AI-assisted Eczema Herpeticum Diagnosis from Digital Atopic Dermatitis Images

Undergraduate Research Opportunity Program (UROP) Summer 2024

Jiho Shin, Yuichiro Minamikawa

Supervisors: Leo Huang, Dr. Wai Hoh Tang, Prof. Reiko Tanaka

Tanaka Lab, Department of Bioengineering, Imperial College London

Contents

1.Introduction

Commented [WHT1]: Use the Justify alignment for the paragraphs. It makes the presentation neater.

Atopic Dermatitis, also known as Eczema, is a chronic skin disease that causes the skin to become red, itchy and inflamed. It occurs in 20% of children and 10% of adults in the developed world [1]. While AD itself is generally not fatal, the complications that arise from the weakened skin barrier and immune dysregulation make the body prone to bacterial and viral infections. One complication is Eczema Herpeticum (EH), caused by the Herpes simplex virus (HSV-1) being in contact with the impaired skin due to eczema.

The symptoms of EH are very noticeable, including monomorphic vesicles, lesions and punched out erosions with hemorrhagic crusts. These are not only painful, but psychologically also very damaging, often affecting self-esteem and wellbeing. Furthermore, mortality rates of EH can be very high, which range from 10% to 50% without treatment, which is heavily reduced by antiviral therapies, specifically acyclovir [2]. The effectiveness of acyclovir is only possible when EH is diagnosed at an earlier stage, ideally within 48 hours. In a study from 2018, 4,655 hospitalized children with EH only had a mortality rate of 0.1%, with most patients having only a minor loss of physical function from illness during hospitalization, due to acyclovir [3]. This shows the extreme importance of a quick diagnosis of EH to minimize mortality.

Commented [WHT2]: What's the minor loss of function?

1.1 Motivation

The high visual clarity of EH and the need to quickly diagnose make EH a very good candidate for Artificial Intelligence (AI) assisted diagnosis, which can be done by the patients at home, or at GPs or emergency services with a digital image, where misdiagnosis as a severe form of AD can be common due to the rarity of EH. This would ensure that EH patients are quickly alerted before conditions worsen and referred to dermatological specialist consultations and treatments.

AI assisted telemedicine has been on the rise, particularly for visually distinct diseases, following the trend of Machine Learning and Deep Learning. For example, EczemaNet is a ML pipeline that can detect severity of Eczema with digital images.[4]. The complications of EH, which is a by result of AD, are deadly, thus we seek to create an extension to EczemaNet that would show warnings when there is suspected EH. In the future, this has the potential to be added to the EczemaNet pipeline as an add-on feature to expand its diagnostic capabilities.

1.2 Project Objectives

The project aims to tackle these four goals:

- Build an accurate Eczema Herpeticum diagnosis model using machine learning

Commented [WHT3]: Include page numbers on all pages (except the cover page).

- Compare the performance of two different approaches – Deep Learning and Traditional Feature Extraction
- Provide explainability and transparency using interpretation techniques – explaining the diagnosis process is crucial in clinical environments
- Provide recommendations for future work by combining merits of both approaches

2.Background

The rise of deep learning and AI has led to an increase in automatic detection of skin diseases [5], such as acne, psoriasis and eczema. Using the deep learning-based Convolutional Neural Network (CNN) method has its advantage on automatically learned complex features and good performance on large datasets, but on the other hand is computationally intense due to its reliance on large datasets and has higher risk of overfitting on small datasets. Moreover, the 'black box' nature of deep learning, being that the features are less interpretable and understandable, remains a barrier for medical diagnosis. [6]. This is an issue for clinical situations such as EH diagnosis, as explainability of the diagnosis is essential.

Commented [WHT4]: Is there a citation for this point?

Commented [WHT5]: The acronym CNN has not been defined yet. Write out the whole phrase when it's introduced.

Commented [WHT6]: What's the high computational demand?

While current approaches for skin diagnosis models are focused on deep learning-based approaches, another possible approach is using the traditional feature extraction method. The traditional feature extraction (TFE) method consists of human-made features that are interpretable such as colors and textures, providing explainability which is helpful in diagnosis. Although CNN-based approaches are widely believed to outperform TFE methods due to their ability to capture complex features that deep learning models can learn, there is a lack of studies comparing the two. Given the importance of the explainable nature of TFE approaches, this study aims to compare the accuracy of these methods and explore their combination to achieve both high accuracy and explainability

3.Methods

Two methods of machine learning were explored for EH diagnosis: Deep learning-based CNN and different combinations of TFE and a pre-trained deep learning model.

3.1 Dataset and split

The entire dataset consisted of the positive EH (222) and negative non-EH (4,271) labelled images. The non-EH images originated from Softened Water Eczema Trial (SWET) + Temperature controlled Laminar Airflow (TLA) (2,008) [7] and Skin Condition Image Network (SCIN) (2,263) images [8]. The positive dataset was labelled as 1, and the negative as 0.

Commented [WHT7]: Let's be consistent and use dataset, i.e. without the spacing.

Commented [WHT8]: Let's use thousand separator for non-year values. It's neat and it makes reading of numbers easier too.

The dataset was split into 80:20 Training: Testing, where the training data underwent 5-fold cross validation. The EH and SWET+TLA were used for training and testing, while the SCIN was only used for testing to ensure the model was not overfitted to SWET+TLA dataset features.

