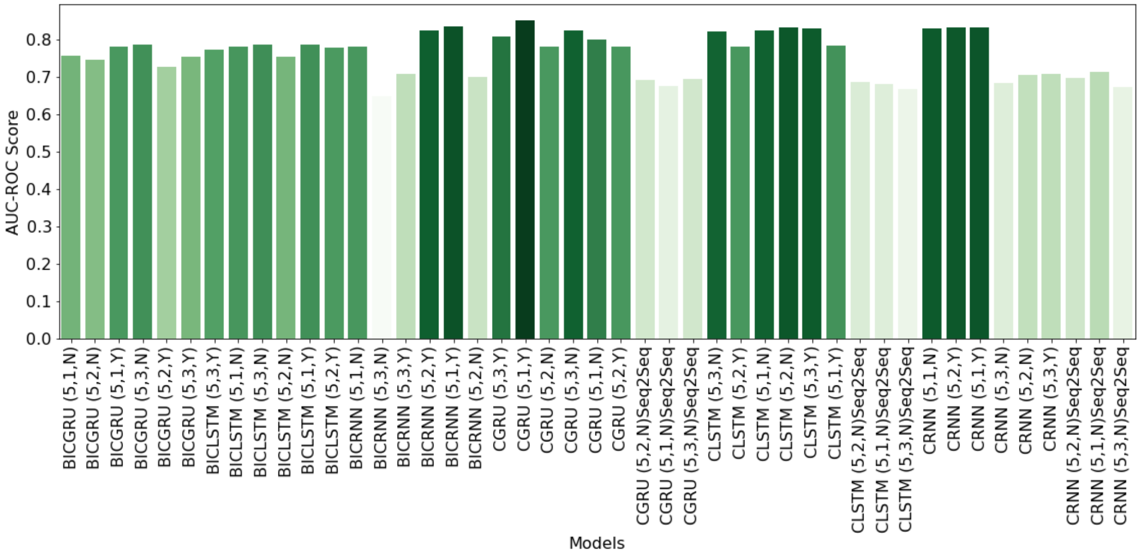


Figure 4.14: AUC-ROC scores of different recurrent convolutional models on AVENUE dataset

frameworks pertaining to the error values. To accommodate popular explainability frameworks that were initially proposed for classifiers, minor modifications are to be made by employing a parameter-less loss layer at the end connecting inputs and reconstructions of the BCAE1 model. The experiments discussed in this section are easily extendable to the video datasets by operating frame-wise although the content of this section essentially

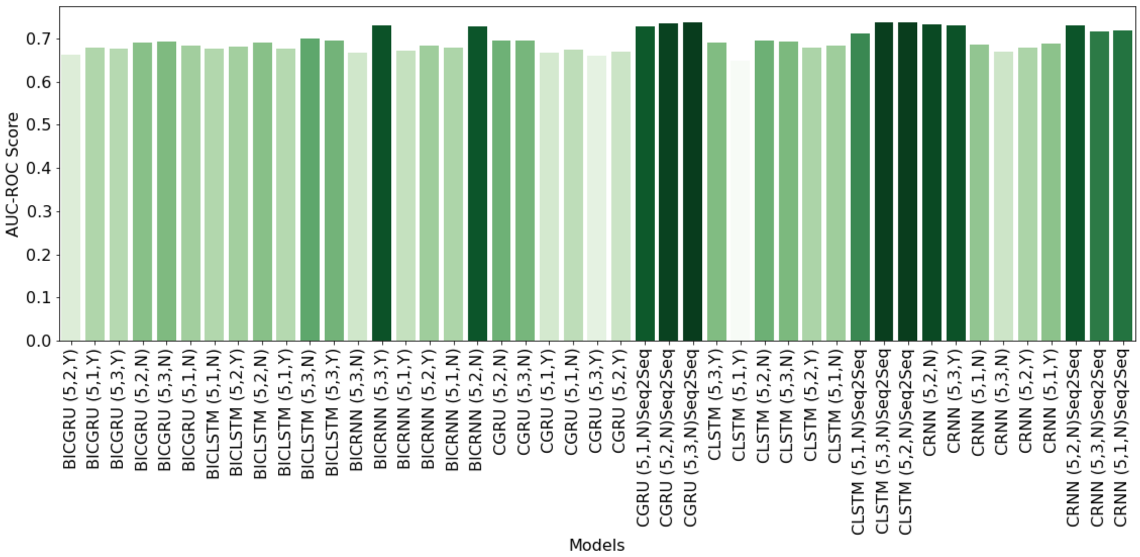


Figure 4.15: AUC-ROC scores of different recurrent convolutional models on UCSD1 dataset

