INTERPRETABLE AI FOR PROTEIN EXPRESSION IMAGES AND ASSOCIATED CLINICAL METADATA

Sushanth Shivpura Ramesh
School of Computing
Newcastle University
s.shivpura-ramesh2@newcastle.ac.uk

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

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

ABSTRACT

Mitochondrial diseases are genetically inherited disorders caused due to mitochondrial dysfunction [1]. The Study of skeletal muscle biopsy is the most common way to understand the pathology of the mitochondrial disease [2]. Image Mass Cytometry (IMC) is used to quantify and analyse ten mitochondrial proteins present in patients with genetically characterised mitochondrial disease from the skeletal muscle tissues [2]. In this research, we use the pseudo protein expression images from the IMC to train a binary classification Convolutional Neural Network (CNN) with transfer learning to classify the images into control or patient classes. Existing pre-trained models VGG16 and ResNet101v2 with fine-tuned fully connected layers are trained with single protein expression images. Of the two CNN configurations for single protein image classification, the best accuracy and F1-score of 95% and 0.94 respectively was achieved on the VGG16 model. To validate the features used by the network, class saliency map, a post hoc explainable method has been implemented which showed good results by highlighting the pixels used by the network to make the classification. Finally, the VGG16 and VGG19 models are modified to train on the stacked 10-channel protein images. The trained VGG16 model for ten channel protein images has an accuracy of 98% and F1-score of 0.95 which had better performance than VGG19 model. The proposed approach with the application of Interpretable AI methods on the pseudo protein expression images from IMC is the first prototype study which helps in understanding the underlying pathology of mitochondrial diseases.

Keywords Image Mass Cytometry · Convolution Neural Network · Transfer Learning · Class Saliency Map

1 Introduction

Mitochondria are double-membrane subcellular organelles, also known as the powerhouse of the cells [3]. A normally functioning Mitochondria generates most of the energy i.e., adenosine triphosphate (ATP) to power the cells by oxidative phosphorylation (OXPHOS) [4]. Mitochondrial disorders manifests because of mutation in the maternally inherited mitochondrial DNA (mtDNA) or nuclear DNA (nDNA) genes that encode the mitochondria [1]. The functioning of multiple organs are affected when the mitochondria fails to produce the required energy (ATP) [1]. Mitochondrial diseases are highly heterogeneous and adversely affect high-energy demanding cells like neurons and skeletal muscle cells [5]. Several diseases are associated with mitochondrial dysfunction, such as neurodegenerative disorders, cardiovascular disorders, neurometabolic diseases, cancer, obesity, and diabetes [6]. Currently, there is no cure for mitochondrial diseases, and it affects 1 in 5000 people [7]. Mitochondrial diseases are studied by finding the relationship between genome, OXPHOS proteins & disease symptoms [1].

There are several ways to diagnose mitochondrial diseases, such as evaluating selected mitochondrial biomarkers in blood, genetic testing and a clinical scoring called Newcastle Mitochondrial Disease Scale (NMDS)[8, 9]. However, testing of tissue biopsy from skeletal muscle is regarded as the most accurate and gold standard for mitochondrial diagnosis and understanding the underlying pathology of the disease [8]. Scientists at the Wellcome Center for Mitochondrial Research (WCMR) have identified ten protein targets from all five OXPHOS complexes and observed their expression using Image Mass Cytometry (IMC) [2]. These pseudo images of protein expressions from IMC are then segmented, quantified using various statistical methods, and analysed to understand relationship amongst various protein expression levels and disease pathology [2]. This is currently not very useful due to high dimensionality of data and limitation of statistical methods with high throughput image data.

With the protein expression images from the image mass cytometry, we can build and train a Deep learning (DL) model to classify different mitochondrial disorders and look for the basis of the prediction, which could uncover underlying pathology of the disease. High-stake decision-making settings such as health care, as in our case, require the model to be interrogated using interpretability approaches such as saliency maps and neural disentanglement [10]. This will help us understand if the network is picking the right features to make the prediction rather than picking up stray artifacts such as folds in the tissue samples or cell shapes. This would also highlight the features that may explain or lead to understanding of the disease pathology. DL networks are considered black boxes due to their opaque nature, even though they may have promising performance [11]. Implementing interpretability increases transparency and unveils the limitations and biases if any in the network training [11]. This would also increase clinical trust in Artificial Intelligence [AI] in decision-making by medical professionals.

In this project, we have fine-tuned the existing VGG16 and ResNet101v2 pre-trained CNN models to classify single protein expression images into patient and control classes and achieved the best accuracy of 95% and 0.94 F1-score for the VGG16 model. Saliency map, an explainable method, is implemented on VGG16 to visualize, and analyse the features used by the network to make the prediction. Finally, the VGG16 and VGG19 models are modified to accommodate and train with stacked 10-channel protein images. VGG16 has an accuracy and F1-score of 98% and 0.95, respectively for the multi-channel image classification model.

The rest of the paper gives the detailed analysis of the proposed approach and is organized as follows: Section 2 provides the background and relevant work and in section 3 we discuss the methodology of the proposed approach. In section 4 we discuss the performance and results for the experiments carried out. Finally, section 5 gives the conclusion of the project and provides future work recommendations.

1.1 Aim

To use Interpretable and explainable artificial intelligence approaches such as saliency maps and Neural Disentanglement to comprehend the underlying pathology of mitochondrial diseases with the features extracted by deep learning models.

2 Background and Related Work

2.1 Biological Background

Oxidative phosphorylation is the process by which ATP is synthesised through the transfer of electrons between the inter mitochondrial membrane due to an electrochemical gradient across the electron transfer chain, which consists of protein complexes I–V [12, 13]. In a healthy mitochondrion, all the oxidative phosphorylation proteins must be expressed in the right quantity for the ATP to synthesise normally [2]. In a defective mitochondrion, some proteins are not expressed in the correct quantity, for which other proteins are upregulated to compensate [2]. By accurately quantifying and assessing these heterogeneously expressed OXPHOS protein deficiencies using IMC will help in understanding the pathology of mitochondrial diseases [2].

Skeletal muscle cells are high-energy demanding cells which consists of large number of mitochondria, so a biopsy of skeletal muscle tissue is used from a patient diagnosed with mitochondrial disease [1]. To quantify the protein of interest the tissue is stained with metal labelled antibody conjugate, which binds to the targeted protein [2]. This tissue is then laser ablated, and the pseudo image is captured using an image mass cytometer. These pseudo protein expression images are then segmented, quantified using various statistical methods, and analyzed to understand relationship amongst various protein expression levels and disease pathology [2, 14]. In our proposed approach, we use the pseudo protein expression image to classify them into Patient and Control images using deep learning models. We then apply explainability and interpretability methods to understand the underlying pathology of mitochondrial diseases with the features extracted by the model. Figure 1 shows the application of IMC on skeletal muscle tissues to generate pseudo protein expression images and our proposed workflow.

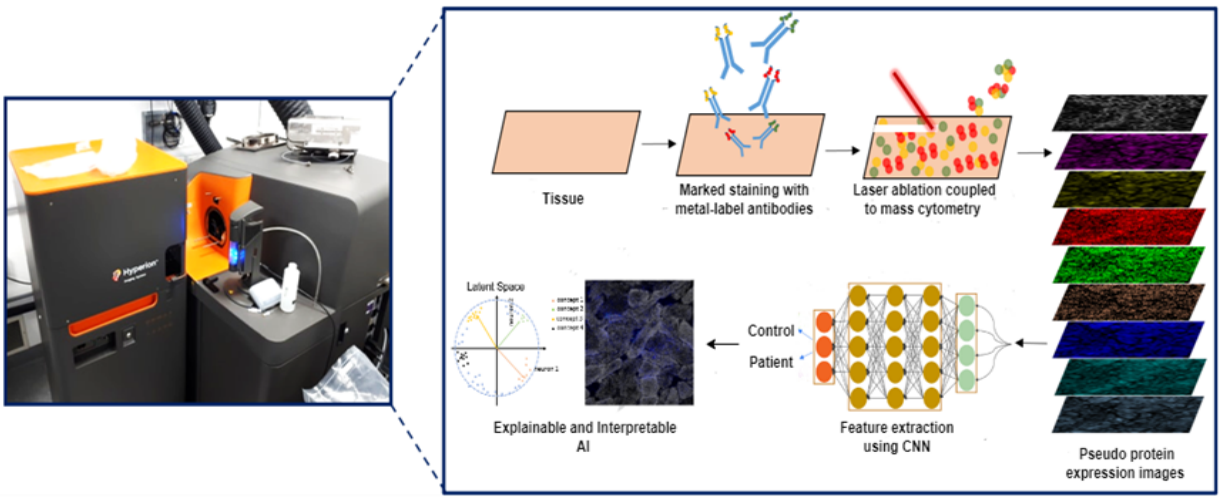

Figure 1: Image Mass cytometer and associated workflow [7]

2.2 CNN Transfer Learning

There has been an increased application of CNN in image classification tasks, especially in the field of medical sciences [15]. Transfer learning methods with pre-trained CNN models such as VGG, ResNet, and Inception are used extensively in extracting features to identify anomalies with respect to healthy individuals from histopathological images [16, 17]. Some examples of implementations are using VGGNet CNN model with transfer learning to diagnose breast cancer using biopsy of breast tissue images by Weiming Zhi et al., which has a prediction accuracy above 85% [18]. Similarly, Claudia Mazo et al. have experimented with four popular CNN architectures - ResNet, VGG16, VGG19 and Inception to classify cardiovascular tissues in histological images, where they have achieved the best F1-scores ranging between 0.712 and 0.955 on the ResNet models [19]. Despite the promising performance, the adoption of artificial intelligence in the clinical workflow is very slow due to the opaque and complex nature of the network [20]. Most deep learning models have many dense layers and are considered black box models as they do not explain the basis of the predictions [21]. It is therefore required to use explainable and interpretability methods in high-stake decisions such as health care settings to know if the network is extracting the right features to make the prediction and, in our case discover unknown features to reveal the underlying pathology. [21, 22] In section 2.3 and 2.4 we discuss different explainable and interpretable methods that were used for medical image analysis.

