
# Integrating feature attribution methods into the loss function of deep learning classifiers

James Callanan

This thesis is submitted to University College Dublin in part fulfillment of the requirements for the degree of Master of Biomedical Engineering.

**Supervisor:** Professor Kathleen M. Curran and Professor Madeleine Lowery.

April 2022

# Abstract

Feature attribution methods are typically used post-training to determine whether a deep learning classifier is basing classifications off meaningful features in an input image. In this study, a novel technique involving the integration of feature attribution methods into a model's loss function was proposed. These loss functions were coined heatmap loss functions. They were given this name as feature attribution methods produce heatmaps that highlight the relative importance of regions in an image for a given class classification. The heatmap loss function enables the provision of automated feedback to a model about where it should or shouldn't be looking within an image when making a classification.

Two groups of models were trained, one group with a heatmap loss function and the other using categorical cross entropy. Prior to model training, regions of input images were deemed irrelevant for making classifications. The heatmap loss function served to disincentivise models from basing classifications off of features present within those regions.

Models trained with the heatmap loss function achieved equivalent classification accuracies on a test dataset of synthetic cardiac MRI cross sections. Moreover, HiResCAM heatmaps suggest that these models rely to a greater extent on features found within the heart when performing classifications on this test dataset.

A further experiment demonstrated that a heatmap loss function could be used to prevent deep learning classifiers from using irrelevant features that disproportionately co-occur with certain classes when making classifications.

The feature attribution method HiResCAM was integrated into the loss function for the aforementioned experiments. However, models were also successfully trained using a Grad-CAM loss function.

A heatmap loss function could be useful in overcoming the issue of learned biases and to train more skillful classifiers* by directing where a model should look when making classifications in training. An IDF has been filed with NovaUCD and is pending patent investigation.

*Skillful classifiers are classifiers that look at information from the correct regions of images when classifying these images as examples of a given class.

# Statement of Original Authorship

I hereby certify that the submitted work is my own work.

# Acknowledgements

I would like to thank my supervisor, Professor Kathleen M. Curran for her support and guidance over the past year. Kathleen has gone above and beyond to help me and has pushed me to achieve far more than I thought was accomplishable. On top of that, thank you for welcoming me into the Machine Learning in Medical Imaging group.

I would also like to thank the other members of the Machine Learning in Medical Imaging group, specifically Niamh Belton, Carles Garcia Cabrera, Gennady Roshchupkin, Misgina Tsighe Hagos, Katie Noonan, Ronan Hearne and Adil Dahlan. You have all gone completely out out of your way to help me and I really appreciate it.

Thanks to my family and friends too for their support throughout the year!

# Contents

# List of Acronyms

**ACDC** Automated Cardiac Diagnosis Challenge. 2, 3, 4

**ARV** arrhythmogenic right ventricular cardiomyopathy. 20, 21, 22, 23, 26, 27, 28, 33, 36

**DCM** dilated cardiomyopathy. 20, 22, 23, 26, 27, 28, 33

**DL** Deep Learning. 1, 2, 3, 4, 5

**HCM** hypertrophic cardiomyopathy. 20, 21, 22, 23, 26, 27, 28, 32, 33

**MSE** mean squared error. 3

**RL** Reinforcement Learning. 4, 8

**TCAV** Testing with concept activation vectors. 2

Chapter 1

# Introduction

In recent years, fully autonomous vehicles have taken to the roads (Schwall *et al.*, 2020) and several natural language processing models with hundreds of billions of parameters have been unveiled (Du *et al.* (2021), Rae *et al.* (2021), Brown *et al.* (2020)). Given these recent advancements in Deep Learning (DL), one would expect that it will not be long before tasks such as medical image screening and disease diagnosis are fully automated using DL models. As outlined by Litjens *et al.* (2017), DL models that outperform medical experts in image classification tasks have been around for many years, with the diabetic retinopathy classifier of Gulshan *et al.* (2016) and the dermatologist level skin lesion classifier of Esteva *et al.* (2017). Litjens *et al.* (2017) go as far as stating, 'deep learning techniques have permeated the entire field of medical image analysis'. However, in spite of all this, there has been limited uptake of DL techniques in clinical settings and resistance from consumers (Longoni *et al.*, 2019).

This is due to several obstacles that are unique to the training and deployment of DL models in the field of medical imaging.

Shorten and Khoshgoftaar (2019) discuss one major obstacle preventing widespread development of high performing DL medical imaging models, which is limited datasets. According to Shorten and Khoshgoftaar (2019), medical imaging datasets are often far smaller than traditional image classification datasets due to; the expensive nature of acquiring medical images (i.e. the purchasing and staffing of medical imaging machines), the privacy laws surrounding the sharing of medical data and the reliance on medical experts to label data. Moreover, due to the rarity of certain diseases, medical datasets are often highly imbalanced. This is a big issue as often the minority class (i.e. the rare disease) is the class we are most interested in classifying. These factors hinder the development of high-performing DL models which typically require large quantities of high-quality data to train.

Moreover, as of 2018, GDPR requires algorithms that use an individual's data to make decisions that significantly affect the individual, to provide explanations (Goodman and Flaxman, 2017). This is a major obstacle to the uptake of DL models in medical imaging. DL models are renowned for their lack of interpretability and are often referred to as 'black-box' models as a result.

Solving these issues surrounding the training and deployment of DL models in the field of medical imaging is a worthwhile endeavour. DL solutions promise improved performance and speed over their human counterparts. Furthermore, DL solutions are inherently scalable, this means medical image analysis could be made more widely accessible with them.

The broad focus of this project has been on exploring the issue of explainability. This is the more pertinent problem to solve, as the issue of limited datasets will become less relevant with the accumulation of medical imaging data over time. On top of this, the decision was made to work with cardiac MRIs. Cardiac MRIs are a worthy topic of investigation as cardiovascular diseases are the leading cause of death globally (WHO, 2021) and cardiac MRIs are 'an indispensable imaging modality in the investigation of patients with suspected heart disease' (Seraphim *et al.*, 2020).

In the case of cardiac disease diagnosis, expert level classification and segmentation algorithms have been developed as outlined by Bernard *et al.* (2018). However, there is a lack of research into explainable DL cardiac disease classifiers, especially considering the severity and prevalence of cardiovascular diseases. Hence, the initial project aim was to train such a classifier on the Automated Cardiac Diagnosis Challenge (ACDC) dataset and to apply several explainability methods to this model.

The explainability methods; Testing with concept activation vectors (TCAV), Discovery-TCAV, Grad-CAM, Guided Grad-CAM and HiResCAM were all implemented or partially implemented. However, the direction of this research project changed due to difficulties encountered working with the ACDC dataset and the conception of a novel idea along the way. Consequently, the majority of this thesis pertains to a different idea, a feature attribution loss function. These loss functions are called heatmap loss functions. They were given this name as feature attribution methods produce heatmaps. These heatmaps highlight the relative importance of the different regions in an image for a given class classification.

These feature attribution methods are currently used post-training to judge if a DL model is using meaningful features in an input image to make class classifications. Examples of such feature attribution methods include; saliency maps, CAM, Grad-CAM and HiResCAM. Below are some sample Grad-CAM heatmaps taken from (Burduja *et al.*, 2020). They highlight the relative importance of regions in CT scan slices for hemorrhage classifications.

The idea to incorporate existing feature attribution methods into a model's loss function occurred to me when implementing these methods in code. All the feature attribution methods that were

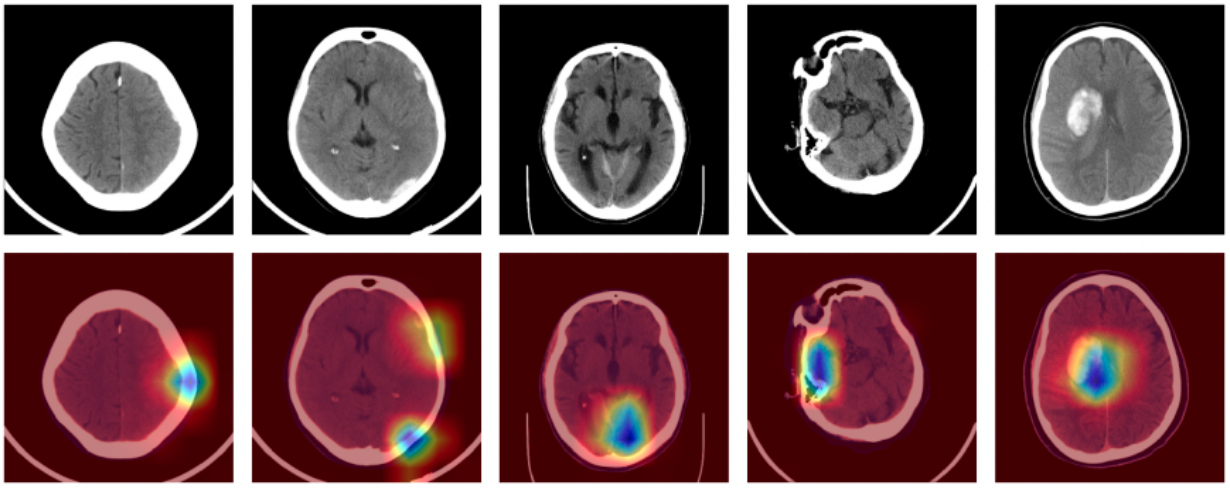

Figure 1.1: Grad-CAM heatmaps for hemorrhage classification (Burduja *et al.*, 2020)

implemented were differentiable with respect to the network's weights and biases. This makes it possible to integrate them into a model's loss function.

The heatmap loss function used in this thesis consisted of a weighted sum of a heatmap component and a mean squared error (MSE) component. The heatmap component serves to disincentivize the classifier from relying on irrelevant portions of images when making classifications and the MSE component acts to incentivise the model to make correct class classifications.

To keep in spirit with the original project's aims, the heatmap loss function was used to train cardiac disease classifiers on datasets of synthetic cardiac MRIs as well as MRIs from the ACDC dataset. Thus, the areas of the MRIs outside of the heart were deemed irrelevant for making classifications. Consequently, the heatmap component of the loss function was set equal to the sum of the heatmap values that lay outside of the heart. This penalises the model for looking outside of the heart when making heart disease classifications in training. Many other metrics have been proposed to evaluate the degree of overlap between ground truth and predicted segmentation masks in segmentation problems (Dice (1945), Belton *et al.* (2021)). Many of these could be easily adapted for use in a heatmap loss function.

The goal of this project is to test whether a heatmap loss function can be used to;

- Train DL classifiers that achieve comparable classification accuracies to equivalent models trained using traditional loss functions.
- Successfully discourage models from using specific features in an input image when making class predictions.

The decision was made to test the above hypotheses by training models on a dataset of synthetic cardiac MRIs. The inspiration for this came from DeepMind. Deepmind are a leading DL research group that perform the majority of their Reinforcement Learning research on simplistic computer generated environments such as atari game worlds (Mnih *et al.*, 2013). Moreover, synthetic datasets have also been employed in the field of DL and biomedical imaging (Kim *et al.*, 2019).

Using synthetic cardiac MRIs is advantageous for several reasons;

- Data availability is not a limiting factor, as more samples can always be generated.
- The classification task can be made harder or easier by making the systematic differences between MRIs of different disease classes larger or smaller.
- The factors which the model should and shouldn't be using to base classifications are known.

Progress made in applying a heatmap loss function to train binary classifiers on the ACDC dataset, which classifies MRIs as either healthy or diseased, is also discussed.

As mentioned previously, a heatmap loss function could be applied outside of medical imaging domains. This approach could be useful in overcoming the issue of learned biases and to train more skillful classifiers by directing where a model should look when making classifications in training.

Chapter 2

# Literature Review

## 2.1 Introduction

In this review, the background knowledge required to engage with this project is provided.

Previous research that compares the performance of models trained using different loss functions is critiqued. Learnings from this research that have shaped this project's methodology are then summarised.

A brief overview of DL explainability methods is then provided.

This is followed by analysis of several popular feature attribution methods. This analysis determined the feature attribution method that was integrated into the heatmap loss function in this study.

A wide range of techniques commonly employed to train DL models on limited datasets are then explored. Learning how researchers have successfully applied these state of the art techniques was relevant to this project as the ACDC dataset consists of only 100 cardiac MRIs.

A brief explanation of why the classifiers in this study were based off VGG and UNet networks is given, with reference to the literature.

## 2.2 Comparing the performance of models trained with different loss functions

Few studies have been carried out to compare the performance of models trained with different loss functions. However, substantial flaws were identified in the approaches of those that were found. Kim *et al.* (2019), Yessou *et al.* (2020) and Cho *et al.* (2019) compared performance metrics achieved by models that were trained using different loss functions. However, they trained a single model for each loss function and fixed the learning rate for all training runs. Moreover, Yessou *et al.* (2020) and Cho *et al.* (2019) did not perform early stopping.

These experimental design choices make it impossible for conclusions to be made about the effects of different loss functions on the performance of models that a DL engineer is likely to train.

This is the case as:

- Statistical inferences cannot be made with samples of size one.
- Different loss functions will result in different loss surfaces for a given dataset. Consequently, a learning rate that is optimal for one loss function is likely suboptimal for another. For this reason, models trained with an arbitrarily chosen fixed loss function are not representative of models that an engineer would deploy, because a DL engineer would tune a model's learning rate.
- Early stopping is usually employed in order to avoid overfitting. Thus, training each model for an arbitrarily chosen fixed number of epochs makes these models unrepresentative of models that an engineer would train.

The aforementioned studies made it clear that, *the performance of models trained with a given loss function*, is an insufficiently defined parameter. In theory, all models are possible regardless of the loss function chosen. Thus, all performances are possible regardless of the loss function chosen. This is the case, as by varying the values of the weights and biases of the neurons in a neural network, a given network architecture can yield an infinite number of model configurations (models).

However, the loss function does affect the models that a DL engineer would consider for deployment*. This is the case as,

- During training, engineers seek models that minimise the loss function for a batch taken from the training dataset via gradient descent.
- A model's loss on unseen data is taken into account when deciding whether to deploy the model.

I propose comparing the performance metrics of models that are trained with different loss functions that a DL engineer is likely to deploy. A set of models that an engineer is likely to deploy is a subset of the set of models that yield small losses for the given loss function. These sets are not equal as there exist many model configurations which yield small losses that an engineer would likely never

*A model applied on data outside of the training, validation and testing dataset is considered deployed.

stumble upon. This is the case because an engineer is likely to follow accepted heuristics when making hyperparameter choices, such as how to initialise the weights of a model.

Thus, in this study, attempts will be made to collect a sample of models that are representative of the population of models that an engineer is likely to deploy. Compiling a representative sample of such models is challenging as there is no consensus on how deep learning models should be trained.

