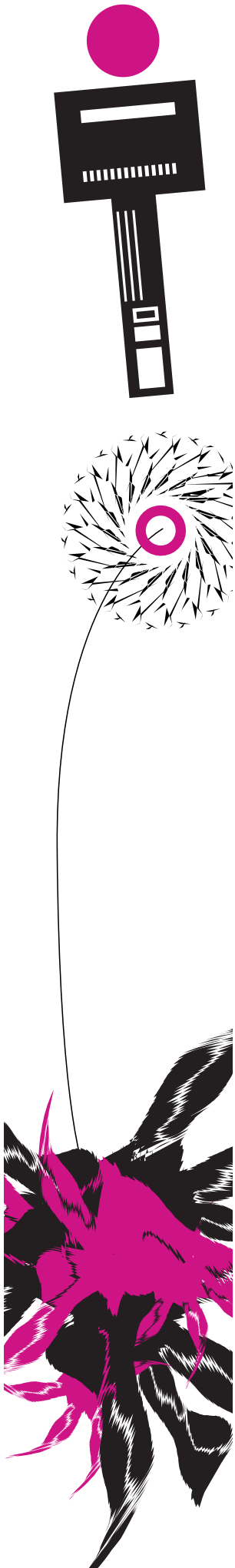

RAM ROBOTICS AND MECHATRONICS

Convolutional autoencoders to process 4D gynaecological data

M.T. (Merijn) Hofsteenge

MSC ASSIGNMENT

Committee:
prof. dr. ir. C.H. Slump
F. van Limbeek-van den Noort, MSc
dr. J.M. Wolterink

August, 2020

041RaM2020
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Acknowledgements

First of all, I would like to thank Frieda. She guided me well throughout my thesis, and was always quickly available for help. Her positivity and friendliness also helped me. Next, I would like to thank Kees for being the chair of my graduation committee and the main supervisor of my thesis. Furthermore, I would like to thank Jelmer Wolterink for being part of my graduation committee. Lastly, I would like to thank my friends and family for the support they provided me throughout my thesis.

Acronyms

3D-AE 3 Dimensional Autoencoder. 4, 5, 6, 7, 8, 10, 14, 16

CAE Convolutional Autoencoder. 1, 2, 3, 5, 6, 7, 8, 10, 16

CNN Convolutional Neural Network. 2, 4, 5, 6, 7, 15, 16

GMM Gaussian Mixture Modeling. 3, 5, 6, 7

GPU Graphics Processing Unit. 4, 5, 8

LAM Levator Ani Muscle group. 1, 2, 3, 4, 7, 8, 16, 17

MRI Magnetic Resonance Imaging. 1, 2, 3

MSE Mean Squared Error. 4, 5, 6, 10, 15, 16

OAE Orthogonal Autoencoder. 4, 7, 10

PCA Principle Component Analysis. 3, 5, 6, 7, 11, 12, 13, 14, 15, 16

ReLU Rectified Linear Unit. 4, 5, 10

RNN Recurrent Neural Network. 8, 15, 17

t-SNE t-distributed Stochastic Neighbor Embedding. 3, 5, 6

Contents

Acknowledgements **ii**

Acronyms **iii**

1 Introduction **1**

2 Article **1**

3 Additional content **11**
3.1 PCA on latent feature videos 11
3.2 Evaluation metrics 14
3.2.1 Rest, contraction and valsalva 15
3.2.2 Pelvic organ prolapse 16

4 Conclusion **16**

5 References **17**

1 Introduction

Deep learning is a hot topic among researchers. The amount of publications on deep learning in the ScienceDirect database grew from 6837 in 2006 to 16288 in 2016, and the number of publications in the Springer database grew from 39 to 706 in the same period [1]. Deep learning can be split into three groups: supervised, unsupervised and semi-supervised, the latter being a combination of the other two. For supervised deep learning, data sets require labeling of the data points. The labeling process costs time, and thus labels are not always available. Unsupervised deep learning is done without any labels. A key reason for the success of unsupervised learning is that it can be applied on any specific domain or data set where annotations are not always available in quantity [2].

The medical field is one of the only sources of volumetric 3D data, which is obtained through e.g. ultrasound or MRI. For this data type, large annotated data sets are not readily available. The available data sets are often not annotated or only annotated for a specific region of interest, e.g. a specific organ [3]. This can be attributed to the long time it takes to label the data, and the limited time of doctors.

In this research, we will attempt to apply unsupervised deep learning to 4D ultrasound scans of the pelvic floor. The scans were made in the GynIUS project, a collaboration between UMC Utrecht, University of Twente and Radboudumc. The aim of GynIUS is to get functional information of the Levator Ani Muscle group (LAM), which is critical in pelvic floor (dys)function. Because the LAM is studied, the ultrasound videos show a patient performing a certain maneuver. This is either contraction, which contracts the LAM, or valsalva, which stretches the LAM. The data set is hard to interpret, mostly because the data is very large. We will attempt to gain more knowledge on this data set through unsupervised methods.

The unsupervised deep learning method we chose is the Convolutional Autoencoder (CAE) [4]. Through convolution, CAEs can preserve spatial information in data on multiple dimensions at once. Therefore CAEs can work well for unsupervised learning of multidimensional data.

The core of this thesis is an article. An article was chosen because of the success of the CAE in determining the maneuvers performed in the ultrasound videos. This section serves as an introduction to the project, where the article presents the work that was done. Section 3 contains additional content, which was outside of the scope of the article.

2 Article

The article is presented on the next page.

Convolutional autoencoders to process 4D gynaecological data

M.T. Hofsteenge, University of Twente

***Abstract*—Unsupervised deep learning is a great way to gain more understanding on data sets. We applied unsupervised methods to gain insight on 4D ultrasound data of the pelvic floor. We reduced the dimensionality of the 3 physical dimensions of the ultrasound with a convolutional autoencoder, in an unsupervised manner. This reduced the data from 4D to 2D, maintaining the time dimension. Every ultrasound shows a patient performing a maneuver, which is either contraction or valsalva. These are thought to be prevalent features in the ultrasounds. Using the dimensionality reduced data, we successfully classified the maneuver performed in the ultrasound, with supervised and unsupervised methods. The supervised classification resulted in 80-95% accuracy, and unsupervised in 75-90% accuracy. This demonstrates that useful data representations can be found in very large data by using an unsupervised convolutional autoencoder for dimensionality reduction.**

I. Introduction

Deep learning has become a major field of scientific study in recent years. Its exponential growth started with the deep learning network AlexNet [1] winning the ImageNet competition in 2012 by a large margin. Deep learning was facilitated by advancements in computational power and the availability of large annotated data sets. It has been beneficial for many different applications, from self driving cars [2] to the medical field [3].

The medical field now widely applies deep learning in various specializations. In particular, Convolutional Neural Networks (CNNs) have had the greatest impact [4]. In medical imaging, end-to-end trained CNNs are often integrated into existing image analysis pipelines and replace traditional handcrafted machine learning methods [5].

Volumetric 3D data is common in the medical field, but not seen much elsewhere. It is obtained by e.g. MRI or ultrasound scans. Due to the uniqueness of this type of data, the intrusive nature of these medical scans, and the lack of time of doctors to properly label scans, large annotated data sets are not readily available for 3D volumetric data [6]. The available data sets are often not annotated or only annotated for a specific region of interest, e.g. a specific organ.

Many different methods have been developed for using 3D data. 3D data can be processed in a 2D slice-by-slice manner, but that does not take full advantage of the spatial information in the 3D scan [7]. More and more methods that do use the spatial information are being developed. Some researches use the full volumetric data in a convolutional model [7], [8], while others split the volumetric data into 2D slices along the 3 principle axes and tackle the data that way [9], [10].