2.3 Explainable Artificial Intelligence(XAI)

Saliency maps are one of the commonly used explainable methods by researchers for medical image analysis which gives visual insights into the features used by the neural network to make decisions [23, 24]. Most saliency map methods use backpropagation and give a post hoc explanation of the network [23]. We have tried to use some of these techniques such as Class saliency map and Grad-Cam to visualize the features extracted by the network. An overview of some of the explainable AI methods used in medical image analysis are discussed below.

Class Saliency Map: The method aims to highlight the pixels in the image using gradient calculation and assigning an importance score to an individual feature which reflects the influence on the model prediction [25]. One of the implementations of class saliency extraction is by de Vos et al. to estimate calcium content in the coronary artery using chest and cardiac Computed Tomography (CT) images [26].

Class Activation Mapping (CAM): This method replaces the CNN model's fully connected layers with the Global Average Pooling (GAP) layer [27]. Each feature map's activations are averaged and concatenated by the GAP layers and output them as a vector [27]. The last layer i.e., SoftMax is then fed with the weighted sum of the vector [27]. The important features used by the network are highlighted by projecting back the output weights on the convolutional feature maps [27]. CAM has been implemented on Inception-V3, ResNet-152, and Inception-ResNet-V2 CNN models to classify and visualize the severity of diabetic retinopathy using retina fundus images by Jiang et al. [28].

Gradient-weighted class activation mapping (Grad-CAM): Grad-CAM is a generalized version of CAM, which works with any type of CNN as it does not require global average pooling compared to CAM [29]. The gradient information fed into the network's final convolution layer is used to assign importance values to the individual neuron

for the class of interest.[29] This can be applied to any layer of the CNN, though usually applied to the output layer.[29] Windisch et al. used the ResNet50 model to classify brain tumours from MRI slices and validate the network output with GRAD-CAM [30].

2.4 Interpretable Artificial Intelligence

Most explainable methods provide post hoc analysis of the network mostly based on the output layers.[31] Interpretability distinguishes a model's passive feature by explaining the overall relationship between features and labels based on the entire model. [32] Interpretability methods are applied by adding additional layers or mechanisms to an existing model without altering the model's performance [31]. Some of the interpretability methods used in image classifications are summarised below.

Interpretable Image Recognition with Hierarchical Prototypes: CNN models are more interpretable when they classify images based on features that a person can understand clearly, so visual prototype methods have been developed for interpretable image classification [33]. One of the recently implemented methods by Peter Hase et al. is hierarchically organized prototypes to classify images based on predefined taxonomy, which helps to know the reason for the forecast of an image at each level of the taxonomy [34]. This also interpretably classifies images from an unknown class at the taxonomy level to which they correlate [34]. The authors have demonstrated this method on a subset of images from the ImageNet database [34].

Concept Whitening(Neural Disentanglement): One of the other interpretability approaches in image recognition is to analyze the neural network's hidden layers by a mechanism called concept whitening (CW), which was introduced by Zhi Chen et al. [32]. When the CW module is added to the neural network, the latent space's axis is aligned with the concept of interest.[32]. It is an alternative to batch normalization that decorrelates and normalizes the latent spaces [32]. Adding a CW module to the network does not deteriorate the model's predictive performance [32].

Interpretable Artificial Intelligence Algorithm for Breast Lesions (IAIABL):A. J. Barnett et al. have introduced an interpretable computer vision machine learning algorithm which uses case based reasoning for mammography to predict if a breast lesion is cancerous or benign that has an accuracy comparable with radiologist[35] .It has been demonstrated better accuracy and interpretability can be achieved even with small number of images by incorporating image object labelling along with pixel level annotations in the prediction model.[35] The model highlights the parts of images that were relevant to the classification as a part of interpretability approach[35].

Few advantages with the application of the interpretability and explainability approaches based on the recent studies include

- **Understanding the complex structures:** Understanding of various natural phenomena such as protein folding and complex histological features which otherwise require trained medical professionals are made easy with implementation of interpretability methods [36].
- **Increased transparency:** Implementation of interpretability and explainability can increase trust by explaining how an AI system came to a particular decision [37].
- **Model development:** Explanations generated through interpretability and explainability approaches can aid in the comprehension of learnt rules, enabling the detection of errors, and enhancing the models [38].
- **Result tracing:** With the features extracted by the interpretable methods it allows for back tracing it to the original input data by decolorating the model layers [39].

3 Methodology

The project was executed in three phases. In the first phase of the project, the objective was to build a pipeline to classify the single protein expression images into patient and control classes by fine-tuning existing pre-trained CNN models such as VGG and ResNet using transfer learning. The second phase was to apply interpretability and explainability methods to analyse and visualize the features that were used by the network to classify the images. The final phase involved stacking all protein images together and building CNN models that work with multichannel images. A generalized architecture of the implementation is shown in Figure 2 below. This section focuses on understanding the data and the methodology used in building the models in detail.

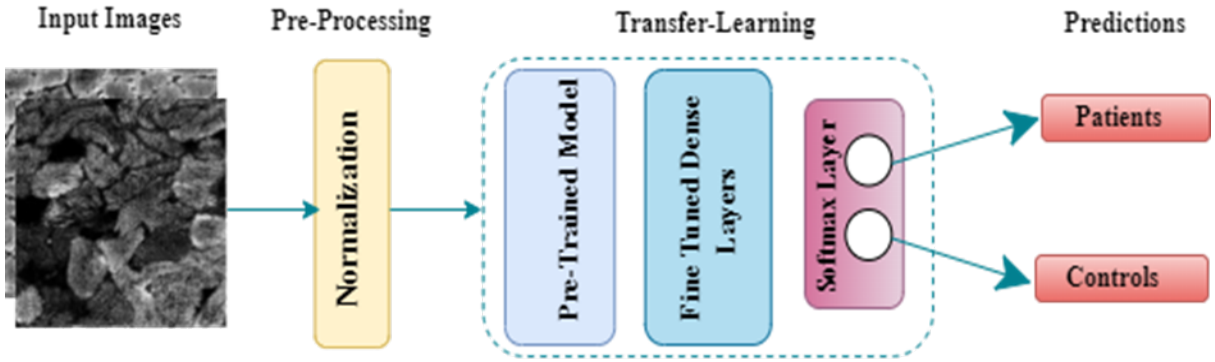

Figure 2: Basic architecture for protein image classification

3.1 Data Understanding and Pre-Processing

The data set used in this project consists of skeletal muscle tissue biopsy pseudo images obtained from Image mass cytometry (IMC) to quantify different proteins associated with mitochondrial diseases. The data set was provided by Wellcome Centre for Mitochondrial Research (WCMR).

The images are classified into two categories i.e.

- **Patient (P):** Tissue image from patients with clinical and genetic characteristics of mitochondrial diseases. The data set consists of images from 10 patient samples.
- **Control (C):** Protein expression tissue image from healthy individuals without any history of mitochondrial disease. The data set consists of images from 4 control samples.

Each patient or control sample consists of 10 images, each corresponding to the type of protein expression namely SDHA, TOM22, NDUFB8, OSCP, GRIM19, VDAC1, COX4, MTCO1, UQCRC2, and DYSTROPHIN. In total, there are 140 unique images stored in both JPEG and TIFF formats. Figure 10 in Appendix shows the folder structure of the dataset.

The images are pre-processed before they are fed into the model to train. The dimensions of protein expression images from IMC are large, and the number of images for the model to train is very small. So, each image is split into 512 X 512 images, which would reduce the image size and increase the number of images for the model to train efficiently. By splitting the images, we also have fewer individual muscle fiber cells in each image, allowing the network to identify and extract the features more effectively rather than having a larger number of cells from the original protein expression IMC images. Next, the data was split into train, test, and validation sets with an 80:10:10 ratio, respectively, ensuring no data leakage. Finally, the images were normalised before training. A summary of the number of images after the data is split for both single protein and stacked protein classification is given in Table 1 and Table 2, respectively. Figure 3 shows some of the sample patient and control images of TOM22 protein images after splitting.