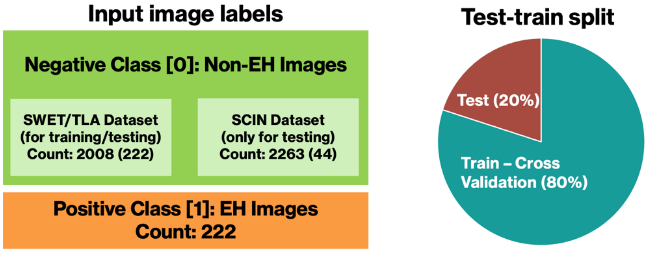


Figure 1. Left – types of data as input, including two datasets for the negative non-EH class and one dataset for the positive EH class. Notice the large class imbalance. Right – training: testing data split

3.2 Evaluation metrics

Metrics used to assess the performance of the model include accuracy, sensitivity, F1 score and Receiver Operating Characteristic (ROC) curve. ROC [9] is a graphical representation used to evaluate the performance of a binary classification model where the others are defined as:

Commented [WHT9]: Move the equations to the Methods section.

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

$$Sensitivity = \frac{TP}{TP + FN}$$

$$F1\,Score = \frac{2 \cdot Sensitivity \cdot Precision}{Sensitivity + Precision}\; where\; Precision = \frac{TP}{TP + FP}$$

3.3 Convolutional Neural Network-based Diagnostic Model

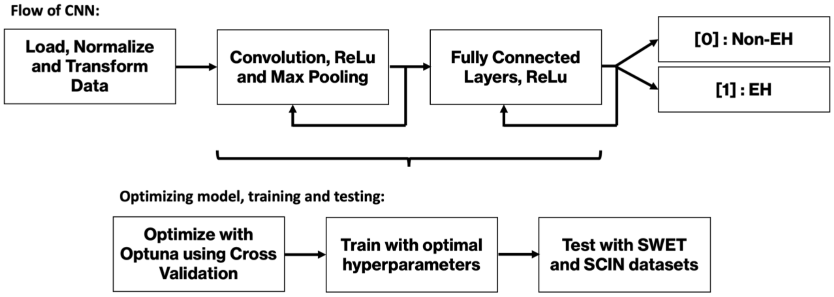

Flow of CNN:
Load, Normalize and Transform Data
Convolution, ReLu and Max Pooling
Fully Connected Layers, ReLu
[0] : Non-EH
[1] : EH
Optimizing model, training and testing:
Optimize with Optuna using Cross Validation
Train with optimal hyperparameters
Test with SWET and SCIN datasets

Figure 2. Workflow of the CNN model and optimization process

The workflow starts with pre-processing the data, then the convolutions and fully connected layers, of which are optimized and trained with Optuna [10]. The optimal hyperparameters are used to train the model which is then tested to predict whether the digital image has signs of EH.

Commented [WHT10]: Provide a citation for Optuna.

During the testing stage, incorrectly classified images are displayed to find artefacts and features of the dataset. This helps with understanding the model and why it is incorrectly classifying certain images, providing insight to how we can improve it further.

3.3.1 Data preprocessing

The first step involved preprocessing the images, including transformations, augmentations and further data splits.

Transformations involved include:

1. Resizing: All images were resized to 512x512 ensure uniform input layers in the CNN layer calculations. Different sizes were examined to optimize accuracy and computational load.
2. Normalization: The standard deviation and mean of the training dataset were calculated, and all data being passed through the pipeline undergo normalization with the standard deviation and mean. This improves speed, convergence, reduces bias to larger scales and allows the scale necessary for activation functions to be used more effectively.
3. Data augmentation: Due to the class imbalance, the positive class underwent augmentation and more SWET/TLA images were sampled to create a larger dataset. Only augmentations that would mimic realistic situations were chosen, including 90°, 180° and 270° degree rotations, Horizontal Flips and Vertical Flips. Other

Commented [WHT11]: What does this mean?

augmentations that would be unrealistic for patients to upload, such as distortion and Gaussian Blur were not included. Data augmentation increases the dataset from 175 to 1,050 images, providing more images for the training process.

For the optimization stage, the 80:20 ratio of training: testing split was used. For the training stage, the training data was further split into 90:10 training: validation, to allow the model to be examined at different points of training.

3.3.2 Optimizing hyperparameters

Optuna was used along with 5-fold cross-validation to optimize the hyperparameters.

During the optimization process, the Imperial High-Performance Computing (HPC) Services were used for training to speed up optimization. The results of the Optuna trials are saved in an SQLite database, which enables training and optimization in batches and for future use. As the number of trials increases, the validation accuracies and sensitivities increased.

Commented [WHT12]: Please provide a table to show the hyperparameters, the corresponding ranges, and settings (e.g. float, log, classes, etc.) in Optuna. If there are some training parameters that are kept fixed, for example number of epochs, do report them too.

Commented [WHT13]: As a rule of thumb, we should write the Methods section such that readers can follow the steps and reproduce the experiments. Image ourselves trying to reproduce an experiment in a paper, we would want to be guided through the steps.

Commented [WHT14]: Use the full phrase of HPC since the acronym is introduced the first time here.

