University of Sheffield

Real-Time Robustness to Modality Corruption in Multimodal Machine Learning

Harry Woods

Supervisor: Valentin Radu

Module Code: COM3610

A report submitted in fulfilment of the requirements
for the degree of BSc in Computer Science

in the

Department of Computer Science

18th May 2021

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Harry Woods

Abstract

This project discusses problems associated with modality corruption in multimodal contexts, and explores current techniques of anomaly detection. A novel anomaly detection pipeline based on Canonical Correlation Analysis is presented, able to detect corruption in all modalities simultaneously, provided at least 2 remain clean. The pipeline is applied to a multimodal MNIST classifier, achieving accuracy increases of up to 28.1% on highly corrupted data, whilst giving no detriment to accuracy when no corruption is present.

COVID-19 Impact Statement

The lockdown imposed because of COVID-19 caused additional challenges for the completion of this project. In the second semester of the project, the university switched to online delivery of all teaching, and university buildings were closed. All project meetings were shifted to email correspondence and video meetings.

Acknowledgements

I would like to thank my supervisor, Valentin Radu, for his help and support throughout this project, Andrei Popescu for his valuable feedback, and the Ubiquitous AI lab for keeping me motivated. I would also like to thank Tubby for remaining largely intact.

Contents

Abstract **ii**

COVID-19 Impact Statement **iii**

Acknowledgements **iv**

1 Introduction **1**

2 Background **2**

3 Literature Review **4**

3.1 Distribution shift detection 4

3.2 Modality reconstruction 5

3.3 Robustness 6

4 Analysis **7**

5 Method **9**

5.1 Canonical Correlation Analysis 9

5.1.1 Linear CCA 9

5.1.2 Generalised Canonical Correlation Analysis 10

5.1.3 Deep Generalised Canonical Correlation Analysis 10

5.2 Noise Generation 10

5.3 Corruption detection 11

5.3.1 Pairwise corruption detection 11

5.3.2 Modality corruption detection 12

5.4 Pipeline 14

6 Evaluation **16**

6.1 Multimodal-MNIST 16

6.1.1 Classifier 16

6.1.2 Correlation analysis 18

6.1.3 Anomaly detection 19

6.1.4 Modality cleaning 21

6.1.5 Single sample anomaly detection 24

7 Conclusion 26

7.1 Future work 26

A Relationship between k and c 30

B Full pipeline results 31

List of Figures

5.1 An example MNIST image under varying types and strengths of noise, with all gaussian mixture models containing 5 components. 11
5.2 Pairwise correlation distributions using two different threshold methods. Note the probability of a false negative can be chosen using the percentile method, giving a much smaller probability in Figure (b) than in Figure (a). 12
5.3 Example of relative corruption detected in modality pairs when modality 2 is corrupt in a 4 modality anomaly detector. Pairs containing modality 2 have correlations below their thresholds whilst the rest have correlations above their thresholds. The leading diagonal is ignored. 13
5.4 Multimodal classifier and corruption detection pipeline architecture. 15

6.1 Baseline accuracy for corrupt and subsequently cleaned data with 0, 1 or 2 corrupted modalities. The MM-MNIST classifier appears robust to the noise used above a SNR of around 0.87 with 1 or 2 corrupt modalities. Beyond this sensitivity threshold accuracy is greater if data is not cleaned. 17
6.2 Distribution of correlations under different noise and sample size conditions. . 18
6.3 Comparison of pairwise and modality corruption detection accuracy over different threshold generation methods. 20
6.4 Proportional and delta corruption classification methods. 20
6.5 Classifier results on corrupted and cleaned data using two different anomaly detector specifications. 21
6.6 Classifier results on corrupted and cleaned data with 1 or 2 corrupt modalities 23
6.7 Correlation between clean and corrupt modalities with a single sample. 24
6.8 Classifier results on corrupted and cleaned data using a single sample 25

List of Tables

6.1 MM-MNIST CNN classifier architecture 17

6.2 Anomaly detector classifier performance compared to performance on uncleaned and perfectly cleaned data. Threshold method: 0.0005th percentile, Classifier: Delta, Noise: All. Full table in appendix B 23

B.1 Anomaly detector, classifier, and pipeline accuracy. Threshold method: 0.0005th percentile, Classifier: Delta, Noise: All. 31

1 Introduction

Machine learning has been an active area of research in recent years and has been deployed in a number of tasks. Machine learning models often use data from a single view, or modality, such as an array of sensors on a single device or the image stream of a video. However, research has shown that combining data from multiple modalities can increase performance [1, 2, 3]. Multimodal learning comes with a number of challenges, including deciding when to fuse data from each modality and dealing with failure in individual modalities.

Existing research and implementations often assume that the multimodal data supplied to the model is complete and clean. However, in real world conditions data from a particular modality or sensor may be corrupted or completely missing, resulting in inaccurate results from such a model. Multimodal models should either understand when they have been fed corrupted samples, suggesting the output could be inaccurate, or be trained in such a way that they are robust to them.

This project aims to maximise model performance in the presence of missing and corrupt modalities.

Chapters 2 and 3 will present existing methods of anomaly and distribution shift detection, data reconstruction, and model robustness, as well as other technologies that could be adapted for these purposes. Chapter 4 presents possible next steps for exploration for the problem and its solutions. Chapters 5 and 6 detail a novel real-time anomaly detection pipeline and present results of its application to a multimodal dataset. Finally, Chapter 7 gives concluding remarks and possible future areas of exploration.

2 Background

Machine learning has been used over a wide range of tasks including computer vision, recommendation, medical diagnosis and human activity recognition. Shallow methods such as Random Forest and Support Vector Machines have been superseded by deep learning techniques as computational power availability has increased. Various additions to the basic feed-forward neural network have been made to better take advantage of spatial and temporal relationships between features in data sources such as images, audio and video streams, and real time sensor data.

A Convolutional Neural Network (CNN) [4] uses convolutional layers to identify spatial or temporal features. Each layer learns a number of kernels with small receptive field which are convolved over the input to produce an activation map. Kernels represent features in the input, e.g a 2D kernel in an image could represent a vertical line, with its activation map showing the location of all vertical lines in the image. Stacking multiple convolutional layers identifies more complex, larger scale spacial features in the input in the same way as the human eye. CNNs can also identify temporal features by stacking sequential frames of data and applying kernels over the temporal dimension.

State of the art image classifiers have achieved high accuracy with CNNs on general image recognition datasets such as ImageNet. One drawback of CNNs is the relatively large storage and computational requirements needed to use them during both training and inference.

Another method of identifying temporal features is with a Recurrent Neural Network (RNN). Recurrent layers contain cells which receive their output from the previous forward pass as an additional input. This allows information to flow through the network temporally, and features can be identified over multiple time steps.

An extension of this, as used in [5], is the Long Short Term Memory unit (LSTM) [6]. An LSTM consists of the cell itself and input, forget and output gates, controlling flow of information. These gates allow values into the cell, to remain in the cell, and to flow out of the cell via the activation function, respectively. Although basic recurrent cells can theoretically learn features over arbitrary timescales, finite-precision arithmetic used in computation makes identifying long term features difficult during training. LSTM cells do not suffer from the same issues and are therefore better suited to learning over longer timescales.

Proposed multimodal architectures tend to differ on how early or late data from different modalities are brought together, known as sensor fusion [7]. On one extreme is Ensemble Classification, where a classification for each modality is obtained using an appropriate classifier. The results of these separate classifiers are used with a majority voting scheme for the final classification. The other extreme is Feature Concatenation, where features from all modalities are concatenated into a single vector used for classification. This has the advantage of being able to learn cross-modality relationships, but can hinder learning of intra-modality relationships.

A split multimodal network architecture [7] has been found to have better results than architectures that have early or late fusion. A small network for each modality is used to generate a set of features. These are then concatenated and fed into a further combined network. The modality specific networks allow the model to learn intra-modality relationships before being fused with the other modalities to learn cross modality relationships. Modout [8] is capable of learning when to share information between modalities during training.

Implementations often assume that the multimodal data supplied to the model is complete and clean. However, in real world conditions data from a particular modality or sensor may be corrupted or completely missing, resulting in inaccurate results from such a model. Methods of detecting or mitigating data modality corruption are discussed in the following section.