The goal with medical 3D models is often segmentation or automatic detection of abnormalities, which can support decision making. This is typically done by using supervised learning. Another branch of deep learning is unsupervised learning. Roughly speaking, unsupervised learning involves observing several examples of a random vector x and attempting to implicitly or explicitly learn the probability distribution $\rho(x)$, or some interesting properties of that distribution [11]. Unsupervised learning has some major benefits over supervised [12]. No labeling is required, which saves time of experts. Also, because it does not have a strictly defined task, it may find interesting patterns above and beyond what we initially were looking for [13]. This allows finding patterns that humans cannot detect. Lastly, unsupervised models are generally more scalable and more easily applied to other problems.

A popular form of unsupervised deep learning is the autoencoder. The idea of the autoencoder dates back to 1986 [14]. An autoencoder uses an encoder and sequentially a decoder. The encoder reduces the dimensionality of the data to its latent features. The decoder attempts to closely resemble the original input using only these latent features. The better the decoding is, the more probable it is that the latent features contain relevant information of the input. One of the benefits is, that if the amount of latent features is much smaller than the input data, subsequent models on the data can be trained much faster and have low computer specification requirements. In 2011 the Convolutional Autoencoder (CAE) was presented [15]. By using convolutional layers, a CAE can preserve information on data proximity in multiple dimensions at once. This makes it favorable for higher dimensional data. 3D CAEs are already being used in the medical field with good results, e.g. for segmentation purposes [16], [17] or disease prediction [18].

Here, we will use a 3D CAE to study pelvic floor problems of women. Many women suffer trauma to the pelvic floor after childbirth. It is not well understood why some women experience pelvic floor problems after delivery while others do not [19]. An important muscle group that is associated with these defects, is the Levator Ani Muscle group (LAM). The LAM encircles the rectum, urethra and vagina. Dietz and Lanzarone [20] found that approximately a third of women have traumas in the LAM after vaginal childbirth. DeLancey *et al.* [21] showed that women with pelvic organ prolapse were much more likely to have such trauma than healthy women. Azpiroz *et al.* [22] found that around 60% of patients with fecal incontinence had impaired contractive strength in the LAM.

Our goal is to gain more knowledge on 4D ultrasound scans of the pelvic floor, and to find easier ways of interpreting the data it contains. Our data set was created to gain functional

information on the LAM. Interpretation of this data is difficult, mostly because the scans are very large. We will attempt to find relevant information on this data using a 3D CAE, as it can take full advantage of the data in the scan through convolution. A 4D CAE is unfeasible, because the model would be too large to fit in computer memory, so the time dimension is omitted at first.

Since the scans were made for investigating the LAM, the patients perform a maneuver that activates the muscles. This is either a vaginal contraction, or the valsalva maneuver. These maneuvers are prevalent in the ultrasound videos, so we will investigate if we can classify them using CAEs. This determination of maneuver is not medically relevant, but is used as a proof of concept for this type of unsupervised data processing, since it is one of the most noticeable features for a human observer. The evaluation of the maneuvers is done in a two-step process. First, we build a 3D CAE to reduce each separate frame of the video to its latent features, in a completely unsupervised manner. The latent space should then contain the most prevalent features of the video frames. In the second step we use multiple methods, supervised and unsupervised, on these latent feature videos to determine the maneuver performed in it. Our main focus is attempting to do this in an entirely unsupervised manner.

II. Background

An autoencoder is a type of neural network that attempts to recreate its input. A perfectly mapping autoencoder is essentially the identity function. It is made with two sequential neural networks, an encoder and a decoder. The input gets encoded to a latent space using the encoder, and this latent space is used to reconstruct the input via the decoder. By applying constraints on this latent space, such as lowering the dimension relative to the input, the network is forced to learn structure in the data in order to reconstruct the input. The encoder and decoder get trained at the same time through backpropagation. The goal of training an autoencoder is to minimize its loss function,

$$L(x, z) = L(x, g(y)) = L(x, g(f(x))) \tag{1}$$

Here x is the input data, z is the autoencoder output, y is the latent space representation, f is the encoder and g is the decoder. The encoder and decoder typically have a similar, but reversed structure. In this research, we use a combination of convolutional and fully connected layers for both the encoder and decoder, mapping the 3D input to a 1D latent space.

CAEs are already applied in the medical field, but they typically include an additional term to the loss function. This term is tailored specifically for the goal of the CAE and contains a label on the data. The additional term uses just the latent features, but is trained at the same time as the autoencoder. Zeune *et al.* [23] used a 2D CAE which classifies tumour cells. Basu *et al.* [18] used a 3D CAE on MRI images to make predictions about the progression of Alzheimers disease, using the disease labels that were obtained 6 months after the MRI. Myronenko [16] used a 3D CAE that essentially has two decoders, one for reconstructing the MRI of a brain, and one for reconstructing a segmentation of a tumour cell in the brain. These are all supervised learning methods, as additional information on the data was introduced.

The key difference between supervised and unsupervised learning is that unsupervised requires no labels for training. An autoencoder can be unsupervised because it aims to recreate its own input, and therefore requires no labels. A key reason for the success of unsupervised learning is that it can be applied on any specific domain or data set where annotations are not always available in quantity [24]. We will focus on unsupervised learning for dimensionality reduction and clustering, which are two large fields in it. To the best of our knowledge, 3D CAEs have not been used for dimensionality reduction in a purely unsupervised manner on volumetric data.

Dimensionality reduction facilitates, among others, classification, visualization, and compression of high-dimensional data [25]. In our 3D CAE, we use a latent space much smaller than the input data, to reduce the dimensionality of the 4D ultrasound data by a significant amount in each frame. Two other methods of dimensionality reduction are Principle Component Analysis (PCA) [26] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [27]. These are common dimensionality reduction methods, but they unfortunately tend to fail for very large data.

For cluster classification, K-means [28] and Gaussian Mixture Modeling (GMM) [29] are two popular methods. They are both applied in a wide array of clustering tasks. K-means has been used in e.g. finding potential customers for a company [30] or profiling people's internet usage [31]. GMM is used similary, but in some cases outperforms K-means [32]. K-means attempts to put each data point as close as possible to a cluster center. Because of this, K-means is good at detecting spherical clusters of similar density and size, but tends to fail when this is not the case. This is why we also introduced GMM, which is a probabilistic approach to clustering. It employs the expectation-maximization algorithm [33] to maximize the likelihood that data points belong to clusters, and accordingly updates the clusters. These clusters are Gaussian distributions, which do not need to be spherical as in K-means.

III. Methods

The data set used is comprised of 4D ultrasound videos of the pelvic region of women. A 2D slice of one of the videos is presented in Figure 1. The data was collected in the GynIUS project, a collaboration between UMC Utrecht, University of Twente and Radboudumc.

For every patient we use one contraction and one valsalva ultrasound video. Vaginal contraction contracts the LAM, while the valsalva maneuver stretches out the LAM. The valsalva maneuver is performed by closing the mouth and nose, and breathing out. This forces pelvic organ descent. At the start of each video the LAM is in resting position. From around halfway in the video, either contraction or valsalva is performed. There is no label indicating whether the videos are of contraction or valsalva, but we know the intended acquisition order of scans per patient. However, this order can

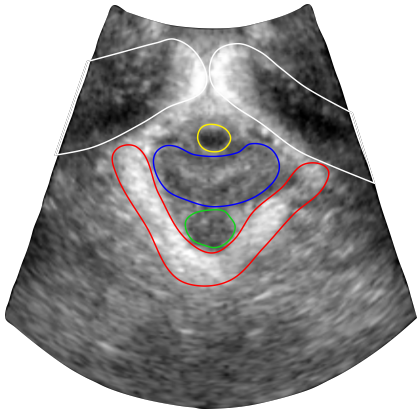

Fig. 1: A 2D slice of one of the ultrasound scans. It shows bone (white), the urethra (yellow), the vagina (blue), the rectum(green) and the LAM (red).

differ in practice. For each patient, the acquisition starts with contraction, so we estimate the first video shows contraction >95% of the time. For valsalva, we took the third video of a patient. This tends to be valsalva, but for clinical reasons there were sometimes more than 2 videos of contraction. Therefore, we estimate valsalva labels to be correct 80-90% of the time.

