Interpretable AI For Protein Expression Images and Associated Clinical Metadata

Satish Pilla
School of Computing
Newcastle University
Newcastle, UK
S.Pilla2@newcastle.ac.uk

Atif Khan
School of Computing
Newcastle University
Newcastle, UK
a.khan21@newcastle.ac.uk

Stephen McGough
School of Computing
Newcastle University
Newcastle, UK
stephen.mcgough@newcastle.ac.uk

September 6, 2022

Abstract

Mitochondrial diseases are among the most common inheritable diseases, with a conservative estimated prevalence of roughly 1:5,000 [1]. Diagnosis and understanding the different mitochondrial diseases are extremely difficult because they have a wide range of symptoms in each patient and affect different organs and tissues of the body [2]. However, recent studies show deep learning algorithms with interpretability and explainability, especially Convolutional Neural Networks (CNN), can help us automatically diagnose and evaluate different diseases by detecting the different patterns in the images. In this research, we investigated different deep learning models such as VGG16 and ResNet50 with transfer learning techniques for predicting mitochondrial diseases using protein expression images (pseudo images of 10 proteins generated by image mass cytometer) and then evaluated the models using different interpretability and explainability methods, for understanding the underlying pathology of the mitochondrial diseases. Of the 2 CNN architecture configurations, VGG16 produced the best results. It achieved an accuracy of 95.50%, a recall (sensitivity) of 94.50%, and an F1 score of 95.25%. For evaluating the model, we used the Grad-CAM (gradient-weighted class activation mapping) technique for explainability and the neural disentanglement (concept whitening) technique for interpretability. When the concept whitening technique was introduced in the model, the accuracy obtained was only 67%. The unavailability of metadata concepts was a major factor in reducing the accuracy of the concept whitening model. Whereas, the Grad-CAM technique showed great results by highlighting the important region of the images. The results suggest that computer-aided diagnosis can be a reliable method for predicting mitochondrial diseases. This is the first study to attempt to predict and understand the pathology of mitochondrial diseases from protein expression images using a deep learning approach.

***K*eywords** Transfer Learning, Interpretability, Explainability, Grad-CAM, Concept Whitening

1 Introduction

Mitochondrial diseases are a group of inherited diseases caused by the dysfunction of mitochondria, a type of cellular organelle [3]. The most common cause of many mitochondrial diseases is impaired oxidative phosphorylation, which results in reduced cellular energy adenosine triphosphate (ATP) production [4]. Mitochondrial diseases are currently incurable, affecting 1 in 5000 [1] and are highly heterogeneous, and adversely affect high-energy demanding cells e.g., skeletal muscle cells and neurons [5]. Mitochondrial diseases are extremely difficult to diagnose and much more difficult is to understand the pathology because they have a wide range of symptoms in each patient and affect different organs and tissues of the body [2]. Many human

diseases, including cancer, obesity, neurodegenerative diseases, cardiovascular diseases, and neurometabolic diseases, are associated with mitochondrial dysfunction [6], [7].

There are several methods to identify whether someone has a mitochondrial disease or not, scientists at the Wellcome Centre Mitochondrial Research [WCMR] use observational data from a variety of patient groups and controls to profile mitochondrial disease and uncover patterns in data. Genetic diagnostic tests, genetic or biochemical tests of diseased tissues such as muscle or liver, and screening for certain mitochondrial biomarkers in spinal fluid, urine, and blood are some additional techniques for diagnosing mitochondrial diseases [8], [9]. Out of all the different techniques for figuring out the underlying pathology of mitochondrial disease, the study of skeletal muscle research is one of the crucial methods that helps a little bit in the diagnosis of mitochondrial malfunction and in determining the underlying pathology of mitochondrial disease [10]. However, the above-mentioned processes are very time-consuming, require clinical specialists with a lot of experience in the field, and even after conducting many different clinical examinations, clinicians, and scientists have never been able to correctly diagnose mitochondrial diseases due to the wide range of symptoms in each patient [4]. Present, there is no treatment or cure for mitochondrial diseases, the only choice for treatment is for medical professionals to help patients with a few symptoms [9]. Clinicians and scientists want to identify and understand the pathology of mitochondrial disease, in the hope that pathology will lead them to effective treatments and cures for people with mitochondrial disease. Clinicians and scientists are finding it hard to understand the pathology, and there is an immediate need for a new solution to identify and understand the underlying pathology of mitochondrial diseases.

As far as we are aware, computer-aided diagnosis has never been used for the diagnosis or in understanding the mitochondrial disease pathology. The traditional machine/deep learning algorithms provide powerful classification and decision-making techniques for image analysis, but most algorithms are black boxes to humans [11]. But the healthcare industry and our problem require AI models to be interpretable and explainable due to the ethical issue of transparency and lack of trust in the AI system's black-box operation [11], [12]. Hence, with the rise of deep learning-based techniques, specifically in high-stakes decision-making areas like medical image analysis, there is an increasing need for explainability and interpretability methods [12]. Various explainability and interpretability techniques have been introduced in recent years, such as saliency maps [13], Grad-CAM (gradient-weighted class activation mapping) [14], and neural disentanglement [15] to evaluate the model performance and to understand, which features are leading to the prediction. Since our main problem is not to classify the images but rather to understand the features that are causing mitochondrial dysfunction, we can apply these explainability and interpretability techniques to identify and understand the underlying pathology of mitochondrial diseases.

In this study, we used protein expression images obtained by imaging mass cytometry techniques [10], [16] to train and build a deep learning model using transfer learning to identify various mitochondrial disorders, and then we are applying interpretability and explainability approaches to look at the basis of this prediction, for the discovery of the disease's underlying pathology. We build a VGG16 model, which was able to distinguish between control (tissue biopsy images from persons who are not suffering from mitochondrial diseases) and patient (tissue biopsy images from persons suffering from mitochondrial diseases) with an accuracy of 95.25%. Then, to increase trust and transparency, we evaluated the model using Grad-CAM and concept whitening techniques. Grad-CAM highlighted the important regions of the images, and with the concept whitening technique, we were able to obtain an accuracy of only 67%. The unavailability of metadata concepts was the key problem that impacted the concept whitening model's accuracy. The results suggest that deep learning with explainability and interpretability can be one of the reliable methods for predicting mitochondrial diseases from protein expression images.

1.1 Research Aim

The project's main aim is to build deep learning models using transfer learning to predict mitochondrial diseases and use existing machine learning interpretability and explainability AI approaches for computer vision, such as saliency maps, Grad-CAM, and Neural Disentanglement (concept whitening) to help us understand the artifacts or features that led to the prediction of mitochondrial disease from protein expression images.

1.2 Research Outline

The rest of the research paper is structured as follows. In section 2, background and related work, we discuss an overview of explainability and interpretability and existing explainability and interpretable AI/ML approaches for medical image analysis published in peer-reviewed articles and journals. Section 3, methodology

describes the dataset and research methods used to determine the aims and objectives of the research. Section 4, results and discussion present a critical evaluation of the research findings. Section 5, conclusions and recommendations summarise the main findings of the study and provide recommendations for further research based on these conclusions.

2 Background and Related Work

2.1 Overview of Explainability and Interpretability

When talking about machine learning and artificial intelligence, the term explainability and interpretability are frequently used interchangeably. Despite how similar they appear, it's crucial to recognise the distinctions. In the paper [17], Rudin clearly distinguishes between interpretable and explainable AI, while explainable AI attempts to offer post-hoc explanations for currently used black-box models, which are proprietary or incomprehensible to humans, interpretable AI focuses on building models that are intrinsically interpretable from the outset. Lipton [18] emphasizes the distinction between the questions that each family of techniques seeks to answer, interpretability asks, "How does the model work?" while explanation methods seek to respond, "What else can the model tell me?" Since there is no general agreement on what either interpretability or explainability means, researchers have looked to various desiderata to motivate different types of techniques [19].

Since our main aim is to understand the pathology of mitochondrial disease, we are applying different interpretability and explainability method. So, in the research, we used Grad-CAM for explainability and concept whitening for interpretability. Most of the approaches mentioned below in sections 2.2 and 2.3 are quite similar with very little difference. For example, most of the explainability approaches use the back-propagation method to highlight the important regions of the image. Any CNN model initially captures the low-level features like edges and corners, but when the model is trained multiple times and passed through different layers it learns the important features related to the class to which it belongs. These important feature scores related to the particular class are stored in the last convolution layers, and most of the explainability methods such as CAM, Grad-CAM, and the deconvolution approach use these feature scores present in the last layers to highlight the important regions of the images. Similarly, most of the interpretable approaches mentioned below try to find out the prototype parts which are very closely related to the class to which it belongs. So, in sections 2.2 and 2.3, one can find full details of different explainable and interpretable AI techniques used in medical image analysis.

2.2 Explainable AI in Medical Image Analysis

In medical image analysis, visual interpretation (also known as saliency mapping) is the most commonly used type of explainable AI. The features of a picture that affect a decision are highlighted by saliency maps. Most saliency mapping techniques use back-propagation based methods. This section provides an overview of explainable AI algorithms used in medical image analysis [20].