3 Literature Review

3.1 Distribution shift detection

Canonical Correlation Analysis (CCA) [9] is a method for learning linear transformations of two random variables such that their representations are maximally correlated. In a multimodal setting, highly correlated representations could suggest that modalities are behaving the same way they did during training, whilst low correlation could suggest that one of the modalities is misbehaving. Clear limitations of this method, constraining it to linear transformations between two modalities, have been addressed separately by Deep CCA [10] which learns nonlinear transformations, and Generalised CCA [11] which can be applied to arbitrarily many modalities. Deep Generalised CCA (DGCCA) [12] addresses both simultaneously, being able to learn maximally correlated nonlinear representations between many views.

DGCCA representation are used for for K-nearest neighbour classification on XRMB and Twitter user datasets [12], obtaining better results than other CCA methods. Using a more complex classifier could improve these results. Alternatively, DGCCA could be used to identify corrupt modalities for removal or reconstruction before running a separate model on the resulting data.

CCA embeddings have been used for detection of single anomalous modalities [13] on the MM-Fit HAR dataset [14] achieving good accuracy increases compared to corrupted data. However, the limitation of only detecting a single modality is problematic for real world scenarios, as it requires knowledge of corrupted modalities beforehand.

Lipton et al. [15] present black box shift detection (BBSD), a method of detecting label shift. They show that, given any classifier f with invertible confusion matrix, to detect that the training distribution p differs from the real world distribution q it is sufficient to detect that $p(f(\boldsymbol{x})) \neq q(f(\boldsymbol{x}))$. Although created for label shift, studies show it can be applied to a variety of other shift types with good results, with [16] finding it performed better than other shift detection techniques in their comparison paper.

They note challenges with adapting BBSD to streaming scenarios. Accuracy increases as the number of samples considered increases, so best results would be obtained with a large window. However, this makes the estimate less fresh, limiting its utility for detecting short term shifts or anomalies in a single sample.

3.2 Modality reconstruction

A Cascading Residual Autoencoder (CRA) [17] can be used to impute data from missing modalities using those available. A Residual Autoencoder (RA) is trained to minimise the difference between a complete training sample and the same sample with corruption. Multiple RA's are stacked, with the sum of the input and output of each RA, the current reconstruction, used as the input for the next RA. The difference between the current reconstruction and complete data reduces after each RA and deeper CRAs produce better imputations.

CRA achieved good results on 4 datasets [17], outperforming all other matrix completion or autoencoder based imputation methods they test against. However, the datasets do not show much variation in modalities. For example in the HSFD dataset modalities are images of the same face taken in different spectral ranges, and in the RGB-D dataset the two modalities are RGB and Depth images of an object. It is unclear whether CRA could be effectively applied to modalities of significantly different types, such as audio, video, and caption data in movies.

Srivastava and Salakhutdinov [18] use a Deep Boltzmann Machine (DBM) to learn representations of multimodal data. A DBM is a multi-layered version of a Restricted Boltzmann Machine (RBM) [19], a generative neural network that can learn a joint probability distribution over its inputs. They independently pretrain a pathway for each modality and fuse them into a single representation with a joint layer, resulting in a network which performed better than other unimodal and multimodal models at the time. When modalities are missing the model is given the remaining modalities and the missing data can be generated by sampling the missing modality pathways. This generated data can be fed back into the network along with the available modalities for inference, giving higher accuracy than using the available modalities by themselves.

Unlike previous methods mentioned, the MIR Flickr dataset used for evaluation consists of two very different modalities, images and image tags. It was also adapted to use video and audio data from the CUAVE and AVLetters speech recognition datasets, achieving good results. This suggests a DBM could be a more generalizable method of multimodal learning than others and could be applied to more diverse modalities. Notably, the image features used were generated using PHOW, a shallow feature extractor from 2007, and modern image feature extraction methods such as CNNs could improve this performance.

Generative Adversarial Networks (GAN) have been used to reconstruct depth data from an RGB image [20]. A GAN consists of a Generator network and a Discriminator network.

Given an input, in this case RGB satellite data, the generator attempts to create a depth map. Rather than training this network with a loss function directly, the discriminator attempts to decide whether the generated depth map is real or fake and this decision is used to train the generator. Training is scheduled so that both networks get better at the same pace, resulting in a generator that can create realistic depth images from an RGB image.

This method does give modest gains in accuracy over using the corrupted modality or ignoring the modality completely. However, as with CRA the two modalities are similar, both representing spacial data with features in similar positions. Again it is unclear whether these could be applied generally to multiple different modalities.

3.3 Robustness

ModDrop [21] is a modality-scale regularisation method which makes predictions robust to missing or corrupted modalities. ModDrop works similarly to the node-scale regularisation method of dropout [22], where each node in a layer is temporarily removed from the network with a specified probability. Dropout ensures the network does not overfit the training data and reduces reliance of the network on individual connections. ModDrop removes entire modalities instead, having a similar effect of reducing reliance on individual modalities.

Networks trained with and without ModDrop are compared on two datasets, MNIST with each corner of an image considered a separate modality, and the ChaLearn LAP gesture recognition dataset augmented with their own audio data. Their augmented ChaLearn dataset contains RGB-D streams for each hand, an audio stream, and a motion captured articulated pose, giving six modalities of three distinct types. Removing individual modalities caused a mean accuracy reduction of 9%, compared to 23% when ModDrop was not used, suggesting it is a good method of dealing with missing modalities. They also obtained good results when individual modalities were corrupted with 50% pepper noise. However, this noise still retains much of the information contained in the original data, as suggested by the comparable scores without ModDrop, and may not be as challenging as the corruption encountered in the real world.

4 Analysis

The challenges involved in dealing with missing and corrupt modalities appear to be quite different. For starters, detection of a missing modality in a system is likely trivial as the data will be nonexistent, or the device missing. Corrupt data will be harder to detect as it will be presented to the model in the same way as a clean sample and these external indicators cannot be relied upon. Likewise, using ModDrop [21] during training appears to make a model largely robust to missing modalities, but not necessarily on those with corruption.

Supposing our model is robust to missing modalities, and possess a reliable method of detecting corruption, we may simply remove any corrupt modalities and allow our model to deal with the gaps. This would limit the effect of inaccurate information from that modality on the output and give the same performance as with missing modalities. However, this method would not be able to take advantage of any useful information that may remain in the corrupt modality.

A main focus of this project is therefore to detect any corruption in input modalities, ideally on a per-sample basis.

The BBSD method outlined in the previous section works well but requires many samples of data from a shifted distribution, in our case caused by corruption, to be accurate. This could detect longer term causes of corruption, for example a malfunctioning sensor, or any other changes that cause a modality to shift from the training data for a period of time. A key advantage is that it requires only an existing form of dimensionality reduction, which can be trained to best perform the task at hand. However, BBSD would be unable to detect corruption on a per-sample basis which could limit its applications in some environments.

CCA and its variants could be used for a novel method of corruption detection. If a CCA model is trained such that clean data samples are always transformed into maximally correlated representations, there is no guarantee that the representation of a corrupted modality will be correlated with them. Assuming this is true, modalities with low correlation can be considered corrupted. A possible advantage of this method over BBSD is that it may be able to detect corruption in single samples.

This method could be applied to detect and remove corruption before running the desired model on the remaining data. Alternatively, the representations learned by CCA, once corruption has been removed, could be used as features for inference. This may not perform as well as a purpose built model, but saves the computational cost of carrying out dimensionality reduction twice.

More exploration is needed to see if CCA can be used as proposed, for both corruption detection and informative dimensionality reduction, and this will be a major part of this project.

An alternative to inference without missing or corrupt modalities is to reconstruct them and use these instead of the missing data. Inference using the reconstruction methods outlined in the previous section have been found to increase accuracy over using missing modalities. It may be possible to incorporate corrupted data into the reconstruction process, allowing it to take advantage of any information that remains.

A more elegant solution than detecting corruption could be training the model so that it is robust to it. Further evaluation of ModDrop is necessary to see if it achieves this.

To evaluate any of these methods, it is necessary to understand what form corruption can take. [16] use a variety of techniques to alter the distribution of MNIST but many of the images are quite close to their usual form. [21] use pepper noise for testing, which could occur in some scenarios, but there are other forms of corruption to be addressed.

5 Method

The proposed anomaly detection system uses the correlations between inputs produced by DGCCA to detect which modalities, if any, are corrupted.

