Deep learning methods for cardiomyocyte motion analysis

Tijmen de Wolf
4604903

August 26, 2022

In partial fulfilment of the requirements for the degree of

Master of Science
in
Biomedical Engineering & Nanobiology

at the Delft University of Technology

Department of Cell Biology Erasmus MC
Rotterdam, The Netherlands

Contents

1 Abstract

Heart failure is a leading cause of death and forms a growing health concern. The development of novel drugs is however hampered by the absence of adequate screening methods and disease models. Cardiomyocytes derived from patients could assist in the development of a patient specific drug screen method to test the efficacy and safety of putative drugs. Simultaneously, deep learning has been applied to a variety of biomedical datasets, achieving state-of-the-art performance. Previous methods for the classification of cardiomyocytes as healthy or diseased only focused on machine learning methods. We present the first deep learning approach to perform this classification task together with a novel artificial intelligence interpretability method called Contraction Analysis Local Interpretable model-agnostic explanations (CA-LIME), able to explain the predictions made by the classifier. The proposed classifier is shown to outperform previously developed methods to classify cardiomyocytes, obtaining 97.5% accuracy. Our results indicate this classifier could aid in the development of a high throughput drug screening system for cardiac drug development. The explanations made by CA-LIME are in correspondence with previous observations of drugs with known effects, verifying the effectiveness of our approach. Together with CA-LIME, the processing pipeline could lead to the discovery of new differences between the motion of healthy and aberrant beating cardiomyocytes.

2 Introduction

2.1 Heart failure

Heart failure (HF) describes the clinical conditions which impair the heart from ejecting enough blood to fulfill the metabolic demands [1, 2, 3]. Underlying causes include damage to the heart muscle by hypertension (high blood pressure), genetic mutations and myocardial infarction (heart attack) [1]. With an estimated 64 million people diagnosed in 2017, HF represents a growing health concern [4, 5]. The chronic disease is increasing among young adults ($\leq$ 50) [6] and results in increased mortality rates [7]. With HF as a leading cause of death, novel medication must be developed [8, 9]. Development of cardiac drugs has however made slow progress, for the amount of approved drugs declined over the last decade [10]. The decline is largely attributed to lack of efficacy or cardiac toxicity observed during late stages of drug development [10]. Improved models for cardiac disease can aid the efficacy and toxicity screening in early stages of drug development. Induced pluripotent stem cells derived cardiomyocytes (hiPSC-CMs) provide a promising patient-specific method to model cardiac diseases [11, 12]. Grown in a monolayer, the hiPSC-CMs display an organized periodic motion of contraction and relaxation. Analysis of this motion enables the screening of novel drugs during early stages of development in a patient-specific cardiac model [11, 13]. Here, we propose a novel approach to analyze the motion of beating cardiomyocytes using artificial neural networks and modern deep learning methods.

2.2 Deep learning

Artificial neural network, their structure and how they process data, are based on ideas, structure and functioning of the actual neural connections in the brain [14]. During training (also called "learning"), all connections in the network are tuned to optimize the performance for a specific task. Among these tasks are: segmentation, classification, denoising, data compression and object detection. Deep learning is an approach to artificial intelligence (AI) in which many layers of artificial neurons are used to construct the network. The method shows state-of-the-art performance in many fields, such as computer vision and medical image processing [15]. The use of many layers enables the extraction of useful features from the data to be learned during training [14]. Supervised deep learning is a sub-domain in which data and the known classification is available during training [14]. Following the training procedure, the algorithm can be used to make inferences about unseen data.

Developed deep learning algorithms are often treated as "black box" models [16, 17]. The extracted features and their relative importance are unknown, and ignored as merely the obtained accuracy metrics are considered [17]. Interpretability methods for AI provide explanations about the predictions made by deep learning algorithms, explaining why that specific prediction was made [18, 19]. Explanations can be in visual or textual form, and serve three goals [17, 20]. Firstly, explanations provide trust for the model by providing insight into the extracted features, which is important when using deep learning models in real world applications [20, 17]. Secondly, points of improvement for the algorithm can be determined. Finally, explanations could provide novel scientific insights by looking at the extracted features and their importance for the made prediction [20, 16, 21]. Multiple interpretability methods have recently been developed, such as SHapley Additive exPlanations (SHAP) [20] and Local Interpretable Model-agnostic Explanations (LIME) [17] which provide general strategies to explain deep learning models for any field of application.

Despite the advances of deep learning, it has not yet been applied to aid a high throughput drug screening system for HF. A single machine learning based method which utilizes the motion of beating cardiomyocytes was previously developed, but remains limited to the features extracted in the pre-processing steps [11]. Furthermore, the use of AI interpretability methods could result in novel insight for HF, but has, to the best of our knowledge, not yet been investigated.

2.3 Deep learning based contraction analysis

This study describes a novel method based on deep learning to classify healthy and diseased beating cardiomyocytes, based on the observed motion. The developed method is trained to distinguish healthy and aberrant behavior of beating hiPSC-CMs imaged using phase contract microscopy time series. Image registration is used to extract the motion at each spatial location. Therefore, the proposed deep learning method can use the available spatial and temporal information and learns to extract useful features. With this automated method, we aim to contribute to the development of a high throughput drug screen system able to detect efficacy and cardiac toxicity.

To gain insight into HF, we developed a novel AI interpretability technique, called Contraction Analysis LIME (CA-LIME). CA-LIME is specifically designed to explain the predictions of contraction analysis

data. Unlike other interpretability techniques, it takes into account the periodic nature of the contraction profiles. Using CA-LIME, the importance of manually extracted features in the input images is determined, providing explanations which are easy to interpret. We propose that CA-LIME, together with existing AI interpretability techniques, can be used to unravel the difference between healthy and aberrant cardiomyocytes and explain the cardiac toxicity of therapeutic agents.

3 Theory

3.1 Induced pluripotent stem cell derived cardiomyocytes

Cardiomyocytes are the muscle cells of the heart, responsible for the orchestrated beating motion that pumps blood through the body. Cultured cardiomyocytes display a cyclic motion consisting of contraction followed by relaxation [22]. Heart diseases can be studied from abnormalities in the observed motion of beating cardiomyocytes [23]. Obtaining cardiomyocytes from the heart of patients is however not practical, due to the highly invasive nature of the procedure. Furthermore, the resulting cardiomyocytes are difficult to culture and the obtained count of cells is often low, obstructing the use of cardiomyocytes in high throughput drug screening [24]. Novel stem cell technology, however, enables the derivation of cardiomyocytes from patients without harvesting from the heart [25, 26]. Fibroblasts taken from patients can be reprogrammed into human induced pluripotent stem cells. These stem cells can subsequently be differentiated into cardiomyocytes-like cells while retaining the genotype of the cell doner [25, 26]. This enables high throughput drug screening by using patient specific cardiomyocytes.

3.2 Neural networks

Deep learning algorithms use artificial neural networks to perform complex tasks in a seemingly intelligent way [14]. Similar to brain architectures, the algorithms consist of multiple artificial neurons connected to each other [27]. Artificial neural networks represent the state-of-the-art processing method for many tasks such as object detection, natural language processing, data compression or image classification [28]. Image classification is the task of assigning a label or target class to an image, examples include skin lesion classification or detecting pneumonia from chest X-rays [29, 30]. Mathematically each neuron in a fully connected layer of the network produces its output according to the following rule:

$$f(x) = \Phi(x^T w + b) \tag{1}$$

here f is the output vector, x represents the input vector which is multiplied by the weights matrix w, after which a bias b is added. These weights and biases are determined during training, and represent the learnable parameters. The transpose operation indicated with T, matches the dimension between the weights matrix and input vector for matrix multiplication to occur. To enable the neuron to be non-linear, the non-linearity Φ is added. The non-linearity can be ReLU, sigmoid, or any other non-linear function [31, 14]. Artificial neural networks contain many neurons connected in successive layers, where the output of the previous neuron is passed as input to the next neuron. A network of multiple layers, a multi-layer perceptron (MLP), can now be formed (see Figure 1). Mathematically, a network of three layers can be described as:

$$y = f_3(f_2(f_1(x))) = F(x) \tag{2}$$

where y is the output of the network resulting from the input x and f_i indicates layer i of the network. The general approximation theorem now states that given a sufficiently large network, any function can be approximated [32, 33, 34]. Which function is approximated depends on the weights and biases, w and b respectively of equation 1. Given the artificial neural network F in equation 2, the training procedure optimizes the weights and biases such that input x is mapped to the desired output. During training, input samples x together with the known outputs $\hat{y}$ (ground truth) are provided. Based on a loss function, the performance of the network is determined during training. Using gradient descent, the learnable parameters are updated to optimize this loss function [35, 36, 14]. Cross entropy loss is an often used loss function defined as [37, 38, 14]:

$$CE_{loss} = -\sum_{c=1}^{C} \hat{y}_c \log(y_c) \tag{3}$$

In which $\hat{y_c}$ and y_c are respectively the ground truth and output of the network for class c, the total number of classes is indicated with C. The loss function assures all learnable parameters are updated such that the difference between the network output y and the ground truth $\hat{y}$ is minimized.

3.2.1 Convolutional neural networks

Convolutional neural networks (CNNs) are similar to neural networks, but the used mathematical operation is a convolution [39, 40, 14, 14]. Convolutions use a small kernel that slides across the input image, and takes the weighted average of the values in the kernel and the intensity in the image at each location [41].

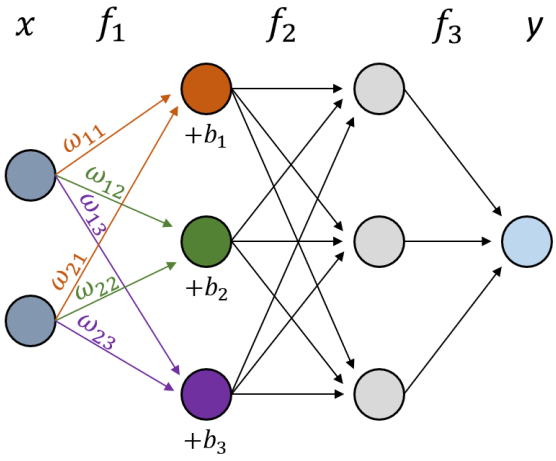

Figure 1: *Example neural network consisting of three layers. Acting on the input layer x, the mathematical operation of equation 1 is indicated for the first layer f_1. The output y follows the relation as in equation 2.*

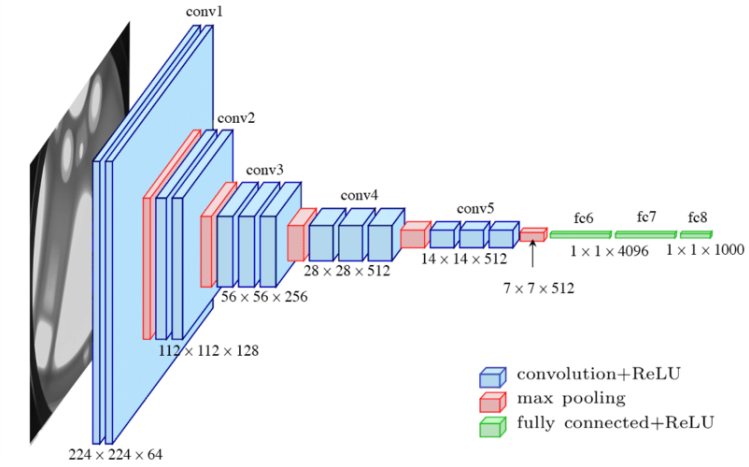

Figure 2: *Example CNN showing a deep architecture able to learn the feature extraction. The max pooling operations in red reduce the size of the images after the convolutional layers. Obtained from Ferguson et al. [48].*

The kernel is often chosen to be 3 by 3 or 5 by 5 pixels for image classification tasks. The resulting output after the convolution is a new image, in which patterns in spatially neighboring regions can be detected. The learnable parameters in CNNs are the numbers (weights) inside the kernel. The training procedure changes the weights and bias of the kernel such that the correct features are extracted from the images to perform the specified task. Because CNNs use convolutions, the number of learnable parameters is reduced compaired to fully connected layers. This enables the design of CNNs with many layers, which is required to achieve state-of-the-art performance in deep learning [42].

Convolutions can consist of multiple output channels, to extract multiple features from the same input image. To do so, multiple kernels are used, each performing a convolution operation and calculating separate output images as result. Similar to the neural networks described above, the output of one layer is used as input in the subsequent layer in the CNN (see Figure 2). Before passing the output on to the next layer, the size of the image is reduced by average or max pooling operations [43, 44, 14]. The subsequent layer of the CNN can now extract features at smaller resolution scales, which facilitates the detection of features larger than the kernel size, by increasing the receptive field [43, 14]. CNNs have been the state-of-the-art method for image processing over the recent years [45]. Convolutions however, depend on local context as the operations acts in a small neighborhood of the size of the kernel. As a result, long range relationsships are difficult to model [46, 47]. Contractions far apart in time, or distant parts in images are therefore difficult for CNNs to base a decision on. This issue was recently solved using transformer architecture for images [47].

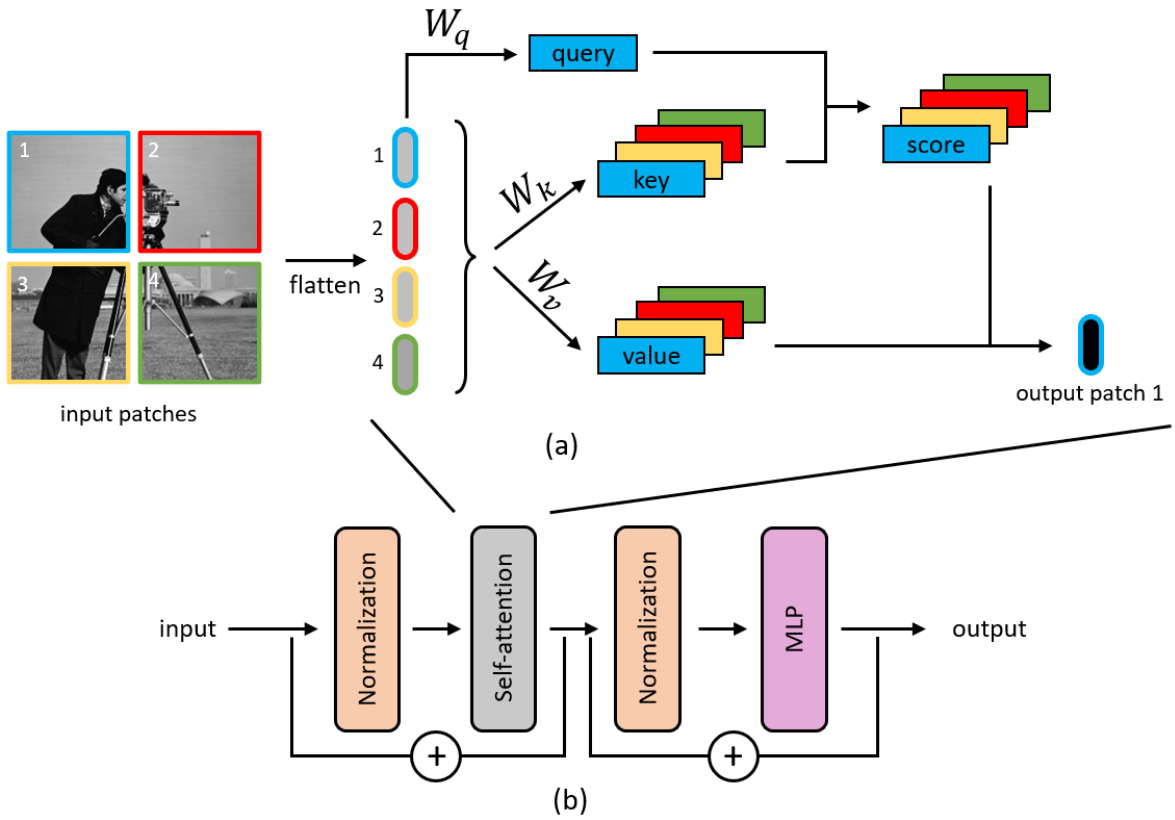

Figure 3: *Self-attention mechanism used to construct a transformer block. (a) the self attention mechanism acting on an image divided in 4 patches. The schematic shows the steps to construct the output of the first patch. The mechanism is repeated to construct the output of the other patches. (b) The transformer block consisting of normalization layers, self-attention and a MLP layer.*