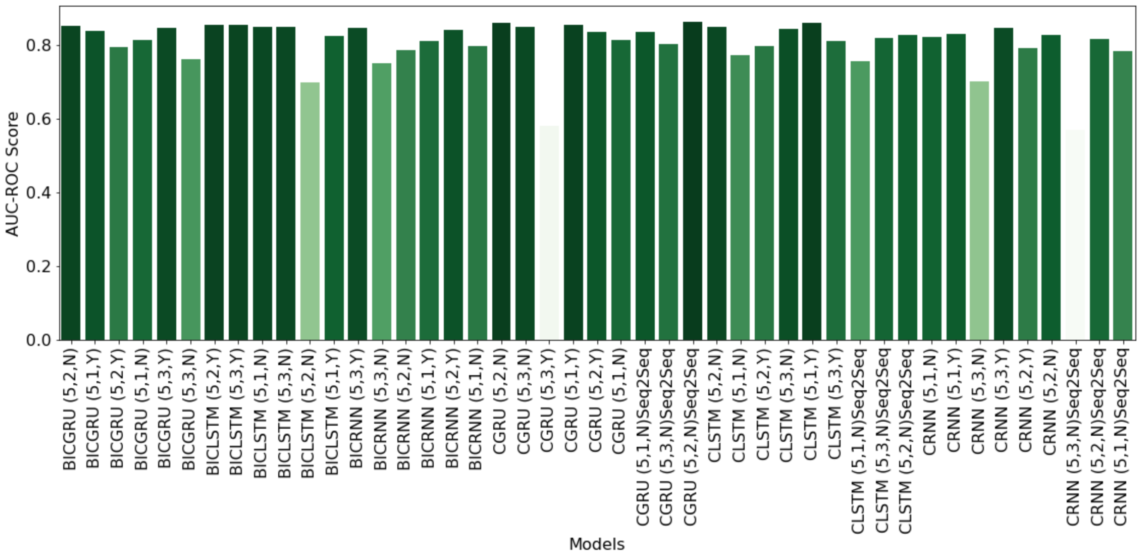


Figure 4.16: AUC-ROC scores of different recurrent convolutional models on UCSD2 dataset

