Universitatea
Transilvania
din Brașov

ȘCOALA DOCTORALĂ INTERDISCIPLINARĂ

Facultatea: Inginerie Electrică și Știința Calculatoarelor

Ing. Costin Florian CIUȘDEL

Învățare profundă pentru analiza imaginilor de diagnosticare în bolile cardiovasculare

Deep Learning for diagnostic image analysis in cardiovascular disease

REZUMAT / ABSTRACT

Conducător științific

Prof.dr.ing. Lucian Mihai ITU

BRAȘOV, 2022

Universitatea
Transilvania
din Brașov

D-lui (D-nei) ....

Componența
Comisiei de doctorat

Numită prin ordinul Rectorului Universității Transilvania din Brașov

Nr. .... din ....

PREȘEDINTE:	- Conf. univ. dr. ing. UNGUREANU Delia Prodecan Universitatea Transilvania din Brașov
CONDUCĂTOR ȘTIINȚIFIC:	- Prof. univ. dr. ing. ITU Lucian Mihai Universitatea Transilvania din Brașov
REFERENȚI:	- Prof. univ. dr. ing. POPESCU Dan Universitatea Politehnica din București - Prof. univ. dr. ing. SALOMIE Ioan Universitatea Tehnică din Cluj Napoca - Prof. univ. dr. ing. MOLDOVEANU Florin Dumitru Universitatea Transilvania din Brașov

Data, ora şi locul susținerii publice a tezei de doctorat: ...., ora ....., sala ....

Eventualele aprecieri sau observații asupra conținutului lucrării vor fi transmise electronic, în timp util, pe adresa costin.ciusdel@unitbv.ro

Totodată, vă invităm să luați parte la şedința publică de susținere a tezei de doctorat.

Vă mulțumim.

Contents

1. Introduction

1.1 Deep Learning for Large Medical Datasets: Overview, Challenges and Future

A formal definition of Deep Learning (DL) would describe it as branch of Machine Learning (ML), containing techniques for training multi-layered neural networks for various prediction tasks, in which relevant features are automatically inferred from the training data, in contrast to classical ML methods which often require manual feature engineering.

An informal description of DL would picture it as a momentum-gaining phenomenon on multiple aspects. In the **engineering** of:

- **hardware**: more and more hardware segments are developed for accelerating Deep Neural Networks inference: CPUs with vectorized instructions for high-throughput floating point operations; GPUs augmented with dedicated Tensor processing cores; Tensor processing units (TPUs) deployed in cloud infrastructures; low power embedded hardware capable of running general purpose DNN models.

- **software**: several Deep Learning frameworks and ecosystems have been developed to support model training and deployment. New software architectures leverage powerful remote computing engines through the Infrastructure-as-a-Service paradigm.

Inside the **business logic** of applications used in content creation, medical imaging, financial transactions and many more, DNN models have been integrated with wide-spread success. Commercial applications have state-of-the-art features and capabilities which are powered by DL solutions. In **academia and research**, the topic of DL is very hot and constantly attracts new attention. Latest breakthroughs in fundamental fields such as computer vision and natural language processing have propagated quickly to user-available products such as real-time translation services, self driving cars, etc.

The deployment of artificial-intelligence (AI) based algorithms have produced a boost in human productivity similar to the effect of the wide-spread software-solutions adoption from several decades ago. Many research groups are striving to reach the prospects of artificial general intelligence, which can then be repurposed to solve novel and hard problems for the benefit of humanity.

Although neural networks are not a new concept (as their investigation started in the second half of the last century), only in the last decade have their adoption and growth started to attain the high rate that practitioners are experiencing today. The 2011-2012 period was a turning point for deep learning. [1] and [2] were two publications whose state-of-the-art performances sparked the community's interest. E.g., in the 2012 ImageNet Large Scale Visual Recognition Challenge, the "AlexNet" model introduced in [2] obtained a top-5 error of 15.3%, more than 10.8 percentage points lower than that of the runner up (a non-DL solution).

The advent of DL-powered solutions would not have been possible without 2 key ingredients: fast computation and large amounts of data. A DNN model typically has a parallelizable structure, allowing the efficient usage of multi-core processors. GPUs have historically had much larger single-precision floating point operations per second (FLOPs) throughput than regular desktop CPUs, therefore they were suitable candidates for highly parallelizable computations. However, GPUs were primarily designed for graphical tasks and accessing their processing prowess required reformulating the initial problem as one involving only graphics primitives. The usage of graphics-specific APIs for general purpose GPU (GPGPU) was complicated and restricting. In 2007, nVIDIA launched the first version of CUDA, a "Compute Unified Device Architecture" which exposed an API for general purpose programming of GPUs. Nowadays, CUDA reached its 11th version and it is heavily used as backend in all DL frameworks. The option of an user-friendly GPGPU API coupled with the exponential increase in FLOPs performance ensured the necessary computational requirement for DL advancement.

The other key ingredient, data, was also made readily available by other technological innovations: the world wide web and personal devices such as digital cameras, smartphones, etc. Users would create content and share it on the internet. Overtime, large pools of data were curated and used as reference datasets, an example being the ImageNet Challenge dataset.

Another aspect which favored the adoption of DL was its generalization capabilities. No longer relevant was the bottleneck of engineering or implementing numerical features suitable for each specific task and data pair, as a DL model would learn its own features directly from the data, given enough compute time and training data. In the AlexNet paper, the authors stated that "All of our experiments suggest that our results can be improved simply by waiting for faster GPUs and larger datasets to become available". The potential of DL was discovered to be very high, as simply leveraging more resources led to steady increases in performance. During the next years, larger models could be trained faster on more advanced GPUs on larger and larger training sets, with the resulting test set accuracy approaching or even surpassing human level performance on many prediction tasks.

Novel research directions were also pursued, such as generative modeling. Unsupervised learning offered ways to further improve discriminative modeling performance in the face of large datasets which had only a fraction of it associated with ground truth annotations. Even without any labels, DL models could still learn from raw data, inferring low- and high-level features about the underlying structure inside the data set.

Medical imaging is a domain in which DL models achieved remarkable performance. The existence of large data collections containing numerous patients and population subgroups fueled the research and development of deep neural models which performed many tasks towards aiding patient diagnosis, among which [3]:

- **classification**. E.g. of lung adenocarcinoma subtypes; of tumors on breast ultrasound images; of non-solid nodules; of skin cancer; of pulmonary abnormalities; of benign and malignant lesions; of evolutive lymphoma and residual masses; of cervical cancer; of EGFR mutation; of status for lung cancer; of brain tumor; etc.

- **localization**. E.g. of lung tumors; of organ landmarks; of breast lesions; etc.

- **segmentation**. E.g. of pulmonary nodules; of liver tumors; for atherosclerotic plaque; of uterine endometrial cancer; of tumor in retinoblastoma; of pancreatic cancer; etc.

- **image enhancement**. E.g. for denoising; for domain translation (from one imaging modality to another); for speeding-up image transforms (CT reconstruction); etc.

Developing a DL model for a medical imaging task usually requires obtaining a labeled dataset curated by expert readers and/or medical practitioners. The manual annotation process is usually slow and expensive. The strive is therefore to design methods which are very data-efficient, i.e. obtaining high performance models from moderate-size train sets. In deploy setups, models are typically included in pipelines with constraints on runtime. Therefore, parameter- and FLOP-efficient models are preferred. However, state-of-the-art results on standardized prediction tasks have showed that large models trained on large-scale datasets perform best. This indicates a compromise that often must be done between, on one side, the cost of annotating training data and of the hardware on the target platform and, on the other side, the expected prediction performance, robustness and generalization that is to be expected on-site.

Fortunately, there are methods to boost model performance. Pretraining is a powerful technique which can leverage high amounts of unlabeled data to yield data-efficient training procedures for the final target tasks. Model robustness can be linked with prediction uncertainty and out-of-distribution detection. Inputs that are unknown to the model (with respect to the data distribution observed during training) can be flagged and the prediction on such input data can be treated separately inside post-processing stages. In a diagnosing pipeline, imaging data that is too noisy or lacking required quality is better to be discarded than employed as evidence for setting a diagnostic.

Explainability is another important aspect in DL, especially in medical applications. Being able to argue why a model reached a certain prediction allows a user to build confidence in the model itself. Analyzing failure cases with explainable models also allows DL practitioners to make better informed decisions inside an iterative model development process. On the other side of the spectrum, treating DL models as black-boxes suggests being ignorant of their intrinsics and this approach hardly adds any value to a DL application in the face of possible model artifacts and failure modes.

A futuristic idea in modern medicine is the digital twin concept, in which personalized models are built for patients, which are continuously adjustable based on each patient's health history. As the data of a single patient is insufficient to build a standalone DL model, a meta-patient model can be instead conditioned on specific markers based on each patient's parameters to yield personalized predictions. With the current rate of development, it is not unrealistic to expect that the prediction accuracy of DL-based algorithms will reach or even surpass expert radiologists or sonographers on most tasks.

Medical imaging equipment will be continually upgraded with the latest DL-based algorithms for real-time acquisition analysis. Precision auto-diagnosis may be unlocked in the future, wherein the imaging scanner can self-adjust the scanning protocol based on what has been observed so far in a specific patient. In the current timeline, the medical acquisition is inspected by an expert reader often hours or days after the imaging procedure was completed and the patient has left the premises. If the findings warrant further deeper investigations, the patient must return and another imaging procedure must be conducted. An AI-powered automated imaging analysis solution could drastically reduce the time between first consult and final diagnosis.

1.2 Role of Cardiac Imaging

The structure of this thesis follows the clinical workflow for a patient suspected with cardiovascular disease. After an initial assessment, the physician may request cardiac imaging for a reliable diagnosis [4]. Individual patient characteristics and local accessibility influence the selection of the cardiac imaging modality. Echocardiography (including stress echo) is able to provide the required clinical information, for a large percentage of cases; it also has the advantage of avoiding radiation exposure. On the other hand, computerized tomography (CT) coronary angiography is increasingly employed to detect coronary artery disease in patients with ambiguous stress test results and in those with an intermediate risk [4].

Assessment of dyspnoea and investigation for coronary artery disease are two of the most common clinical scenarios that may require cardiac imaging [4]. For dyspnoea, two imaging modalities are often used:

- Chest X-ray: in the preliminary assessment of cardiovascular disease. It can observe an increase in heart size and indicators of pulmonary congestion. It cannot reliably exclude cardiac aetiology.

- Echocardiography: extremely useful for evaluating dyspnoea as it provides structural and functional information that indicates a particular diagnosis.

When imaging for dyspnoea is inconclusive and respiratory conditions have been excluded, coronary artery disease (CAD) is next considered. Accurate CAD diagnosis is crucial in treatment planning for patients presenting with chest pain. Among others, the following imaging investigations can be performed [4]:

- Stress ECG: low cost and safe test, but with low sensitivity and specificity for CAD.
- Stress echocardiography: effective and non-invasive. Provided information is often very useful. It is diagnostically reliable and generally accessible.
- CT coronary angiography: directly visualizes the coronary arteries for both obstructive and non-obstructive CAD. Useful for patients with ambiguous stress test results.
- CT coronary calcium scoring: may predict the risk of future cardiovascular events in individuals at intermediate risk of CAD.

For diagnosing cardiomyopathy, structural heart disease and congenital heart disease, transoesophageal echocardiography and cardiac MRI can be employed when there is not enough diagnostic information [4].

Therefore, cardiac imaging is a key ingredient in diagnosing and monitoring cardiovascular diseases. CT coronary angiography and cardiac MRI are both relatively recent in their clinical use compared to echocardiography [4]. Having such great importance in the clinical practice, further augmenting the imaging-based cardiac assessment procedures with DL-powered algorithms can only boost their impact on patient care, by automating measurements and increasing their precision.

1.3 Thesis Structure and Content

This thesis presents Deep Learning methods and results from applications in medical imagining. The thesis is structured in 6 chapters. This **first chapter** presented an overview on the impact and potential of DL algorithms for medical imaging.

The **second chapter** introduces ultrasound as an imaging modality. The typical DL tasks/steps comprised in the development cycle of an AI-powered solution are followed: first, pretraining methods are introduced. Such techniques are used to learn general representations from unlabeled data, to bootstrap the performance of downstream related supervised tasks. Another option is to generate synthetically more data through generative modeling. A framework for generating apical echocardiographies conditioned on heart-chamber segmentations masks is investigated. A typical measure of interest in assessing the cardiac function is the Ejection Fraction (EF). An auto-EF solution requires the automation of 2 key steps: detecting the cardiac phase in echo video acquisitions, specifically selecting the end-diastolic (ED) and end-systolic (ES) frames, and semantic segmentation of heart chambers for obtaining the corresponding contours to estimate the chamber volumes. To achieve the successful automation of these tasks, an RNN-based model is developed for cardiac phase detection while several model architectures are investigated and augmented for semantic segmentation and landmark detection. Finally, the 2nd chapter investigates the use of uncertainty estimation frameworks; they are adapted for running the segmentation models, therefore obtaining an uncertainty map attached to the predicted masks.

The **third chapter** casts landmark detection in 3D echocardiographies as a reinforcement learning problem. Six landmarks are to be detected for the LV chamber inside echo volumes at specific ED/ES time points. Instead of having a single large convolutional network operating on the entire input volume, a series of agents observing local regions take successive actions towards minimizing the distance between their position and the target landmark locations. This approach leads to a fast and resource-friendly method for 3D landmark detection. The training protocol is described along with the mathematical framework which governs it. Deployment aspects are considered to minimize runtime.

In the **fourth chapter**, unsupervised methods are explored for out-of-distribution detection. Normalizing flows models are developed to detect incorrect 3D lumen segmentations in coronary computed tomography angiographies. They are placed downstream of a pipeline stage of contouring, analyzing provided inputs either from the user or from a DNN segmentor, therefore acting as an Audit model. A novel NF architecture is compared against a baseline for their semantic analysis capabilities. Investigations reveal that by following certain constraints, an NF model capable of better semantic interpretation and hence better OoD detection capabilities is obtained. Sampling procedures also show that the baseline model focuses more on texture instead of semantics, as previously reported in literature, while the proposed model manages to produce realistic synthetic samples.

Chapter 5 presents a video classification problem, specifically detecting cardiac phase in invasive coronary angiographies. The proposed model is presented along with detailed results and analyses on considered test sets for assessing the entire solution pipeline. Similar to the cardiac phase detection on apical echocardiographies, the cardiac phase in invasive coronary angiographies is inferred purely based on the imaging frames, avoiding the need to process noisy ECG signals (which may also be absent for some acquisitions).

The **final chapter** presents the conclusions, highlights the original contributions and the dissemination of research results, and indicates possible directions of improvement as future work.

2. Deep Learning Solutions for 2D Cardiac Ultrasound Imaging

2.1 Introduction

Echocardiography is essential in cardiology, as it assists in the evaluation of heart valve function, such as stenosis or insufficiency, strength of cardiac muscle contraction, and hypertrophy or dilatation of the main chambers. Usually, during an echo study, multiple acquisitions are made from various views in order to capture sufficient information for determining quantities of interest such as: chamber dimensions and volumes, ejection fraction, wall thickening, global longitudinal strain, etc.

Ejection fraction (EF) is an important measurement in assessing the cardiac function [8]. It is usually measured only in the left ventricle, the heart's main pumping chamber. Reduced EF values may be caused by cardiomyopathy, heart muscle damage from a heart attack, heart valve problems or long-term high blood pressure. EF is defined as:

$$EF = \frac{EDV - ESV}{ESV} \tag{2.1}$$

where EDV and ESV are the LV volumes at two distinct time moments: end-diastole (ED) and end-systole (ES), respectively. To estimate the 2 volumes, the apical 2-chamber and 4-chamber views can be employed.

Given the LV segmentations for the two views, Simpson's method can be employed to obtain a bi-plane estimate of the ventricular volume, which should be closer to the actual physical value. This method considers partitioning the LV along its mitral-valve-to-apex axis into thin ellipses. The two ellipse axes are given the sizes of corresponding LV diameters in the A2C and A4C views at corresponding locations along the LV axis. A workflow for estimating EF would comprise the following steps:

1. acquire A2C and A4C echocardiographies (each one containing at least one full cardiac cycle)
2. detect and select the ED and ES frames inside the 2 views
3. segment the LV inside the 4 selected frames
4. apply Simpson's rule to estimate EDV and ESV and compute EF

Steps 2 and 3 can be automated using Deep Learning (DL) techniques, while step 4 can be implemented inside post-processing stages of an auto-EF solution. Other methodologies [8] train models to directly predict EDV, ESV and EF as regression tasks, however such approaches are lacking in terms of explainability.

2.2 Pretraining Methods[1]

2.2.1 Introduction

Developing a DNN model to perform predictions on a specific task requires a train set containing pairs of input samples and their corresponding target outputs. When training from scratch and the train set size is relatively small, the model performance on a hold-out test set may be sub-optimal. The training procedure may fail altogether, since overfitting a small train set is a well known phenomenon.

Pretraining is technique which has helped low-data scenarios to achieve higher test-time performance than training from scratch. This technique usually involves changing the initial weights (i.e. before starting the final training procedure) of the model under development to those taken from another model which was trained on another similar task. The donor model does not need to have an identical architecture as the final model, since it may have been trained on different types of prediction problems, e.g. the donor was trained on a large classification dataset while the final model is to be trained on a smaller segmentation dataset.

Obtaining labels for medical datasets usually involves expert readers manually annotating each individual case, which is a time consuming and expensive process. In contrast, there are large collections of medical acquisitions available for use, an opportunity which may be missed when considering only the small fractions of the collections which do have expert annotations. A method having high potential is therefore to learn the inherent data structure inside these large-scale collections, without using any labels. General features may be learned first on a surrogate task which holds no clinical importance. These general features are then finetuned specifically for a target task bearing clinical relevance.

The main motivation is that when training from scratch, the model must also discover these general features which describe semantically the input data. Having a pretrained model as an initial starting point makes the training procedure deal with a much easier problem, since the model already has a decent understanding of the typical semantic content. An easier problem needs less training data to achieve high detection performance. Therefore it is of high importance to find good pretraining methods which put downstream final tasks at a considerable advantage.

2.2.2 Methods and Results

Self-supervised learning is an array of methods aimed at training DNN models to produce meaningful learned representations. The usual recipe typically employs data augmentation and pretext tasks. These tasks formulate simple training objectives, which have little practical use by themselves, but it order for the model to solve them it needs to learn useful representations, which can later be repurposed when training for the final task.

This section deals with heuristic pretext tasks, in which a transform T_i is randomly chosen from a predefined set $\{T_1, ..., T_n\}$ and is applied to the input image. The model is trained on a classification task, trying to guess which transform was applied. In [9], two self-supervised pretraining methods were tested on apical BMode echocardiographies: random horizontal flips and frame ordering.

For the first method, individual echo frames were considered and flipped along the center vertical axis with a probability of 50%. A binary classification task was formulated in which the model should predict "false"/0 on original images and "true"/1 on flipped ones. In an initial training, the model obtained 95.8% accuracy on the test set (with 100% train time accuracy). It was previously reported in literature that when using such simple pretext task, the model can find means to cheat, i.e. instead of learning useful representations of the semantic content, the model may look for certain image artifacts which correlate well with the target labels.

[1] This section describes experiments done in [9], which represents previously published work of the author, under the PhD research program.

To test this, another experiment was conducted in which each training sample was injected with an artifact in the bottom right corner, namely a small white square always in the same position on the black echo background. Flipping an image would also modify the artifact location to bottom left. Therefore, there existed an easy cue which can be leveraged instead of trying to interpret the echo image content. Compared to the case of unperturbed data, the model quickly converges to 0 train and validation loss and to a maximum train and validation accuracy. When tested on the original test set (i.e. without the artifacts) the model performance decayed to random guessing: 51%, suggesting that nothing useful was learned by the model.