Firstly, there is no consensus on which hyperparameters one should tune when training DL models. Many hyperparameters other than the learning rate and batch size can be tuned. For example, Gulshan *et al.* (2016) tuned the learning rate, early stopping and their image pre-processing methods when training their state of the art diabetic retinopathy classifier. In contrast, Li *et al.* (2020) focused on the effects of tuning a model's momentum. Furthermore, establishing which hyperparameters were tuned in research papers proved to be a difficult task. For example, the researchers that trained classifiers on the ACDC dataset did not mention the hyperparameters that they tuned ((Isensee *et al.*, 2017),(Khened *et al.*, 2017),(Wolterink *et al.*, 2017),(Cetin *et al.*, 2017),(Zheng *et al.*, 2019)). Details of the hyperparameter tuning process were also omitted from the following papers corresponding to the ImageNet winning classifiers ((Krizhevsky *et al.*, 2012),(Simonyan and Zisserman, 2014)). The hyperparameter tuning process inevitably took place when training all the aforementioned models. However, instead of discussing this process, values for parameters that may or may not have been tuned were given. These parameters include the regularisation techniques employed by the researchers (e.g. dropout and dropout rate), the network weight initialisations, the number of training epochs and the different structural parameters of the network architecture. Examples of structural parameters that could be tuned include the number of hidden layers in a network, the number of neurons in these layers and the activation functions applied.

The decision was made to only tune the learning rate of the models used in the experiments in this study. This decision was made due to the ambiguity surrounding the hyperparameters that are frequently tuned. Furthermore, according to Goodfellow *et al.* (2016), 'The learning rate is perhaps the most important hyperparameter. If you have time to tune only one hyperparameter, tune the learning rate.' Moreover, a VGG network that had been pre-trained on the ImageNet database was altered and re-trained for this classification task. The majority of the hyperparameters used in this VGG network were left untouched. Thus, these should be set to reasonable values.

Secondly, there is no consensus on how to tune hyperparameters in machine learning problems. According to Cho *et al.* (2020), some of the most popular methods to optimise a model's hyperparameters are based on Bayesian optimisation. These have been successfully employed in tuning the hyperparameters of medical imaging classifiers such as the diabetic retinopathy fundus classifier of Shankar *et al.* (2020). However, alternative search methods have also been employed, such as random searches (Wang *et al.*, 2018) and full and fractional factorial searches ((Staelin, 2003), (Lujan-Moreno *et al.*, 2018)). Researchers have also investigated the use of Reinforcement Learning agents to tune a DL model's hyperparameters. Neary (2018) employed reinforcement learning to tune the hyperparameters of a CNN trained to classify images from the MNIST dataset and Rijsdijk *et al.* (2021) employed RL to tune the hyperparameters of a DL model trained to detect side-channel security attacks.

A popular library called Keras Tuner (O'Malley *et al.*, 2019) was used to perform the hyperparameter searches. These hyperparameter searches employed a Bayesian optimisation technique to find learning rates which yielded small validation losses.

The weakest assumptions made in this experiment are that:

1. The performance metrics achieved by models trained with the VGG16 architecture, whose only tunable hyperparameter is the learning rate, are representative of the performance metrics achieved by all models a DL engineer would likely train.

2. All of the models that achieved a validation accuracy above 95% are representative of models an engineer would deploy.

Regarding the first assumption. The simplifying approximation to use the VGG network was justified as it is used disproportionately in computer vision tasks, as are all ImageNet winning architectures ((Krizhevsky *et al.*, 2012),(Simonyan and Zisserman, 2014), (Szegedy *et al.*, 2015), (He *et al.*, 2016)).

Regarding the second assumption. An engineer would likely only deploy the model which achieved the maximum validation accuracy among all trained models. The maximum classification accuracy achieved would depend on many factors such as the difficulty of the classification task and the length of time the engineer could afford to model training.

Due to project time constraints, these flawed assumptions were deemed necessary. Despite these assumptions, this methodology is thought to be far superior than the methodologies of Kim *et al.* (2019), Yessou *et al.* (2020) and Cho *et al.* (2019).

## 2.3 DL explainability methods

Rudin (2019) defines DL explainability methods as methods that are applied post-training to provide explanations that help make sense of the decision making process of 'black box' models. These methods are valuable as;

- Models that make decisions which significantly affect users are required by law to provide explanations (Goodman and Flaxman, 2017). Thus, the application of DL models in high-stake scenarios is dependent on DL explainability methods.
- Gaining an understanding of DL models will aid in training models that are aligned with the engineers goals and that achieve higher performances. Large DL models are extremely expensive to train. Thus, methods that could potentially reduce the number of training runs would be extremely valuable. According to Sharir *et al.* (2020), one training run of a language model with 1.5 billion trainable parameters cost approximately $80,000 in 2020.
- The prevalence of DL models is likely going to increase with inevitable developments in the field of DL. Interest in the field of DL is growing, with Alphabet Inc. and Meta, increasing their AI research investments by several billion in 2021 (Rosenbush, 2022).

## 2.4 Feature attribution methods

According to Olah *et al.* (2017), there are two categories of DL explainability methods; feature attribution[†] and feature visualisation methods. The focus of this study was to integrate feature attribution methods into the loss functions used to train models. These loss functions were called heatmap loss functions. They were given this name as feature attribution methods produce heatmaps. These heatmaps highlight the relative importance of the different regions in an image for a given class classification. As mentioned in the introduction, feature attribution methods are generally differentiable with respect to the network's weights and biases. Consequently, it is possible to integrate these methods into a model's loss function.

[†]Feature attribution methods are also called saliency maps.

A heatmap loss function inherits the limitations of the feature attribution method that is integrated into it. Consequently, choosing a feature attribution method to integrate into a heatmap loss function warrants a lot of consideration. This is not a trivial task as there is a lack of consensus on which feature attribution methods work and which ones do not. Moreover, there is not an accepted list of tests to assess a feature attribution method nor are there ground truth heatmaps that can be used to compare generated attribution heatmaps to.

As a result, the evaluation criteria applied to new feature attribution methods before their dissemination have been informal and not quantifiable. Adebayo *et al.* (2018) demonstrated that this approach to evaluating feature attribution methods is vulnerable to human confirmation bias. It has undoubtedly led to the acceptance of feature attribution methods which appear to behave very similar to edge detectors. Adebayo *et al.* (2018) illustrated how many attribution methods produce similar attribution heatmaps for a given image regardless of the configurations of the weights and biases of the model which they are meant to provide insight into.

In the sections that follow, explanations will be given as to why popular feature attribution methods were not chosen to be integrated into a heatmap loss function in this study.

### CAM

CAM was not considered as it can only be applied to a limited set of networks[‡]. Although, to the best of my knowledge, CAM heatmaps provide meaningful explanations.

### Grad-CAM

Grad-CAM succeeded CAM. Unlike CAM, Grad-CAM can be applied to a much wider range of DL models[§]. The Grad-CAM loss function was successfully integrated into a heatmap loss function and used to train a model on the dataset of synthetic cardiac MRIs. Grad-CAM passed the assessments laid out by Adebayo *et al.* (2018) and has been widely used to bring explainability to medical imaging DL models ((Pasa *et al.*, 2019),(Baltruschat *et al.*, 2019),(Belton *et al.*, 2021)). However, the main experiment was not carried out with a Grad-CAM loss function due to a major flaw with Grad-CAM that was highlighted by Draelos and Carin (2020). Draelos and Carin (2020)

[‡]CAM can only be applied to networks that apply global average pooling to the feature maps that are output from the last convolutional layer of a network. These pooled feature maps must also be followed by a single fully connected layer which makes classifications.

[§]Grad-CAM can be applied to all models whose class logits are differentiable with respect to the feature maps that are output from the network's last convolutional layer.

demonstrated that Grad-CAM highlights regions of images as important that were not used to base classifications. This is due to the importance of features in a feature map being computed by multiplying the activations of the neurons in that feature map by the average gradient of the class logit[¶] with respect to that feature map. By averaging the gradients in a feature map, information is being lost and less accurate attribution heatmaps are produced. According to Draelos and Carin (2020), the averaging of gradients was likely inspired by the global average pooling of feature maps in CAM.

### HiResCAM

To overcome the limitation of Grad-CAM, Draelos and Carin (2020) proposed HiResCAM. The importance of features in a HiResCAM feature map are computed by performing an element wise multiplication between the activations of the neurons in that feature map and the gradient of the class logit with respect to that feature map. Thus, the gradient of the class logit with respect to a given neuron in the feature map is multiplied by that neuron's activation. Consequently, this illustrates the regions which the model is placing importance on more truthfully than Grad-CAM heatmaps do. HiResCAM heatmaps were found to be more precise than GradCAM on a high performing model as a result (Draelos and Carin, 2020). The assessments outlined by Adebayo *et al.* (2018) and Kindermans *et al.* (2019) have not yet been performed on HiResCAM. However, as HiResCAM is very similar to Grad-CAM, one would expect it to pass these assessments too. Moreover, the method has not received many citations. Thus, it likely needs to be further verified.

### Saliency maps

Saliency maps are found by computing the derivative of the activation of the output neuron that corresponds to a particular class classification with respect to the pixels in an input image. According to the authors, 'The magnitude of the derivative indicates which pixels need to be changed the least to affect the class score the most' (Simonyan *et al.*, 2013). Saliency maps were not integrated into a heatmap loss function due lack of time. They passed the feature attribution method assessments outlined by Adebayo *et al.* (2018) and Kindermans *et al.* (2019). To the best of my knowledge, they provide meaningful explanations.

[¶]Class logits are the activations of neurons in the output layer of a network that correspond to the different class classifications **before** these activations have been normalised by passing them through a normalisation function (e.g. sigmoid function)

**Gradients x input**

These feature attribution maps are produced by performing an element wise multiplication between the pixels in the aforementioned saliency maps with the values of the pixels in the input image. This method was not considered as it failed the tests outlined by Adebayo *et al.* (2018) and Kindermans *et al.* (2019). Kindermans *et al.* (2019) demonstrated that input images could be altered so that the activations of the output neuron corresponding to a given class classification would not change yet the *gradients x input* attribution heatmap would change substantially. To quote the authors, they were able to 'purposefully create a deceptive explanation of the network' by superimposing a hand drawn cat image over images from the MNIST dataset. An illustration of the results from this test can be seen below.

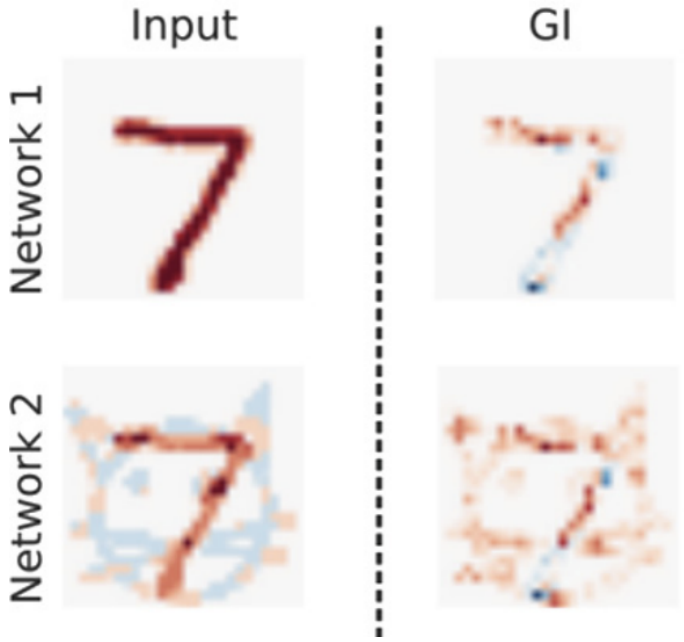


Figure 2.1: The two images in the leftmost column produce identical network classifications yet yield very different Gradients x input attribution maps as can be seen in the rightmost column. Image adapted from Kindermans *et al.* (2019)

Before any feature attribution method is integrated into a heatmap loss function to train a model in a high-stake classification task, a consensus should be reached on an assessment protocol to test the validity of feature attribution methods in general. Until we can be sure that the feature attribution method that is integrated into the heatmap loss function provides truthful explanations, it is likely that the trained classifier would learn undesirable behaviours.

## 2.5 Working with small datasets

Without sufficiently large datasets, DL models are prone to overfitting. Overfitting occurs when a model fits to the training dataset but does not generalise well on unseen data. Ying (2019) highlights three causes of overfitting, these include; the presence of noise in the training dataset, too small a training dataset and the use of overly complex classifiers for the task at hand.

Approaches to combat overfitting include dataset expansion, data augmentation and the application of other regularisation techniques such as; L1, L2 and dropout regularisation.

One cannot always build high performing models using small datasets. If a dataset is too small, it simply can not adequately represent the diversity of the population from which it was sampled even with the application of the aforementioned regularisation techniques.

### 2.5.1 Data Augmentation

Data augmentation is a method employed to artificially increase the size and improve the quality of a training dataset. It can help combat overfitting by making a training dataset more representative of the population from which it was sampled. Shorten and Khoshgoftaar (2019) divide data augmentation techniques into two categories; data warping and oversampling.

Data warping techniques involve altering existing data in the training dataset in order to increase its diversity.

Examples of data warping techniques include the application of:

- Geometric transformations such as; rotating, translating, scaling, cropping and elastic deformations.
- Colour transformations such as colour shifting and lighting changes.
- Noise injection.
- Random erasing, where patches of varying shapes and sizes are 'erased' (i.e. replaced with pixels/voxels of a constant value).
- Applying kernel filters to sharpen/blur images or techniques such as neural style transfer.

Care must be taken when applying data warping augmentations to ensure that the data's label is preserved. For example, applying random erasing to mammograms could lead to all evidence of cancerous growth in a cancerous mammogram being removed.

Wolterink *et al.* (2017) applied 90° rotations to the MRIs in the ACDC dataset when training their classifier. This was likely done as MRIs in the ACDC dataset appear to be roughly aligned with either the x or y axes. However, applying rotations within a random range would likely lead to a model which generalises better. Isensee *et al.* (2017) applied random rotations to cardiac MRIs in the ACDC dataset when training classification and segmentation models. Isensee *et al.* (2017) also applied other data warping techniques such as; gamma-corrections, elastic deformations, the mirroring of MRIs along the x and y axes and slice translations to mimic motion augmentation. Similarly, Khened *et al.* (2017) applied translations along the x and y axes as well as random rotations on cardiac MRIs from the ACDC dataset. However, Khened *et al.* (2017) also applied random zoom factors between 0.6 and 1.4 to Cardiac MRIs in order to increase the dataset diversity. Although not mentioned by Baumgartner *et al.* (2017*a*) in their paper, based on the source code corresponding to their implementation[‖], random rotations within a range of -15° and 15° as well as left right flips were applied to cardiac MRIs. According to the original paper on the UNet architecture (Ronneberger *et al.*, 2015), the application of elastic deformations was crucial to training biomedical segmentation networks on small datasets. These transforms were said to simulate deformations of biological tissue efficiently. Translations and random rotations were also applied.