The examined space for the hyperparameters is shown below:

Hyperparameter	Type	Range
Learning rate	Float	loguniform(5e-5, 5e-4)
Optimizer	Categorical	['SGD', 'RMSprop', 'Adam']
Number of Conv Layers	Integer	int(1,6)
Number of FC Layers	Integer	int(1, 4)
Number of Epochs	Integer	Fixed

Table 1: Examined space of hyperparameters

Optuna Visualizations are used to understand the hyperparameter behaviors for the model.

3.3.3 Training, testing and evaluating the model

Once the optimal hyperparameters have been extracted, the model was trained using the 90:10 training: validation split. The model at the epoch with the highest validation accuracy is extracted.

Commented [WHT15]: This was mentioned in Section 3.2.1.

This model then underwent testing with both the SWET/TLA and SCIN dataset. During the testing process, images that were classified incorrectly are displayed to find any artefacts associated with the incorrect classification. The testing results include F1 score, accuracy, sensitivity and Receiver Operating Characteristic (ROC) .

Commented [WHT16]: The acronym ROC is used the 1st time here.

3.4 TFE and pre-trained VGG16 based Diagnosis Method

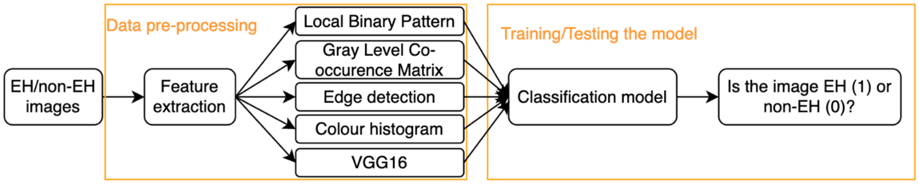


Figure 3. Workflow of the TFE model

The overall flow of building the model is illustrated by Figure 3. The EH/non-EH images undergone pre-processing which includes extracting image features. These features were then used as input to build a classification model to predict whether the input image has EH or not. The detailed procedures are followed.

3.4.1. Data preprocessing

The images were resized to 512x512 and were gone through feature extraction with the following 4 traditional features:

- Local Binary Pattern (LBP)
- Gray Level Co-occurrence Matrix (GLCM)
- Edge detection
- Color histogram

And a single pre-trained deep learning feature:

- VGG16

Note that data augmentation was not performed here to test performance of the TFE model on small dataset. The details of the used features are illustrated in the next section.

3.4.2. The feature extraction process

To preprocess images and extract traditional features, the OpenCV library was used, which is an open-source computer vision and machine learning library [11].

Commented [WHT17]: Mention that OpenCV library is used.

The LBP feature [12] is a texture descriptor that captures patterns in the image by comparing each pixel with its neighboring pixels. To do this, first the image was converted to grayscale and for each pixel, the method compares the intensity of the central pixel with 24 equally separated pixels that draw a circle of radius 8 pixels around the central pixel. If the neighbor's intensity is greater than or equal to the central pixel, it gets a value of 1; otherwise, it gets a 0 these values are then recorded as a 2D array corresponding to each pixel in the image.

Commented [WHT18]: The example in the reference website shows the 8 neighbouring pixels. How do we get the 24 pixels and what is the 8 radius pixels?

The GLCM feature [13] is a global texture descriptor that captures texture by examining how often pairs of pixels with specific values appear across the image. The image is converted to grayscale to focus on intensity values and the *graycomatrix* function computes the co-occurrence matrix, which computes specific measures including *contrast, dissimilarity, homogeneity, energy,* and *correlation.*

The edge detection feature [14] is used to detect object boundaries in an image. The image is converted to grayscale and is applied with the *cv2.Canny()* function which detects edges by finding areas in the image with rapid intensity changes and maps it into a 2D-array. The lower and upper thresholds used for edge detection are 100 and 200.

Commented [WHT19]: Mention that OpenCV library is used.

The Color histogram feature [15] captures the distribution of colors in the image where the *cv2.calcHist()* function computes a 2D histogram for each color channel (Red, Green, Blue) with 8 bins per channel.

The VGG16 [16] is a CNN architecture that has 13 convolutional layers and 3 fully connected layers that are each followed by a Rectified Linear Unit (ReLU) activation function. The early layers detect simple patterns like edges, while deeper layers detect more complex patterns such as shapes and objects that is hard to capture by human-made features. The output feature map is a 3D tensor of size 7x7x512 that captures complex, high-level features of the image.

The following 4 traditional features and 1 deep learning feature are flattened and converted into vectors which can be split into integers to be used as an input for the machine learning processes. Some of the extracted features are illustrated by Table 2.

image_labels	vgg16_0	vgg16_25087	color_histogram_0	color_histogram_511	edge_0	edge_255	lbp_0	lbp_25	glcm_0	glcm_4	label
1	0.0	0.0	0.357	0.001	597837	59235	29886	495249	72.247	0.979	1
2	0.0	0.0	0.015	0.013	589861	67211	25236	432580	100.236	0.973	1
3	0.0	0.0	0.56	0.014	599242	57830	28786	439052	66.943	0.99	1
4	0.0	2.575	0.081	0.0	8937	667	352	5306	103.653	0.953	1
5	0.0	0.0	0.759	0.012	194123	9927	6877	139280	131.873	0.98	1
6	0.0	0.0	0.0	0.001	647372	9700	23851	452627	27.099	0.991	1
7	0.0	0.0	0.05	0.005	65566	5582	1984	39184	128.598	0.965	1
8	0.0	0.0	0.549	0.0	637765	19307	25975	420151	42.835	0.983	1
9	0.871	0.0	0.04	0.002	299696	7504	10268	213248	38.459	0.982	1
10	0.0	0.0	0.207	0.242	625430	31642	23140	346924	52.303	0.992	1

Table 2. Table of extracted features of 10 EH images labelled as 1, only the first and last column for each feature are displayed for presentation; All features are used to train the model.