Saliency analysis was conducted in order to confirm that the model focuses on the artifacts. A flipped image was employed as test input. The artifact was applied with gradual intensity in the bottom left corner (the location for flipped images), starting from a pixel intensity of 0 (i.e. no artifact) up to full intensity (i.e. white color, as used when training). Initially, with no artifact applied, the model outputs a wrong label close to 0. When the artifact intensity is large enough, the model output has a sharp increase to value 1, the correct label for flipped inputs. One can observe that when the artifact is visible enough, the model focuses solely on it, disregarding any semantic content present inside the echocardiography.

These experiments highlight a shortcoming of such simple whole-image pretext task: the training data must be carefully pre-processed to exclude any possible artifacts which give away the applied transform during training and thus render useless such pretraining procedures.

Another pretraining method investigated in [9] used the pretext task of correctly ordering a sequence of systolic frames. Three frames were considered: beginning of systole (1), mid-systole (2) and end-systolic (3) frames, from the same cardiac cycle. There are six possible permutations in the ordering of the 3 frames: (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2) and (3,2,1). During training, an input tuple of frames was randomly applied a permutation and the target label was a one-hot encoding of the permutation index. Categorical cross-entropy was used as loss function.

Four different scenarios were considered, as combinations between:

- model initialization: either training from scratch or using the pretrained model from the flip detection task (without artifacts) as the initialization for the encoder. The final output stage always had random initialization.

- the encoder responsible for frame embeddings computation was either frozen or trainable. The final output stage was always trainable.

Tab. 2.1 displays the accuracy metrics on the test set. Whenever bootstrapping from the flip detection task, the accuracy increases. When freezing the encoder, the accuracy is slightly better with transfer learning (+5%) but massively lagging behind the scenarios when the model is fully trainable. This suggests that the first task managed to extract representations slightly more informative than random features, but not fully relevant for a much harder task such as frame ordering. This is intuitive since the relevant cues for flip detection (e.g. placement of heart chambers) might not be as relevant as the cues for correct frame ordering (e.g. mitral valve opening and LV size).

Therefore, when using heuristic pretext tasks, special care must also be taken to pair the final target task with a pretext one that depends, at least in part, on the same image regions and structures.

2.2.3 Conclusions

For medical imaging modalities such as ultrasound echocardiographies, special care must be taken when choosing the data augmentation operations. Unlike natural images, a regular BMode echo acquisition has a constant shape (i.e. the US cone in the middle of the image) and the color content is usually not relevant. Instead of color jittering, contrast and brightness perturbations should be employed.

Table 2.1: Frame ordering accuracies in different initialization and training scenarios.

Initialising way	Accuracy
Random initialization	92.38%
Initialization with the parameters learned from the binary classification task (transfer learning)	95.43%
Random initialization and frozen layers	34.51%
Initialization from the flip detection task (transfer learning) and frozen layers	39.59%

When random image cropping is employed, it should be constrained to capture relevant heart structures and sufficient overlap between patches should be ensured. Otherwise, the model can be forced to produce similar encodings from disjointed regions of an echo, e.g. constraining that the representation of a patch around the RA chamber should be similar to one around the LV may not produce useful learned representations. Instead, inclusive patches can be used to enforce that the model can localize local heart chamber structures inside larger whole-chamber views and overlapped adjacent patches can be employed to ensure that the model can detect if two adjacent patches are capturing the same heart chamber from the same acquisition.

2.3 Conditional Generative Modeling of Echocardiographies

Generative modeling involves learning the distribution of the training data, either explicitly (e.g. as in a Variational Autoencoder, a Normalizing Flow model, etc.) or implicitly (as in Generative Adversarial Networks - GANs). In conditional generative modeling, a new sample is being drawn from the learned distribution based on some external conditioning signals. Such modeling techniques have great utility since they allow sampling from under-represented regions of the original sample space and can aid the development of supervised models by pretraining them on synthetically generated data.

This section deals with the conditional generative modeling of apical BMode echocardiographies, specifically generating photo-realistic echo frames from input chamber-segmentation masks. This goal is the opposite of a regular segmentation task:

- in segmentation, the model outputs a mask correlated with the spatial content of an input echocardiographic frame; the semantic content is inferred from the input and spatial details are used to position and tune the predicted mask.
- in generation, the model outputs an echocardiographic frame which is plausible under the provided conditioning masks, i.e. the placement, size and shape of the cardiac chambers match with the provided segmentations.

This section follows the conditional generative framework introduced in [15]. A GAN is developed in which the generator employs a novel type of normalization layer called "Spatially Adaptive (De)Normalization" (SPADE). The conditioning masks are processed by convolutional layers to compute spatial demodulation tensors:

$$h^{i+1}_{n,c,y,x} = \gamma^{i}_{c,y,x}(m)\frac{h^{i}_{n,c,y,x} - \mu^{i}_{c}}{\sigma^{i}_{c}} + \beta^{i}_{c,y,x}(m) \tag{2.2}$$

where $h^{i}_{n,c,y,x}$ and $h^{i+1}_{n,c,y,x}$ are feature maps before (at layer i) and after (at layer $i+1$) the SPADE normalization, respectively. Indexes n, c, y and x are along batch, channel, height and width axes, respectively. μ^{i}_{c} and σ^{i}_{c} are per-channel normalization coefficients similar to the functioning of a regular BatchNorm layer without the affine transformation, i.e. mean and standard deviation.

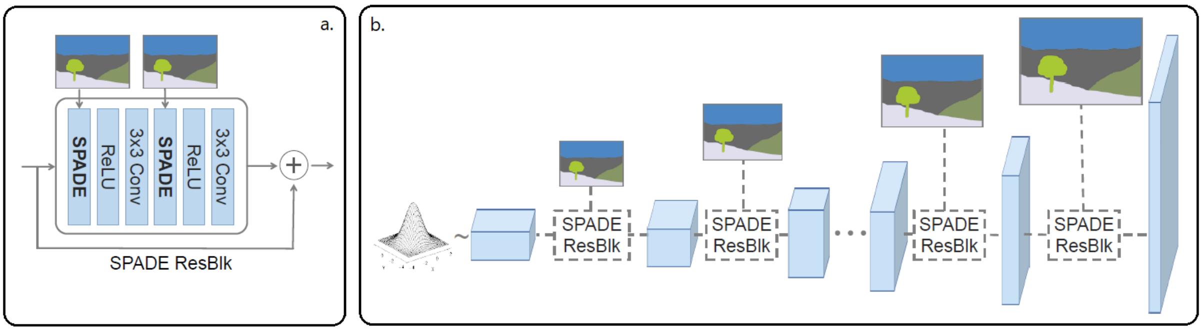

Figure 2.1: a: Residual block employing SPADE modules. b: Generator architecture. At each stage, conditioning masks having the same spatial scale are used for feature-map conditioning through SPADE modules. Figure is a modified version of [15].

γ^i and β^i are the demodulation tensors having the same spatial dimensions and number of channels as h^i, therefore each spatial location of tensor h^{i+1} is updated based on the corresponding region of the conditioning masks. The (de)modulation tensors are computed using a small DNN module consisting of several convolutional and activation layers, without employing any normalization in-between. In [15] it was empirically shown that this conditioning module prevents semantic information loss from the conditioning masks, in contrast to other types of normalization layers.

The Generator network architecture is similar to an image decoder (see Fig. 2.1). Random noise z can be used as network input to modulate style and textures (while semantic content is dictated by the conditioning masks). After each generator stage, upsampling operations are employed to successively increase feature-map resolution. Each stage consists of SPADE residual blocks. The conditioning masks are rescaled to match the resolutions of each generator stage and are used for conditioning inside the residual blocks. The target output resolution was 384x512 grayscale. Six upsampling stages were used to increase the resolution from an initial size of 6x8.

Three multiscale discriminators were employed to perform patch-wise classification between real and synthetic images at multiple scales: original scale, 0.5x and 0.25x. The input was the concatenation between the conditioning masks and a real/synthetic echo image. The first scale discriminator has a Receptive FoV $\sim 18\%$ of the height size and therefore focuses more on texture, while the smallest scale discriminator has a FoV $\sim 73\%$ of the height size and thus focuses more on the shapes/structures and semantic content.

The Wasserstein GAN [16] variant was employed for training. Starting from the Earth-Mover distance, applying the Kantorovich-Rubinstein duality leads to the following adversarial loss for the discriminator:

$$\max_w \mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))] \tag{2.3}$$

and for the generator:

$$\max_\theta \mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))] \tag{2.4}$$

where w and θ are parameters for the (multiscale) discriminator f_w and generator g_θ, respectively; $\mathbb{P}_r$ is the distribution of real data samples and $z \sim p(z)$ is the distribution of latent variables.

The discriminator has no output activation function. In order to apply the Kantorovich-Rubinstein duality, the discriminator is required to be a 1-Lipschitz continuous function [17]. To ensure this constraint, multiple methods have been proposed in literature, such as weight clipping [16]; gradient penalty loss [18]; spectral normalization [19]; etc. Herein, spectral normalization is employed inside the discriminator.

To obtain a scalar loss value for all discriminator stages, the losses are computed and averaged across all multi-scale spatial patches. Additional loss components, such as discriminator or perceptual feature matching, have been employed [15] to boost the quality of synthesized samples.

Spectral normalization was employed also in the generator network. The latent space dimension was $z \in \mathbb{R}^{256}$. An initial fully connected layer projected z onto a low resolution 6x8 feature map used as input into the first generator stage. Adam optimizer was employed with a two-times update rule, in which the discriminator learning rate was 4 times larger than the generator's.

In a first experiment, a model was trained on a dataset comprising A2C echocardiographic views. The conditioning masks had 3 channels: LV, LA and background. One can observe that the model manages to produce photo-realistic outputs while respecting the segmentation masks, however, each echo frame is modeled independently and therefore ED/ES frame pairs may be synthesized with different appearance (i.e. textures, LV papillary muscles, etc.) between ED/ES, depending on how much the segmentation masks have changed between the two cardiac phases. Also, the opening of the mitral valve cannot be conditioned, since the generator cannot reliably infer the cardiac phase from the conditioning segmentation masks. A solution to this issue is to jointly model pairs of ED/ES frames, instead of disjoint frames. Both generator and multiscale discriminators need to be modified:

- Generator: the same latent z is employed for both ED/ES frames. Each residual block is applied twice, sharing all convolutional parameters excepting the layers inside the SPADE (de)normalization blocks; there are separate layers computing modulation coefficients $(\gamma^i_{ED}, \beta^i_{ED})$ and $(\gamma^i_{ES}, \beta^i_{ES})$. The classic BatchNorm applied before the SPADE modulation processed both ED and ES frames inside the same batch.

- Discriminator: the final layer in each multiscale discriminator is removed and each discriminator is applied twice (once for ED and once for ES frames); their output is concatenated along the channel axis and fed into a fully convolutional output stage which does spatial patch-wise classification as before, except that each patch is computed based on the corresponding patches from both ED and ES tensors.

The multiscale discriminators will try to distinguish if frame pairs are either real or synthetic, enforcing that the generator produces ED/ES frames which have coherent appearance, i.e. the frames have similar positioning between ED/ES, the mitral valve should have different positions according to the corresponding cardiac phases and textures/speckles are similar. Two such models were trained on datasets composed of A2C and A4C views, respectively. Fig. 2.2 shows sampling examples from both models. One can observe that synthesized frames are photo-realistic and relevant anatomy structures are generated consistently between ED and ES frames.

.

2.4 Cardiac Phase Detection on Echocardiographies using Recurrent Neural Networks

2.4.1 Introduction

Cardiac phase detection is an important step in auto-EF solutions. Typically, an echocardiography recording contains multiple heart cycles. From each one, of interest are the ED and ES frames, which would be used in downstream DNN models.

Compared to other medical imaging modalities such as coronary angiographies, echocardiographies tend to have a much wider variation for the frames-per-second (fps) rate of video acquisition. Also, the heart movement depicted such as valves rapidly closing and opening may cause subsampling to a constant low fps rate to miss out important movement patterns (due to sampling aliasing effect) and therefore cause sub-optimal DNN performance when detecting the cardiac phase. Using echocardiographies at original fps values raises additional problems:

- the network must account for the wide fps ranges. On videos with high fps rates, the heart structures movement is spanned over many more frames than compared with low fps video (where there are much more pronounced changes going from one frame to the next).

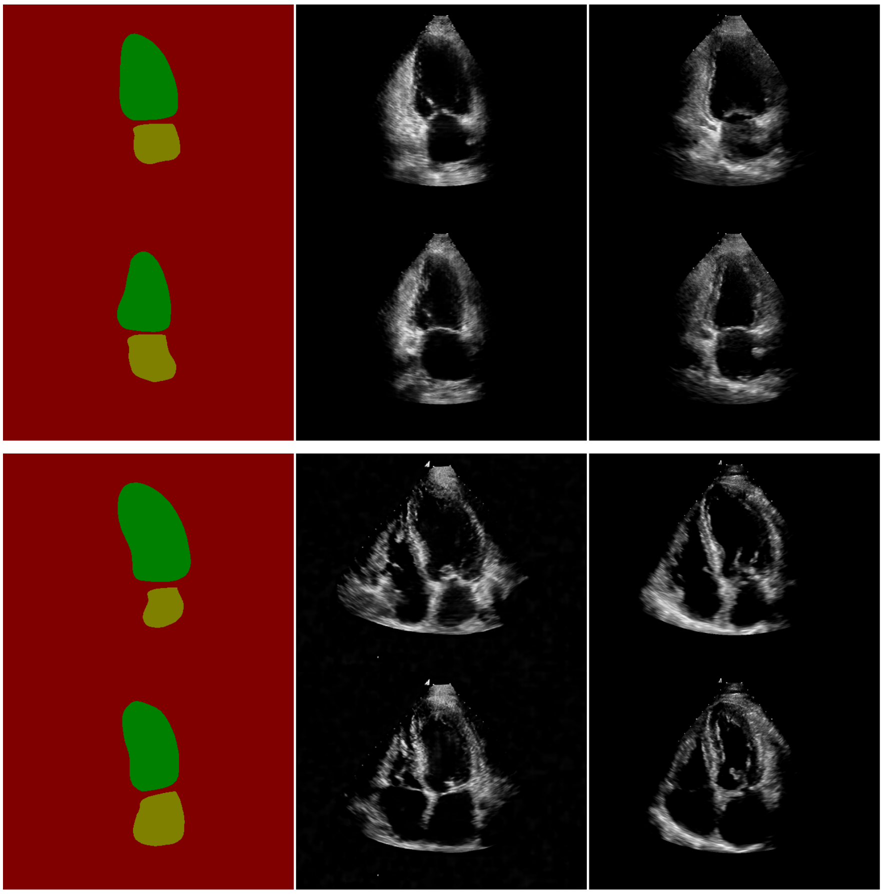

Figure 2.2: Examples of synthetically generated A2C (top group) and A4C (bottom group) ED/ES pairs (middle column) based on conditioning masks pair (left column: green is LV, yellow is LA, red is background). Right column shows equivalent real data samples. In each plot subgroup, top rows show ED while bottom ones ES.

- ideally, the prediction for one frame should be computed based on semantic content from all the other frames. Sliding window approaches only capture local temporal content. As the number of frames inside an acquisition is highly heterogeneous, the network must account for variable-length echocardiographies.

2.4.2 Methods

Recurrent neural networks are a natural solution to the above issues, as they can operate on arbitrary length sequences. The fps value can be injected as an additional model input, allowing the model to cope with the wide fps intervals usually encountered in datasets. A novel architecture is proposed, allowing for the determination of the end-diastolic and end-systolic frames solely from medical images, without requiring additional input signals (e.g. ECG, etc.).

The ground truth may be generated either manually, i.e. by expert annotators, or automatically, e.g. from simultaneously acquired signals like the ECG. In the latter case, special care must be taken to ensure that the signal is noise-free, well synchronized, and the algorithms for ED / ES detection based on ECG are accurate enough.

In the following experiments, the ground truth signal consisted of a single binary signal along with a loss-weighting 1D signal (generated by subtracting Gaussians centered on the ED and ES frame indices from a constant signal; the Gaussians' standard deviation was defined as a function of systole/diastole length expressed in number of frames). For a wide range of patient heart beats-per-minute values, the diastole has a longer time duration than the systole. In order to have a balanced systole-diastole contribution to the aggregated loss value, the systolic duration was further weighted

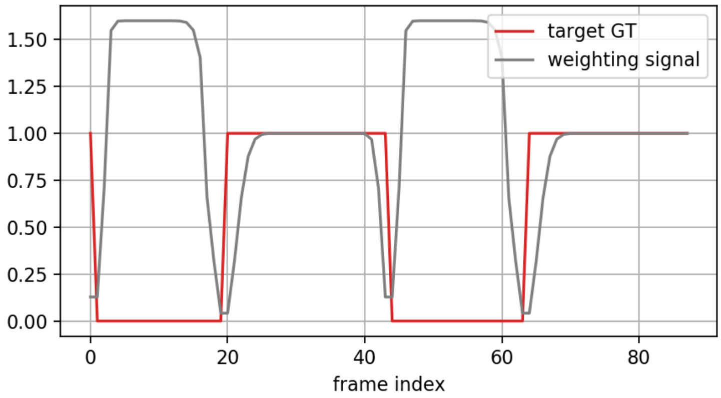

Figure 2.3: Example GT signal from a sample used in training the recurrent network.

inversely proportional to the mean systole/diastole ratio across the training set. Fig. 2.3 shows an example of ground-truth signal along with its per-frame-loss weighting signal. During training, binary cross-entropy loss was computed at each frame index and the weighted sum of all individual frame-losses yielded the training sample loss.

The model architecture combines both convolutional and recurrent sub-networks and is capable by design to generate the prediction both a posteriori (i.e. once the entire acquisition becomes available) or in real-time (as soon as new frames are acquired, the model can be applied to generate prediction for the current time-step and possibly update past predictions).

In the first stage of the network, each frame of the acquisition is run independently through a convolutional neural network. This encoder takes grayscale input images of fixed size, and outputs one-dimensional feature vectors, e.g. of length 64. The encoder employs simple layer structures consisting of 2D convolutions followed by max pooling, each one reducing the input size by a factor of two and creating higher level features. A final, fully connected layer, responsible for generating the output, takes as input feature volumes at two different spatial scales. The goal is to enhance the encoder performance in the face of ventricular size variability (due to zooming, view type, etc.). In the next stage, chunks of adjacent feature vectors (e.g. tuples of 3) are run through a temporal CNN. This CNN consists of two layers: a separable convolutional layer, and a 1D convolutional layer. The goal of this CNN is to output features capturing the local variation in time between adjacent frames. For each batch of spatial-feature vectors a temporal-feature vector of greater length is generated (e.g. 128).

Next, these temporal feature vectors are passed through a bidirectional recurrent neural network. The same RNN parameters are used for both forward and backward passes. The idea behind this RNN is to perform a nonlinear averaging in time of the local variation between image frames, thus allowing the network to process image sequences recorded at different frame rates. The internal state of the RNN is outputted at each time instant, for both directions, feeding a time-distributed fully connected layer. This dense layer acts as a temporal-feature detector, e.g. for detecting changes in ventricular volume or mitral valve position.

The output of this detector is fed into another bidirectional RNN, now with separate parameters for the two forward / backward passes. The idea behind this RNN would be to learn the normal alternation between systole and diastole. The final network layer is a distributed fully-connected binary classifier with a sigmoid activation. It outputs the probability of each component frame to be part of a diastole. The network can be trained end-to-end. Different activation nonlinearities may be employed, e.g. 'SeLU' (for convolutional layers), sigmoid and tanh (for recurrent layers). Data augmentation and drop-out layers may be employed as means for avoiding overfit. The building block of the RNN layers was the LSTM unit, which was initialized at training start to alleviate the vanishing gradient problem which usually happens for long sequences, by using a large forget-gate bias.

