UNIVERSITÉ CLERMONT AUVERGNE

Doctoral School of Sciences for Engineer

Laboratory of Informatics, Modelling and Optimization of the Systems - LIMOS - UMR 6168

Doctoral Thesis

Presented By

Sk Imran Hossain

To obtain the title of

Doctor of Philosophy in Computer Science

Contribution to Multimodal Classification Problem With Data Scarcity: Application to the Early Diagnosis of Lyme Disease

---

Presented and defended publicly on May 11, 2023, before the jury composed of:

Vincent Barra	Université Clermont Auvergne	President
Isabelle Bichindaritz	State University of New York at Oswego	Reviewer
Anna Fabijańska	Lodz University of Technology	Reviewer
Germain Forestier	Université de Haute-Alsace	Reviewer
Richard Emilion	Université d'Orléans	Examiner
Olivier Lesens	CHU de Clermont-Ferrand	Examiner
Jocelyn de Goër de Herve	INRAE	Co-supervisor
Engelbert Mephu Nguifo	Université Clermont Auvergne	Thesis director

Acknowledgments

I would like to express my heartfelt gratitude to all those who have helped me throughout my PhD journey. First and foremost, I am deeply grateful to my supervisors Pr. Engelbert Mephu Nguifo and Dr. Jocelyn de Goër de Herve for their unwavering guidance, encouragement, and expertise throughout this research. Their constructive feedback and continuous motivation have been instrumental in shaping my ideas and refining my work. I am also indebted to Pr. Richard Emilion for his invaluable suggestions and help.

I would also like to thank my collaborators from the DAPPEM project, especially the members from INRAE and CHU Clermont-Ferrand, for their stimulating discussions, insightful feedback, and willingness to share their knowledge and expertise. Their contributions have enriched my work and broadened my perspective on the field.

My heartfelt thanks go to the members of the Miners research team for constructive discussions and also for helping me out with non-academic activities. Their unconditional support made it possible for me to survive in a foreign country with a language barrier.

I would also like to extend my sincere gratitude to the members of the jury for agreeing to assess my work and providing constructive feedback.

I would like to express my heartfelt gratitude to my family and friends, for their mental support, unconditional love, and encouragement throughout this journey. Their belief in me and my abilities has been a constant source of inspiration and motivation, and I am grateful for their presence in my life.

This research was funded by the European Regional Development Fund, project DAPPEM–AV0021029. The DAPPEM project («Développement d'une APPlication d'identification des Erythèmes Migrants à partir de photographies»), was coordinated by Pr. Olivier Lesens and was carried out under the Call for Proposal 'Pack Ambition Research' from the Auvergne-Rhône-Alpes region, France. This work was also partially funded by Mutualité Sociale Agricole (MSA), France. I am immensely grateful for the financial assistance, which enabled me to carry out my research with the necessary resources and facilities.

I am deeply appreciative of the contributions of all those who have helped me along the way, and I am proud to have had the opportunity to work with such talented and dedicated individuals.

Finally, I am grateful to the almighty for showering his blessing on me.

Abstract

Recent advancement in deep learning techniques has eased the creation of artificial intelligence (AI) solutions to aid in skin disorder diagnosis. Studies showed that incorporating data from multiple modalities in the analysis process significantly improves the AI based model's performance compared to a single modality based analysis for many medical diagnosis tasks. Although deep learning based systems compete on par with expert dermatologists for diagnosing skin cancer, their application is limited in diseases like Lyme disease, where it is difficult to collect training data.

In this thesis, we particularly focused on AI assisted Lyme disease analysis because it requires both patient data and lesion images for a proper diagnosis, but there is no available dataset comprising both of these modalities. Self-supervised pre-training is effective to address the data scarcity problem of lesion images when a large number of in-domain images are available. But for some diseases, it's difficult to collect a lot of in-domain images. To tackle this problem we proposed a customized transfer learning approach to improve ImageNet pre-trained convolutional neural network's performance by utilizing additional pre-training with an out-of-domain dataset. To deal with the lack of training data for patient data modality, we proposed an expert opinion elicitation approach to create a model that calculates disease probability from patient data with intuitive model validation based on decision tree and formal concept analysis. The proposed questionnaire based elicitation approach is less demanding for the experts. We also proposed an approach for combining disease probability scores from multiple modalities by ensuring veto power for a modality, based on expert choice.