(Guided) Backpropagation and Deconvolution: The earliest saliency mapping methods highlighted the pixels that had the most influence on the results of the analysis. Examples include deconvolution [21], displaying partial derivatives of the output at the pixel level [22], and guided backpropagation [23]. These methods have been used in the analysis of medical imaging data. For example, Vos et al. [24] assessed coronary artery calcium for each slice of a heart or chest computed tomography (CT) image and applied deconvolution to reveal where in the slice the decision was made.

Class Activation Mapping (CAM): Zhou et al. [25] introduced a new technique, Class Activation Map (CAM), for the visual inspection of deep learning models. On the final convolutional feature maps, they applied global average pooling to replace the fully connected layers at the end of the CNN, then a weighted linear sum was employed by the class activation map to reflect the presence of visual patterns (recorded by the filters) at different spatial locations. To detect acute intracerebral haemorrhage, Lee et al. [26] constructed a CAM from the output of an ensemble of four CNNs: VGG-16, Inception-V3, and ResNet-50.

Gradient-weighted Class Activation Mapping (Grad-CAM): Grad-CAM is a generalisation of CAM and is introduced by Selvaraju et al. [14]. In contrast to CAM, Grad-CAM does not require global average pooling and can provide a post-hoc local interpretation using any type of CNN. Selvaraju et al. also introduced Guided Grad-CAM, which is an element-wise multiplication of Grad-CAM and guided back-propagation. Ji et al. [27] constructed Grad-CAM and demonstrated the use of a classifier to determine metastatic tissue in

histological lymph node sections. Kowsari et al. [28] deployed Grad-CAM technology to identify small bowel enteropathy on histology.

Layer-wise Relevance Propagation (LRP): In 2015, Bach et al. [29] introduced this technique (LRP). LRP back propagates the neural network's output, for example, a classification score between 0 and 1 iteratively throughout the network. It assigns a relevance score to each of the input neurons from the preceding layers in each iteration. According to the conservation law, the sum of these distributed relevance scores must match that of their source neuron. LRP has been applied to image analysis in medicine. For instance, Bohle et al. [30] employed LRP to locate Alzheimer's disease-causing areas in brain MR images. They contrasted the saliency maps produced by guided backpropagation with LRP and discovered that LRP was more accurate in detecting areas known to have Alzheimer's disease.

2.3 Interpretable AI in Medical Image Analysis

This Looks Like That: Interpretable Image Recognition Using Deep Learning: Chaofan Chen et al. [31] introduced the deep network architecture ProtoPNet, a new concept for interpretable deep learning that basically finds some prototype parts of an image to classify itself, making the classification process interpretable. The algorithm's reasoning was qualitatively comparable to how ornithologists, doctors, geologists, architects, and other professionals would instruct humans on how to do difficult image classification jobs. They demonstrated the method on the two datasets CUB-200-201 and the Stanford Cars dataset. The same approach can be extended to other datasets as it only uses image-level labels for training.

Interpretable Image Recognition with Hierarchical Prototypes: Hase et al. [32] introduced a new model that uses hierarchically arranged prototypes (HPnet) to classify objects at every level in a predefined taxonomy. Consequently, one can discover different reasons for the forecast of an image at each level of the taxonomy. Using a subset of the ImageNet dataset, the authors assessed the model's performance against a black-box model for two tasks: 1) classifying data into known classes, and 2) classifying data into classes that had not yet been found at the correct level of the taxonomy.

A case-based Interpretable Deep Learning Model for Digital Mammography Mass Lesion Classification (IAIA-BL): Barnett et al. [33] introduced another new framework the Interpretable Artificial Intelligence Algorithm for Breast Lesions (IAIABL). The framework not only classifies a lesion as benign or malignant, but it also intends to mimic the radiologists' thought processes in identifying clinically significant semantic components of each image. They show that even with a small number of photos, the algorithm can combine data with whole-image labeling and data with pixel-by-pixel annotations, improving accuracy and interpretability.

Concept Whitening for Interpretable Image Recognition: In 2020, Z. Chen, Bei, and Rudin [15] introduces a new method called concept whitening (CW) for altering a network layer to better comprehend the computation leading up to that layer, such that the axes of the latent space are aligned with known concepts of interest when a concept whitening module is added to a CNN. Concept samples can be manually selected to feed the CNN network. They demonstrated that CW can provide a clearer picture of how the network progressively learns concepts across layers through the experiment. The CW layer normalises and decorrelates the latent space and it is an alternative to a batch normalization layer. CW can be used at any layer of the network without altering expected performance. They demonstrated the method on the ISIC dataset to classify the skin lesion as benign or malignant by selecting the concept of age and lesion mass size. The original architecture explaining the concept whitening can be found in the appendix section under section methodology.

2.4 Advantages of Using Interpretability and Explainability in the Medical Field

Below we summarise, based on [17], [18], [34], [35] a few advantages that researchers working in the healthcare industry can gain from the interpretability and explainability of AI techniques:

- Increased transparency: The interpretability and explainability approaches increase transparency in how AI systems operate and can develop higher levels of trust by describing how an AI system arrived at a certain choice [18], [34].
- Result tracing: Using the explanations produced by interpretability and explainability approaches, it is possible to identify the variables that the AI system took into account when making a prediction [17], [35].

- Model Development: To make predictions, AI systems learn from data. Sometimes, incorrect learning rules can result in incorrect predictions. Explanations produced by interpretability and explainability approaches can help understand learned rules, allowing for the identification of flaws and improving models [17], [18], [34].
- Privacy: Privacy can be a concern when systems rely on sensitive personal data. Interpretability and explainability approaches can help us understand if user privacy is preserved or not [18], [34].
- Understanding complex structures: With the help of interpretability and explainability one can easily understand many complex natural phenomena such as protein folding or in our case trying to understand the pathology of mitochondrial diseases [36].

3 Methodology

The experiment was conducted in two phases, In the first phase, we built a pipeline for the classification of two sub-experiments. In the first sub-experiment, we are using single channel protein images (SDHA) to classify the images into control and patient. In the second sub-experiment, at first, we are stacking all the protein expression images into one single image (multi-channel image) for different controls and patients, and then using the pipeline we are classifying the images as control and patient. For these two sub-experiments, we adapted two deep neural networks, VGG16 and ResNet50, for classification and comparison. VGG16 and ResNet50 are state-of-the-art feature extractors that are trained on the ImageNet dataset. The main components and architecture for the classification of mitochondrial diseases are shown in Figure 1.

The classification pipeline for both experiments is divided into two main steps: the first step is to preprocess the images, dealing with data augmentation, data splitting, image stacking, and normalisation. For the data augmentation, we used rescaling, random rotation, vertical and horizontal flipping, and zooming for the training dataset, and for the test & validation dataset, we're just rescaling the dataset. For data splitting we used patchify, and for stacking different protein images we first converted the images into a numpy array and stacked all 10 channels using numpy concatenation method. Full details of data augmentation, data splitting, and image stacking can be found in the appendix under the methodology section. The second step, classification, involves applying transfer learning to a previously trained model such as VGG16 and ResNet-50 for making predictions.

Since our main goal is to know, which features are leading to the prediction of control and patient, in the second phase we are applying the interpretability (Concept whitening) and explainability (Grad-CAM) methods to the classification models to understand the features used to predict mitochondrial diseases. Architectures explaining these techniques can be found in Figures 6 & 7.

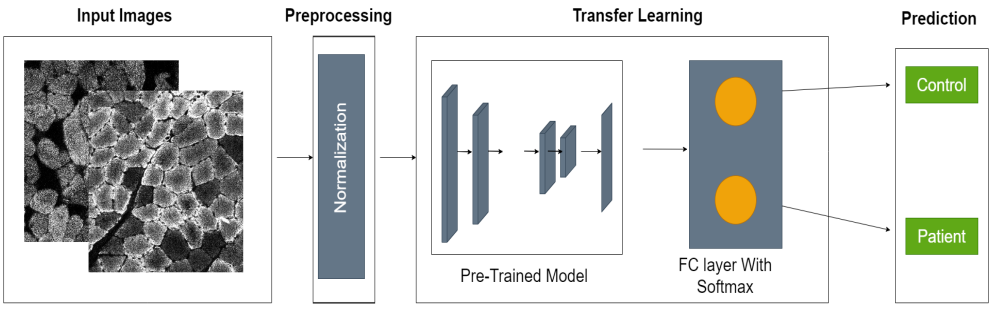

Figure 1: Overview of the methodology transfer learning for mitochondrial disease diagnosis

3.1 Dataset

In this study, we used protein expression images obtained from image mass cytometry (IMC). The two main classifications of the dataset are control and patient. Controls are further divided into 4 folders containing data for 4 different controls, whereas patients are divided into 10 folders containing data for 10 different patients. Each folder contains 10 different protein expression images (pseudo images of 10 proteins generated by image mass cytometer). The proteins that were observed are (Y SDHA, TOM22, NDUFB8, OSCP, GRIM19, VDAC1, COX4, MTCO1, UQCRC2, and Dystrophin). A total of 280 images were provided by Wellcome Centre Mitochondrial Research [WCMR]. Each image was provided in two different formats 140 JPEG images and 140 TIFF images.