Table 2.2: Recurrent cardiac phase detector performance on the test set. TP and FN count how many GT transitions were detected or missed, respectively. Frame index errors are computed for the detected transitions, with distances normalized at a reference fps rate of 30. The last two columns measure the percentage of detected transitions which are ± 2 or ± 1 frames away from the GT, respectively.

Phase	View	TP	FN	Recall [%]	Mean abs. diff. ± std [frames]	Mean diff. ± std [frames]	Accuracy [%] ($\pm$2 fr. tol.)	Accuracy [%] ($\pm$1 fr. tol.)
ED	A4C	50	0	100	0.363 ± 0.410	0.158 ± 0.524	100	96
	A3C	50	0	100	0.377 ± 0.556	-0.129 ± 0.659	98	92
	A2C	49	1	98	0.368 ± 0.758	-0.117 ± 0.834	97.96	95.92
ES	A4C	50	0	100	0.584 ± 0.591	-0.099 ± 0.825	100	84
	A3C	49	1	98	0.752 ± 0.688	-0.173 ± 1.004	93.88	83.67
	A2C	50	0	100	0.718 ± 0.566	-0.547 ± 0.733	100	82

2.4.3 Results

After training for several tens of epochs on $\sim$1000 cases, performance was evaluated on the test set employed in section 2.5 (which consisted of 50 patients, each having one acquisition for each cardiac apical view: A2C, A3C and A4C). An expert reader annotated one cardiac cycle per each acquisition. Table 2.2 shows the recurrent model detection performance, for each view and cardiac phase. Frame index distances between GT annotations and predicted frames were computed and the values were normalized at 30 fps (to allow metrics on acquisitions with vastly different framerates to be directly comparable).

For any GT transition, if there is not exactly one predicted transition inside a predefined temporal window, then that case is counted as a False Negative (FN). One can observe that the model shows good detection performance, as $> 90\%$ of cases have the predicted transition frames less than 2 (in normalized distance) frames away from the GT annotation. Performance is similar across the views and mean error values indicate that there is no prediction bias, for neither cardiac phase.

2.5 Semantic Segmentation and Landmark Detection

2.5.1 Introduction

Heart chamber contouring is a core task in auto-EF solutions. A contour can be approximated by an array of 2D points. Such arrays can be obtained from interpolating between predicted landmarks, extracting the edge of predicted segmentation masks or by employing heuristics which combine both types of predictions. In the following experiments, the investigated DL solutions had to accomplish both goals:

- **Segmentation**: produce 3 segmentation maps for LV, LA and background.
- **Landmark detection**: produce spatial probabilities for 6 points of interest (LV and LA each having 2 annuluses and one apex).

The input resolution was set at 320px width by 256px height, single channel (grayscale images). Starting from a collection of annotated medical acquisitions, pre-processing techniques were employed to construct datasets suitable for DL model development and evaluation. The ultrasound cone was cropped, the resulting image was converted to grayscale and resized. Each medical acquisition was a 3D pixel buffer with image height and width as the first two axes and the time dimension as the last axis. From each buffer, only the frames around the ED and ES timepoints were extracted and annotated by expert readers. Experiments were conducted using datasets consisting of the 3 apical views (A2C, A3C, A4C). The train and validation sets had more than 12000 frames from 1800+ dicoms. The test set consisted of 50 patients, each having one dicom for each view. The annotations consisted of 17 points around the LV and atrium walls, where the first and last points were the two annuluses and the middle point was the LV/LA apex. Target segmentation maps were generated by filling the polygon obtained from a spline interpolation of the 17 annotated points.

2.5.2 Methods

2.5.2.1 Baseline Encoder-Decoder Architectures

The first approach is a baseline architecture where the segmentation and landmark detection tasks were treated separately. The DNN architecture is a basic cascade of convolutional layers forming an encoder (where the image resolution is progressively downsampled and higher level features are computed) and a decoder (where the original input resolution is recovered through successive upsampling operations, starting from the encoder output). The two network parts have 5 stages each, consisting of sequential blocks of convolution, batch normalization, activation and max pooling in the encoder, and bilinear upsampling, convolution, batch normalization and activation in the decoder. All convolution kernels had 3x3 size with stride 1x1. After each encoder/decoder stage, the resolution is decreased/increased by a factor of 2x, while the number of feature channels is doubled or halved, respectively.

Considering the case of training separate models for each goal, the SoftMax activation can be used as the output layer's activation function, for both models. For segmentation, the number of output channels is equal to the number of segmented items (in this case 3: LV, LA and background). The Softmax operation is applied channel-wise, independently for each pixel. The resulting tensor has the same spatial size as the input image, wherein each pixel contains a discrete probability distribution over the 3 segmentation classes. The DICE loss was employed during training:

$$DICE(P_i, T_i) = \frac{2\sum_j P_{i,j}T_{i,j} + \epsilon}{\sum_j P_{i,j} + \sum_j T_{i,j} + \epsilon} \tag{2.5}$$

where P_i and T_i are the predicted and target (spatial, unnormalized) probability distributions, respectively, for the object modeled on channel i. ϵ is small constant used for numerical stability. The sums are computed over all spatial locations j inside the output/target channel i. The final loss value is the sum of all DICE coefficients across all channels.

For landmark detection, the number of output channels is equal to the number of target landmarks. The SoftMax activation is applied spatial-wise across the HxW dimensions, independently for each channel (landmark). Intuitively, the resulting tensor is a spatial probability distribution similar to a heatmap, where plausible landmark locations have large values, while other regions have low probability values. The employed loss function was Adaloss [6]. In this framework, target point-locations are approximated by Gaussian Distributions centered on the ground-truth locations, but with varying standard deviation values. In the beginning of training, localization is (still) a new and hard problem for the DNN model, therefore the target distribution is relatively wide, i.e. the requested landmark localization precision is relatively low. As training progresses, the framework monitors the loss function value over a preset window of the last T epochs. If the loss variance is decreasing, then it means that the training is converging at the current difficulty setting and thus the problem can be set harder by reducing the standard deviation of the target Gaussian, therefore requesting higher localization precision from the DNN model. The loss value's statistics are continuously monitored and the target Gaussians are updated concordantly, in order to avoid oscillatory or divergent training behavior.

The advantage of this baseline architecture is its simplicity. The downside is that the model may produce coarse segmentation maps, as the fine spatial details may get lost. The encoder cascade causes the feature map resolution to get progressively smaller, therefore possibly sacrificing fine spatial details such as sharp edges. While the decoder does upsample successively the feature maps in order to recreate the original resolution, at each stage only the inbound features from the previous stage are transformed; therefore if some spatial details are lost in the preceding layers due to pooling, it will not be possible to recover them due to the sequential nature of this architecture. In [7] it is shown that reusing denser feature maps (produced by earlier layers inside the encoder) in the decoder may recover spatial details and thus produce segmentations with higher fidelity.

2.5.2.2 Using Skip-connections and Multi-task Learning

Based on principles outlined in [7], [5] introduces "UNet", a fully convolutional DNN employing skip connections between every encoder-decoder stages operating at the same resolution. At each decoder stage, the matching encoder feature map is reused by concatenating it to the current decoder feature map. Upsampling the bottleneck feature map retains the large FoV attached to each spatial pixel, while the matching encoder-generated feature map contains lower-level features which are richer in spatial details. Therefore, the concatenation of these 2 types of feature maps produces tensors which contain both semantic information (relative to global image content) and texture/shapes information (relative to local image content). The end result is a network capable of segmenting fine details in a coherent manner (e.g. without predicted mask artefacts such as holes or leakages).

The task of heart chamber segmentation and landmark localization are related, since in this example the landmarks are part of the annotated contour which generates the target mask. Therefore, the two task share structure, in the sense that features learned for one task may also generalize to the other. Instead of training two different model instances, computations can be shared to yield one model which produces the logits for both tasks.

Fig. 2.4 shows an improved DNN architecture combining the above ideas. The network components are:

- one encoder with 6 stages (2x downsampling operations)
- a bottleneck layer containing 3x3 and 1x1 conv2D layers
- 3 decoders, each with 6 stages (2x bilinear upsampling operations). The first decoder has 3 output channels and is responsible for heart chamber segmentation. The last 2 decoders have

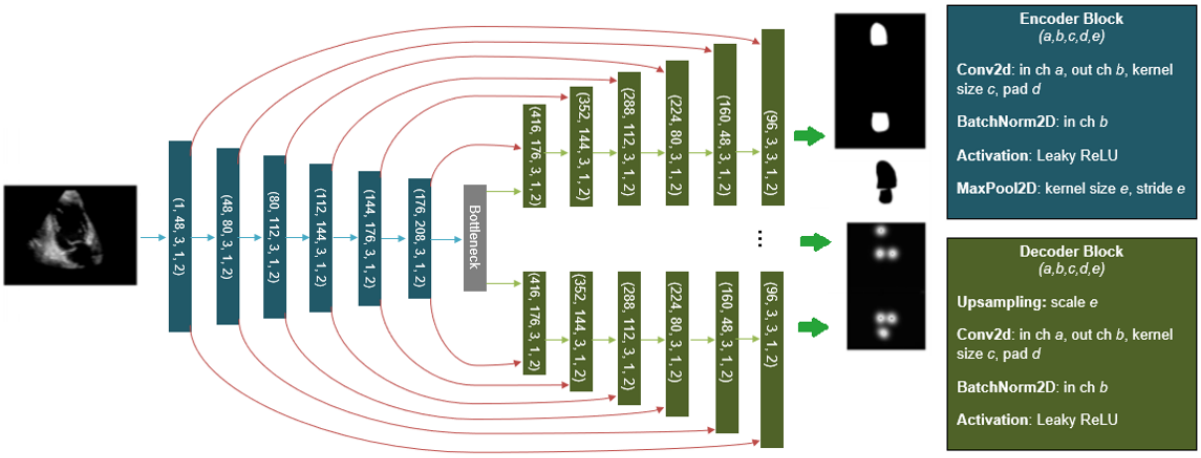

Figure 2.4: Improved DNN architecture which employs skip-connections and multitask learning.

3 output channels each and each one is responsible for detecting the 3 landmarks for LV and LA chambers, respectively.

The output from an encoder stage is used 4 times: once for the downstream encoder stage and once for each decoder as a skip connection. The same loss functions are employed as in the previous section. The first head has a DICE loss, while the last 2 each have an Adaloss. The final loss value is a weighted sum of the 3 sub-losses. Due to the skip connections, the network can be made deeper without sacrificing spatial fidelity in the outputs. The effective receptive FoV can therefore increase, as each output pixel is computed as function of its corresponding image patch which ideally should cover the entire image, at every output pixel position.

The multitask learning setup has several advantages:

- no need to run multiple training procedures (i.e. by not having one model instance for each task).
- inference runtime requirements are lowered due to shared computation.
- supervision for one task can increase performance for the other tasks.
- the decoder heads can produce outputs which are more spatially correlated than dealing with predictions coming from independent models. Having multiple heads sharing the same encoder yields a model which is much more likely to have better spatial correlation in its outputs.

2.5.2.3 Predicting on ED and ES Frames Simultaneously

As mentioned in a previous section, to compute an EF estimate, predictions are required both on ED and ES frames from the same medical acquisition. To obtain robust estimates of EF, robust estimates of individual volumes are needed. Specifically, there should be no bias between the way ED and ES contours are predicted. During the medical video acquisition, the heart is beating and the captured views show the relevant anatomical movement. However, due to the transducer's placement on the subject, some heart structures can be fully visible during one cardiac phase, but be missing in the other. For robust and consistent contouring, the model should segment similarly the ED and ES frames in this visually uncertain region. E.g., it may be possible to infer some missing information from the ES frame and apply it to the ED segmentation, in order to maximize the consistency of contouring between ED and ES frames.

Previous architectures did not explicitly link the processing of the 2 ED/ES frames, but only implicitly through the batch normalization layers, as the ED and ES could be put together to form one batch. Therefore, a new architecture is proposed, building on top of the one from the previous subsection. Fig. 2.5 depicts the overall architecture. The same encoder is applied once on a batch consisting of the 2 input frames. After the bottleneck, the feature maps are reshaped by splitting the batch axis in half and doubling the channels axis. Independent 1x1 conv2D "merging" blocks are applied to get 2 embeddings for the ED frame and for the ES frame, respectively, starting from the concatenated features of both frames. Each decoder head has its own sets of "merging" layers, possibly acting on multiple spatial scales.

The training procedure is the same as in previous sections. The model can be trained end-to-end in a multitask setup. This revised architecture allows mutual conditioning between output ED and ES logits. The net result of the proposed architectural changes is a model capable of superior performance in the 2 main tasks, which can provide robust and consistent contouring therefore allowing more precise estimation of EF values.

2.5.3 Results

Tab. 2.4 and 2.3 show performance comparisons for the 3 described architectures on the test set, for segmentation and landmark detection, respectively. Compared to the baseline, both updated models show superior performance on both tasks across views, chamber and cardiac phases.

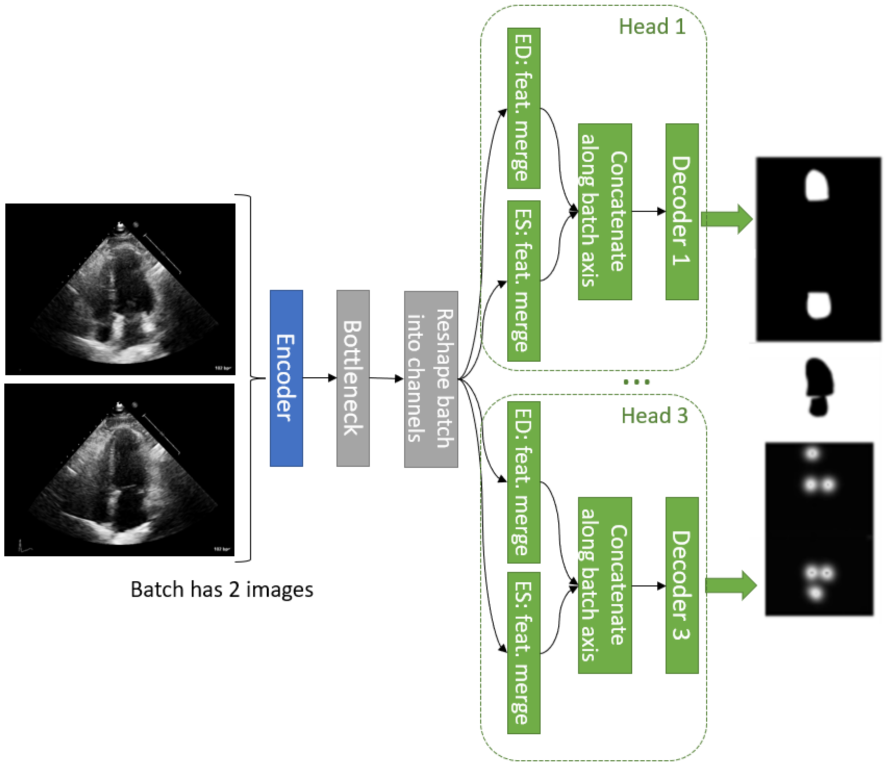

Figure 2.5: DNN architecture which explicitly links the processing of ED and ES frames. Merging layers are used to construct features for ED and ES frames, respectively, employing feature maps from both frames.

Table 2.3: Performance comparison on landmark detection. Each cell shows euclidean distances between predicted and ground-truth locations, averaged over the 3 landmarks. Errors are measured in pixels (relative to the input resolution of 320x256). (SC = skip connections, MT = multitask learning, ED/ES = frame conditioning)

View	A2C				A3C		A4C				Mean Lmk Error
Chamber	LA		LV		LV		LA		LV		
Phase	ED	ES	ED	ES	ED	ES	ED	ES	ED	ES	
Baseline	5.6	5.5	4.9	5.0	5.9	5.6	5.9	5.7	4.7	4.9	5.4
SC & MT	5.2	5.2	4.2	4.4	5.4	5.3	5.4	5.0	4.2	4.4	4.9
SC, MT & ED/ES	5.3	5.1	4.1	4.3	5.4	5.2	5.5	5.0	4.1	4.3	4.8

Table 2.4: Performance comparison on heart chamber segmentation.

View	A2C				A3C		A4C				Global DICE
Chamber	LA		LV		LV		LA		LV		
Phase	ED	ES	ED	ES	ED	ES	ED	ES	ED	ES	
Baseline	90.18 ± 5.2	91.66 ± 4.0	91.72 ± 2.7	90.16 ± 4.1	91.61 ± 3.9	90.09 ± 6.1	88.60 ± 6.7	90.23 ± 5.6	92.02 ± 2.6	89.35 ± 4.3	90.56
SC & MT	90.56 ± 4.6	92.56 ± 3.3	92.66 ± 2.5	91.10 ± 2.8	92.33 ± 3.2	90.54 ± 3.6	89.89 ± 6.4	92.20 ± 5.4	93.5 ± 2.4	90.39 ± 4.0	91.57
SC, MT & ED/ES	90.69 ± 5.0	92.85 ± 2.9	92.54 ± 2.1	90.88 ± 2.9	92.48 ± 3.2	90.51 ± 4.3	90.19 ± 6.5	92.08 ± 5.3	93.66 ± 2.0	90.96 ± 4.1	91.68

Table 2.5: EF correlation metric comparison between the updated architectures.

Architecture	Single-Plane EF		BiPlane EF
	A2C	A4C	
SC & MT	0.82	0.83	0.84
SC, MT & ED/ES	0.86	0.86	0.88

Even though the updated models perform similarly in terms of DICE and L_2 landmark errors, when considering the correlation of predicted versus ground-truth single-plane and bi-plane EF values, the mutual conditioning of the last model yields an improvement in the Pearson product-moment correlation coefficient, as seen in Tab. 2.5.

In conclusion, the heart chamber contouring task can be successfully automated inside an EF estimation solution, yielding good performance when compared against expert annotators.

2.6 Uncertainty in Deep Neural Networks

2.6.1 Introduction

Regular deep neural networks tend to be overconfident in their prediction, even when it is completely wrong. Their prediction is a point-estimate, i.e. only one tensor is provided as output for each input sample. While the notion of uncertainty affects both the range of actual possible output values and downstream tasks, it is completely ignored by classical DL algorithms. Fortunately, simple extensions allows practitioners to infer more information from their already-trained DL models.

Uncertainty in the predicted outputs depends on the input uncertainty and on the model itself (on the architecture and on the trained weights). Ideally, output uncertainty should be high only for certain reasons:

- the query input sample is out-of-domain (OoD), i.e. is inside a region in the input space not covered by any training points.
- the query input sample is placed near the learned class separation border.
- especially for higher dimensional inputs (where sparsity is even more prevalent given the finite nature of training data [13]) the query sample can be far away from neighboring training datapoints, even though it may be placed ”inside” the training distribution.

At a basic level, many supervised DL tasks can be decomposed into 2 main categories:

- **Regression**: output uncertainty is usually approximated using a Gaussian distribution. Its mode is the most likely value while the standard deviation is a direct measure of confidence.
- **Classification**: the entropy of the resulting probability distribution is an indicator of confidence. High entropy is associated with a uniform distribution over possible classes, while low entropy occurs when the bulk of the probability mass is placed only on one class.

Augmenting the prediction with uncertainty information is valuable for any DL workflow, especially in the medical domain. Automated medical imaging analysis software which offers pre-computed measurements based on visual detections should be able to asses the quality of its output and of its underlying DNN predictions. Recent research has shown that uncertainty estimates can be extracted from already trained models using simple techniques. If, e.g., batch normalization or dropout layers have been employed during model training, the inference procedure at test time can be augmented to also yield output uncertainties along with the model predictions, by running multiple forward passes for the same test sample.

Subsequent sections present methods that have the advantage of requiring a single forward pass to get both the prediction and the attached uncertainty. Semantic segmentation, as a task, can be decomposed into a set of mini-classification tasks at pixel level. Uncertainties can be estimated in parallel for all pixels, since the network outputs logits for all pixels in parallel.