The videos contain $277 \times 352 \times 229 \times 22$ voxels, which is very large for a single data point. This makes it computationally expensive to train networks on. Computations within deep learning are often done on the GPU, as they allow parallel computations making training much faster. GPUs unfortunately do not have enough memory for us to directly apply convolution on the data. Therefore, we split the data processing in multiple steps.

A) First, the time dimension is not processed yet. We designed a 3D autoencoder, which we will refer to as 3D-AE, to process one video frame at a time and encode it to 128 features. We used the trained 3D-AE to encode the videos into latent feature videos with dimension 22×128. This put the time dimension in again, for further processing.

We processed the latent feature videos in multiple ways.

B) We used it in a supervised manner, by training a CNN on the labels of contraction and valsalva. This led to a semi-supervised result, as the reduction of the 3D-AE is unsupervised, but the CNN is supervised.

C) We classified it in an unsupervised manner, by applying different clustering techniques. This led to truly unsupervised classification results.

B) and *C)* were also applied on just the last 4 frames of the latent feature videos, so on data of size 4×128. We did this because contraction or valsalva should always be visible in the last frames. The usage of just the last frames filters out data that is likely unnecessary.

A. 3D-AE

We trained the 3D-AE on single frames of the ultrasound videos. The videos were resized from $277 \times 352 \times 229 \times 22$ to $192 \times 256 \times 192 \times 22$ by removing voxels from the edges. These removed voxels mostly contained no information, since the data is cubical while the ultrasounds are not. Also, the gel padding needed for the ultrasound scans was discarded.

The structure of the 3D-AE can be seen in Figure 2. The depth of the convolutional layers and the number of latent features is relatively low. This is necessary to prevent overloading of the GPU memory. The memory bottlenecks are the depth of the initial convolutional layer, and the fully connected layer before the latent features. These compete for memory and require a balance. We set this balance to a convolution depth of 8 and 128 latent features. This leads to a dimensionality reduction of factor $\frac{192 \times 256 \times 192}{128} = 73728$.

We normalized the input data from 0 to 1. The output function is linear, but with a minimum of 0 and a maximum of 1. All other activation functions are the swish function [34]. The swish function was used as it has shown minor improvements over the often applied Rectified Linear Unit (ReLU) function. We applied a dropout of 0.5 before the latent space to reduce overfitting. The loss function we used is:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2 + \gamma|1 - \sum_{j=1}^{m} y_j^2| \qquad (2)$$

Here x_i is the input, $\hat{x}_i$ is the decoded reconstruction, n is the number of voxels, m is the number of latent features, y are the latent features and γ is a scaling factor.

The first part of Equation 2 is the reconstruction loss, which is the Mean Squared Error (MSE) loss between input and reconstruction. To ensure that the latent features are nonzero, we introduced the second part. It is inspired by the Orthogonal Autoencoder (OAE) [35], and will be referred to as the OAE loss. It takes the sum of the squares of the latent features and forces it to a constant. This ensures that the latent features can not explode to a high number, and it also makes them nonzero. We chose the constant to be 1, which means the loss is 0 when the latent space vector has length 1.

γ was set to 10^{-3}. This allowed for the latent features to be normalized, while only having a minor effect on the reconstruction. The model uses the Adam optimizer [36] with parameters $\alpha = 3 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-7}$.

B. Supervised

The labels for the CNN are binary, a 0 for a latent feature video showing contraction and a 1 for a latent feature video showing valsalva. The same simple model structure was used for the 22×128 and 4×128 data. The CNN starts with a 2D convolutional layer with depth 8. Sequentially, it applies max-pooling with pool size 2×2 and flattening to a vector. The flattened vector goes into a fully connected layer of size 8, and finally the output is given by a fully connected layer of size 1. The activation functions for all layers are ReLUs, except for the output activation function which is the Sigmoid function. We applied a dropout of 0.5 between the fully connected layers. The loss function is the log loss function [37]. We used the Adam optimizer again, with the same parameters as used for the 3D-AE.

We also used this CNN to find mislabeled data points. We used 5-fold cross validation [38] for this, on the 22×128 data. For the 5 models trained in the cross validation, we compared

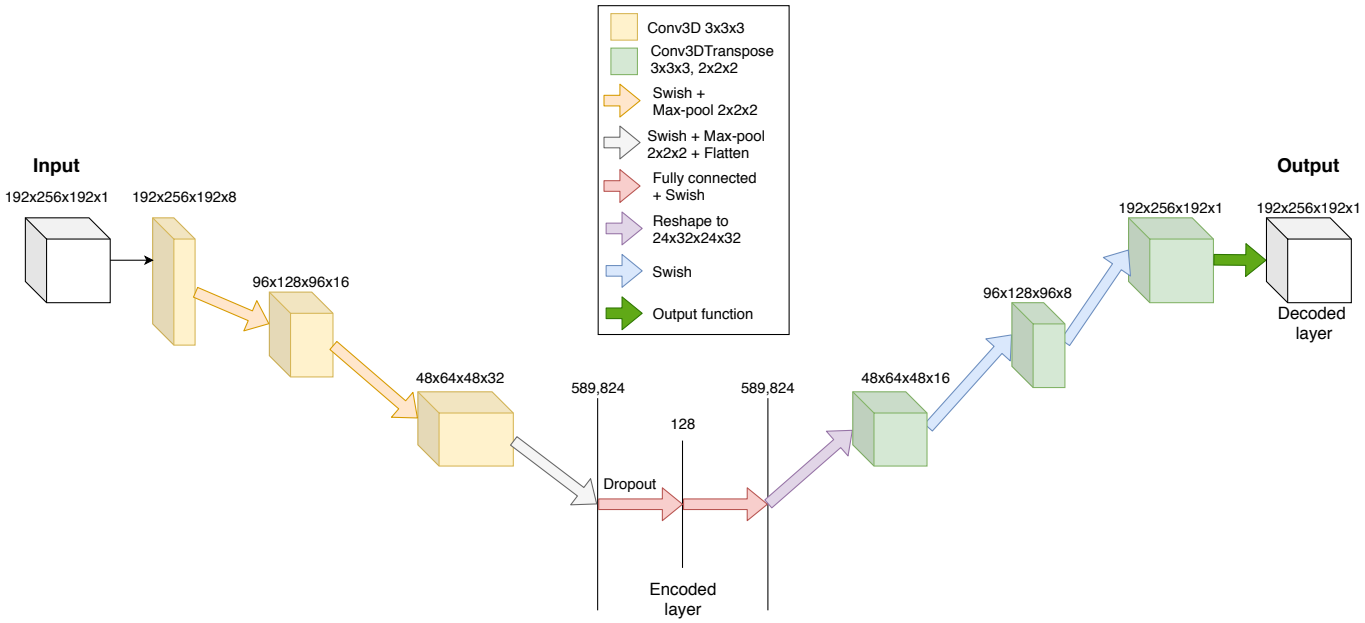

Fig. 2: Model structure of the 3D-AE. Yellow blocks are 3D convolution, green blocks are 3D transpose convolutions.

the classifications with its label. All data points that were misclassified by at least two models were sent to an expert for manual evaluation. We updated the labels according to the manual verification, and retrained the CNN with these labels. These updated labels were also used for the validation of the unsupervised methods.

C. Unsupervised

We used multiple unsupervised methods for data analysis. We classified the data clustering with two algorithms, K-means and GMM. We applied them to the latent feature videos, concatenated into a single vector. We also used three methods that first applied dimensionality reduction to the latent feature videos, down to 2 data points, before cluster classification. This allowed us to also perform a visual inspection of the clustering. These three methods are PCA, t-SNE and a CAE.