Class	Training	Validation	Test	Total
Patient	770	79	89	938
Control	212	31	33	276
Total	982	110	112	1214

Table 1: Data split for TOMM22 Single protein images

3.2 Single Protein Image Classification Model Architecture

CNNs with transfer learning are most extensively used for computer vision tasks and are increasingly used in medical image analysis for disease classification, detection, and image segmentation. VGG16 and ResNet101v2 are some of the best-performing CNN models that are trained on 1.3 million ImageNet dataset, which was used to classify 1000 classes

Class	Training	Validation	Test	Total
Patient	461	116	144	721
Control	146	36	46	228
Total	607	152	190	949

Table 2: Data split for multi-channel protein images

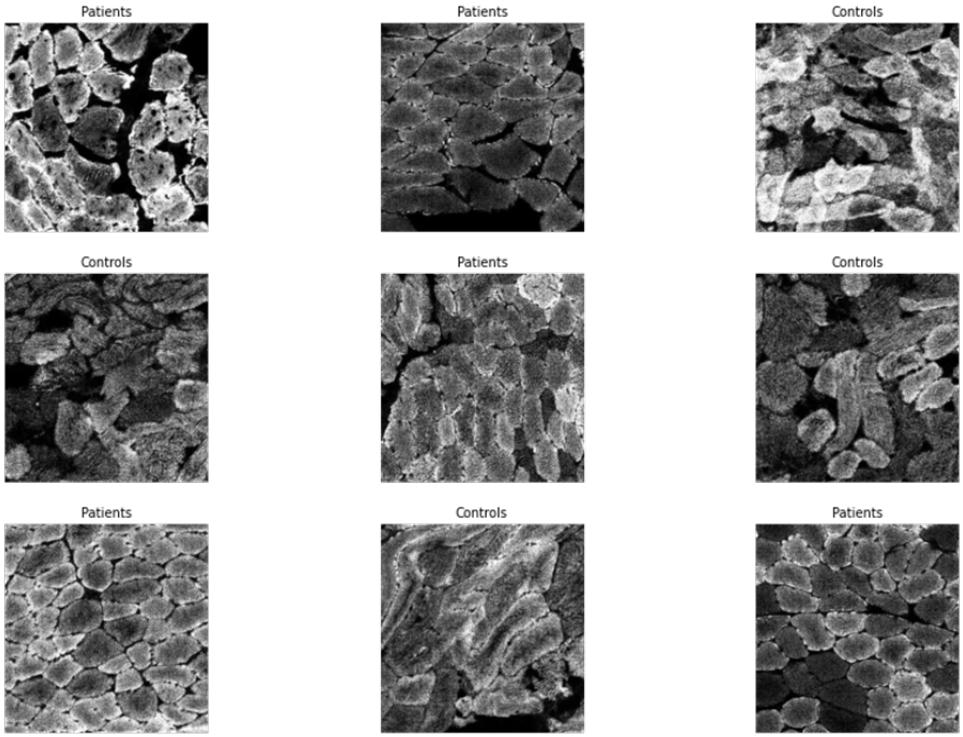

Figure 3: Sample images of TOM22 protein images after splitting

of images. VGG16 contains 16 deep layers i.e., 13 convolution layers and 3 fully connected layers. The ResNet101v2 network contains 101 deep layers built by stacking residual blocks and uses skip connections to address the problem of vanishing gradient. In this experiment, we used transfer learning with base models VGG16 and ResNet101v2 with pre-trained ImageNet weights. All the layers in the base model are frozen except for the last layer. In addition to the base layers, the network was fine-tuned with the addition of 6 dense layers. Dense layer 1 contains 2048 units with ReLU activation function. The subsequent dense layers contain 1024, 512 and 256 units receptively with the ReLU activation function. A dropout layer with 0.5 size is added, and the output is flattened between dense layer 4 and dense layer 5, which contains 128 units with the ReLU activation function. The final two layers are the dropout layer with 0.5 size. The final output layer is a dense layer with 2 units and a Softmax activation function which is used to calculate the probability of the classes, which in our case is patient and control. The architecture of both the models is shown in the Figure 4 and 5.

3.3 Class Saliency Maps

Saliency maps are the most frequently used explainable method to investigate and interpret the hidden layers of the network. Saliency maps helps to highlight the pixels in an image that are significant in making class perditions. If an image is represented by I belonging to class c and $S_c(I)$ represents the class score function of the convolution network, then the significance of the pixel in image I is based on the influence of the $S_c(I)$ score. As the class score is highly nonlinear in a deep convolution network, we can approximate the class core $S_c(I)$ by computing the first order Taylor expansion in Equation 1.

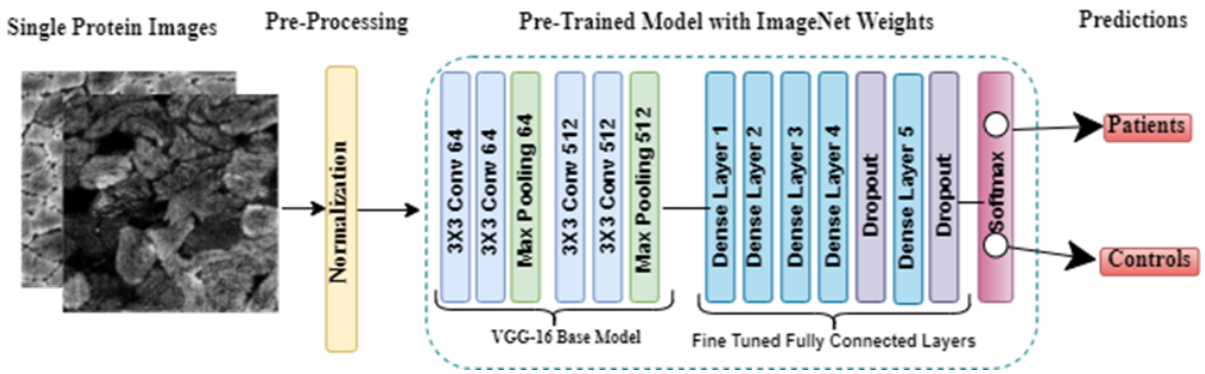

Figure 4: VGG16 architecture for single protein image classification

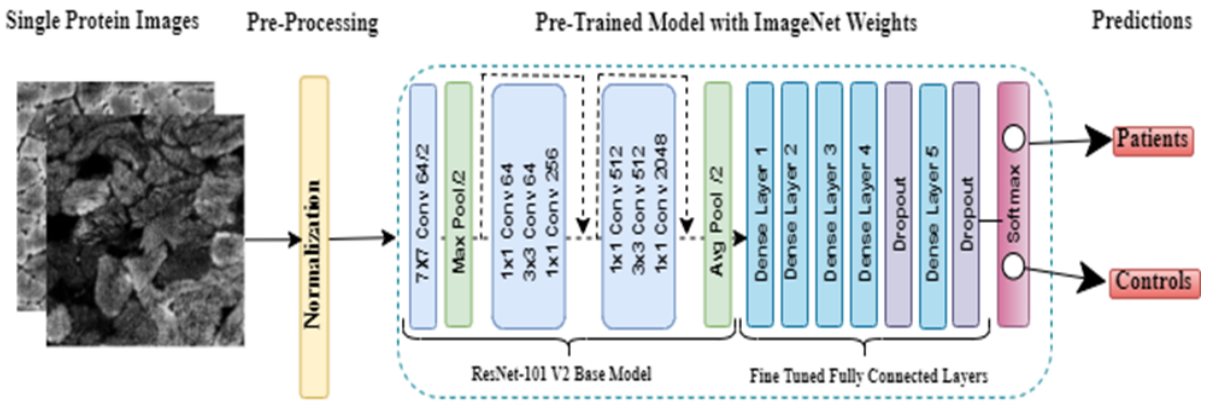

Figure 5: ResNet101v2 architecture for single protein image classification

$$S_c(I) \approx \omega^T I + b \tag{1}$$

$$\omega = \frac{\partial S_c}{\partial I} \tag{2}$$

where ω is the derivative of class score $S_c(I)$ with respect to image I and b represents the bias of the model. The magnitude of the derivative ω shows which pixels needs to be changed the least to affect the class score the most. These pixels will most likely correspond to features of interest in the image. The intuition behind producing the saliency map of the image I is as follows: first, the derivative ω is calculated by back-propagation with Equation 2. Then, the saliency map is generated by rearranging the elements of vector ω. If the image is grey-scale, then class Saliency map is computed using Equation 3, where $g(i, j)$ represents the index the element of ω corresponding to pixel in i-th row and j-th column. For a multi-channel RGB images, the salience map is generated by taking the maximum magnitude of ω across all channels (c) and is represented by Equation 4. Figure 6 show the implementation of class saliency map on VGG16 model.