2.6.2 Energy-based Models

It is shown [10] that a classification model having a softmax final activation contains implicitly an input density estimator. An energy function is proposed, which employs the same input logits as the softmax layer:

$$E(x; f) = -T \log \sum_{i}^{K} \exp\left(\frac{f_i(x)}{T}\right) \tag{2.6}$$

where T is the softmax temperature and $E(x; f)$ is the attached energy value of input x under the model f.

The energy score is linearly aligned with the log-probability density of the input samples:

$$\log p(x) = \frac{-E(x; f)}{T} - \log Z \tag{2.7}$$

where $\log Z$ is constant for all inputs x.

A simple criterion for OoD detection can be formulated based on the observed energy scores on the validation set: a threshold τ can be derived and compared with energy scores of test samples. Usually, negative energy scores are used, to align with the conventional definition where positive (in-distribution) samples have higher scores [10]:

$$G(x; \tau, f) = \begin{cases} 0 \text{ (OoD)}, & \text{if } -E(x; f) \leq \tau, \\ 1 \text{ (in-distrib)}, & \text{if } -E(x; f) > \tau. \end{cases} \tag{2.8}$$

The energy score can also be incorporated into the training objective, along with the categorical cross-entropy classification loss. The training dataset would be consisting of 2 parts:

- an in-distribution subset, on which the classification loss is computed and for which the energy score is minimized.
- an OoD subset, on which only the energy score is computed and maximized.

The energy method is applicable to an already-trained model, extending it as an OoD classifier. A segmentation and landmark detection model trained on Apical BMode echocardiographies (with its architecture presented in section 2.5.2.2) was applied on a test set containing apical echocardiographies graded on a scale of 1 to 5 with respect to LV region image quality (1 - best image quality, 5 - worst image quality). Two bins were considered:

- grade 1 & 2 & 3: acquisitions where the LV is fairly visible, with reduced pixel noise, etc. Only these levels of quality were employed in the model's original train set.
- grade 4 & 5: acquisitions where some parts of the LV wall regions are not visible, high image noise, poor transducer positioning, etc. Data of this low quality never reached the model during training, therefore it should be out-of-distribution with respect to higher quality data.

Only the segmentation head of the pretrained model (i.e. not finetuned on energy scores) was considered. As semantic segmentation involves pixel-level classifications over the entire input image, the softmax logits on each pixel location can be employed to compute pixel energy scores. Fig. 2.6 shows an example segmentation with the attached energy map. Eq. 2.7 states that the energy score is linearly aligned with the likelihood of the input sample. In the segmentation context, each pixel is classified based on its effective FoV-sized patch from the input.

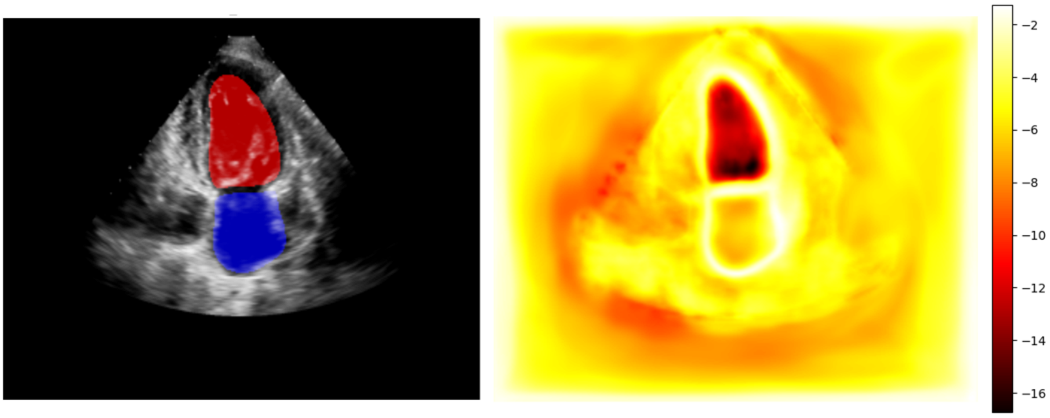

Figure 2.6: Example LV (blue mask) and LA (red mask) segmentation for an A2C acquisition (left). Right subplot shows pixel energy scores. Around the segmented masks edges, the energy scores have relatively large values.

Therefore, the energy score of a pixel indicates the likelihood of its corresponding FoV-sized patch (from the input image) to be inside the distribution of image patches seen during training. To compute an energy score describing the entire LV region, a Region-of-Interest (RoI) was constructed on top of the LV chamber using the model-predicted LV contour and landmarks. The energy scores pertaining to pixels inside this RoI can be averaged out to obtain an estimate of the whole-region uncertainty. The classification capabilities of the model between the two LV-region visual quality grade bins was tested. By varying the threshold employed to separate the two score distributions, the AuRoC metric obtained by the pretrained model was ~ 0.81.

The pixel-wise energy scores can also be averaged-out across the entire image, yielding a global measure of sample uncertainty, useful when tested against datasets which potentially may be very different compared to the training set. On such data, regular predictions (i.e. segmenting the LV chamber) may not be performed due to the lack of appropriate semantic content. This procedure can act as a data curation mechanism.

2.6.3 Sparse Gaussian Processes

2.6.3.1 Gaussian Process Formulation

A Gaussian Process (GP) represents a distribution denoted as $\mathcal{GP}$ over real-valued functions $f(\cdot) : \mathcal{X} \to \mathbb{R}$ defined over an input domain $\mathcal{X}$ [12]. Analog to a multivariate Gaussian which represents a distribution over finite dimensional vectors, a GP represents a distribution over uncountably infinite-dimensional functions: vector indexes in multivariate Gaussian random variables conceptually correspond to specific evaluation points $X \in \mathcal{X}$ in GP random functions [12].

A GP is defined through 2 real-valued functions:

$$f(\cdot) \sim \mathcal{GP}\left(\mu(\cdot), k(\cdot, \cdot')\right) \tag{2.9}$$

where $\mu(\cdot)$ is the mean value of function f and $k(\cdot, \cdot')$ forms the covariance matrix of function f evaluations at points $(\cdot)$.

Considering a partition of the uncountably infinite set of random variables represented by a GP into two sets:

- one containing a finite subset denoted as **u** evaluated at a finite set of evaluation points **Z** $= Z_1, ...Z_M \in \mathcal{X}$, such that **u** $= f($**Z**$)$, with mean μ_u and covariance matrix $\mathbf{K}_{uu}$.
- one that contains the remaining uncountably infinite set of random variables denoted as $f(\cdot)$ evaluated at all locations in $\mathcal{X}$ excluding the points in **Z**.

The GP can be rewritten as:

$$\begin{pmatrix} f(\cdot) \\ \mathbf{u} \end{pmatrix} \sim \mathcal{GP}\left(\begin{pmatrix} \mu(\cdot) \\ \mu_{\mathbf{u}} \end{pmatrix}, \begin{pmatrix} k(\cdot,\cdot') & \mathbf{k}_{\cdot\mathbf{u}} \\ \mathbf{k}_{\mathbf{u}\cdot'} & \mathbf{K}_{\mathbf{uu}} \end{pmatrix}\right) \tag{2.10}$$

where $\mathbf{k}_{\cdot\mathbf{u}}$ and $\mathbf{k}_{\mathbf{u}\cdot'}$ denote vector-valued functions that express the cross-covariance between the finite-dimensional random variable u and the uncountably infinite-dimensional random variable $f(\cdot)$, i.e. $\mathbf{k}_{\cdot\mathbf{u}}$ is a row vector valued function equal to $k(\cdot, Z_m)$.

The conditional GP of $f(\cdot)$ conditioned on **u** is:

$$f(\cdot)|u \sim \mathcal{GP}\left(\mu(\cdot) + \mathbf{k}_{\cdot\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{u} - \mu_{\mathbf{u}}), k(\cdot,\cdot') - \mathbf{k}_{\cdot\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}\cdot'}\right) \tag{2.11}$$

In above equation, **u** can be a vector of scalar measurements (e.g. target ground truth values) corresponding to a set **Z** of training input samples. $\mu(\cdot)$ can be set to zero and kernel k is applied on all pairs $(\cdot, Z_i)$, yielding the required matrices.

2.6.3.2 Deterministic Uncertainty Estimation in Deep Learning

Considering image datasets, applying standard kernels k directly on pairs (X_i, X_j) of sample images would lead to poor modeling expressiveness, since all kernel operations would be performed in pixel space. Instead, Deep Kernel Learning (DKL) [14] tries to combine the expressiveness of deep neural networks with the probabilistic prediction ability of GPs [11] by using a DNN-based feature extractor to feed a GP output layer. The kernel would therefore contain a deep feature extractor [11]:

$$k_{l,\theta} \rightarrow \bar{k}_l(f_\theta(x_i), f_\theta(x_j)) \tag{2.12}$$

where $f_\theta(\cdot)$ is a DNN parameterized by θ, $\bar{k}_l$ can be a standard kernel (such as radial basis function or Matérn) with hyperparameters l (e.g. kernel length and output scale).

The variational inference training procedure can be employed to avoid storing the feature embeddings of all N training samples and computing the inverse of an N-by-N matrix. Instead, M inducing points and M latent inducing variables are learned in the feature space, with $M << N$.

Feature collapse is a phenomenon [11] in which input sample embeddings can be placed on a low dimensional manifold, effectively collapsing certain regions of the input space into very narrow neighborhoods in the feature space, and therefore making it impossible to distinguish between some input points even though they may be placed very far away from each other. In [11] it is proposed that the feature extractor is constrained to be bi-Lipschitz to alleviate feature collapse:

$$\frac{1}{K} d_X(x_1, x_2) \leq d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2) \tag{2.13}$$

where $K \geq 1$, $f : X \rightarrow Y$ is the DNN feature extractor and d_X and d_Y are distance functions on set X and Y, respectively. Therefore the feature extractor must have 2 properties: sensitivity and smoothness.

The output of a GP output layer is a Gaussian distribution. Regarding semantic segmentation, the output is a distribution of per-pixel class logits. However, directly applying a GP output layer on the pixel-wise embeddings across the full input image is problematic due to memory and computational constrains. Instead, a classic DNN segmentation head can pre-select smaller regions in the input image which are of interest in a particular task and discard the other non relevant areas to save memory and computation.

An experiment was conducted in which a model consisting of a bi-Lipschitz deep feature extractor feeding two decoder heads was trained on the echocardiography LV-chamber segmentation task. The first head is responsible for performing a full-image semantic segmentation based on which the GP regions of interest are extracted. In this experiment, the first decoder head segments the LV chamber and its contour is extracted and dilated to obtain a thicker band around the LV. The uncertainty is likely to be more meaningful inside this "ring" surrounding the LV, because there may be multiple plausible contour paths segmenting the LV, given a specific input image.

The embeddings of pixels inside the "ring" are extracted from their feature map and reshaped into b tensors of shape (p_i, c), where b is the number of samples inside the input batch of the feature extractor, p_i is the number of selected pixels for batch sample i and c is the number of channels in the embedding feature map, i.e. the embedding size for each pixel. This pre-selection procedure models jointly pixels from the same image, but acts independently on each input image.

During each training step, the two decoder heads share intermediate features computed by the bi-Lipschitz deep feature extractor (consisting of a Spectral Normalized Residual UNet). The first head has a DICE loss while the second head averages the sample-based ELBO losses across the batch. In order to avoid data biases due to the pre-selection procedure, the second head is also applied on embeddings sets from other randomly chosen pixel locations. This way, the GP is trained not only on regions around the LV, but (in expectation) covers the entire image space.

Training can be performed end-to-end. During test time, the same pre-selection procedure can be employed. Fig. 2.7 shows an output example. The inner part of the predicted mask by the classic head can be stitched with the outer mask component (along the thick LV ring) predicted by the GP head. One can observe that the uncertainty is concentrated along the middle of the GP-modeled pixel-band, i.e. along the edge of the final predicted mask. Since the GP output is a distribution, drawing multiple samples can be used to construct multiple outer mask components and, through stitching with the constant inner part, can yield several predicted LV mask variants.

2.6.4 Conclusions

Uncertainty estimation frameworks offer valuable insight into the potential quality of a DNN model prediction. Such frameworks can also be used to further improve the model, by detecting the data subsets on which the model uncertainty is consistently high. The train set can be updated with more such data and a retraining should lower the uncertainty while improving prediction accuracy. Some uncertainty frameworks have the advantage of being readily applicable to existing already-trained models, adding value to pre-existing AI pipelines by allowing them to detect faulty inputs or outputs with too-high uncertainty at arbitrary stages.

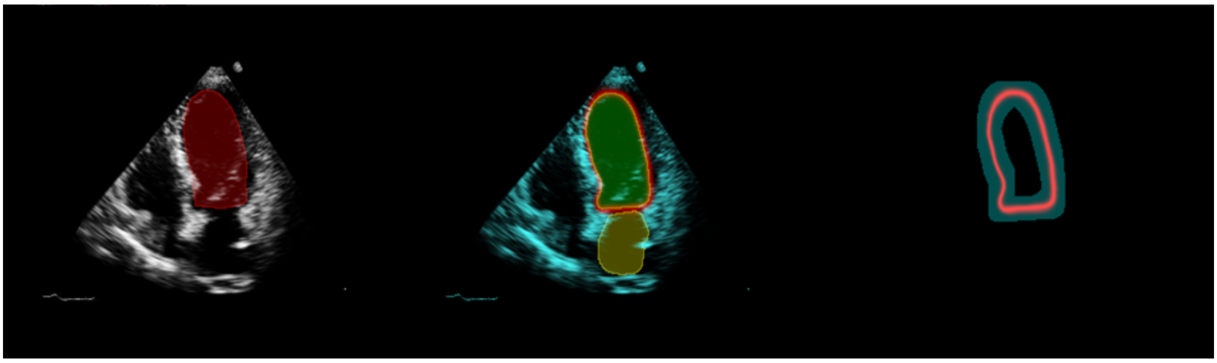

Figure 2.7: Output example from the sparse variational GP trained on the LV segmentation task. Left subplot shows the input along with the GT LV mask; middle subplot shows stitched LV (green) and LA (yellow) predicted masks; the right subplot shows the predicted mask uncertainty map (red) and the area of the GP modeled pixels (blue band).

3. Landmark Detection in 3D Echocardiographies using Deep Reinforcement Learning

3.1 Introduction

Reinforcement Learning (RL) is a category of machine learning methods in which an agent interacts with an environment based on a predefined set of actions (continuous or discrete), observes its current state and receives a reward given by the environment as an effect of a specific action taken in a specific state.

A multiscale landmark searching framework based on deep reinforcement learning was introduced in [21] for solving the task of anatomical landmark detection in CT scans. It achieved state-of-the-art accuracy while being several orders of magnitude faster than reference methods. Artificial agents are trained not only to distinguish organ appearance but also to navigate towards target landmarks along optimal trajectories.

In this chapter, the framework from [21] is adapted for detecting 6 LV landmarks on TTE BMode 3D echocardiographies. Three multi-scale agents are trained to follow optimal trajectories while searching for the landmarks inside the echo volume. The suite of RL-powered agents alleviate the need of running a 3D convolutional DNN on the entire echo volume, thus leading to computational and runtime savings.

3.2 Methods

3.2.1 Reinforcement Learning

The classical mathematical model which describes the operation of the agent inside the environment is a Markov Decision Process (MDP), which is a discrete-time stochastic control process. It consists in [20]: a set of environment states: $S_t \in \mathcal{S}$; a set of agent actions (possible at any given state s): $A_t \in \mathcal{A}(s)$; a set of numerical rewards: $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$; a function p describing the dynamics of the environment. Given a current state s and chosen action a, p yields the probability of the next state s' and returned reward r:

$$p(s', r|s, a) \doteq \text{Pr}\left\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\right\} \tag{3.1}$$

If sets $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ have finite numbers of elements, then the described MDP is finite. Subscripts t refer to the discrete timepoint. A trajectory of the MDP is a sequence of states, actions and rewards obtained by running the agent inside the environment:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, ... \tag{3.2}$$

The agent's goal is to maximize its cumulative reward. Because of the possible existence of very long episodes, the sum of rewards may tend to very large values during agent runtime, therefore the concept of discounting is used to compute the expected discounted return, provided that the agent tries to choose actions that maximize rewards:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{3.3}$$

where γ is the discount rate with $0 \leq \gamma \leq 1$.

Solving a reinforcement learning tasks involves finding a good policy π that achieves large future rewards. A policy π is considered better than (or equal to) another policy π' if its expected return is greater than (or equal to) that of π' for all states:

$$\pi \geq \pi' \text{ if and only if } v_\pi(s) \geq v_{\pi'}(s), \forall s \in \mathcal{S} \tag{3.4}$$

An optimal policy is better than or equal to all other policies. There may be more than one optimal policy. All of them are denoted by π_* and they share the same (optimal) state value function:

$$v_*(s) \doteq \max_\pi v_\pi(s), \forall s \in \mathcal{S} \tag{3.5}$$

Optimal policies also have the same optimal action-value function:

$$q_*(s,a) \doteq \max_\pi q_\pi(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A} \tag{3.6}$$

For a state s and a performed action a, this function gives the expected return if the optimal policy is followed thereafter:

$$q_*(s,a) = \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \tag{3.7}$$

3.2.2 Multiscale Agents for 3D Echocardiographies

In this section, the framework from [21] was adapted for detecting 6 LV landmarks on TTE BMode 3D echocardiographies: 4 annulus and 2 apex landmarks. Given a 128x128x128 sized echocardiography volume, the detection pipeline involves successively applying agents trained on several scales: 32x32x32 (coarsest), 64x64x64 and 128x128x128 (finest).

The environment is the grid world indexing the echo volume and a state is completely described by the current agent position inside its scaled volume. The action set $\mathcal{A}$ has 6 possible actions (see Fig. 3.1): up/down, left/right and front/back, moving the agent position one increment along the specified directions.

A state is considered terminal if either it is sufficiently close to the GT target location or if it is out-of-boundary with respect to the input echo volume. The agent observes a 23x23x23 region centered on its current location. The region chunk is extracted from the scaled echo volume and fed into the agent's Q network which predicts action-value pairs for all 6 possible actions. When close to the scale echo volume border, the observed region chunk uses zero-padding when indexing outside the echo volume.

The adopted rewarding system was the sum of 3 components: (i) one proportional to the relative change in distance towards the GT; (ii) a fixed cost for any action; (iii) a special reward whenever the GT

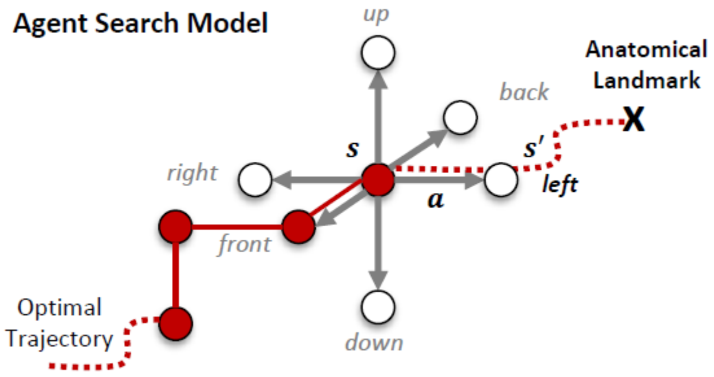

Figure 3.1: Landmark agent search model. There are 6 possible actions at each state. The state tensor is represented by an echo volume chunk centered at the current agent position. Figure taken from [21].

is reached (i.e. a positive reward) or whenever the agent goes out of volume boundary (i.e. a negative reward).

This rewarding system causes the state value function to decrease as the states get further away from the GT. If the agent reaches the GT location a large reward is given while if it steps outside the volume it is heavily penalized. As the reward is always negative (except when reaching the terminal GT) an optimal control policy is to take as few steps as needed towards the GT.