3.2.2 Transformer architectures

Similar like CNNs, transformer architectures are artificial neural networks. In order to deal with global relations, transformers were introduced [49]. Transformer architectures rely on self-attention mechanisms [49]. Attention mechanisms were first introduced in sequence models, which deal with sequential input data, like words in a sentence [49, 50]. Self-attention determines how important each word in a sentence is, based on all other words that occur in the sentence. It was shown that self-attention mechanisms alone, are sufficient to reach state-of-the-art performance on tasks like language translations or text classification [49]. Following this work, transformer architectures deploying self-attention were used for image classification tasks as well, outperforming commonly used CNNs [47]. These transformers for image-related tasks are called vision transformers (ViT). To treat an image as sequence, the input image is divided in patches (see Figure 3a). The different patches are now similar to words in a sentence, all distinct parts which together form a whole. Because self-attention mechanisms are based on vectors as input, each patch is flattened to a vector. Consider an image I divided into P patches. In self-attention, each patch I_p is used to calculate a "key", "query", and "value" by taking the input patch I_p and apply linear transformations:

$$query = Q = W_q I_p, \quad key = K = W_k I_p, \quad value = V = W_v I_p,$$

in which W_q, W_k, W_v, are the learnable parameters. Using the key and query, the attention scores ω can be calculated:

$$\omega_{pj} = softmax(Q_p^T K_j),$$

The softmax operation serves as normalization, to bound the attention score between 0 and 1 [51]. Finally, the output patch S_p is calculated as:

$$S_p = \sum_{j=1}^{P} \omega_{pj} V_j$$

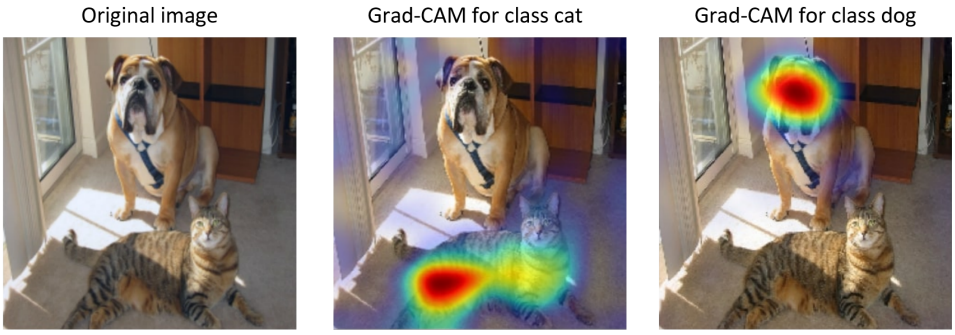

Figure 4: *Result of Grad-CAM on an example image containing a cat and dog. The original image is indicated on the left. The middle pannel shows the heatmap for the class cat, the right pannel for the class dog. Red colors correspond to higher scores for the class of interest. Adapted from Selvaraju et al. [19].*

Self-attention is a sequence-to-sequence operation, meaning that P input patches result in P output patches. The attention mechanism can be divided into multiple heads, allowing a specific patch to have multiple relations with the other patches. A transformer block is constructed of a normalization layer, followed by self-attention with another normalization layer, and finally a MLP (see Figure 3b). Transformer architectures usually contain multiple layers of transformer blocks to form a deep architecture. Unlike CNNs which rely on local dependencies in the image, transformers are able to capture global context by relating each patch to all other patches [47]. Because of their state-of-the-art performance, transformer architectures are increasingly used for medical image processing [52, 53].

3.2.3 Interpretability

Neural networks are used for an increasing number of tasks [52]. Given a network (i.e. equation 2), one is often interested in the output y, but not how the function F (the network) generates this output [16, 17]. Such usage treats the network as a black box. Interpretability is the task of providing an explanation how the network made a specific decision in terms understandable for humans [54]. Local explanations concern individual instances (explain the prediction for a specific sample from the dataset), whereas global explanations cover the network as a whole (explain prediction for the entire dataset) [17]. Many methods for interpretability tasks have been developed, an excellent review of available methods can be found in the work of Linardatos and colleagues [55]. The following methods will be considered here: Gradient-weighted Class Activation Mapping (Grad-CAM) [19], LIME [17] and SHAP [20].

In the present study, Grad-CAM and SHAP will be used to generate explanation and change the input to improve the performance of our classifier. LIME is at the basis of SHAP and provides fundamental understanding of how the explanations are generated. We propose an adapted version of LIME, called CA-LIME, which is specifically adapted to explain the contractility profiles of beating cardiomyocytes. The provided background information about LIME helps to understand CA-LIME and the motivation behind the alterations made.

Grad-CAM

Grad-CAM is a local interpretability method that produces visual explanations for individual input samples. Explanations for any CNN architecture can be obtained without the need for retraining or adaptation of the architecture [19]. The method depends on the gradients of the neural network, which during training are used to update the weights. Using a class of choice, the gradients with respect to the last convolutional layer in the network are visualized. The gradients are visualized as a heatmap, indicating which regions in the image contribute to the output of that class that is explained. Because of pooling operations, the final convolutional layer is usually much smaller than the original input image. To correct for this, the obtained heatmap is upsampled using interpolation, after which it can be depicted as overlay on top of the image that is explained (see Figure 4). As explanations for each output class can be obtained, the contribution of each part of the image for all possible classes can be determined.

LIME

Unlike Grad-CAM, which requires convolutional layers to be present for the generation of explanations, LIME is able to explain any black box model B without changing the model [17]. LIME is a local interpretability

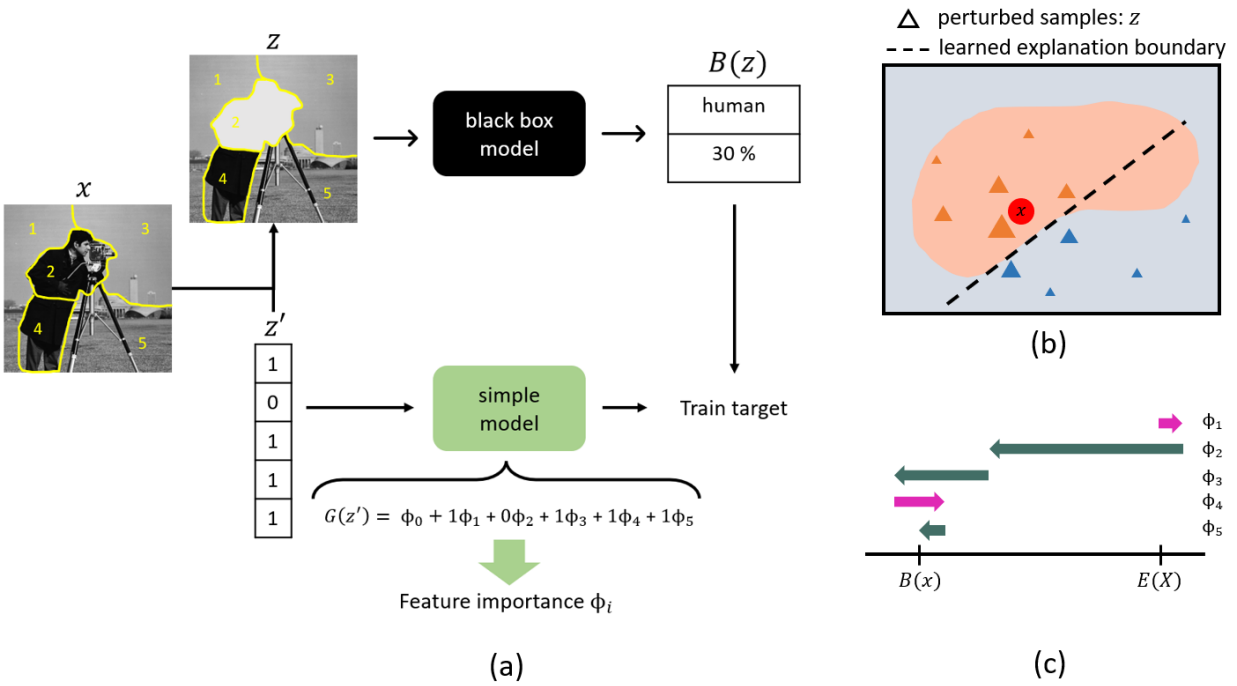

Figure 5: *LIME and SHAP interpretability. (a) Defining 5 superpixels on the image instance x. The feature vector z' determines which superpixels are present in the image, resulting in the perturbed sample z. A simple model (in green) is trained to reproduce the output of the black box model B. The parameters ϕ_i of the simple model now indicate feature importance. (b) Perturbation (triangles) of the image instance x are created. The weight of each perturbed sample is determined by the proximity to x, indicated by marker size here. The dashed line indicates the learned explanation, which is a local explanation and not a global. (c) SHAP values together explain the difference between the mean of the dataset $E(X)$ and the explained instance x.*

method, as individual input instances are explained. To generate an explanation, LIME takes an input sample x and generates local perturbations of this sample. In the context of images, these perturbations are generated by dividing the images in multiple regions, called superpixels, and switching these on or off. Which superpixels are turned on or off, is kept track of by a binary feature vector z', in which 1 indicates on, and 0 indicates off (see Figure 5a). A superpixel is switched off by setting the pixel values to a set value, like zero. The perturbed image z with switched off superpixels, is similar to the unperturbed image x, but not the same. Feeding perturbed samples through the black box model, will therefore result in a different output $B(z)$.

The general idea behind LIME is to create a dataset Z consisting of perturbed samples, to which a simple model G can be fitted which is more interpretable. This simple model can for example be a decision tree, or a linear regression model. When fitting a linear regression model, a weighted fit can be made, taking into account the similarity between x and z for each perturbed sample (see Figure 5b). Similarity can be measured using a distance metric, like the cosine distance or the amount of perturbed superpixels, resulting in the weight $\pi_x(z)$. The weighted fit puts more emphasis on samples closer to x. Following this procedure, an explanation ξ can be obtained by training a simple surrogate model:

$$\xi(x) = \arg\min_{G} \quad \mathcal{L}(B, G, \pi_x) + \Omega(G) \tag{4}$$

in which Ω represents the complexity of the simple model (the amount of learnable parameters or amount of superpixels). To generate explanations, the function $\mathcal{L}$ is minimized, while keeping the complexity low to maintain the interpretability of the simple model. While training a linear regression model, $\mathcal{L}$ is defined as weighted square loss:

$$\mathcal{L} = \sum_{z,z' \epsilon Z} \pi_x(z)(B(z) - G(z'))^2 \tag{5}$$

such that the simple model G with the binary feature vector z' as input, is trained to produce the same

output as the black box model B for the perturbed image z. LIME considers the simple model to be a linear model of the binary feature vector:

$$G(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{6}$$

where M is the number of features and ϕ represents the learnable parameters of the explainable model. The simple model attributes a score ϕ_i to each superpixel, indicating the importance of that superpixel. Similarly as Grad-CAM, individual output classes can be explained for multiclass classification problems.

SHAP

Inspired by game-theory, SHAP has a larger mathematical foundation compared to previous methods [20]. The values predicted by SHAP represent shapley values [56], and they indicate the importance of each feature that is explained. For shapely values, the parameter ϕ_0 in equation 6 is set to the expectation of the model. As a result, the values ϕ_i represent the contribution of each feature in explaining the difference between the mean and the prediction of the black box on sample x, such that $\sum_{i=1}^{M} \phi_i = B(x) - E(X)$ (Figure 5c). Here, $E(X)$ represents the expectation value on the entire dataset X. The predicted values by SHAP satisfy the following properties:

1. Local accuracy, stating that the explanation model G should match the original black box model B for the simplified input $z' = 1$ (all superpixels turned on).
2. Missingness, which requires that a feature not present in the original input x, have no impact on the output.
3. Consistency, which states that a change in the model that makes a specific feature more important, cannot result in a decrease of the predicted shapley value for that feature.

SHAP provides a unified approach based on six other methods, among which is LIME, that predicts values for feature importance with the above properties. LIME on the contrary, only satisfies the property of missingness [20, 57]. Under the correct choice of the parameters in equation 4, the values predicted by LIME, however, approximate shapley values and satisfy the above properties. This method is called Kernel SHAP and it connects LIME to shapley values under the following parameters of equation 4:

$$\Omega(G) = 0, \quad \pi_x(z') = \frac{M-1}{(M choose |z'|)|z'|(M-|z'|)}$$

in which M is the number of features in the feature vector and $|z'|$ is the number of non-zero elements in z'. The function $\mathcal{L}$ should be chosen as in equation 5. The sample weighting function π_x now puts large weights on samples close to the original input x, as well as on samples in which almost all features are perturbed. Intuition for this choice is to isolate a feature by perturbing all other features, this will provide a lot of information only about the feature that was not perturbed. In order to predict shapley values using Kernel SHAP, the feature vector z' must be a binary vector.

Grad-CAM and SHAP will be used to generate explanations for a deep learning classifier trained to classify the cardiomyocytes as healthy or aberrant. Using these explanations, the input data can be altered to achieve improved classification results. Finally, we propose CA-LIME, a novel interpretability method based on LIME, which is specifically designed to generate explanations for cardiomyocyte contractility profiles.

4 Methods

4.1 Data acquisition

Movies of healthy and aberrant hiPSC-CMs were used for this study (see Figure 6a). A total of 33 healthy, and 33 aberrant movies were available. The aberrant cardiomyocyte phenotype was generated by adding chemical agents to healthy cardiomyocytes (see Figure 6b). Isoproterenol or endothelin-1 was used for this purpose. Moreover, a cell line from a Hypertrophic cardiomyopathy patient with a p.Trp792ValfsX41 mutation in the MYBPC3 gene was used as aberrant phenotype. The healthy phenotype contains cardiomyocytes grown on regular MR44 media, and on MR056 media without glucose. Phase contrast timeseries imaging was performed by a commercial facility: Ibidi (Munich, Germany), partners on the HeartCHIP project using their incubation system. All cardiomyocytes are grown in monolayers on a polydimethylsiloxane substrate with a known stiffness of 15 kPa, produced by Ibidi. Imaging was performed at 24 frames per second using a Ti-Eclipse microscope equipped with an ORCA Flash 4.0 LT camera (0.33 μm/pixel). Movies ranged from 1200x1200 to 2048x2048 pixels in the spatial dimension and 240 to 600 frames along the temporal dimension. A full description of treatment and cell culture conditions can be found in the work of Snelders and colleagues [58].

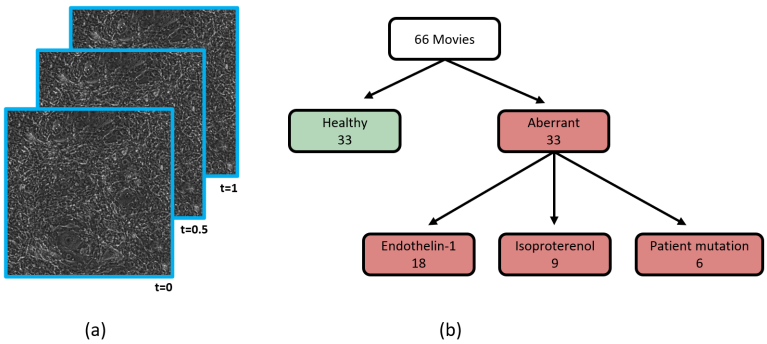

Figure 6: *Imaging and description of the dataset. (a) Movie that results after imaging, three different frames are shown as example. (b) Different conditions used during imaging. Both healthy and aberrant cardiomyocytes are imaged. The aberrant cardiomyocytes can be subdivided into drug induced (Endothelin-1 and Isoproterenol) or genetically induced aberrant phenotypes. The numbers indicate the amount of movies included in the dataset.*

4.2 Decision tree classifier

To the best of our knowledge, only a single paper proposing machine learning methods to classify the motion of beating cardiomyocytes as healthy or diseased is currently published. Teles et al. test multiple machine learning methods for the classification based on 8 calculated parameters, among which the maximum displacement, beating frequency and duration of the contraction [11]. A random forest of decision trees obtained the best performance, with an accuracy of 92% and F1-score of 91%. Based on their result, and an optical flow based analysis previously developed in the group [58], we investigated the performance of a random forest consisting of 1-30 decision trees, with increments of 1. The previously developed optical flow based analysis yields 19 extracted parameters represented as tabular data, where each entry has a well defined meaning (see Figure 7a). The analysis is able to determine absolute pressure values (in pascal) because the stiffness of the substrate underneath the cardiomyocytes is known. Data are z-score standardized such that each parameter has zero mean and unit variance before the decision tree, similar to Teles et al. [11] (see Figure 7b). A fourfold cross validation is used to determine the performance over the entire dataset. Importance of each feature is determined using the normalized Gini importance [59], indicating how much the Gini impurity index decreased by including the feature in the random forest.