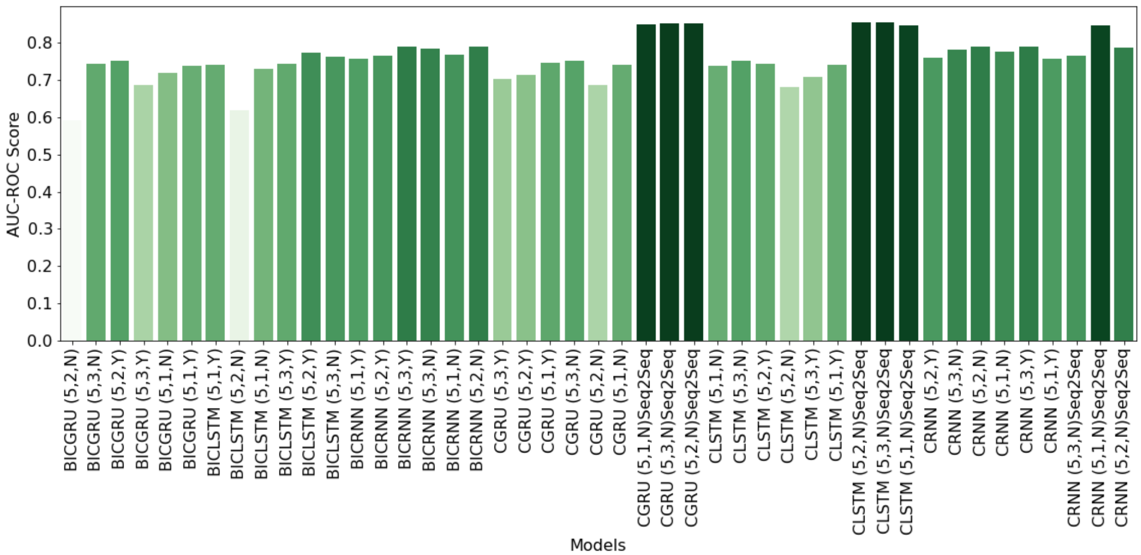


Figure 4.17: AUC-ROC scores of different recurrent convolutional models on SUBWAY ENTRANCE dataset

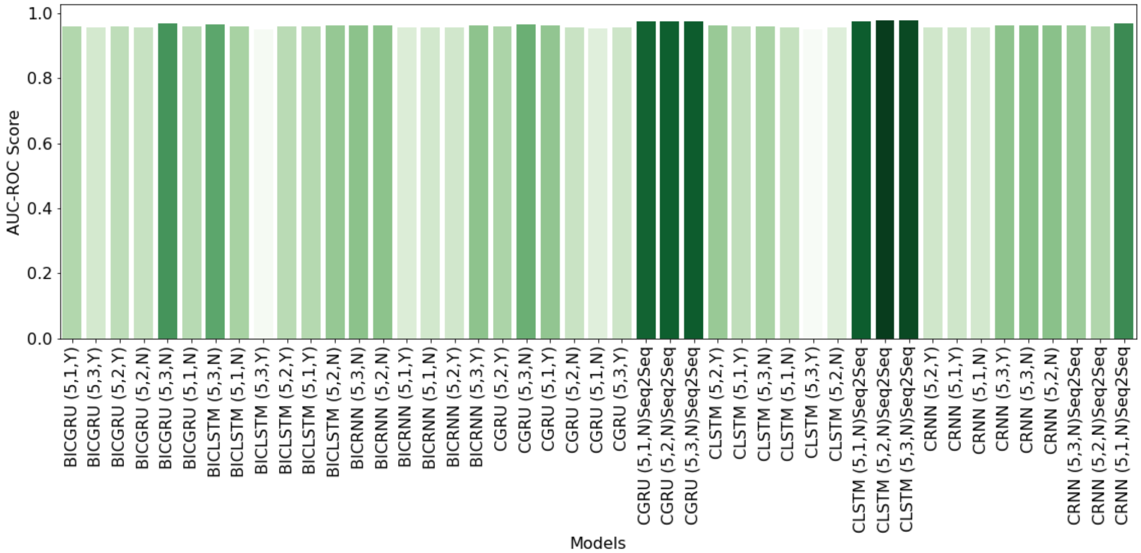


Figure 4.18: AUC-ROC scores of different recurrent convolutional models on SUBWAY EXIT dataset

focuses on the explainability image anomaly detection models.

4.10.1 Experiments

The experiments to explore the feasibility of applying generic CNN based explainability frameworks to CAEs are performed on BCAE1 models on *MVTec* and *DDDS* datasets as it is easier to understand the nature of anomalies in them. Four explainability methods which are e-LRP, LIME, SHAP and CounterFactuals are evaluated for the purpose of the experiments. The modification procedure to BCAE1 discussed in section 3.6.5 is applied to the frameworks that are tightly built around results of classification i.e. prediction confidence such as LIME and SHAP. For the other two approaches e-LRP and CounterFactuals, the reconstruction based CAE model without this modification is used. The reconstructions and the error values for the models are obtained for the test inputs and analysed through the frameworks after normalization of error values which mimic probability scores of normality.