As part of the thesis, we prepared a dataset of Lyme disease related skin lesion images with labeling from expert dermatologists. We also created another skin lesion hair mask annotation dataset for dealing with lesion hair artifacts in an efficient manner. The proposed techniques in this thesis were applied to create a mobile application for assisting with early Lyme disease diagnosis but they will be useful for other similar diseases where there is a problem of data scarcity.

Keywords: Artificial intelligence; Data scarcity; Deep learning; Multimodality; Lyme disease; Erythema migrans

Résumé

Les récents progrès des techniques d'apprentissage profond ont permis la mise au point de modèles d'intelligence artificielle (IA) pour aider au diagnostic des maladies de la peau. Dans la littérature il est montré que l'intégration de données provenant de plusieurs sources dans le processus d'analyse de données peut améliorer considérablement les performances du modèle d'IA par rapport à une analyse basée sur une source unique, notamment dans le cas du diagnostic médical. Bien que les systèmes basés sur l'apprentissage profond rivalisent avec les experts dermatologues pour le diagnostic du cancer de la peau, leur utilisation reste limitée au niveau de maladies telles que la maladie de Lyme, où les données d'entraînement sont rares.

Dans cette thèse, nous nous sommes focalisés sur le développement d'un modèle d'IA appliqué à la maladie de Lyme avec la particularité que cette maladie nécessite à la fois des données contextuelles de patients et des images de lésions cutanées pour pouvoir établir un diagnostic correct. En outre, il n'existe aucun jeu de données comprenant ces deux modalités. Le pré-apprentissage auto-supervisé est efficace pour résoudre le problème de la rareté des données lorsqu'un grand nombre de données du domaine sont disponibles par ailleurs. Cependant, pour certaines maladies comme la maladie de Lyme, il est difficile de collecter un grand nombre d'images du domaine. Pour faire face à ce problème, nous avons proposé une approche personnalisée d'apprentissage par transfert afin d'améliorer les performances du réseau de neurones convolutifs pré-entraîné ImageNet, en mettant en place une phase de pré-entraînement supplémentaire avec un ensemble de données hors-domaine. En outre, pour faire face au manque de données d'entraînement concernant les données contextuelles des patients, nous avons proposé une approche d'élicitation d'opinion d'experts (médecins) pour créer un modèle qui calcule la probabilité de la maladie à partir des données relatives à un patient avec une validation intuitive du modèle basée sur un arbre de décision et une analyse formelle des concepts. L'approche d'élicitation proposée, basée sur un questionnaire, est moins exigeante pour les experts. Nous avons également proposé une approche pour combiner les scores de probabilité de la maladie provenant de plusieurs modalités en assurant un droit de veto pour une modalité, en fonction du choix d'un expert.

Dans le cadre de cette thèse, nous avons constitué un jeu de données d'images de lésions cutanées, liées à la maladie de Lyme avec une classification réalisée par un panel de dermatologues experts. Nous avons également créé un autre jeu de données d'annotation de masque de poils de lésions cutanées permettant de traiter les artefacts liés aux poils sur les lésions, de manière efficace. Les techniques proposées dans cette thèse ont été utilisées pour créer une application mobile d'aide au diagnostic précoce de la maladie de Lyme, mais elles pourraient être utiles à d'autres maladies similaires pour lesquelles il existe un problème de pénurie de données.

Mots clés : Intelligence artificielle ; Rareté des données ; Apprentissage profond ; Multimodalité ; Maladie de Lyme ; Erythème migrant

Contents

1 Introduction

This chapter sets the thesis motivation, summarizes our research problem and main contributions, and also contains the thesis organization.