In this study, the source code provided by (Baumgartner *et al.*, 2017*b*) was adapted to train a 3D UNet to segment cardiac MRIs from the ACDC dataset. Consequently, when the UNet was transformed into a classification model, the augmentation methods applied by Baumgartner *et al.* (2017*a*) were still used. These included minor rotations within a range of -15° and 15° as well as the random application of left right flips. As high-performing classifiers were not trained using this set up, additional data warping techniques were applied. These included the application of elastic transforms as recommended by Ronneberger *et al.* (2015) and as successfully implemented by Isensee *et al.* (2017), the implementation of translations along the x and y dimensions to mimic motion artefacts and the injection of Gaussian noise. The regularisation technique dropout was also trialled as this was used in the original UNet paper (Ronneberger *et al.*, 2015) and by Khened *et al.* (2017).

[‖]See *augmentation_function()* in **train.py** (Baumgartner *et al.*, 2017*b*)

**Oversampling:**

Oversampling involves adding synthetic images to your training dataset.

Examples of oversampling methods include,

- Random Oversampling (ROS), where random examples of the minority class are duplicated.
- Using generative models such as variational autoencoders or generative adversarial networks to create synthetic images.

Oversampling is typically performed only on imbalanced datasets where the classes in a dataset are not equally represented. As the ACDC dataset was balanced, containing 20 cardiac MRIs from each disease class, none of previously mentioned researchers implemented oversampling techniques when training classification or segmentation algorithms. However, in this study, the decision was made to simplify the classification task as no progress was made training a multiclass classifier on the ACDC dataset. Consequently, cardiac MRIs were split into two classes, normal and diseased. The resulting dataset was imbalanced, with 20 cardiac MRIs corresponding to normal patients and 80 corresponding to patients with heart disease. As a result, the issue of class imbalance had to be solved. The decision was made to use an alternative approach which involved leaving the dataset untouched and altering the weighting assigned to samples of data from each class in the model's loss function. The latter approach can be easily applied in modern deep learning frameworks. For example, Keras' *model.build()* method allows you to provide different weightings to sample data from different classes via a *class_weight* parameter.

### 2.5.2 Transfer learning

Transfer learning is used as it can reduce the time taken for training convergence and can lead to improved model performance on tasks with small datasets. In their survey paper on transfer learning, Weiss *et al.* (2016) define transfer learning as the process of improving model performance at a target task (task B) by using learnings from a model that was trained on a different dataset to carry out a different task (task A).

Ng (2017), states that transfer learning only makes sense when:

- Tasks A and B take the same type of data as inputs (e.g. both take image data as inputs).
- There is a lot more data for task A than task B. This is the case as data for task A is inherently less valuable than data for task B (i.e. if you had sufficient data from task B, then there would be no incentive to use data from task A).
- Low level features learned from task A can be helpful in task B. Feature extractors such as edge and high-low frequency detectors have been discovered in many computer vision models trained on different datasets (Olah *et al.*, 2020). It is likely that such feature extractors would be useful for most computer vision tasks.

Models trained on the ImageNet dataset are often chosen for transfer learning in computer vision tasks due to the sheer size of the dataset. However, these models are not ideally suited for classifying volumetric medical data such as cardiac MRIs. 2D pre-trained models have been used to segment 3D data by dividing 3D volumes into 2D slices ((Yu *et al.*, 2018), (Han, 2017)). However, using 3D CNNs to segment 3D data has been found to give far superior segmentation performance than splitting this data into 2D slices and using 2D CNNs to perform sequential segmentations (Lai, 2015). It was for this reason that Med3D was developed (Chen *et al.*, 2019). Med3D is a 3D CNN trained on volumetric medical images of several organs gathered from several medical imaging modalities. The use of Med3D in this project was considered. However, it was not used as it was implemented using a DL framework that I was not familiar with. Instead the decision was made to train a 3D UNet to segment MRIs from the ACDC dataset and to build a classifier around the encoder from this UNet. This decision was made as open source code to train a 3D UNet to classify cardiac MRIs using a DL framework familiar to me was found online (Baumgartner *et al.*, 2017*b*). This was deemed a reasonable approach as encoders from UNets had been successfully used as feature extractors for classification models working with x-ray data ((Soulami *et al.*, 2021), (Dong *et al.*, 2020)). Furthermore, encoders from auto-encoders are frequently used as feature extractors for classification tasks. For example, Liu *et al.* (2021) and Betechuoh *et al.* (2006) based disease classifiers around encoders taken from auto-encoders.

## 2.6 Model architectures used

### 2.6.1 VGG network

A VGG16 network that had been pre-trained on the ImageNet dataset was used for the experiments involving the synthetic cardiac MRIs.

The VGG network was developed by Simonyan and Zisserman (2014). This network placed first in the image localisation task and second in the image classification task in the ImageNet Large Scale Visual Recognition Challenge in 2014. The VGG network succeeded and outperformed AlexNet (Krizhevsky *et al.*, 2012). The VGG network differs from AlexNet by using smaller convolutional filters with a receptive field of (3x3). The use of smaller filters enabled the VGG network to be much deeper than AlexNet. There are two combinations of the VGG network, VGG16 and VGG19. These have 16 and 19 convolutional layers respectively. The classifier used in this experiment was based on the VGG16 network.

VGG16 was chosen for the task of classification on the dataset of synthetic cardiac MRIs as it had been successfully applied by many researchers in medical imaging domains. A VGG16 architecture was found to outperform approximately 20 other commonly used CNN architectures for the task of cardiac short axis slice classification by Ho and Kim (2021). Moreover, a transfer learned VGG16 network was shown to outperform a transfer learned Xception network at classifying chest X-rays as normal or pneumonic by Ayan and Ünver (2019). VGG16 also achieved 94% classification accuracy at classifying skin cancer classification from skin lesion images (Khamparia *et al.*, 2021).

Below is a diagram of the VGG16 network 2.2. This model architecture differs slightly from the architecture used in the experiments in this project. The final three fully connected layers as well as the softmax layers from the pre-trained VGG16 model were removed. These were replaced with a single fully connected layer consisting of 100 neurons, followed by a fully connected layer with four neurons (one for each disease class) and a softmax layer.

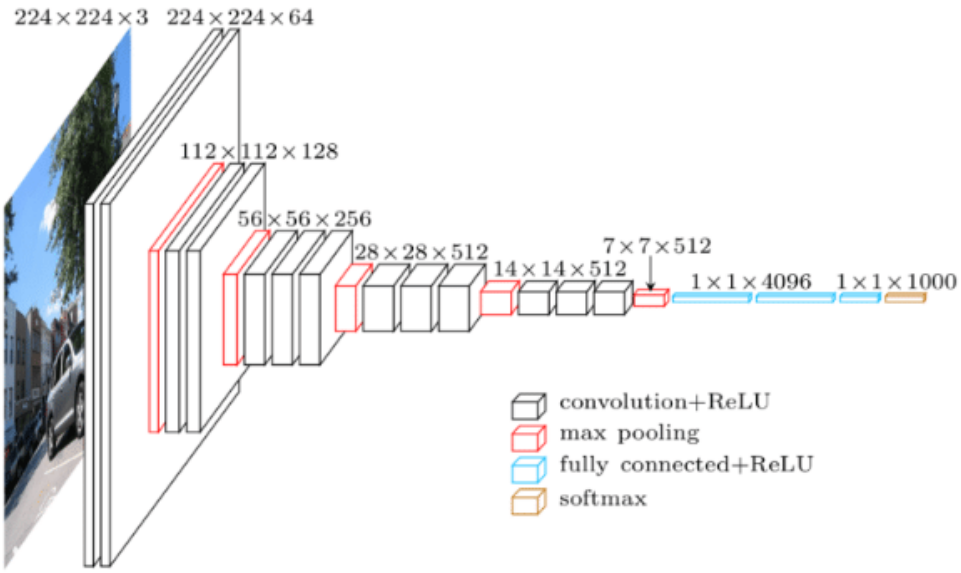


Figure 2.2: Structure of VGG16 (Sugata and Yang, 2017)

### 2.6.2 UNet architecture

The UNet architecture was developed by Ronneberger *et al.* (2015) to segment cells in transmitted light microscopy images. It won the ISBI cell tracking challenge in 2015 and is one of the most popular architectures for semantic segmentation today. Nine out of the ten cardiac segmentation algorithms referenced in the ACDC study were UNet or 'UNet like' networks (Bernard *et al.*, 2018). No CNN had been trained to classify cardiac disease from MRIs on the ACDC dataset. This was likely due to the small dataset, which contained only 100 MRIs. However, features extracted from the segmentation masks produced by UNets had been used to train machine learning classifiers such as random forests, support vector machines and multi layer perceptrons ((Khened *et al.*, 2017), (Cetin *et al.*, 2017), (Isensee *et al.*, 2017)). Thus, it was hypothesised that a fully convolutional network which made use of the encoder from a UNet could be used to train the first CNN to classify cardiac MRIs on this dataset. Thus, a 3D UNet was trained to segment cardiac MRIs from the ACDC dataset. The decoder from this UNet was then discarded and it was replaced with two fully connected layers with many neurons, followed by a fully connected layer with four neurons (one for each disease class) and a softmax layer. This formed the classification architecture.

Below is an image of a 2D UNet architecture 2.3. It consists of two sections, the encoder and decoder. These are also known as the contracting path and the expanding path. The encoder, which can be seen on the left, was kept for use in the classifier. The decoder, which can be seen on the right, was discarded. The diagram below is of a 2D UNet, however, a 3D UNet was used in this

study. A 3D UNet was chosen as 3D CNNs trained to segment 3D data were found to outperform 2D CNNs used to sequentially segment 2D slices from 3D data (Lai, 2015). Thus, it was inferred that 3D CNNs would serve as better feature extractors for 3D data. The only difference between a 3D UNet and a 2D UNet is that 3D convolution, pooling and up convolution operations are performed instead of their 2D counterparts.

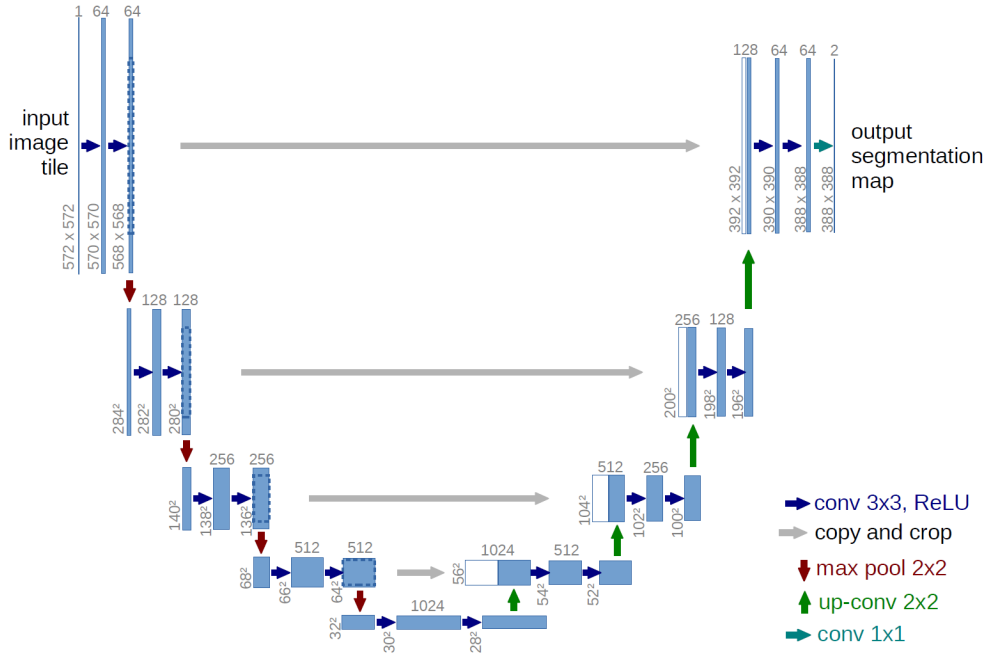


Figure 2.3: Structure of a 2D UNet (Ronneberger *et al.*, 2015)

Chapter 3

# Methodology and Results

## 3.1 Dataset

The purpose of the synthetic dataset of cardiac MRIs was to perform the experiments proposed below in order to test the feasibility of a heatmap loss function. Attempts were made to make the synthetic dataset mimic the ACDC dataset. These attempts were made as the heatmap loss function would be employed to train models on the ACDC dataset if results were promising on the synthetic dataset.

The synthetic datasets consisted of the disease classes; normal, hypertrophic cardiomyopathy, dilated cardiomyopathy and arrhythmogenic right ventricular cardiomyopathy. This dataset was balanced much like the ACDC dataset. However, there were several discrepancies between the two datasets. The major differences included;

Dataset size: The ACDC dataset contains 100 cardiac MRIs whereas it was decided to include 10,000 MRIs in the synthetic training dataset and 3,000 in the validation dataset. Large dataset sizes were chosen to reduce the likelihood that a limited dataset would interfere with testing the feasibility of the novel loss function.

MRIs in the ACDC dataset are a time series of volumetric data. MRIs in the synthetic dataset were 2D MRI cross sections. This simplification was made as the dimensionality of the input data was deemed irrelevant for the purposes of testing the feasibility of the loss function

MRIs of patients with myocardial infarction were not included in the synthetic dataset. The biomarkers of this disease were not as straightforward to model as the other four diseases in the ACDC dataset. The decision to omit this disease class was made as perfectly mimicking the ACDC dataset was deemed irrelevant for the purposes of testing the feasibility of the loss function.

Several attempts were made to make the synthetic cardiac MRI cross sections representative of the underlying diseases and of real world cardiac MRIs. An explanation of how these MRIs were generated is given below. However, one may find reading the source code used to generate these

MRIs more insightful. The code can be found here. The function used to generate the MRIs was called *make_mri_and_seg_mask()*.

### 3.1.1 Synthetic cardiac MRI details

The prevalence of the diseases among male and female sexes was accounted for, with approximately 2.7 times more male cases of ARV than female cases. Similarly, there were approximately 1.3 times more male cases of HCM than female cases.

A size multiplier was used to scale the radii of the heart chambers and the degree of body fat in MRIs based on the individual's body fat percentage. As chamber radii were made proportional to the size multiplier, the area of these chambers in the 2D cross sections were proportional to the square of the size multiplier.