4.3 Data preprocessing

Next to classical machine learning methods like decision trees, deep learning methods for the classification task are explored in this study. For this purpose, the motion of the beating cardiomyocytes is extracted from the movies with the use of image registration. Image registration finds the transformation that puts two different frames in the same coordinate system. As such, the movement between different frames can be

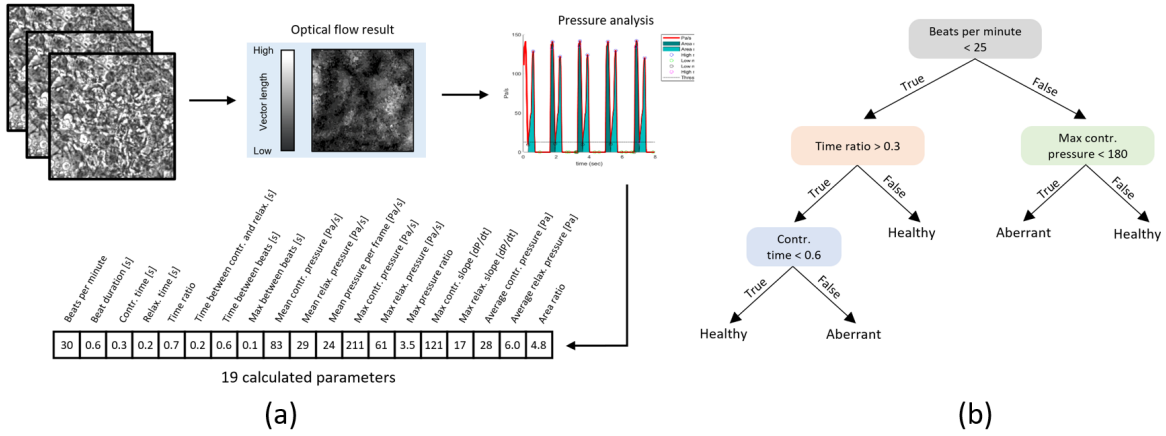

Figure 7: *Optical flow method and decision tree classification. (a) Input movies on a substrate of known stiffness are acquired. Using optical flow, the motion of the pixels over time is estimated. The results are used to perform the pressure analysis, in which 19 different features are calculated that describe the motion of the cardiomyocytes. The values indicated in the vector represent typical values for that feature. Adapted from Snelders et al. [58]. (b) Decision tree classifier where each level of the tree takes one feature and splits the data depending on the value of the features. Subsequently the data is split again based on a different feature. Which features are used and the depth of the tree is optimized during training.*

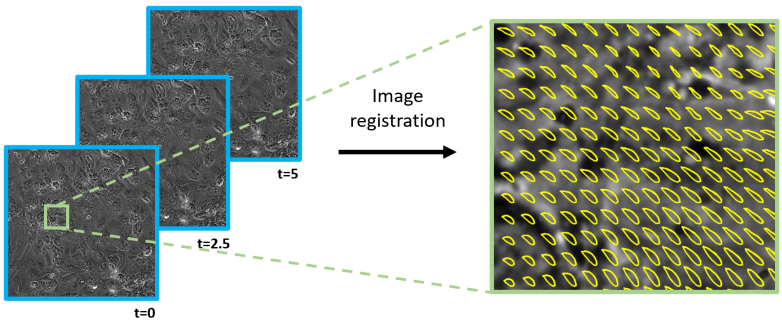

Figure 8: *Movement extraction using image registration. Image registration is used on the input movies to register the first frame to all consecutive frames. Using the obtained deformation field, the movement of every fifteenth pixel is determined. When the x and y-coordinates of a pixel are displayed, a track with a periodic shape arises as shown by the zoom in (the right panel).*

determined (see Figure 8). Registration was performed using Elastix [60], implemented for the CellsOnCHIP ImageJ plugin by Ihor Smal. The mutual information cost function is optimized with gradient-descent, to maximize the correspondence in the joint image histogram of different frames in the movie. The first frame is registered to all subsequent frames, to extract the movement for all time steps. After registration, the obtained deformation field is applied to every fifteenth pixel of the first frame along both spatial dimensions. As a result, the movement of these sampled pixels is extracted over time, and describes a track (see Figure 8).

Using image registration, the x and y-coordinates of the sampled pixels are determined for each time step. The x and y-coordinates are used to calculate four measures which describe the motion, these are: the distance from the median position, distance from the previous frame, angle relative to the median position and the angle relative to the previous position (see Figure 9). Supplementary Figure S1 indicates how the angle from previous is defined. These metrics are calculated for each time step, and used to construct movies in which the metrics are represented as the intensity in different channels, these movies are referred to as feature movies. These feature movies with calculated features are used for classification with a CNN. The CNN can now deploy both the spatial and temporal dimension of the data. To unify the spatial dimension of the resulting data, all feature movies are rescaled to 80x80 pixels using nearest neighbor interpolation, before further processing. The value of 80 pixels arises after subsampling the movie with the smallest spatial dimension of 1200 pixels with a factor of 15. Following registration, each movie is split in equal parts of 120 frames each, to have input movies of equal size and to generate more training data. The angle features are

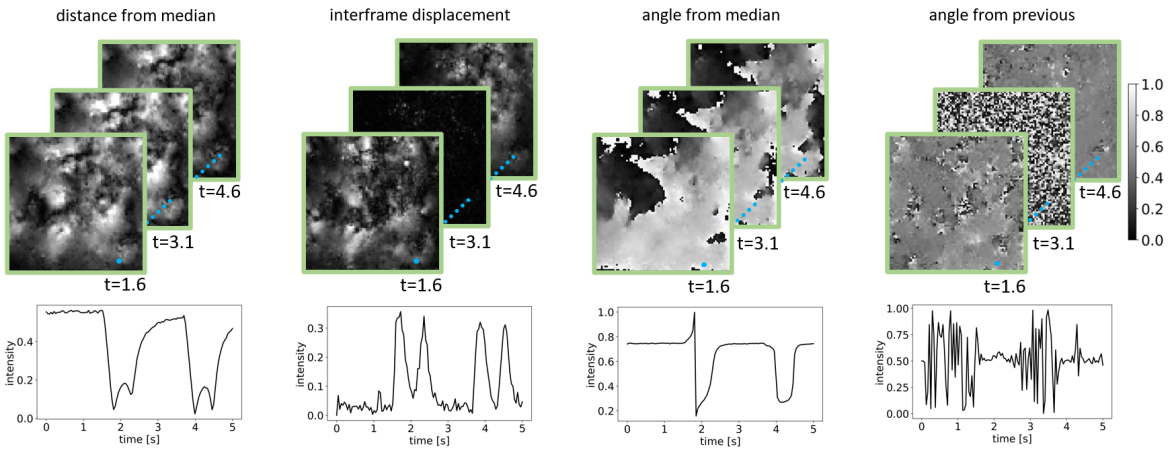

Figure 9: *Calculation of four different features that describe the motion of beating cardiomyocytes in the feature movies. The intensity in the resulting feature movies represent the calculated feature. The features are calculated for all time steps, the top panel indicates three different frames, taken at t=1.6, t=3.1 and t=4.6 seconds. The colorbar on the right applies to all the frames. The bottom panel indicates the intensity profile for each feature plotted along the dashed line in blue.*

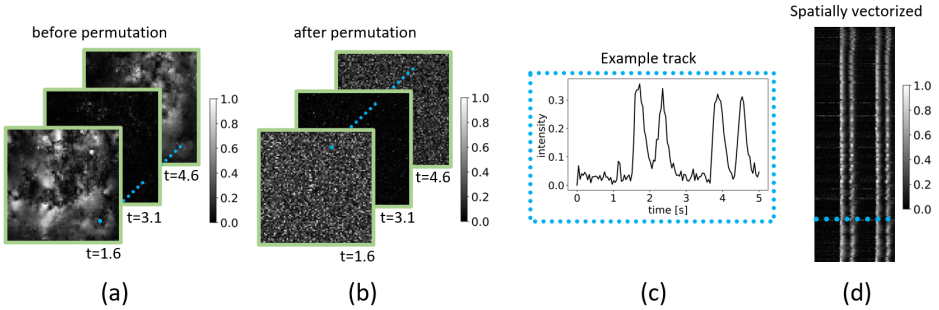

Figure 10: *Experiments to determine the contribution of the spatial dimension, indicated using the interframe displacement as channel. (a) Original 3D input containing two spatial and one temporal dimension. (b) The result after permuting the spatial location. The time dimension is unaltered, but tracks are relocated to a random location in the image. (c) Plot of the track located at the dotted blue line in (a), (b) and (d). The temporal dimension is the same, but the track is found at a different location. (d) Result after vectorizing the spatial dimension, the resulting 2D image here contains 350 tracks ($T_N = 350$).*

normalized between zero and one, the distant from median is normalized using a value of 15, the interframe displacement with a value of 4.2. These values are heuristically determined based all on the values that occur in the dataset. This was done to prevent normalization using an outlier in the datasets.

4.3.1 Importance of spatial dimension

The input data with calculated features contains both spatial and temporal information. To investigate the contribution of the spatial dimension, two experiments were performed. The first used the 3D input volumes with calculated features (see Figure 9 and 10a) but permutes the spatial locations of the pixels. As such, each track contains the same information along the temporal dimension, but is relocated to a random location in the feature movie (see Figure 10b,c). During training, the spatial location of the tracks is changed as data augmentation, as such the spatial location is different each time a feature movie is presented to the network. The second experiment takes the 3D features movies and vectorizes the spatial dimension to generate a 2D image. Each row contains a track from a different spatial location, the columns indicate the temporal dimension (see Figure 10d). The resulting image can be constructed using all the tracks in the 80x80 image, or only a selection of the tracks. In selecting which tracks to use, the tracks are sorted based on the maximum interframe displacement value. The T_N tracks with the largest interframe displacement are used to construct the 2D image. The effect of omitting tracks is explored by varying the value of T_N, values of 6000, 3000, 1500, 750 and 350 are used during this study. The order of the rows is randomly permuted as data augmentation.

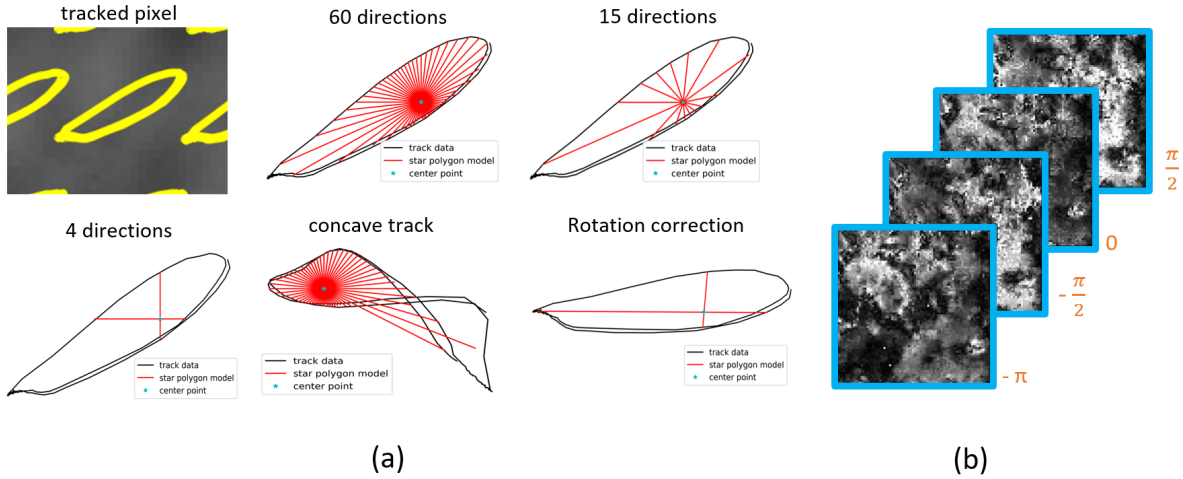

Figure 11: *Star-polygon representation of the data. (a) top left panel indicates the shape described by the x and y coordinates of the track. Using the star-polygon representation with different numbers of radial directions, the shape of the track can be described. The same shape is indicated for 60 directions (top middle panel), 15 directions (top right panel) and four directions (bottom left panel). For concave shapes, star polygon models fail to describe the shape accurately, as depicted in the bottom middle panel. Furthermore, the track can be rotated such that the largest distance from center to boundary is along the $-\pi$ direction (bottom right panel). As this direction is different for each track, the orientation of the sample under the microscope is lost. (b) the resulting output image containing channels for the different radial directions. The intensity now indicates the distance from the center to the boundary of the track for each radial direction.*

4.3.2 Star-polygon and levelset transformation

Because of the cyclic nature of the contraction-relaxation cycle, the x and y-coordinates of each track describe a closed shape. We propose two data transformations to investigate the importance of the temporal dimension of the data, by classifying the shape of the track alone, without any temporal information. The first method is a star-polygon model, the second deploys a distance map to describe the shape, similar to levelset descriptions of shapes.

Star-polygon

Star-convex polygons were originally proposed to segment cell nuclei from 2D fluorescence microscopy images [61]. Starting from a center point inside the object, a star-convex polygon describes the distance to the boundary along K radial directions [61, 62]. As such, convex shapes like cells, or cell nuclei can be accurately segmented [61]. Although not all our tracks describe a convex shape, we deploy star-polygons to describe the shape of each track as a vector of length K (see Figure 11a). This vector contains the distance from the center of each track, to a point on the boundary of the track along a specified direction. To that end, the center is determined by introducing a Fibonacci lattice on top of the track. The x and y-coordinates of the lattice are defined as:

$$(x_i^f, y_i^f) = (\frac{i}{\Psi} \bmod 1, \frac{i}{N}) \quad \text{for } 0 \leq i < N$$

where N is the number of points in the lattice, mod is the modulo operator and Ψ is the golden ratio: $\Psi = \frac{1+\sqrt{5}}{2}$. This study uses a lattice of $N = 50$ points. For every point in the lattice as putative center, the x and y-coordinates of the track are transformed to polar coordinates to get the polar angle

$$\theta_i = \tan^{-1}\left(\frac{y - y_i^f}{x - x_i^f}\right)$$

Here x and y indicate coordinates of the track and x_i^f and y_i^f indicate the coordinates of point i of the Fibonacci lattice. The center point that maximizes the standard deviation of the polar angle θ is picked as center of the track. Intuition behind this choice is that a good center point will have the x and y-coordinates of the track in all possible directions. The distance towards the boundary is determined using linear interpolation by transforming the x and y-coordinates to polar space using the determined center point.

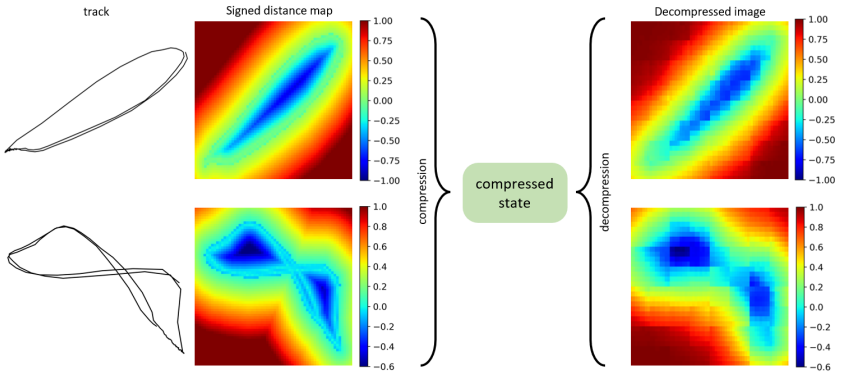

Figure 12: *Transformation of a track to a distance map. The left panel indicates two different tracks. Each track is represented as a signed distance map, in which negative distances indicate that the point is inside the track. The distance map is compressed and decompressed, the decompressed image is indicated on the right. The compressed representation now represents the shape of the track similar like the cutting plane defines a shape in levelsets.*

The final vector containing the distance to the track in all radial directions is normalized between [0,1], the shape of the track is therefore mapped to fall inside a circle with radius 1. As a result of this normalization, only the shape, and not the amplitude of the motion is captured. Final output is a 2D image with K channels, where the pixel value represents the distances to the track along radial direction k (see Figure 11b). These images are classified as healthy or aberrant with the use of a CNN. To investigate the effect of the parameter K, experiments with the following number of radial directions are performed: 4, 8, 15, 30, 60. Using the representation with 60 radial directions, additional experiments are performed in which each track is rotated such that the $-\pi$ direction contains the maximum distance to the boundary (see Figure 11b). As a result, the orientation of each track is different, and the orientation of the sample under the microscope is masked.

Levelset

Star-polygons are not appropriate to describe non-convex shapes. Therefore, a second method, that is based on levelsets, is introduced to describe the shape of the tracks. For this purpose, a square grid of 64x64 pixels is placed on top of each track. The pixel values represent the distances from that pixel to the closest point on the track, called a distance map. Using linear interpolation, the number of points of the track is increased from 120 to 400, to make the distance map more accurate. Finally, using a flooding-algorithm starting from the top left pixel of the grid, pixels inside the track get a negative sign and pixels outside the track get a positive sign, creating a signed distance map for each track (see Figure 12). The distance map is computed to a distance of +4 outside the track, and -2 inside the track. Larger or smaller values are clipped to these limits. This allows for normalization of the maps in the range [-1,1], while values of 0 remain 0 after normalization.

The height where the signed distance map equals zero, describes the shape of the track. Similar to levelsets, we seek to find a function L_S which is able to describe all possible shapes in the data. Depending on the height at which you asses L_S you get a different shape. To find the function L_S, we deploy a autoencoder neural network. Autoencoders, often used for data compression, take an image as input and try to reconstruct that same input after a bottleneck [63]. The bottleneck reduces the amount of parameters, allowing for the images to be compressed (see Figure 12). Similarly, we propose a convolutional autoencoder, which takes the 64x64 pixels signed distances map as input, and compresses them to 20 parameters. The network now represents the function L_S, the 20 parameters define the height at which to cut the function L_S to describe the shape of a specific track.