For both experiments, the dataset was too small, so we used Patichfy [37] to divide the images into 512 x 512 small images to feed to the CNN network. After dividing the images into smaller images, the dataset was split into 80:10:10 for training, testing, and validation for both experiments. The details of the data splitting for both experiments are shown in Tables 1 & 2. A normalization range of 0 to 1 was chosen i.e., from 8 bit (0-255). A sample of the results obtained after the data augmentation is shown in Figure 2.

Table 1: Experiment 1(Single Protein Images (SDHA))

Classes	Training	Validation	Testing	Total
Controls	212	31	33	276
Patients	770	79	89	938
Total	982	110	112	1214

Table 2: Experiment 2(Multi-Stacked Protein Images)

Classes	Training	Validation	Testing	Total
Controls	146	36	46	228
Patients	461	116	144	721
Total	607	152	190	949

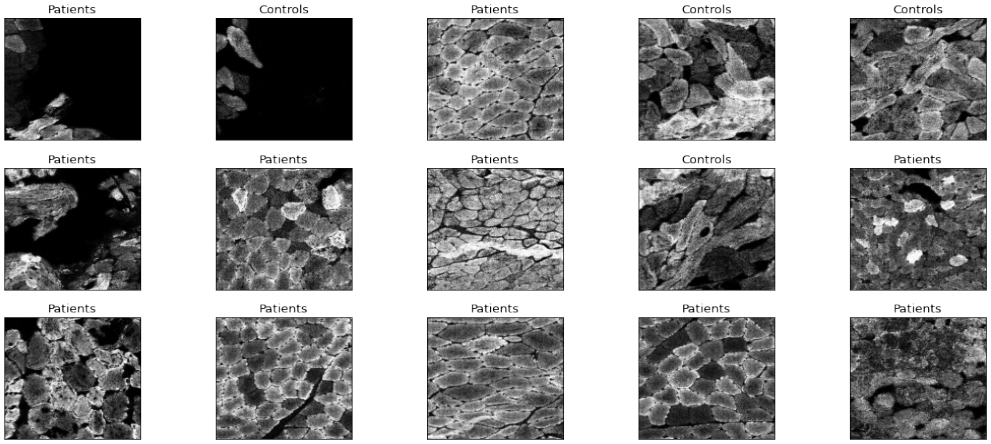

Figure 2: Samples of Images after splitting into train, test, validation and after applying data augmentation

3.2 Architectural Overview of Pre-Trained Models for Single Protein Expression Images

For the first experiment, we adapted two models VGG16 and ResNet50. There are 13 convolutional layers in the VGG16 model, along with 3 fully connected layers. Having 50 layers, ResNet-50 is a deep convolutional neural network (one MaxPool layer, one Average Pool layer, and 48 Convolution layers). A residual neural network (ResNet) is constructed by stacking residual blocks on top of one another.

The models considered in this experiment were trained using the ImageNet weights. We trained the model using weights that had already been trained on the ImageNet dataset. We applied transfer learning by freezing all except the final layer of the model's pre-trained layers. For both models, the input image parameters were constant at 224 x 224 x 3 pixels. The base model was built using pre-trained ImageNet weights. To the base model, we added 6 dense layers. First, dense layer 1 is added to the base model of 2048 units with ReLU activation. After dense layer 1, a total of 3 additional dense layers (dense layer 1, dense layer 3, dense layer 4) of 1024, 512, and 256 units with ReLU activation functions are added to the model. A dropout layer with a dropout size of 0.5 is placed after dense layer 4. Next, the model is flattened and another dense layer is added on top, Dense Layer 5 with 128 units and a ReLU activation function. At the very end, a dense layer 6 with two units that have the softmax activation function is added. To fine-tune the model, fully connected (FC) layers and a dropout of 0.5 are implemented in the research. Since we had 10 channels (10 different proteins), we chose the SDHA protein at random from all the other proteins as our input channel for our first experiment. We then copied the input channel values for each image three times to turn them into three channels. The selection of SDHA protein was made solely to demonstrate how well the CNN models worked, and the conversion of each image into three channels was done to show how well Grad-CAM and concept whitening approaches work because only images with three channels are supported by Grad-CAM and concept whitening. The architectural layouts of the transfer learning models are depicted in Figures 3 and 4.

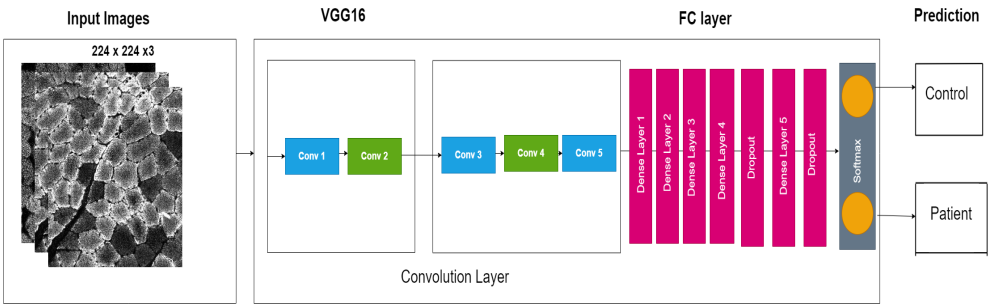

Figure 3: Summary of the VGG16 model architecture adapted in this research

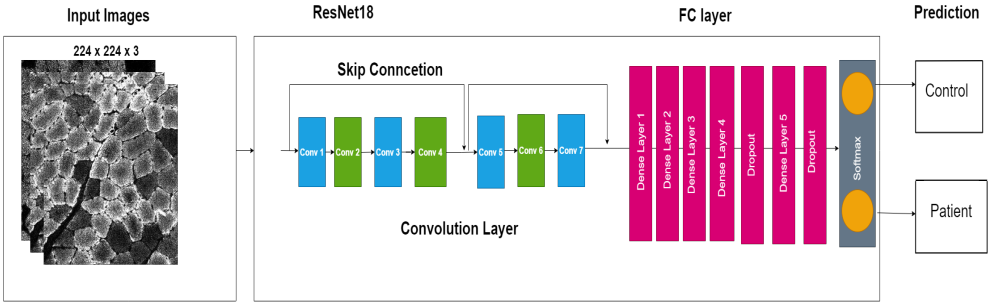

Figure 4: Summary of the ResNet-50 model architecture adapted in this research

3.3 Architectural Overview of Pre-Trained Models for Stacked Protein Expression Images

For the second experiment, we created a 10-channel image by combining all 10 protein expression images from the patient and control data, and we used these images to train a CNN model using transfer learning. The input images for all the pre-trained models, including VGG16 and ResNet50, which we used to classify single protein images, must have a size of 224 x 224 x 3, where 3 indicates the number of channels and contains ImageNet weights on all three channels. To account for the extra channels, we modified the pre-trained VGG-16 architecture. In the input layer of the network, we first define the image's height, width, and channels as 224, 224, and 10. When the number of channel was increased to 10, the only change was the number of trainable parameters in the first convolution layer, and the rest of the trainable parameters in the remaining convolution layers remain the same as in the original VGG-16 model.

Second, to use transfer learning on 10 channels, we must replace the random weights that would have been allocated to the extra 7 channels with ImageNet weights. We do this by averaging the ImageNet weights in the kernels of the first three channels, then copying the averaged weights to each of the seven individual channels, excluding the first three channels, which would already have the ImageNet weights from the original VGG16 model. Once the ImageNet weights were copied to the rest of the 7 channels, we froze all the layers of the pre-trained model and added the same additional dense and dropout layers that were used for single protein image classification.

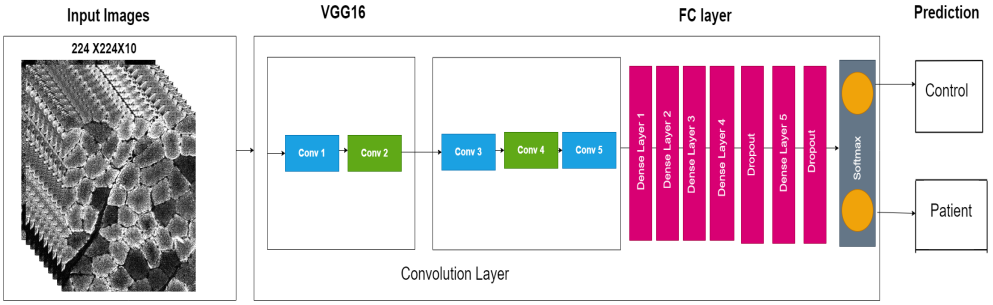

Figure 5: Summary of the VGG16 model architecture adapted in this research

3.4 Gradient-weighted class activation mapping (Grad-CAM)

Grad-CAM [14] technique is used to evaluate and find which features of protein expression images are leading to the predictions of mitochondrial disease. This method draws attention to the areas of the input image that the model focused on during the classification process, indicating that the feature maps created in the final convolution layer hold the spatial information needed to effectively capture the visual pattern. These visual patterns help in classifying the classes. The layers and abstracted features from the trained model are used to apply the Grad-CAM technique. The architecture explaining the Grad-CAM technique is shown in Figure 6.