Size multipliers for female MRIs were pulled from a Gaussian distribution with a mean of 1 and standard deviation of 0.15. Size multipliers for male MRIs were pulled from a Gaussian distribution with a mean of 1.3 and a standard deviation of 0.15. The inspiration to model size multipliers this way was based on the distribution of male and female heights. As can be seen below, male and female heights can be well modelled by Gaussian distributions where both distributions have approximately equal standard deviations and the mean male height is approximately 2 standard deviations larger than the mean female height.

Subcutaneous body fat was also included in the synthetic MRIs. The amount of pixels taken up by body fat in an MRI cross section was determined by multiplying the individual's body fat percentage by their size multiplier. Data regarding the distributions of male and female body fat percentages among Mongolian Adults was found (OtgontuyaE, 2009). As body fat was not deemed crucial to this experiment, these distributions of body fat percentage were used. Body fat percentage among patients in the synthetic dataset were normally distributed with males having a body fat percentage of 26% and a standard deviation of 7.9% and females having a body fat percentage of 36.4% and a standard deviation of 9.7%.

Prior to being scaled by the size multiplier, radii of heart chambers and chamber wall thicknesses were assumed to be normally distributed for a given disease class. The mean radius/thickness for each disease class were loosely estimated based on descriptions for the diseases found on radiopaedia ((Feger and Radiopaedia, 2021), (Saber and Radiopaedia, 2021), (Weerakkody and Radiopaedia, 2021)) and inferred from diagrams of characteristic hearts with the different diseases found online

((Clinic, 2021), (Clinic, 2020), (UMichigan, n.d.)). The final radius/wall thickness for an individual MRI was found by multiplying the individual's size multiplier by the sampled value and adding some individual variability. This variability was assumed to be normally distributed with a mean of 0 and a standard deviation which was proportional to the size of the radius/thickness prior to adding the variability.

Below, the main deviations between MRIs of the different disease classes are described:

- **HCM:** These MRIs had thicker left ventricle walls.
- **DCM:** These MRIs had more dilated left ventricles (i.e. larger left chamber radius). The body fat of individuals with this disease were also pulled from a different Gaussian distribution because obesity is a contributing factor for this disease.
- **ARV:** Two subtypes of ARV were modelled, fatty ARV and fatty fibro ARV. These MRIs had more dilated right ventricles and fat was included in the chamber walls. The percentage of fat was normally distributed.

Other less substantial systematic differences in heart dimensions were modelled. They can be discovered by looking at the source code if required.

## 3.2 Experiment 1

### 3.2.1 Methods

The aim of the first experiment was to compare the performance of models trained with the novel heatmap loss function to those trained with CCE. Below is the experimental plan which had been devised prior to performing this experiment.

Gather a collection of models trained with the heatmap loss function that achieve a validation accuracy of over 95%. These represent models a DL engineer would likely choose to deploy. Gather a collection of models trained with CCE that achieve a validation accuracy of over 95Compute the average degree of overlap between the HiResCAM heatmaps and the heart for all selected models in both groups. All metrics will be evaluated on the same test dataset. Perform a two-sided statistical test to test for statistically significant differences in the distributions of the above metrics among both groups of models. Perform a non-parametric statistical test (Mann-Whiteney U-test) if the

distribution of performance metrics among either group is not normally distributed. Otherwise, perform a parametric test (t-test).

In an attempt to remove some sources of variability, it was decided that all models would be trained and evaluated on the same datasets. As larger datasets yield more accurate estimations, a dataset of 10,000 synthetic cardiac MRIs was used to estimate a model's mean heatmap-heart overlap on unseen MRIs. Substantially larger dataset sizes were ruled out as they would not fit in a single NumPy array on the low RAM Colab GPU instances. Fitting the dataset into a single NumPy array was desirable as inference would be quicker if the data only needed to be loaded once. Otherwise, the dataset size of 10,000 was arbitrarily chosen.

Below are sample cardiac MRI cross sections used in this experiment.

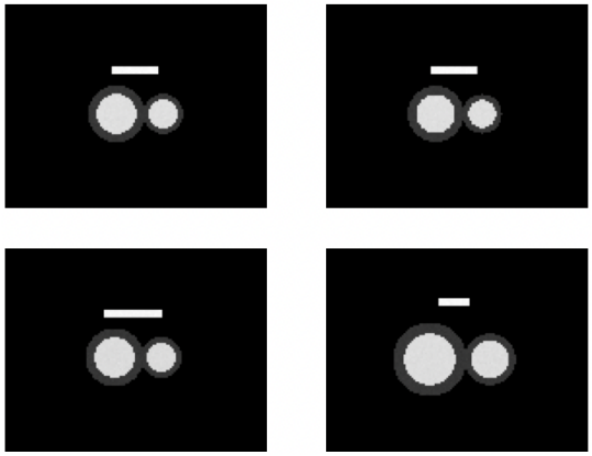

Figure 3.1: Exp 1: Normal MRIs

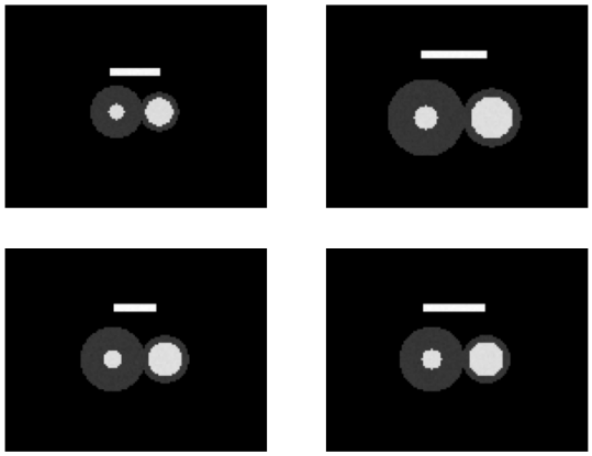

Figure 3.2: Exp 1: HCM MRIs

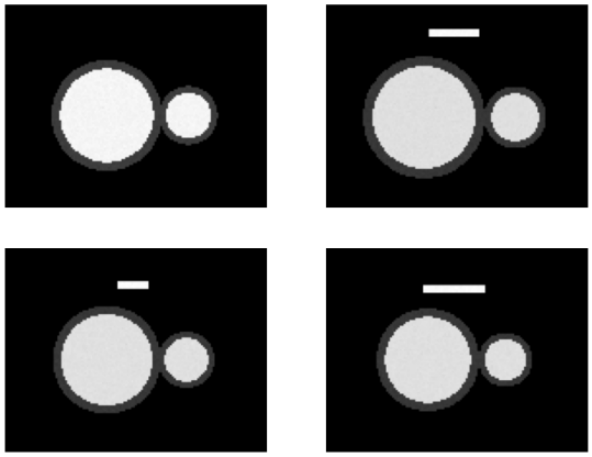

Figure 3.3: Exp 1: DCM MRIs

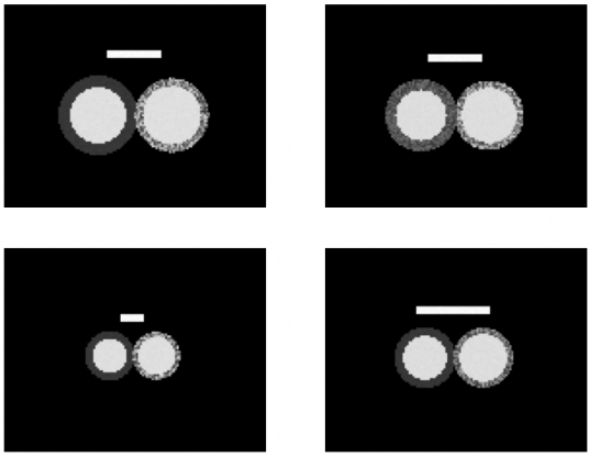

Figure 3.4: Exp 1: ARV MRIs

Initially two models were trained, one model using the heatmap loss function and one model using the CCE loss function. Many learning rates were trialled before a model was found whose validation

loss and accuracy stably converged. These training runs were performed manually in several Colab notebooks. Tracking model results and the latest code got messy very quickly as several different implementations had been run in different Colab notebooks. Thus, several process improvements were made. These included;

- Making a GitHub repository to track all code that was shared between multiple notebooks. This meant the Git repository could be cloned into each Colab notebook and the necessary functions could be imported. This meant there were never several implementations of functions across different notebooks.
- Using a library called Keras Tuner to automate the hyperparameter searches.
- Saving trained models in a directory on Google Drive along with their performance metrics (as evaluated on a test set) in a json file.
- Setting up a local PostgreSQL database and using a tool called ngrok to expose the Postgres server to the internet. This made it possible to have multiple hyperparameter searches be carried out simultaneously. Prior to this, some model's performance metrics were lost as two notebook instances were simultaneously writing to the json file used to track results.

Two models were then trained with each loss function using the aforementioned workflow. The mean HiResCAM-heart overlap were computed on the test dataset for these four trained models. The mean and standard deviations of the overlap metrics for models trained with both loss functions were estimated. Using these estimates, a Cohen's d effect size of 7.6 was calculated. Given an estimated effect size of 7.6 and an alpha level of 0.05, three models would need to be trained using each loss function in order to achieve an arbitrarily chosen desired power of 0.9999. In hindsight, a more conservative alpha level should have been set given how few samples were needed in order to conduct a test with such a high power given the estimated effect size.

As multiple hyperparameter searches could be conducted in parallel using the automated workflow that had been developed, it was decided to train approximately thirty models using the heatmap loss function and the CCE loss function overnight. A large fraction of these models were expected to be discarded because they would not have achieved a validation accuracy of over 95%. After filtering for models which achieved a validation accuracy of greater than 95%, 23 models trained with the heatmap loss function and 25 models trained with CCE remained.

Mean HiResCAM-heart overlaps were computed for models in both groups. Two Shapiro-Wilkes tests ($\alpha = 0.05$) were performed to test whether the distributions of these metrics within each group were normally distributed. The null hypothesis for a Shapiro-Wilkes test is that the data in a sample was drawn from a normal distribution. Both Shapiro-Wilkes tests returned p-values less than 0.01. As these are less than the preset significance level of 0.05, the null hypothesis that the distribution of mean heart overlap metrics were drawn from a normal distribution was rejected.

Consequently, a two-sided Mann-Whitney U-test ($\alpha = 0.05$) was performed to test for a statistically significant difference between the two group's HiResCAM-heart overlaps. The null hypothesis was that there is no systematic difference between the mean HiResCAM-heart overlaps of models trained with a heatmap loss function and the mean HiResCAM-heart overlaps of models trained with a CCE loss function.

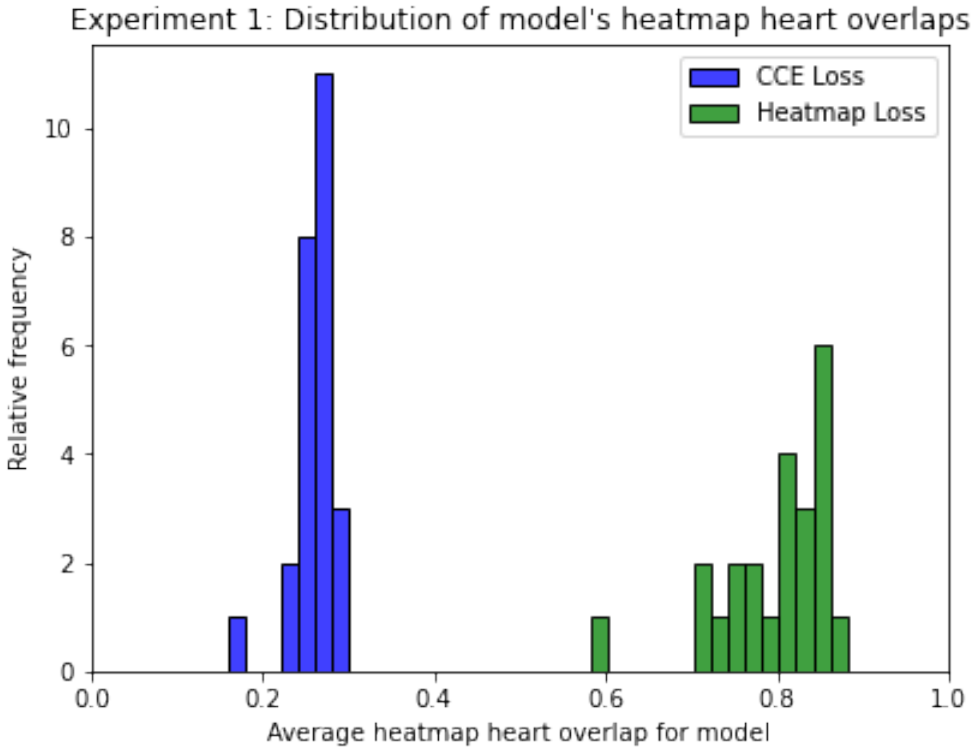


Figure 3.5: Experiment 1: Distribution of HiResCAM-heart overlaps

### 3.2.2 Results

This Mann-Whitney U-test returned a p-value $\approx 1 \times 10^{-9}$. This is less than the preset significance level of 0.05. Thus, the null hypothesis is rejected. These results support the alternative hypothesis that the mean HiResCAM-heart overlaps of models trained with the heatmap loss function are systematically higher than the mean HiResCAM-heart overlaps of models trained with CCE. As

all mean HiResCAM-heart overlaps for models trained with CCE were smaller than all mean HiResCAM-heart overlaps for models trained with the heatmap loss function, a common language effect size of 0.0 was calculated. The Cohen's d effect size was calculated to be -10.78, this is an extremely large effect size. Assuming the HiResCAM heatmaps provide truthful visualisations, these results suggest that models trained with the HiResCAM loss function appear to base their classifications to a higher extent off of information lying in regions within the heart than models trained with CCE.

Below are sample HiResCAM heatmaps produced by one randomly chosen model trained with a CCE loss function.