The autoencoder is trained on a random sample of 142 000 tracks taken from the entire dataset. After compressing the signed distance maps, each movie can be represented as a 2D image of 80x80 pixels (see Figure 13), consisting of 20 channels which are classified with a CNN. Each channel contains one of the parameters that define the cutting plane of L_S. Unlike the star-polygon model, where each shape is mapped to a radius of 1, the size of each shape persists here, because distance maps are used that carry information about the size and therefore the amplitude of motion of each track.

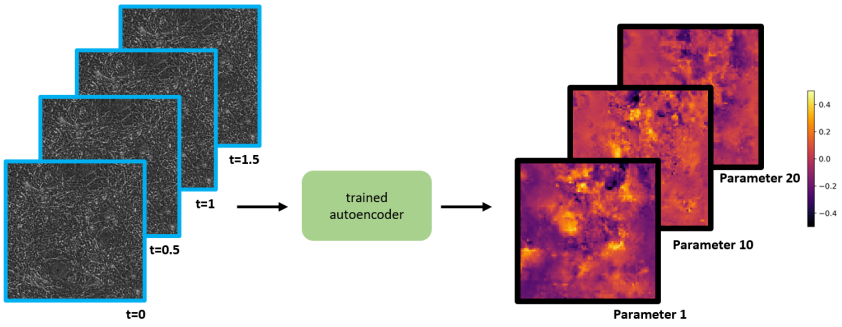

Figure 13: *Resulting output after representing each track in the image as a vector of 20 numbers. The different channels together describe the shape of the track. The compression is done by training an autoencoder.*

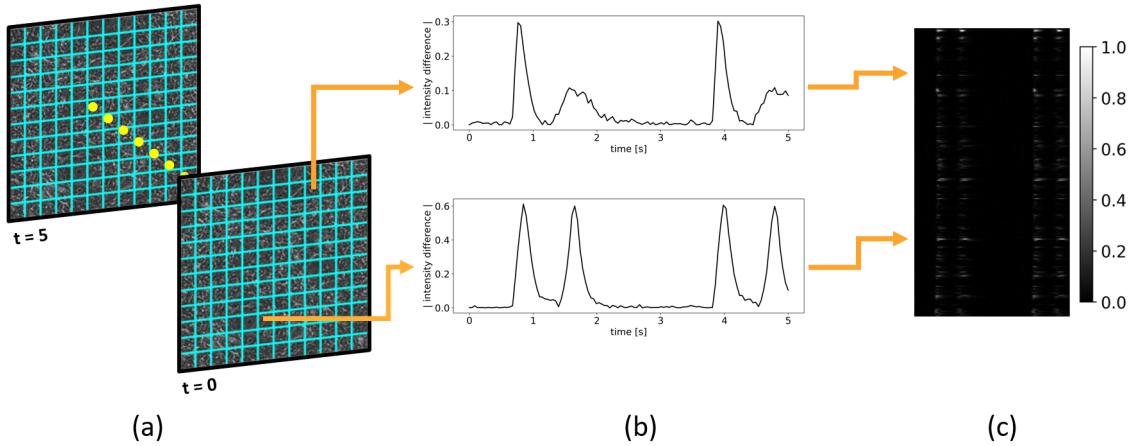

Figure 14: *Extracting the motion of beating cardiomyocytes without image registration by averaging the pixel intensity over time in small local neigborhoods. (a) A square grid is placed on top of the input images. The intensity in each grid is averaged over time. (b) After averaging, the change in intensity in one of the grid cells is plotted over time. (c) The final output of averaging the pixel intensity. Each row contains the average intensity for a different grid cell. The resulting output image is classified with a CNN.*

4.4 Local averaging

Image registration is a time consuming step in the processing pipeline. Therefore, a processing pipeline without image registration was developed. For this purpose, raw movies from the microscope without any preprocessing steps are used. Using a square grid, the average pixel intensity inside each square is determined for each time step (see Figure 14a). The motion of the beating cardiomyocytes causes pixels to enter or leave a grid cell. Because of this, the dynamics of the motion can be captured by taking the absolute value of the interframe change in the average intensity in each square of the grid (see Figure 14b). The averaging operation is implemented in the CellsOnCHIP ImageJ plugin by Ihor Smal. Final output format is an image in which each row contains the absolute change in the average intensity of a single grid cell for all time points along the columns (see Figure 14c). As before, movies are split in multiple parts, each of 120 frames length to generate more training for the CNN. Before classification each image is normalized to be in the interval $[0, 1]$. The effect of the grid size is determined by running the experiment for a grid size of 20x20, 40x40, 60x60 and 80x80 pixels.

4.5 Network architectures

Throughout this study, four different network architectures are used to perform the classification task. Moreover, an additional autoencoder architecture is used to encode the levelset images. Group normalization is used to facilitate training on small batch sizes [64]. For consistent comparison between the different architectures, all networks are implemented in Pytorch and trained with a batch size of 5 on a single NVIDIA Quadro T1000 GPU with 4 GB of memory. All architectures used to perform the classification were trained for 70 epochs using a cross entropy loss function (equation 3). The Adam algorithm with an initial learning

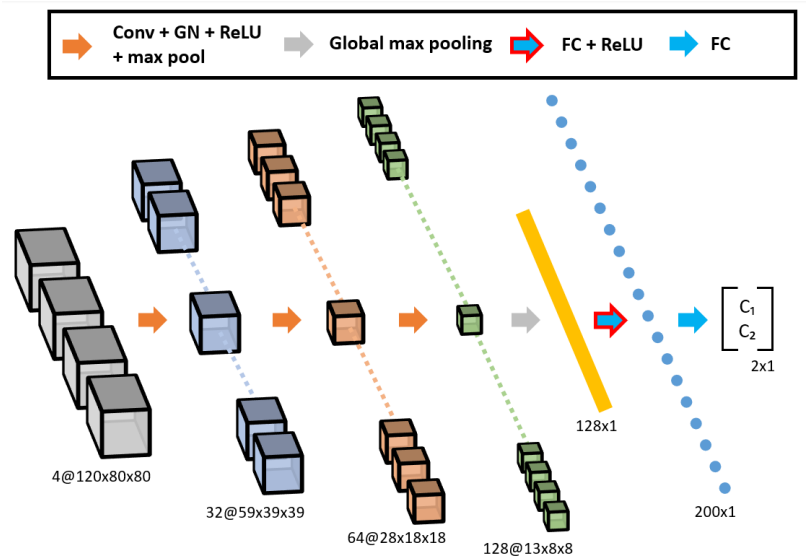

Figure 15: *3D-CCNET architecture, using 3D convolutions. The different channels are indicated using the different cubes. Convolution followed by group normalization and ReLU activation is used. To reduce the size of the output, max pooling with a 2x2x2 kernel size is used. Following a global max pooling layer, two linear layers are used to predict each input video as healthy or aberrant. The size of each block is indicated in the figure, the number before @ indicates the number of channels. Conv stands for convolution, GN for group normalization and FC for a fully connected layer.*

rate of 0.001 was used as optimizer [65]. Results are generated on the entire dataset using a fourfold cross validation. As we split each feature movie in equal parts of 120 frames to generate more training data, the cross validation puts all the parts originating from the same movie into the same fold. As such when the first 120 frames of a sample are in the training set, the subsequent 120 frames can not be in the test set of the cross validation.

4.5.1 3D-CCNET

To classify feature movies, the 3D cardiomyocyte classifier network (3D-CCNET) uses convolutions along the temporal and two spatial dimensions. The architecture is depicted in Figure 15. The feature movies have four channels, containing the calculated features from Figure 9. The kernel size for the 3D convolution is 3x3x3, max pooling is performed with a 2x2x2 kernel. Group normalization where each group contains four channels is used. Final output is a score for both classes (healthy and aberrant), the largest value determines the final prediction. Data augmentation consists of rotation, vertical flipping and horizontal flipping along the spatial dimensions.

4.5.2 2D-CCNET

To explore the effect of removing the spatial dimension, a 2D convolution cardiomyocyte network (2D-CCNET) is used (see Figure 16). The input now contains a different track in each row, the columns represent the time dimension, only the interframe displacement is used as input channel. Convolution with a 5x5 kernel is used, max pooling is performed with a 4x2 kernel. The groups used for group normalization contain four channels. As data augmentation the order of the rows was randomly permuted.

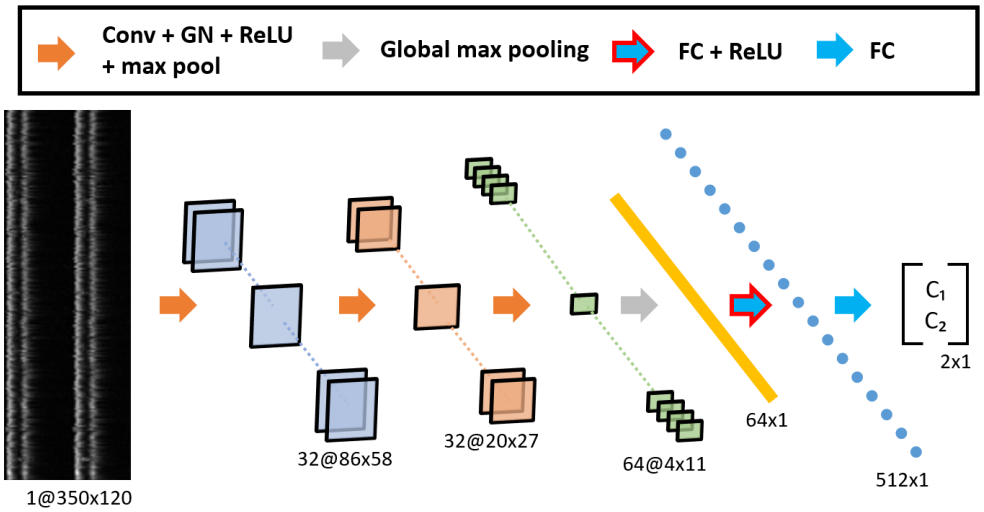

Figure 16: *2D-CCNET architecture used to classify the motion of beating cardiomyocytes as healthy or aberrant. The size of each image is indicated in the figure when an input image containing 350 tracks is used, the number before @ indicates the number of channels. Conv stands for convolution, GN for group normalization and FC for a fully connected layer.*

4.5.3 Star-polygon network

To study the contribution of the temporal dimension, a network is used which performs 2D convolution along both spatial dimensions with a 3x3 kernel (see Figure 17). The channels represent the k different radial directions in the star-polygon representation. Max pooling is performed with a 2x2 kernel. Note the similarity to the 2D-CCNET architecture, besides an additional convolutional layer which was added for increased performance. Data augmentation consists of rotation, vertical and horizontal flipping of the input images.

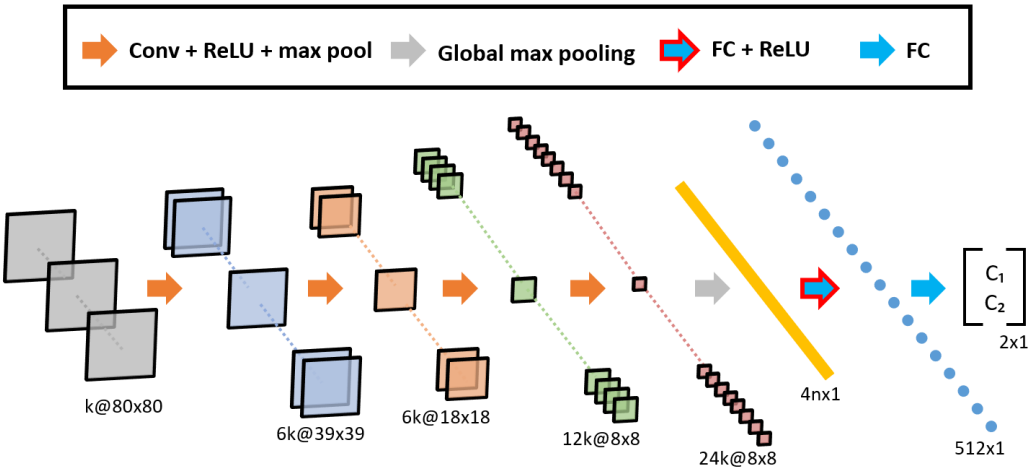

Figure 17: *Network architecture that performs the classification after the star-polygon or levelset transformation. A 2x2 kernel is used during max pooling, a 3x3 kernel is used for the convolutions. The size of the images is indicated in the figure, the number before the @ indicates the number of channels. Conv stands for convolution and FC for a fully connected layer.*

4.5.4 Levelset transform networks

The contribution of the temporal dimension is also studied using a levelset transformation. The levelsets are compressed using an autoencoder network depicted in Figure 18. During training, the signed distance maps are horizontally and vertically flipped. Training is performed with a batch size of 512, the mean squared error between the input and decompressed output is used as loss function. Convolution with a kernel size of

3x3 and a stride of 2 is used to compress the images. To decompress the images, transposed convolutions (up-convolutions) with a 2x2 kernel and a stride of two are used. The compressed state of the autoencoder facilitates the construction of a 80x80 image with 20 channels, without a temporal dimension (see Figure 13). Subsequently, classification into healthy or aberrant is performed using the architecture in Figure 17, with *k=20* channels. Data augmentation consists of rotation, vertical and horizontal flipping of the input images.

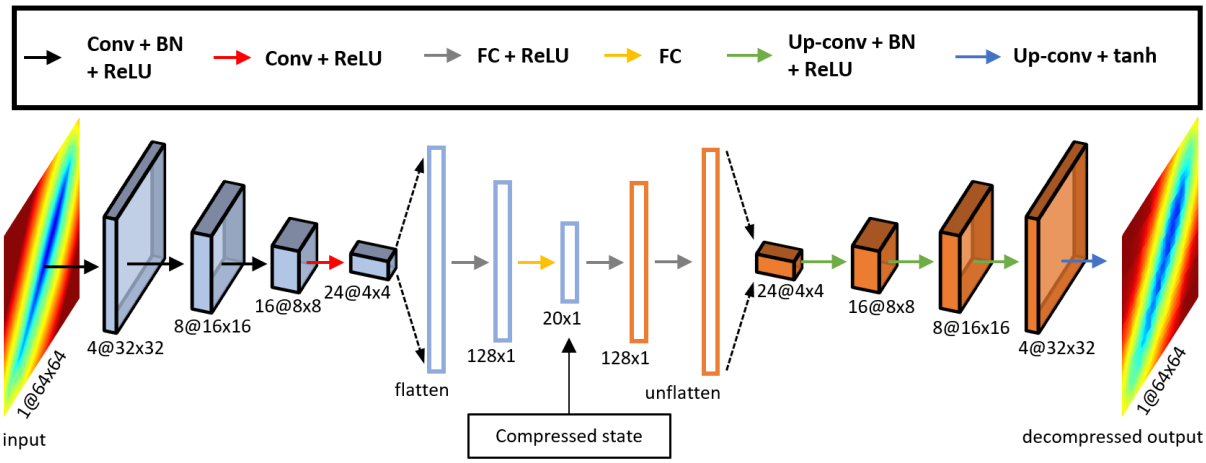

Figure 18: *Autoencoder architectures used to compress the signed distance maps. The distance maps are compressed to a vector of 20 numbers, which represent the cutting plane of the levelset function L_S which is modeled by the network. Conv stands for convolution, performed using a 3x3 kernel with a stride of 2. Up-conv stands for up-convolutions performed with a 2x2 kernel and a stride of 2. FC stands for a fully connected layer, and BN for batch normalization. A hyperbolic tangent activation function is used in the final layer to enable the output to be between -1 and 1.*

4.5.5 CaTNET

Convolutions depend on local context and are not able to relate each specific track from an image to all other tracks. To alleviate this issue, we propose a cardiomyocyte transformer network (CaTNET), based on ViT architectures. Input images in which the spatial dimension is vectorized with 750 tracks are used to perform the classification. Only the interframe displacement was used as input channel. The image is divided into patches of 50x3 pixels (see Figure 19). The patches are flattened and projected using a linear layer, followed by two consecutive single headed transformer blocks (see Figure 3). Linear projection and cutting the image in patches is implemented as a convolution with a kernel size of 50x3. Finally, a max pooling layer along the time dimension and a fully connected layer is used to construct the final output of the architecture. The rows of the input image are randomly permuted as data augmentation.