The CAE uses two convolutional layers and two densely connected layers for both its encoder and decoder, and has a latent space dimension of 2. Its structure is shown in Appendix A. The CAE has the same structure for both the 22×128 and 4×128 data. All activation functions are ReLUs, but the output has no activation function. The MSE is used as the loss function, and the same Adam optimizer is used again.

For all methods, we compared the unsupervised classification with the true (updated) labels of the data points to get a clustering accuracy. We compared this to the classification ability of the supervised CNN, for a clearer judgment of how well it performs.

D. Data preparation and network training

We used ultrasounds of 292 patients for training the 3D-AE. For every patient, we took one valsalva and one contraction video. From every ultrasound, 4 frames were used. These 4 frames are the first frame, the last frame and two random frames in between. We did not use all frames, because we needed to limit the amount of data for memory management on the external server. We chose the first and last frame, because they are mostly consistent over the data. We also used two random frames in between, such that the 3D-AE does see the complete video, albeit over multiple patients. We then discard two-thirds of the entire data set in a completely random manner, to further reduce its size. This results in a training and validation set of 790 frames, 700 for training and 90 for validation. The mini-batch size is 1, which is the maximum that fits in the GPU memory. 20 epochs were done, which took around 15 hours. The loss was calculated after every 30 frames. After training, we used the model with the lowest reconstruction (MSE) loss.

We encoded ultrasound videos of 200 patients with the 3D-AE, which creates 22×128 latent feature videos. We also took the last 4 frames of each video for the 4×128 data set. We discarded some ultrasound videos, because they had less than 22 frames. This resulted in 379 encoded videos, of which 194 are of contraction and 185 are of valsalva. For the CNN we used a 80-20 training-validation split, resulting in 304 latent feature videos for training and 75 for validation. 17 patients that were unused for the training of the 3D-AE were used to construct a testing set. All testing set videos were manually verified to be either contraction or valsalva by an expert. The testing set contains 29 latent feature videos, since some ultrasounds had less than 22 frames. We trained the CNN for 100 epochs, with mini-batch size 8. After updating the labels according to the outcome of the 5-cross validation, we retrained the model in the same way.

We trained the unsupervised classification on the latent feature videos with the updated labels. The K-means was applied with 10 different random seeds and 300 iterations. The GMM used 100 expectation-maximization iterations. The PCA and t-SNE were both set up to return 2 components. The

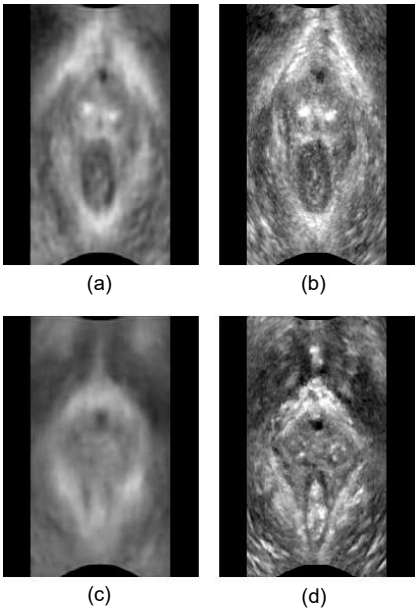

Fig. 3: Slices of the reconstructed 3D images by the 3D-AE. (a) is the reconstruction of training frame (b), and (c) is the reconstruction of validation frame (d). The MSE for (a) is 3.91×10^{-3}, for (c) it is 8.36×10^{-3}.

PCA transformation was fit without testing data, for t-SNE we included the testing data since its transformation is iterative. The t-SNE was set to a perplexity of 15 for the 22×128 data and to 25 for the 4×128 data. We used a learning rate of 200, with 5000 iterations. The CAE used the same training-validation split as the CNN, and was trained for 100 epochs with mini-batch size 16. We converted the training, validation and test data to 2 latent features after training. The K-means and GMM were fit using the validation data, and then used for classification on the testing set.

All models were made using Keras in the TensorFlow 2.1.0 library in Python. The t-SNE, PCA, K-means and GMM were computed using the scikit-learn 0.22.1 library. We used an NVIDIA Titan X for training the 3D-AE, and an NVIDIA Quadro M1200 for all other models.

IV. Results

A. 3D-AE

The 3D-AE has a validation loss of 7.72×10^{-3}. Appendix B shows the loss of the model during training.

Figure 3 shows slices of ultrasound frames and its reconstruction. The training image shows a recognizable image, similar to its input. For the validation it is still recognizable, but less so.

B. Supervised

The accuracy of the trained CNNs can be seen in Table I. We achieved high accuracy for the training, validation and testing set for the 22×128 data. The accuracy is worse for 4×128, especially for the testing set.

TABLE I: CNN accuracy for both latent feature video sizes, on original and updated labels. The highest accuracies are highlighted.

Data set	22×128 accuracy (%)	4×128 accuracy (%)
Training	95.4	91.1
Validation	90.7	90.7
Test	**93.1**	82.8
Updated labels training	**100**	96.7
Updated labels validation	**94.7**	93.3
Updated labels test	86.2	82.8

TABLE II: Validation and testing set accuracy of unsupervised methods on the contraction and valsalva labels. The highest accuracies are highlighted.

Method	22×128 accuracy (%) validation / test	4×128 accuracy (%) validation / test
CNN	94.7 / 86.2	93.3 / 82.8
K-means	83.1 / 79.3	89.7 / **86.2**
PCA + K-means	81.6 / 79.3	89.2 / **86.2**
t-SNE + K-means	72.8 / 68.9	87.0 / **86.2**
CAE + K-means	82.3 / 79.3	81.8 / 79.3
GMM	82.5 / 79.3	87.0 / 62.1
PCA + GMM	**83.4** / 79.3	89.4 / **86.2**
t-SNE + GMM	75.9 / **82.8**	88.0 / **86.2**
CAE + GMM	81.0 / 75.9	**91.0** / **86.2**

The 5-fold cross validation on the 22×128 data returned 32 latent feature videos that were misclassified 2 times or more, 28 valsalva videos and 4 contraction videos. For the valsalva, 20 of these videos were indeed contraction, 6 were improperly performed valsalvas and 2 were normal valsalvas. For the contraction, 1 video was a strangely performed valsalva and the other 3 were poorly performed contractions. We changed the labels for the 20 valsalva videos that showed contraction, and the 1 contraction video showing valsalva.

After the label change, we saw an improved accuracy for the training and validation of both the 22×128 and 4×128. The testing accuracy however decreased for both.

C. Unsupervised

Table II shows the accuracies achieved by the different methods. The CNN accuracy was included, for easy comparison. We achieve decent to good accuracy on the 22×128 data. The 4×128 data however shows highly improved accuracy, with the CAE+GMM method achieving accuracy similar to the CNN. Figure 4 and 5 show 2D plots of the dimensionality reduced data with its labeling. Here the difference between 22×128 and 4×128 data can be seen, where the second clearly has better clustering. This trend of better observable clusters was seen for all dimensionality reduction methods.

V. Discussion

In this study, we explored the effectiveness of using CAEs to interpret 4D ultrasound data. We constructed the 3D-AE to reduce the dimensionality of the data, and sequentially classified the maneuvers performed in the ultrasounds successfully, both with supervised and unsupervised methods.