5.1 Canonical Correlation Analysis

5.1.1 Linear CCA

Canonical Correlation Analysis [9] is a a generalised way of analysing cross covariance matrices between two random variables. Given two random variables $\mathbf{X} = (X_1, ..., X_n | X_i \in \mathbb{R}^{d_X})$ and $\mathbf{Y} = (Y_1, ..., Y_n | Y_i \in \mathbb{R}^{d_Y})$, CCA learns transformations $\mathbf{u}_1 \in \mathbb{R}^{d_X}$ and $\mathbf{v}_1 \in \mathbb{R}^{d_Y}$ such that

$$corr(\mathbf{u}_1^T \mathbf{X}, \mathbf{v}_1^T \mathbf{Y}) \text{ is maximised.}$$

$(\mathbf{u}_1^T \mathbf{X}, \mathbf{v}_1^T \mathbf{Y})$ are known as the first pair of canonical variables. Further canonical variables maximise correlation orthogonally to the first pair of canonical variables, with the i^{th} pair of canonical variables $(\mathbf{u}_i^T \mathbf{X}, \mathbf{v}_i^T \mathbf{Y})$ maximising

$$corr(\mathbf{u}_i^T \mathbf{X}, \mathbf{v}_i^T \mathbf{Y})$$

subject to

$$cov(\mathbf{u}_i^T \mathbf{X}, \mathbf{u}_j^T \mathbf{X}) = cov(\mathbf{v}_i^T \mathbf{Y}, \mathbf{v}_j^T \mathbf{Y}) = 0, \forall j < i$$

In practice, CCA is carried out by performing singular value decomposition on the covariance matrix and taking the top k eigenvectors.

The transformations learned by CCA give the basis for the anomaly detection system. When two modalities contain clean data of the type encountered during training, it is expected that their canonical variables have a high correlation. If a modality contains corrupted data which differs enough from the training distribution, the transformation will not be optimal it is expected that the correlation will be lower.

As discussed, CCA is limited to maximising correlations between two sets of variates using linear transformations. To use CCA efficiently with many modalities a method of learning transformations for more than two sets of variates is required.

5.1.2 Generalised Canonical Correlation Analysis

GCCA [11] constructs a shared representation G and maximises the correlations between each set of variates and the shared representation.

5.1.3 Deep Generalised Canonical Correlation Analysis

DGCCA builds upon Deep CCA [10] to extend GCCA to learn nonlinear transformations. A neural network is trained for each modality to learn a nonlinear transformation to a new representation, which is then used for GCCA. The networks are trained by backpropagating the objective of GCCA, maximising the ability of their outputs to be correlated.

Benton et al. [12] derive a loss function for GCCA based on pairwise correlations between modalities.

Using DGCCA, representations of any number of modalities can be obtained, with the expectation that correlation between any two representations is high.

5.2 Noise Generation

For training and evaluation of the anomaly detector, a method of adding noise to the input modalities is required. A number of different methods of noise generation are used:

- Gaussian noise: Sample from the standard normal distribution for each feature in the input modality.
- Feature GMM: Sample from 1-D GMMs fitted to each feature of the input modality.
- Modality GMM: Sample from multi dimensional GMMs fitted to each modality.
- Full GMM: Sample from a multi dimensional GMM fitted to the entire dataset.

It is expected that each method produces more realistic and harder to detect noise than the previous, with more gmm components producing more realistic noise. Noise is added to the raw data on a per-modality basis before it is passed through the first stage of the classifier based on the signal to noise ratio specified (Figure 5.1), with a corrupt modality given by $corrupt = \frac{snr \times \mathbf{raw} + \mathbf{noise}}{snr + 1}$.

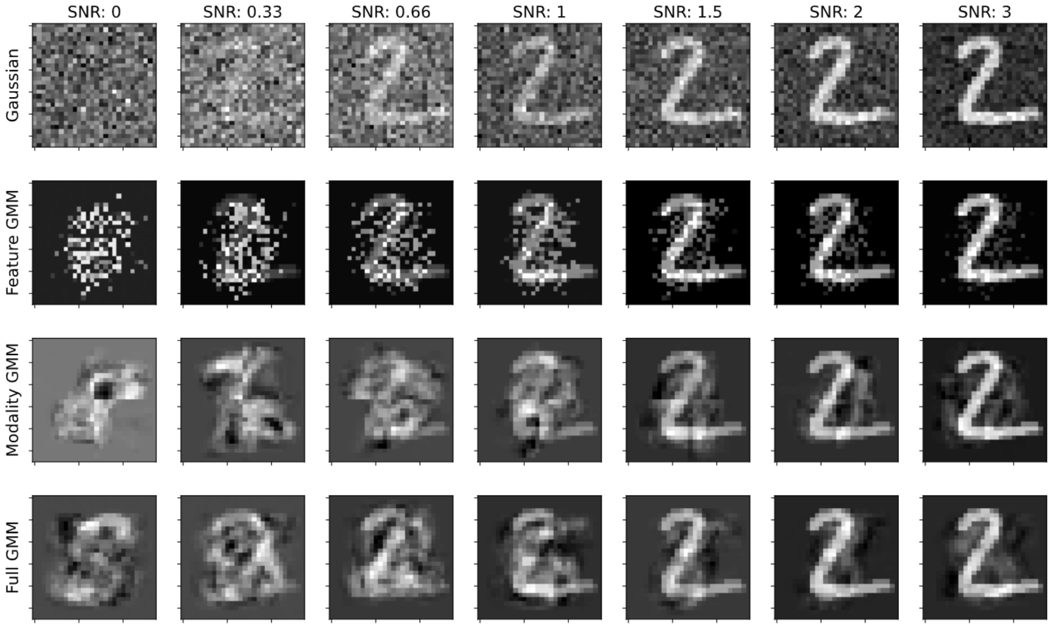


Figure 5.1: An example MNIST image under varying types and strengths of noise, with all gaussian mixture models containing 5 components.

5.3 Corruption detection

The general method for detecting corrupt modalities involves first looking at correlation between each of the $\frac{1}{2}(m-1)m$ possible pairs of modalities and using the information gained from each of these comparisons to infer which modalities are corrupt.

5.3.1 Pairwise corruption detection

To level of correlation between two modalities can be measured over one or many samples. The transformations learned by DGCCA are used to generate a representation of each modality. To calculate the correlation between modalities a and b based on N samples with cca dimension C, canonical variate matrices $A, B \in \mathbb{R}^{N\times C}$ are compared using one of two different methods.

- Combined correlation, the sum of correlations over each canonical variate:

$$CombinedCorr(A,B) = \sum_{d=1}^{C} corr((A_{s,d}|s=1..N),(B_{s,d}|s=1..N))$$

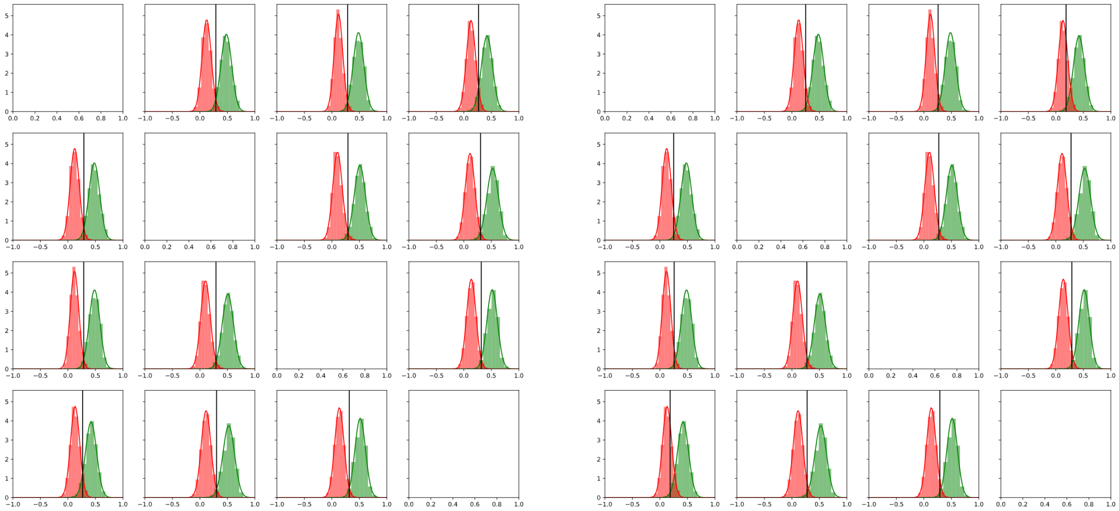
- Flat correlation, the correlation between the flattened canonical variate matrices:

$$FlatCorr(A,B) = corr((A_{s,d}|s=1..N, d=1..C),(B_{s,d}|s=1..N, d=1..C))$$

The expectation is that these measures are high when samples are drawn from a similar distribution to the training set, but low when samples are corrupted and fall outside of it. Note that *CombinedCorr* requires at least two samples, whilst *FlatCorr* can be used to calculate correlation between single samples.