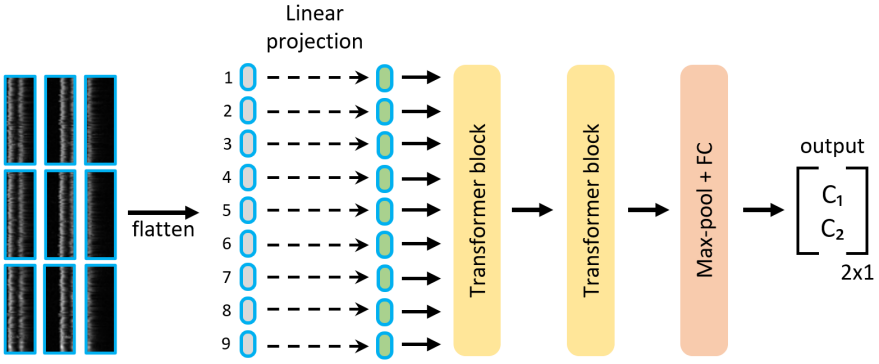

Figure 19: *CaTNET architecture to classify images containing different tracks along the rows, and the time dimension along the columns. The image is divided into patches of 50x3 pixels. Before the single headed transformer blocks, a linear projection is performed on each patch. The final output is constructed using a max pooling operation followed by a fully connected layer (FC).*

4.6 Grad-CAM and SHAP

Explanations using Grad-CAM and SHAP are constructed for the 3D data containing two spatial and one temporal dimension (see Figure 9). Both methods are used to explain the 3D-CCNET classifier. Grad-CAM is implemented as described by the original authors [19]. To extract the gradients in Pytorch, gradient hooks are used. The obtained heatmaps are up sampled with the use of linear interpolation. The mean L_2 and L_1 distance between the obtained heatmap and the four different input channels is calculated to determine the importance of the different channels $\mathcal{C}_i$. The mean L_2 and L_1 distance are defined as:

$$L_2 = \frac{1}{N} \sum_{x,y,t \epsilon N^{X,Y,T}} (H[x,y,t] - \mathcal{C}_i[x,y,t])^2, \qquad L_1 = \frac{1}{N} \sum_{x,y,t \epsilon N^{X,Y,T}} |H[x,y,t] - \mathcal{C}_i[x,y,t]|$$

here, H indicates the heatmap generated by Grad-CAM, N is the total number of pixels, x and y indicate the spatial dimension and t the temporal dimension. As SHAP contains a Pytorch implemented, it is directly installed from the github of the authors [20]. As the preliminary Pytorch implementation of SHAP does not support group normalization, a 3D-CCNET architecture with batch instead of group normalization is used. The background model of SHAP is constructed with 6 images due to memory constraints. The 3D-CCNET trained on all available data is used to generate explanations, as such the explanations hold for the entire dataset.

4.7 CA-LIME

To obtain a more thorough understanding between the motion of healthy and aberrant beating cardiomyocytes, we propose CA-LIME, an interpretability method which does not depend on the use of superpixels. After image registration, a large set of similar motion patterns from different spatial locations is obtained and each track contains multiple contractions because of the periodic nature of the motion. Similar information is presented in subsequent beats, or tracks from different spatial locations. As a result, superpixels fail to mask features successfully (see Figure 20a). Even more, a superpixel based approach indicates the importance of different parts of the contraction, but not how they differ between healthy and diseased cardiomyocytes.
As a solution, CA-LIME does not depend on superpixels, but uses the interframe displacement to automatically detect the contractions and relaxation peaks (see Figure 20b). Using handcrafted features, perturbations to the detected regions can be defined, which are applied to each track and period of the instance x that is explained (see Figure 21). Currently, six handcrafted features M_H are implemented that can be perturbed: contraction and relaxation peak height, upwards contraction and relaxation time and finally downwards contraction and relaxation time.
A continuous feature vector z' determines which features will be perturbed and the size of the perturbation, resulting in the perturbed image z (see Figure 21). Positive values in z' indicate an increase of peak height or time interval, whereas negative numbers indicate a decrease of the feature. A value of zero corresponds to an unperturbed feature. How much the peak height or time interval is changed, is indicated by the number in the feature vector, 0.3 indicates a 30% increase. Peak heights are varied between -30 and +30%, time intervals between -70 and +70%. Similar as in LIME [17], an explainable linear regression model G is trained in correspondence with equation 4 and 5 with π_x defined as:

$$\pi_x = 1 - \frac{1}{M_H} \sum_{j=1}^{M_H} |z'_j|$$

The following linear model with parameters ϕ is used as the surrogate model:

$$G(z') = B(x) + \sum_{i=1}^{M_H} \phi_i z'_i \tag{7}$$

which is similar to equation 6, but the choice of $\phi_0 = B(x)$ ensures the local accuracy property from SHAP values, albeit only for sample x and not for all samples in the dataset, as this would require getting the expectation value of the black box model. Moreover, as a result of this choice, a positive value of ϕ_i after fitting, corresponds to a feature i of which an increase in its attribute (positive value of z'_i) increases confidence for the class that is explained. A negative value of ϕ_i corresponds to a feature i of which the attribute should be decreased (negative value of z'_i) to increase confidence in the class that is explained. Equation 7 is implemented by subtracting $B(x)$ from all the predictions, before fitting with zero offset ($\phi_0 = 0$). To fit the surrogate model, 1000 perturbed samples are generated. The method is used to generate explanations for the 2D-CCNET trained on images that contain 750 tracks.

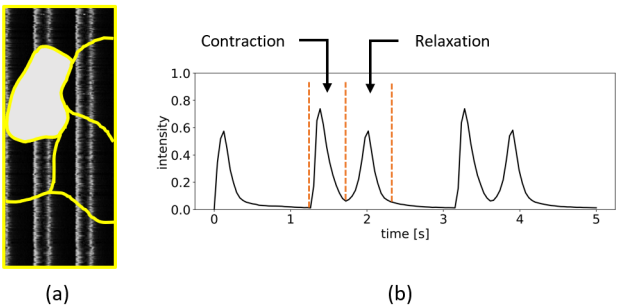

Figure 20: *Superpixels for periodic data and the detection of contraction and relaxation peaks. (a) Input image containing different tracks along the rows, pixel values indicate the interframe displacement. In yellow different superpixels are defined, the grey superpixel is turned off. Defining superpixels on top of the image is ineffective as similar contraction profiles can be found in other rows or the subsequent beat in the same row. (b) Result of averaging all the rows from (a). The contraction and relaxation peak are indicated for one of the beats.*

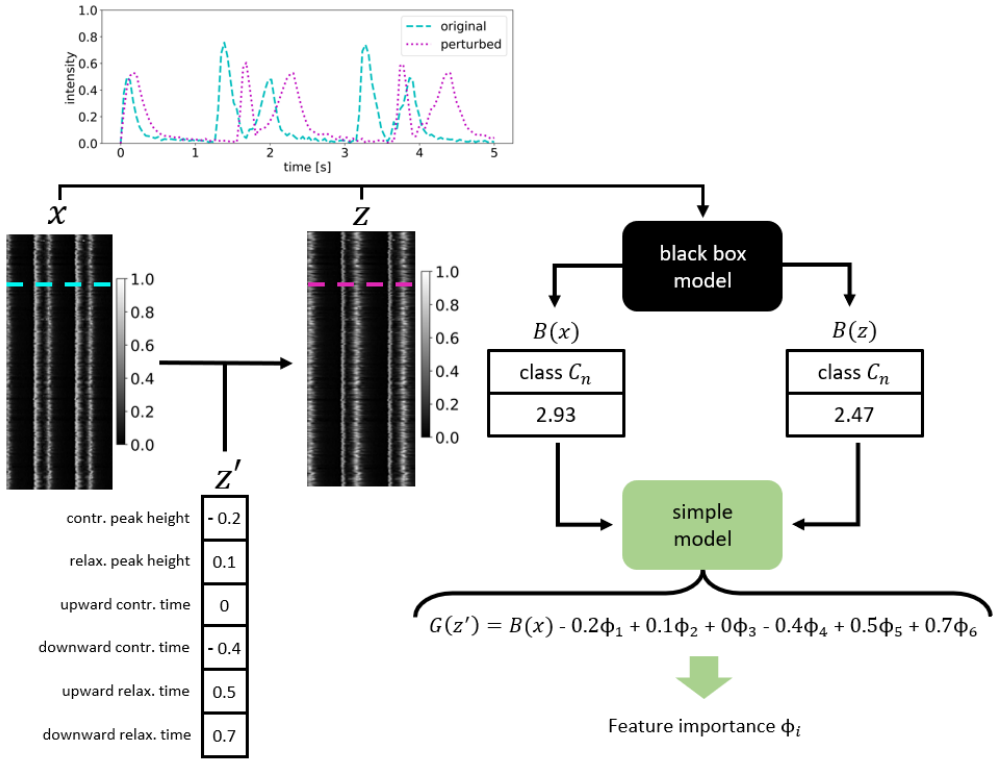

Figure 21: *Generation of explanations using CA-LIME. Unperturbed sample x is indicated on the left. The continuous feature vector z' determines if a feature is increased (positive numbers) or decreased (negative numbers). The feature vector is used to generate the perturbed sample z. The panel on top indicates the example track along the dashed line before and after the perturbation. Both x and z are passed through the black box model, the resulting output for the class of interest $C_n$is used to train a linear regression model, indicating the importance for each feature.*

Figure 22: *Accuracy and F1 score of the random forest classifier for different number of trees. The vertical axis indicates the obtained score in %. The best performance is obtained using nine decision trees, resulting in 85% accuracy and an F1 score of 87%.*

5 Results

Classification of the motion of beating hiPSC-CMs as healthy or aberrant is assessed using multiple experiments. The first method utilizes an optical flow analysis yielding 19 parameters about the pressure and temporal dimension of the motion. The second method uses image registration adopting both spatial and temporal information. Subsequently, a deep learning algorithm is trained to classify the motion based on the spatial and temporal information.

5.1 Decision tree classifier

First, we asses an optical flow based analysis resulting in 19 parameters that describe the motion (see Figure 7). A random forest of decision trees, ranging from 1 to 30 trees is used to perform the classification. The best performance was obtained using nine decision trees, resulting in 85% accuracy and a F1 score of 87% (see Figure 22). Increasing the number of trees beyond 19 resulted in a decrease of the accuracy and F1 score.

To test the importance of each of the 19 parameters, the Gini importance metric is used. The five most important features and their normalized Gini importance are shown in Table 1. The most important features relate to the contraction and relaxation time and the beating frequency of the cardiomyocytes. The summed importance of the five most important features equals 0.50, indicating that these five features alone are responsible for half of the reduction of the Gini impurity loss during training of the random forest.

Feature	Normalized Gini importance
Contraction time [s]	0.112
Relaxation time [s]	0.105
Beats per minute	0.100
Max pressure ratio	0.097
Time between contraction and relaxation [s]	0.086

Table 1: *Top 5 most important features determined by the random forest classifier. The values indicate the normalized Gini importance. The five shown features together have a summed Gini importance of 0.50.*

5.2 Deep learning approach

Besides the machine learning approach using decision trees, a deep learning approach is investigated which utilizes both the spatial and temporal dimension available in the data. For this purpose, image registration is used to extract the motion of the beating cardiomyocytes. The registration data is used to calculate the distance and angle from the median x and y-coordinate, and the distance and angle from the previous frame (see Figure 9). The classification is done using the 3D-CCNET (see Figure 15). Using a fourfold cross validation, an accuracy of 78% and F1 score of 80 % is obtained.

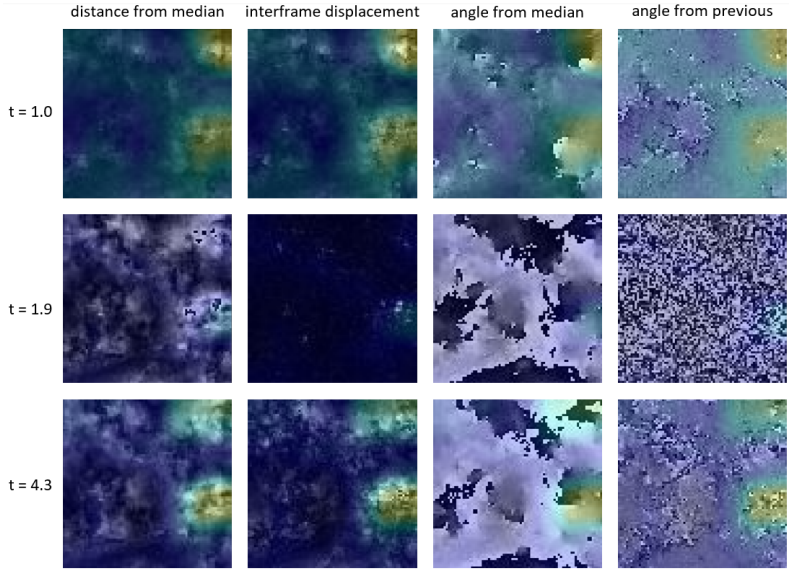

Figure 23: *Heatmaps generated by Grad-CAM depicted as overlay for the different input channels (along the columns) for a healthy sample. The rows indicate the results for different time points in the feature movie. At each time point a single heatmap is generated for all input channels. Red regions indicate more importance, blue regions indicate less importance.*

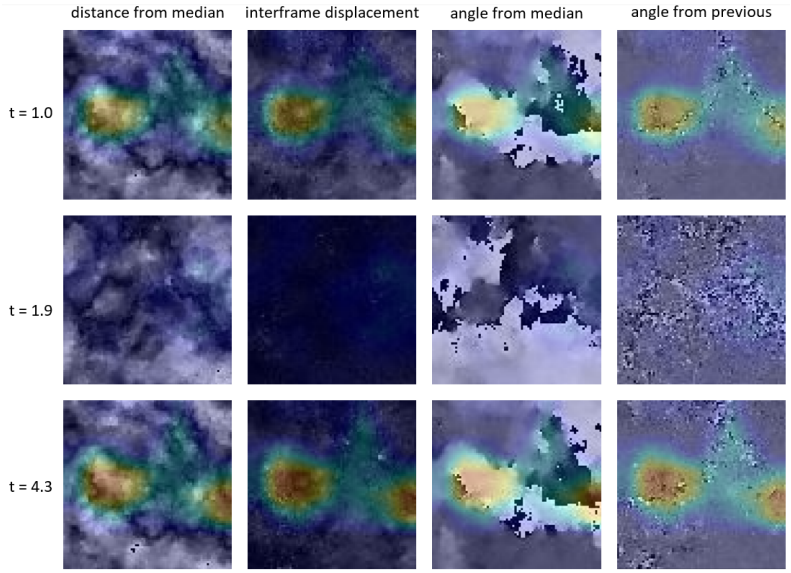

Figure 24: *Heatmaps generated by Grad-CAM depicted as overlay for the different input channels (along the columns) for an aberrant sample. The rows indicate the results for different time points in the feature movie. At each time point a single heatmap is generated for all input channels. Red regions indicate more importance, blue regions indicate less importance.*

The channels that represent angles are normalized between zero and one. As a result, -180 degrees corresponds to zero and +180 degrees to 1 in the feature movies. These angles are close together but the intensity values are far apart. To solve this issue, angles were represented using the x and y-coordinates on the unit circle corresponding to the angle. Each angle was now represented as two seperate channels. Using a fourfold cross validation, an accuracy and F1 score of 77% was obtained. Based on this result, the mapping of angles to the x and y-coordinates on the unit circle is not used in further experiments.

5.2.1 Grad-CAM generated explanations

To explain the classification made by the 3D-CCNET and possibly improve the performance, Grad-CAM [19] is used to generate a heatmap depicting which regions of the feature movies are important for the classification

	mean L_2 distance		mean L_1 distance	
Input feature	healthy	aberrant	healthy	aberrant
Distance from median	0.048 ± 0.03	**0.062** ± 0.03	0.145 ± 0.06	0.183 ± 0.05
Interframe displacement	**0.042** ± 0.02	0.063 ± 0.02	**0.132** ± 0.04	**0.178** ± 0.04
Angle from median	0.225 ± 0.02	0.246 ± 0.03	0.392 ± 0.03	0.408 ± 0.03
Angle from previous	0.227 ± 0.03	0.202 ± 0.03	0.379 ± 0.02	0.394 ± 0.02

Table 2: *Similarity between the heatmap and the different input channels. Results indicate the average value of the entire dataset. Values in bold indicate the lowest value along the columns. Standard deviations are indicated behind the ±. The results indicate the interframe displacement and the heatmaps are most similar to each other.*

task. The results of Grad-CAM for an input image containing the motion of healthy cardiomyocytes is depicted in Figure 23. Here, the gradient with respect to the class healthy is taken. Red regions correspond to more important regions, blue indicates the least important parts. Similarly, the results for an aberrant sample is shown in Figure 24, where the gradient with respect to the aberrant class is taken.

The resulting heatmaps for both samples seem to indicate spatial regions where the displacement from previous is large. As such, these spatial regions contribute most to the classification. To quantify this observation, the mean $L_1$and L_2 distance between the heatmaps and each input channel is calculated and averaged over the entire dataset (see Table 2). The L_1 distance of the heatmap is smallest with respect to the interframe displacement for both the healthy and aberrant samples. The L_2 distance of the heatmap with respect to the interframe displacement is the smallest for the healthy samples. For the aberrant samples, the distance from median has the smallest L_2 distance to the heatmap, with a difference of 0.001 compared to the interframe displacement.