The Q network architecture was inspired from MobileNetV2 [22] and was designed to be very lightweight, having only 87k parameters. To reduce runtime to only a few milliseconds, each stage had only one inverted residual block and therefore no skip connections were possible. LeakyReLU activations were placed between all stages instead. The network had 2 input channels: the echocardiographic volume pixels and an LV-wall segmentation mask (provided by an existing upstream 3D segmentation model).

Three agents were trained on two 0.25x and 0.5x downsampled volumes and on the original volume. The first agent performs a coarse localization. As its input size is 23x23x23 inside a 32x32x32 volume, the agent has an equivalent relative receptive FoV of $\sim 72\%$, similar to bottleneck layers in a classical image-to-image DNN model. During test time, the next agents progressively refine the search location, as the starting point of the next stage is the converged point of the previous stage (see Fig. 3.2). Similar to the final layers of an image-to-image network which deal with progressively finer spatial details, as the echo volume scale grows towards the original scale, the multiscale agents learn to interpret from coarse spatial details on large FoVs (when operating on smaller scales) to fine spatial details on small FoVs (when operating on larger scales).

At all echo volume scales, every training episode draws random starting points inside a predefined fixed-size cubic neighborhood n_{start} around the GT location, while the allowed region of travel n_{travel} during an episode is larger than n_{start} by 1 pixel position on each 6 sides. Reaching a state/location outside n_{travel} was also considered terminal.

Each training episode had an imposed maximum length of 36 steps, a number proportional to the needed steps by the coarsest volume agent to reach the GT in the worst case scenario. A training epoch lumped together the simultaneous training steps of all agents and was sized such that 5% of the train set is covered during one full epoch. A complete training procedure involved 200 epochs, thus covering 10 times the entire train set. The training set included 2000 patients, each having two

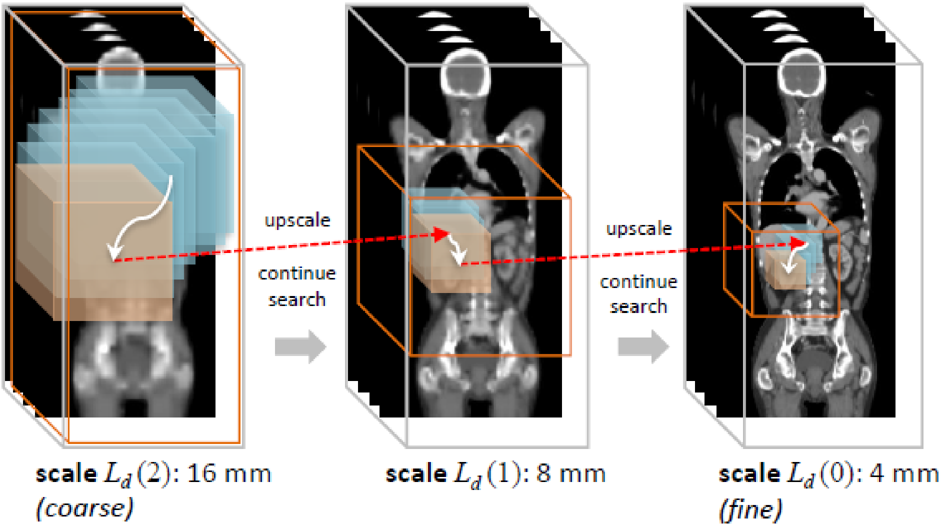

Figure 3.2: Multiscale landmark detection pipeline. The final location of the previous stage is used as starting point for the next stage. Each stage involves a 2x increase in spatial resolution. White arrows depict local search trajectories. Blue and brown cubes denote state chunks of the echo volume along the agent trajectory (i.e. input tensors into the agent's Q network). Figure taken from [21].

volumes corresponding to ED and ES time points. As training progressed, the ϵ coefficient used in the ϵ-greedy policy was linearly annealed from an initial value of 1 towards a minimum value of 0.05 across the first 80% of epochs. As a result, the policy progressively acted more greedily during training and the percentage of converged episodes increased inversely proportional with ϵ.

At test time, the GT is used only for distance metric computation and therefore it cannot be used to mark a specific location as terminal inside the scaled echo volumes. A termination criterion was therefore used to detect agent convergence:

- each visited location was placed inside a list as a tuple of volume indexes (x, y, z).
- if the current location was visited before by the agent, then a cycle has occurred on its trajectory. All locations comprising the cycle were averaged out to get the final estimate of the target landmark location (at the current echo volume scale).

The starting point for the coarsest agents was the LV-wall mask's center of mass, for all 6 landmarks. Successive agents used the previous stage converged landmark location as starting point, as in Fig. 3.2.

3.3 Results

Tab. 3.1 shows error metrics on a test set for all 6 LV landmarks when using the RL-based multiscale search pipeline. Some landmarks exhibit better detection performance, although all of them have a mean landmark error of $\leq 2.1\%$ of original echo volume size. More than 90% of cases are detected within a 5 pixels distance from the GT (on an original resolution of 128x128x128), across all landmarks.

3.4 Conclusions

This chapter has presented an approach to 3D landmark detection using a cascade of multi-scale RL agents. Using the versatile framework of deep Q-learning, three agents learned to follow optimal trajectories towards the target locations, navigating inside the echo volume. This framework had several benefits such as data efficiency and training stability. A lightweight DNN architecture was investigated for fast and robust action-value prediction.

Table 3.1: LV TTE landmark detection pixel-distance metrics (on original 128x128x128 echo volumes) using multiscale RL agents. The last column shows the percentage of test cases for which the prediction was inside a 5pixel-radius sphere around the GT.

Lmk ID	mean	std	max	min	median	% <5px
0	2.53	1.39	6.93	0.77	2.38	91.8
1	2.18	1.17	5.68	0.45	1.74	97.9
2	2.64	1.51	8.28	0.75	2.25	91.8
3	2.05	0.95	4.95	0.23	1.88	100
4	1.94	1.36	7.40	0.27	1.69	93.8
5	2.46	1.60	8.75	0.43	2.06	95.9

4. Detecting Incorrect Lumen Segmentations in Coronary Computed Tomography Angiographies[1]

4.1 Introduction

Coronary computed tomography angiography (CCTA) is an effective imaging modality, increasingly accepted as a first-line test to diagnose coronary artery disease (CAD). Advancements in CCTA have allowed for minimal radiation exposure, effective coronary characterization, and detailed imaging of atherosclerosis over time. Due to the increasing body of evidence showing the effectiveness of CCTA [28, 29], recent ACC/AHA chest pain guidelines recommend CCTA as a first line test for patients with stable and acute chest pain.

While the performance of AI based methods has improved markedly over the years, given the importance of an accurate lumen detection, semi-automated approaches are currently still being employed. Thus, the lumen is first automatically detected, and then manually inspected and edited by the radiologist if deemed necessary. This process, together with coronary artery centerline editing, required, e.g., between 10 and 60 minutes in a study assessing the diagnostic performance of ML-based CT-FFR for the detection of functionally obstructive coronary artery disease [30]. One potential approach for significantly reducing the time required for a semi-automated CCTA lumen analysis is to pre-select locations which are likely to require inspection and editing, and to present only those for review to the radiologist. Considering that a Deep Neural Network (DNN) is responsible for generating the lumen segmentation masks, this pre-selection step can be linked to the topic of confidence and out-of-distribution detection in Deep Learning.

Normalizing Flow (NF) models can be trained explicitly to model input data probability densities. Given a downstream target task *T*, if only its input data is employed for building the NF model, then estimating the likelihood of input samples for the target task can be obtained through the NF model. Input samples with low probabilities can be flagged as out-of-distribution and the target model's output should be considered unreliable, as it would operate outside its training distribution. An NF model can also be built by stacking the input samples with their expected GT output. This way, the NF model can be placed downstream of the target task and act as an Audit model, detecting cases where the previous model provided faulty predictions. In either scenario, the NF is a separate model and therefore imposes no constraints on the model responsible for the target task. NF can usually operate efficiently in both directions: forward (or inference) direction, where input samples x from the input domain X are transformed into embeddings z which are likely under a chosen distribution Z. At each layer, the input is modified towards Z and the *logDet* value (i.e. $\ln\left(\left|\det\left(\frac{\partial f}{\partial x}\right)\right|\right)$, where f is the NF) is summed with the current layer contribution. The backward (or generative) direction employs the bijection property of the NF to transform an embedding z into a synthetic sample x_{new}.

In this paper we present an approach based on NF for the OoD detection of coronary lumen seg-

[1] The following chapter describes work published in:

- **Ciusdel, C.F.**, et al., 2022. Normalizing Flows for Out-of-Distribution Detection: Application to Coronary Artery Segmentation. Appl. Sci. 12, 3839. `https://doi.org/10.3390/app12083839`

Some sections were quoted verbatim from the above reference, which represents previously published work of the author, under the PhD research program.

mentations. NF models which are built from coupling layers as proposed in [23, 24] tend to focus on local pixel correlations instead of the global semantic meaning [25, 26] and, as a result, OoD samples may in fact produce larger log-probability values than in-distribution data. We investigate the usage of a new type of coupling layer, which employs reversible 1x1 convolutions in which the filter parameters are computed based on the passed-through components. We compare the proposed architecture against a Glow-like architecture on the task of detecting mismatched pairs of CCTA lumen images and their corresponding lumen segmentations. The coronary lumen images and masks are 3D volumes stacked along the channel axis. We also employ synthetic perturbations on the binary masks and use the perturbed samples as explicit outliers to further shape the learnt probability density of "correct" image-mask pairs. The end goal is to flag those samples for which the given segmentation does not properly match with the lumen image. Overall, we assess the performance of the NF models as follows: (i) against the synthetic mask perturbations, and (ii) using expert annotations.

4.2 Methods

4.2.1 NF Architectures

We investigated the use of a Glow-style NF architecture, combining layers previously introduced in [23, 24], such as checkerboard and channel masking coupling layers, invertible 1x1 Convolutions, Split and Squeeze layers. Our baseline network is depicted in Fig. 4.1 and described in Tab. 4.1. We employed affine coupling layers as in eq. (4.1), where x and y are the input and output tensors, respectively. Subscripts a and b typically denote the two halves of the tensors: one which is passed-through unchanged and the other one which is updated in a linear fashion with respect to itself, but in a highly non-linear fashion with respect to the former half, through functions s and t (which are Deep Neural Networks).

$$\begin{aligned} y_a &= x_a \\ y_b &= \left(x_b - t_{DNN}\left(x_a\right)\right)\ s_{DNN}\left(x_a\right) \end{aligned} \tag{4.1}$$

Networks s and t are in our case a two-head 3D CNN. The final activation function of head s was chosen as exp(tanh(x)) in order to easily compute the contribution to logDet (as $\sum$ tanh(x) across all spatial dimensions and channels) and provide a bound of [e^{-1}, e^{1}] to the scaling done at each coupling layer, ensuring numerical stability and a bounded global maximal value of logDet.

The input samples consist of chunks of 8 adjacent cross sections (down-sampled to 32x32 resolution) and 2 channels (the concatenation of the angiography and the binary mask volumes).

In [25] it has been shown that NF which employ affine coupling layers are prone to focus more on local pixel correlations instead of semantic content and exploit coupling layer co-adaptation in order to maximize the final log-probability. Motivated by these findings, we propose the use of a novel type of coupling layer, one which can operate efficiently for both NF directions, does not focus on

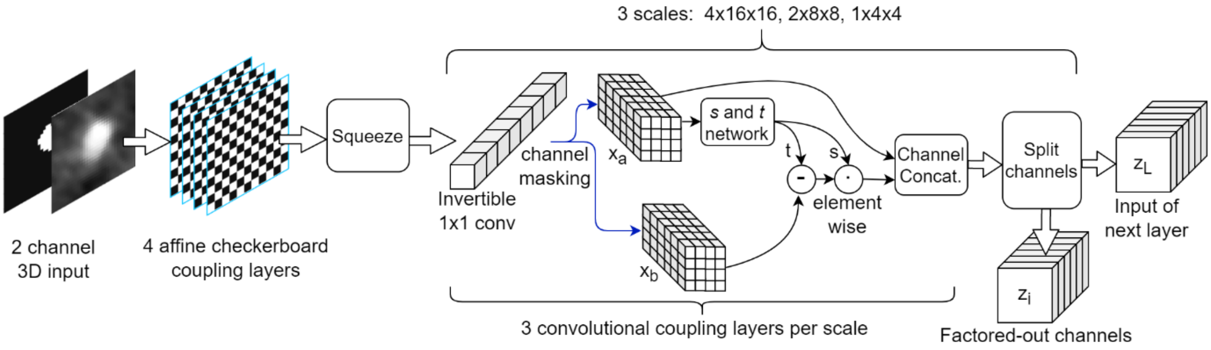

Figure 4.1: Baseline model architecture. Activation norm not depicted.

Table 4.1: Glow-style baseline architecture

Stage	No. Blocks	Block description	Resolution	No. Channels	Total number of parameters
1	4	Affine coupling layer using checkerboard mask; Activation Norm (if not last block)	8x32x32	2	~ 2 millions
2 3 4	1	3D Squeeze operation	Stage 2: 4x16x16 Stage 3: 2x8x8 Stage 4: 1x4x4	Stage 2: 16 Stage 3: 64 Stage 4: 256	
	3	Activation Norm; Invertible 1x1 Convolution; Affine coupling layer using channel-wise masking			
	1	Split channels		After stage 2: 8 After stage 3: 32	

local pixel correlations and has an inductive bias similar to conventional CNNs. The layer resembles a standard Glow-like sequence of 1x1 Invertible Convolution, channel masking, affine coupling layer. However, the last step is replaced with a 1x1 convolution (with applied bias) whose parameters are computed based on the passed-through channels, as in [27]. The applied bias is broadcasted to all spatial positions, therefore is it the same across the width, height and depth of the resulting tensor, meaning that the layer is no longer capable to reproduce masked pixel values as revealed in [25]. The same (sample specific) convolution kernel is applied at all spatial positions, in contrast to the element-wise computation done in (4.1). This behavior is similar to classical CNNs, with the exception that now the filter weights are not the same for all samples. Eq. (4.2) describes the layer's operation, with simplified notation: $*$ means 1x1 Convolution with kernel k and $+$ is a broadcasting sum. k is computed by a CNN and has shape c_{modif}-by-c_{modif}, where c_{modif} is the number of channels which are updated. b is a vector of c_{modif} elements.

$$\begin{aligned} y_a &= x_a \\ y_b &= x_a * k\left(x_a\right) + b(x_a) \end{aligned} \tag{4.2}$$

A new NF architecture was designed employing the above coupling layer. The first stage is a sequence of Additive Coupling Layers with checkerboard masking. According to [25], these layers will focus mainly on local pixel correlations, but this is equivalent to the functioning of the first layers in classical CNNs, where the receptive field-of-view is small and the filters tend to search for simple patterns such as corners, edges, textures, etc. As opposed to affine couplings, additive couplings are volume preserving, i.e. they do not contribute directly to logDet and final log $(p(x))$, but indirectly through the upstream layers.

The next stages consist in cascades of coupling layers, as described in Fig. 4.2 and Tab. 4.2. In contrast to a classical CNN, where filters of shape 3x3 (or larger) and strides larger than 1 are used (either in convolutional or max pool layers) to increase the effective field-of-view (FoV), in our architecture the FoV in these stages is increased solely by the squeeze operations. After squeezing, a 1x1x1 patch of pixels is formed from a patch of 2x2x2 pixels which were flattened spatially into the channel dimension. Therefore, the FoV doubles on each spatial axis for each squeeze step. This allows 1x1 Convolutions to operate on increasingly larger FoV, similar to the functioning of a classical CNN, while still retaining the capability of efficient forward/backward NF computation. There are enough squeeze operations so that the resolution on the last stage decays to 1x1x1. Naturally, we restrict the input spatial dimensions to be powers of 2.

In all our experiments, the network weights are initialized such that the layers are an identity mapping in the beginning of training, as suggested in [24]. We employed the PyTorch DL framework

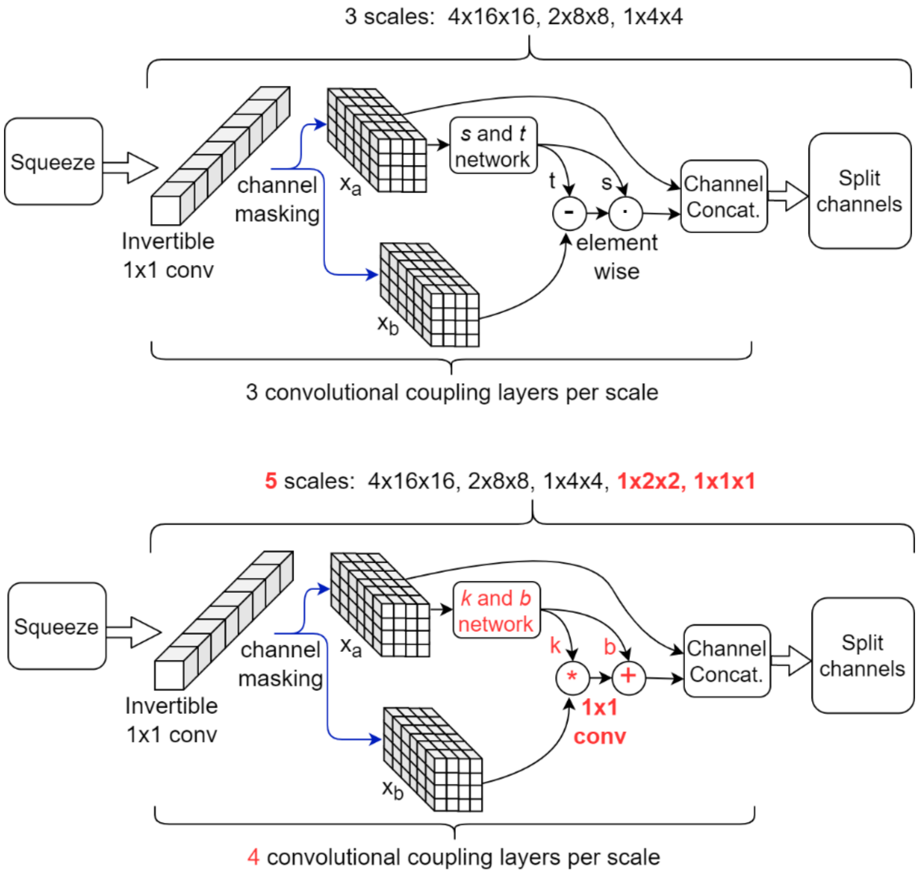

Figure 4.2: Comparison between baseline inner architecture (top) and proposed inner architecture (bottom) employing the novel coupling layer. Updated parts are highlighted in red. Normalization layers not depicted.

Table 4.2: Improved NF architecture employing the novel coupling layers.

Stage	No. Blocks	Block description	Resolution	No. Channels	Total number of parameters
1	4	Additive coupling layer using checkerboard mask; BatchNorm (if not last block)	8x32x32	2	∼8.7 millions
2 3 4 5 6	1	3D Squeeze operation	Stage 2: 4x16x16 Stage 3: 2x8x8 Stage 4: 1x4x4 Stage 5: 1x2x2 Stage 6: 1x1x1	Stage 2: 16 Stage 3: 64 Stage 4: 256 Stage 5: 512 Stage 6: 1024	
	4	BatchNorm; Invertible 1x1 Convolution; convolutional coupling layer using channel-wise masking			
	1	Split channels		After stage 2: 8 After stage 3: 32 After stage 4: 128 After stage 5: 256	

with the Adam optimizer with a learning rate of 1e-4 and trained until the validation loss plateaued.