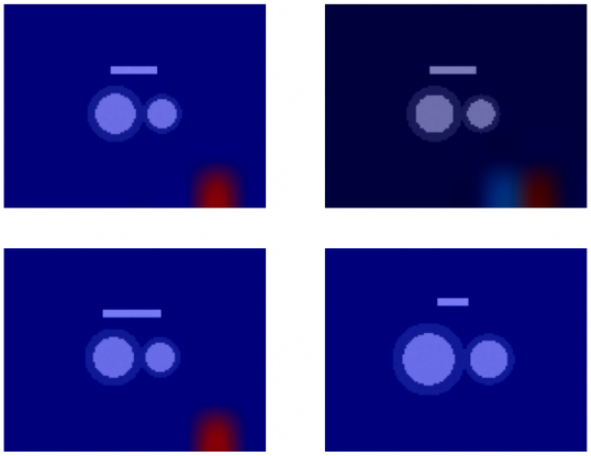

Figure 3.6: Exp 1: Normal MRI heatmaps

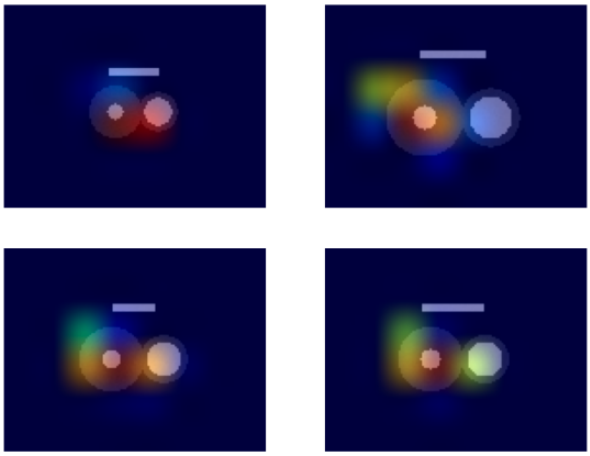

Figure 3.7: Exp 1: HCM MRIs heatmaps

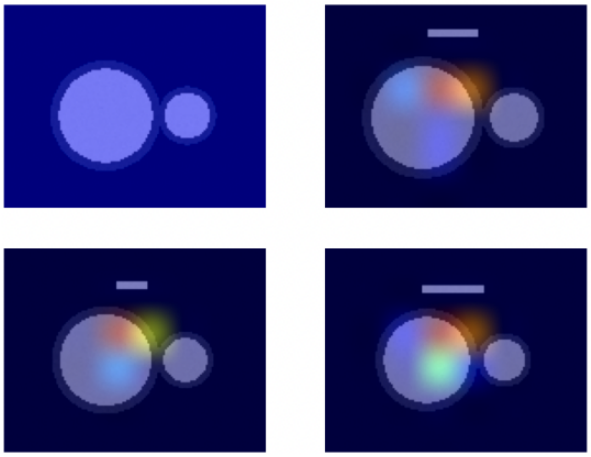

Figure 3.8: Exp 1: DCM MRIs heatmaps

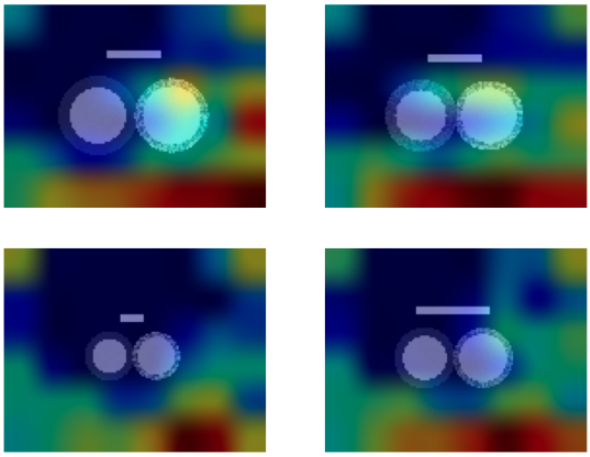

Figure 3.9: Exp 1: ARV MRIs heatmaps

Below are sample HiResCAM heatmaps produced by one randomly chosen model trained with a heatmap loss function.

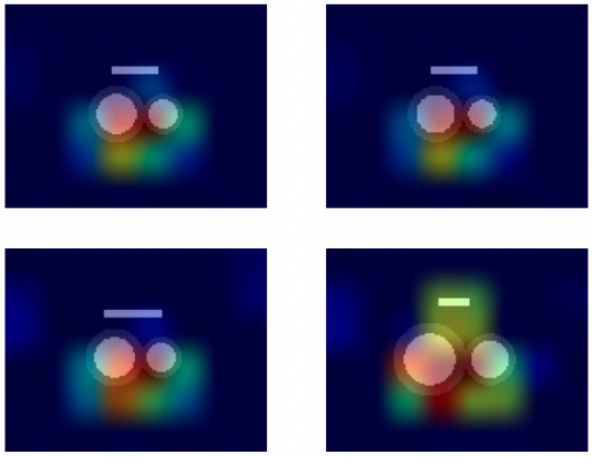

Figure 3.10: Exp 1: Normal MRIs heatmaps

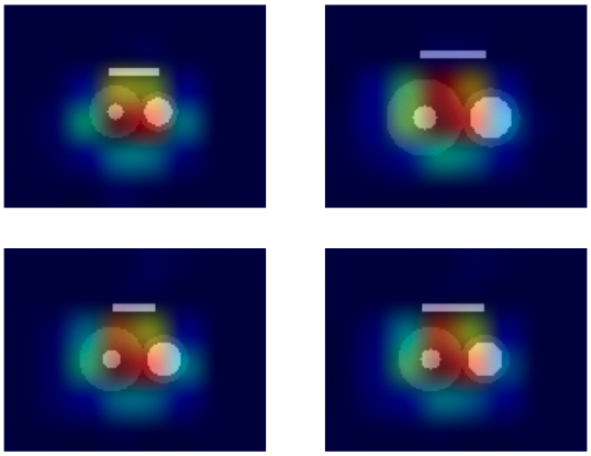

Figure 3.11: Exp 1: HCM MRIs heatmaps

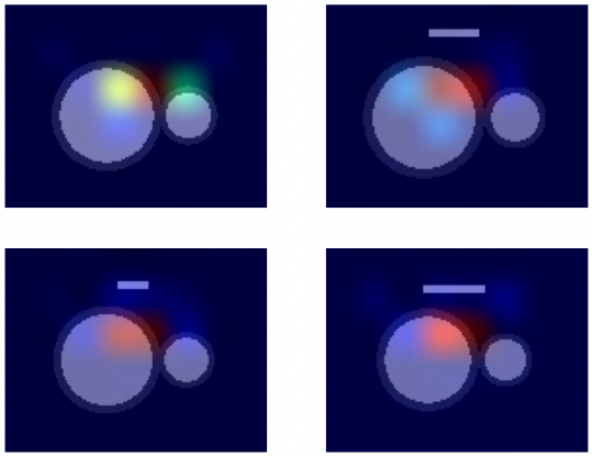

Figure 3.12: Exp 1: DCM MRIs heatmaps

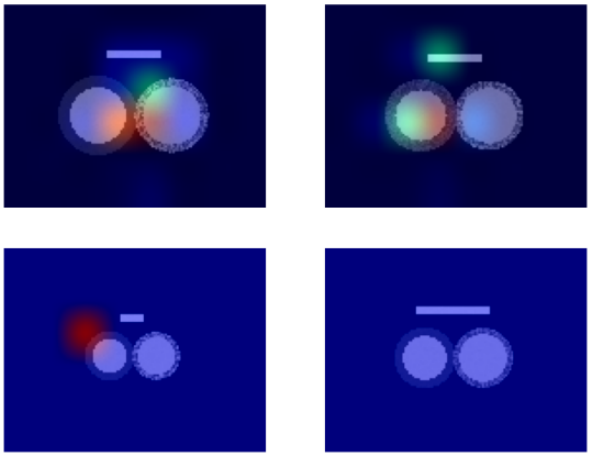

Figure 3.13: Exp 1: ARV MRIs heatmaps

## 3.3 Experiment 2

### 3.3.1 Methods

A second experiment was carried out to test whether models trained with either loss function were relying on knowledge of the patient's sex when making classifications. It was theorised that a model trained with a traditional loss function may base disease classifications based off knowledge of a patient's sex. We hypothesised that this would be most likely to occur when classifying diseases such as ARV, as this disease occurs a disproportionate amount in males in our training dataset (as well as in real life).

For this experiment, all systematic differences between the MRIs of males and females were removed. This was implemented by pulling the size multiplier described in the dataset section above from the same distribution for male and female patients. Thus, there should be no systematic differences between the chamber radii, wall thicknesses and body fat in male and female MRIs. However, a label was included in the bottom corner of male patient's MRIs. This enabled the separation of the feature, sex, from the heart. Thus, a model's reliance on information of the patient's sex could be calculated by measuring the degree of overlap between the feature attribution heatmaps and the sex labels.

Below are sample cardiac MRI cross sections used in this experiment.

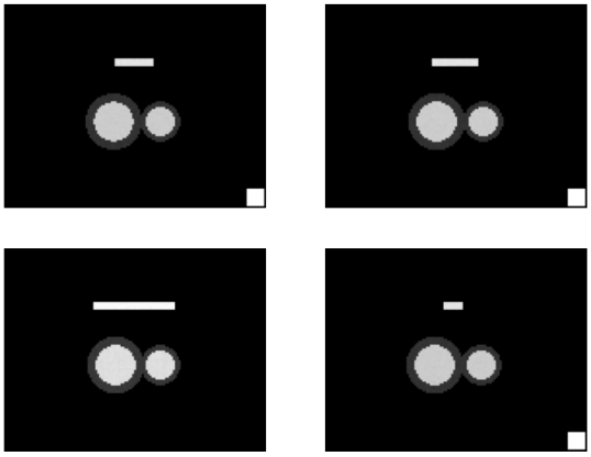

Figure 3.14: Exp 2: Normal MRIs

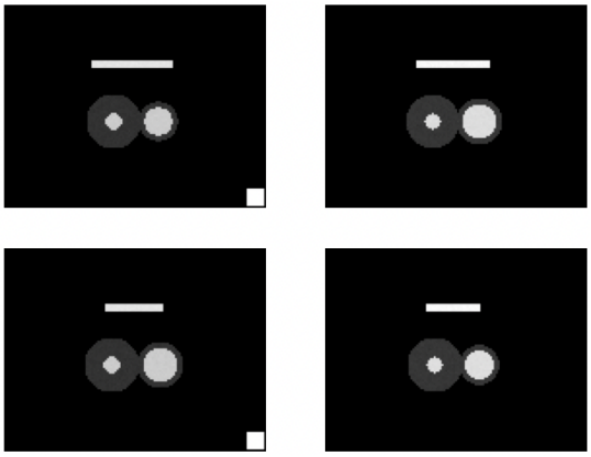

Figure 3.15: Exp 2: HCM MRIs

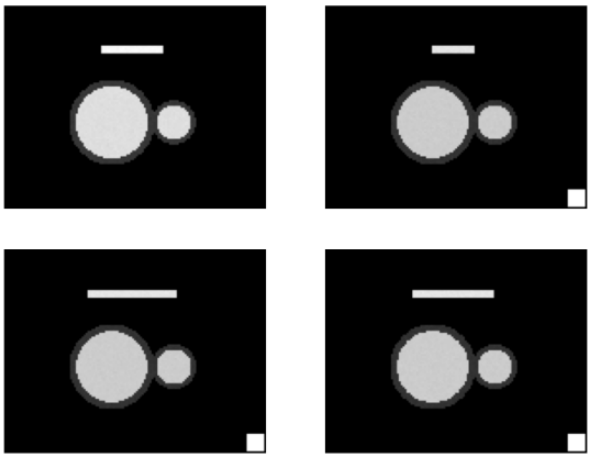

Figure 3.16: Exp 2: DCM MRIs

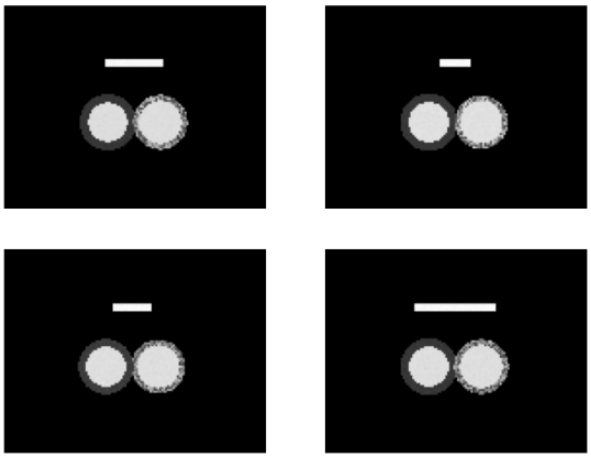

Figure 3.17: Exp 2: ARV MRIs

It was assumed that a similar effect size would be observed in this experiment. Thus, it was assumed that training three models with each loss function should be sufficient to achieve a sufficiently high power for an alpha level of 0.05. Once again, a more conservative alpha level should have been set for this experiment.

The automated workflow was then employed to train approximately thirty models using the heatmap loss function and the CCE loss function. A large fraction of these models were expected to be discarded because they would not have achieved a validation accuracy of over 95%. After filtering for models which achieved a validation accuracy of greater than 95%, 22 models trained with the heatmap loss function and 23 models trained with CCE remained.

Mean HiResCAM-heart overlaps were computed for models in both groups. Two Shapiro-Wilkes tests ($\alpha = 0.05$) were performed to test whether the distributions of these metrics within each group were normally distributed. The null hypothesis for a Shapiro-Wilkes test is that the data in a sample was drawn from a normal distribution. A p-value of 0.49 was returned from the Shapiro Wilkes test performed on the overlap metrics of models trained with CCE. Thus, we fail to reject the null hypothesis. However, a p-value of 0.008 was returned from the Shapiro Wilkes test carried out on the sample of overlaps from the models trained with the heatmap loss function. This supports the alternative hypothesis that the distribution of the overlap metrics were not drawn from a normal distribution and consequently are not normally distributed.

Consequently, a two-sided Mann-Whitney U-test ($\alpha = 0.05$) was performed to test for a statistically significant difference between the two group's HiResCAM-heart overlaps. The null hypothesis was that there is no systematic difference between the mean HiResCAM-heart overlaps of models trained with a heatmap loss function and the mean HiResCAM-heart overlaps of models trained with a CCE loss function.