5.2.2 Explanations by SHAP

Similar like Grad-CAM, we use SHAP [20] to generate explanations for the pretrained 3D-CCNET. The maximum absolute SHAP value for each frame is used to indicate the importance. Unlike Grad-CAM, SHAP returns separate importance scores for each of the input channels. Explanations for a healthy sample (Figure 25) and an aberrant sample (Figure 26) are generated. The highest SHAP sample is obtained for the interframe displacement channel for both samples. The SHAP values increase during contractions of the cardiomyocytes, but the maximum absolute SHAP values vary strongly between the different successive contractions. To quantify the importance of the different input channels, explanations for 30 input samples are generated using SHAP. Out of the 30 samples, in 26 the interframe displacement contained the highest SHAP value (see Table 3).

Feature	Occurence of highest SHAP value
Displacement from median	3
Interframe displacement	**26**
Angle from median	0
Angle from median	1

Table 3: *Number of occurences of the highest SHAP for the different calculated features, determined for 30 different samples picked from the dataset at random. The values in the second column indicate the number of samples in which the maximum absolute SHAP value was obtained for the four different input channels.*

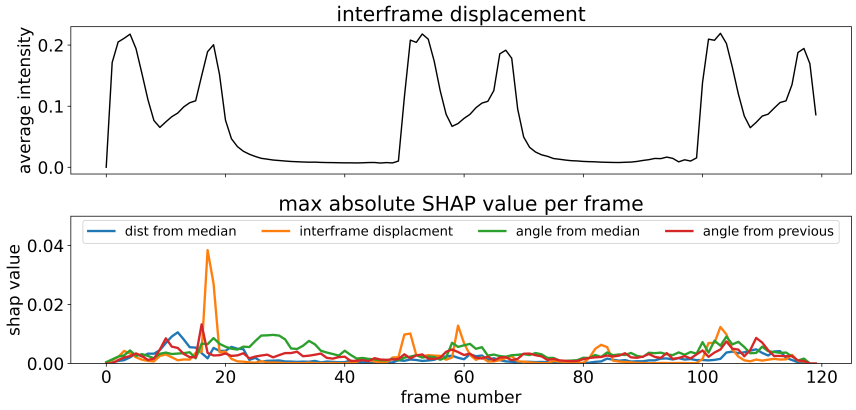

Figure 25: *SHAP values for a healthy input sample when the class healthy is explained. The top panel depicts the interframe displacement, to indicate which frames contain movement of the cardiomyocytes. The bottom panel indicates the maximum absolute SHAP value for each frame in the feature movies. Higher values indicate larger importance.*

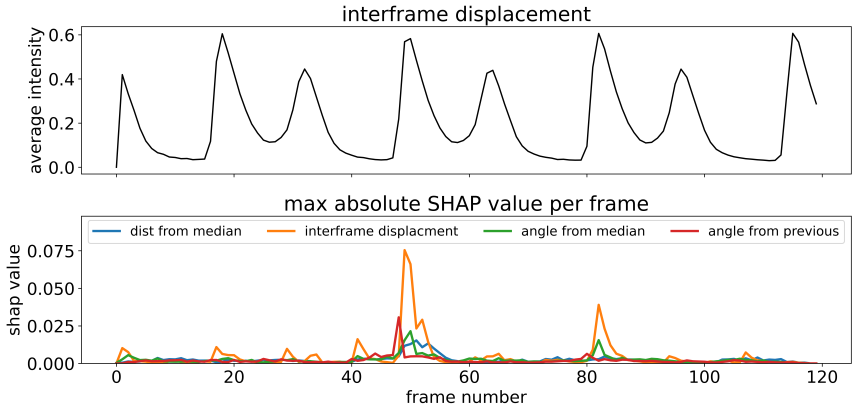

Figure 26: *SHAP values for a aberrant input sample when the class aberrant is explained. The top panel depicts the interframe displacement, to indicate which frames contain movement of the cardiomyocytes. The bottom panel indicates the maximum absolute SHAP value for each frame in the feature movies. Higher values indicate larger importance.*

5.3 Spatial dimension

Grad-CAM indicates that specific spatial regions contribute to the classification task in different amounts. To assess the importance of the spatial dimension for the classification task, two experiments are performed in which the classification is performed without the spatial dimension. Based on previous results, only the interframe displacement and angle from previous is used. The angle from previous is chosen because this metric still contains information about the shape of each track and does not depend on a fixed coordinate, like the median position. The first experiment determines the effect of permuting the spatial dimension (see Figure 27a). Before permuting the spatial dimension, an accuracy of 88.1% and F1 score of 88.4% is obtained. Performing the classification task after shuffling the spatial dimension results in an accuracy of 89.5% and F1 score of 90.0%. Shuffling the spatial dimension as form of data augmentation results in an accuracy and F1 score of 95%.

The second experiment vectorizes the spatial dimension, after which each row in the resulting input image contains a track from a different spatial location (see Figure 27b). The classification is performed using the 2D-CCNET network. During training, the order of the rows is randomly permuted. The number of rows in the image is varied between 6000 and 350 by selecting the tracks that contain the largest interframe displacement. Using 6000 tracks, an accuracy and F1-score of 96% is obtained (see Figure 27c). Reducing the number of tracks results in a decreased performance, but each experiment resulted in an accuracy and F1 score above 92%. The 2D-CCNET architecture uses a 5x5 kernel to perform the convolution. As the data represents track from different spatial locations along the different rows, we assessed the effect of using

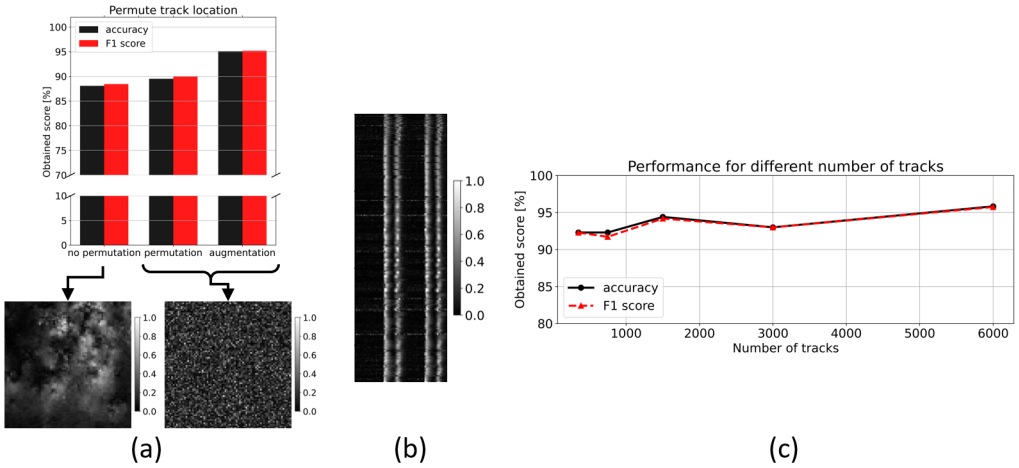

Figure 27: *Experiments to test importance of the spatial dimension. Results are indicated using the accuracy and F1 score (a) Randomly permuting the spatial location of the data. The top panel indicates the performance in case the spatial information is present, the locations are randomly shuffled, or the shuffling is added as data augmentation. The bottom panel indicates examples before and after permuting the spatial location. (b) Resulting image after vectorizing the spatial dimension in the data and using the 350 tracks that contain the largest interframe displacement. (c) Performance when the spatial dimension is vectorized for different number of tracks along the horizontal axis. An accuracy and F1 score above 92% were obtained independent from the number of tracks.*

a 1x5 kernel. The convolution is now performed over a single track. The resulting accuracay and F1 score are depicted in supplementary Figure S2a, b. Due to the inferior performance, a 5x5 kernel is used in further experiments.

5.4 Temporal dimension

Grad-CAM and SHAP indicate that frames during the contraction and relaxation are more important for the classification compared to frames with less displacement. To investigate the effect of omitting the temporal dimension, we propose two transformations that describe the shape of each track without any temporal component. The shape of each track is described as convex star-polygon (see Figure 11) or as a levelset using a signed distance map (see Figure 13).

5.4.1 Star-polygon representation

Using the network architecture depicted in Figure 17, the data containing the star-polygon representation is classified. Experiments for a different number of radial directions are performed (see Figure 28a). The best F1 score of 68.7% was obtained using 30 radial directions. Using eight radial the best accuracy of 67.6% was obtained. Despite that four radial directions do not describe the shape of the track accurately, an accuracy of 63.3% is obtained. To investigate this effect further, the distribution of the distance from the center to the boundary of the track for the entire dataset is depicted in Figure 28b. Along all radial direction, the distance is smaller for tracks belonging to healthy cardiomyocytes.

As the star-polygon model is normalized such that each track falls within a circle with radius 1, the observed distribution could indicate that healthy tracks are more elliptical compared to aberrant tracks. Tracks which are almost circular contain large radial distances to the boundary along all radial directions, whereas elliptical tracks contain large distances along two directions, and small distances along two directions orthogonal to it. To test this hypothesis, the major and minor axis of each track are determined (see Figure 29a). Using this, the distribution of the ratio minor/major axis is calculated for each track in the dataset (see Figure 29b) and no difference is observed. Indicating that healthy tracks are not more elliptical. Alternatively, the distribution of the star polygon model with four directions (Figure 28) could be explained by the orientation of the samples under the microscope, as the radial directions are in specified directions and therefore contain information about the global orientation of each track. To investigate this effect, a network is trained in which each track is rotated such that the longest distance from center to the boundary of the tracks is aligned with the $-\pi$ direction. After training with 60 radial directions, an accuracy of 56.3% and F1 score of 45.1% is observed.

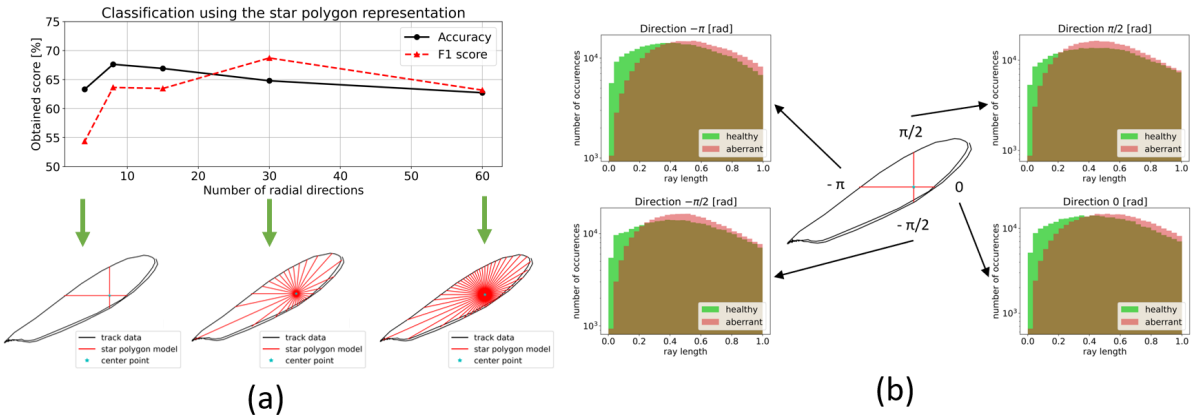

Figure 28: *Classification of the shape of each track using the star-polygon description. (a) Top panel indicates the accuracy and F1 score for different number of radial directions. The bottom panel indicates examples when 4, 30 or 60 radial directions are used respectively. Independent on the number of radial directions, accuracy and F1 scores below 70% are obtained. (b) Distribution of the distance from the center to the boundary of the track in case 4 radial directions are used. Healthy track have a higher occurrence of short distances, whereas aberrant tracks have a higher occurrence of longer distances.*

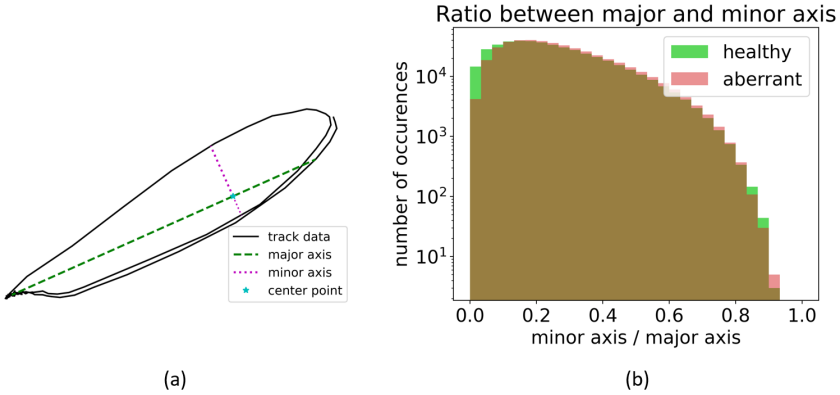

Figure 29: *The ratio between the major and minor axis of each track. (a) Example of a track in which the major axis is indicated in green and the minor axis is indicated in pink. The minor axis is formed by the two radial directions with a difference of π radians that together form the smallest distance from center to boundary. (b) Distribution of the ratio between the minor and major axis calculated over all tracks.*

5.4.2 Levelset representation

The star-polygon representation facilitates the description of convex shapes. As the tracks in our data also contain non convex shapes, a levelset like description of the shapes is used. The shape of each track is described as a signed distance map. The complex function L_S, able to describe all possible shapes, is given by an autoencoder (see Figure 18). The 20 parameters in the bottleneck layer represent the cutting plane of L_S to describe each specific shape. The resulting compressed signed distance maps are used to classify the data as healthy or aberrant using a convolutional architecture (see Figure 17), resulting in an accuracy of 73.3% and F1 score of 74.1%.

5.5 Motion extraction using intensity averaging

Image registration is a time consuming step in the processing pipeline. We investigated the extraction of the motion of beating cardiomyocyte by calculating the change in the average pixel intensity of the raw microscopy data inside square grid cells (see Figure 14). The resulting traces of the change in average intensity over time are used to construct 2D images, used for training and testing the 2D-CCNET (Figure 16). Experiments are performed for different dimensions of the grid, ranging from 20x20 to 80x80 pixels. The best accuracy and F1 score were obtained using a 20x20 pixel grid, resulting in 86.8% and 87.0% respectively (see Figure 30). Using a 60x60 pixel grid resulted in the lowest accuracy of 84.9% and an F1 score of 85.8%.

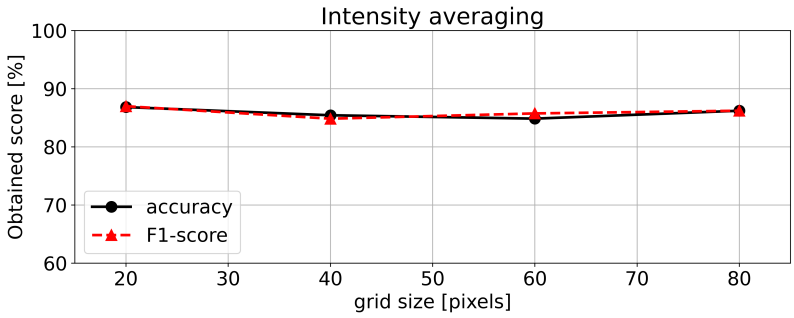

Figure 30: *Accuracy and F1 score using an averaging method with a square grid to extract the motion of the beating cardiomyocytes from the raw microscopy data. The vertical axis indicates the obtained score, the horizontal axis gives the size of the square grid cell used in the intensity averaging. The best score is obtained using 20x20 pixel grid cells, resulting in an accuracy of 86.8% and F1 score of 87.0%.*

5.6 Transformer architecture

Unlike transformer layers, convolutions depend on global context to classify between healthy or aberrant beating cardiomyocytes [47]. The performance of our transformer architecture CaTNET (Figure 19) is therefore compared to the 2D-CCNET (Figure 16) which utilizes convolutional layers. The CaTNET is similar to ViT architectures [47]. Tenfold cross validation is used to quantify the performance on data in which the spatial dimension is vectorized. On all metrics, the 2D-CCNET outperformed the CaTNET, reaching an accuracy of 97.5% compared to 88.2% for the transformer architecture. A similar difference in performance is observed for the precision, recall and F1 score (see Table 4). Training the 2D-CCNET took on average 180.2 ± 0.9 seconds compared to 148.9 ± 1.5 seconds for the CaTNET. Predicting a single sample during testing time required on average over the tenfold cross validation 12.3 milliseconds using the 2D-CCNET architecture compared to 10.7 milliseconds for CaTNET.