$$S_{map(i,j)} = \left| \omega_{g(i,j)} \right| \tag{3}$$

$$S_{map(i,j)} = max_c \left| \omega_{g(i,j,c)} \right| \tag{4}$$

3.4 Multi-Channel Image Classification Model Architecture

In the third phase of our experiment, we combine all 10 protein expression images from each patient and control data into a 10-channel image and train a CNN model using transfer learning with these multi-channel images. All the

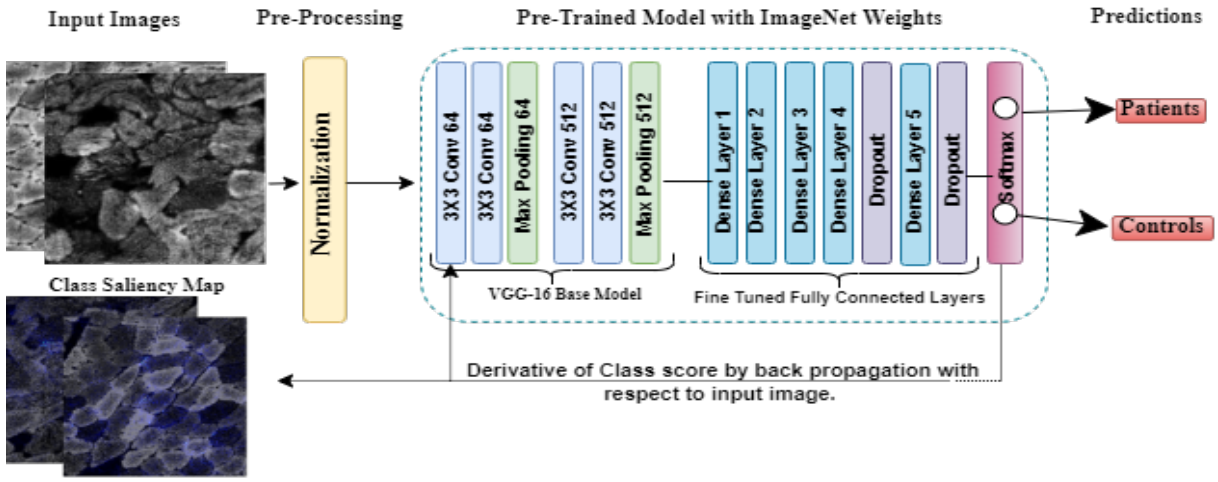

Figure 6: Class saliency Map implementation on VGG16 model

existing pre-trained models, such as VGG, and ResNet, which have been used for single protein image classification, take an input image with a maximum of 3 channels and have the ImagNet weights on these channels. We have modified the existing pre-trained VGG16 and VGG19 architecture to accommodate the extra channels. First, we define the image's height, width, and channels as 224, 224, and 10 in the network's input layer. When the number of channels is increased to 10, the trainable parameters in the first convolution layer are also increased to 5824 from 1792 in 3 channel model. The rest of the trainable parameters in the remaining convolution layers remain the same as in the original models. The additional 7 channels that we have added would have random weights assigned, so we must replace them with ImageNet weights to use transfer learning. For this, we average the ImageNet weights in the kernels of the first three channels and copy the averaged weights to all the individual 7 channels, excluding the first three channels, which already have the ImageNet weights from the original model. All the top layers are frozen, and the same dense layers are added, which is used in the single protein image classification CNN models, as explained above. The modified VGG16 architecture for multi-channel image classification is shown in Figure 7.

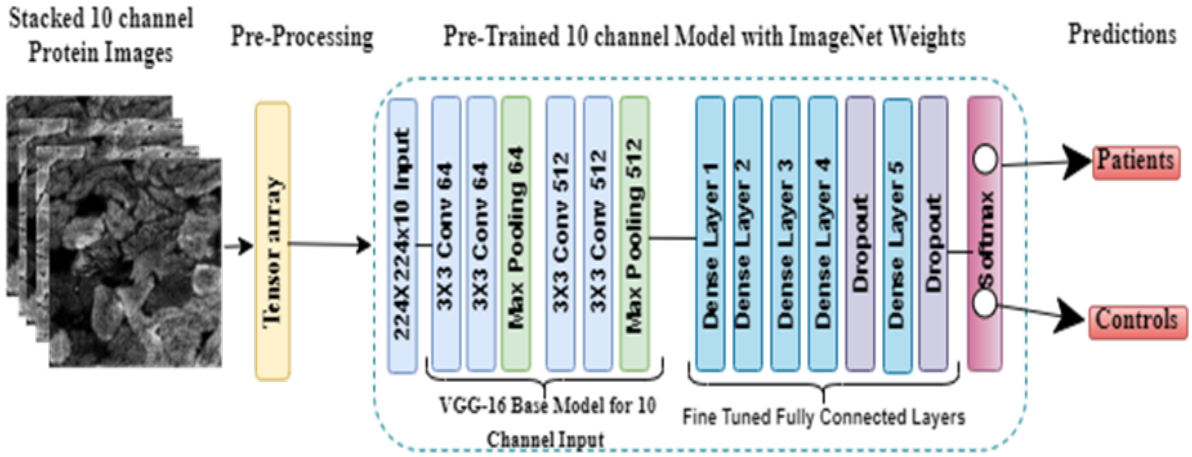

Figure 7: Modified VGG16 architecture for 10-channel image classification

3.5 Environment and Implementation

The project was implemented using Python with open-source machine learning libraries such as Keras and TensorFlow. As our dataset is images, we require GPU to train our models efficiently. We used both physical hardware and a cloud-based environment to execute this project. For training the single protein image classification model, we used physical hardware with a configuration of core i7 processor, 16GB of memory, and NVIDIA RTX 3050 graphic processor. For pre-processing multichannel protein images and training the model, we used the Google collab cloud

environment which gave us better GPU performance and reduced training time. The collab environment was equipped with a Tesla T4 graphic processor and a shared memory of 16GB.

The objective of Phase 1 was to build a binary classification model for single protein image classification into patient and control classes. Existing pre-trained CNN models such as VGG16 and ResNet101v2 were fine-tuned and implemented with transfer learning. The base model was frozen with ImageNet weights, and the fully connected layers were optimized by the trial-and-error method to achieve the best model performance. The CNN architecture implemented for single protein image classification is shown in Figure 4 and 5. The sole purpose of using the CNN classification model is to understand the features used by the network to differentiate between patients suffering from mitochondrial disease (Patient class) and healthy individuals (Control class). So, in the project's 2nd phase, we have implemented class saliency map on single protein images, which is an explainable method to visualize where the network is looking to make the predictions. The implementation of the class saliency map is explained in Figure 6. In the 3rd phase of the project, all the 10 protein images were combined into a 10-channel image, and the existing fine tuned VGG16 and VGG19 CNN model used for single protein image classification was modified to accommodate for multi-channel images and transfer learning was implemented to classify the images into patient and control class. The model architecture is explained in Figure 7. The results of each phase are discussed in section 4 below.

4 Results and Discussion

As discussed above the experiments were conducted in three phases. In the first phase of the experiment, we built a pipeline to classify the single protein expression images into patients and control classes using pre-trained CNN with ImageNet weights. There were a number of architectures explored such as VGG, ResNet and InceptionNet to efficiently classify the images. By fine-tuning the fully connected layers by trial-and-error method and modifying the hyper parameters while training the model we achieved an accuracy of above 85% on VGG16 and ResNet101v2 models.

In the next phase of our experiment, we applied class saliency map on the single protein expression models that is VGG16 and ResNet101v2 models that were trained in the first phase of the experiment. This helped in visualizing the pixels used by the network to make its predictions and aim to understand the underlying pathology of mitochondrial disease.

In the final phase our experiment we stacked all the protein images of the individual control and patient data in to 10-channel image. We modified the VGG16 and VGG19 to train with multi-channel images and classify the images into patient and control. The model was trained multiple times while iterating over different input layer configurations and hyperparameter settings to achieve an accuracy of above 90% on both the models. The result of each experiment is discussed in detail below.

Phase 1: Single Protein Image Classification

VGG16 and ResNet101v2 models using transfer learning to classify single protein images into patient and control classes were trained in the project's first phase. The top layers were frozen with ImageNet weights and the fully connected layers are fine-tuned in the model as discussed in section 3.2. The models were trained on TOM22 protein jpeg images from all the patient and control data. The data was split into training, validation, and test sets in $80:10:10$ ratio, respectively as shown in Table 1. The images were resized to 224 X 224 pixels and normalized before training. As the images need to be classified into patient and control classes, categorical cross entropy was used to encode the class labels into integers i.e., 0 for controls and 1 for patients. Adam, an efficient, memory-optimized, and widely used stochastic optimizer in deep learning computer vision has been used with the learning rate set to 0.001. The batch size was set to 2 as the data was small, allowing the model to train slowly and learn the features efficiently. A summary of hyper-parameters used is shown in Table 3. The model was set to train for 100 epochs with an early stopping set on validation accuracy for no improvement after 20 consecutive epochs.