4.2.2 Synthetic Mask Perturbations

Our application's goal is to detect incorrect pairs of (angiography image, lumen mask), i.e., samples where the segmentation is not in full agreement with the image. To test our models, we devised a method to obtain "wrong" datapoints (or samples which are not in the distribution of "correct" image-mask pairs) starting from our initial data (considered to be "correct").

We augmented the datasets by applying preset perturbations on the lumen segmentation binary mask, while keeping the angiographic image untouched. Three types of mask perturbations were employed:

- **zooming:** we applied zoom in/out operations on the mask image with respect to the mask center, so that the resulting mask is still aligned with the angiography, but larger/smaller than before.

- **morphing**: we applied dilations or erosions along 4 directions on the height*width plane: left-right, top-bottom, topLeft-bottomRight and topRight-bottomLeft. This perturbation only affects one part of the mask (the eroded or dilated part), while the other part is left untouched. By convention, negative and positive levels refer to the two ways in the selected direction, with zero meaning original mask position (levels are expressed as ratios of the original mask size along the chosen direction). At every level, either dilation (resulting in prolonged masks) or erosion (resulting in shortened masks) can be applied.

- **translations**: in the same 4 directions on the height*width plane, we translated whole mask images. Each level increment signifies a pixel shift.

For each network architecture, we performed two training procedures: one employing only original (unperturbed) data and one employing a dataset consisting of the original data and its perturbed version. The perturbations are applied during train time, similar to data augmentation techniques, such that each original data sample gets perturbed on all perturbation types, levels and directions over the training epochs. At each epoch, the ratio between untouched and perturbed data is 1-to-1.

Training only on original data and then testing on synthetic perturbations gives insight into the OoD detection capability which stems purely from the inductive bias of the NF architecture. Also, we argue that NF models, being a class of generative models, provide a form of explainability by being able to produce samples from their learnt probability density. By sampling repeatedly from the model and computing the associated log-probs, one can observe the kind of samples which the model considers to be in-distribution.

4.3 Results and Discussion

4.3.1 Evaluation on Synthetic Mask Perturbations

First, we evaluated the baseline and the proposed networks trained on original (unperturbed) data. We applied the synthetic perturbations on the testset in increasing levels of severity and measured how well the models can distinguish between log-probs of original and log-probs of perturbed samples. At each perturbation level, we computed the area under the RoC curve. We use AuRoC as a metric for assessing how well two individual data distributions can be separated by using a probability threshold.

One can observe that the proposed model has superior performance across all perturbation types and levels. Zooming under 1.0x actually yields higher log-probs for the baseline model, resulting in AuRoC values under 0.5. Even at small mask perturbation levels (e.g. 0.9x/1.1x zooming, ± 2 pixels translation), the proposed model has much larger sensitivity in detecting the mask alterations (even though it was not trained explicitly to do so) in contrast to the baseline model, where the log-probs start to decrease more significantly only at larger perturbation levels. The mask morphing is the hardest to detect since part of the mask remains the same. Thus, the baseline model is largely insensitive to this type of perturbation as the maximum AuRoC at a high perturbation level of 60% is under 0.65. In comparison, the AuRoC for the proposed network has a much faster variation for increasing perturbation severity, achieving values over 0.9 for some directions at 60% morphing.

To obtain a log-probability signal which describes the likelihood of an entire vessel segment, a sliding window approach was employed in which overlapping chunks of 8 adjacent cross-sections are fed through the NF model to obtain the log-prob values for each chunk. Using this procedure, middle cross-sections can participate in at most 8 chunks, therefore there may be up to 8 predicted log-probability values linked to each middle cross-Section. A voting scheme based on averaging is employed, where the final log-prob value for each cross-section is computed by averaging the linked predicted log-probs. Fig. 4.3 depicts such an example, where a synthetic perturbation is applied with a known severity variation. The proposed NF model detects when the perturbation severity is high enough, while outputting high log-prob values when the perturbation is negligible.

4.3.2 Evaluation on Expert Annotations

A selected test set was manually and independently annotated by three expert readers at lesion-level: each lesion was marked as being either "correctly" or "incorrectly annotated", based on the following instructions: a lesion should be marked as "incorrectly annotated" if *at least one* cross-sectional contour would require editing, otherwise if *no* cross-sectional contour requires editing, then the lesion should be marked as "correctly annotated". The annotator instructions were devised so as to match the procedure employed to construct this separate test set, with the goal of being able to directly compare the labels from the NF model to the ones provided by the annotators.

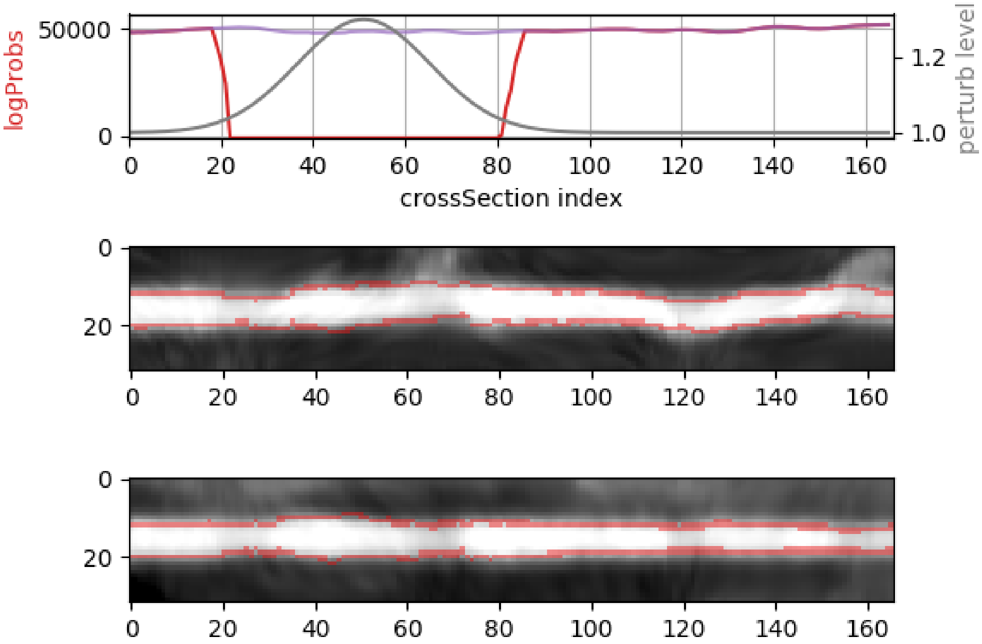

Figure 4.3: Whole vessel-segment prediction using a sliding window approach and the proposed architecture. A zooming perturbation with a known severity variation (top plot, gray signal) is applied (note that zoom level 1.0x is an identity transform). The resulting log-prob signal (top plot, red signal) dips whenever the perturbation is severe enough, compared to the original log-prob signal in the absence of any perturbation (top plot, purple signal). The bottom 2 plots display two lateral views of the vessel segment (2 projections on different axes), with the perturbed mask contour overlaid.

Table 4.3: Metrics on the secondary dataset for the baseline and the proposed model. The proposed model consistently outperforms the baseline and has metric values close to inter-expert agreement.

Metric	Inter-Expert Agreement Average [Min, Max]	Baseline Model	Proposed Model
Accuracy	0.81 [0.79, 0.86]	0.64	0.79
Sensitivity	0.79 [0.70, 0.87]	0.48	0.76
Specificity	0.83 [0.76, 0.90]	0.77	0.81
PPV	0.79 [0.70, 0.87]	0.63	0.76
NPV	0.83 [0.76, 0.90]	0.65	0.81

Evaluating the two NF models against the human annotations was framed as a binary classification problem. Tab. 4.3 summarizes relevant metrics (accuracy, sensitivity, specificity, PPV and NPV) for the proposed and baseline models. Annotation consensus was obtained through a majority vote between the three annotators. The mean inter-user metric values were obtained by averaging all 6 possible metric values pertaining to pairs of annotators, e.g., Annotator_1 (as GT) versus Annotator_3 (as Prediction), Annotator_3 (as GT) versus Annotator_1 (as Prediction), etc. When compared against annotation consensus, the proposed model has higher performance than the baseline on all considered metrics.

We observe that the proposed model has sensitivity of 76.0%, close to the inter-user value of 79.0%, while the baseline model only achieves 48%. The overall accuracy score also increases to 78.6% (close to the inter-user value of 80.9%) for the proposed model, as compared to an accuracy value of 64.3% for the baseline Glow-like model. These results reinforce the observation that the baseline model is unable to fully capture semantic content while the proposed model does. Similar behavior was observed in the previous section, where the proposed model had better AuRoC values in detecting synthetic perturbations when trained only on original data.

4.3.3 Sampling from the Models

We employed the models trained on the augmented trainset to generate novel samples. Similar to sampling procedures in [24], we employed $\mathcal{N}(\mathbf{0},\ 0.6 \cdot I)$ instead of the actual prior distribution (i.e. standard normal multivariate distribution) in order to produce samples with larger log-probs and which look more realistic. Each new sample was run back through the model in the forward direction to compute the log-probs, confirming that the sample is in fact in-distribution (the sampling procedure may seldomly generate samples of lower log-probability). Fig. 4.4 shows samples from the two models.

As already revealed in [25], the baseline model tends to focus more on textures and is unable to capture the semantics of the training data. We observed that in most of the generated samples, the segmentation mask is lacking (i.e. only zeros are generated on the mask channel). Also, the usual round shape of the lumen is not distinguishable in the image channel. In contrast, the proposed architecture manages to capture the semantic content of a usual data point: the lumen has the typical shape in the image channel, the segmentation mask is present (with plausible pixel-values, e.g., close to either 0 or 1) and respects the shape and position of the lumen in the image channel.

We argue that inspecting the generated samples is an explainability mechanism which offers insight into the learnt probability distribution, i.e. the model can provide example inputs which are very likely under the learnt density and by repeating the sampling procedure enough times, an approximation of the typical set of the learnt distribution may be constructed. If a generative model consistently produces samples with high associated log-probabilities but which have low quality under manual inspection and are implausible considering the specific topic/domain, then this is proof that the learnt probability density is not a good approximation of the true probability density and,

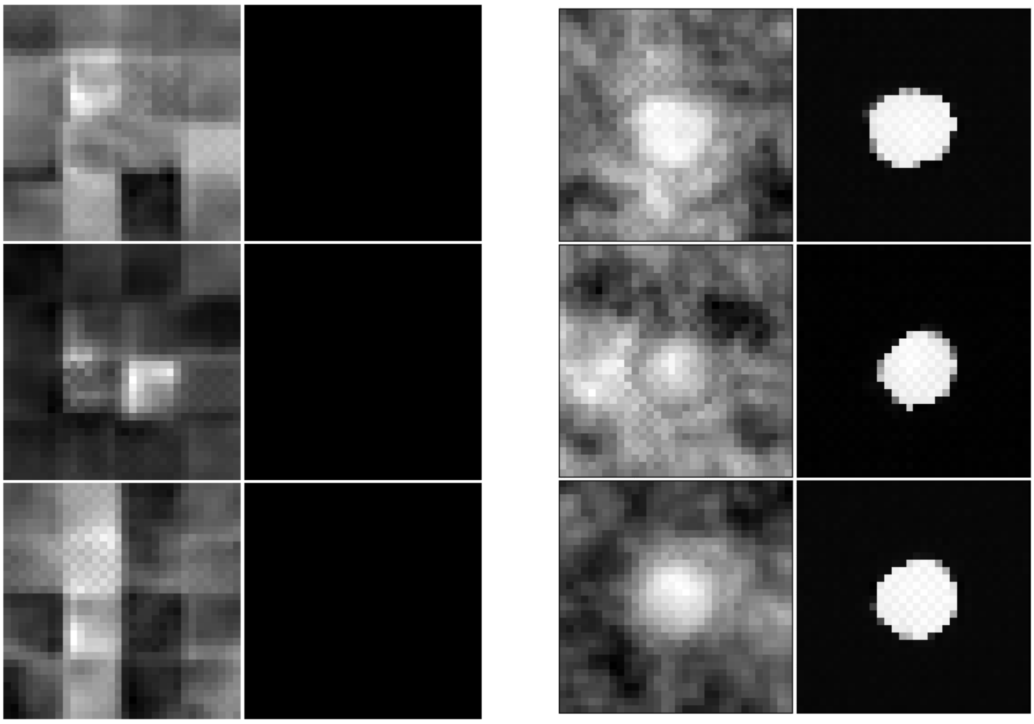

Figure 4.4: Three samples (pairs of lumen image and segmentation mask) generated by the proposed network (right) and by the baseline model (left).

therefore, the model cannot be reliably used for OoD detection.

4.4 Conclusions

While the performance of AI based methods has improved markedly over the years, semi-automated approaches are currently still being employed. One potential approach for significantly reducing the processing time is to pre-select regions of interest which are likely to require manual inspection and editing. Herein we have linked this pre-selection step to the topic of confidence and out-of-distribution detection, based on NF. The usage of a novel coupling layer which exhibits an inductive bias favoring the exploitation of semantical features instead of local pixel correlations was investigated on the task of detecting mismatched pairs of CCTA lumen images and their corresponding lumen segmentations. A network architecture employing such layers was tested against a Glow-like baseline. The proposed network showed better performance in OoD detection when tested against synthetic perturbations, while the sensitivity of detecting faulty annotations was close to inter-expert agreement. Samples from the model confirm that the learnt probability density managed to capture the relevant informational content from the training samples, instead of just modelling plain textures.

The method proposed herein allows for more confident decision making using CCTA imaging alone. Using the proposed out-of-domain detection method, the gray zone in the clinical interpretation can potentially be narrowed down.

5. Cardiac Phase Detection on Invasive Coronary Angiographies[1]

5.1 Introduction

Invasive coronary angiography (ICA) represents the gold standard in CAD imaging [31], providing important information about the structure and function of the heart (annually, more than a million ICA procedures are performed in the USA alone [32]). It enables the assessment of the anatomical severity of coronary stenoses either visually or by computer-assisted quantitative coronary angiography (QCA) [33]. In view of the limitations of the pure anatomical evaluation of CAD, and given the recent technological advances, methods for image-based functional assessment of coronary stenoses based on ICA have been introduced and validated, e.g. image based Fractional Flow Reserve (FFR) computation [34, 35, 36]. In this and other clinical settings based on the use of ICA, detection of the end-diastolic frames (EDF) and, in general, cardiac phase detection on each temporal frame of a coronary angiography acquisition is of significant importance.

Currently, the selection of the EDF and the identification of the cardiac phase are performed either manually or automatically based on simultaneously acquired ECG signals [35]. This has a number of drawbacks: ECG signals may not always be available, and cardiac phase detection based on ECG can be challenging if the signal-to-noise ratio is too low to accurately detect end-diastole or the signal presents artefacts [37], [38].

The main challenge addressed within this work is the development of a purely image-based method for performing cardiac phase detection at frame level and EDF detection on invasive coronary angiographies.

Furthermore, we addressed two additional challenges related to the main one. The first one was to demonstrate that the methodology introduced herein provides similarly good results across all types of coronary angiographies, with variations in view (left / right coronary artery), primary and secondary acquisition angles, heart rate, and patient type and condition (stable / acute, previous Percutaneous Coronary Intervention (PCI), previous Coronary artery Bypass Graft (CABG), coronary total occlusions, etc.). The second additional challenge was to leverage the very large dataset (77435 coronary angiographies images acquired from 13081 patients) without performing manual annotations. Exploiting such a large dataset is beneficial due to the extreme variability in X-ray coronary angiography but makes the use of ground truth information defined from manual annotations unfeasible.

Thus, we introduce herein a methodology based on deep neural networks (DNN) for cardiac phase (systole or diastole) and EDF detection on X-ray coronary angiographic images. Ground truth labels, employed for training the model, are derived from simultaneously acquired ECG signals.

[1] The following chapter describes work published in:

- **Ciușdel, C.**, et al., 2020. Deep Neural Networks for ECG-free Cardiac Phase and End-Diastolic Frame Detection on Coronary Angiographies. Comput. Med. Imaging Graph. 84, 101749, `https://doi.org/10.1016/j.compmedimag.2020.101749`

Some sections were quoted verbatim from the above reference, which represents previously published work of the author, under the PhD research program.

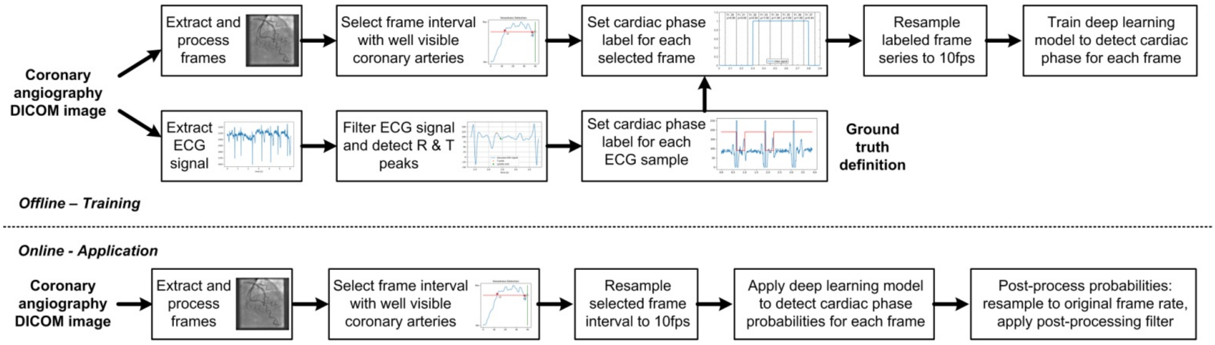
Coronary angiography DICOM image
Extract and process frames
Select frame interval with well visible coronary arteries
Set cardiac phase label for each selected frame
Resample labeled frame series to 10fps
Train deep learning model to detect cardiac phase for each frame
Extract ECG signal
Filter ECG signal and detect R & T peaks
Set cardiac phase label for each ECG sample
Ground truth definition
Offline – Training
Online - Application
Coronary angiography DICOM image
Extract and process frames
Select frame interval with well visible coronary arteries
Resample selected frame interval to 10fps
Apply deep learning model to detect cardiac phase probabilities for each frame
Post-process probabilities: resample to original frame rate, apply post-processing filter

Figure 5.1: Overall workflow of the proposed methodology for offline training and online application.

5.2 Methods

Coronary arteries display significant motion on angiographic images during one heart cycle. The motion is the compound effect of cardiac contraction, respiratory motion, and possibly patient or table displacement, usually appearing as panning in angiographic images. The main goal of the proposed workflow is to employ deep learning based techniques to determine the cardiac phase of each angiographic frame by implicitly analyzing the motion of arteries and structures visible on consecutive angiographic frames, and without using any associated ECG information.

Fig. 5.1 displays the overall workflow of the proposed methodology for cardiac phase detection, including offline training and online application. In brief, for the online application, a first DNN, trained to detect coronary arteries, is employed to preselect a subset of frames in which coronary arteries are well visible. A second DNN predicts cardiac phase labels for each frame: it takes as input sequences of 10 frames from the preselected frame interval, performs spatial and temporal convolutions, and outputs predictions for the middle 4 frames of the sequence. The second DNN is applied with a sliding window mode approach. Only in the training and evaluation phases for the second DNN, ECG signals recorded simultaneously with angiographic images are used to provide ground truth labels: R-wave and T-wave peaks, employed for determining the onset and the end of systole respectively, are algorithmically detected, and the corresponding cardiac phase labels are mapped to each angiographic frame. The preliminary step of processing angiographic sequences to extract frames in which coronary vessels are clearly visible is also applied when building the training database for the deep learning method determining the cardiac phase of each angiographic frame. While the ECG signal is typically available for the entire angiographic acquisition, we have chosen to detect the cardiac phase only for those frames on which the coronary arteries are well visible, because these are the frames which are relevant in the clinical decision making process. Moreover, by allowing the deep learning model for cardiac phase detection to learn only based on frames with well visible arteries, presumably, a more specialized and, thus, better performing model is obtained.