The ability of discrimination between valsalva and contraction, which is even present in unsupervised classification, indicates that the 3D-AE effectively reduced the dimensionality

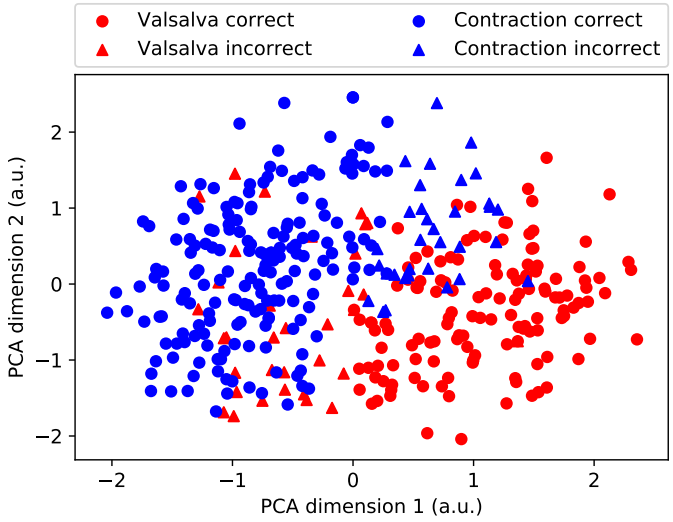

Fig. 4: Latent feature videos (22×128) reduced to 2 dimensions by PCA and classified with K-means. The colors (red and blue) indicate the true labels, the shape (circle and triangle) indicates whether K-means predicts the same.

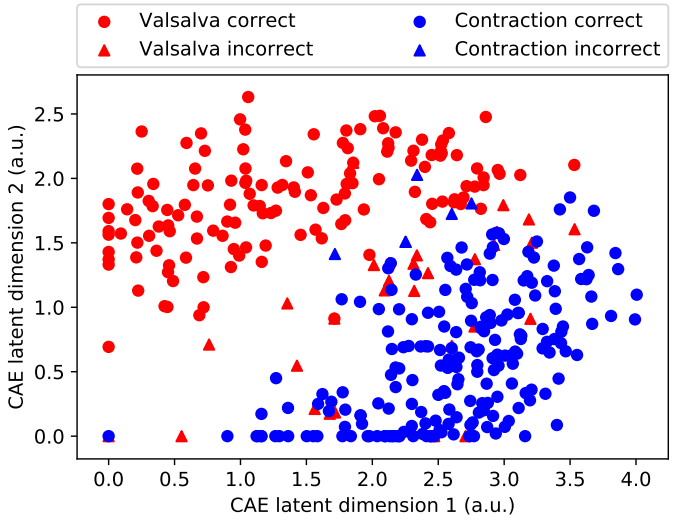

Fig. 5: Last 4 latent video frames (4×128) reduced to 2 dimensions by a CAE and classified with GMM. The colors (red and blue) indicate the true labels, the shape (circle and triangle) indicates whether GMM predicts the same.

of the ultrasounds. Both supervised and unsupervised methods show excellent classification. The supervised methods show an accuracy of 80-95%, while unsupervised methods show 75-90% accuracy. The unsupervised accuracy significantly improves on the 4×128 data, where the frames showing the resting position are discarded.

We constructed the 3D-AE in a fully unsupervised way, as opposed to other researches using CAEs [16], [18], [23]. With just the OAE loss, which is unsupervised, the information about the maneuvers was successfully extracted from the latent features. This is a great result, as it proves that even without labeling or a data set specific loss function, useful information can still be extracted.

An improvement with minimal supervision could be to introduce the number of clusters to the model. This can be used to assign clusters to the latent features during training of the 3D-AE, using Kullback-Leibler divergence, as done by Xie *et al.* [39] and Guo *et al.* [40]. The goal of this study was however not to obtain optimal clustering on valsalva and contraction, but to find out if a CAE would be able to classify some of the most prevalent features in the data in an entirely unsupervised manner, as a proof of concept.

A problem with the data set is that some patients are unable to properly perform contraction or valsalva, either psychologically or physically. Psychologically, not all patients are used to performing these maneuvers, which can cause problems in making scans that properly show muscle movement. A physical problem can be that through damage in the LAM, patients can be unable to properly perform contraction. Damage to the LAM should have less impact on the valsalva maneuver, since the LAM is not actively involved in valsalva. The inability to perform maneuvers can lead to predictions that do not follow the labels, which means a 100% accuracy is not thought to be achievable on this data set.

The CNN used to classify the maneuver was kept simple, since our emphasis was to obtain proper dimensionality reduction with the 3D-AE. The simple structure still identified the maneuvers very well, achieving a highest test set accuracy of 93.1%. It also allowed for finding wrong data points with the 5-fold cross validation, showing how the CNN even learned from the latent feature videos when mislabeled data was present.

The unsupervised methods allowed for decent to good classification on the 22×128 data, and excellent classification on the 4×128 data. With PCA and K-means, a validation and testing accuracy of 89.2% and 86.2% can be achieved. This is an impressive result, as this classification was done in a fully unsupervised way starting from the ultrasound data, without introducing a clustering incentive to the 3D-AE. With the CAE and GMM the highest validation accuracy was achieved. This validation however introduced some bias, as the resulting clustering was manually observed. The testing accuracy is still 86.2%, which makes it in line with the other best methods.

We intentionally did not use state-of-the-art clustering and dimensionality reduction methods, such as Invariant Information Clustering [41] or Feature Selection method for Balanced Clustering [42]. This was done to demonstrate that the real power comes from the 3D-AE, and not a powerful clustering technique.

For all but three methods, the testing set showed an accuracy of at least 79.3%. The testing set was unfortunately small, as it required manual annotation. All methods, supervised and unsupervised, misclassified 2 valsalva points consistently. These were 2 out of the 3 valsalva points in the testing set that had annotations that the patient could not perform valsalva properly. The other misclassifications were also relatively consistent throughout the different methods. This indicates that the unsupervised methods find a similar split in the data as the supervised methods, without receiving a label on the data. Overall, we think the testing classification is good, but future

work could include a larger testing set. More annotation does require time of experts.

One of the major constraints in designing the 3D-AE was the GPU memory. This left little room for the structure of the 3D-AE. Myronenko [16] used an NVIDIA Volta V100 32GB GPU while using 3D data, to have more GPU memory to fit their model in. We used an NVIDIA Titan X which has a memory of 12GB. Using a GPU with more memory should improve the obtained results, by increasing the depth of the initial convolution. Deeper convolution allows the model to process more local features.

The distinction between valsalva and contraction is unfortunately not a very relevant result, since the patients are explicitly asked to perform a certain maneuver in the making of the ultrasounds. Its current application could be ordering the data set, sorting it by contraction and valsalva, as this was not properly documented.

We believe the 3D-AE structure shows promising results for dimensionality reduction of very large data. It does need to be studied more before it can be applied in a relevant manner. Applications of the 3D-AE will probably remain in the medical field, as that is the common source of volumetric 3D data.

We only showed that the most prevalent features can be found using the 3D-AE. Further research has to be done to see if underlying features can also be processed. This will most likely involve introducing labels after the 3D-AE, since our unsupervised methods distinguish the most prevalent features. This can be seen in the difference in clustering accuracy between using the 22×128 data and the 4×128 data.

A future work that could return medically relevant information, is constructing a Recurrent Neural Network (RNN) on the latent feature videos. Currently, we can not distinguish contraction from the resting position. With a RNN, changes in the latent features between frames can be observed. This could give a precise indication of when the patient starts performing the maneuver, and also how well this maneuver is performed. If a patient shows proper valsalva performance and poor contraction performance, it could be an indication that the LAM is damaged in such a way that contraction is no longer possible. This would be medically relevant, and a RNN combined with the 3D-AE could automate this process.

In conclusion, we have constructed a 3D CAE to apply unsupervised dimensionality reduction on 4D ultrasounds. The dimensionality reduced data was used to classify the maneuver performed in the ultrasounds with excellent accuracy, both with supervised and unsupervised methods. This shows that useful information can be extracted from 4D ultrasounds in an unsupervised manner by using a 3D CAE.

VI. References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Neural Information Processing Systems* 25 (2012).

[2] Q. Rao and J. Frtunikj. "Deep learning for self-driving cars". In: *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems - SEFAIS '18*. New York, New York, USA: ACM Press, 2018, pp. 35–38.

[3] A. Esteva et al. "A guide to deep learning in healthcare". In: *Nature Medicine* 25.1 (Jan. 2019), pp. 24–29.