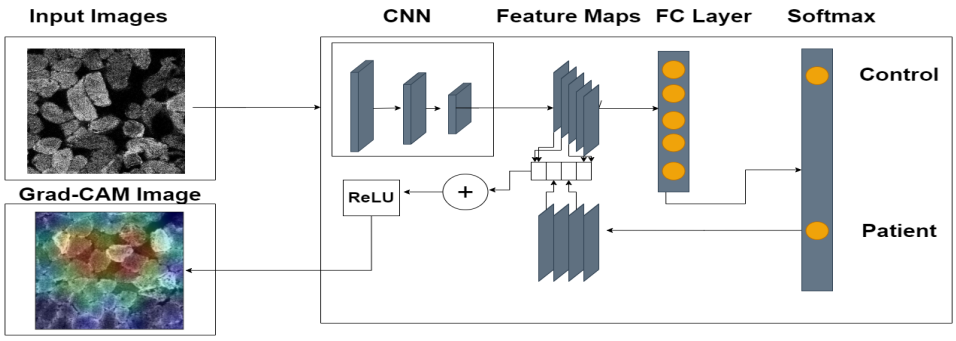

Figure 6: Architecture describing the Grad-CAM technique [38]

The architecture provides a high-level description of the Grad-CAM's workflow. From the architecture, one can see that, in the beginning, we are using images as input to build a model that is cut off at the layer for which we want to create a Grad-CAM heatmap. The completely connected layers for prediction are then attached. After that, we are running the input through the model, grabbing the layer output and loss. Then, we are determining the gradient of the output of our chosen model layer with respect to the model loss. In the final phase, we take the gradient sections that are contributing to the prediction and scale, resize, and reduce the images such that the heat map may be overlaid with the original image.

3.5 Neural Disentanglement (Concept whitening)

Concept whitening [15] technique was used to make the model interpretable. We applied this technique by adding the concept whitening layer before the softmax layer to our pre-built ResNet-18 model. Since concept samples can be manually selected, we created our own concept dataset by putting some images from the test data into the concept train and concept test for experiment 1 (single-channel protein expression images (SDHA)). The folder structure for the training of the model consists of 5 folders, concept train, concept test, train, validation, and testing each containing the images of the two classes (patient and control). The architecture explaining the Concept Whitening technique is shown in Figure 7.

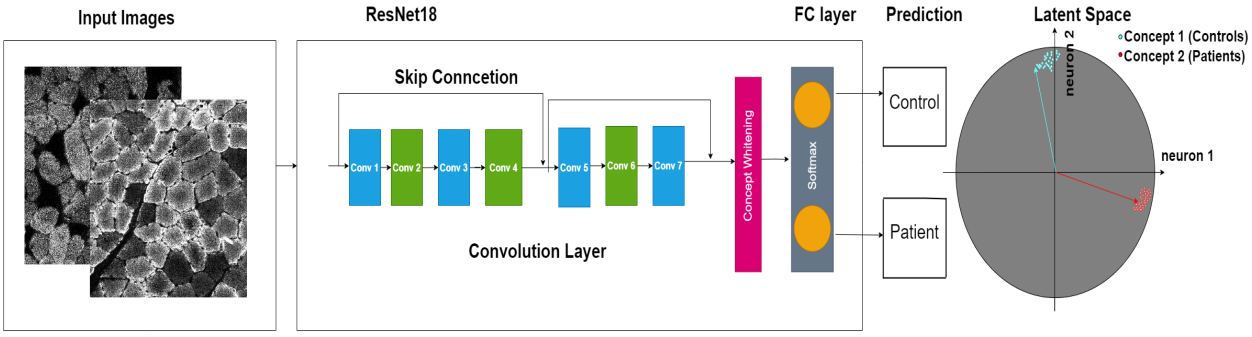

Figure 7: Architecture to describe the Concept Whitening technique [15]

From the architecture, one can see that, in the beginning, we are feeding the training images to train the ResNet-18 model. In place of the batch normalization layer at the end of layer four of ResNet-18, we are adding the concept whitening layer. The ResNet-18 architecture has eight residual blocks. Since we can

replace the batch normalization layer at one of these blocks, we are replacing the last block with the concept whitening layer to produce the best results. Each image is resized to 224 x 224 and fed to the modified ResNet-18. The input image goes through these eight blocks, five of which resize the image in half. So, each dimension image will be reduced from 224 x 224 to 7 x 7 size. The output after going through the concept whitening layer would be 7 x 7 x 512 where 512 is the number of channels.

Then, we are using images from the concept test folder as inputs to the model. For each input, we know which concept it belongs to. So, we are feeding the inputs through the network to get 7 x 7 x 512 as above. Next, we are flattening these vectors into 1 D vectors and then we are using the cosine similarity to check the orthogonality of the concepts. To calculate the concept scores, we are using the images from the test folder as the inputs. Similarly, again for each input, we will get a 7 x 7 x 512 output vector. We need a scaler for each of these scores. To calculate the scaler of these scores, we are applying the max pooling on the feature map, and then we are taking the mean to reduce a tensor from 7 x 7 x 512 to a 512- dimensional vector. From this, the score for each concept is the value at the corresponding index of the vector. For example, if concept control has an index of 0, then the score of concept control is the value at index 0 of the vector. Similarly, the concept patient will take the value at index 1 of the vector as the concept score.

4 Results and Discussion

4.1 Experiment Plan

As discussed above, the experiment was conducted in two phases. The first phase was further divided into two sub-experiments 1) Single channel image classification and 2) multi-channel image classification. For these two sub-experiments, we applied CNN to divide mitochondrial diseases into control and patient groups. For this, we adapted two popular pre-trained models VGG16 and ResNet-50. Transfer learning was used to add new layers, freeze the top layer, and train the models. In each sub-experiments, the models were trained 2-3 times by freezing different number of layers to find out the best optimal layer settings across all the models. Figures 3, 4, and 5 represent the model architectures used in this research, and table 3, represents the model hyperparameters used in the research across all the models. Since our main goal is not to classify, but rather to understand the underlying pathology of mitochondrial diseases. So, in the second phase, we applied the interpretability and explainability methods to single-channel images to understand, which features are responsible for predicting controls and patients. With Grad-CAM, we are hoping to find out the regions where the model is paying more attention, whether it is focusing on the edges or individual cells in the images. To identify which images are more activated in relation to the concepts, the concept whitening method is used. The environmental setup and implementation details are discussed in the next section.

4.2 Implementation Details

Using the two pre-trained models, VGG16 and ResNet-50—the prepared dataset is assessed. The data was divided into an 80:10:10 ratio for training, testing, and validation. Tables 1 and 2 provide more information on data splitting. The images from the dataset were downsized to 224 by 224 pixels for training, testing, and validation. Since we are dealing with a classification problem, we used the categorical cross-entropy to encode the class labels as normal integers. The batch size was set to 2 due to the small training dataset. The default standard learning rate value of 0.001 was chosen because we wanted to train the model slowly to make sure it could capture all relevant information. The optimizer chosen for all the models was adam because it gives better results than other optimization algorithms, offers faster computation time, and require fewer parameter for training. The number of epochs was set to 100 as an ideal choice, and early stopping on validation accuracy to stop the training when there is no improvement in the validation accuracy after 30 consecutive epochs. Details of model hyperparameters used in the research are shown in table 3.

A core i5 laptop served as the workstation for this study. Google Colab's integrated RAM and GPU are used in place of the laptop's inbuilt RAM (8 GB) and graphics (4 GB). The above methods and data pre-processing were run on Google Colab with the help of the Python 3 Google compute engine backend GPU and shared RAM (12.69 GB). TensorFlow, Keras, and PyTorch were introduced as free and open-source Python libraries. These libraries are used to do machine learning techniques using dataflow. It also helped train models and compute values. The dataset was uploaded to Google Drive for easy use on Google Colab. Four separate Colab files have been used to compare and assess four different models and approaches (VGG16, ResNet50, ResNet18 + CW, and VGG16 for stacked protein images). All the approaches and models worked well and generated the results we wanted, which are discussed in section 4.3.

Table 3: Model Hyperparameters across all the models used in the research

Hyperparameter	Values
Optimizer	Adam
Loss Function	Categorical-Cross-Entropy
Epochs	100
learning rate	0.001

4.3 Results and Discussion:

Training accuracy, validation accuracy, training loss, and validation loss at the end of 100 epochs were used to evaluate the training performance of the models. These parameters are calculated to determine how well the trained model fits the data. We plotted the training accuracy vs validation accuracy and training loss vs validation loss curves for models VGG16 and ResNet-50 for both experiments over 100 epochs to see whether the model is overfitting, underfitting or it is a good fit. The model is said to be a good fit when the training accuracy and validation accuracy are increasing over the number of epochs until the stable point is reached, and similarly, training loss and validation loss are decreasing over the number of epochs until the stable point is reached. For the VGG16 model in both experiments, the training accuracy and validation accuracy curve increases over the number of epochs and reaches a point of stability after 50 epochs, and has a small gap with the training accuracy. Similarly, the training loss and validation loss curve also decreases over the number of epochs and reaches a point of stability at 50 epochs, and has a small gap with the training loss. So, from the plots, we can say that the VGG16 model for both experiment 1 (single channel protein images (SDHA)) and experiment 2 (multi-channel stacked images) is a good fit. Figures 14, 15, and 16 show training accuracy vs validation accuracy and training loss vs validation loss for all the models and one can find the figures in the appendix under section results and discussion.