3.4.3 Model training

Having the extracted features that are single integers, they are first standardized using *fit()* and transform*()* functions from sklearn.preprocessing.*StandardScalar* class that computes the mean and standard deviation of the input feature data and standardize them to have mean of 0 and standard deviation of 1.

These standardized features were used as an input to train a *RandomForestClassifier* model [17]. To improve performance, *Stratified K-fold cross validation* [18] technique was implemented where the training data was divided into 5 folds where one of the folds are used as the validation set while the others are used as training set. This process is repeated 5 times and each time using a different fold for validation. For hyper parameter tuning, *GridSearchCV* [19] was implemented to optimize the number of trees in the forest, maximum depth of the trees, minimum number of samples required to split an internal node, minimum number of samples required to be at a leaf node, and whether to use bootstrap samples. The cross-validation step reduces the likelihood of overfitting, and the grid search step optimizes the model's hyperparameter allowing it to achieve higher accuracy.

4.Results and Evaluation

4.1 Optuna Results

The optimized hyperparameters, parameter numbers and a summary of the CNN model is shown below.

Hyperparameter	Optimized Value
Learning rate	0.000233
Optimizer	RMSProp
Number of Conv Layers	4
Number of FC Layers	3
Number of Epochs	10 Fixed)

(a)Values of optimized hyperparameters

Parameter	Number
Weights	12,341,280
Biases	602
Total	12,341,882

(b)Numbers of parameters

Layer Type	Output Shape	Parameter Count
Conv2d	[-1, 16, 508, 508]	1,216
ReLU	[-1, 16, 508, 508]	0
MaxPool2d	[-1, 16, 254, 254]	0
Conv2d	[-1, 32, 250, 250]	12,832
ReLU	[-1, 32, 250, 250]	0
MaxPool2d	[-1, 32, 125, 125]	0
Conv2d	[-1, 64, 121, 121]	51,264
ReLU	[-1, 64, 121, 121]	0
MaxPool2d	[-1, 64, 60, 60]	0
Conv2d	[-1, 128, 56, 56]	204,928
ReLU	[-1, 128, 56, 56]	0
MaxPool2d	[-1, 128, 28, 28]	0

Linear	[-1, 120]	12,042,360
ReLU	[-1, 120]	0
Linear	[-1, 120]	14,520
ReLU	[-1, 120]	0
Linear	[-1, 120]	14,520
ReLU	[-1, 120]	0
Linear	[-1, 2]	242

(c)Summary of the CNN, including layer types, outputs of the layers and parameters.

Table 3: Three tables showing the hyperparameters, parameters and CNN structure

Below are three visualizations that capture the importance of different hyperparameters and relationships between different combinations.

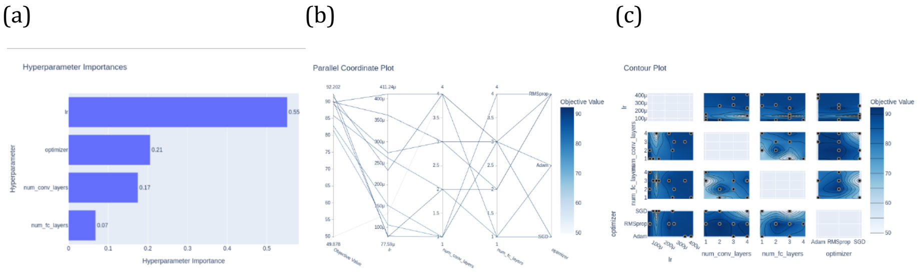


Figure 4. (a) – Shows the relative importance of hyperparameters, with learning rate being the most significant, and number of FC layers being least important. (b) – Parallel Coordinate Plot, showing the different hyperparameters of each trial and its objective value. (c) – Contour plot, showing the relationship between two hyperparameters

Through optimization with different input sizes, it was found that 512x512 offered maximum accuracy with the most efficient computational load. Anything more, the accuracy started to plateau, and computational efficiency is compromised.