[4] D. Ravi et al. "Deep Learning for Health Informatics". In: *IEEE Journal of Biomedical and Health Informatics* 21.1 (Jan. 2017), pp. 4–21.

[5] G. Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (Dec. 2017), pp. 60–88.

[6] S. Chen, K. Ma, and Y. Zheng. "Med3D: Transfer Learning for 3D Medical Image Analysis". In: *arXiv e-prints* (Apr. 2019), arXiv:1904.00625.

[7] Q. Dou et al. "3D deeply supervised network for automated segmentation of volumetric medical images". In: *Medical Image Analysis* 41 (Oct. 2017), pp. 40–54.

[8] Ö. Çiçek et al. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Cham: Springer International Publishing, 2016, pp. 424–432.

[9] R. Indraswari et al. "Multi-projection deep learning network for segmentation of 3D medical images". In: *Pattern Recognition Letters* 125 (July 2019), pp. 791–797.

[10] P. Moeskops et al. "Deep Learning for Multi-task Medical Image Segmentation in Multiple Modalities". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Cham: Springer International Publishing, 2016, pp. 478–486.

[11] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[12] K. Raza and N. K. Singh. "A Tour of Unsupervised Deep Learning for Medical Image Analysis". In: *arXiv e-prints* (Dec. 2018), arXiv:1812.07715.

[13] A. A. Patel. *Hands-On Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data*. O'Reilly Media, 2019.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning internal representations by error propagation". In: 1986.

[15] J. Masci et al. "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction". In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 52–59.

[16] A. Myronenko. "3D MRI brain tumor segmentation using autoencoder regularization". In: *arXiv e-prints* (2018), arXiv:1810.11654.

[17] Y. He et al. "DPA-DenseBiasNet: Semi-supervised 3D Fine Renal Artery Segmentation with Dense Biased Network and Deep Priori Anatomy". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Cham: Springer International Publishing, 2019, pp. 139–147.

[18] S. Basu et al. “Early Prediction of Alzheimer’s Disease Progression Using Variational Autoencoders”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Cham: Springer International Publishing, 2019, pp. 205–213.

[19] F. Van Den Noort et al. “Automatic segmentation of puborectalis muscle on three-dimensional transperineal ultrasound”. In: *Ultrasound Obstet Gynecol* 52 (2018), pp. 97–102.

[20] H. P. Dietz and V. Lanzarone. “Levator trauma after vaginal delivery”. In: *Obstetrics and Gynecology* 106.4 (Oct. 2005), pp. 707–712.

[21] J. O. L. DeLancey et al. “Comparison of Levator Ani Muscle Defects and Function in Women With and Without Pelvic Organ Prolapse”. In: *Obstetrics & Gynecology* 109.2, Part 1 (Feb. 2007), pp. 295–302.

[22] F. Azpiroz et al. “The puborectalis muscle”. In: *Neurogastroenterology and Motility* 17.s1 (June 2005), pp. 68–72.

[23] L. L. Zeune et al. “Deep learning of circulating tumour cells”. In: *Nature Machine Intelligence* 2.2 (Feb. 2020), pp. 124–133.

[24] M. Caron et al. “Deep Clustering for Unsupervised Learning of Visual Features”. In: *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018, pp. 139–156.

[25] L. van der Maaten, E. Postma, and H. Herik. “Dimensionality Reduction: A Comparative Review”. In: *Journal of Machine Learning Research - JMLR* 10 (2009), pp. 66–71.

[26] J. E. Jackson. *A User’s Guide to Principal Components*. Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc., Mar. 1991.

[27] L. Van Der Maaten and G. Hinton. *Visualizing Data using t-SNE*. Tech. rep. Nov. 2008, pp. 2579–2605.

[28] J. Macqueen. “Some methods for classification and analysis of multivariate observations”. In: *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[29] M. P. Deisenroth, A. A. Faisal, and C. S. Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020, pp. 348–369.

[30] P. Anitha and Malini M. Patil. “RFM model for customer purchase behavior using K-Means algorithm”. In: *Journal of King Saud University - Computer and Information Sciences* (Dec. 2020).

[31] M. Zulfadhilah, Y. Prayudi, and I. Riadi. “Cyber Profiling Using Log Analysis And K-Means Clustering”. In: *International Journal of Advanced Computer Science and Applications* 7.7 (2016).

[32] Z. Wang et al. “Comparison of K-means and GMM methods for contextual clustering in HSM”. In: *Procedia Manufacturing*. Vol. 28. Elsevier B.V., Jan. 2019, pp. 154–159.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[34] P. Ramachandran, B. Zoph, and Q. V. Le. “Swish: a Self-Gated Activation Function”. In: *arXiv: Neural and Evolutionary Computing* (2017).

[35] W. Wang et al. “Clustering with Orthogonal AutoEncoder”. In: *IEEE Access* 7 (2019), pp. 62421–62432.

[36] D. P. Kingma and J. L. Ba. “Adam: A method for stochastic optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. Dec. 2015.

[37] K. Janocha and W. M. Czarnecki. “On Loss Functions for Deep Neural Networks in Classification”. In: *Schedae Informaticae* 25 (Feb. 2017), pp. 49–59.

[38] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. 2nd ed. New York: Springer, 2009.

[39] J. Xie, R. Girshick, and A. Farhadi. “Unsupervised Deep Embedding for Clustering Analysis”. In: *33rd International Conference on Machine Learning, ICML 2016* 1 (Nov. 2015), pp. 740–749.

[40] X. Guo et al. “Deep Clustering with Convolutional Autoencoders”. In: *ICONIP*. 2017.

[41] X. Ji, A. Vedaldi, and J. Henriques. “Invariant Information Clustering for Unsupervised Image Classification and Segmentation”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9864–9873.

[42] P. Zhou et al. “Unsupervised feature selection for balanced clustering”. In: *Knowledge-Based Systems* 193 (Apr. 2020).

APPENDIX

A. Structure of the CAE on latent feature videos

The structure of the CAE used on the 22×128 data can be seen in Figure 6. The CAE for 4×128 data has the same structure.

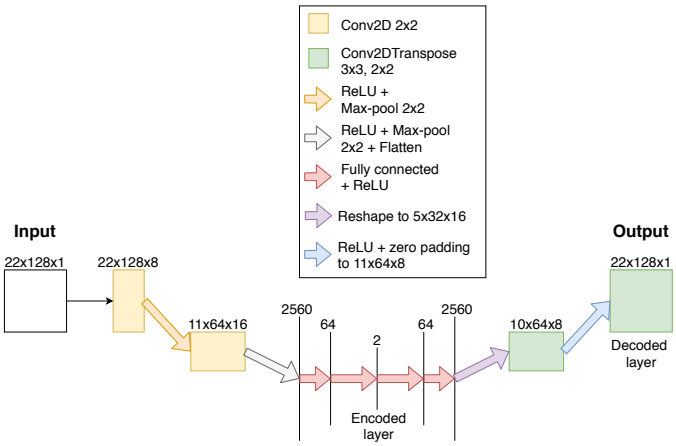

Fig. 6: Model structure of the CAE used for dimensionality reduction of the latent feature videos.

B. Training loss of the 3D-AE

Figure 7 shows the reconstruction loss per epoch during the training of the 3D-AE. The minimum reconstruction loss was 7.35×10^{-3}. The OAE loss is shown in Figure 8. The OAE loss for the 3D-AE was 0.368. The total validation loss is nearly identical to the reconstruction loss.