Furthermore, to evaluate the model results for both experiments, accuracy, precision, recall, and f1-score were calculated. The confusion matrix was used to generate the equations (1) - (4) which are used to calculate these parameters and can be found in the appendix under the methodology section. The confusion matrix was generated using a test dataset of 124 images. Confusion matrices for each model and each experiment are shown in Figure 17 in the appendix under the results and discussion section. Since our first goal was to adapt different pre-build models for the classification of different mitochondrial diseases, so for experiment 1 (single channel protein images (SDHA)) we compared three pre-build models VGG16, ResNet-50, and ResNet-18 +CW, and then for the second experiment (multi-channel stacked protein images), we adapted the best pre-build model obtained during the experiment 1. Table 4 shows the accuracy, precision, recall, and f1-score for different models (VGG16, ResNet-50, ResNet-18 + CW) for experiment 1 (single channel protein images (SDHA)) whereas table 5 shows the same parameters for model VGG16 for experiment 2 (multi-channel stacked protein images). From tables 4 and 5, we can see that after fine-tunning several transfer learning models VGG16 model attained the highest accuracy, precision, recall, and f1-score among the other models with values of 95%, 96%, 95%, and 95% respectively for both experiments. Whereas when we tried to make the model interpretable by adding concepts in the training phase the accuracy of the model ResNet-18 + CW was decreased attaining accuracy of 67% for experiment 1 (single channel protein images).

Table 4: Experiment 1 single channel protein images(SDHA)

Model	Accuracy	Precision	Recall	F1_Score
VGG16	0.95	0.96	0.94	0.95
ResNet50	0.87	0.85	0.85	0.85
ResNet18+CW	0.67	0.65	0.65	0.65

Table 5: Experiment 2 multi-channel stacked protein Images

Model	Accuracy	Precision	Recall	F1_score
VGG16	0.95	0.96	0.96	0.95

Prediction

The predictions of VGG16 accurately determined whether the protein images under test belonged to the control or patient for both the experiments, but ResNet50 and resNet50+CW had the lowest prediction accuracy and misclassified the test images. The protein expression image prediction results of the model VGG16 for single channel on the test dataset are shown in Figure 8.

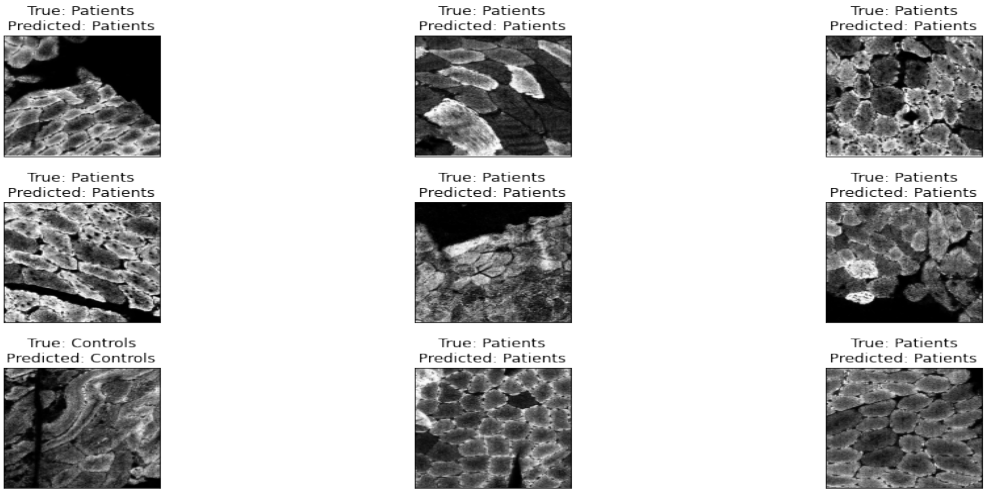

Figure 8: Results of prediction on a test dataset for single protein images

Grad-CAM technology was used on the VGG16 model of experiment 1 (single-channel protein image), which emphasised the areas that the model paid the most attention to during the feature extraction process. The discovery of the VGG16 model by the Grad-CAM technology is shown in Figure 9 The colour palette used in this study is jet. The blue hues in this palette represent lower values that exhibit no feature extraction for a given class, while the yellow and green hues represent medium values that exhibit some feature extraction. Red and dark red hues represent higher values that exhibit feature extraction for regions relevant to a given class.

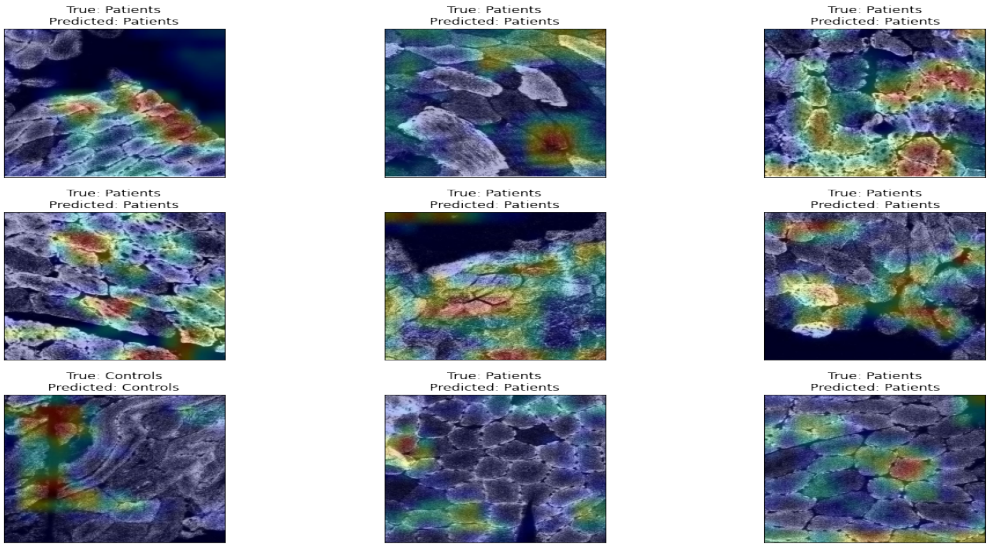

Figure 9: Grad Cam results on on a test dataset for single protein images(SDHA)

From the figures in 9, we see that model can pay attention to different areas of the images. The model is can distinguish different cells in the images based on the intensity value of the images. The red and dark hues in

the images represent where the models were paying more attention to each class. We think that the cells which are getting focused on by the model must contain the proteins with which the tissues were stained. The results of the Grad-CAM were sent to bio-medical scientists at WCMR for verification.

ResNet-18, for Experiment 1 (single-channel protein images (SDHA)) to make the model interpretable. After the model was trained, we achieved an accuracy of 67%. We then saved the top 50 images, which got the greatest activation with respect to the concepts for scientists at WCMR to study and tell us whether those images belong to the right classes or not. From start, we didn't have any actual metadata concepts for our experiment, we manually created our own concepts. These concepts are not clinically approved. The sole purpose of implementing the concept whitening was to demonstrate its working to the WCMR scientists, so that in the future if the scientists can gather the relevant concepts related to mitochondrial diseases, the concept whitening technique can help them a better way to understand the underlying pathology of the disease.

5 Conclusion and Recommendations

In the research, we applied transfer learning with fine-tuning on the two pre-trained models (VGG and ResNet) for experiment 1 (single channel protein images (SDHA)) and experiment 2 (multi-channel protein images) to predict mitochondrial diseases using protein expression images. The models' performance was verified using a number of metrics, including recall, precision, F1-score, accuracy, accuracy and loss graphs, and confusion matrix. ResNet-18 with the CW model showed the least accurate performance in classifying mitochondrial diseases, but the VGG16 model showed promising results in both experiments.

For experiment 1, the accuracy of VGG16, ResNet-50, and ResNet-18+CW was 95.25%, 87%, and 67%, respectively. For Experiment 2, VGG16 achieved 95% accuracy. To further evaluate the model, we used Grad-CAM and concept whitening for experiment 1 (single channel protein images (SDHA)). Grad-CAM highlighted the areas where the model was paying more attention. Whereas with concept whitening, we were able to attain only 67% accuracy, the main issue was the unavailability of clinically approved metadata of the protein expression images. Since the Grad-CAM technique focuses on highlighting the important pixels in the images, these highlighted regions might help scientists to focus only on those highlighted regions for a prediction and better understanding of the underlying pathology of mitochondrial diseases.

The results and the two approaches were demonstrated to bio-medical scientists at WCMR. The scientists were very happy to see how deep learning was able to highlight the important pixels in the images even with the small amount of data. They were also happy to see how well deep learning models were able to distinguish the difference between controls and patients. At present, scientists are verifying the results of both approaches. If scientists can confirm that the model is looking into the right areas then we can conclude that deep learning with interpretability and explainability techniques might help us in the prediction and understanding of mitochondrial diseases from protein expression images.