Network	accuracy	precision	recall	F1
2D-CCNET	**97.5**	**98.2**	**97.1**	**97.4**
CaTNET	88.2	84.5	95.4	89.0

Table 4: *Result of the 2D-CCNET and CaTNET determined using a tenfold cross validation. The 2D-CCNET outperformed the transformer architecture on all metrics. Numbers in bold indicate the best score along the columns.*

5.7 CA-LIME interpretability

A superpixel based approach to generate explanations, like LIME [17] or SHAP [20] is not effective for masking out features in case many periodic beats and similar contraction-relaxation profiles are presented in the image. We therefore propose CA-LIME, which perturbs all tracks and all subsequent contraction-relaxation cycles in the input samples. To show the effectiveness of our approach, CA-LIME is used to generate explanations for four different input samples, one sample from each aberrant condition (Figure 31a,b,c) and one healthy sample (Figure 31d).

Treatment with Endothelin-1 is known to cause a hypertrophic-like state and is associated with decreased contraction pressures and increased beating frequencies [66, 58] (see Figure 32a). Following Endothelin-1 treatment, the importance for each feature is determined using CA-LIME (see Table 5). The most important feature is the relaxation peak height, with an importance of -5.81, indicating that decreasing the relaxation peaks results in more confidence for predicting the sample as aberrant. The second most important feature is the downward relaxation time, with an importance of 5.01. An increase of the duration of the downward relaxation time is therefore associated with more confidence in predicting the sample as aberrant. An example of the three most important features and how they should be perturbed to make the sample more aberrant is indicated in Figure 31a.

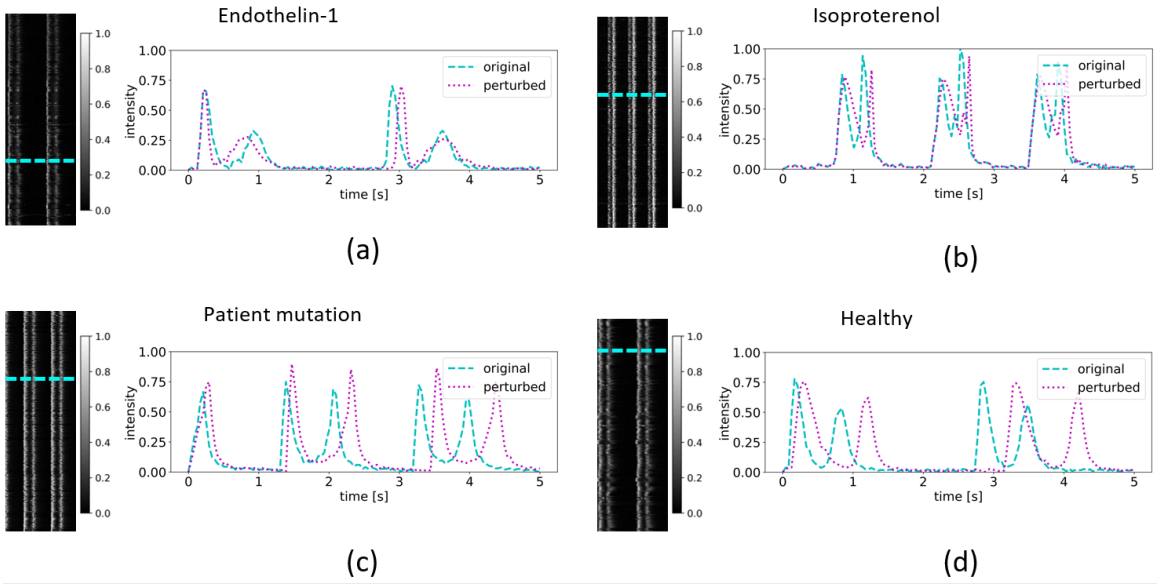

Figure 31: *Samples for which explanations are generated using CA-LIME together with example perturbations shown on a single track from the image. (a) left panel depicts a sample treated with Endothelin-1, the track along the dashed line is indicated in the right panel before and after applying the perturbations. (b) left panel depicts a sample treated with Isoproterenol, the track along the dashed line is indicated in the right panel before and after applying the perturbations. (c) Sample belonging to the patient mutation together with the track along the dashed line before and after applying the perturbation. (d) left panel depicts a healthy sample, the track along the dashed line is indicated in the right panel before and after applying the perturbations. All the plots show the three perturbations with the highest importance score as determined by CA-LIME. The perturbations are such that the confidence for the ground truth class of each sample is increased.*

	Importance ϕ_i			
CA-LIME feature	Endothelin-1	Isoproterenol	patient mutation	healthy
Contraction peak height	1.33	0.25	3.80	-0.12
Relaxation peak height	-5.81	4.11	11.40	3.25
Upward contraction time	-0.96	1.00	0.59	3.29
Downward contraction time	-2.72	1.55	-1.57	6.24
Upward relaxation time	0.64	0.90	3.57	0.44
Downward relaxation time	5.01	-1.78	-0.2	-1.20

Table 5: *Feature importance determined by CA-LIME for the different aberrant conditions and a healthy sample. Positive importance values indicate that the feature should be increased to elevate the prediction for the ground truth class of the sample. Similarly, negative features should be decreased to elevate the confidence of the network. The three most important features for each condition are highlighted in blue if they have a negative importance value and in yellow for positive importance values.*

Treatment with Isoproterenol causes increased beating frequencies and relaxation pressures to be observed [58] (see Figure 32b). According to CA-LIME, the most important feature in the sample treated with Isoproterenol, is the height of the relaxation peak with a value of 4.11 (Figure 31b). An increase of the relaxation peak causes more confidence for predicting the sample as aberrant. The second most important feature is again the duration of the downward relaxation, which should now be shortened in order to increase the confidence for predicting the sample as aberrant. Figure 31b shows one of the tracks and how it should be perturbed, in order to more confidently predict it as aberrant.

Like Endothelin-1, the patient mutation is associated with hypertrophy. CA-LIME predicts the most important feature to be the relaxation peak height with a value of 11.40, followed by the contraction peak height with a value of 3.80. Both peak heights should be increased in order to increase confidence for the

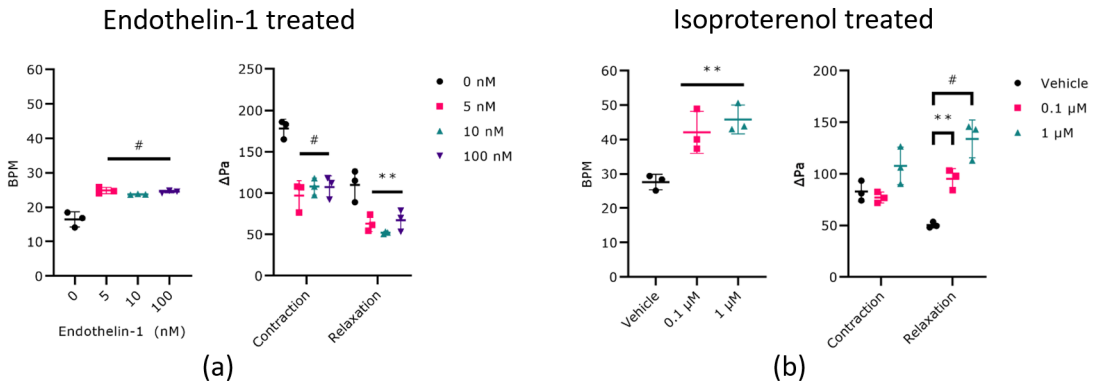

Figure 32: *Results after treating beating cardiomyocytes with Endothelin-1 and Isoproterenol from Snelders and colleagues [58]. The vertical axis indicates the beats per minute (bpm) or the pressure. Results are determined using an optical flow based analysis. (a) Treatment with Endothelin-1 resulting in increased beating frequencies (left panel). Both the contraction and relaxation pressure are shown to decrease when treated with Endothelin-1 (right panel). (b) Treatment with Isoproterenol resulting in increased beating frequencies (left panel). Furthermore, the relaxation pressure was shown to decrease when treated with Isoproterenol (right panel).*

class aberrant. The sample and how the top 3 features should be perturbed is given in Figure 31c.

Finally, CA-LIME is used to explain one of the healthy samples (see Figure 31d). The most important feature is the duration of the downward contraction, with a weight of 6.24, followed by the upward contraction time with 3.29. The duration of both time periods should be increased for an elevated confidence in the class healthy. The three most important perturbations are indicated in Figure 31d.

6 Discussion

Previously published methods use calcium signaling and machine learning methods to classify cardiomyocytes as healthy or diseased [67, 68, 69]. The fluorescent dyes used for this imaging procedure can however, result in low temporal resolution and interfere with the motion of the beating cardiomyocytes because of their toxicity [13, 11, 70]. Therefore, a label-free method which enables long term studies is preferred, like presented in the work of Teles and colleagues [11]. None of these works however discuss the importance of the spatial and temporal dimension or deploy current state-of-the-art deep learning methods to perform the classification. Our present study shows the use of Elastix as label-free method to extract the motion of beating cardiomyocytes and a deep learning based classifier. With 97.5% accuracy, we outperform all the previously mentioned machine learning methods [67, 68, 69, 11]. Moreover, unlike these methods, we generate explanations that indicate the difference between healthy and diseased tracks to provide novel insights into the used disease models. For this purpose, CA-LIME was developed, which adapts a superpixel based approach to generate explanations for periodic data. The generated explanations are easy to interpret and resemble the explanations of tabular data, in which a short list of understandable features, their importance and how they should be perturbed are indicated.

According to CA-LIME, the relaxation peak height belongs to the top 3 most important features for all the explained samples (Table 5). Even more, it was the most important feature for all the samples belonging to the class aberrant. The observations for the Endothelin-1 and Isoproterenol treated sample are consistent with previous results established by the group [58]. This increases confidence in the predictions of our 2D-CCNET and provides evidence for the effectiveness of CA-LIME. To the best of our knowledge, the duration of the upward and downward relaxation times as a result of treatment has not been reported previously.

A random forest classifier used to classify motion as healthy or diseased reported by Teles and colleagues [11] achieved an accuracy of 92% and F1 score of 91%. Eventhough we utilize a similar processing pipeline and extract more features including pressure characteristics about the motion, the performance is not matched in our work (Figure 22). Our random forest classifier achieved 85% accuracy and 87% F1 score, using nine decision trees. The difference in performance could be explained by the size of the training set, consisting of 322 healthy videos and 148 videos of a patient mutation compared to 33 healthy and aberrant videos in our work. The reduced training size could result in overfitting and decreased performance figures. The reduced accuracy and F1 score with a random forest over 19 trees or more observed in Figure 22 indicates that these random forests consisting of many trees start to overfit. Alternatively, the difference could also be explained by the different conditions of the diseased group. Our work contained an aberrant condition in which multiple different disease models, drug induced or patient mutations, are included. The work of Teles and colleagues [11] acquired 148 diseased samples from the same patient suffering from Timothy Syndrome, which might result in a more distinct difference between the healthy and aberrant (diseased) population.

The determined feature importance in Table 1 only indicates the importance over the entire dataset. Smaller populations within the data, like those treated with Isoproterenol could therefore have different features with the highest importance score, like the relaxation pressure for example. Using all the available data, these features might not be included in the top 5 most important features, as the samples treated with Isoproterenol make up less than 15% of the entire dataset. The most important features relate to the time component, which could be expected as the temporal component relates to the force and acceleration of the beating cardiomyocytes and is known to be different for healthy and aberrant cardiomyocytes [58, 71, 72].

A deep learning approach able to use both spatial and temporal information resulted in accuracy of 78% using the 3D-CCNET, a 7% decrease compared to the decision tree classifier. The reason for this can be attributed to overfitting on the spatial information. The architecture was difficult to optimize and often resulted in a large performance gap between the training and test set. Removing two input channels, using only the interframe displacement and angle from previous reduced this issue, boosting the accuracy to 88%, slightly outperforming our random forest classifier. Having less input channels available reduced overfitting and therefore resulted in better performance of the test set. Removing the spatial dimension entirely by permuting the spatial information as data augmentation reduced overfitting further resulting in a 95% accuracy outperforming the random forest of Teles and colleagues [11] using much less training data. A similar observation was made when the spatial information was removed by vectorizing the spatial dimension and using our 2D-CCNET. From these observations, we conclude that the spatial information does not provide additional information to differentiate healthy from aberrant tracks. The addition of disease models in which cell signaling for example is effected, could result in altered spatial information and should be further investigated.

As discussed above, the spatial information does not contribute to the classification task. Despite this, Grad-CAM [19] indicates specific spatial regions that contribute most to the decision, mainly in the interframe

displacement channel. Based on experiments with images in which the spatial dimension is vectorized, we conclude that Grad-CAM does not point to spatial information here, but to the most informative tracks. Since Grad-CAM shows that tracks with large interframe displacements carry the most importance, selecting only these tracks should be sufficient for an accurate classification. This is also observed in the experiments with images in which the spatial dimension is vectorized and only 750 tracks are used, while still reaching 92% accuracy (Figure 27b).
The high accuracy of the 2D-CCNET suggests there is a clear difference between the healthy and aberrant population. Therefore, efforts were made to explore an unsupervised classification method that clusters the 350 tracks with the highest interframe displacement from each sample ($T_N = 350$). Using t-SNE [73] or UMap [74] did not yield seperated clusters between the healthy or aberrant population. Future work with more advanced or partially supervised clustering methods should be investigated to explore new differences or sub-populations within the dataset [75]. Methods like convolutional embedding networks [76] or ClusterNET [77] are good candidates and should be further investigated.

Both the explanations by Grad-CAM and SHAP [20] indicate that the interframe displacement is the most important feature for the classification task. This observation is further confirmed by the experiment in which the 2D-CCNET is compared to the CaTNET. During training, solely the interframe displacement is used, still resulting in 88.2% accuracy for the CaTNEt and 97.5% for the 2D-CCNET. Furthermore, the interframe displacement also depicts clearly separable contractions and relaxation peaks, which aid in the connection to literature and for generating a comprehensible explanation for different parts of contraction-relaxation cycle using CA-LIME. The low mean L_2 and mean L_1 distance values between the heatmaps and the distance from median (Table 2), can be explained by the similarity between the interframe displacement and distance from median channel. Both channels depict a similar intensity pattern for most samples, as the median position is often located near the resting position of the cycle.

Explanations generated by SHAP indicate a large increase in the maximum absolute SHAP value for specific regions during a single contraction-relaxation cycle, but in subsequent beats, this rise is not observed (Figure 25, 26). This observation could indicate overfitting, in which only very specific parts of the input data are used which are not necessarily different between the two classes. This could also be the effect of using a superpixel based approach, in which specific parts of each cycle are masked ineffectively, still displaying similar information despite masking out the superpixel.

The levelset representation is more suited for describing the shapes of the tracks as it enables the description of non-convex shapes. The star polygon representation, however, allows all amplitudes to be normalized to study the effect of shape alone. Having shape and the global orientation of each sample available, 68% accuracy is obtained (Figure 28a). Removing the global orientation by rotating each shape, did not alter the description of the shape but did reduce the accuracy to 56%. We conclude from this that the global orientation of the samples caused a bias, which allowed a relatively high classification accuracy of 68% to be reached. This effect is, however, not attributed due to a difference in the shape of the track between healthy or aberrant cardiomyocytes. The observation that there is no difference in the distribution of the ratio minor/major axis confirms this. As the levelset representation has both the global orientation and amplitude of motion available, the performance increase of 5% relative to the star polygon model could be attributed to the beating amplitude. Alternatively, this could also be caused by including an accurate description of non-convex shapes using the levelset representation. Both the star polygon and levelset representation however confirm that without the temporal dimension, the accuracy and F1 score decrease more than 20%. Indicating the importance of the temporal dimension. This conclusion is further confirmed by the top 5 most important features determined by the random forest classifier and the feature importance scores determined by CA-LIME.

Using the average intensity to extract the motion of beating cardiomyocytes, instead of image registration, results in a 10% decrease of performance (Figure 30, 27b). Unlike registration methods like Elastix [60] that determine the deformation field, the amplitude of the contraction is not accurately determined using the change in average intensity. This lack of the amplitude component could explain the difference in the observed performance compared to the pipeline which included image registration.

6.1 Conclusion

In summary, we present a novel deep learning approach that could contribute to a fast drug screening system for heart failure. The deep learning algorithm classifies the motion of beating cardiomyocytes as healthy or aberrant. The aberrant phenotype is modelled using a patient mutation or by the treatment with Endothelin-

1 or Isoproterenol. The motion is extracted using image registration, extracting both spatial and temporal information of the contraction-relaxation cycle. The registration data is used to calculate four different features which are used as input channels for the deep learning classifier.

Experiments representing the shape of the tracks as star-polygon or levelset, showed the temporal information is most important for the classification task. The final model, utilizing only the temporal information, achieved 97.5% accuracy, outperforming previously published machine learning algorithms. Using Grad-CAM and SHAP to generate explanations for the deep learning classifier, the interframe displacement was shown to be the most important channel. Besides the classifier, we introduced CA-LIME, a novel AI interpretability method specifically tailored to explain the predictions of cardiomyocytes contractility profiles. The explanations by CA-LIME are in correspondence with previous observations of the effects of Endothelin-1 and Isoproterenol. The explanations by CA-LIME represent those of tabular data and are easy to interpret. CA-LIME could contribute to the detection of novel differences between the motion of healthy and aberrant beating cardiomyocytes.