The main methodological contribution in this work is the approach for offline training and online application (see Fig. 5.1):

- Coronary angiographic frame intervals relevant from a clinical point of view are automatically selected. This is has a positive impact both on the training process (by allowing the cardiac phase prediction model to learn only on relevant frames, with well visible arteries, its prediction accuracy increases for the relevant frames), and for the online application (e.g. only relevant EDF frames will be selected)

- No intermediate steps or tasks performing the detection, localization or segmentation of anatomical structures are included in the workflow. Moreover, no surrogate measures must be defined. The models take directly the images as input and output the detected vessels / cardiac phase

- Ground truth labels are automatically derived from ECG signals

5.2.1 Offline Training Process

5.2.1.1 Pre-processing of ECG Signals and Angiographic Frames

Anonymized DICOM-formatted coronary angiographic acquisitions are used as input data. In the pre-processing stage, each DICOM file is parsed to extract the imaging data and the raw ECG signal used for ground truth definition (Fig. 5.2 left). Diastole-systole and systole-diastole transitions are detected to define the ground truth cardiac phase labels. For the former, BioSPPy, an open-source toolbox for biological signal processing, is employed [39]. First, BioSPPy applies a default processing pipeline to the original ECG signal, based on a dual-pass zero phase delay band pass linear finite impulse response filter. Next, the R peaks are detected as surrogate for diastole-systole transition (Fig. 5.2 middle). The prerequisite for identifying the systole-diastole transition is the detection of the T peak. High frequency oscillations are typically still present after the first filtering step, hindering a precise T peak detection. Therefore, another stage of zero phase delay low pass filtering is applied. Finally, the T peaks are identified according to the following rules (Fig. 5.2 right):

- The T peak time point should reside between two predefined limits, expressed relative to the heartbeat duration: herein, the limits of 20% and 65% were selected [40]
- The T peak should be a local maximum or minimum of the filtered ECG signal
- The T peak should be the local maximum or minimum with the largest temporal span, with respect to neighbouring maxima locations, in the considered window

Finally, the time point corresponding to the systole-diastole transition is defined as the first local minimum or maximum after the T peak, or as the time point where the filtered signal decreases or increases below the 20%-65% window mean value, whichever time point is encountered first [40]. A binary classification signal is then generated based on the detected time points: '0' for systole and '1' for diastole.

The goal of the herein proposed method is to correctly predict the cardiac phase labels and detect EDFs for the angiographic frames on which coronary arteries are well visible, i.e. filled with contrast agent, as this corresponds to the set of clinically relevant frames for the applications that we are considering. For each frame the visibility of the coronary arteries is determined using a deep learning classifier trained to detect pixels which represent coronary arteries, i.e. to estimate the 'vesselness' of the image. All frames with visibility higher than a given threshold, defined relatively to the maximal number of pixels per frame representing coronary arteries, are marked as candidates; the largest continuous sequence of candidate frames is selected for further processing.

Furthermore, to develop a cardiac phase detection model which is independent from the acquisition frame rate, the labeled frame series is resampled at 10 fps.

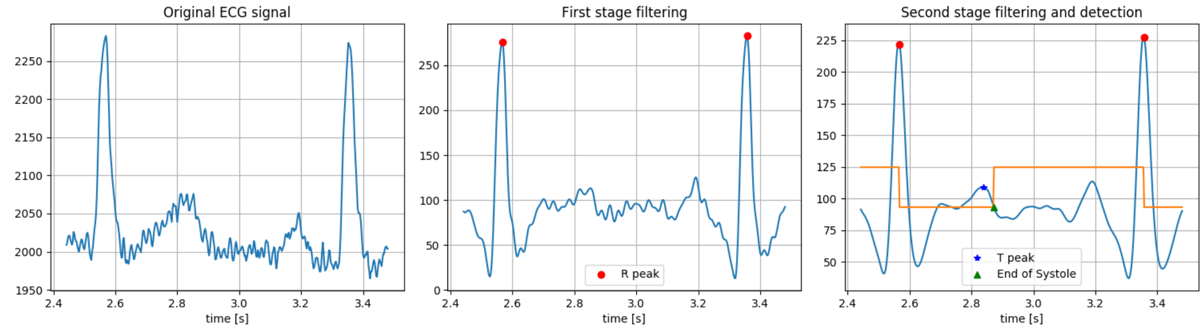

Figure 5.2: Processing of ECG signals: original ECG signal, first stage filtering, i.e. dual-pass zero phase delay band pass linear finite impulse response filter and R peak detection, second stage filtering, i.e. zero phase delay low pass filtering, and T peak and end of systole detection.

5.2.1.2 Vesselness Deep Learning Model Architecture and Training Process

A deep learning based model is employed to determine the angiographic frame interval on which coronary arteries are well visible, i.e. filled with contrast agent. The classifier is a deep neural network having an architecture similar to U-Net [41], trained to classify pixels using the Adam optimizer [42] by minimizing a custom loss function derived from the Jaccard index:

$$L = 1 - \frac{\mu + \sum P_i T_i}{\mu + \sum P_i^2 + \sum T_i^2 - \sum P_i T_i} ,$$

where P_i and T_i are the predicted and the expected probability that the i-th pixel is part of a coronary artery, and $\mu = 0.1$ is a smoothing factor.

Training data are angiographic frames in which coronary vessels are manually annotated. Each artery is annotated as a set of centreline points, and to each point an approximate estimate of the local vessel radius is associated. Using these annotations, a binary mask is generated for each angiographic frame, in which pixels found in the neighbourhood of centreline points are set to one; all other pixels are set to zero.

During inference, the model processes independently each angiographic frame and outputs a probability map with the same size as the input image, where the value of each pixel represents the probability that the associated pixel from the input image is part of a coronary artery. By summing all pixel-wise probabilities, an overall vesselness score of the frame is determined, i.e. the likelihood of the frame to display coronary vessels.

For each angiographic frame in the series, the vesselness score is determined. Frames having a vesselness score lower than 0.4 of the maximum vesselness score in the series are discarded. From the remaining frames, the longest subsequence of consecutive frames is selected for further processing.

5.2.1.3 Cardiac Phase Deep Learning Model Architecture and Training Process

The cardiac phase predictor is a deep neural network taking as input a sequence of angiographic frames (Fig. 5.3). The output of the network is the classification of the middle frames of the input sequence, in terms of the probability of each frame being a diastolic or systolic frame. Since we consider a binary classification problem, with the diastolic and systolic classes being associated with class index 1 and 0 respectively, the output probability can be interpreted as the predicted class index. We designed the network to process a sequence of 10 frames, covering exactly one second of acquisition. This choice was made as a trade-off between accuracy of detection and memory usage during the training phase. The network outputs the classification only for the middle 4 frames of the sequence, so that for each frame the classification task uses information both from prior as well as following frames.

As shown in Fig. 5.3, a first convolutional neural network (CNN) is employed to perform spatial convolutions, mapping each 512 x 512 input image to a 64 dimensional feature vector. The spatial CNN employs a classic structure of 2D convolutional layers followed by max pooling layers, and, finally, by a fully connected layer. The same CNN is applied independently to each frame, yielding a 10 x 64 feature matrix. A second CNN performs temporal convolutions on this matrix, independently for all 64 image features. Its output is a 64 x 8 matrix, which contains 8 temporal features for each one of the 64 spatial features. This matrix is then fed to a dense classifier, which computes the output probabilities for the middle 4 frames in the input sequence (frames 4-7).

After performing the selection of relevant frames using the vesselness model, and the resampling at 10 fps, the training datasets are generated as sequences of 10 frames. Ground truth labels for the middle 4 frames of each sequence are obtained by processing the associated ECG signal as described in the previous section.

Training is performed using the Adam [42] optimizer and a custom loss function based on the Poisson distribution divergence [43] which penalizes the output probability less significantly if the class is predicted correctly with respect to the 0.5 threshold value.

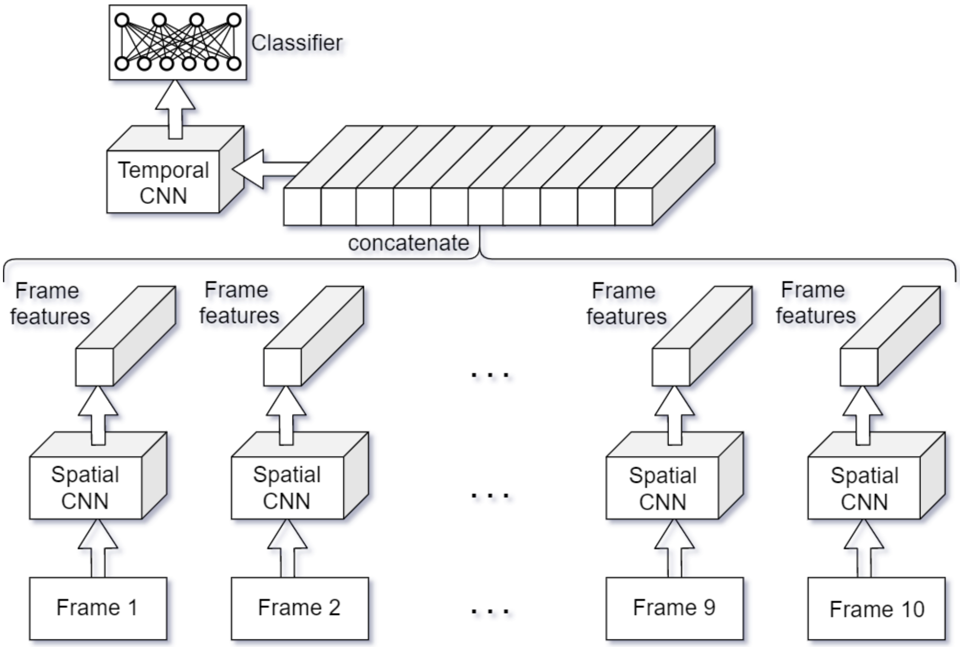

Figure 5.3: Overview of the deep neural network employed for cardiac phase detection.

5.2.2 Online Application Process

During the online application of the trained deep learning model for cardiac phase prediction, no ECG information is used. The angiographic frames are pre-processed with the same methods used to populate the training database: they are cropped and rescaled, and the DL based vesselness estimator is employed to select the relevant frame interval. Next, the frames within the chosen interval are resampled at 10 fps and provided as input to the cardiac phase predictor.

To minimize the amount of redundant operations, first the spatial CNN is applied to all input frames, and the computed frame features are stored for further use. Next, the temporal CNN and the output classifier are applied with a sliding window approach, with step size one, using the stored frame features as input. Since the model outputs probabilities for the middle four frames, several frames end up with multiple predictions, and we consider for the final cardiac phase predictions only the frames for which at least two probabilities were generated. The final prediction is selected as the probability value closest to one of the class indices. In an additional post-processing step, the resulting sequence of probability values is resampled to the original frame rate using linear interpolation. The output is a binary classification signal, where zero corresponds to systole and one to diastole.

5.3 Results

5.3.1 Vesselness Detection

The DNN performing vesselness detection was trained on 1174 coronary angiographies and validated on 293 coronary angiographies. Annotations were performed for five consecutive frames for each angiography, resulting in 5870 training frames and 1465 validation frames. Additionally, data augmentation was performed on the training set, consisting in random scaling with a factor between 0.8 and 1.2, gaussian intensity shifts with mean 0 and standard deviation 0.1, and gaussian noise with mean 0 and standard deviation 0.2. On a separate testing set consisting of frames extracted from 91 coronary angiographies (i.e. 455 frames in total), the model achieved an average Dice score of 0.86. The training, validation and test sets contain acquisitions depicting both RCA and LCA, from different acquisition angles.

5.3.2 Cardiac Phase Detection

The workflow for cardiac phase detection was trained and evaluated using independent datasets. The training dataset consisted of 56655 coronary angiographies acquired from 6820 patients. A 90 – 10 % split was applied at patient level during training for setting up the actual training and validation datasets.

The main evaluation dataset consisted of 20780 coronary angiographies from 6261 patients and was acquired at a different clinical site. This completely independent dataset was used for quantitatively assessing the performance of the cardiac phase and end-diastolic frame detection workflow, and for performing various subgroup analyses based on acquisition angles, heart rate, and vessel of interest. Furthermore, the ECG signals of a subset of coronary angiographies, extracted from the main evaluation dataset, were annotated by experienced clinical examiners, and the cardiac phase detection performance was assessed separately on this subset.

No exclusion criteria were formulated in the training and the main evaluation dataset related to acquisition angles, heart rate, contrast agent intensity, patient characteristics, and patient diagnosis.

5.3.2.1 Cardiac Phase Detection Performance on the Main Evaluation Dataset

For the quantitative evaluation of the cardiac phase detection workflow performance, we excluded frames with intermediate ground truth label values, and report in the following results for the remaining frames, and for the case when additionally the frames adjacent to a cardiac phase transition are excluded.

Tab. 5.1 displays for these two configurations the statistical measures for cardiac phase detection on the evaluation dataset, under two settings: (a) all coronary angiographies have the same weight, i.e. statistics are first computed independently for each coronary angiography, and then an average value is computed for all angiographies, and (b) all frames have the same weight.

To assess the performance of the image based detection independently for the two cardiac phase transitions, systole-to-diastole and diastole-to-systole, we determined the cardiac phase misclassifications at frame level around each type of transition for the coronary angiographies with annotated ECG signals. We found that 65.0% / 65.2% of the errors corresponded to the systole diastole transition, and 35.0% / 34.8% of the errors corresponded to the diastole systole transition.

5.3.2.1.1 Cardiac Phase Detection Performance vs. Vessel of Interest Coronary angiographies are recorded either for the left coronary artery (LCA) or the right coronary artery (RCA). Using a DL based classifier (98.0% accurate) we determined the coronary artery visible in each angiography, and computed the mean accuracy for each class: 98.9% for LCA, i.e. 73.7% of all coronary angiographies, and 98.5% for RCA, i.e. 26.3% of all coronary angiographies. A similar LCA / RCA distribution could be observed in the training dataset.

Table 5.1: Statistical measures on the evaluation dataset, computed in two variants: (a) all coronary angiographies have the same weight, and (b) all frames have the same weight.

Statistical measure	Including frames adjacent to transitions		Excluding frames adjacent to transitions	
	Identical weights at coronary angiography level	Identical weights at frame level	Identical weights at coronary angiography level	Identical weights at frame level
Accuracy	96.4%	96.6%	98.8%	98.8%
Sensitivity	97.1%	97.3%	99.3%	99.4%
Specificity	95.2%	95.3%	97.6%	97.5%
PPV	97.0%	97.1%	98.9%	98.9%
NPV	95.5%	95.7%	98.6%	98.7%

5.3.2.1.2 Cardiac Phase Detection Performance vs. Expert Annotations To further validate our approach, we considered a subset of 300 patients randomly chosen from the main evaluation dataset. For each of the patients we randomly picked one angiographic acquisition which fulfilled the inclusion criteria, and two experienced clinical examiners independently annotated all R and T peaks on the corresponding ECG signals. Since T peaks could not always be reliably detected on the ECG signals extracted from the angiographic images, we finally only retained those angiographic acquisitions on which R and T peaks were identified by both clinical examiners, leading to an evaluation dataset consisting of 207 coronary angiographies. Starting from the R and T peak annotations we defined the ground truth cardiac phase labels as described in the methods section. Tab. 5.2 displays the statistical measures for cardiac phase detection on this evaluation dataset.

5.3.2.1.3 End-diastolic Frame Detection Performance on the Main Evaluation Dataset One of the goals for performing cardiac phase detection on coronary angiographies is the detection of EDFs suitable for 3D QCA and coronary artery segmentations. Tab. 5.3 displays the statistical measures for EDF detection on the main evaluation dataset, obtained with a tolerance range of ±1 frames around the ground-truth EDF as detected based on the ECG signal. For the image based predictions, each diastolic frame which is followed by a systolic frame is considered to be an EDF. For the ground truth signal, each diastolic frame which is followed by a transition frame or a systolic frame is considered to be an EDF.

5.4 Discussion

Cardiac phase and end-diastolic frame detection are essential steps for the quantitative processing of coronary angiographies. Currently, the selection of the end-diastolic frame is performed either manually or automatically based on the simultaneously acquired ECG signal [35]. The ECG signal may not always be available, and the ECG-based cardiac phase detection has several drawbacks: the signal-to-noise ratio may be too low to accurately detect end-diastole or the signal may present artefacts [37, 38]. Herein we report, to the best of our knowledge, the first deep learning based workflow for purely image-based cardiac phase classification of angiographic frames validated on a large, real-world dataset.

Training data also included the natural variation in angiographic acquisitions of each view, including variations in image quality. We specifically avoided idealization of training and evaluation

Table 5.2: Statistical measures on the evaluation dataset consisting of coronary angiographies for which the ECG signals were annotated by two experienced clinical examiners.

Statistical measure	GT defined by clinical examiner 1	GT defined by clinical examiner 2
Accuracy	97.6%	97.6%
Sensitivity	97.2%	97.3%
Specificity	98.3%	98.3%
PPV	99.3%	99.3%
NPV	93.6%	93.5%

Table 5.3: Statistical measures for EDF detection, with a tolerance range of ±1 frames around the ground-truth EDFs as detected based on the ECG signal.

Statistical measure	Performance
Precision	98.4%
Recall	97.9%
F1 score	98.2%

datasets, to ensure that the model is applicable in daily clinical practice with the accuracy reported herein. Both training and evaluation datasets were very large, consisting of a total of 77435 coronary angiographies acquired from 13081 patients. This ensures both model generalization and accurate prediction statistics. Moreover, the training / validation and evaluation datasets were acquired at different clinical sites. While we have used exclusively CNNs for the cardiac phase detection, similar results may be obtained with recurrent neural networks (RNNs) for this type of task.

5.5 Conclusions

Given the very large datasets employed during training and evaluation, we adopted a strategy, where the cardiac phase labels are determined automatically from the simultaneously acquired ECG signals (only those acquisitions were selected for which the ECG signal was reliable). This strategy allowed us to reduce drastically the annotation costs (in terms of both time and expenditures).

We conclude that the proposed image-based workflow, employing deep neural networks, demonstrated good performance, thus potentially obviating the need for manual frame selection and ECG acquisition, representing a relevant step towards automated CAD assessment.

6. Final Conclusions

6.1 Conclusions

The focus of the thesis was to develop, apply and evaluate Deep Learning based methods on large datasets comprised of medical imaging acquisitions, to demonstrate the utility of such methods in assessing and diagnosing patients with cardiovascular disease. The DL models were augmented with architecture customizations while the training protocols were personalized to address the specifics of the tasks, in face of the heterogeneous parameters such as amount and type of data, prediction complexity and runtime requirements.

Semantic segmentation is a fundamental task, especially in the medical domain. Recent research proposed novel DNN architectures (e.g. UNet) which achieved state-of-the-art segmentation and detection performance on medical data. In this work, accuracy was incrementally improved by using such novel architectures, and by researching a series of custom modifications, such as joining segmentation and landmark detection under a multitask learning problem, using novel loss functions (i.e. Adaloss), and jointly modeling pairs of ED and ES frames in echocardiographies to improve the final performance of a DL-powered pipeline for estimating Ejection Fraction.

The trend in medical imaging is to include AI-algorithms inside the imaging equipment, to support diagnosis and to automate repetitive tasks, leading to time and resource savings. Therefore, the predictive performance of such algorithms have a direct impact on the usability of entire pipelines. A valuable measure besides the actual prediction performance is the estimate of uncertainty. Simply treating the DNN models as black-boxes affects the explainability and the trustworthiness of an automated medical analysis pipeline. When large uncertainty is associated with model predictions, the input can be flagged and presented to expert readers which take a decision on the next pipeline steps or validate/correct the model prediction. Uncertainty estimation is a hot research topic. Several such methods were investigated in the context of semantic segmentation, with interesting results. Regions of the input image which have inadequate quality can be marked and the associated prediction variants can be proposed for finetuning the output to medical practitioners.