Chapter Contents

1.1 Context

Diagnosing skin disorders requires a careful inspection from dermatologists or infectiologists but their availability, especially in rural areas is scarce [39]. As a result, the diagnosis is generally carried out by non-specialists, and their diagnostic accuracy is in the range of twenty-four to seventy percent [143, 163]. The wrong diagnosis can result in improper or delayed treatment which can be harmful to the patient. Recent advancements in Artificial intelligence (AI) especially deep learning techniques have found applications in many medical domains including medical image analysis tasks [9, 30, 76, 91]. It has eased the creation of AI solutions to aid in skin disorder diagnosis. AI powered diagnostic tools can help with the scarcity of expert dermatologists.

Many works have been done utilizing deep learning techniques specifically convolutional neural networks (CNNs) for diagnosing cancerous and other common skin lesions from dermoscopic images. Dermoscopic images have unique lighting and low level of noise because they are captured using a dermatoscope device having a lighting system and a high-quality magnifying lens [154]. Dermoscopic images require dermatoscopes from dermatology clinics so other works focused on diagnosing skin diseases using deep learning from clinical skin lesion images acquired mostly using mobile phones and digital cameras [154].

Several studies have shown that deep learning-based systems' disease diagnosis capability from clinical and dermoscopic images is on par with experienced dermatologists [13,

26, 34, 49, 101, 165]. Considering patient data with skin lesion images can boost the performance of AI model for skin disease diagnosis and for some diseases like Lyme disease, it is crucial to consider both modalities for a proper diagnosis. But the scarcity of training data is a big challenge for creating robust AI models. In this thesis, we tried to tackle the data scarcity issue of multimodal skin disorder diagnosis by addressing the issues of a limited number of clinical skin lesion images, unavailability of patient data, and hair artifacts on dermoscopic lesion images.

1.2 Research Problems

For image classification problems with limited labeled data, pre-training the model with a large number of unlabeled domain specific images can significantly improve the model performance [6, 31, 52, 93, 148, 188]. Often practitioners work with clinical skin lesion images and it is difficult to gather a large number of unlabeled images from the same domain for rare diseases. Many datasets of dermoscopic images are easily accessible however, their image modality is significantly different from clinical skin lesion images. Our first research concern is about the utilization of dermoscopic images for improving the performance of Clinical image classification.

Without taking into account the additional context from patient data, a correct diagnosis solely on skin lesions is ineffective for some conditions, such as Lyme disease. However, it is time-consuming and costly to collect training data for patient data modality let alone creating a dataset comprising multiple modalities. So, our second research concern is how to utilize patient data to assist deep learning based skin lesion image classifier in the absence of training data.

The effectiveness of computer-assisted lesion analysis algorithms is affected by the occlusion of skin lesions in dermoscopic images caused by hair artifacts. Our third research concern is about efficiently handling hair artifacts in dermoscopic images.

In this section, we have briefly stated our research concerns. These issues are described in detail in Section 2.3 in context of related works.

1.3 Contributions

In this thesis, we have made the following contributions addressing the stated research problems:

- A strategy to improve transfer learning based clinical skin lesion image classifier's performance with additional pre-training using dermoscopic images.
- A flexible questionnaire based expert opinion elicitation method to assist skin lesion image classifier with patient data in the absence of training data.

- An approach for combining independent disease probability scores from multiple modalities by ensuring veto power for a modality based on expert choice.
- A dataset of Lyme disease related skin lesion images with labeling from expert dermatologists.
- A fine-grained skin lesion hair mask annotation dataset for dealing with lesion hair artifacts in an efficient manner.