Future research can include collecting metadata for concept whitening, applying other explainability and interpretability methods like Layer-wise relevance propagation (LRP), Interpretable Image Recognition with Hierarchical Prototypes (HPnet), Interpretable Artificial Intelligence Algorithm for Breast Lesions (IAIABL), and ProtoPNet to evaluate the model performance and to understand the underlying pathology of the mitochondrial diseases.

Acknowledgments

I would like to thank Atif Khan and Dr Stephen McGough at Newcastle university, as well as bio-medical scientists at the Wellcome Centre Mitochondrial Research [WCMR], for their helpful contributions and feedback on the work.

6 Appendix

6.1 Objectives:

1. Determine whether it is possible to accurately diagnose mitochondrial diseases using adapted pre-trained model architectures like VGG16 and ResNet-50 using protein expression images obtained by image mass cytometry.
2. Evaluate, compare, and fine-tune different pre-trained model architectures using parameters like accuracy, precision, recall, f1 score, and confusion matrix.
3. Understand the underlying pathology of mitochondrial diseases by applying different interpretability and explainability AI approaches such as Grad-CAM and Neural Disentanglement to the best pre-trained model obtained after fine-tuning.

6.2 Methodology

6.2.1 Data Engineering:

A total of 280 images were provided by Wellcome Centre Mitochondrial Research [WCMR]. Each image was provided in two different formats 140 JPEG images and 140 TIFF images. To train the CNN model appropriately we required more number images, and one of the solutions was to divide the images into smaller patches to increase the data and then feed them to the neural network. Here patches implies a group of pixels in an image. since we had images with different shape, we divided each images into square patches of 512 x 512 pixels each. We used patchify one the python package to divide the images into small patches. Patchify divides an image into small overlapping sections based on the patch unit size specified, and then fuse the areas with the original image. Figure 10 shows the patchify results on a random image of the control.

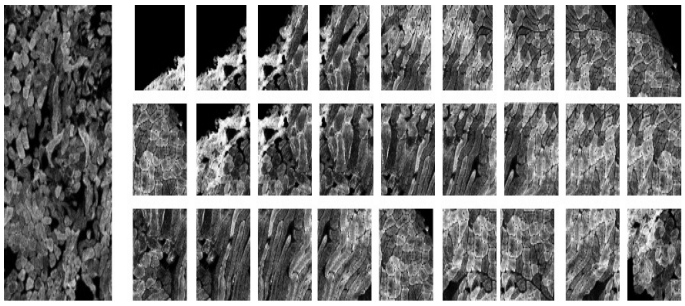

Figure 10: patchify results on a test dataset for single protein images

For experiment 2 first, we converted the images into a numpy array for each 10 channel and then stacked all the 10 channel as one by using numpy concatenation method. The end results after concatenation were multi-staked 10 channel numpy array. We used the same patichy function to divide the 10 channel numpy array into smaller patches of numpy for training.

6.2.2 Data Augmentation

By using several image augmentation techniques, we artificially expanded the size of the image training dataset. In general, image augmentation increases the dataset's size by creating modified versions of the training set images. This helps to boost dataset variation and, in turn, enhances the model's ability to know new images. For our two experiments we constructed an image data generator for the train dataset using the tensorflow.keras.preprocessing.image module that randomly applies preset parameters to the train dataset.

Image Data Generator parameters used to increase the size of the training dataset:

- Rescale — Each pixel in a digital image has a value between 0 and 255, with 0 representing black and 255 representing white. In order to make the original image's pixel values more evenly contribute to the overall loss, rescale the scales array of those values to be between [0,1]. If not, a lower learning rate should be utilised and a higher learning rate would be needed for a lower pixel range image because it causes more loss.

- Zoom range — A zoom of less than 1.0 enlarges the image. The image has been zoomed out by more than 1.0.
- Horizontal flip and vertical flip — A random horizontal and vertical flip is applied for images in the training dataset.
- Roataion range — At random, the image is rotated from 0 to 180 degrees.

With the exception of the rescale, these transformation techniques are applied at random to the images in the training dataset. Every image has been resized.

6.2.3 Concept Whitening Technique:

Concept whitening is an approach that aims to supply neural networks with latent spaces that are aligned with concepts relevant and important to the task for which they were trained. By using this approach, the CNN network's output and the features of an input image may be related more easily, making the deep learning model interpretable. Concept samples can be manually selected to feed the CNN network. The deep learning model runs through two parallel training cycles when implementing concept whitening. Concept whitening, alters neurons in each layer to align them with the classes in the concept dataset as the neural network adjusts its parameters to represent the classes in the main task. Consequently, concepts are divided into each layer and aligned with the concepts for which neurons were activated in a disentangled latent space. The demo video of concept whitening form authors (Link).

Ideas of CW

Concept: A concept may be anything related to images. For example, for skin lesion the concept may be clinical defined lesion size and age of the patients, Or in an normal image the concept may be anything that represent the images. For example if an image consist of many animals, we can round box one animal and use it has a concept.

Convolutional Neural Network's BatchNorm layer can be replaced by CW, a module that does the Whitening transformation in a neural network environment (CNNs). In order to produce interpretable models, CW decorrelates the CNN's latent space in addition to normalisation effects like BatchNorm's and aligns the latent space's axes with the concepts during training. To develop a better understanding of how the model learns, CW can describe the contributions made by each concept. Three potential data distributions in the latent space are shown in the figure below. Latent vectors can be disentangled using CW, making them suitable for use as a concept.

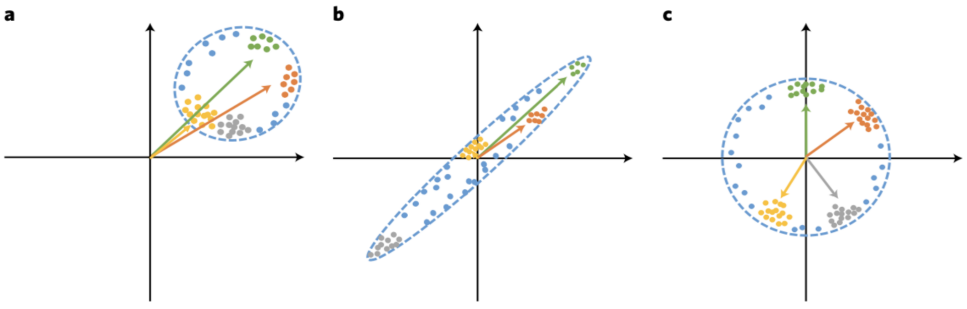

Figure 11: Data distributions in the latent space. (a) Not mean-centered; (b) standardized; (c) standardized and decorrelated (CW). [15]

How the model chooses to proceed during the training process may be seen from the trajectory of a sample in the space of concepts over several network levels.

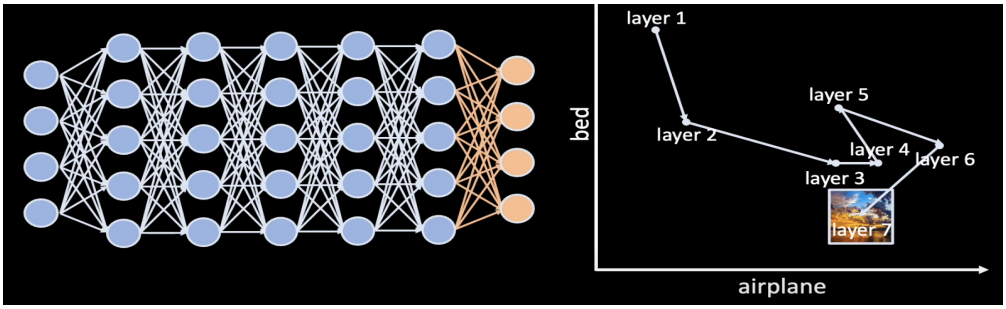

Figure 12: At the first layers, the model thinks that the image may contain a bed, but later changes its mind to airplane. [15]

How CW works

Standard settings call for mapping input x to a latent feature z=$\phi(x)$, and then make a prediction y=g(z)y=g(z). In the CW's scenario, in addition to minimising the loss from g, we simultaneously optimise ϕ with g so that z aligns with the desired concept. Suppose if we are interested in cj, j j=1 to k, where k are the concepts. Then we have two sets of dataset. Concept dataset X and normal train dataset D for classification task.

The two components of CW are orthogonal transformation and whitening. First, from Xcj (i.e., nj samples that activate the most in concept cj) we get its latent representation matrix Zcj (d x nj) which comprises the latent features of the ith sample of Xcj in each d-dimensional column. Next, to activate the data from the ZCA whitening, we must find a rotation matrix with the formula matrix Q(d x d) from Xcj in cj. So the jth axis of Q is represented by column qj. In particular, we must maximise the subsequent objective:

$$maxq1, q2, q3, q4....., qk \sum_{j=1}^{k} \frac{1}{nj} q_j^T \psi(Zcj) 1nj \times 1$$

where ψ is a whitening transformation with sample mean μ as a parameter and whitening matrix W.