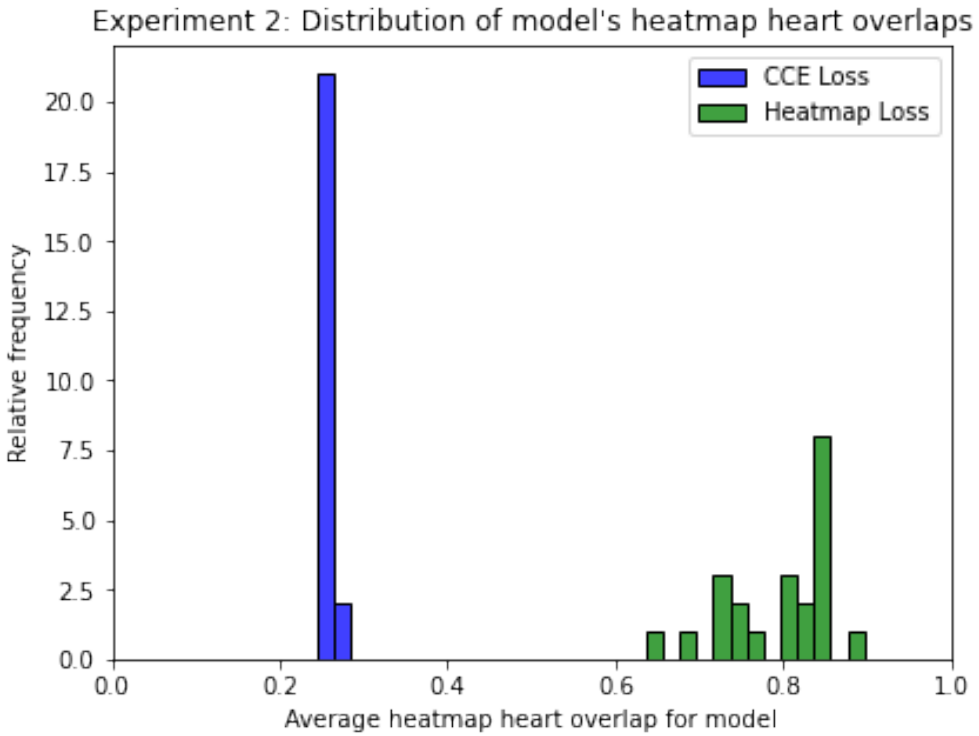


Figure 3.18: Experiment 2: Distribution of HiResCAM-heart overlaps

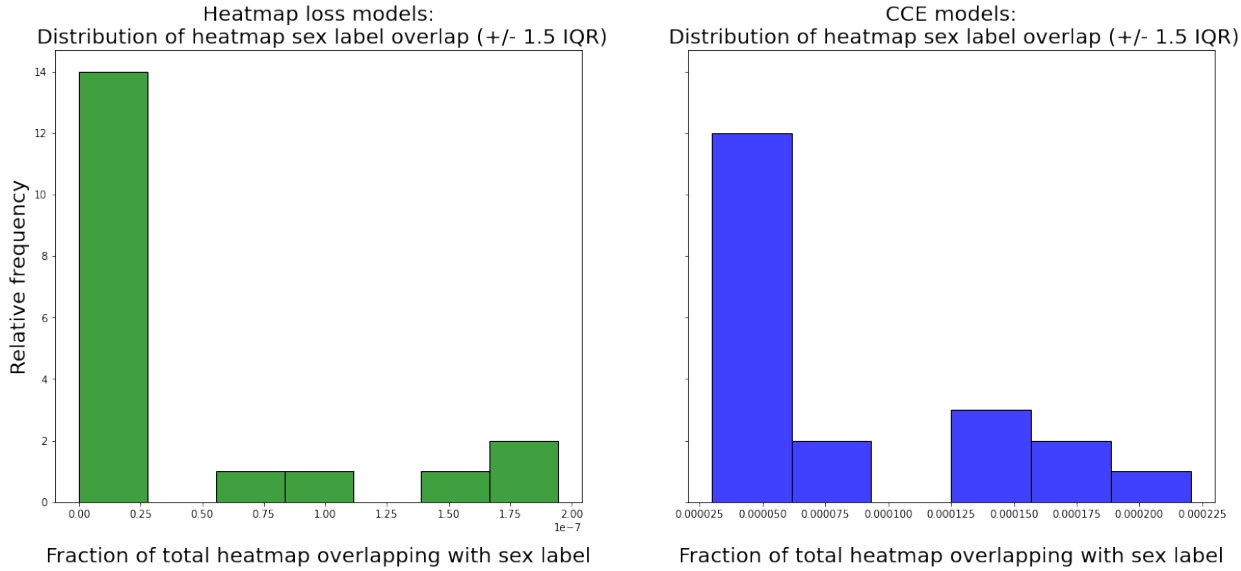


Figure 3.19: Distribution of HiResCAM-sex label overlaps (+/- 1.5IQR from median) for
a) Models trained with heatmap loss function on the left (Note, order of magnitude of overlaps $\approx 1 \times 10^{-7}$).
b) Models trained with CCE loss function on the right (Note, order of magnitude of overlaps $\approx 1 \times 10^{-4}$).

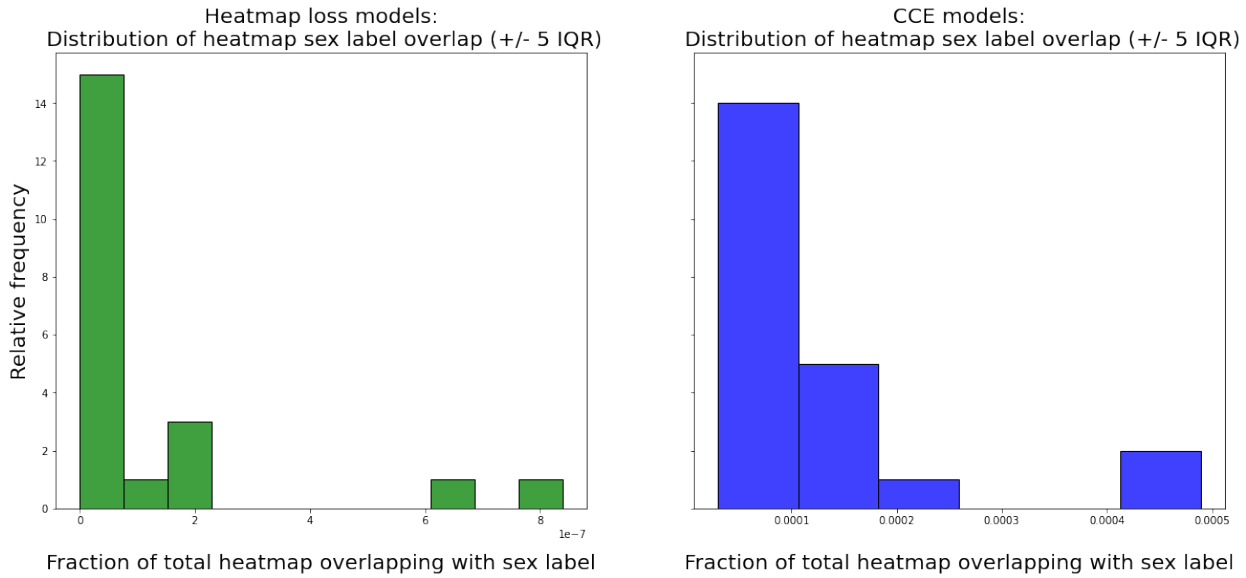


Figure 3.20: Distribution of HiResCAM-sex label overlaps (+/- 5IQR from median) for models trained with the heatmap loss function on the left and models trained with CCE on the right.

### 3.3.2 Results

This Mann-Whitney U-test returned a p-value $\approx 1 \times 10^{-8}$. This is less than the preset significance level of 0.05. Thus, the null hypothesis is rejected. This result supports the alternative hypothesis that the mean HiResCAM-heart overlaps of models trained with the heatmap loss function are systematically higher than the mean HiResCAM-heart overlaps of models trained with CCE. As all mean HiResCAM-heart overlaps for models trained with CCE were smaller than all mean HiResCAM-heart overlaps for models trained with the heatmap loss function, a common language effect size of 0.0 was calculated. The Cohen's d effect size was calculated to be -11.88, this is an extremely large effect size. Assuming the HiResCAM heatmaps provide truthful visualisations, these results suggest that models trained with the HiResCAM loss function appear to base their classifications to a higher extent off of information lying in regions within the heart than models trained with CCE.

A two-sided Mann-Whitney U-test ($\alpha = 0.05$) was performed to test for a statistically significant difference between the two group's HiResCAM-sex label overlaps. The null hypothesis was that there is no systematic difference between the mean HiResCAM-sex label overlaps of models trained with a heatmap loss function and the mean HiResCAM-sex label overlaps of models trained with a CCE loss function.

This Mann-Whitney U-test returned a p-value $\approx 1 \times 10^{-7}$. This is less than the preset significance level of 0.05. Thus, the null hypothesis is rejected. This result supports the alternative hypothesis that the mean HiResCAM-sex label overlaps of models trained with the heatmap loss function are systematically lower than the mean HiResCAM-sex label overlaps of models trained with CCE. As the majority of mean HiResCAM-sex label overlaps for models trained with CCE were larger than all mean HiResCAM-sex label overlaps for models trained with the heatmap loss function, a common language effect size of 0.95 was calculated. There existed some large outliers in both groups overlaps. The rank sum test is not vulnerable to outliers. However, the Cohen's d effect size is. Thus, I will quote the Cohen's d effect size after removing outliers. The Cohen's d effect size was calculated to be 1.84 after values +/- 1.5 IQRs above the medians were removed from the samples of HiResCAM-sex label overlaps and it was calculated to be 1.25 after values +/- 5 IQRs above the medians were removed from the samples of HiResCAM-sex label overlaps. These results suggest there was a large effect between the two groups HiResCAM-sex label overlaps.

Below are sample HiResCAM heatmaps produced by one randomly chosen model trained with a CCE loss function.

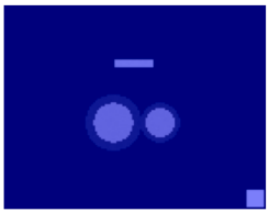

Figure 3.21: Exp 2: Normal MRI heatmaps

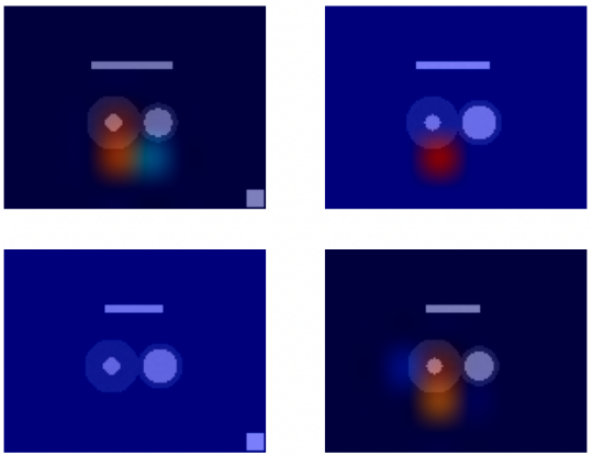

Figure 3.22: Exp 2: HCM MRIs heatmaps

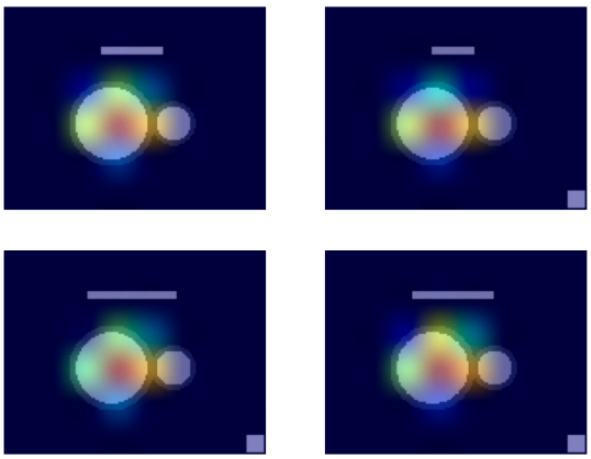

Figure 3.23: Exp 2: DCM MRIs heatmaps

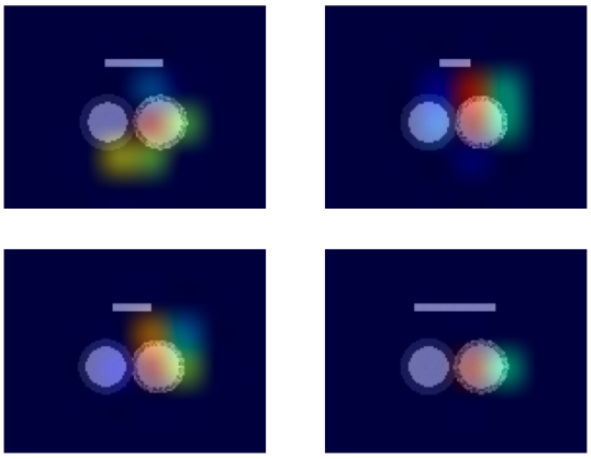

Figure 3.24: Exp 2: ARV MRIs heatmaps

Below are sample HiResCAM heatmaps produced by one randomly chosen model trained with a heatmap loss function.

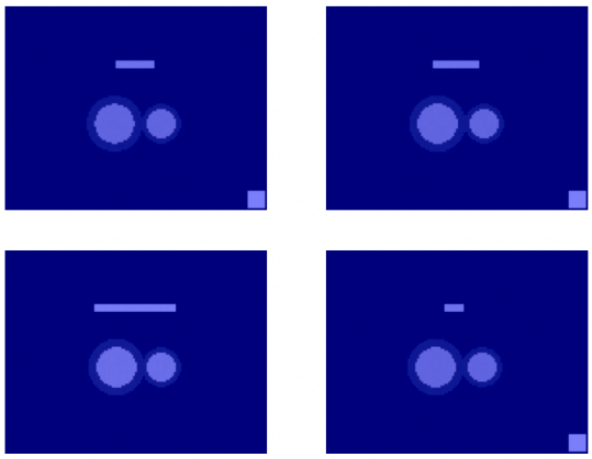

Figure 3.25: Exp 2: Normal MRIs heatmaps

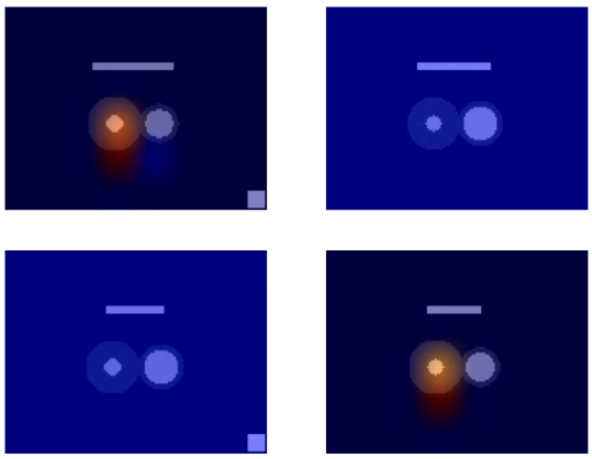

Figure 3.26: Exp 2: HCM MRIs heatmaps

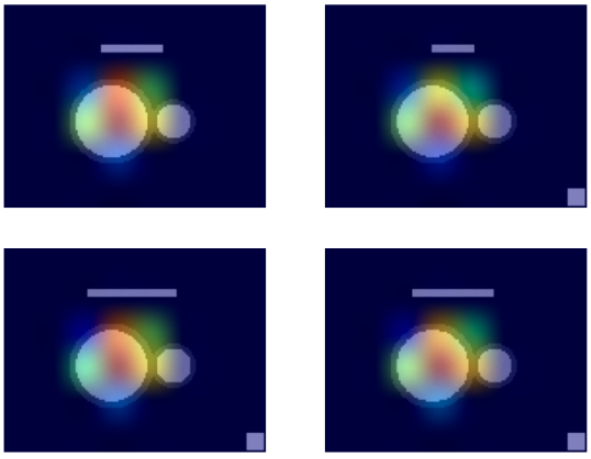

Figure 3.27: Exp 2: DCM MRIs heatmaps

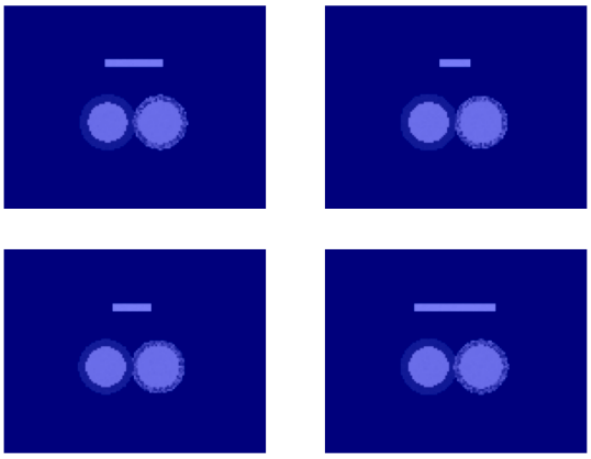

Figure 3.28: Exp 2: ARV MRIs heatmaps

Chapter 4

# Discussion & Conclusion

## 4.1 Discussion

### 4.1.1 Can models be trained to simultaneously achieve good classification accuracy as well as to look in the right regions of input images?