1.4 Thesis Organization

The thesis is organized into six chapters:

- Chapter 1 (introduction) sets the thesis motivation, summarizes the research problems and our main contributions, and also contains the thesis organization.
- Chapter 2 provides the required theoretical background and literature review, and states the research questions in context of related studies.
- Chapter 3 presents our pre-training strategy for improving clinical skin lesion image classification performance of ImageNet pre-trained convolutional neural networks by utilizing additional pre-training with dermoscopic images.
- Chapter 4 presents the questionnaire based expert opinion elicitation method for calculating disease probability from patient data and an approach for combining independent probability estimates from multiple modalities.
- Chapter 5 presents our ongoing works on efficiently dealing with dermoscopic skin lesion hair artifact, custom architecture for Lyme disease image classifier, and an application utilizing our research findings.
- Chapter 6 (conclusions) presents a summary of the key findings of this thesis and possible future research directions.

2 Background

This chapter provides the required theoretical background, and literature review and states the research questions in context.

Chapter Contents

2.1 Theoretical Background

The required theoretical concepts to understand the rest of the manuscript are briefly described in the following subsections.

2.1.1 Convolutional Neural Network

Convolutional neural network (CNN) is a kind of neural network that simulates some actions generated in the human visual cortex using convolution mathematical operation to extract features from input and pass these features through successive layers to generate more abstract features to yield a final output [83]. It processes data having grid patterns (images for example) and learns spatial feature hierarchies adaptively and automatically, from basic to complex patterns [178].

Feedforward neural networks, also called deep feedforward networks, or multilayer perceptrons (MLPs), are classic examples of neural networks [43]. A mathematical representation of a biological neuron is called a perceptron [104]. Figure 2.1 the representation of a biological neuron. While the axons of other neurons provide electrical signals to the dendrite in actual neurons, these electrical signals are represented as numerical values in perceptron. Electrical impulses are regulated in various amounts at the synapses between the dendrite and axons. The perceptron models this by multiplying each input value by a value referred to as the weight. Soma also called the cell body is responsible for input processing and decision making in a biological neuron. Only when the sum of the input signals is greater than a predetermined threshold does a real neuron really fire an output signal. By computing the weighted sum of the inputs, which represents the entire strength of the input signals, and applying an activation function to the sum to determine the output, we may mimic this phenomenon in a perceptron. Nonlinear activation functions can be used for performing nonlinear transformations with perceptrons. Appendix Table A.1 lists some of the commonly used activation functions. Perceptron calculates the dot product between a learnable weight vector and an input vector and passes it through an activation function after adding a scalar bias term. Mathematically, the operation of perceptron can be represented as Equation 2.1.

$$y_{out} = f_{act}(\sum_{i=1}^{n} w_i x_i + b) \tag{2.1}$$

where y_{out} is the output of the perceptron, f_{act} is the activation function. w_i is the weight corresponding to input x_i, and b is the bias. Figure 2.2a shows the schematic representation of a perceptron. MLP consists of many layers of perceptrons where each perceptron of a layer is fully connected to every other perceptron in the previous layer as shown in Figure 2.2b. Neural networks are trained using optimization algorithms. An optimization algorithm updates learnable parameters (weights, biases) of the network to minimize a task-specific loss function. Gradient descent [133] and its extensions like RMSprop [161] and Adam [80] are some of the popular choices for optimization. We have used Adam optimizer in this study and Appendix Section A.2 contains a brief overview of Adam optimizer. Interested readers are suggested to consult the study by Ruder [134] for a detailed overview of gradient descent based optimization algorithms. Traditional MLPs are not

well suited for image processing as they require a large number of parameters and can not take into consideration the spatial information in images. Although, modern MLPs like ResMLP [162] have been customized for image classification tasks CNNs are preferred over them.