4.10.2 Results and Analysis

The results[7] of the experiments from different XAI frameworks with the inputs and reconstructions of the models can be seen in 4.19 for both anomalous and normal samples. Overall, it is observable from the figure that the explanations from SHAP and Counterfactuals are able to meticulously identify the anomalous patterns better. For example, the hand movements responsible for anomalous behaviour of *operating GPS* while driving is indicated by SHAP that marks the entire region of the right hand of the person and by the counterfactual through a bounding-box enclosure around the region. Although LIME is able to identify the anomalies with regional neighbourhood of annotations, is not able to localize to the region with precision and the explanation from e-LRP is not visually understandable. Similarly, SHAP is able to precisely mark the phone in the person's hand, marking the anomaly of *texting* while driving, counterfactual indicates the same with a larger bounding box and LIME with a region around the person's upper-front body. Likewise, on *MVTec* dataset, SHAP marks the anomalies like *crack in a tile* and *print on a hazelnut* sharply without touching the other regions while counterfactual and LIME perform fairly well in indicating the regions although they are larger. The explanations from

[7] More results of test samples are available in the supplementary material pertaining to Paper [5] in the statement of contribution

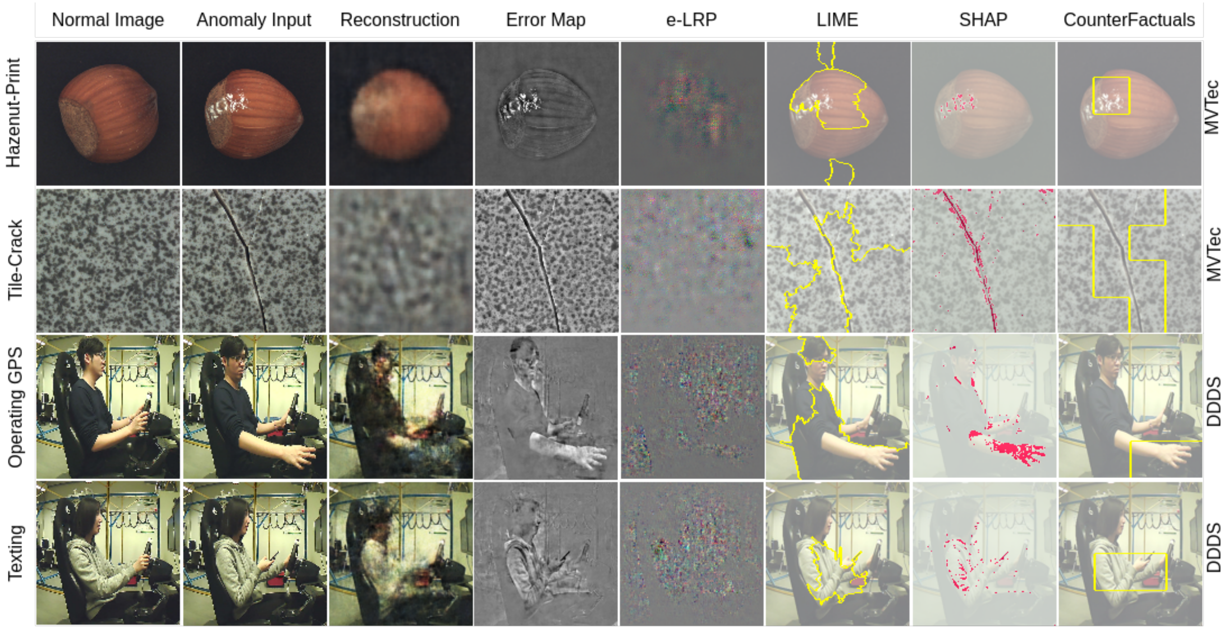


Figure 4.19: Comparison of the XAI frameworks on MVTect and DDDS for the anomalies *Print in Hazelnut*, *Crack in a tile*, *Operating GPS* and *Texting* respectively

e-LRP are the approximate pixelated contributions towards the decision made by the network and hence are incomprehensible in all the cases rendering it to be ineffective for this task as the prediction is based on value from a single output value from the modification made.

The inability of LIME to indicate the location of the anomalies as well as the other two methods can be attributed to the fact that the LIME surrogate model is meant to be an approximation of the original model and the performance depends on the degree of approximation. SHAP can be regarded as a more flexible version of Counterfactual since SHAP replaces features with random variables to determine their contributions towards the final output prediction through the relative difference from the original prediction. As a result of this, it allows SHAP to create masks of the arbitrary shape of feature importance as opposed to Counterfactual where only rectangular shapes are possible. It is important to note that the shape and size of block explanations in Counterfactuals can be customized to make the explanations more relevant to the use case. As we compare the explanations from the frameworks with the conventionally used residual reconstruction error maps to explain anomalies, the ineffective nature of error maps to correctly identify and precisely explain them can be emphasized through the visually interpretable results from figure 4.19. This shows the effectiveness of the proposed solution to use XAI frame-works with structural modifications to CAEs and the potential applications in visual anomaly detection tasks.