6.2 Future prospects

CA-LIME, similar like LIME [17], is sensitive to picking favorable samples in which the generated explanations show the desired result. Moreover, the weighting function used to determine the weights of the samples can influence the explanation. To alleviate this issue, CA-LIME should be extended to predicted shapley values, similar like SHAP [20]. SHAP however, utilizes a superpixel based approach not effective for masking features in our input data. Moreover, SHAP uses a binary feature vector, meaning explanations only indicate the effect of masking parts of the track. On the contrary, CA-LIME indicates if peaks heights or durations should be decreased or increased to put more confidence in the decision. Using a binary feature vector is not possible in that case. Since currently only six features are implemented, without too much computational load, shapley regression values [78], or shapley sampling values [79] might be calculated by computing the effect of all possible combinations of the perturbations. As a result, the importance values will fulfill the properties of shapley values. Additional recordings of the disease conditions would enable the aberrant class to be separated into different treatment conditions, allowing the detection of different compounds or disease models like in the work of Lee and colleagues [80]. This could also enable CA-LIME to make more accurate explanations as the aberrant condition is currently a mixture of different effects. Additional data could also contain calcium signaling, which could be added as input channel besides the interframe displacement.

References

[1] Anke J Tijsen, Yigal M Pinto, and Esther E Creemers. Non-cardiomyocyte micrornas in heart failure. *Cardiovascular research*, 93(4):573–582, 2012.

[2] Sharon Ann Hunt, William T Abraham, Marshall H Chin, Arthur M Feldman, Gary S Francis, Theodore G Ganiats, Mariell Jessup, Marvin A Konstam, Donna M Mancini, Keith Michl, et al. 2009 focused update incorporated into the acc/aha 2005 guidelines for the diagnosis and management of heart failure in adults: a report of the american college of cardiology foundation/american heart association task force on practice guidelines developed in collaboration with the international society for heart and lung transplantation. *Journal of the American College of Cardiology*, 53(15):e1–e90, 2009.

[3] Boback Ziaeian and Gregg C Fonarow. Epidemiology and aetiology of heart failure. *Nature Reviews Cardiology*, 13(6):368–378, 2016.

[4] Spencer L James, Degu Abate, Kalkidan Hassen Abate, Solomon M Abay, Cristiana Abbafati, Nooshin Abbasi, Hedayat Abbastabar, Foad Abd-Allah, Jemal Abdela, Ahmed Abdelalim, et al. Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 392(10159):1789–1858, 2018.

[5] Véronique L Roger. Epidemiology of heart failure. *Circulation research*, 113(6):646–659, 2013.

[6] Mia N Christiansen, Lars Køber, Peter Weeke, Ramachandran S Vasan, Jørgen L Jeppesen, J Gustav Smith, Gunnar H Gislason, Christian Torp-Pedersen, and Charlotte Andersson. Age-specific trends in incidence, mortality, and comorbidities of heart failure in denmark, 1995 to 2012. *Circulation*, 135(13):1214–1223, 2017.

[7] Scott D Solomon, Joanna Dobson, Stuart Pocock, Hicham Skali, John JV McMurray, Christopher B Granger, Salim Yusuf, Karl Swedberg, James B Young, Eric L Michelson, et al. Influence of nonfatal hospitalization for heart failure on subsequent mortality in patients with chronic heart failure. *Circulation*, 116(13):1482–1487, 2007.

[8] JE Sanderson and TF Tse. Heart failure: a global disease requiring a global response, 2003.

[9] Houman Savoji, Mohammad Hossein Mohammadi, Naimeh Rafatian, Masood Khaksar Toroghi, Erika Yan Wang, Yimu Zhao, Anastasia Korolj, Samad Ahadian, and Milica Radisic. Cardiovascular disease models: a game changing paradigm in drug discovery and screening. *Biomaterials*, 198:3–26, 2019.

[10] Jennifer M Beierlein, Laura M McNamee, Michael J Walsh, Kenneth I Kaitin, Joseph A DiMasi, and Fred D Ledley. Landscape of innovation for cardiovascular pharmaceuticals: from basic science to new molecular entities. *Clinical therapeutics*, 39(7):1409–1425, 2017.

[11] Diogo Teles, Youngbin Kim, Kacey Ronaldson-Bouchard, and Gordana Vunjak-Novakovic. Machine learning techniques to classify healthy and diseased cardiomyocytes by contractility profile. *ACS Biomaterials Science & Engineering*, 7(7):3043–3052, 2021.

[12] Arun Sharma, Paul W Burridge, Wesley L McKeithan, Ricardo Serrano, Praveen Shukla, Nazish Sayed, Jared M Churko, Tomoya Kitani, Haodi Wu, Alexandra Holmström, et al. High-throughput screening of tyrosine kinase inhibitor cardiotoxicity with human induced pluripotent stem cells. *Science translational medicine*, 9(377):eaaf2584, 2017.

[13] Eeva Laurila, Antti Ahola, Jari Hyttinen, and Katriina Aalto-Setälä. Methods for in vitro functional analysis of ipsc derived cardiomyocytes—special focus on analyzing the mechanical beating behavior. *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research*, 1863(7):1864–1872, 2016.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[15] Md Zahangir Alom, Tarek M Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C Van Essen, Abdul AS Awwal, and Vijayan K Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3):292, 2019.

[16] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv e-prints*, pages arXiv–1605, 2016.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[18] Alun Preece. Asking 'why'in ai: Explainability of intelligent systems–perspectives and challenges. *Intelligent Systems in Accounting, Finance and Management*, 25(2):63–72, 2018.

[19] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[20] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[21] Tsehay Admassu Assegie, Thulasi Karpagam, Radha Mothukuri, Ravulapalli Lakshmi Tulasi, and Minychil Fentahun Engidaye. Extraction of human understandable insight from machine learning model for diabetes prediction. *Bulletin of Electrical Engineering and Informatics*, 11(2):1126–1133, 2022.

[22] Shota Yanagida, Ayano Satsuka, Sayo Hayashi, Atsushi Ono, and Yasunari Kanda. Comprehensive cardiotoxicity assessment of covid-19 treatments using human-induced pluripotent stem cell-derived cardiomyocytes. *Toxicological Sciences*, 183(1):227–239, 2021.

[23] Amy Pointon, Alexander R Harmer, Ian L Dale, Najah Abi-Gerges, Joanne Bowes, Christopher Pollard, and Helen Garside. Assessment of cardiomyocyte contraction in human-induced pluripotent stem cell-derived cardiomyocytes. *Toxicological Sciences*, 144(2):227–237, 2015.

[24] Maggie Zi Chow, Kenneth R Boheler, and Ronald A Li. Human pluripotent stem cell-derived cardiomyocytes for heart regeneration, drug discovery and disease modeling: from the genetic, epigenetic, and tissue modeling perspectives. *Stem cell research & therapy*, 4(4):1–13, 2013.

[25] Jianhua Zhang, Gisela F Wilson, Andrew G Soerens, Chad H Koonce, Junying Yu, Sean P Palecek, James A Thomson, and Timothy J Kamp. Functional cardiomyocytes derived from human induced pluripotent stem cells. *Circulation research*, 104(4):e30–e41, 2009.

[26] Limor Zwi, Oren Caspi, Gil Arbel, Irit Huber, Amira Gepstein, In-Hyun Park, and Lior Gepstein. Cardiomyocyte differentiation of human induced pluripotent stem cells. *Circulation*, 120(15):1513–1523, 2009.

[27] Anders Krogh. What are artificial neural networks? *Nature biotechnology*, 26(2):195–197, 2008.

[28] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[29] Khalid M Hosny, Mohamed A Kassem, and Mohamed M Foaud. Skin melanoma classification using deep convolutional neural networks. In *Deep Learning in Computer Vision*, pages 291–314. CRC Press, 2020.

[30] Okeke Stephen, Mangal Sain, Uchenna Joseph Maduh, and Do-Un Jeong. An efficient deep learning approach to pneumonia classification in healthcare. *Journal of healthcare engineering*, 2019, 2019.

[31] Kunihiko Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, 20(3):121–136, 1975.

[32] Alexander N Gorban and Donald C Wunsch. The general approximation theorem. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 2, pages 1271–1274. IEEE, 1998.

[33] Boris Hanin and Mark Sellke. Approximating continuous functions by relu nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

[34] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

[35] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

[36] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[37] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[38] Douglas M Kline and Victor L Berardi. Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications*, 14(4):310–318, 2005.

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[40] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[41] A.V. Oppenheim and A.S. Willsky. *Signals and Systems: Pearson New International Edition*. Always learning. Pearson Education Limited, 2013.

[42] Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*, 2016.

[43] Rajendran Nirthika, Siyamalan Manivannan, Amirthalingam Ramanan, and Ruixuan Wang. Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. *Neural Computing and Applications*, pages 1–27, 2022.

[44] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[45] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1):1–74, 2021.

[46] Sucheng Ren, Qiang Wen, Nanxuan Zhao, Guoqiang Han, and Shengfeng He. Unifying global-local representations in salient object detection with transformer. *arXiv preprint arXiv:2108.02759*, 2021.

[47] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[48] Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee, and Kincho H Law. Automatic localization of casting defects with convolutional neural networks. In *2017 IEEE international conference on big data (big data)*, pages 1726–1735. IEEE, 2017.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[51] John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2, 1989.

[52] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. *arXiv preprint arXiv:2201.09873*, 2022.

[53] Christos Matsoukas, Johan Fredin Haslum, Magnus Söderberg, and Kevin Smith. Is it time to replace cnns with transformers for medical images? *arXiv preprint arXiv:2108.09038*, 2021.

[54] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[55] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: a review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[56] LS Shapley. Quota solutions op n-person games1. *Edited by Emil Artin and Marston Morse*, page 343, 1953.

[57] Jesse He and Subhasish Mazumdar. Comparing lime and shap using synthetic polygonal data clusters. *International Journal for Infonomics (IJI)*, 2021.

[58] Matthijs Snelders, Iris H. Koedijk, Julia Schirmer, Otto Mulleners, Juancito van Leeuwen, Nathalie P. de Wagenaar, Oscar Bartulos, Pieter Voskamp, Stefan Braam, Zeno Guttenberg, A.H. Jan Danser, Danielle Majoor-Krakauer, Erik Meijering, Ingrid van der Pluijm, and Jeroen Essers. Contraction pressure analysis using optical imaging in normal and mybpc3-mutated hipsc-derived cardiomyocytes grown on matrices with tunable stiffness. *To be published*, page 1, 2022.

[59] L. Breiman, Jerome H. Friedman, Richard A. Olshen, and C. J. Stone. Classification and regression trees. In *Wiley, International Biometric Society*, 1983.

[60] Stefan Klein, Marius Staring, Keelin Murphy, Max A Viergever, and Josien PW Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2009.

[61] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018.

[62] Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers. Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3666–3673, 2020.

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

[64] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[65] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[66] Henriëtte W de Jonge, Dick HW Dekkers, Adriaan B Houtsmuller, Hari S Sharma, and Jos MJ Lamers. Differential signaling and hypertrophic responses in cyclically stretched vs endothelin-1 stimulated neonatal rat cardiomyocytes. *Cell biochemistry and biophysics*, 47(1):21–32, 2007.

[67] Martti Juhola, Henry Joutsijoki, Kirsi Penttinen, and Katriina Aalto-Setälä. Machine learning to differentiate diseased cardiomyocytes from healthy control cells. *Informatics in Medicine Unlocked*, 14:15–22, 2019.

[68] Henry Joutsijoki, Kirsi Penttinen, Martti Juhola, and Katriina Aalto-Setälä. Separation of hcm and lqt cardiac diseases with machine learning of ca2+ transient profiles. *Methods of Information in Medicine*, 58(04/05):167–178, 2019.

[69] Hyun Hwang, Rui Liu, Joshua T Maxwell, Jingjing Yang, and Chunhui Xu. Machine learning identifies abnormal ca 2+ transients in human induced pluripotent stem cell-derived cardiomyocytes. *Scientific reports*, 10(1):1–10, 2020.

[70] Matthew F Peters, Sarah D Lamore, Liang Guo, Clay W Scott, and Kyle L Kolaja. Human stem cell-derived cardiomyocytes in cellular impedance assays: bringing cardiotoxicity screening to the front line. *Cardiovascular Toxicology*, 15(2):127–139, 2015.

[71] Marita L Rodriguez, Brandon T Graham, Lil M Pabon, Sangyoon J Han, Charles E Murry, and Nathan J Sniadecki. Measuring the contractile forces of human induced pluripotent stem cell-derived cardiomyocytes with arrays of microposts. *Journal of biomechanical engineering*, 136(5):051005, 2014.

[72] Daisuke Sasaki, Katsuhisa Matsuura, Hiroyoshi Seta, Yuji Haraguchi, Teruo Okano, and Tatsuya Shimizu. Contractile force measurement of human induced pluripotent stem cell-derived cardiac cell sheet-tissue. *PloS one*, 13(5):e0198026, 2018.

[73] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[74] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[75] Md Rezaul Karim, Oya Beyan, Achille Zappa, Ivan G Costa, Dietrich Rebholz-Schuhmann, Michael Cochez, and Stefan Decker. Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, 22(1):393–415, 2021.

[76] Md Rezaul Karim, Michael Cochez, Achille Zappa, Ratnesh Sahay, Dietrich Rebholz-Schuhmann, Oya Beyan, and Stefan Decker. Convolutional embedded networks for population scale clustering and bioancestry inferencing. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.

[77] Ankita Shukla, Gullal S Cheema, and Saket Anand. Semi-supervised clustering with neural networks. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pages 152–161. IEEE, 2020.

[78] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

[79] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.

[80] Eugene K Lee, David D Tran, Wendy Keung, Patrick Chan, Gabriel Wong, Camie W Chan, Kevin D Costa, Ronald A Li, and Michelle Khine. Machine learning of human pluripotent stem cell-derived engineered cardiac tissue contractility for automated drug classification. *Stem cell reports*, 9(5):1560–1572, 2017.

7 Acknowledgement

I would like to thank Ihor Smal and Matthijs Snelders for their help during the project. They provided me with feedback and helped me analyze the data. I would further like to thank all members of the group for their fruitful discussions that resulted in new ideas. I would also like to thank Jeroen Essers who gave me the opportunity to work on this project. Lastly, I would like to thank Mischa Hoogeman for his help and feedback.

8 Clarification between degrees

This report is for my graduation of the master Nanobiology and the master Biomedical Engineering (track Medical Physics). The project represents my graduation work for both masters. Although the subject covers aspects of both studies, the developed interpretability methods were developed for Nanobiology, as they provide more insight into the disease models. The proposed classifier and experiments for the spatial and temporal dimension are included for Medical Physics. For in this master program, image processing for clinically relevant applications is considered and developed.

9 Supplementary figures

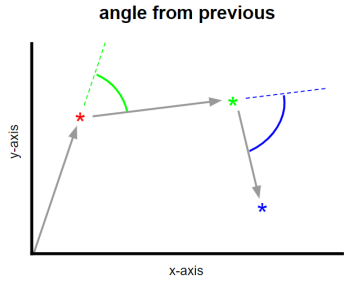

Figure S1: *Angle from previous indicated for a small example track. the arrows indicate the track, the green and blue line indicate how the angle from previous is defined.*

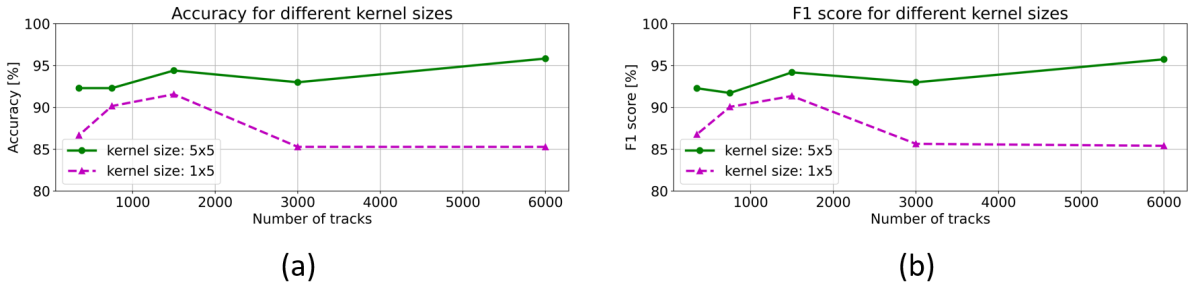

Figure S2: *Accuracy and F1 score for different kernel sizes. Input images in which the spatial dimension is vectorized are used, classification is performed using the 2D-CCNET. Experiments are performed for different number of tracks. (a) Accuracy using a 5x5 and a 1x5 kernel. The 1x5 kernel might be more suited for the input data, as the different rows in the input image represent spatially unrelated tracks. The 5x5 kernel is however observed to achieve better accuracy scores. (b) F1 score using a 5x5 and 1x5 kernel. The 5x5 kernel results in superior F1 scores independent of the number of tracks used.*