This study has demonstrated that models trained to minimise a heatmap loss function can also yield low losses as measured by traditional loss functions such as CCE. Moreover, the heatmap loss function was shown to successfully incentivise models to base their classifications off relevant portions of input images.

However, the classification task was evidently not very challenging, as many models from both groups achieved perfect test set classification accuracies. Thus, further research is needed to test whether models trained with a heatmap loss function on harder classification tasks can yield low losses as measured by traditional loss functions. This experiment could be carried out by training models on a dataset of synthetic MRIs which contain smaller systematic differences between the MRIs of the different disease classes.

It would also be worth testing whether the heatmap loss function can be used to dissuade a model from looking at features in an image that perfectly co-occur with images from a given disease class. As the heatmap loss function is a weighted sum of a CCE and a heatmap component, one could imagine a model sacrificing the heatmap component of the loss function for an increased classification accuracy, especially in harder classification tasks. This could be tested by superimposing class identifiers on images from any image dataset and training models using the heatmap loss function.

### 4.1.2 How could the heatmap loss function be altered or improved?

The biggest change one could make would involve changing the loss metrics or feature attribution methods used in the loss function. The feature attribution method HiResCAM was used in this study. This could be replaced by a myriad of other feature attribution methods such as GradCAM,

CAM or saliency maps to name but a few. Moreover, MSE was used to incentivise the model to correctly classify images, this could be replaced by a CCE loss for example.

One could imagine tuning the weights of the heatmap and CCE components of the loss function depending on the model's performance. For example, if the model was incurring large CCE losses, one could decrease the relative weighting of the heatmap component of the loss function. However, if CCE losses were sufficiently small, it may be beneficial to further incentivise modes to look at relevant portions of input images by increasing the relative weighting of the heatmap component of the loss function.

One could also alter how the feature attribution map component of the loss function is calculated. In this study, irrelevant portions of input images were highlighted and the model was penalised for looking within those regions when making classifications. Alternatively, one could highlight portions of input images that are necessary for a class classification and penalise a model for not looking within those regions when making classifications. For example, ARV is characterised by the replacement of the myocardium with fat tissue. Thus, a model could be penalised for not looking at the fatty myocardium of the MRIs of patients with ARV when making ARV classifications. Alternative metrics to measure overlaps between the irrelevant to relevant regions and the feature attribution masks could be developed. Inspiration could be taken from metrics used in segmentation problems. Different metrics to measure the overlap between predicted segmentation masks and ground truth segmentation masks have been developed there. Many of these metrics could likely be adapted for use in a heatmap loss function.

### 4.1.3 What concerns do you have about the heatmap loss function?

The heatmap loss function could facilitate the deployment of DL models that are not understood in high stake scenarios. Models trained with this loss function could conceivably gain undeserved trust due to the confirmation bias of humans and the convincing feature attribution maps produced by models trained with this loss function. Those assessing the suitability of a DL model for deployment would need to be familiar with the limitations of explanations provided by the different feature attribution methods. I believe interpretable models should be used in high-stake scenarios in place of non-interpretable models where possible.

More experiments need to be carried out to test whether minimising this heatmap loss function (and others) is desirable. Answering this question is akin to solving the outer alignment issue in

reinforcement learning problems. According to Hubinger *et al.* (2019), the outer alignment problem involves aligning an agent's reward function with the programmer's intentions for the agent. For example, would an agent that actually optimises for the reward function do what the programmer wants the agent to do. Based on countless examples of unforeseen outer alignment failures in reinforcement learning scenarios (Krakovna *et al.*, 2020), brainstorming possible shortcomings of a heatmap loss function is likely not sufficient, albeit worth doing nonetheless. To solve this outer alignment problem, models would need to be trained with this loss function to perform a variety of tasks on a variety of datasets and their behaviour analysed. Analysing these models with feature attribution methods other than the methods used in the loss function would likely be worth using here. The HiResCAM heatmaps produced by this randomly sampled model **??** that was trained with the heatmap loss function suggest that class classifications for normal and ARV are unresponsive to features anywhere in the input image. This is likely due to a flaw in the visualisation, as the heatmap overlays were not normalised in these visualisations. However, it could also be due to models learning to minimise their heatmap loss by being unresponsive to input features anywhere in an input image, not just outside the heart. Further investigation is needed.

I am unsure whether models should always be discouraged from basing classifications on non-causal features that disproportionately co-occur with a given class. By non-causal features, I mean features whose presence were not caused by the class of interest and do not determine an image being classified as an example of the class of interest. It seems feasible that in hard classification tasks, knowledge of certain non-causal co-occurrences could lead to an increased classification accuracy. I would argue that non-causal co-occurrences are already used in disease screening today. For example, men are not encouraged to get screened for breast cancer as the probability of an individual having breast cancer given that they are male is extremely low. However, when it comes to breast cancer occurrence, sex is a non-causal feature. By that I mean an individual's sex is not caused by the presence of breast cancer nor does an individual's sex cause breast cancer. In an ideal world, we could develop screening and diagnostic tools that do not depend upon such non-causal co-occurrences. In the case of breast cancer screening, ideally the only factor that influences breast cancer screening and diagnosis would be the presence or absence of cancerous cells in breast tissue.

Models that are dependent on non-causal features are especially problematic in scenarios where the co-occurrence of that non-causal feature and examples of the class of interest changes. This

might be less problematic for a disease classifier (assuming the disease does not change rapidly) as human evolution is a slow process. However, societal changes can happen quickly. For example, models trained 10 years ago that assign higher probabilities of classifying individuals as presidents if the individual is white and male, will likely perform extremely poorly in 10 years time.

### 4.1.4 Where would this approach be useful?

Heatmap loss functions could be useful in overcoming learned biases. Learned biases are a known issue in top performing ImageNet models (). Learned biases occur when a DL classifier recognises features that disproportionately co-occur with examples from certain classes. The classifier then learns to base class predictions on the presence or absence of these features. Identifying and eliminating all features that happen to disproportionately co-occur with specific classes of images in a dataset is likely an impossible task. Take the much simpler task of compiling a dataset where all genders and races are equally represented for all relevant classes. Compiling adequately large sets of sample images (to train a DL classifier) of builders, nurses, hurling players and presidents that are equally represented by people of all genders and races would be a very challenging task. In contrast it would likely be easier to use a heatmap loss function to disincentivise a model from looking at the skin and hair/footnoteHair length is meant to serve as a proxy variable for gender. Regardless of how good or bad a proxy variable hair length is, an individual's hair length should not be used to base job classifications. of individuals when making such classifications. Moreover, by incentivising a model to look at smaller portions of images when making classifications, the total number of features that happen to disproportionately co-occur with images from a given class that a model could undesirably detect would be reduced. This is worth mentioning, as the DL engineer training a model is likely not aware of the majority of the features that disproportionately co-occur with the different classes in the training dataset. This could be due to our limited ability to detect certain features when we are looking for them or our inability to track the degree to which features co-occur with images from a given class.

Moreover, by limiting the number of irrelevant features that a model could feasibly* discover and use when making classifications, it seems feasible that a model could be on smaller dataset sizes.

*The word feasibly was used, as in theory all features that co-occur with examples from a given class are learnable. However, with a heatmap loss function many of these features would likely not be learned as they lie in regions deemed irrelevant for classifications. Basing classifications on such features would incurr large losses.

### 4.1.5 Limitations of this approach

A major limitation of the heatmap loss function is that it inherits all the limitations of the feature attribution method used. Should a flawed feature attribution method be used in the loss function, it is likely that the model would not behave desirably. This would be the case even if high degrees of overlap between the feature attribution maps and the relevant/irrelevant portions of the input images were achieved. HiResCAM was integrated into the loss function in this study as it was believed to be superior to GradCAM. However, the paper on HiResCAM is a preprint and has not been cited by many researchers. Thus, it is possible this method is flawed. Should that be the case, models trained with this loss function are likely behaving no more desirably than models trained with a traditional loss function.

The additional training times required to train models with a heatmap loss function could be a limiting factor. Models trained using the heatmap loss function took approximately twice as long to train as models trained using CCE. For this implementation of the heatmap loss, only the heatmaps of the class of the image being classified were compared to regions outside the heart. Moreover, models trained with the heatmap loss functions were trained for approximately 3 times fewer epochs. These models were trained over fewer epochs as validation accuracies and losses stably converged quicker when using the heatmap loss function. Further research to test whether this is typical of models trained using heatmap loss functions is needed. It is conceivable that models trained with a heatmap loss function would stably converge in fewer epochs due to the reduced number of features that they could feasibly use to base classifications off of.

The differences in training durations observed in this study are not necessarily representative of the differences in training durations that one would observe should they implement their own heatmap loss function. Firstly, it is likely that the heatmap loss function which was implemented in this study was not implemented as efficiently as it could have been. No attempts were made to make the heatmap loss function more efficient after the first working implementation. Thus, the implementation of the CCE loss function provided by Keras was likely more efficient. Furthermore, precise training durations were not measured for these experiments. Precise training times could be calculated using the Python library 'time'. These durations could then be stored with the rest of the model information which was entered into a table in the database. Model training durations were tracked using this approach when models were trained on the ACDC dataset.

Applying this loss function would demand substantial amounts of additional labelling in some scenarios. However, it may be possible that semantic segmentation algorithms could be employed to automate this procedure. For example, segmentation algorithms could be used to create masks of regions of exposed skin and hair in images. These masks could then be used to disincentivise a model from looking at the skin and hair of individuals when making classifications related to a profession (i.e. builder/nurse/president classifications). Moreover, segmentation masks already exist for some medical imaging datasets. These could be adapted to train a model with a heatmap loss function. In this study, the segmentation masks which came with the ACDC dataset were used for this purpose, with regions outside of the heart deemed irrelevant for making heart disease classifications.

It is worth mentioning that the increased labelling requirements and training times could be minimised by selectively applying the heatmap component of the loss function to images from certain classes. For example, the heatmap loss component could be applied only for classes where bias in the training dataset had been identified or bias in a previously trained model. The MSE component of the loss function could be applied on all images.

If highlighting features in an image that we do not want the classifier to base classifications off of, these features need to be separable from the regions that we want the classifier to base classifications off of. For example, size and colour could be irrelevant when classifying images of a given class, however, it would be impossible to highlight these features as irrelevant without highlighting the entire object as irrelevant. In such scenarios, highlighting specific regions within the object that are essential to make classifications could be done. Then a different heatmap metric could be used to incentivise classifiers to look in these regions when making classifications. Alternatively, there may exist transforms that could be applied to images to make different features of the object separable from the object itself in a similar manner to how a Fourier transform applied to a time-domain audio recording makes frequencies in the recording separable.

Furthermore, the success of the heatmap loss function heavily depends on the accuracy of the image regions that are highlighted as relevant or irrelevant. By wrongly highlighting regions of images as irrelevant, the network would be unlikely to base decisions off of meaningful co-occurrences between features in those regions of images of the class of interest. For example, one could incorrectly deem the backgrounds of images irrelevant. This is likely an unfair assumption as the context from an image's background is often necessary to classify the object in the foreground. For example, when classifying diseases that originate from a specific organ one might wrongly assume that a model

should not look outside of that organ for signs of the disease. This could prevent the network from finding detectable systemic disease symptoms that are present outside of the primary organ associated with the disease.

Moreover, if the regions are formed based on existing knowledge of where a classifier should be looking when making classifications this would prevent the model from discovering undiscovered disease biomarkers that lie outside of these regions. However, by incentivising models to look within smaller regions of images when making classifications, the number of possible features which the network could learn to base classifications off of would decrease. This decreased feature set may make it easier for a human interpreter to discover meaningful biomarkers from.

## 4.2 Conclusion

This study has demonstrated that models trained to minimise a heatmap loss function can also yield low losses as measured by traditional loss functions such as CCE. This approach could be useful in overcoming the issue of learned biases and to train more skillful classifiers[†].

Several questions about the heatmap loss function have been posed such as;

- Its utility on harder classification tasks.
- Its ability to prevent a model from basing predictions off features that perfectly co-occur with a single disease class.
- Incorporating different feature attribution methods into the loss function.
- Tuning the weights of the heatmap and MSE components like any other hyperparameter in a hyperparameter search.
- Quantifying the additional training time requirements and potentially making the heatmap loss function implementation more efficient.
- Trying to identify models which achieve low heatmap losses but behave undesirably. (Answering the outer alignment question).
- Experimenting to see if semantic segmentation algorithms can be used to automate the irrelevant/relevant regions of images from a given class.

[†]Skillful classifiers are classifiers that look at information from the correct regions of images when classifying these images as examples of a given class.

- Test whether models trained with a heatmap loss function can be trained on smaller datasets and if these models can be trained over fewer epochs.

Attempting to use this method to train an ImageNet classifier would be an interesting challenge which would require several of the above issues be addressed.