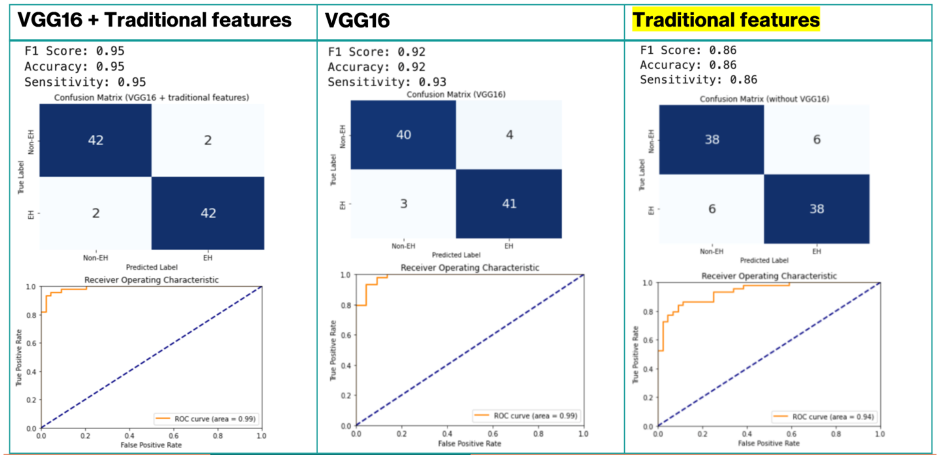
Commented [WHT20]: This should be in the discussion.

Augmenting the data, which corresponded to an increase in the training dataset size from 175 to 1050, resulted in an increase in test accuracy and sensitivity, as shown below:

Dataset Size:	175 (Without augmentation)	1050 (With augmentation)
Accuracy:	69.3%	83.2%
Sensitivity:	45.5%	80.1%

Table 4: Summary of the change in training due to an increase in dataset size by augmentation

4.2 Results comparison for model built on different combinations of traditional features and VGG16 model.



VGG16 + Traditional features

```
F1 Score: 0.95
Accuracy: 0.95
Sensitivity: 0.95
```

VGG16

```
F1 Score: 0.92
Accuracy: 0.92
Sensitivity: 0.93
```

Traditional features

```
F1 Score: 0.86
Accuracy: 0.86
Sensitivity: 0.86
```

Figure 5. Left: Model built using deep learning (VGG16) and traditional features combined; Middle: Model built using deep learning (VGG16) feature alone; Right: Model built using traditional features alone.

Different results using different combinations of features for the TFE model are displayed in Figure 5. The performance is best when traditional features are combined with VGG16 features, and poorest when traditional features are used alone. This indicates how pretrained deep learning VGG16 features outperform traditional features and at the same time combining them gives the best performance with 95% accuracy.

4.3 Traditional Features visualization

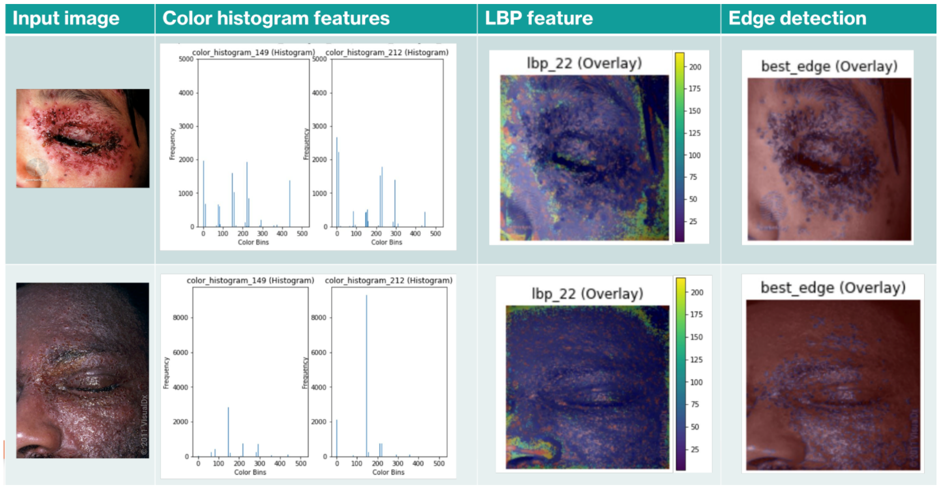


Figure 6. Traditional features visualization of two EH images with different skin colors. Above: EH patient with white skin, and Below: EH patient with darker skin.

To visualize how the traditional features work to diagnosis input image, Figure 6 displays selected features according to feature importance [20]. There are two color histogram features, one LBP and edge detection feature visualized. Note that GLCM features are excluded due to its minimal importance and VGG16 feature requires separate techniques (eg. Grad-CAM [21]) for visualization.

Commented [WHT21]: Include a citation.

According to Figure 6, color histogram features are successfully capturing color distributions within the image which turned out to be the most important features to diagnosis EH. While the LBP feature is performing well with the white skin to indicate EH lesions, for the dark skin, it is not due to EH appearing differently for different skin types which was not successfully captured. On contrary, the edge detection feature is performing well for both skin types to capture the EH lesion areas. This leaves area for improvement by manually altering feature weightings to correctly diagnosis EH for different skin types.