VGG16 had better training and validation accuracy compared to ResNet 101v2 model. The VGG16 model stopped training after the 46th epoch as the validation accuracy stabilized, and there were no further improvements. To validate if the models are under-fitting or over-fitting, loss and accuracy curves with respect to epochs were plotted for all the models, as shown in Figure 11 and 12 in the appendix. VGG16 had a good fit compared to all models as the accuracy increased for both training and validation set at nearly the same rate, with loss decreasing simultaneously over the training epochs.

To quantify the performance of the model's; Precision, Recall, F1-scores, and Accuracy metrics are calculated using Equations 5 - 8 in appendix. The summary of the metrics for all the models is shown in the Table 4. Confusion matrices were also generated for all the models with 124 images of the test dataset, as shown in Figure 13 in the appendix. The accuracy of VGG16 and ResNet101v2 is 95% and 89% respectively. As the data is imbalanced, it is better to use

Hyperparameters	Values
Loss Function	Categorical Cross Entropy
Optimizer	Adam
Learning Rate	0.001
Max Epochs	100
Early stopping patience Epochs	20
Batch Size	2

Table 3: Hyperparameters used in training the models

F1-scores to compare the model performance, and we see the F1 score of VGG16 is 0.94 and 0.87 for ResNet101v2. With the VGG16 model having a better F1-score, we use this model in the next phase of our analysis.

Model	Precision	Recall	F1-Score	Accuracy
VGG16	0.96	0.95	0.94	0.95
ResNet101v2	0.87	0.86	0.87	0.89

Table 4: Model performance parameters for single protein image classification models

The predictions of VGG16 model for some of the TOM22 single protein test images, where they have been classified into their True and Predicted labels, are shown in the Figure 8 below.

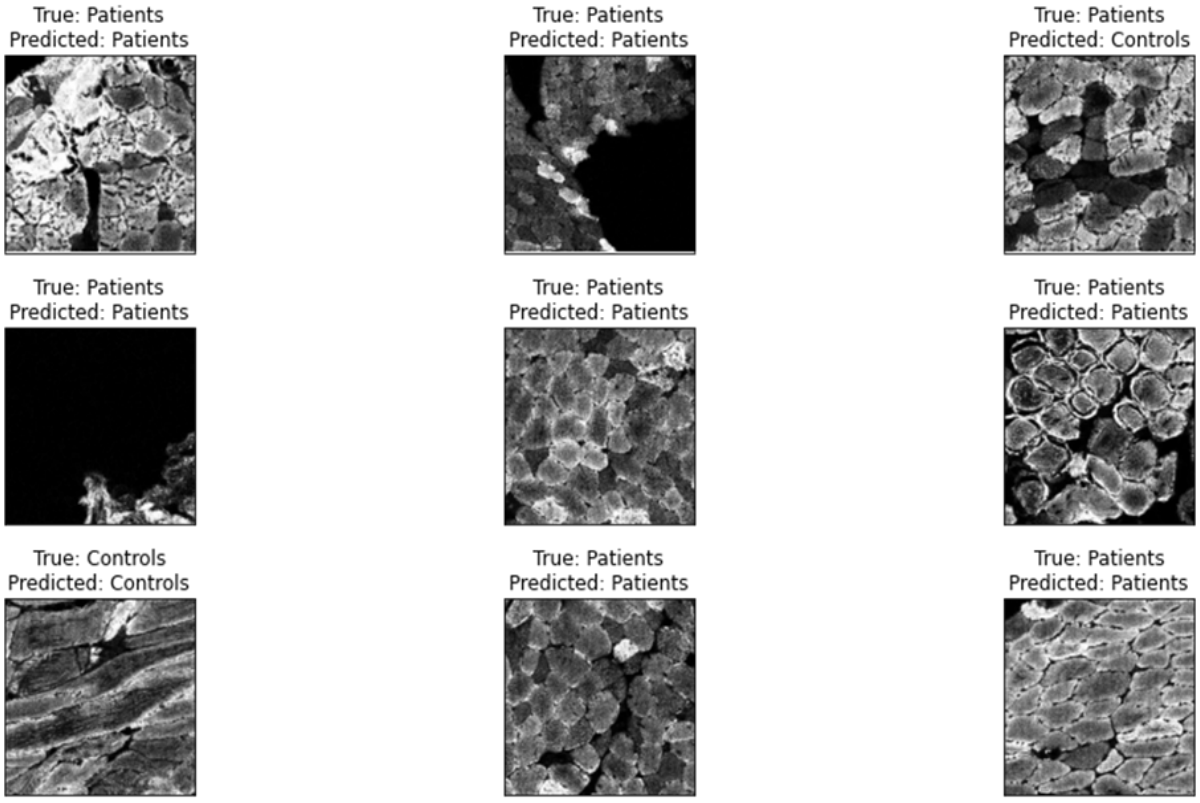

Figure 8: Predictions for TOM22 single protein test images

Phase 2: Class Saliency Maps

With the VGG16 CNN model now trained to classify the protein expression images, the network must be interrogated to see if the right features are being used to make the predictions. Class saliency map is one of the post hoc explainable methods implemented to highlight the pixels used by the network to classify the images. As explained in section 3.3, the main idea is to compute the gradient of the SoftMax output of the VGG16 model with respect to the input. The interpretation of the output is brighter the pixel colour, more the pixel contributes to the prediction. Figure 9 shows the sample output of saliency maps for the VGG16 model of both patient and control test images with their overlay on the original image. The brighter pixels that is dark blue colour in the images represent where the model is paying more attention to predict the class. We can infer from the images that most features are picked from the cell boundaries, as the area near the cell boundaries appear to be illuminated more. To validate the output saliency map, images were shared with the scientists at WMRC for their feedback.

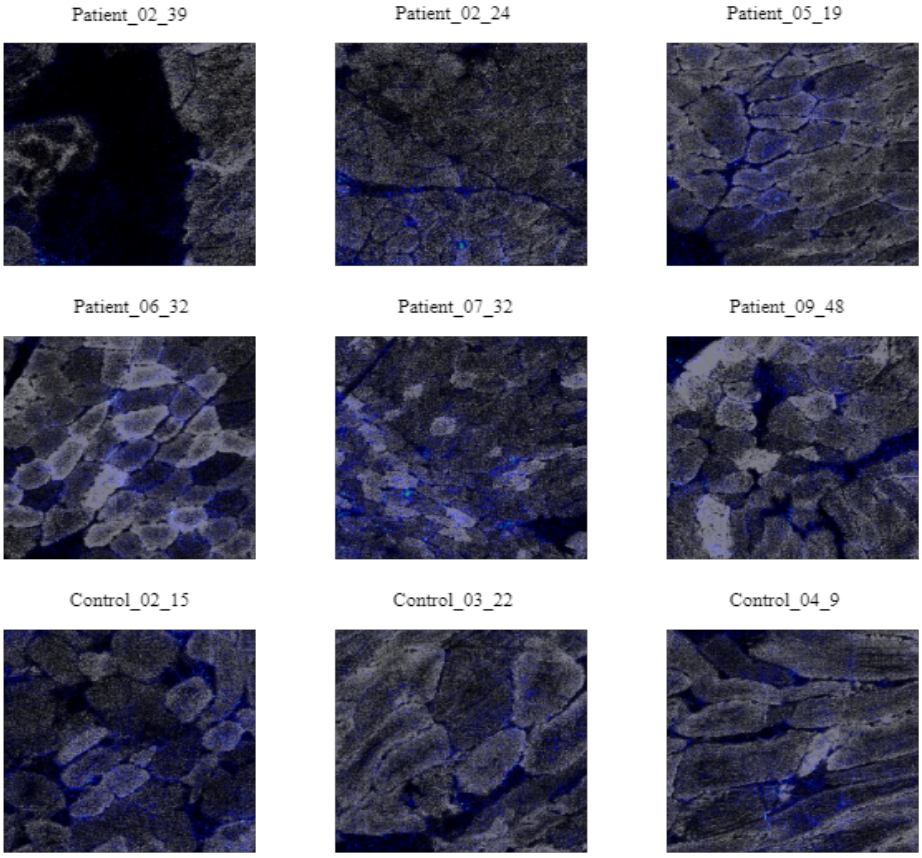

Figure 9: Class Saliency Map implementation on TOM22 single protein images

Phase 3: Multi-Channel Image Classification

To understand the underlying pathology of mitochondrial disease, all the 10-protein expression in combination needs to be analysed for each patient, so all the images are converted into NumPy arrays and concatenated vertically to have a multi-channel image. Each control and patient will be a 10-channel image. The existing pre-trained models VGG16 and VGG19 are modified to take 10-channel images as input and use transfer learning to train the network. The data is split into training, validation, and test sets in $80:10:10$ ratio, respectively, as shown in Table 2. Before the data is fed into the model, each image is converted into a tensor array for training. The hyperparameters used to train the model are the same as the single protein image classification model, summarised in Table 3.