To detect whether a given pair of modalities is corrupted, their correlations are compared with a threshold learned during training. Combined correlation is calculated between cca embeddings of clean data, as well as between clean and corrupt data. The intensity of corruption used affects the distribution of corrupt correlations, and therefore the learned threshold.

It is assumed the correlations follow a normal distribution, and two methods of choosing a threshold are proposed. To maximise accuracy of each pairwise anomaly detector (Figure 5.2a), thresholds are chosen as the intersection between probability density functions of clean and corrupt modalities. To minimise the number of false negatives (Figure 5.2b) (clean modalities erroneously classified as corrupt) thresholds are chosen as a given percentile of the clean distribution.



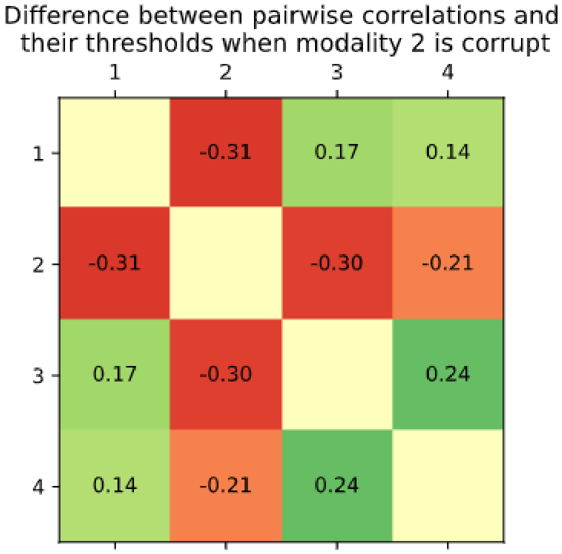
(a) Correlation thresholds using the intersection method. $P(\text{type I error}) = 0.028, P(\text{type II error}) = 0.023$

(b) Correlation thresholds using the percentile method with p-value 0.01. $P(\text{type I error}) = 0.010, P(\text{type II error}) = 0.074$

Figure 5.2: Pairwise correlation distributions using two different threshold methods. Note the probability of a false negative can be chosen using the percentile method, giving a much smaller probability in Figure (b) than in Figure (a).

5.3.2 Modality corruption detection

The pairwise corruption detection stage yields a matrix $M \in \mathbb{B}^{m \times m}$, where $M_{i,j}$ is true if both modalities are clean, and false if one or both modalities are corrupted. it is expected that when modality i is corrupted, all entries of the ith row and column are false. The remaining modalities are true for all entries except that of the corrupted modality. This means that when many modalities are corrupted the row corresponding to a clean modality may still contain many corrupt pairs, making classification at this stage more difficult. A number of different methods of classification have been used. Note that at least 2 modalities must be clean for any classification to be carried out, fewer clean modalities will result in all pairwise classifications being negative, so any system using pairwise correlations can detect corruption in at most $m - 2$ modalities.



	1	2	3	4
1		-0.31	0.17	0.14
2	-0.31		-0.30	-0.21
3	0.17	-0.30		0.24
4	0.14	-0.21	0.24	

Figure 5.3: Example of relative corruption detected in modality pairs when modality 2 is corrupt in a 4 modality anomaly detector. Pairs containing modality 2 have correlations below their thresholds whilst the rest have correlations above their thresholds. The leading diagonal is ignored.

Proportional classification

Let k be the proportion of pairs that are classified as corrupt, and let k_i be the proportion of pairs in row i that are corrupt. It is expected that $k_i = 1$ when modality i is corrupt, and $k_i = \frac{c}{m-1}$ otherwise. When M contains both clean and corrupt pairs and pairwise classifications are 100% accurate, $1 > k > \frac{c}{m-1}$. Therefore, by measuring k, modality i can be marked as corrupt if $k_i > k$, or clean otherwise.

In practice the accuracy of this method suffers when pairs are falsely classified as corrupt. If all modalities are clean, a single false negative pair classification can cause up to 2 clean modalities to be classified as corrupt, significantly impacting accuracy.

Corruption estimation

k can be calculated from the number of corrupt modalities c by

$$k = 1 - \frac{(n-c)(n-c-1)}{n(n+1)} \qquad (Appendix A)$$

As an estimate of k can be calculated directly from M, the formula can be used to estimate the number of corrupt modalities c'. If c' is an integer, there is a higher confidence that the number of pairwise classification errors is low, otherwise it is certain that there are errors. c' can also be used to limit the number of modalities classified as corrupt. Tests have been carried out limiting the number of corrupt modalities by rounding c' to the nearer, higher, or lower integer.

Delta/Probability classification

Rather than using the matrix M, the difference between pairwise correlations and their trained thresholds can be used to get a deeper view of corruption, based on the assumption that a genuinely corrupt pair will have a correlation further below their threshold than a false negative. The differences across each row of pairs are summed and the modality with the greatest negative value is marked as corrupt, removing its correlations from M. This continues until no row has a negative total correlation. A variation on this method uses the probability distributions learned during threshold training to estimate the probability correlation belongs to the clean and corrupt distributions. If the sum of corrupt probabilities is greater than the sum of clean probabilities, the modality is classified as corrupt, again removing its probabilities from M.

5.4 Pipeline

Figure 5.4 shows the full proposed classifier pipeline, using the decisions made by the anomaly detection system to remove corrupt modalities from the intermediate data representation fed to the final classification system.

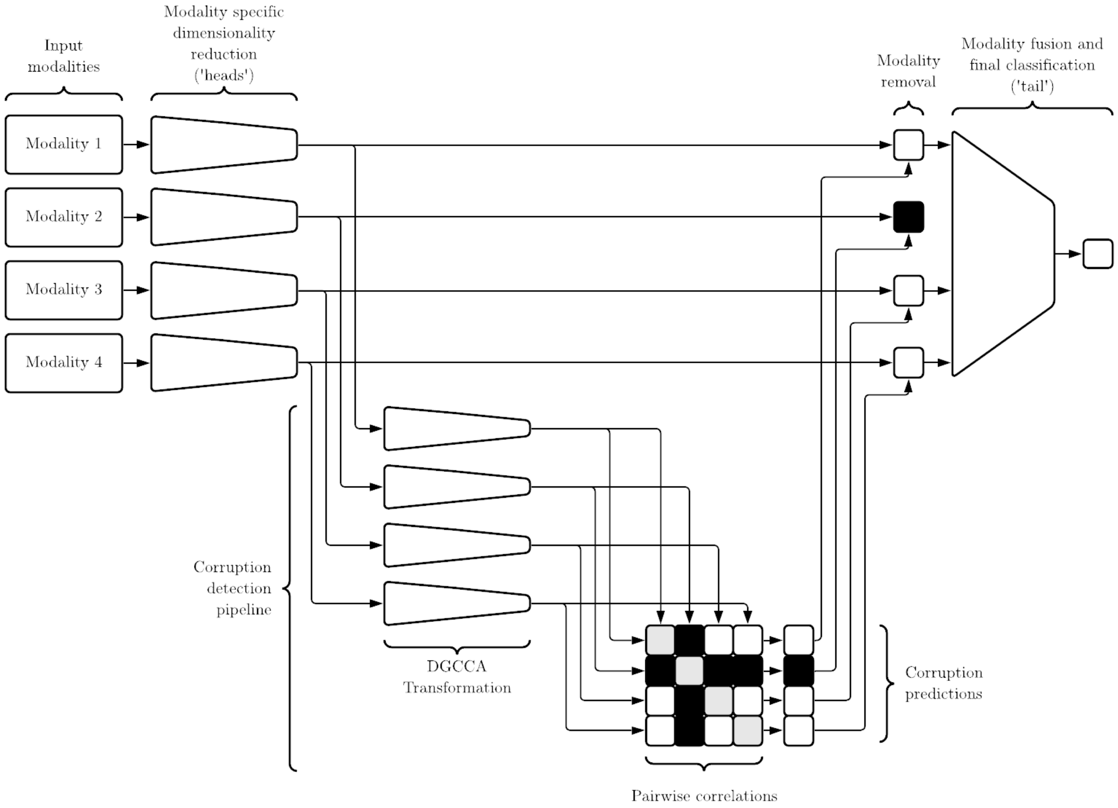


Figure 5.4: Multimodal classifier and corruption detection pipeline architecture.