4.4 Result comparison using CNN model and TFE model

	CNN (SWET)	CNN (SCIN)
F1 Score	0.83	0.83
Accuracy	0.83	0.84
Sensitivity	0.80	0.80

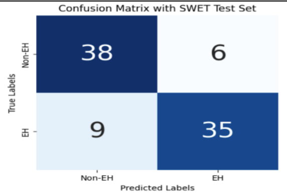
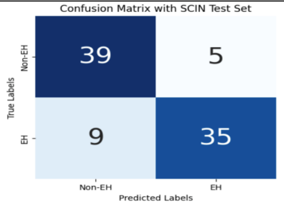
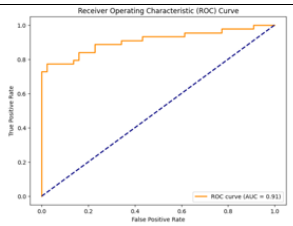
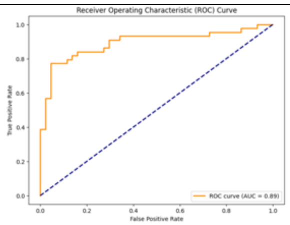
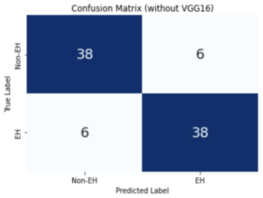
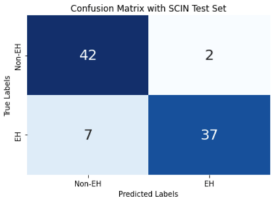
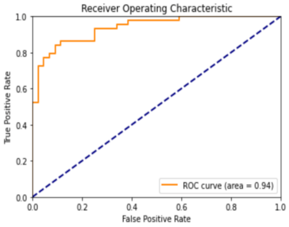
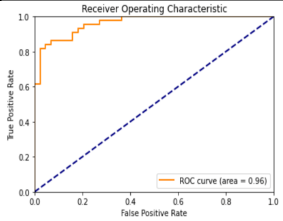
Confusion Matrix		
ROC Curve		
	TFE (SWET)	**TFE (SCIN)**
F1 Score	0.86	0.89
Accuracy	0.86	0.90
Sensitivity	0.86	0.84
Confusion Matrix		
ROC Curve		

Figure 7. Results comparing the testing of CNN and TFE models using SWET/TLA and SCIN test datasets

As shown in Figure 7, TFE yields better results than deep learning, while also providing interpretability to the model. This shows that models that include human-selected features perform better – leaving potential for combining these with the CNN. However, the CNN model still has room for improvement by running more trials with Optuna and exploring a larger hyperparameter space, as well as incorporating transfer learning.

Commented [WHT22]: How is this demonstrated in the results, i.e. a comparison of training with and without a pre-trained model?

Commented [WHT23]: It will be interesting to report the training time for both CNN and TPE. That will provide readers an idea how many experiments can reasonably be done.

In addition, the similarity in accuracy and sensitivity results between SWET and SCIN show that the two models do not overfit to a particular feature of the SWET dataset.

The presence of a high False Negative Rate (FNR) for both models and datasets, i.e. 9 for CNN, and 6 and 7 for TPE, is of concern as misdiagnosing positive EH cases can result in fatalities.

Upon displaying the misdiagnosed images with the CNN, it was shown that the less severe EH images were being predicted as non-EH. Some images in the positive EH dataset are visually very mild forms, which can confuse the model and cause it to predict it as a non-EH image. Below are some examples of the images that were incorrectly classified. Below are two images that were incorrectly classified. Compared to other EH images, these are visually much milder and have smaller EH regions, which can confuse the model.

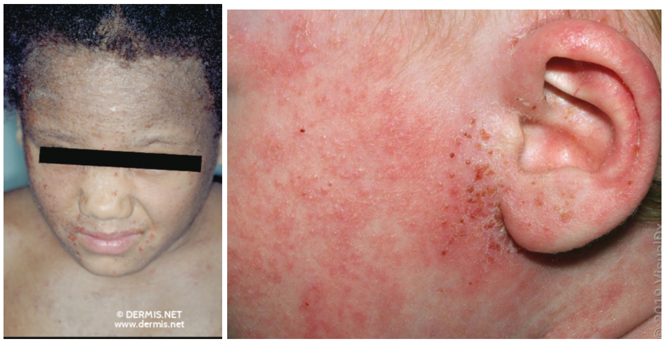


Figure 8. Positive EH images that were incorrectly classified as negative (False Negative)

The time taken to train the models were as follows. The optimizing time is significantly longer than the train time, due to 5-Fold Cross Validation and the many number of trials. 10 trials of optimization were done at a time. Note that the deep learning model was significantly sped up by using the HPC clusters at Imperial College London.

Model	TFE	Deep Learning (10 epochs)	Deep Learning with Augmentation (10 epochs)
Optimize time		~1 hour	~5 hours
Train time	~1 hour	~2 minutes	~10 minutes

Table 5: A comparison of the time taken for training and optimizing (only for deep learning).

5. Discussion

The biggest challenge is the limited size of the dataset, with different types of images being in the dataset. This variation ranges from skin color, body part and magnification of the EH regions. This comes with the risk of overfitting to certain artefacts of the dataset, or even having an ineffective model due to the wide variety and small amounts of each type of data.

In addition, there is a wide range of EH severity in the data, which can often confuse the model as some EH images look like a non-EH AD image. In order to tackle this problem, the positive class was augmented to create more versions of the data for the CNN model, to utilize the larger negative dataset. This helped test accuracy increase from 69.3% to 85.2%. Transfer Learning for the CNN model can be used to utilize a model that has been trained on a similar large dataset, such as a skin dataset. The ideal trajectory for this project would be, to increase the dataset size and potentially separate the EH images based on severity, through examination by a dermatological professional. This will allow the model to examine EH as a spectrum of severity, which can prevent the model from being confused due to the range of severity in EH. Furthermore, sectioning or standardization of EH magnification will be very helpful, to prevent the model from picking up size and magnification related artefacts and reducing complexity of data to improve accuracy [22] .