VGG16 had better training and validation accuracy compared to the VGG19 model. After the 64th epoch, the VGG16 model stopped training as the validation accuracy stabilized. Loss and accuracy curves with respect to epochs are plotted and are shown in Figure 14 and 15 in the appendix. To quantify the performance of the model's; Precision, Recall, F1-scores, and Accuracy metrics are calculated using Equations 5-8 in the appendix. The summary of the performance metrics for both models are as shown in Table 5. As the data is imbalanced with more patient data compared to control,

F1 scores are used to compare the model performance, and we see the F1-scores for VGG16 is 0.95 better than VGG19 model, which is 0.92. The confusion matrices generated for the test data of both the models are shown in Figure 16

Model	Precision	Recall	F1-Score	Accuracy
VGG16	0.96	0.96	0.95	0.98
VGG19	0.94	0.91	0.92	0.93

Table 5: Model performance parameters for 10-channel protein image classification models

5 Conclusion

To summarize, in this project we have fine-tuned existing pre-trained CNN networks such as VGG and ResNet Models to classify Protein expression images from IMC into patient and control classes to understand the underlying pathology of mitochondrial diseases with the features extracted by the network. In the project's initial phase, we experimented with VGG-16 and ResNet101v2 models for classifying single protein expression images (TOMM22). The model performance was evaluated using Recall, Precision, F1-score, and Accuracy parameters. The accuracy of VGG16 and ResNet101v2 is 95% and 89%, respectively. As the data is imbalanced F1-score metrics are also considered to evaluate the model performance. VGG16 had a better F1-score with 0.94 compared to ResNet101v2. The best fit for the data is seen from the accuracy and loss function curves of the VGG16 model with respect to the number of epochs.

In the project's next phase, the network is interrogated using explainable and interpretable methods to see if the network is using the right features to make the prediction and analyse the output to understand the underlying pathology of mitochondrial disease. We have implemented class saliency map, a post hoc explainable method for the VGG16 single protein image classification model. The saliency map's output highlights the pixels that contribute to the model predictions. Brighter the pixel intensity, higher is its contribution towards the network prediction and appear to be more concentrated near the cell boundaries.

In the final phase of the project, we combined all the protein expression images into a multi-channel image and trained it with a modified multi-channel VGG16 and VGG19 pre-trained model that takes 10-channel images as input. We see that VGG16 had a better model performance than VGG19, where VGG16 had an accuracy of 98% and F1-scores of 0.95.

The results from the saliency maps focus on highlighting the important pixels in the single protein expression image. These results will help bio-medical scientists to recognise and compare the pattern produced by the saliency map to differentiate from health tissue and a tissue from individual suffering from mitochondrial disease. The results of these approaches were presented to scientists at WCMR and were happy with the results obtained. If the scientists can confirm that the model is picking the right features from the images, we can conclude that the applied interpretability techniques will help in understanding the pathology of mitochondrial disease.

One of the limitations identified while performing the experiments is, there is a need to collect more control samples, as patient images are more compared to control. This can lead to model bias towards patient class which was discussed with scientists while presenting the results. To produce more accurate results, we were suggested by the scientists to removes defects in the images such as folds, holes in tissue samples and over stained regions using segmentation prior to training the model, so that the network would not have any influence on the stray artifacts which can impact the results. This could be one of the proposed future works.

Further work can include the implementation of saliency maps on the 10-channel image classification output to visualize the features exacted in each channel of the image and explore more interpretable methods such as Layer-wise relevance propagation (LRP), Interpretable Image Recognition with Hierarchical Prototypes(HPnet).

Acknowledgments

I would like to thank Atif Khan and Stephen McGough at Newcastle University for the guidance throughout the project. I would also extend my regards to bio-medical scientists at the Wellcome Centre Mitochondrial Research [WCMR], for sharing the data and providing feedback on the work.

References

[1] Salvatore DiMauro. Mitochondrial diseases. *Biochimica et Biophysica Acta (BBA) - Bioenergetics*, 1658(1-2):80–88, July 2004. doi:10.1016/j.bbabio.2004.03.014.

[2] Charlotte Warren, David McDonald, Roderick Capaldi, David Deehan, Robert W. Taylor, Andrew Filby, Doug M. Turnbull, Conor Lawless, and Amy E. Vincent. Decoding mitochondrial heterogeneity in single muscle fibres by imaging mass cytometry. *Scientific Reports*, 10(1), September 2020. doi:10.1038/s41598-020-70885-3.

[3] Philip Siekevitz. Powerhouse of the cell. *Scientific American*, 197(1):131–144, July 1957. doi:10.1038/scientificamerican0757-131.

[4] Laura C Greaves, Amy K Reeve, Robert W Taylor, and Doug M Turnbull. Mitochondrial DNA and disease. *The Journal of Pathology*, 226(2):274–286, November 2011. doi:10.1002/path.3028.

[5] Mauro Scarpelli, Alice Todeschini, Irene Volonghi, Alessandro Padovani, and Massimiliano Filosto. Mitochondrial diseases: advances and issues. *The Application of Clinical Genetics*, Volume 10:21–26, February 2017. doi:10.2147/tacg.s94267.

[6] Kumarasamy Thangaraj, NahidAkhtar Khan, Periyasamy Govindaraj, and AngamuthuKannan Meena. Mitochondrial disorders: challenges in diagnosis & treatment. *Indian Journal of Medical Research*, 141(1):13, 2015. doi:10.4103/0971-5916.154489.

[7] Gráinne S. Gorman, Andrew M. Schaefer, Yi Ng, Nicholas Gomez, Emma L. Blakely, Charlotte L. Alston, Catherine Feeney, Rita Horvath, Patrick Yu-Wai-Man, Patrick F. Chinnery, Robert W. Taylor, Douglass M. Turnbull, and Robert McFarland. Prevalence of nuclear and mitochondrial dna mutations related to adult mitochondrial disease. *Annals of Neurology*, 77(5):753–759, March 2015. doi:10.1002/ana.24362.

[8] Sumit Parikh, Amy Goldstein, Mary Kay Koenig, Fernando Scaglia, Gregory M. Enns, Russell Saneto, Irina Anselm, Bruce H. Cohen, Marni J. Falk, Carol Greene, Andrea L. Gropman, Richard Haas, Michio Hirano, Phil Morgan, Katherine Sims, Mark Tarnopolsky, Johan L.K. Van Hove, Lynne Wolfe, and Salvatore DiMauro. Diagnosis and management of mitochondrial disease: a consensus statement from the mitochondrial medicine society. *Genetics in Medicine*, 17(9):689–701, September 2015. doi:10.1038/gim.2014.177.

[9] A. M. Schaefer, C. Phoenix, J. L. Elson, R. McFarland, P. F. Chinnery, and D. M. Turnbull. Mitochondrial disease in adults: A scale to monitor progression and treatment. *Neurology*, 66(12):1932–1934, June 2006. doi:10.1212/01.wnl.0000219759.72195.41.

[10] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. doi:10.1038/s42256-019-0048-x.

[11] Zohaib Salahuddin, Henry C. Woodruff, Avishek Chatterjee, and Philippe Lambin. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. *Computers in Biology and Medicine*, 140:105111, January 2022. doi:10.1016/j.compbiomed.2021.105111.

[12] M Ahmad, A Wolberg, and CI Kahwaji. Biochemistry, electron transport chain.[updated 2021 sep 8]. *StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing*, 2022.

[13] Deirdre Nolfi-Donegan, Andrea Braganza, and Sruti Shiva. Mitochondrial electron transport chain: Oxidative phosphorylation, oxidant production, and methods of measurement. *Redox Biology*, 37:101674, October 2020. doi:10.1016/j.redox.2020.101674.

[14] Qing Chang, Olga I. Ornatsky, Iram Siddiqui, Alexander Loboda, Vladimir I. Baranov, and David W. Hedley. Imaging mass cytometry. *Cytometry Part A*, 91(2):160–169, February 2017. doi:10.1002/cyto.a.23053.

[15] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, September 2017. doi:10.1162/neco_a_00990.

[16] Rajesh Godasu, David Zeng, and Kruttika Sutrave. Transfer learning in medical image classification: Challenges and opportunities. *MWAIS 2020 Proceedings*, 2020.

[17] Pronnoy Dutta, Pradumn Upadhyay, Madhurima De, and R.G. Khalkar. Medical image analysis using deep convolutional neural networks: CNN architectures and transfer learning. In *2020 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, February 2020. doi:10.1109/icict48043.2020.9112469.

[18] Weiming Zhi, Henry Wing Fung Yueng, Zhenghao Chen, Seid Miad Zandavi, Zhicheng Lu, and Yuk Ying Chung. Using transfer learning with convolutional neural networks to diagnose breast cancer from histopathological images. In *Neural Information Processing*, pages 669–676. Springer International Publishing, 2017. doi:10.1007/978-3-319-70093-9_71.