6 Evaluation

6.1 Multimodal-MNIST

The anomaly detection pipeline is first applied to Multimodal MNIST (MM-MNIST), proposed in [21]. This dataset consists of the classic MNIST dataset [23] with each image split into quarters, giving four $14 \times 14 \times 1$ pixel modalities.

Full pipeline results are compared based on 4 metrics:

- Accuracy of the anomaly detection pipeline at classifying modalities as corrupt or clean.
- Accuracy of the MM-MNIST classifier when input data contains corrupt modalities. This is independent of the anomaly detector architecture and is uniform across settings.
- Accuracy of the MM-MNIST classifier when all corrupt modalities are removed. As all noise is removed this is independent of the type or strength of noise, and is equivalent to cleaning data with a 100% accurate anomaly detector.
- Accuracy of the MM-MNIST classifier when corrupt modalities are removed by the anomaly detector. This is the metric that should ultimately be maximised when compared to the previous two.

6.1.1 Classifier

A convolutional neural network is implemented with a separate head for each modality, and a tail which combines the head outputs and completes the classification (Table 6.1).

ModDrop [21] is used before the tail to improve robustness to missing modalities modalities will be removed at this stage. The classifier achieves 98.9% accuracy on the clean test set.

The full GMM method above is used to generate a baseline for accuracy on corrupted data (Figure 6.1). Repeated experiments across signal to noise ratio with 0, 1, or 2 corrupted modalities are carried out, and accuracy is measured on both the corrupted data, and the data with corrupted modalities removed.

Table 6.1: MM-MNIST CNN classifier architecture

Layer	**Kernel size/units**
Per-modality heads	
Input	$14 \times 14 \times 1$
Dropout	$p = 0.2$
Conv1	$32 \times 5 \times 5$
Conv2	$32 \times 3 \times 3$
MaxPool	2×2
Cross modality tail	
ModDrop	$p = 0.1$
Conv3	$64 \times 5 \times 5$
Conv4	$64 \times 3 \times 3$
MaxPool	2×2
Fully Connected	10
Log SoftMax	10

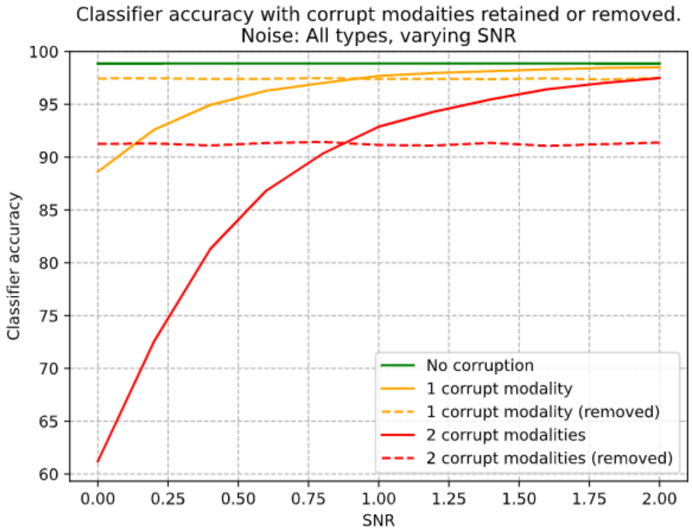


Figure 6.1: Baseline accuracy for corrupt and subsequently cleaned data with 0, 1 or 2 corrupted modalities. The MM-MNIST classifier appears robust to the noise used above a SNR of around 0.87 with 1 or 2 corrupt modalities. Beyond this sensitivity threshold accuracy is greater if data is not cleaned.

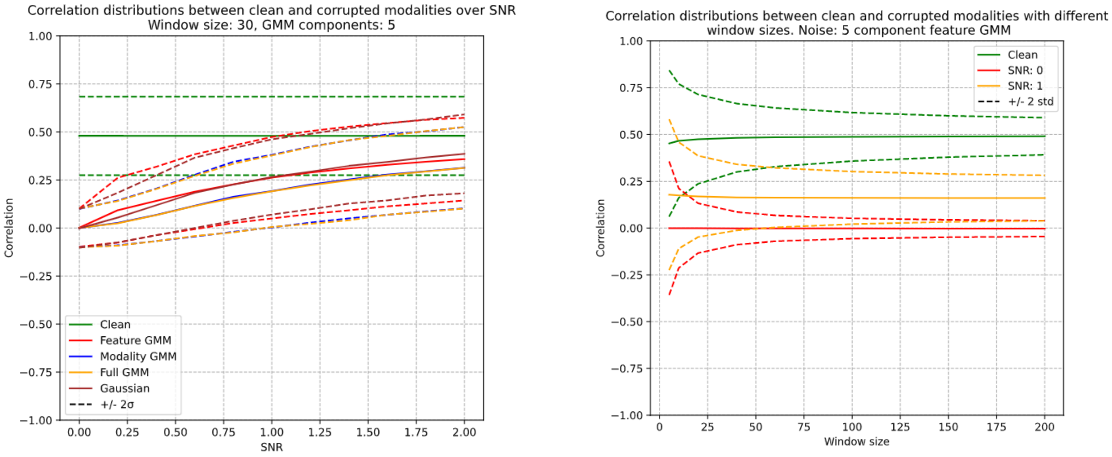
Figure 6.1 shows that unless a modality contains a high amount of corruption, the modality is still useful for overall classification. Removal of all corruption causes a drop in accuracy under many conditions. The the threshold at which accuracy is better on corrupt data than clean in Figure 6.1 lies at a signal to noise ratio of around 0.87, which in figure 5.1 is still reasonably obfuscated to the human eye.

As discussed in sections 3.2 and 4, attempting to clean each detected modality instead of removing them could make the most of the remaining signal.

6.1.2 Correlation analysis

The intermediate representations output by the head networks are used for anomaly detection, with the outputs flattened into 512 element vectors. The networks used in the first stage of DGCCA are fully connected, with layer sizes [512, 125, 64, 32]. These networks are trained on 50000 training samples, with the remaining 10000 held out for corruption detection training. Once the network heads are trained, a linear GCCA is trained on the resulting embeddings to produce transformations to the top *cca_dim* canonical variates.

Figure 6.2a gives an indication of separability of clean and corrupt data. When the amount of noise is high, there is a significant difference between correlations of clean and corrupt data, which suggests easy classification. However, as the amount of noise reduces, correlation and spread both increase and there is more overlap between distributions making classification more difficult. Correlation could also be used as an indicator for how much noise is present, though the usefulness of this may be limited due to the high spread. The distributions also suggest that the Gaussian and feature GMM noise generation methods are more difficult to detect than the more complex GMM methods.



(a) Distribution of correlations under various types and amounts of noise.

(b) Distribution of correlations as window size increases.

Figure 6.2: Distribution of correlations under different noise and sample size conditions.

Varying the number of samples used to calculate correlation could reduce spread of correlation distributions and increase separability. Figure 6.2b shows the spread of correlations reducing when more samples are used. Increasing the number of samples is not always desirable, as for real-time tasks waiting for a large number of undetected corrupt samples to be collected is not always possible.

For remaining experiments a sample size of 30 is used as it gives a reasonable balance of correlation spread and detection latency.