When large-scale annotations are not available, raw imaging pixels can still be employed to pretrain large DNN models. Heuristic pretext tasks were investigated for self-supervised learning on echocardiographies. Although being a simple framework with high potential, special care must be taken with respect to data artifacts having high correlation to the pretext task. DNN behavior in a self-supervised regime was investigated when synthetic artifacts were injected in the training set, revealing that such simple heuristic pretext tasks offer models the possibility of cheating, instead of encouraging them to infer actual relevant data features. More robust approaches have been described, such as contrastive learning techniques.

When working on 3D input volumes, convolutional networks tend to be resource intensive, both in terms of the required amount of memory and in the number of FLOPs. For a landmark localization topic in 3D echocardiographies, deep Reinforcement Learning techniques were investigated and a suite of multiscale RL agents were developed to replace the equivalent large CNN needed for such a task. Instead of operating on the entire input domain, a reward-maximizing trajectory is learned by each agent. This avoids unnecessary computations on irrelevant regions of the input echo volume. As each agent is powered by a relatively-small DNN model, fast landmark search speeds were attained, despite the sequential nature of the RL algorithm.

Unsupervised learning is a large sub-domain in the Deep-Learning world. Generative modeling involves learning, implicitly or explicitly, the underlying data distribution inside the train set. Normalizing Flows models have been researched in this work to explicitly and efficiently model the probability density of pairs of CT lumen slices and segmentation masks. Previous research highlighted some flaws in the operation of deep NF models, limiting them in capturing and focusing on semantic content. A revised architecture was proposed and investigated which displayed much improved performance in outlier detection when compared to a standard baseline. Element wise affine transformations were replaced with kernel-based ones, similar to the inductive bias in regular convolutional networks. This structural change had a positive impact on the semantic capabilities of the proposed NF model. Another option is to model implicitly the underlying data probability distribution, which is what Generative Adversarial Networks do. Such models can be further conditioned on external signals which dictate the way new samples are synthesized. A spatially adaptive (de)normalization layer was investigated for building a model which outputs realistic apical echocardiographies conditioned on user-provided chamber segmentation masks. The architecture was further extended to jointly generate pairs of ED and ES echo frames, as if they pertained to the same medical acquisition. Such powerful generative models allow for the generation of synthetic datasets with custom prescribed properties, such as EF values and chamber size, placement, and / or orientation.

For the quantitative processing of coronary angiographies, cardiac phase and end-diastolic frame detection are considered essential steps. Based on a large dataset, a purely image-based cardiac phase detection algorithm was developed which showed high performance across a wide variety of acquisition angles, heart rates and views. For the more heterogeneous settings in echocardiographies, an extension employing recurrent neural layers was researched, which also achieved good performance and robustness across several views. These image-based algorithms alleviate known problems attached to ECG-based cardiac phase detection.

In conclusion, several deep learning approaches were proposed and investigated for solving existing problems and topics in the field of medical imaging based diagnosis for patients with cardiovascular disease. Extensions were researched and tested, yielding novel and personalized solutions to long standing DL tasks.

6.2 Original Contributions

The personal contributions can be grouped based on the medical imaging modality. Each proposed solution is a step towards a common goal in healthcare applications: reliable and trustworthy automation of medical diagnosis pipelines, leading to improved patient care and quality of service.

6.2.1 Echocardiography

Ultrasound (US) is a widely used imaging technique due its noninvasive, real-time and low-cost nature. Therefore, AI methods developed for US may reach a large client base and their performance may potentially have a large impact.

Developing DNN models usually requires relatively large labeled data. Pretraining methods are known to yield a performance boost when the supervised finetuning on the target labeled dataset starts from pre-optimized parameter vectors; such parameters were obtained by forcing the model to learn expressive representations of unlabeled data which are either invariant under certain augmentations or can instead distinguish between them. An original contribution is the investigation of heuristic pretext tasks for obtaining pretrained models on echocardiographies. Horizontal image flipping and systolic frame ordering were proposed as two pretext task suitable for apical BMode echocardiographies, as they require the model to infer features about chamber placement, mitral valve opening, and chamber area. Such general features are exploited in downstream supervised tasks: since a model no longer needs to learn these general features from scratch, it requires a smaller train set to achieve comparable performance to the case where no pretraining is used.

Another method to improve model performance in the face of limited data (of a certain type) is to synthetically generate it. Synthetic samples from under-represented categories can be conditionally generated to match user-provided semantics and properties. Such a framework was adapted to run on apical 2D echocardiographies. An original contribution in the form of an architectural extension allowed a GAN to sample ED/ES frame pairs instead of individual independent frames, as if they pertained to the same acquisition and heart cycle. Chamber segmentation masks acted as conditioning signals and the generated samples respected the imposed layout. This method allows the generation of synthetic ED/ES pairs having user-controlled EF, which can be of great use when trying to build classifiers on unbalanced train sets, by augmenting poorly represented bins.

Cardiac phase detection is a crucial step in auto-EF solutions. An original contribution consisted in developing a RNN-based architecture capable of handling the heterogeneous nature of echocardiographies. The model exhibited good performance on sequences of various frame rates, lengths and views.

A core task inside cardiac assessment pipelines is auto-contouring of heart chambers, to estimate quantities such as volumes, global longitudinal strains, etc. Another original contribution was to investigate and augment various segmentation and landmark detection architectures. Joining the two prediction types under a multitask loss and using skip connections provided a performance boost. A proposed architectural extension for simultaneously predicting on ED/ES frames with mutual conditioning, improved model robustness in the face of variable acquisition quality.

It has been previously reported in literature that regular DNN models tend to be overconfident in their prediction even when they are wrong. Such a behavior is not beneficial, especially in automated medical imaging pipelines. Another original contribution was to adapt recently developed uncertainty frameworks for the task of semantic segmentation. E.g., when using Gaussian Processes (GP), a classical DNN decoder head was used to compute a region of interest on which to apply the GP decoder head. This region was much smaller than the entirety of the input image and hence it enabled a reduction of several orders of magnitude in required computational resources. An energy based method was also adapted for estimating the contour uncertainty around the LV. Analyses have shown that there is a moderate correlation between the uncertainty estimates from different methods, even when using heuristic metrics for the global uncertainty estimation.

Finally, 3D landmark detection on echo volumes using multiscale RL agents was investigated. An original contribution consisted in finetuning the training procedure, its hyperparameters and deployment strategy to obtain robust and fast agents. The results indicate a good localization performance, with minimal runtime and computation requirements.

6.2.2 Coronary Computed Tomography Angiography

Certain automated imaging pipelines may have an optional stage where the medical practitioner may input annotations. Other pipelines may employ cascades of DNN models. One can note that a faulty annotation / prediction can propagate throughout the pipeline, and directly affect the quality of the final output. An audit model can help in ensuring that the input data, at a certain stage, is valid. This validity can be formulated as an out-of-distribution detection problem, as it may have better generalization capabilities than, e.g., employing a regular classifier scanning for a limited set of possible defects.

Normalizing flows are good candidates since they enable a fast and efficient computation of probability densities. However, it has been reported in literature that standard architectures relying on affine coupling layers tend to focus more on textures instead of the semantic content, leading to poor OoD detection performance. An original contribution was the proposal of a novel NF architecture employing convolutional coupling layers for detecting faulty pairs of CT images of coronary angiographies and corresponding lumen segmentations. A pair was considered faulty if the mask is not fully aligned with the lumen image. Mask perturbations (e.g. zooming, dilations, translations, etc.) were proposed to generate artificial outliers on-the-fly. When comparing against a baseline, analyses reveal that the proposed model has better semantic interpretation capabilities, and, hence, superior detection performance. Sampling experiments revealed that the baseline model indeed focuses on texture, and is unable to produce realistic samples, while the proposed model manages to generate synthetic samples which look both realistic, and are semantically coherent between channels (i.e. the lumen mask matches the CT image).

6.2.3 Coronary Angiography

Cardiac phase detection represents an important pre-processing task invasive coronary angiographies. Instead of using the ECG signal (which may be noisy or missing), a purely image-based solution is proposed. Video acquisitions are preprocessed and resampled at a constant frame rate and fed to a convolutional neural network. In a self-supervised manner, the ECG is processed to extract a binary signal describing the two heart phases; the ECG is used only during training to offer model supervision. Although processing the ECG is not a fully robust procedure, the train set is very large and thus averages out any local errors in the ECG-based phase extraction. The model is applied in a sliding window fashion using a majority voting mechanism, and can therefore be applied on sequences of arbitrary length. Analyses reveal good and robust model prediction performance, for large ranges of viewing angles, and across multiple views. A large evaluation set (collected from a site different than the one from which the training data was collected), was employed to test the robustness of the cardiac phase solution under multiple scenarios.

6.3 Dissemination of Research Results

During the PhD program, the conducted research led to 4 publications as author or co-author. Two journal articles were published as first author:

- **Ciusdel, C.**, et al., 2020. Deep Neural Networks for ECG-free Cardiac Phase and End-Diastolic Frame Detection on Coronary Angiographies. Comput. Med. Imaging Graph. 84, 101749, `https://doi.org/10.1016/j.compmedimag.2020.101749` (impact factor 4.79, Q1 journal)

- **Ciusdel, C.**, et al., 2022. Normalizing Flows for Out-of-Distribution Detection: Application to Coronary Artery Segmentation. Appl. Sci. 12, 3839. `https://doi.org/10.3390/app12083839` (impact factor 2.68, Q2 journal)

During the collaboration inside the consortium of the European ITFoC project (Information Technology for the Future Of Cancer), the following journal article was published as co-author:

- Tsopra, R., ..., **Ciusdel, C.**, et al., 2021. A framework for validating AI in precision medicine: considerations from the European ITFoC consortium. BMC Med Inform Decis Mak 21:274, `https://doi.org/10.1186/s12911-021-01634-3` (impact factor 2.80, Q3 journal)

An article was published as co-author in the proceedings of an international conference:

- Danu, M., **Ciusdel, C.**, Itu, L., 2020. Deep learning models based on automatic labeling with application in echocardiography. 24th Intl Conf. on System Theory, Control and Computing (ICSTCC). DOI: 10.1109/ICSTCC50638.2020.9259701

The following manuscript was submitted as co-author to a journal, and its status is currently as ”major revision”:

- Hatfaludi, CA., Tache, IA., **Ciusdel, C.**, et al., 2022. Towards a deep-learning approach for prediction of fractional flow reserve from optical coherence tomography. Appl. Sci. (impact factor 2.68, Q2 journal)

The following manuscript was submitted as co-author to an international conference, and its status is currently as ”under review”:

- Hatfaludi, CA., **Ciusdel, C.**, Toma, A., Itu, L.M, 2022. Deep Learning based Aortic Valve Detection and State Classification on Echocardiographies. 20th IEEE Intl Power Electronics and Motion Control Conf.

References

[1] Cireșan, D., et al., 2011. Flexible, High Performance Convolutional Neural Networks for Image Classification. Proc 22nd Intl Joint Conf on Artificial Intelligence. Vol. 2: 1237–1242.

[2] Krizhevsky, A., et al., 2012. ImageNet classification with deep convolutional neural networks. Communications of the ACM. 60 (6): 84–90. doi:10.1145/3065386.

[3] Anaya-Isaza, A., et al., 2021. An overview of deep learning in medical imaging. Informatics in Medicine Unlocked. Vol. 26. https://doi.org/10.1016/j.imu.2021.100723

[4] Stokes M.B., Roberts-Thomson R., 2017. The role of cardiac imaging in clinical practice. Aust Prescr. Vol. 40:151-5. https://doi.org/10.18773/austprescr.2017.045

[5] Ronneberger, O., et al., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597

[6] Teixeira, B., et al., 2019. Adaloss: Adaptive Loss Function for Landmark Localization. arXiv:1908.01070

[7] Long, J., et al., 2015. Fully convolutional networks for semantic segmentation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.

[8] Ghorbani, A., et al., 2020. Deep learning interpretation of echocardiograms. npj Digital Medicine (2020) 3:10. doi.org/10.1038/s41746-019-0216-8

[9] Danu, M., **Ciusdel, C.**, Itu, L., 2020. Deep learning models based on automatic labeling with application in echocardiography. 24th Intl Conf. on System Theory, Control and Computing (ICSTCC). DOI: 10.1109/ICSTCC50638.2020.9259701

[10] Liu, W., et al., 2020. Energy-based Out-of-distribution Detection. Proc. NeurIPS

[11] van Amersfoort, J., et al., 2021. Improving Deterministic Uncertainty Estimation in Deep Learning for Classification and Regression. arXiv:2102.11409v1

[12] Leibfried, F., et al., 2020. A Tutorial on Sparse Gaussian Processes and Variational Inference. arXiv:2012.13962v11

[13] Theodoridis, S. Machine Learning: A Bayesian and Optimization Perspective. 2nd Edition. Chapter 3. Elsevier, 2020. https://doi.org/10.1016/B978-0-12-818803-3.00015-5

[14] Wilson, A.G., et al., 2016. Deep Kernel Learning. Proc. 19th Int. Conf. on AI and Stats (AISTATS, Spain 2016). JMLR: vol 51.

[15] Park, T., et al., 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2332-2341. doi: 10.1109/CVPR.2019.00244.

[16] Arjovsky, M., et al., 2017. Wasserstein GAN. arXiv:1701.07875

[17] Foster, D., 2019. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play. O'Reilly Media, USA. ISBN: 978-1-492-04194-8

[18] Gulrajani, I., et al., 2017. Improved training of wasserstein GANs. Proc. 31st Intl Conf. on Neural Information Processing Systems (NIPS'17).

[19] Miyato, T., et al., 2018. Spectral normalization for generative adversarial networks. Proc. Intl Conf. on Learning Representations 2018.

[20] Sutton, R., Barto, A., 2018. Reinforcement Learning: An Introduction. 2nd Ed. MIT Press.

[21] Ghesu, F., et al., 2017. Multi-Scale Deep Reinforcement Learning for Real-Time 3D-Landmark Detection in CT Scans. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.41:1. doi:10.1109/TPAMI.2017.2782687

[22] Sandler, M., et al., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. IEEE Conf on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.

[23] Dinh, L., et al., 2017. Density Estimation using Real NVP. Proc. ICLR.

[24] Kingma, D.P., Dhariwal, P., 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. Proc. NeurIPS.

[25] Kirichenko, P., et al., 2020. Why Normalizing Flows Fail to Detect Out-of-Distribution Data. Proc. NeurIPS.

[26] Nalisnick, E., et al., 2019. Do Deep Generative Models Know What They Don't Know?. Proc. ICLR.

[27] Karami, M., et al., 2019. Invertible Convolutional Flow. Proc. NeurIPS.

[28] Mark, D.B., et al., 2016. Economic outcomes with anatomical versus functional diagnostic testing for coronary artery disease. Ann Intern Med;165:94–102.

[29] Levin, D.C., et al., 2019. Coronary CT angiography: reversal of earlier utilization trends. J Am Coll Radiol;16:147–55.

[30] Coenen, A., et al., 2018. Diagnostic accuracy of a machine-learning approach to coronary computed tomographic angiography–based fractional flow reserve: result from the MACHINE consortium. Circulation: Cardiovascular Imaging, Vol. 11.

[31] Ryan, T.J., 2002. The coronary angiogram and its seminal contributions to cardiovascular medicine over five decades. Circulation. 106, 752-756.

[32] Mozaffarian, D. et al., 2015. Heart disease and stroke statistics-2015 update: a report from the American Heart Association. Circulation. 131, e29-322. http://doi.org/10.1161/CIR.0000000000000152.

[33] Ng, V.G., Lansky, A.J., 2011. Novel QCA methodologies and angiographic scores. Int J Cardiovasc Imaging. 27, 157-165. http://doi.org/10.1007/s10554-010-9787-9.

[34] Tu, S. et al., 2014. Fractional Flow Reserve calculation from 3-dimensional quantitative coronary angiography and TIMI frame count: A fast computer model to quantify the functional significance of moderately obstructed coronary arteries. JACC Cardiovasc Interv. 7, 768-777. http://doi.org/10.1016/j.jcin.2014.03.004.

[35] Tröbs, M., et al., 2016. Comparison of Fractional Flow Reserve Based on computational fluid dynamics modeling using coronary angiographic vessel morphology versus invasively measured Fractional Flow Reserve. Am J Cardiol. 117, 29-35. http://doi.org/10.1016/j.amjcard.2015.10.008.

[36] Itu, L.M. et al., 2016. A Machine Learning Approach for Computation of Fractional Flow Reserve from Coronary Computed Tomography. J App Physiol. 121, 42-52. http://doi.org/10.1152/japplphysiol.00752.2015

[37] Kroft, L.J.M. et al., 2007. Artifacts in ECG-Synchronized MDCT Coronary Angiography. Cardiac Imaging Review. 189, 581-591.

[38] Dehkordi, M.T., 2016. Extraction of the Best Frames in Coronary Angiograms for Diagnosis and Analysis. J Med Signals Sens. 6, 150–157. http://doi.org/10.4103/2228-7477.186887

[39] Carreiras, C. et al., 2015. BioSPPy - Biosignal Processing in Python, https://github.com/PIA-Group/BioSPPy/.

[40] Marzencki, M. et al., 2014. Diastolic Timed Vibrator: Noninvasive Pre-Hospitalization Treatment of Acute Coronary Ischemia. IEEE Trans Biomed. Circ Syst. 8, 313-324. http://doi.org/10.1109/TBCAS.2013.2270181.

[41] Ronneberger, O. et al., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Proc. MICCAI 2015. https://doi.org/10.1007/978-3-319-24574-4_28

[42] Kingma, D. et al., 2014. Adam: A Method for Stochastic Optimization. Proc. Conf Learning Representations, 2014.

[43] Lin, T.Y. et al., 2017. Focal Loss for Dense Object Detection. Proc. IEEE ICCV. http://doi.org/10.1109/TPAMI.2018.2858826

Abstract

Recent progress related to massively-parallel processors and the availability of large data collections have catalyzed the research of Deep Learning methods. There is a trend to include AI-powered algorithms inside medical imaging pipelines or even as part of the medical scanner itself. Given specific task complexities and large volumes of medical acquisitions, Deep Learning models have rightfully earned their place in the center of diagnosis-aiding solutions. This thesis investigates, proposes and extends DL frameworks and model architectures to cater for the particular aspects of several medical imaging applications related to cardiovascular diseases.

This thesis' structure follows the clinical workflow for a patient suspected with cardiovascular disease. Starting with non-invasive 2D Ultrasound imaging, self-supervised pretraining methods using heuristic pretext tasks are investigated for their utility and robustness in building generalizable prototype models from unlabeled data. Conditional generation of echo frames based on prescribed chamber segmentation masks using generative adversarial networks is researched for its novel and custom data synthesis quality. A video classification task in the form of cardiac phase detection on entire heterogeneous echo acquisitions is solved by a novel DNN model. The topics of semantic segmentation and modeling prediction uncertainties are also investigated. Custom model architectures are tested for their ability to solve specific requirements, such as contour consistency between end-systolic and end-diastolic frames. Uncertainty methods such as Gaussian Processes and Energy-based Models are applied in the context of semantic segmentation. For 3D Ultrasound imaging, a deep reinforcement learning based optimization for a landmark localization task is investigated for its savings in runtime and resources. Next, unsupervised learning techniques such as Normalizing Flows are explored for their capability of explicit density modeling, in an out of distribution detection setup for flagging incorrect lumen segmentations in Coronary Computed Tomography Angiographies. A novel model architecture shows superior detection performance by exploiting semantic features. Finally, the important topic of cardiac phase detection is addressed for Invasive Coronary Angiographies, for which an automated and robust solution was developed.

The solutions to all these tasks unlock new prospects in modern medicine, such as completely automated diagnosing and prognostics systems.