4.11 Summary

In this chapter, a variety of datasets were introduced that were used for experiments on several proposed approaches. The quantitative and qualitative evaluation of each of the proposed approaches on multiple datasets demonstrates the effectiveness and superiority of the approaches over the conventional baseline models for the task of anomaly detection in images and videos improving the performance by a significant margin.

Chapter 5

Conclusion

The primary objective of this research was to study and create novel deep learning architectures and incorporate feasible, model-agnostic and computationally efficient modifications to improve visual anomaly detection performance without any increment in data or computational requirements. Another objective was to explore and study various convolutional recurrent architectures for video anomaly detection and validate the effectiveness of the popular ConvLSTM configuration. And finally to complement the improvements in performance with understanding the decision behind different models, this research also tackles the problem of explainability of the convolutional AutoEncoder configuration using simple yet efficient modifications to the architecture to support popular explainability frameworks that were designed around convolutional networks for tasks like classification and object detection.

The open problems that were identified with the current approaches in the literature review section were all tackled with dedicated solutions to each of them. To improve the efficiency of learning in convolutional AutoEncoders to extract better performance from them on the same data without substantial increase in complexity, modifications like incorporating kernel strengthening in convolutional layers, using visual similarity metric into the objective function and novel architecture using visual soft-attention in AutoEncoders were proposed and successfully proven to be performing better than their baseline counter-parts on several datasets. The second challenge of increasing the ability to discriminate between normal and abnormal samples by incorporating the same into the objective function to train the Convolutional AutoEncoders was successfully addressed using SVDD encoder regularization technique which showed significant boost in performance with very little additional effort in terms of fine-tuning. The true effectiveness and feasibility of different convolutional recurrent configurations were explored, compared with other state of the art works and few interesting findings were also presented on multiple video datasets from which it was concluded that ConvGRU variants performed better than the popular ConvLSTM configuration in contrast to what was seen form literature. Finally, it was shown that methods like SHAP and Counterfactuals can be applied to convolutional AutoEncoders with few modifications to the architecture and from the experiments, it was observed that the popular XAI frameworks were more informative in enunciating the decisions of the models based on the variation in reconstruction error along with effectively spotting the anomalies spatially in the input images.

Every experiment in this research is accompanied by comprehensive quantitative analysis with comparisons to both the baseline models and the SOTA results from other works with conspicuous performance improvements. Moreover, the code-base containing all the experiments are made available for repeatability of the experiments and to benefit the AI community. All the proposed solution are model-agnostic and can be easily applied to any convolutional AutoEncoder variant for visual anomaly detection task and most findings from this research can be applied across various models for vision-based tasks to help researchers and AI practitioners to make informed design choices.

Future work

Although many of the proposed data-efficient techniques are able to perform considerably well under most circumstances, there are still room for improvements with potential future work. Some of the important issues and ideas that are still needed to be explored and tackled are listed below:

1. Apart from kernel strengthening, there are many other augmented alternatives for convolutional layers that can impart better learning at the layer level. Effectiveness and impact of such techniques like residual connections and parallel convolutions are to be studied in detail for the task of anomaly detection.

2. The application of visual similarity based loss for video anomaly detection has to be potentially carried out and tested.

3. Hybrid and ensemble models for conjoint image and video anomaly detection are possible using convolutional recurrent networks and it is an interesting path to explore the feasibility of such setups in the future.

4. The effectiveness of generative models for video anomaly detection hasn't been studied closely in the literature. Given that the generative models are proven to perform well on images and translation tasks like text to image generation, studying their effectiveness, visualizing their outputs and comparing them to that of the reconstruction and prediction based approaches could lead to some interesting insights.

5. Though this work studies four popular explainability frameworks for image anomaly detection, due to time constraints other possible frameworks and the applicability of XAI frameworks to video anomaly detection could not be carried out and should be explore in the future.