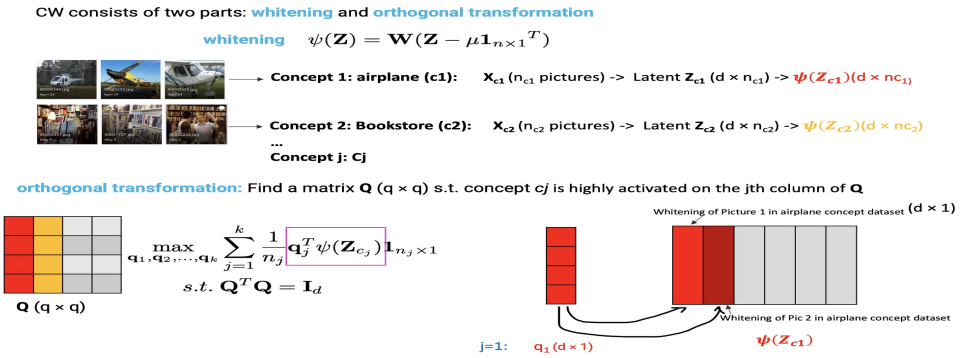

Figure 13: Illustration of Concept Whitening. [15]

6.2.4 Evaluation Metrics

To evaluate the model results for all the above models mentioned, accuracy, precision, recall, and f1-score were calculated. The equations to calculate these parameters can be found in the below. The main metric for assessing classification effectiveness is classifier accuracy (Accuracy). Equation defines it as the ratio of the number of instances (images) correctly classified by the number of examples (images) in the dataset being analysed (1). Precision and recall are the two metrics that are typically used to evaluate how well image classification systems perform. Equation (2) defines precision as the proportion of accurately identified classed images to the total number of images. Equation (3) defines recall as the proportion of correctly classified images in the test dataset to the total number of images. The F1-score is the harmonic mean of precision and recall; a higher value represents the system's ability to forecast the future. Precision and recall alone cannot be used to evaluate a system's efficacy. The F1-score is mathematically represented in Equation (4).

$$Accuracy = \frac{TP + TN}{FP + TP + TN + FP} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{FN + TP} \tag{3}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

6.3 Results and Discussions

Traning accuracy vs validation accuracy and training loss vs validation loss for VGG16, ResNet-50 for experiment 1 (single channel protein images (SDHA)) and experiment 2 (multi-channel stacked protein images) are shown in figures 14, 15 & 16.

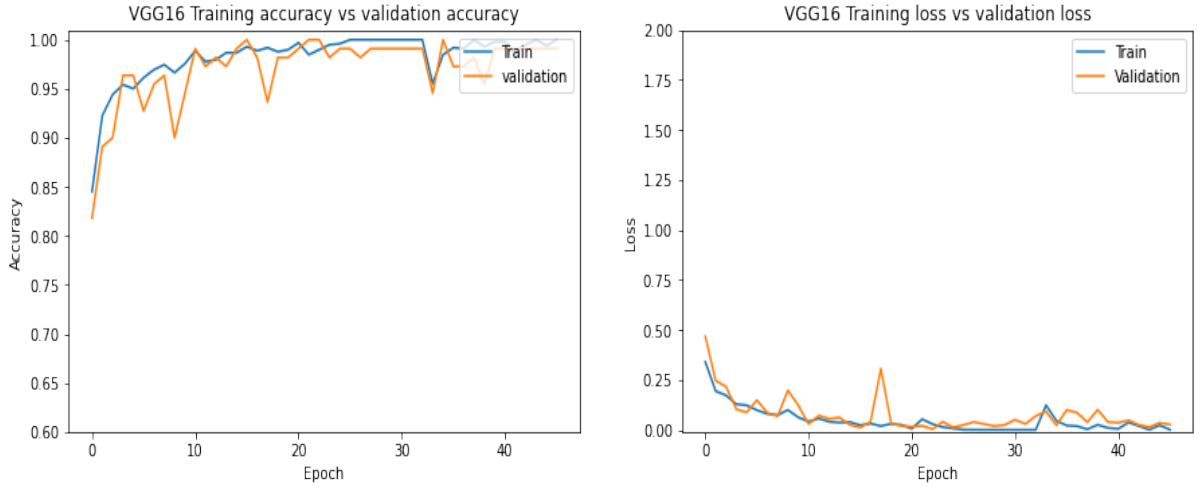

Figure 14: Training accuracies vs. validation accuracies and training loss vs. validation loss for VGG16 single-channel images (SDHA)

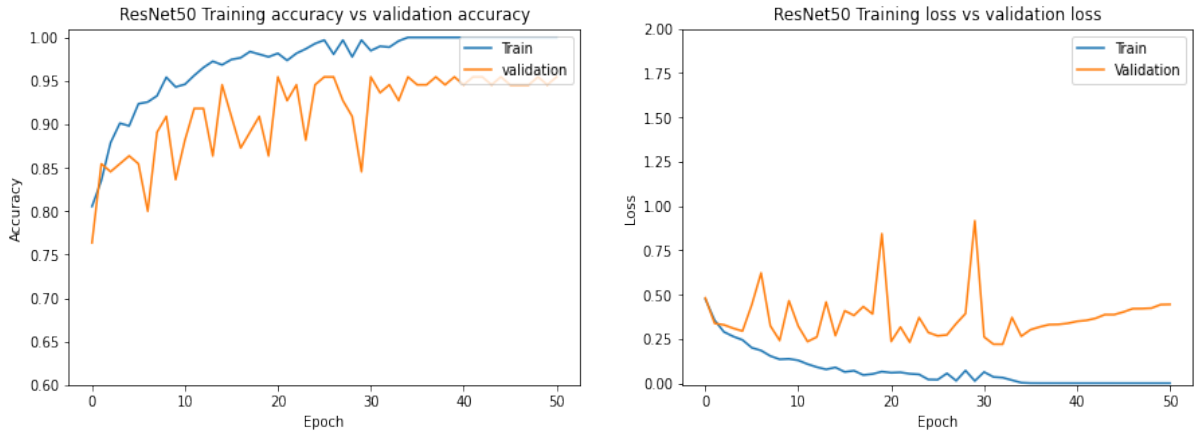

Figure 15: Training accuracies vs. validation accuracies and training loss vs. validation loss for ResNet single-channel images (SDHA)

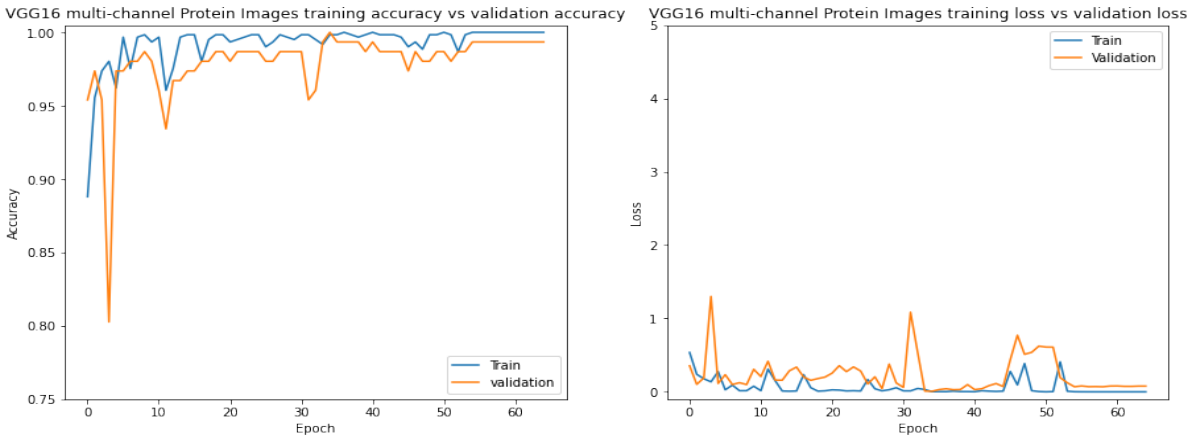

Figure 16: Training accuracies vs. validation accuracies and training loss vs. validation loss for VGG16 stacked images

Confusion Matrix

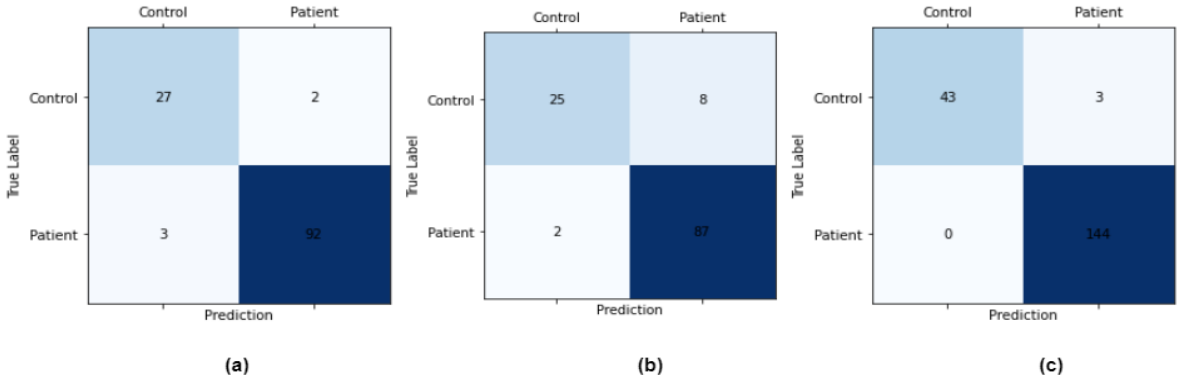

Figure 17: Confusion matrix for the models for the results on the testing dataset:(a) Confusion matrix for VGG16 for experiment 1 (Single Channel Protein images (SDHA)); (b) Confusion matrix for ResNet-50 for experiment 1 (single-channel protein images (SDHA)); (c) Confusion matrix for experiment 2: Stacked protein images(multi-channel protein images)

Reference

[1] Gorman, A. M. Schaefer, Y. Ng, N. Gomez, and Blakely, "Prevalence of nuclear and mitochondrial DNA mutations related to adult mitochondrial disease," *Annals of neurology*, vol. 77. Wiley Online Library, pp. 753–759, 2015.