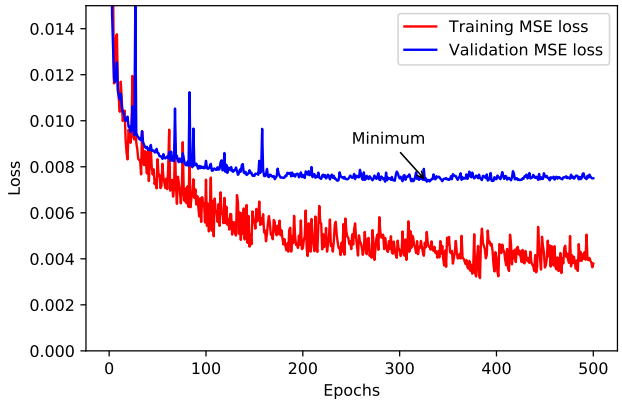

Fig. 7: Training and validation reconstruction loss of the training of the 3D-AE. The arrow shows the epoch with the lowest validation loss.

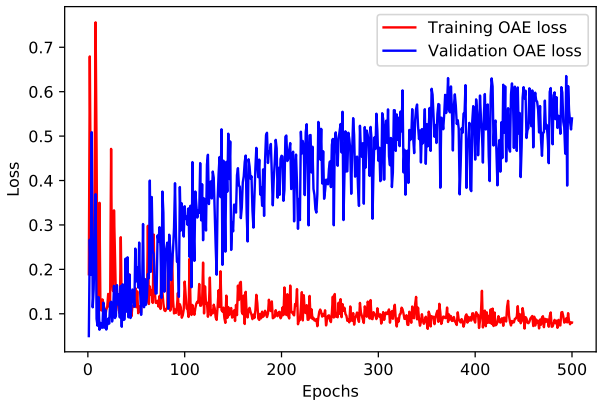

Fig. 8: OAE loss of the training of the 3D-AE.

3 Additional content

In this section we will go over two subjects that were not within the scope of the article, so were omitted there. The first is using the PCA on the latent feature videos for different combinations of frames than in the article. This is an additional source of information on the data. Furthermore, we address evaluation metrics that we attempted to use other than the contraction and valsalva classification. These were unfortunately unsuccessful, so were not included in the article.

3.1 PCA on latent feature videos

In the article, we applied PCA either on all latent feature video frames at once, or on just the last 4 frames. These led to the most relevant results for determining the maneuver performed. Here we show more plots that were made with applying PCA to the latent feature videos, for more insight on the information contained in the latent features.

Figure 9 and Figure 10 show plots of PCA applied to the first frames and last frames of the latent feature videos. The first frames seem to not cluster, which is expected as all videos should be in resting position. For the last frames, there is clear clustering. This is also as expected, as it should be similar to using the last 4 frames.

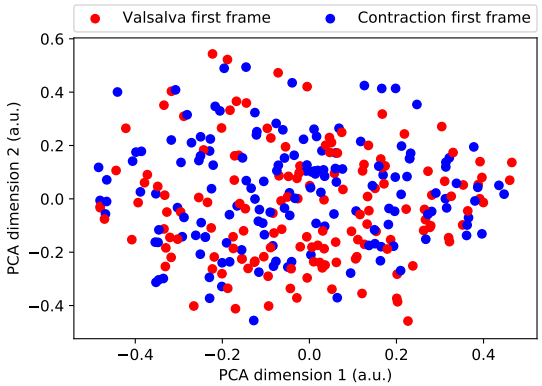

Figure 9: PCA applied to the first frames of valsalva and contraction latent feature videos.

Figure 11 shows a plot where PCA was applied to both the first and last frames of contraction and valsalva. Here, the valsalva first frames, and the contraction first and last frames seem to cluster together, while the last frames of valsalva form a seperate cluster. This indicates that the latent features clearly show valsalva. We think that the separation of contraction and valsalva is mostly done on valsalva versus non-valsalva. There does not seem to be a clear

difference between the resting position and contraction for the first and last frames.

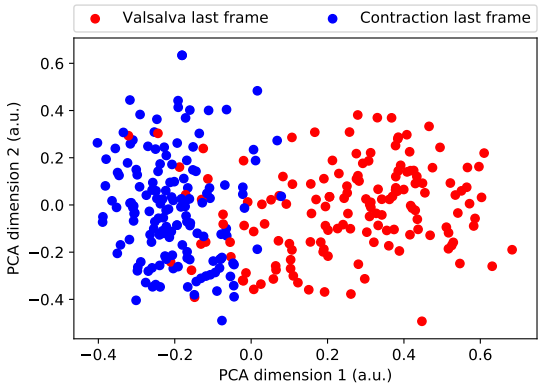

Figure 10: PCA applied to the last frames of valsalva and contraction latent feature videos.

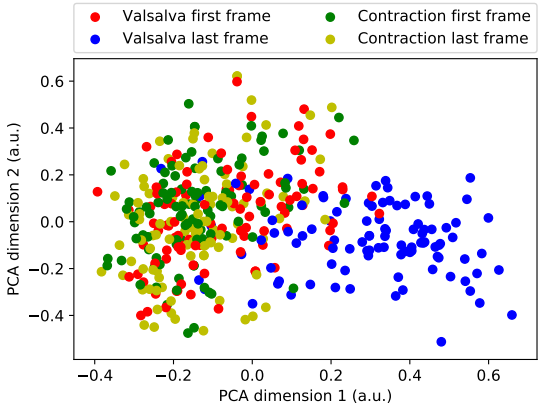

Figure 11: PCA applied to both the first and last frames of valsalva and contraction latent feature videos.

Figure 12 shows PCA applied to the first frames of both the contraction and valsalva video for 8 patients. Generally, there is some clustering per patient, which indicates that the latent feature videos contain information about the shape of the pelvic floor per patient. The similarity between first frames should be there since both the valsalva and contraction video start in resting position, so the shape of the pelvic floor should be the difference between the frames. It is not apparent for every patient however, this could be due to different scanning angles or movement of the patient in between scans.

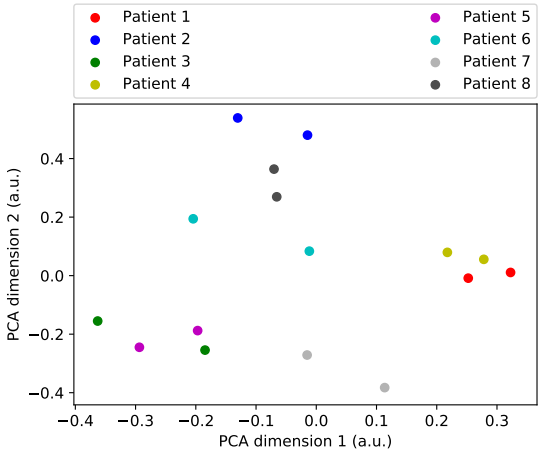

Figure 12: PCA applied to the first frames of valsalva and contraction latent feature videos for 8 patients.

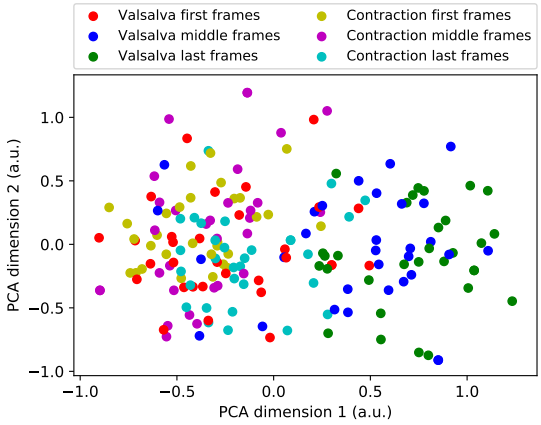

Figure 13: PCA applied to the first 4 frames, middle 4 frames and last 4 frames of valsalva and contraction videos.