6.1.3 Anomaly detection

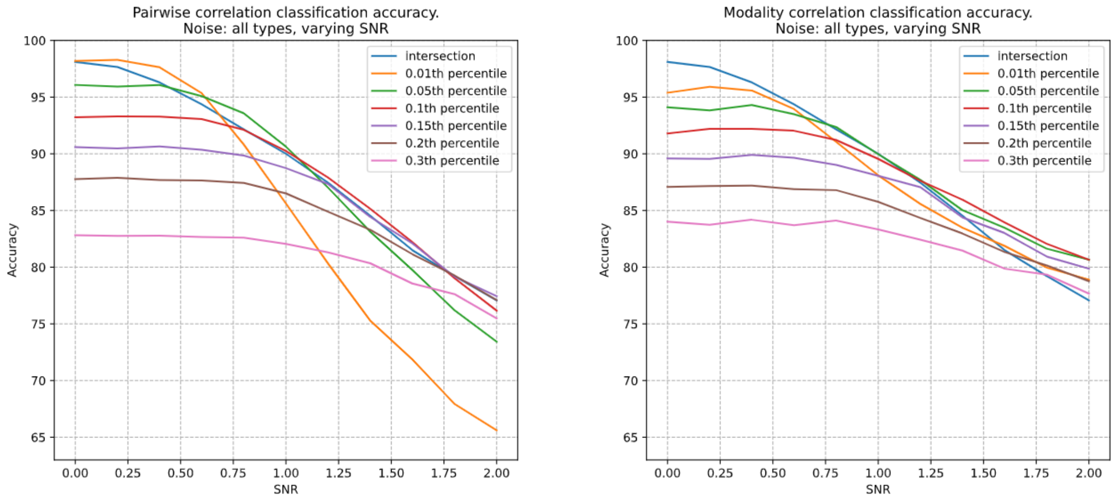
The remaining 10000 elements of the training set are used for training the pairwise correlation thresholds. Correlations between DGCCA embeddings for each pair of modalities are measured on clean data and on data that has been corrupted with noise. Figure 6.3a shows accuracy of pairwise corruption detection using various threshold methods. The best performing method is intersection, having higher accuracy in general than percentile methods. The best performing percentile is 0.05, having slightly lower accuracy than the intersection method.

Figure 6.3b gives modality corruption detection accuracy using the same threshold methods. As SNR increases, the accuracy of the percentile threshold method rises above that of the intersection threshold method. The number of pairwise false negatives is fixed when using the percentile threshold method, so a reduction in accuracy over SNR is purely dependent on an increase in false positives. For the intersection method, both false positives and false negatives increase with SNR. This suggests that false negatives during the pairwise classification stage have a greater negative impact than false positives on overall modality detection accuracy.

Aggregating results over multiple pairs increases robustness to these pairwise errors, and it is expected that increasing the number of modalities would increase the accuracy of the modality anomaly detector.

In addition to limiting of false negatives, the percentile threshold method does not require knowledge of the type of noise that will be encountered. In figure 6.3, the intersection classifier is trained by generating noise of the specified SNR, whilst the percentile method requires no noise during the entire training pipeline. This may make the percentile method more useful in real world scenarios where obtaining corrupt data may not be practical.

Figure 6.4 shows that the delta classifier gives significant accuracy gains over the proportional classifier using the 0.05 percentile threshold method, supporting the idea that correlation can be used as an indicator for quantity of noise, especially when correlations are available across multiple pairs.



(a) Pairwise corruption detection accuracy.

(b) Modality corruption detection accuracy.

Figure 6.3: Comparison of pairwise and modality corruption detection accuracy over different threshold generation methods.

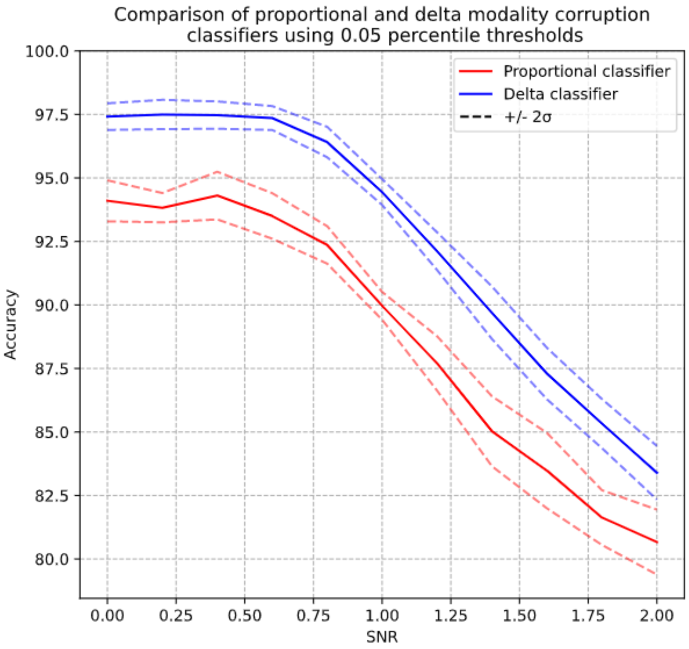


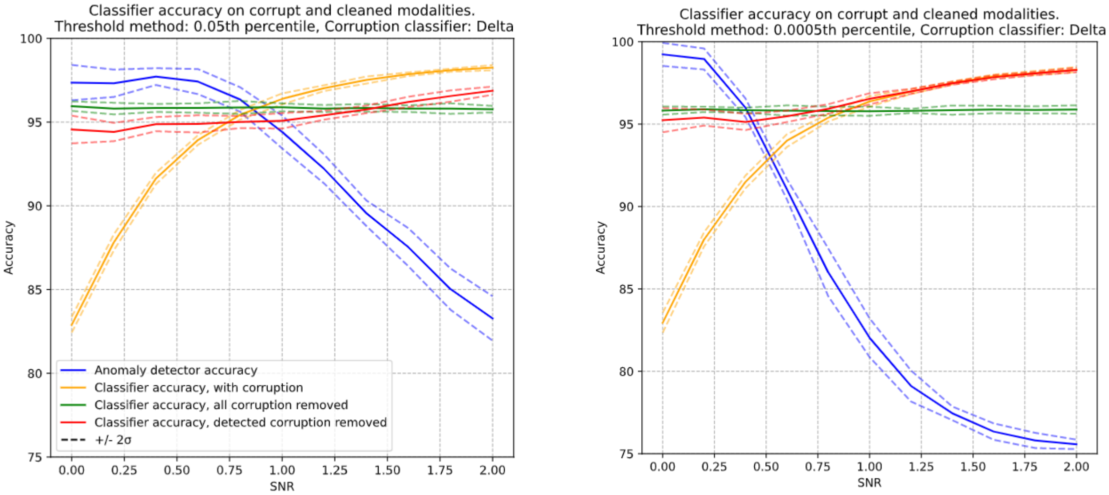
Figure 6.4: Proportional and delta corruption classification methods.

One reason for the significant reduction in detector accuracy as the SNR increases could be down to the design of the MM-MNIST classifier. The dropout, convolutional and MaxPool layers that the data has passed through at the point that CCA is applied could all have the effect of partially denoising the input. Though this reduces anomaly detector accuracy, it is useful for our overall goal of maximising classifier accuracy.

6.1.4 Modality cleaning

An overall goal of this project is to reduce the impact that corruption has on the multimodal network. To remove corruption, modalities identified as corrupt are replaced with zeros before being fed to the tail of the MM-MNIST classifier. Use of ModDrop as the first layer of the tail should make the classifier robust to modality removal at this stage, though performance is still expected to suffer if otherwise useful data is removed.

Classifier accuracy is compared on data with no modalities removed, known corrupt modalities removed, and modalities identified by the anomaly detection pipeline removed. Figure 6.5a shows accuracy across a range of noise.



(a) Performance of anomaly detector and MM-MNIST classifier on corrupted and cleaned data using the best found anomaly detector.

(b) Performance of anomaly detector and MM-MNIST classifier on corrupted and cleaned data using a lower percentile threshold.

Figure 6.5: Classifier results on corrupted and cleaned data using two different anomaly detector specifications.

With higher amounts of noise, the anomaly detector clearly improves classifier performance when compared to classifying using the corrupted data, with an accuracy increase of 11.6% when no signal is present. However, as the SNR increases classifier performance on the raw data exceeds that of the clean data, even when cleaned with a theoretically perfect classifier. The drop in accuracy resulting from cleaning the data with the anomaly detector is 1.3% with an SNR of 2. It is expected that in the real world modalities will be clean most of the time, so this essentially limits overall classifier accuracy.

The approach taken so far has been to maximise anomaly detector accuracy. Given the high classification accuracy on corrupted data above the sensitivity threshold, it is beneficial to allow certain amounts of corruption through to the classifier, which necessarily reduces anomaly detector accuracy.

Notice that the raw data (yellow) can be seen as having been cleaned by an anomaly detector with no false negatives, no true negatives, and many false positives. The perfectly cleaned data (green) has been cleaned by an anomaly detector with no false negatives or false positives. This suggests that for SNRs above the classifier sensitivity threshold, anomaly detector false positives are in fact beneficial, and provided an anomaly detector returned no false negatives it would result in an accuracy somewhere between the two. As the anomaly detector cleaned classifier accuracy is lower than the perfectly cleaned accuracy for some values above the sensitivity threshold, it suggests that the false negative rate of the anomaly detector is too high.