Commented [WHT24]: We need to include a figure or table in the Results section, i.e. the training with or without the data augmentation.

Commented [WHT25]: Is this an opinion or a results from a paper? We'll need a citation if it's from a paper.

According to Figure 5, the model built using VGG16 alone resulted in 92% accuracy, and 95% when combined with traditional features. This is a better result than using the CNN model which had 86% accuracy, indicating how implementing pre-trained model such as VGG16 can benefit the model by initializing the weights. As the best result were seen when traditional features were combined with pre-trained VGG16 features, there is a very promising outlook for combining pre-trained CNN models, TFE, and further data augmentation.

The black box issue for deep learning remains unsolved, creating difficulty in medical diagnosis where explanation of results is crucial. This was the primary reason for exploring Traditional Feature Extraction, as features and trends can be found that can aid diagnosis and consultation with patients. To approach this with the deep learning method, a way of finding wrongly classified images is introduced, to extract any features and trends that can result in incorrect predictions. It was seen that the less severe EH images were responsible for the incorrect predictions. A future approach to make EH prediction as a spectrum or probability rather than a binary classifier, and to apply GradCAM. GradCAM is a tool that creates a localization map highlighting the extent that features in an image contribute to the prediction [23].

Another big barrier to these models were the variability in skin color. With TFE, different skin colors required different features to point out the EH regions. For example, with darker

Commented [WHT26]: Which model? TPE or CNN?

skin, the color feature did not provide as much information compared to lighter skin. Manual altering of feature weightings can further improve the model's performance on different skin colors. If the dataset gets larger, with more datasets of non-white EH, it is proposed that a separate deep learning model can be created for different skin colors, using a skin color classifier such as Fitzpatrick's Skin Phenotypes, due to the high variability in visual features depending on skin color. This would make the model specific to the situation and be more accurate.

In conclusion, to improve the model, we can include:

Commented [WHT27]: Are there citations for the suggested methods?

1. Probability/spectrum-based labelling and prediction to recognize that EH visual features range with severity
2. Easing the criteria for positive prediction, by optimizing with respect to sensitivity, or reducing the threshold for predicting can be considered. In medical situations, a false negative is better than a false positive [24]
3. Consider transfer learning for CNN using pretrained models and feature extraction, incorporating elements of TFE and deep learning to create a hybrid system with higher accuracy
4. Use GradCAM to better understand the deep learning model and provide interpretability [23]
5. Gather a larger dataset or apply further data augmentation techniques

Commented [WHT28]: What does this mean?

6. Reflection

Commented [WHT29]: 👍👏

6.1 Yuichiro

Initially, getting used to Git/GitHub, using the remote servers and the HPC, and file management were challenging. After weeks of hands-on practice and learning from mistakes, I was able to become fluent in these areas and was able to utilize them effectively.

The workflow for the UROP was as follows:

<u>Research and Learning Stage:</u>

Week 1: Read about AD, EH, current deep learning approaches to skin disease diagnosis and PyTorch.
Week 2: Started writing a CNN in PyTorch, organised files and papers, looked through dataset, and learnt how to use Git version control and the terminal commands

<u>Coding and implementation:</u>

Week 3: Data augmentation, started using JupyterHub, getting used to the RDS, file management, sorting datasets and structuring directories.

Week 4: Setting up a virtual environment, using anaconda, started training the CNN and manually exploring hyper parameters such as learning rate and number of epochs.
Week 5: Used Optuna to optimise hyperparameters, learnt how to use the HPC, Optuna visualisations, incorporated more performance metrics, implement cross validation
Week 6: Continued using Optuna with the HPC and SQLite, created a larger augmented dataset, looked at different input sizes, trained model with optimal hyperparameters.

Presenting results stage:

Week 7: Tidying up results, thinking of a narrative, producing and practicing for presentation, and writing the final report.

This UROP was an excellent and very effective way for me to learn about machine learning, improve my coding and develop a research mindset.

6.2 Jiho

Background research (Week 1):

I spent the first week reading about etiology, risk factors and biomarkers of EH which gave me a good background. I then looked at the overall flow of the feature extraction model and some possible features that can be used. I also explored available data (EH/non-EH images) and found out that the images are not cropped, and how EH lesions look different according to different skin types which can decrease the accuracy of the model.

Initialization (Week 2)

The second week was to set up the virtual environments, linking RDS to Jupyter Hub, and to test a classifier model using smaller dataset. The result showed 100% accuracy which gave me some complex emotions.

Fixing and improving the model (Week 3 and Week 4)

I had many up and down emotions during these weeks, suspicious and anxious as the output of the model was 100%, disappointment and relief when I found out that I was mixing up training and testing data which decreased the accuracy as fixed. At the end, I was able to make 95% accuracy by implementing Grid Search and *k*-fold cross validation.

Testing the model, building models of different feature combinations (Week 5)

The model was built using different feature combinations and was tested on different types of non-EH image types, and imbalanced input data.

Traditional features visualization and preparing for presentation (Week 6, Week7)

Traditional features were visualized to provide explainability to the model and started to tidy up the results and prepare for the presentation and final report.