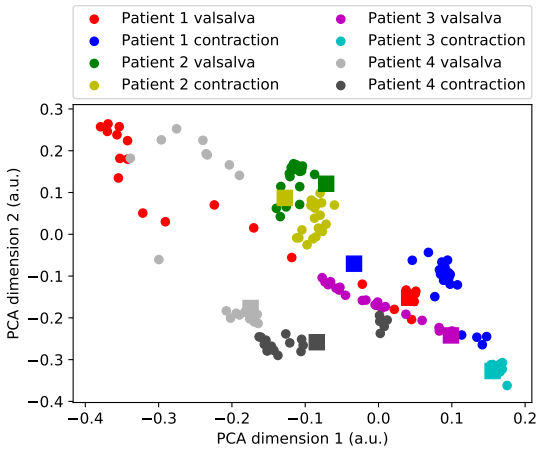

Figure 14: PCA applied frame by frame to valsalva and contraction latent feature videos of 4 patients. The square indicates the first frame of a video.

Figure 13 presents a similar view as Figure 11. It shows the PCA applied to the first 4 frames, middle 4 frames and last 4 frames of both the contraction and valsalva latent feature videos. There seem to be 2 clusters here, a big cluster on the left and a smaller cluster on the right of the plot. The smaller cluster contains the last valsalva frames, and a big portion of the valsalva middle frames. We expect this kind of clustering, as it shows that the last valsalva frames have their distinct cluster, but for the middle frames of the valsalva videos, not all videos are in valsalva yet. That is why some of the middle frames are still in the left cluster. The contraction videos do not change in a visible way in this plot.

Figure 14 shows how the latent feature videos change from frame to frame. For each patient the first frames tend to lie close to each other in PCA space, as also observed in Figure 12. The valsalva videos tend to change the most in PCA space, which is to be expected since the valsalva is most visible in the other PCA plots. However, there is movement in most of the contraction videos as well, which does indicate that the contraction latent feature videos change in time.

3.2 Evaluation metrics

Valsalva and contraction were successfully classified, but we also attempted to use additional evaluation metrics to classify how well the 3D-AE reduced the dimensionality of the ultrasounds. These additional methods were unfortunately unsuccessful. They are treated in this section, to avoid future pitfalls.

3.2.1 Rest, contraction and valsalva

We attempted to distinguish the resting position from the contraction and valsalva maneuvers. The start of each scan should have the patient in resting position, while from around halfway in the video the maneuver will be performed until the end of the video. We attempted to classify the positions using a CNN, which is nearly identical to the CNN used for classification in the article. Only its output and loss function were modified.

We labeled the first 5 frames of each video as being rest, the last 5 frames of valsalva videos to be valsalva, and the last 5 frames of contraction videos to be contraction. With these 3 labels, we attempted to train a CNN. This was done with the latent feature videos, so with 5×128 data.

First we tried the Softmax [5] function as an output, with the log loss [6]. The classes are unbalanced, there are more resting labels. Thus, we tried classification both with unbalanced classes and with classes that were all made equally large. This class balancing was done by discarding data from the resting and contraction sets in a random manner, as the valsalva set was the smallest. The unbalanced Softmax led to a validation accuracy of 68.2%. The validation accuracy of valsalva was 86.1%, the resting position had a 94.6% accuracy and the contraction had a 4.9% accuracy. The model tends to classify valsalva correctly, while the Softmax balances the contraction and resting labels according to their occurrence in the data set. This means that the prediction, which is the highest value in the Softmax, will be the resting position for both the resting and contraction labels. The balanced data set had a validation accuracy of 63.5%. The valsalva had a validation accuracy of 96.3%, the resting position 51.4% and the contraction 45.7%. This is the same trend as the unbalanced classes, where accuracy is good on valsalva, but contraction and the resting position are balanced in the Softmax. This led to random guesses here.

We also tried using a linear output function. Here the contraction was given a 0, the resting position a 1, and valsalva a 2. These labels led to the best results, compared to other label combinations. The MSE was used as the loss function. With unbalanced classes, the average validation output for contraction was 0.696, for resting it was 0.745 and for valsalva it was 1.73. For balanced classes, the average validation outputs were 0.580 for contraction, 0.563 for resting, and 1.71 for valsalva. These outputs line up with the trend that is seen in the Softmax CNNs, where the valsalva can be distinguished, but the contraction and resting position predictions are simply based on their occurrence in the data set.

In Figure 13 it is visible that the distinction between contraction and resting position is not visible in PCA space, while the difference between resting and valsalva is visible. We think this is the reason why the CNN can not make this distinction.

The distinction between rest, contraction and valsalva might be made by using a RNN, as mentioned in the article. Figure 14 does support this idea, as in PCA space the contraction latent feature videos do move from frame to frame.

3.2.2 Pelvic organ prolapse

For 200 patients the measure of pelvic organ prolapse was given. Those measures are estimates made by the gynaecologist, from 0 to 3. We again used the same CNN, while varying the outputs and loss functions. There are 19 patients with a 0, 44 patients with a 1, 96 patients with a 2 and 41 patients with a 3. We used the last 4 frames of the valsalva videos, as we assume that the prolapse is most visible during valsalva.

First we tried to build a CNN with a linear output function and the MSE loss. With the full data set, an accuracy of 45.9% was achieved, which we attribute to random guessing. With balanced classes, that is all classes only have 19 data points, the accuracy drops down to 26.7%. This further indicates that no information is actually learned, and guesses are random.

We also attempted using a Softmax with 4 classes and the log loss. With unbalanced classes the model achieved a validation accuracy of 43.6%. With balanced classes the validation accuracy was 31.6%. This is again shows the guessing is random, only based on the amount of labels per class.

The final attempt was to train a CNN with binary labels and the log loss. One class contains prolapse measure 0 and 1 and the other class contains prolapse measure 2 and 3, splitting the data set on severeness of the prolapse. With unbalanced classes, this led to a validation accuracy of 67.6%. This accuracy was achieved by simply predicting the 2 and 3 class for all data. With balanced data, we got a validation accuracy of 58.3%. However, this accuracy jumped around 50% for multiple runs, so we assume this to be random guessing.

We conclude that the measure of pelvic organ prolapse cannot be found in the dimensionality reduced data. It is not known if the measure of pelvic organ prolapse can be detected using deep learning models, as it is a rough estimate by the gynaecologist. This estimation can be inconsistent, so the prolapse might not be visible in the data. If it is visible, supervised methods should be used to find it.

4 Conclusion

We have studied 4D ultrasound data of the pelvic floor using unsupervised deep learning. This was done by constructing the 3D-AE, an unsupervised 3D CAE which reduces the dimensionality of the 3 physical dimensions of the ultrasound data. We used it to reduce the 4D data to 2D latent feature videos. We successfully classified the maneuvers performed in the ultrasounds using the latent feature videos, with supervised and unsupervised methods. This shows that useful information can be extracted from the 4D ultrasounds in an unsupervised manner.

The latent feature videos unfortunately had no clear distinction between when the LAM was in resting position or in contraction. A CNN was unable to distinguish them, and there was no clear difference between them when PCA was applied. Furthermore, the measure of pelvic organ prolapse of the patients

could not be found in the latent feature videos. This data does not seem to be in the latent feature videos, but it is unknown if it can even be observed in the data because the measure of prolapse is a rough estimate. These failed evaluation metrics show that the 3D-AE has a focus on the most prevalent features within the ultrasounds.

In future work, a RNN could be constructed in an attempt to find when contraction or valsalva starts within a latent feature video. This would be medically relevant because it can give a measure of how much the LAM contracts, which can be an indication of damage to the LAM.

5 References

[1] R. Vargas, A. Mosavi, and R. Ruiz. "Deep Learning: A Review". In: *Advances in Intelligent Systems and Computing* 5 (2017).

[2] M. Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features". In: *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018, pp. 139–156.

[3] S. Chen, K. Ma, and Y. Zheng. "Med3D: Transfer Learning for 3D Medical Image Analysis". In: *arXiv e-prints* (Apr. 2019), arXiv:1904.00625.

[4] J. Masci et al. "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction". In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 52–59.

[5] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[6] K. Janocha and W. M. Czarnecki. "On Loss Functions for Deep Neural Networks in Classification". In: *Schedae Informaticae* 25 (Feb. 2017), pp. 49–59.