The anomaly detector has a false negative rate of 2.2%, higher than the $\approx 1\%$ disparity between raw and cleaned data close to the classifier sensitivity threshold. Since the anomaly detector false negative rate is influenced by the pairwise threshold classifier, and the percentile threshold method essentially allows the pairwise false negative rate to be selected, it is expected that a lower percentile would result in a higher classification accuracy for noise above the sensitivity threshold.

Figure 6.5b and Table 6.2 give performance for the best found percentile threshold of 0.0005. Performance is higher across all noise amounts, and inclusion of the anomaly detector does not reduce classifier accuracy compared to uncleaned data at any point. This ensures that there is no detriment to performance regardless of how much corruption is present, including when there is no corruption at all.

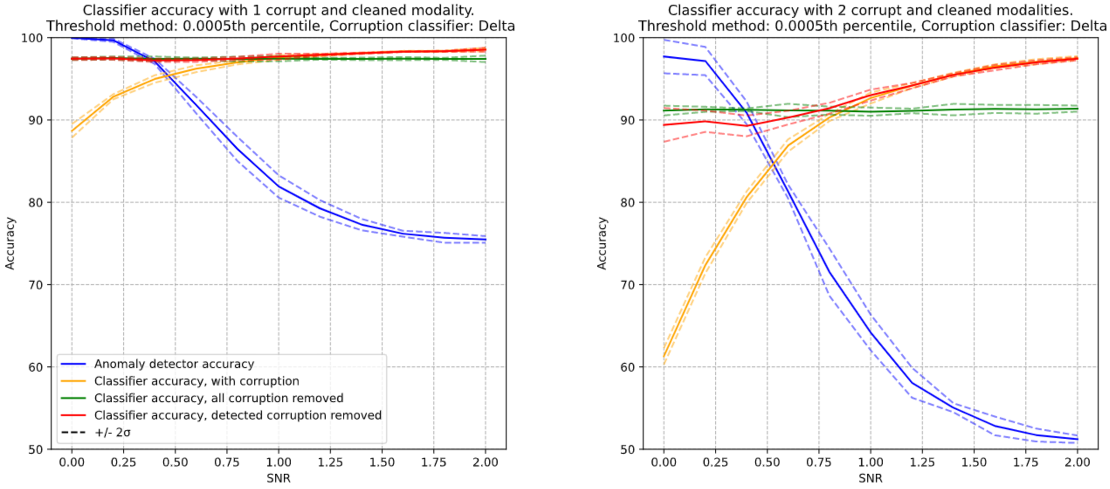
Close to the sensitivity threshold, performance using the anomaly detector is higher than both the cleaned and uncleaned data. At this level of noise some modalities are corrupt enough to reduce performance while others still contain enough information to be useful for classification, and these results show that the anomaly detector can be precise enough to differentiate between them. When the amount of noise is high it should be possible to obtain performance comparable to that of the perfectly cleaned data, representing an accuracy increase of up to 0.7%. However, tuning performance at this end tends to increase the number of false negatives and be detrimental to performance for lower noise amounts.

Note that accuracy of the anomaly detector itself has fallen over most of the noise range as the intention is now to separate modalities based on their utility for classification, rather than whether noise was added. These results are averaged over 0, 1 and 2 corrupt modalities, making 75% of modalities clean, so the accuracy of the anomaly detector as it allows more corrupt modalities through tends towards 75%.

Table 6.2: Anomaly detector classifier performance compared to performance on uncleaned and perfectly cleaned data. Threshold method: 0.0005th percentile, Classifier: Delta, Noise: All. Full table in appendix B

	SNR										
	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
Best accuracy	95.2	95.4	95.1	95.5	95.9	96.5	97.0	97.5	97.8	98.1	98.2
Δ vs. corrupt	12.3	7.4	3.6	1.5	0.5	0.2	0.0	0.0	0.0	0.0	0.0
Δ vs. clean	-0.5	-0.5	-0.7	-0.4	0.1	0.8	1.2	1.6	2.0	2.2	2.4

The 0.0005th percentile threshold used represents the extreme lower end of the correlation distribution and gives very little chance of false negatives. It is possible that a different threshold method that produces a lower threshold could increase accuracy even more. However, different modalities and modality pairs can have differently placed and shaped distributions, resulting in different thresholds, and the percentile threshold method ensures these differences are taken into account.



(a) Performance of anomaly detector and MM-MNIST classifier with 1 corrupted modality.

(b) Performance of anomaly detector and MM-MNIST classifier with 2 corrupted modalities.

Figure 6.6: Classifier results on corrupted and cleaned data with 1 or 2 corrupt modalities

Figure 6.6 compares performance when 1 or 2 modalities are corrupt. Performance with a single corrupt modality is high across the noise range, with only a slight drop below the accuracy of the fully cleaned data with lower SNR. Predictably, accuracy with 2 corrupt modalities (half the image) is worse, but the performance gains over using the corrupted data are as high as 28.1%, with a minimum accuracy of 89.4%. . Performance with 0 corrupt modalities was the same for all classifications (98.9%), with the anomaly detector achieving an accuracy of 100% due to the low false negative rate.

6.1.5 Single sample anomaly detection

So far experiments have focused on using multiple samples for anomaly detection, such as in a real-time streaming scenario. However, there may be times when sets of samples are not available and corruption must be detected in a single sample. In this case, correlation between all canonical variates can be used instead of correlation across all samples (Section 5.3.1). Figure 6.7 shows the distribution of correlations for a single sample.

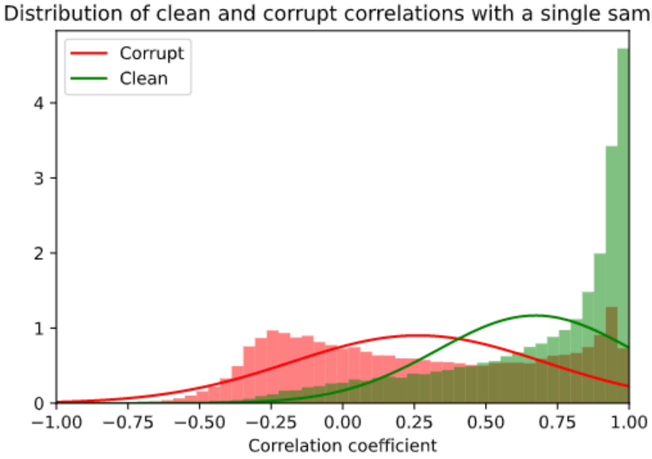
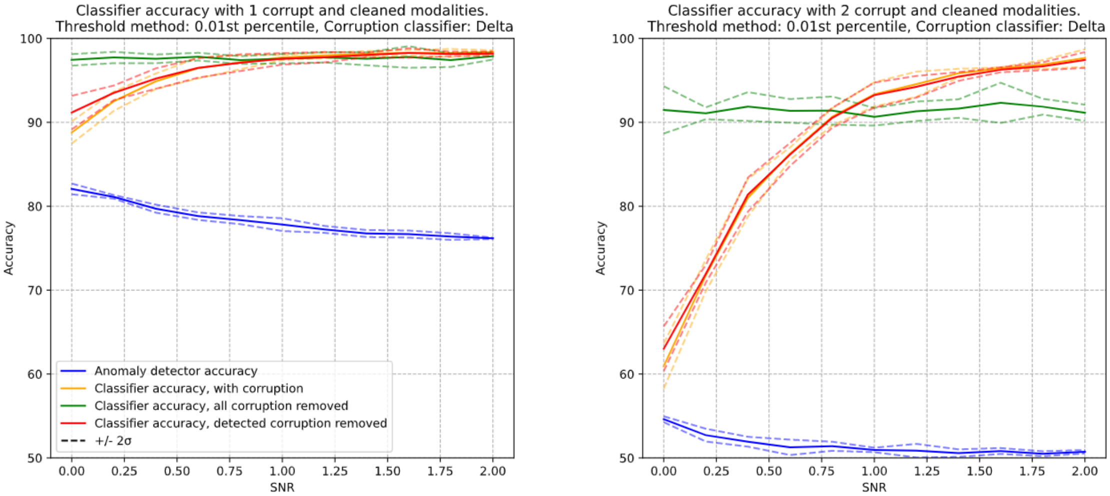


Figure 6.7: Correlation between clean and corrupt modalities with a single sample.