6. Only image level or frame-level anomalies were considered for experiments in this work. Few datasets also contain spatially annotated anomalous frames and the ability of deep learning models to spatially localize to the anomalous regions should be further explored where proposed mechanisms like soft-attention could prove more effective.

References

[1] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE transactions on pattern analysis and machine intelligence*, 30(3):555–560, 2008.

[2] Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.

[3] Charu C Aggarwal. An introduction to outlier analysis. In *Outlier analysis*, pages 1–34. Springer, 2017.

[4] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pages 622–637. Springer, 2018.

[5] H. Akrami, A. A. Joshi, J. Li, S. Aydore, and R. M. Leahy. Brain lesion detection using a robust variational autoencoder and transfer learning. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 786–790, 2020.

[6] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[7] Liat Antwarg, Bracha Shapira, and Lior Rokach. Explaining anomalies detected by autoencoders using SHAP. *CoRR*, abs/1903.02407, 2019.

[8] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[10] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49, 2012.

[11] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI Brainlesion Workshop*, pages 161–169. Springer, 2018.

[12] Momotaz Begum and Fakhri Karray. Visual attention for robotic cognition: A survey. *IEEE Transactions on Autonomous Mental Development*, 3(1):92–105, 2010.

[13] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3286–3295, 2019.

[14] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[15] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[16] L. Bergamasco, S. Saha, F. Bovolo, and L. Bruzzone. An explainable convolutional autoencoder model for unsupervised change detection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020:1513–1519, 2020.

[17] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019.

[18] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

[19] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[20] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[21] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[22] Min Chen, Xiaobo Shi, Yin Zhang, Di Wu, and Mohsen Guizani. Deep features learning for medical image analysis with convolutional autoencoder neural network. *IEEE Transactions on Big Data*, 2017.

[23] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[24] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *International symposium on neural networks*, pages 189–196. Springer, 2017.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[26] Taylor Denouden, Rick Salay, Krzysztof Czarnecki, Vahdat Abdelzad, Buu Phan, and Sachin Vernekar. Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance. *arXiv preprint arXiv:1812.02765*, 2018.

[27] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1911–1920, 2019.

[28] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[29] Ugo Fiore, Francesco Palmieri, Aniello Castiglione, and Alfredo De Santis. Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, 122:13–23, 2013.

[30] Anupriya Gogna and Angshul Majumdar. Discriminative autoencoder for feature extraction: Application to character recognition. *Neural Processing Letters*, 49(3):1723–1735, 2019.

[31] Pegah Sagheb Haghighi, Olurotimi Seton, and Olfa Nasraoui. An explainable autoencoder for collaborative filtering recommendation. *CoRR*, abs/2001.04344, 2020.

[32] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016.

[33] M. Haselmann, D. P. Gruber, and P. Tabatabai. Anomaly detection using deep learning based image completion. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1237–1242, 2018.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[35] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[36] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[37] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[38] Ling Huang, XuanLong Nguyen, Minos Garofalakis, Michael I Jordan, Anthony Joseph, and Nina Taft. In-network pca and anomaly detection. In *Advances in Neural Information Processing Systems*, pages 617–624, 2007.

[39] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[40] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[41] T. Kamitani, S. Fujimoto, H. Yoshimura, M. Nishiyama, and Y. Iwai. Anomaly detection using local regions in road images acquired from a hand-held camera. In *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pages 375–378, 2018.

[42] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8):5455–5516, 2020.

[43] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[44] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018.

[45] Yeara Kozlov and Tino Weinkauf. Persistence1d: Extracting and filtering minima and maxima of 1d functions. *h ttp://people. mpi-inf. mpg. de/weinkauf/notes/persistence1d. html, accessed*, pages 11–01, 2015.

[46] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[47] Kingsly Leung and Christopher Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342, 2005.

[48] Xuan Li, Christian Desrosiers, and Xue Liu. Out-of-distribution detection using vision transformers. *OpenReview.net preprint MIDL 2021*, 2021.