[19] Claudia Mazo, Jose Bernal, Maria Trujillo, and Enrique Alegre. Transfer learning for classification of cardiovascular tissues in histological images. *Computer Methods and Programs in Biomedicine*, 165:69–76, October 2018. doi:10.1016/j.cmpb.2018.08.006.

[20] Zohaib Salahuddin, Henry C. Woodruff, Avishek Chatterjee, and Philippe Lambin. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. *Computers in Biology and Medicine*, 140:105111, January 2022. doi:10.1016/j.compbiomed.2021.105111.

[21] Alfredo Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural Computing and Applications*, 32(24):18069–18083, February 2019. doi:10.1007/s00521-019-04051-w.

[22] Arash Shaban-Nejad, Martin Michalowski, and David L. Buckeridge, editors. *Explainable AI in Healthcare and Medicine*. Springer International Publishing, 2021. doi:10.1007/978-3-030-53352-6.

[23] Bas H.M. van der Velden, Hugo J. Kuijf, Kenneth G.A. Gilhuijs, and Max A. Viergever. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, 79:102470, July 2022. doi:10.1016/j.media.2022.102470.

[24] T. Nathan Mundhenk, Barry Y. Chen, and Gerald Friedland. Efficient saliency maps for explainable ai, 2019. URL: https://arxiv.org/abs/1911.11293, doi:10.48550/ARXIV.1911.11293.

[25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013. URL: https://arxiv.org/abs/1312.6034, doi:10.48550/ARXIV.1312.6034.

[26] Bob D. de Vos, Jelmer M. Wolterink, Tim Leiner, Pim A. de Jong, Nikolas Lessmann, and Ivana Isgum. Direct automatic coronary calcium scoring in cardiac and chest CT. *IEEE Transactions on Medical Imaging*, 38(9):2127–2138, September 2019. doi:10.1109/tmi.2019.2899534.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. doi:10.1109/cvpr.2016.319.

[28] Hongyang Jiang, Kang Yang, Mengdi Gao, Dongdong Zhang, He Ma, and Wei Qian. An interpretable ensemble deep learning model for diabetic retinopathy disease classification. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, July 2019. doi:10.1109/embc.2019.8857160.

[29] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, October 2017. doi:10.1109/iccv.2017.74.

[30] Paul Windisch, Pascal Weber, Christoph Fürweger, Felix Ehret, Markus Kufeld, Daniel Zwahlen, and Alexander Muacevic. Implementation of model explainability for a basic brain tumor detection using convolutional neural networks on MRI slices. *Neuroradiology*, 62(11):1515–1518, June 2020. doi:10.1007/s00234-020-02465-1.

[31] Quan shi Zhang and Song chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, January 2018. doi:10.1631/fitee.1700808.

[32] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. doi:10.48550/ARXIV.2002.01650.

[33] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition, 2020. URL: https://arxiv.org/abs/2012.02046, doi:10.48550/ARXIV.2012.02046.

[34] Peter Hase, Chaofan Chen, Oscar Li, and Cynthia Rudin. Interpretable image recognition with hierarchical prototypes, 2019. URL: https://arxiv.org/abs/1906.10651, doi:10.48550/ARXIV.1906.10651.

[35] Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y. Lo, and Cynthia Rudin. Iaia-bl: A case-based interpretable deep learning model for classification of mass lesions in digital mammography, 2021. URL: https://arxiv.org/abs/2103.12308, doi:10.48550/ARXIV.2103.12308.

[36] Mohammed AlQuraishi and Peter K. Sorger. Differentiable biology: using deep learning for biophysics-based and data-driven modeling of molecular mechanisms. *Nature Methods*, 18(10):1169–1180, October 2021. doi:10.1038/s41592-021-01283-4.

[37] Zachary C. Lipton. The mythos of model interpretability, 2016. URL: https://arxiv.org/abs/1606.03490, doi:10.48550/ARXIV.1606.03490.

[38] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017. URL: https://arxiv.org/abs/1702.08608, doi:10.48550/ARXIV.1702.08608.

[39] Axel Wismüller and Larry Stockmaster. Tracking results and utilization of artificial intelligence (tru-ai) in radiology: Early-stage covid-19 pandemic observations, 2020. URL: https://arxiv.org/abs/2010.07437, doi:10.48550/ARXIV.2010.07437.

A Appendix

A.1 Objectives

- To fine-tune existing convolutional Neural Network (CNN) models with Transfer Learning (TL) to classify single protein expression images into patient (tissue images from patients affected by the mitochondrial disorder) and control (tissue image from a healthy individual).
- Interrogate the CNN models with interpretability and explainability approaches such as saliency maps and neural disentanglement to understand the underlying pathology of mitochondrial diseases.
- To modify the existing pre-trained CNN model to accommodate for stacked 10-channel protein images and train the models to classify the images and evaluate the performance of the model using different interpretability, and explainability AI approaches

A.2 Data Structure

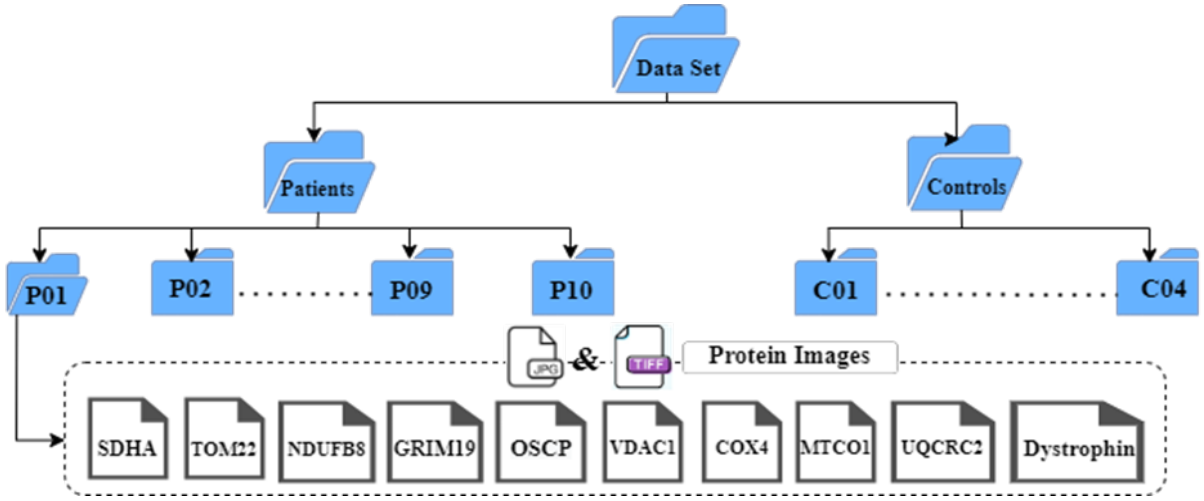

Figure 10: Data Structure

A.3 Evaluation Metrics

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{FN + TP} \tag{6}$$

$$Accuracy = \frac{TP + TN}{FP + TP + TN + FN} \tag{7}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

To evaluate the model performance we use Precision , Recall , Accuracy and F1-score metrics which are calculated from the above equations. Where TP is True Positive, FP is False Positive, TN is True Negative and FN is False Negative