[2] P. Forny, E. Footitt, J. E. Davison, A. Lam, and Woodward, "Diagnosing mitochondrial disorders remains challenging in the omics era," *Neurology. Genetics*, vol. 7. p. e597, 2021. doi: 10.1212/NXG.0000000000000597.

[3] O. M. Russell, G. S. Gorman, R. N. Lightowlers, and D. M. Turnbull, "Mitochondrial diseases: Hope for the future," *Cell*, vol. 181. pp. 168–188, 2020. doi: 10.1016/j.cell.2020.02.051.

[4] N. A. Khan, P. Govindaraj, A. K. Meena, and K. Thangaraj, "Mitochondrial disorders: Challenges in diagnosis & treatment," *The Indian journal of medical research*, vol. 141. pp. 13–26, 2015. doi: 10.4103/0971-5916.154489.

[5] M. Scarpelli, A. Todeschini, I. Volonghi, A. Padovani, and M. Filosto, "Mitochondrial diseases: Advances and issues," *The application of clinical genetics*, vol. 10. pp. 21–26, 2017. doi: 10.2147/tacg.s94267.

[6] L. C. Greaves, A. K. Reeve, R. W. Taylor, and D. M. Turnbull, "Mitochondrial DNA and disease," *The Journal of pathology*, vol. 226. pp. 274–286, 2012. doi: 10.1002/path.3028.

[7] R. McFarland and D. M. Turnbull, "Batteries not included: Diagnosis and management of mitochondrial disease," *Journal of Internal Medicine*, vol. 265. pp. 210–228, Feb. 2009. doi: 10.1111/j.1365-2796.2008.02066.x.

[8] R. J. T. Rodenburg, "Biochemical diagnosis of mitochondrial disorders," *Journal of Inherited Metabolic Disease*, vol. 34. pp. 283–292, Apr. 2011. doi: 10.1007/s10545-010-9081-y.

[9] S. Parikh, A. Goldstein, M. K. Koenig, and Scaglia, "Diagnosis and management of mitochondrial disease: A consensus statement from the mitochondrial medicine society," *Genetics in medicine: official journal of the American College of Medical Genetics*, vol. 17, no. 9, pp. 689–701, 2015, doi: 10.1038/gim.2014.177.

[10] C. Chen, D. McDonald, A. Blain, A. Sachdeva, L. Bone, and Smith, "Imaging mass cytometry reveals generalised deficiency in OXPHOS complexes in parkinson's disease," *NPJ Parkinson's disease*, vol. 7. p. 39, 2021. doi: 10.1038/s41531-021-00182-x.

[11] G. Baselli, M. Codari, and F. Sardanelli, "Opening the black box of machine learning in radiology: Can the proximity of annotated cases be a way?" *European radiology experimental*, vol. 4, no. 1, p. 30, 2020, doi: 10.1186/s41747-020-00159-0.

[12] U. Pawar, D. O'Shea, S. Rea, and R. O'Reilly, "Explainable AI in healthcare." pp. 1–2, 2020. doi: 10.1109/CyberSA49311.2020.9139655.

[13] T. N. Mundhenk, B. Y. Chen, and G. Friedland, "Efficient saliency maps for explainable AI." 2019. doi: 10.48550/ARXIV.1911.11293.

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *International journal of computer vision*, vol. 128. pp. 336–359, 2020. doi: 10.1007/s11263-019-01228-7.

[15] Z. Chen, Y. Bei, and C. Rudin, "Concept whitening for interpretable image recognition," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 772–782, 2020, doi: 10.1038/s42256-020-00265-z.

[16] C. Warren, D. McDonald, R. Capaldi, D. Deehan, and Taylor, "Decoding mitochondrial heterogeneity in single muscle fibres by imaging mass cytometry," *Scientific reports*, vol. 10. p. 15336, 2020. doi: 10.1038/s41598-020-70885-3.

[17] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." 2018. doi: 10.48550/ARXIV.1811.10154.

[18] Z. C. Lipton, "The mythos of model interpretability." 2016. doi: 10.48550/ARXIV.1606.03490.

[19] R. Marcinkevics and J. E. Vogt, "Interpretability and explainability: A machine learning zoo minitour," 2020, doi: 10.3929/ETHZ-B-000454597.

[20] B. H. M. van der Velden, M. H. A. Janse, M. A. A. Ragusi, C. E. Loo, and K. G. A. Gilhuijs, "Volumetric breast density estimation on MRI using explainable deep learning regression," *Scientific reports*, vol. 10. p. 18095, 2020. doi: 10.1038/s41598-020-75167-6.

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *arXiv [cs.CV]*, 2013, doi: 10.48550/ARXIV.1311.2901.

[22] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv [cs.CV]*. 2013. doi: 10.48550/ARXIV.1312.6034.

[23] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv [cs.LG]*. 2014. doi: 10.48550/ARXIV.1412.6806.

[24] B. D. de Vos, J. M. Wolterink, T. Leiner, P. A. de Jong, N. Lessmann, and I. Isgum, "Direct automatic coronary calcium scoring in cardiac and chest CT," *IEEE transactions on medical imaging*, vol. 38. pp. 2127–2138, 2019. doi: 10.1109/TMI.2019.2899534.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 2921–2929. doi: 10.1109/CVPR.2016.319.

[26] H. Lee, S. Yune, and Mansouri, "An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets," *Nature biomedical engineering*, vol. 3. pp. 173–182, 2019. doi: 10.1038/s41551-018-0324-9.

[27] J. Ji, "Gradient-based interpretation on convolutional neural network for classification of pathological images." pp. 83–86, 2019. doi: 10.1109/ITCA49981.2019.00026.

[28] K. Kowsari, R. Sali, L. Ehsan, W. Adorno, A. Ali, and Moore, "HMIC: Hierarchical medical image classification, a deep learning approach," *Information (Basel)*, vol. 11. p. 318, 2020. doi: 10.3390/info11060318.

[29] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Muller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10. p. e0130140, 2015. doi: 10.1371/journal.pone.0130140.

[30] M. Bohle, F. Eitel, M. Weygandt, and K. Ritter, "Layer-wise relevance propagation for explaining deep neural network decisions in MRI-based alzheimer's disease classification," *Frontiers in aging neuroscience*, vol. 11. p. 194, 2019. doi: 10.3389/fnagi.2019.00194.

[31] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin, "This looks like that: Deep learning for interpretable image recognition," *arXiv [cs.LG]*. 2018. doi: 10.48550/ARXIV.1806.10574.

[32] P. Hase, C. Chen, O. Li, and C. Rudin, "Interpretable image recognition with hierarchical prototypes," *arXiv [cs.CV]*. 2019. doi: 10.48550/ARXIV.1906.10651.

[33] A. J. Barnett, C. Schwartz, J. Y. Chen, and C. Rudin, "IAIA-BL: A case-based interpretable deep learning model for classification of mass lesions in digital mammography," *arXiv [cs.LG]*. 2021. doi: 10.48550/ARXIV.2103.12308.

[34] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv [stat.ML]*. 2017. doi: 10.48550/ARXIV.1702.08608.

[35] H. Lakkaraju, S. H. Bach, and L. Jure, "Interpretable decision sets: A joint framework for description and prediction," *International Conference on Knowledge Discovery & Data Mining. International Conference on Knowledge Discovery & Data Mining*, vol. 2016. pp. 1675–1684, 2016. doi: 10.1145/2939672.2939874.

[36] M. AlQuraishi and P. K. Sorger, "Differentiable biology: Using deep learning for biophysics-based and data-driven modeling of molecular mechanisms," *Nature methods*, vol. 18, no. 10, pp. 1169–1180, 2021, doi: 10.1038/s41592-021-01283-4.

[37] *PyPI*. Available: `https://pypi.org/project/patchify/`

[38] C. V. Aravinda, M. Lin, K. R. Udaya Kumar Reddy, and G. Amar Prabhu, "A demystifying convolutional neural networks using grad-CAM for prediction of coronavirus disease (COVID-19) on x-ray images." Elsevier, pp. 429–450, 2021.