I personally really enjoyed every part of the experience, getting to know a good friend, weekly meetings with Leo and Wai Hoh sometimes were joyful talking about some brilliant outputs that I think I got, but the other week being gloomy due to some mistakes made and seeing how my model gradually improves as various techniques were implemented. I was able to learn and apply various machine learning techniques to a real-life problem which made an invaluable experience.

7. Acknowledgements

We would like to thank Leo Huang and Dr Wai Hoh Tang for their constant support and help. The weekly meetings and advice given were invaluable and contributed greatly to the project and the learning.

We would also like to thank Professor Reiko Tanaka for organizing this program and giving us this valuable opportunity.

8. References

[1] Xiao A, Syed HA, Tsuchiya A. Eczema Herpeticum. [Updated 2024 Aug 12]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2024 Jan-. https://www.ncbi.nlm.nih.gov/books/NBK560781/

[2] Wheeler CE, Abele DC. Eczema herpeticum, primary and recurrent. Arch Dermatol. 1966 Feb;93(2):162-73. https://pubmed.ncbi.nlm.nih.gov/4159394/

[3] Hsu DY, Shinkai K, Silverberg JI. Epidemiology of Eczema Herpeticum in Hospitalized U.S. Children: Analysis of a Nationwide Cohort. J Invest Dermatol. 2018 Feb;138(2):265-272. https://pubmed.ncbi.nlm.nih.gov/28927889/

[4] Pan, K., Hurault, G., Arulkumaran, K., Williams, H.C., Tanaka, R.J. (2020). EczemaNet: Automating Detection and Severity Assessment of Atopic Dermatitis. In: Liu, M., Yan, P., Lian, C., Cao, X. (eds) Machine Learning in Medical Imaging. MLMI 2020. Lecture Notes in Computer Science(), vol 12436. Springer, Cham. https://doi.org/10.1007/978-3-030-59861-7_23

[5] Malik SG, Jamil SS, Aziz A, Ullah S, Ullah I, Abohashrh M. High-precision skin disease diagnosis through deep learning on dermoscopic images. Bioengineering. 2024;11(9):867. https://doi.org/10.3390/bioengineering11090867

[6] von Eschenbach, W.J. Transparency and the Black Box Problem: Why We Do Not Trust AI. *Philos. Technol.* **34**, 1607–1622 (2021). https://doi.org/10.1007/s13347-021-00477-0

[7] [WHT1] KS Thomas, K Koller, Taraneh Dean, CJ o'Leary, Tracey H Sach, A Frost, I Pallett, AM Crook, S Meredith, and AJ Nunn. "A multicentre randomised controlled trial and economic evaluation of ion-exchange water softeners for the treatment of eczema in children: the Softened Water Eczema Trial (SWET)". In: Health technology assessment 15.8 (2011)

[8] Google AI. SCIN: A new resource for representative dermatology images [Internet]. Google Research Blog. 2023 [cited 2024 Oct 17]. Available from: https://research.google/blog/scin-a-new-resource-for-representative-dermatology-images/

[9] Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett. 2006;27(8):861-74. doi:10.1016/j.patrec.2005.10.010.

[10] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In KDD

[11] Bradski G. The OpenCV Library. Dr Dobb&#x27;s Journal of Software Tools. 2000;

[12] Fast, OpenCV optimized 'for' pixel loops with, says PP. Local Binary Patterns with Python &OpenCV[Internet].PyImageSearch.2015.Availablefrom: https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/

[13] MATLAB. Texture analysis using the gray-level co-occurrence matrix (GLCM). MathWorks. [Internet]. [cited 2024 Oct 6]. Available from: https://uk.mathworks.com/help/images/texture-analysis-using-the-gray-level-co-occurrence-matrix-glcm.html

[14] Cloudinary. Edge detection: Glossary. Cloudinary. [Internet]. [cited 2024 Oct 6]. Available from: https://cloudinary.com/glossary/edge-detection

[15] CV Explained. Color histograms: Explained. CV Explained. [Internet]. 2020 Jul 21 [cited 2024 Oct 6]. Available from: https://cvexplained.wordpress.com/2020/07/21/10-3-color-histograms/

[16]] ScienceDirect. VGG16: Computer Science. ScienceDirect. [Internet]. [cited 2024 Oct 6]. Available from: https://www.sciencedirect.com/topics/computer-science/vgg-16

[17] Breiman L. Random forests. Mach Learn. 2001;45(1):5–32. doi:10.1023/A:1010933404324.

[18] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1995. p. 1137–43.

[19] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst. 2011;24:2546–54.

[20] Scikit-learn. Forest Importances: Examples. Scikit-learn. [Internet]. [cited 2024 Oct 6]. Available from: https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html

[21] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. International Journal of Computer Vision. 2020 Feb 1;128(2):336–59.

[22] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V. and Garcia-Rodriguez, J., 2017. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*. https://arxiv.org/abs/1704.06857

[23] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2020. Grad-CAM: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, *128*, pp.336-359. https://arxiv.org/abs/1610.02391 False Positives vs False Negatives https://www.datasource.ai/en/data-science-articles/false-positives-vs-false-negatives

[24] False Positives vs False Negatives https://www.datasource.ai/en/data-science-articles/false-positives-vs-false-negatives