[49] Xuan Li, Yuchen Lu, Christian Desrosiers, and Xue Liu. Out-of-distribution detection for skin lesion images with deep isolation forest. In *International Workshop on Machine Learning in Medical Imaging*, pages 91–100. Springer, 2020.

[50] Zhaoyan Li, Yaoshun Li, and Zhisheng Gao. Spatiotemporal representation learning for video anomaly detection. *IEEE Access*, 8:25531–25542, 2020.

[51] C. Lile and L. Yiqun. Anomaly detection in thermal images using deep neural networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2299–2303, 2017.

[52] X Liu and M Milanova. Visual attention in deep learning: a review. *Int Rob Auto J*, 4(3):154–155, 2018.

[53] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013.

[54] Yuchen Lu and Peng Xu. Anomaly detection for skin disease images using variational autoencoder. *arXiv preprint arXiv:1807.01349*, 2018.

[55] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[56] Weixin Luo, Wen Liu, and Shenghua Gao. Remembering history with convolutional lstm for anomaly detection. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 439–444. IEEE, 2017.

[57] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981. IEEE, 2010.

[58] Jefferson Ryan Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[59] Asim Munawar, Phongtharin Vinayavekhin, and Giovanni De Magistris. Limiting the reconstruction capability of generative neural network using negative learning. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.

[60] Rashmiranjan Nayak, Umesh Chandra Pati, and Santos Kumar Das. A comprehensive review on deep learning-based methods for video anomaly detection. *Image and Vision Computing*, page 104078, 2020.

[61] Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1273–1283, 2019.

[62] Trong Nguyen Nguyen and Jean Meunier. Hybrid deep network for anomaly detection. *arXiv preprint arXiv:1908.06347*, 2019.

[63] Chaojie Ou, Qiang Zhao, Fakhri Karray, and Alaa El Khatib. Design of an end-to-end dual mode driver distraction detection system. In *International Conference on Image Analysis and Recognition*, pages 199–207. Springer, 2019.

[64] A. Paul, A. Majumdar, and D. P. Mukherjee. Discriminative autoencoder. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3049–3053, 2018.

[65] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[66] Manassés Ribeiro, André Eugênio Lazzaretti, and Heitor Silvério Lopes. A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters*, 105:13–22, 2018.

[67] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[68] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*, 2011.

[69] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

[70] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[71] David E Rumelhart, James L McClelland, PDP Research Group, et al. Explorations in the microstructure of cognition. *parallel distrubuted processing*, 1, 1986.

[72] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[73] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[74] S. Shashikar and V. Upadhyaya. Traffic surveillance and anomaly detection using image processing. In *2017 Fourth International Conference on Image Information Processing (ICIIP)*, pages 1–6, 2017.

[75] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015.

[76] Vasilis A Sotiris, W Tse Peter, and Michael G Pecht. Anomaly detection through a bayesian support vector machine. *IEEE Transactions on Reliability*, 59(2):277–286, 2010.

[77] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

[78] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018.

[79] Ilya Sutskever, Geoffrey E Hinton, and Graham W Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in neural information processing systems*, pages 1601–1608, 2009.

[80] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[81] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[82] Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt Jack Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7(1):1–30, 2020.

[83] Jing Tian, Michael H Azarian, and Michael Pecht. Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm. In *Proceedings of the European Conference of the Prognostics and Health Management Society*, pages 1–9. Citeseer, 2014.

[84] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, page 1–21, 2020.

[85] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[86] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

[87] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *CoRR*, abs/1812.05069, 2018.

[88] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[89] Chengwei Wang, Krishnamurthy Viswanathan, Lakshminarayan Choudur, Vanish Talwar, Wade Satterfield, and Karsten Schwan. Statistical techniques for online anomaly detection in data centers. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 385–392. IEEE, 2011.

[90] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[91] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015.

[92] Quanshi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology Electronic Engineering volume*, page 27–39, 2018.

[93] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

[94] Qiang Zhao and Fakhri Karray. Anomaly detection for images using auto-encoder based sparse representation. In *International Conference on Image Analysis and Recognition*, pages 144–153. Springer, 2020.

[95] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.

[96] Sijie Zhu, Chen Chen, and Waqas Sultani. Video anomaly detection for smart surveillance. *CoRR*, abs/2004.00222, 2020.