For many samples, the combined correlations approximate a normal distribution in line with the Central Limit Theorem. With a single sample, the distributions have vastly different shapes and are not matched by their normal approximations. There are significant overlaps in the histograms, some clean samples have correlations below 0, and the corrupt distribution is almost uniform between -0.25 and 1, making separation more difficult. To ensure the thresholds are still placed based upon the clean distribution the percentile method is used, but with larger percentiles.

Figure 6.8 shows results using the best found percentile threshold of 0.01. Performance in general is poor, with a slight increase in performance over the corrupt data when the amount of noise is very high. At this threshold level there is generally no accuracy reduction for lower amounts of noise. Other thresholds gave significant performance reductions across the noise range due to large numbers of pairwise misclassifications.



(a) Performance of anomaly detector and MM-MNIST classifier with 1 corrupted modality using a single sample.

(b) Performance of anomaly detector and MM-MNIST classifier with 2 corrupted modalities using a single sample.

Figure 6.8: Classifier results on corrupted and cleaned data using a single sample

Though results are poor, they are not detrimental to performance, and further tuning of the anomaly detection pipeline to the single sample case could overcome these limitations. Using a dataset with more modalities could smooth out the effect of pairwise errors during the modality corruption detection stage.

7 Conclusion

Whilst existing methods increased model robustness to moderate amounts of noise or missing data, performance still suffered under high amounts of noise. Integration of the proposed anomaly detection pipeline into the MM-MNIST multimodal classifier produces very good results, increasing accuracy by an average of 12.3% with large amounts of corruption, only 0.7% lower than a theoretically perfect anomaly detector. Accuracy is not reduced when levels of corruption are low, so the pipeline can generally be used with no detriment to classifier accuracy.

Previous efforts using CCA have been successful in detecting single corrupt modalities [13], but that has limited real world utility as it requires existing knowledge of how many modalities are corrupt. The proposed system improves upon this by being able to detect and localise corruption in up to $m - 2$ modalities simultaneously, or indicate whether more modalities than that are corrupt.

Initial goals were to maximise model performance in the presence of missing and corrupt modalities, and as an intermediate step, detect any corruption in input modalities, ideally on a per-sample basis. The anomaly detector does a good job of detecting corruption in input modalities, though performance does suffer on single samples. Model performance was vastly improved over classification using corrupted data when the anomaly detector was integrated into the classifier.

These performance improvements used a relatively small sample size of 30, and it is likely high performance could be achieved using smaller sample sizes. Though dependent on the data in use, in many cases using 30 samples gives fast enough response times be considered real-time.

7.1 Future work

This project has identified a number of directions for future exploration.

- Tuning the pipeline to perform well on small and single sample sizes could increase its utility in real-time and non temporal scenarios.

- Applying the pipeline to different datasets of different types, such as MM-Fit [14] as used in [13] [1].
- Jointly training the classifier network heads with the classification and GCCA loss functions, so that GCCA can be carried out on the intermediate representations without using further deep networks.
- Applying DGCCA to the raw data, rather than an intermediate representation.
- Attempting to reconstruct missing or removed modalities, rather than simply zeroing them out.

[1] Although significant research and implementation was carried out on the MM-Fit dataset, the results were not complete enough by the project deadline to make it into this dissertation.

References

[1] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski, "Integration of acoustic and visual speech signals using neural networks," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 65–71, 1989.

[2] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia systems*, vol. 16, no. 6, pp. 345–379, 2010.

[3] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic, "Avec 2011–the first international audio/visual emotion challenge," in *Affective Computing and Intelligent Interaction* (S. D'Mello, A. Graesser, B. Schuller, and J.-C. Martin, eds.), (Berlin, Heidelberg), pp. 415–424, Springer Berlin Heidelberg, 2011.

[4] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982.

[5] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] V. Radu, C. Tong, S. Bhattacharya, N. Lane, C. Mascolo, M. Marina, and F. Kawsar, "Multimodal deep learning for activity and context recognition," *Proceedings of ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 4, pp. 1–27, 2018.

[8] F. Li, N. Neverova, C. Wolf, and G. Taylor, "Modout: Learning multi-modal architectures by stochastic regularization," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 422–429, IEEE, 2017.

[9] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[10] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *International conference on machine learning*, pp. 1247–1255, PMLR, 2013.

[11] P. Horst, *Generalized canonical correlations and their application to experimental data.* No. 14, Journal of clinical psychology, 1961.

[12] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang, and R. Arora, "Deep generalized canonical correlation analysis," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pp. 1–6, 2019.

[13] S. P. Jayakumar, "Robust multimodal deep learning for human activity recognition," Master's thesis, University of Edinburgh, 2020.

[14] D. Strömbäck, S. Huang, and V. Radu, "Mm-fit: Multimodal deep learning for automatic exercise logging across sensing devices," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, December 2020.

[15] Z. Lipton, Y.-X. Wang, and A. Smola, "Detecting and correcting for label shift with black box predictors," in *International Conference on Machine Learning*, pp. 3122–3130, 2018.

[16] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," in *Advances in Neural Information Processing Systems*, pp. 1396–1408, 2019.

[17] L. Tran, X. Liu, J. Zhou, and R. Jin, "Missing modalities imputation via cascaded residual autoencoder," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1405–1414, 2017.

[18] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2949–2980, 2014.

[19] P. Smolensky, "The mathematical role of self-consistency in parallel computation," in *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, pp. 319–325, 1984.

[20] B. Bischke, P. Helber, F. König, D. Borth, and A. Dengel, "Overcoming missing and incomplete modalities with generative adversarial networks for building footprint segmentation," *CoRR*, vol. abs/1808.03195, 2018.

[21] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1692–1706, 2015.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

A Relationship between k and c

Let n be the number of modalities, c be the number of corrupt modalities, and k be the proportion of pairs that contain corruption.
n modalities give a triangle of pairs with height $n-1$, so the total number of pairs is given by

$$\frac{1}{2}(n-1)n$$

The first corrupted modality corrupts an entire row of pairs, reducing clean pairs by $n-1$. The second corrupted modality corrupts the next row, bar the already corrupted pair, so reduces the number of clean pairs by $n-2$, and so on.
Corrupting c modalities removes the c largest rows from the triangle, leaving a triangle of height $n-1-c$, and reducing the number of clean pairs to

$$\frac{1}{2}(n-1-c)(n-c)$$

The proportion of corrupt pairs is given by

$$k = 1 - \frac{\textit{clean pairs}}{\textit{total pairs}} = 1 - \frac{(n-1-c)(n-c)}{n(n-1)}$$

During anomaly detection we know n, and k can be calculated directly from the pairwise correlations. Rearranging to calculate c gives

$$c^2 + (1-2n)c + n^2 - n((n-1)(1-x)+1) = 0$$

for which the roots can easily be computed.

B Full pipeline results

Table B.1: Anomaly detector, classifier, and pipeline accuracy. Threshold method: 0.0005th percentile, Classifier: Delta, Noise: All.

	SNR										
	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
0 Corrupt											
Detector	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Corrupt	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9
Cleaned	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9
Pipeline	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9	98.9
1 Corrupt											
Detector	100.0	99.7	97.1	91.8	86.5	81.9	79.2	77.3	76.2	75.7	75.5
Corrupt	88.7	92.8	95.0	96.2	97.0	97.6	97.9	98.1	98.3	98.4	98.5
Cleaned	97.4	97.5	97.4	97.4	97.4	97.5	97.4	97.4	97.5	97.4	97.4
Pipeline	97.4	97.5	97.2	97.3	97.5	97.7	97.9	98.1	98.3	98.3	98.5
2 Corrupt											
Detector	97.7	97.2	90.9	81.3	71.5	64.2	58.1	55.0	52.8	51.7	51.2
Corrupt	61.3	72.3	80.6	86.9	90.3	92.7	94.2	95.5	96.5	97.1	97.5
Cleaned	91.1	91.3	91.2	91.2	91.1	91.0	91.1	91.3	91.3	91.3	91.4
Pipeline	89.4	89.8	89.3	90.3	91.4	93.0	94.2	95.5	96.4	97.0	97.4
Average											
Detector	99.2	98.9	96.0	91.1	86.0	82.0	79.1	77.4	76.3	75.8	75.6
Corrupt	82.9	88.0	91.5	94.0	95.4	96.4	97.0	97.5	97.9	98.1	98.3
Cleaned	95.8	95.9	95.8	95.8	95.8	95.8	95.8	95.8	95.9	95.9	95.9
Pipeline	95.2	95.4	95.1	95.5	95.9	96.5	97.0	97.5	97.8	98.1	98.3