The majority of the time spent on this project was trying to train a baseline DL classifier on the ACDC dataset. This time would likely have been better spent answering some of the proposed questions about this novel heatmap loss function. Given this experience, I believe new DL techniques should be fully developed on computer generated datasets like the one used in this study, or large high-quality datasets such as ImageNet. This would lead to faster research and development. Attempts could then be made to apply proven methods in harder computer vision sub-domains such as biomedical imaging.

# Bibliography


Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M. and Kim, B. (2018), 'Sanity checks for saliency maps', *Advances in neural information processing systems* **31**. 10, 11, 12

Ayan, E. and Ünver, H. M. (2019), Diagnosis of pneumonia from chest x-ray images using deep learning, *in* '2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)', Ieee, pp. 1–5. 17

Baltruschat, I. M., Nickisch, H., Grass, M., Knopp, T. and Saalbach, A. (2019), 'Comparison of deep learning approaches for multi-label chest x-ray classification', *Scientific reports* **9**(1), 1–10. 10

Baumgartner, C. F., Koch, L. M., Pollefeys, M. and Konukoglu, E. (2017*a*), An exploration of 2d and 3d deep learning techniques for cardiac mr image segmentation, *in* 'International Workshop on Statistical Atlases and Computational Models of the Heart', Springer, pp. 111–119. 14

Baumgartner, C. F., Koch, L. M., Pollefeys, M. and Konukoglu, E. (2017*b*), 'An exploration of 2D and 3D deep learning techniques for cardiac MR image segmentation', https://github.com/baumgach/acdc_segmenter. 14, 16

Belton, N., Welaratne, I., Dahlan, A., Hearne, R. T., Hagos, M. T., Lawlor, A. and Curran, K. M. (2021), Optimising knee injury detection with spatial attention and validating localisation ability, *in* 'Annual Conference on Medical Image Understanding and Analysis', Springer, pp. 71–86. 3, 10

Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.-A., Cetin, I., Lekadir, K., Camara, O., Ballester, M. A. G. *et al.* (2018), 'Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved?', *IEEE transactions on medical imaging* **37**(11), 2514–2525. 2, 18

Betechuoh, B. L., Marwala, T. and Tettey, T. (2006), 'Autoencoder networks for hiv classification', *Current Science* pp. 1467–1473. 16

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. *et al.* (2020), 'Language models are few-shot learners', *Advances in neural information processing systems* **33**, 1877–1901. 1

Burduja, M., Ionescu, R. T. and Verga, N. (2020), 'Accurate and efficient intracranial hemorrhage detection and subtype classification in 3d ct scans with convolutional and long short-term memory neural networks', *Sensors* **20**(19), 5611. 2, 3

Cetin, I., Sanroma, G., Petersen, S. E., Napel, S., Camara, O., Ballester, M.-A. G. and Lekadir, K. (2017), A radiomics approach to computer-aided diagnosis with cardiac cine-mri, *in* 'International workshop on statistical atlases and computational models of the heart', Springer, pp. 82–90. 7, 18

Chen, S., Ma, K. and Zheng, Y. (2019), 'Med3d: Transfer learning for 3d medical image analysis', *arXiv preprint arXiv:1904.00625* . 16

Cho, H., Kim, Y., Lee, E., Choi, D., Lee, Y. and Rhee, W. (2020), 'Basic enhancement strategies when using bayesian optimization for hyperparameter tuning of deep neural networks', *IEEE Access* **8**, 52588–52608. 8

Cho, K., Roh, J.-h., Kim, Y. and Cho, S. (2019), A performance comparison of loss functions, *in* '2019 International Conference on Information and Communication Technology Convergence (ICTC)', IEEE, pp. 1146–1151. 5, 9

Clinic, M. (2020), 'Hypertrophic cardiomyopathy', https://www.mayoclinic.org/diseases-conditions/hypertrophic-cardiomyopathy/symptoms-causes/syc-20350198. 22

Clinic, M. (2021), 'Dilated cardiomyopathy', https://www.mayoclinic.org/diseases-conditions/dilated-cardiomyopathy/symptoms-causes/syc-20353149. 22

Dice, L. R. (1945), 'Measures of the amount of ecologic association between species', *Ecology* **26**(3), 297–302. 3

Dong, X., Taylor, C. J. and Cootes, T. F. (2020), 'Defect detection and classification by training a generic convolutional neural network encoder', *IEEE Transactions on Signal Processing* **68**, 6055–6069. 16

Draelos, R. L. and Carin, L. (2020), 'Hirescam: Faithful location representation in visual attention for explainable 3d medical image classification', *arXiv preprint arXiv:2011.08891* . 10, 11

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A. W., Firat, O. *et al.* (2021), 'Glam: Efficient scaling of language models with mixture-of-experts', *arXiv preprint arXiv:2112.06905* . 1

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. and Thrun, S. (2017), 'Dermatologist-level classification of skin cancer with deep neural networks', *nature* **542**(7639), 115–118. 1

Feger, J. and Radiopaedia (2021), 'Hypertrophic cardiomyopathy', https://radiopaedia.org/articles/hypertrophic-cardiomyopathy?lang=gb. 21

Goodfellow, I., Bengio, Y. and Courville, A. (2016), 'Deep learning (adaptive computation and machine learning series), illustrated ed'. 7

Goodman, B. and Flaxman, S. (2017), 'European union regulations on algorithmic decision-making and a "right to explanation"', *AI magazine* **38**(3), 50–57. 1, 9

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J. *et al.* (2016), 'Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs', *Jama* **316**(22), 2402–2410. 1, 7

Han, X. (2017), 'Automatic liver lesion segmentation using a deep convolutional neural network method', *arXiv preprint arXiv:1704.07239* . 16

He, K., Zhang, X., Ren, S. and Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 770–778. 8

Ho, N. and Kim, Y.-C. (2021), 'Evaluation of transfer learning in deep convolutional neural network models for cardiac short axis slice classification', *Scientific reports* **11**(1), 1–11. 17

Hubinger, E., van Merwijk, C., Mikulik, V., Skalse, J. and Garrabrant, S. (2019), 'Risks from learned optimization in advanced machine learning systems', *arXiv preprint arXiv:1906.01820* . 36

Isensee, F., Jaeger, P. F., Full, P. M., Wolf, I., Engelhardt, S. and Maier-Hein, K. H. (2017), Automatic cardiac disease assessment on cine-mri via time-series segmentation and domain specific features, *in* 'International workshop on statistical atlases and computational models of the heart', Springer, pp. 120–129. 7, 14, 18

Khamparia, A., Singh, P. K., Rani, P., Samanta, D., Khanna, A. and Bhushan, B. (2021), 'An internet of health things-driven deep learning framework for detection and classification of skin cancer using transfer learning', *Transactions on Emerging Telecommunications Technologies* **32**(7), e3963. 17

Khened, M., Alex, V. and Krishnamurthi, G. (2017), Densely connected fully convolutional network for short-axis cardiac cine mr image segmentation and heart diagnosis using random forest, *in* 'International Workshop on Statistical Atlases and Computational Models of the Heart', Springer, pp. 140–151. 7, 14, 18

Kim, B., Han, M., Shim, H. and Baek, J. (2019), 'A performance comparison of convolutional neural network-based image denoising methods: The effect of loss functions on low-dose ct images', *Medical physics* **46**(9), 3906–3923. 4, 5, 9

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D. and Kim, B. (2019), The (un) reliability of saliency methods, *in* 'Explainable AI: Interpreting, Explaining and Visualizing Deep Learning', Springer, pp. 267–280. 11, 12

Krakovna, V., Uesato, J. and Mikulik, V. (2020), 'Specification gaming: the flip side of ai ingenuity', https://www.deepmind.com/blog/specification-gaming-the-flip-side-of-ai-ingenuity. 36

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), 'Imagenet classification with deep convolutional neural networks', *Advances in neural information processing systems* **25**. 7, 8, 17

Lai, M. (2015), 'Deep learning for medical image segmentation', *arXiv preprint arXiv:1505.02000* . 16, 19

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R. and Soatto, S. (2020), 'Rethinking the hyperparameters for fine-tuning', *arXiv preprint arXiv:2002.11770* . 7

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B. and Sánchez, C. I. (2017), 'A survey on deep learning in medical image analysis', *Medical image analysis* **42**, 60–88. 1

Liu, X., Kayser, M., Kushner, S., Tiemeier, H., Rivadeneira, F., Jaddoe, V., Niessen, W., Wolvius, E. and Roshchupkin, G. (2021), 'Association between prenatal alcohol exposure and children's facial shape. a prospective population-based cohort study', *medRxiv* . 16

Longoni, C., Bonezzi, A. and Morewedge, C. K. (2019), 'Resistance to medical artificial intelligence', *Journal of Consumer Research* **46**(4), 629–650. 1

Lujan-Moreno, G. A., Howard, P. R., Rojas, O. G. and Montgomery, D. C. (2018), 'Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study', *Expert Systems with Applications* **109**, 195–205. 8

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2013), 'Playing atari with deep reinforcement learning', *arXiv preprint arXiv:1312.5602* . 4

Neary, P. (2018), Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning, *in* '2018 IEEE International Conference on Cognitive Computing (ICCC)', pp. 73–77. 8

Ng, A. (2017), 'Transfer learning (c3w2l07)', https://youtu.be/yofjFQddwHE?t=598. 16

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M. and Carter, S. (2020), 'Zoom in: An introduction to circuits', *Distill* **5**(3), e00024–001. 16

Olah, C., Mordvintsev, A. and Schubert, L. (2017), 'Feature visualization', *Distill* **2**(11), e7. 9

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L. *et al.* (2019), 'Kerastuner', https://github.com/keras-team/keras-tuner. 8

OtgontuyaE (2009), 'Obesity', *Malaysian journal of nutrition* . 21

Pasa, F., Golkov, V., Pfeiffer, F., Cremers, D. and Pfeiffer, D. (2019), 'Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization', *Scientific reports* **9**(1), 1–9. 10

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S. *et al.* (2021), 'Scaling language models: Methods, analysis & insights from training gopher', *arXiv preprint arXiv:2112.11446* . 1

Rijsdijk, J., Wu, L., Perin, G. and Picek, S. (2021), 'Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis', *IACR Transactions on Cryptographic Hardware and Embedded Systems* pp. 677–707. 8

Ronneberger, O., Fischer, P. and Brox, T. (2015), U-net: Convolutional networks for biomedical image segmentation, *in* 'International Conference on Medical image computing and computer-assisted intervention', Springer, pp. 234–241. 14, 18, 19

Rosenbush, S. (2022), 'Big tech is spending billions on ai research. investors should keep an eye out', https://www.wsj.com/articles/big-tech-is-spending-billions-on-ai-research-investors-should-keep-an-eye-out-11646740800. 9

Rudin, C. (2019), 'Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead', *Nature Machine Intelligence* **1**(5), 206–215. 9

Saber, M. and Radiopaedia (2021), 'Arrhythmogenic right ventricular cardiomyopathy', https://radiopaedia.org/articles/arrhythmogenic-right-ventricular-cardiomyopathy?lang=gb. 21

Schwall, M., Daniel, T., Victor, T., Favaro, F. and Hohnhold, H. (2020), 'Waymo public road safety performance data', *arXiv preprint arXiv:2011.00038* . 1

Seraphim, A., Knott, K. D., Augusto, J., Bhuva, A. N., Manisty, C. and Moon, J. C. (2020), 'Quantitative cardiac mri', *Journal of Magnetic Resonance Imaging* **51**(3), 693–711. 2

Shankar, K., Zhang, Y., Liu, Y., Wu, L. and Chen, C.-H. (2020), 'Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification', *IEEE Access* **8**, 118164–118173. 8

Sharir, O., Peleg, B. and Shoham, Y. (2020), 'The cost of training nlp models: A concise overview', *arXiv preprint arXiv:2004.08900* . 9

Shorten, C. and Khoshgoftaar, T. M. (2019), 'A survey on image data augmentation for deep learning', *Journal of big data* **6**(1), 1–48. 1, 13

Simonyan, K., Vedaldi, A. and Zisserman, A. (2013), 'Deep inside convolutional networks: Visualising image classification models and saliency maps', *arXiv preprint arXiv:1312.6034* . 11

Simonyan, K. and Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* . 7, 8, 17

Soulami, K. B., Kaabouch, N., Saidi, M. N. and Tamtaoui, A. (2021), 'Breast cancer: one-stage automated detection, segmentation, and classification of digital mammograms using unet model based-semantic segmentation', *Biomedical Signal Processing and Control* **66**, 102481. 16

Staelin, C. (2003), 'Parameter selection for support vector machines', *Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1* **1**. 8

Sugata, T. and Yang, C. (2017), Leaf app: Leaf recognition with deep convolutional neural networks, *in* 'IOP Conference Series: Materials Science and Engineering', Vol. 273, IOP Publishing, p. 012004. 18

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015), Going deeper with convolutions, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1–9. 8

UMichigan, w. (n.d.), 'Hypertrophic cardiomyopathy', https://www.umcvc.org/conditions-treatments/arrhythmogenic-cardiomyopathy-ac-arvc. 22

Wang, S.-H., Sun, J., Phillips, P., Zhao, G. and Zhang, Y.-D. (2018), 'Polarimetric synthetic aperture radar image segmentation by convolutional neural network using graphical processing units', *Journal of Real-Time Image Processing* **15**(3), 631–642. 8

Weerakkody, Y. and Radiopaedia (2021), 'Dilated cardiomyopathy', https://radiopaedia.org/articles/dilated-cardiomyopathy. 21

Weiss, K., Khoshgoftaar, T. M. and Wang, D. (2016), 'A survey of transfer learning', *Journal of Big data* **3**(1), 1–40. 15

WHO (2021), 'Cardiovascular diseases (cvds)', https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). 2

Wolterink, J. M., Leiner, T., Viergever, M. A. and Išgum, I. (2017), Automatic segmentation and disease classification using cardiac cine mr images, *in* 'International Workshop on Statistical Atlases and Computational Models of the Heart', Springer, pp. 101–110. 7, 14

Yessou, H., Sumbul, G. and Demir, B. (2020), A comparative study of deep learning loss functions for multi-label remote sensing image classification, *in* 'IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium', IEEE, pp. 1349–1352. 5, 9

Ying, X. (2019), An overview of overfitting and its solutions, *in* 'Journal of Physics: Conference Series', Vol. 1168, IOP Publishing, p. 022022. 13

Yu, Q., Xie, L., Wang, Y., Zhou, Y., Fishman, E. K. and Yuille, A. L. (2018), Recurrent saliency transformation network: Incorporating multi-stage visual cues for small organ segmentation, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 8280–8289. 16

Zheng, Q., Delingette, H. and Ayache, N. (2019), 'Explainable cardiac pathology classification on cine mri with motion characterization by semi-supervised learning of apparent flow', *Medical image analysis* **56**, 80–95. 7