A.4 Results

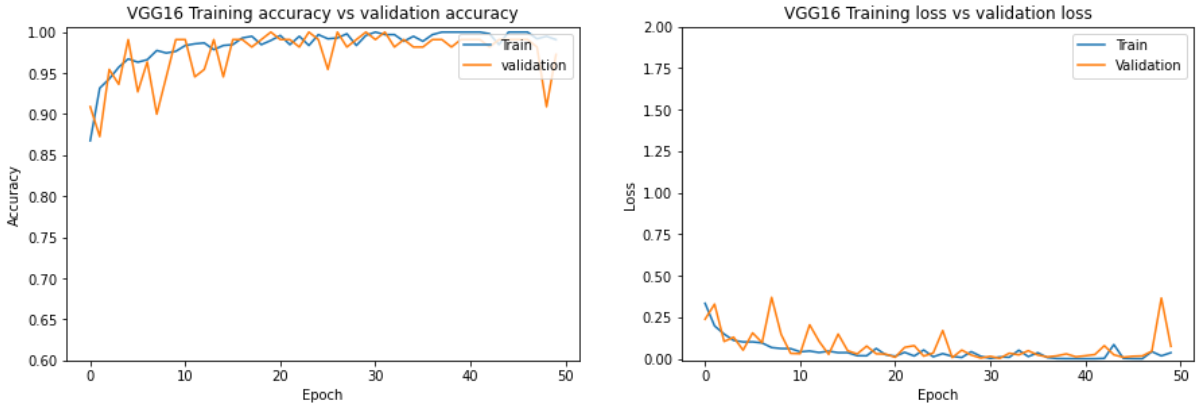

Figure 11: Accuracy and Loss curves for VGG16 single-channel model

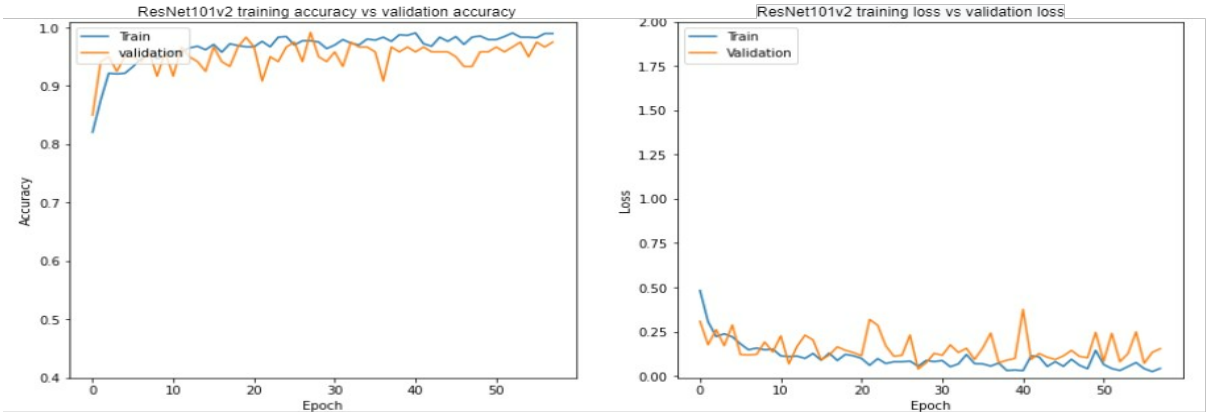

Figure 12: Accuracy and Loss curves for ResNet101v2 single-channel model

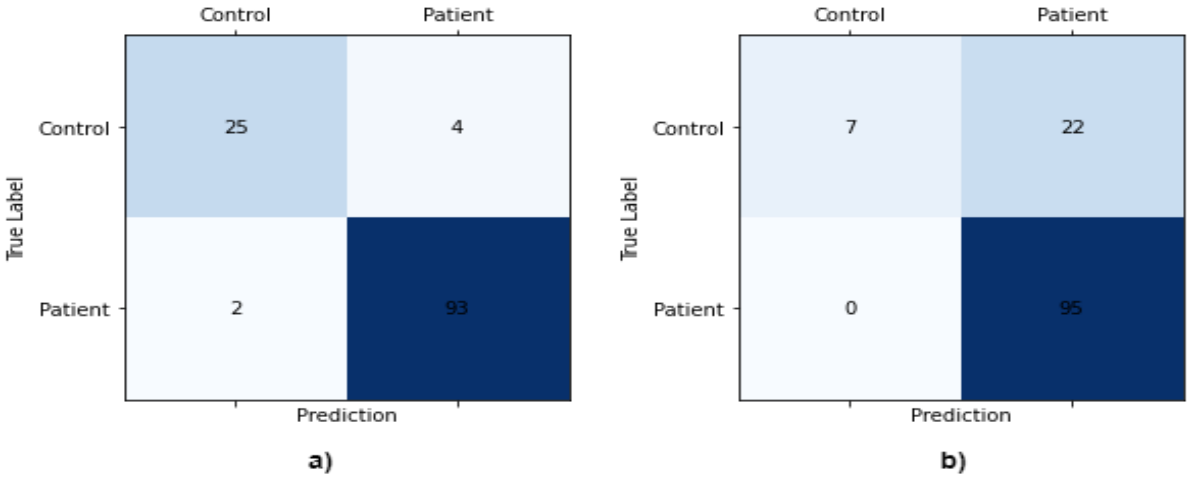

Figure 13: Confusion matrices for test data a) VGG16 single-channel model b) ResNet101v2 single-channel model

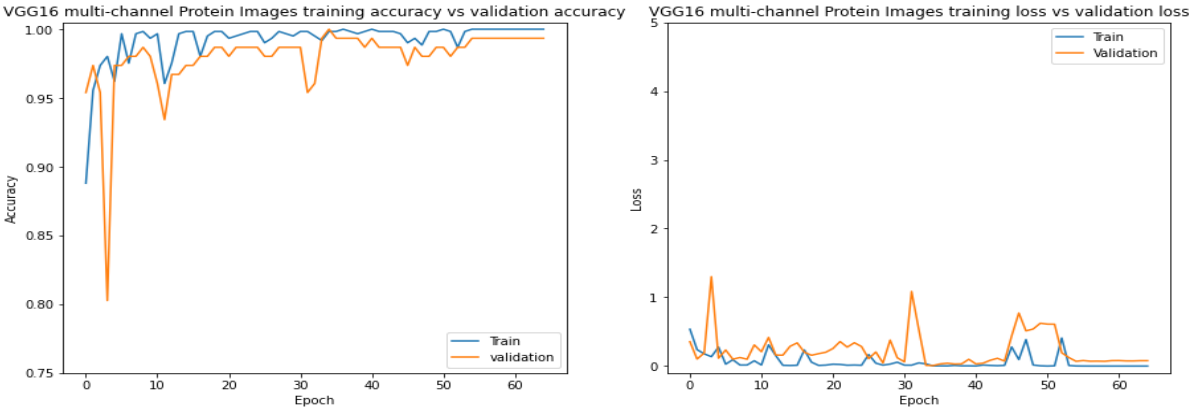

Figure 14: Accuracy and Loss curves for VGG16 10-channel model

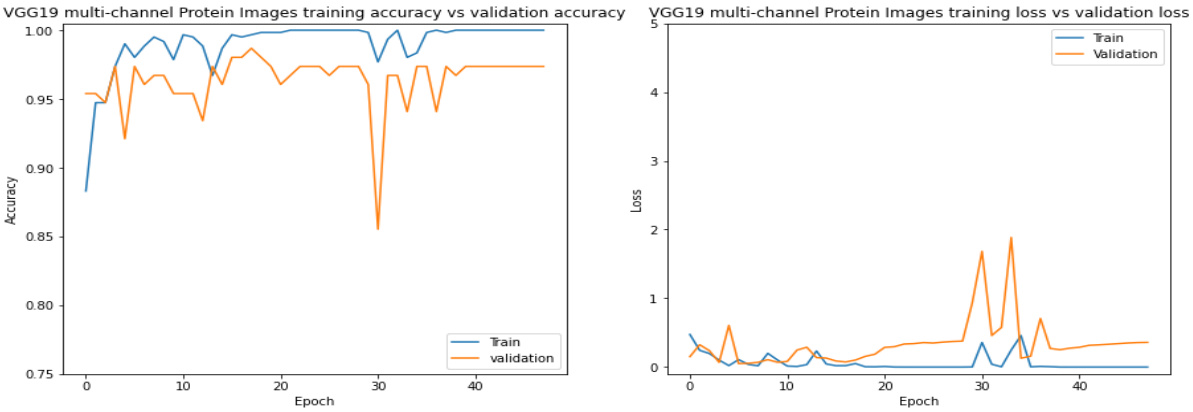

Figure 15: Accuracy and Loss curves for VGG19 10-channel model

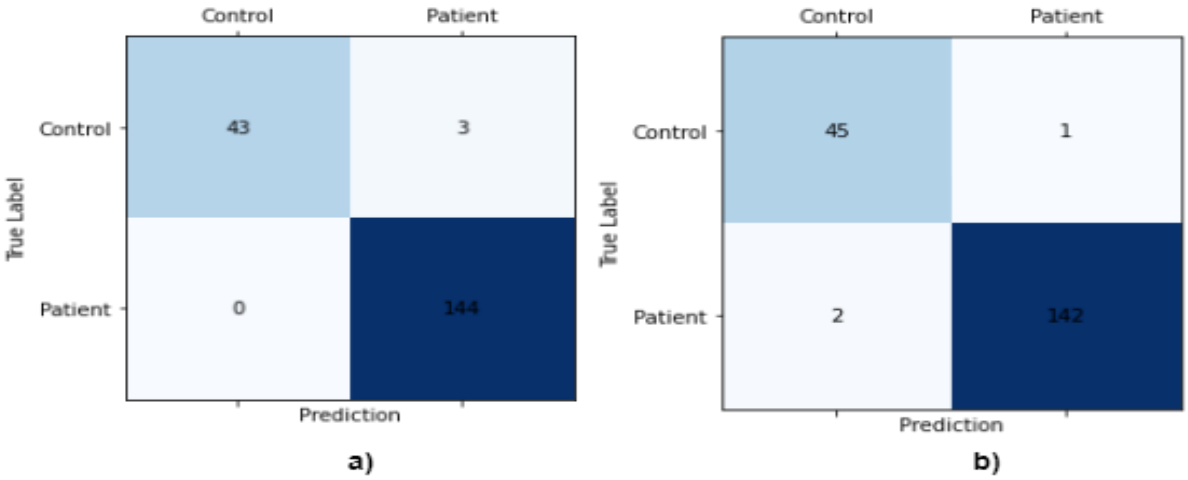

Figure 16: Confusion matrices for test data a) VGG16 10-channel model b) VGG19 10-channel model