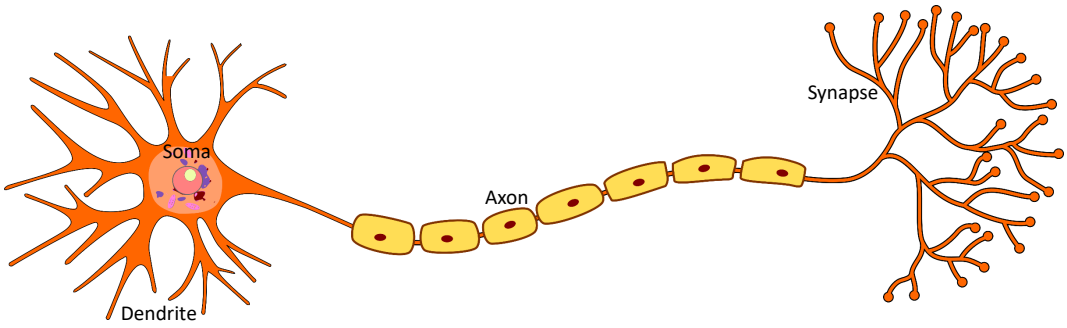


Figure 2.1: Illustration of a biological neuron. Image modified from [151].

The idea of convolution, a mathematical technique that entails swiping a tiny kernel over an image and computing the dot product between the kernel and the corresponding pixels in the image, serves as the foundation for CNNs. By spotting patterns in the data, this procedure aids in the extraction of features from the image. A feature map, which highlights particular aspects of the image such as edges, corners, and textures, is the result of this operation. The convolution operation is shown in Equation 2.3 [43].

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.2}$$

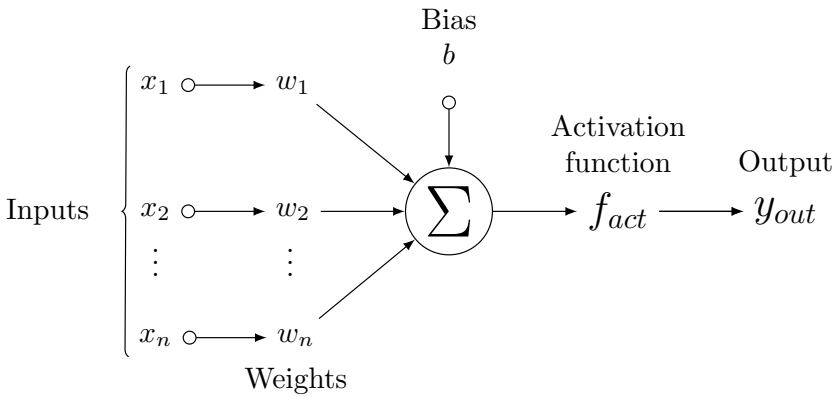
where S is the output feature map, I is the input image, K is the kernel of size $m \times n$, $*$ represents the convolution operation, i and j are the row and column indexes of an element from S. The convolution is commutative, so the equation can be also written as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{2.3}$$

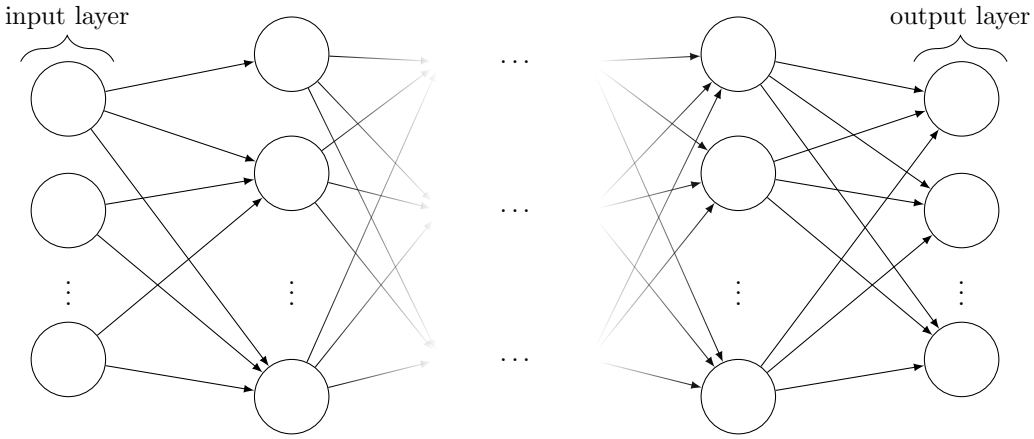
Many deep learning libraries uses cross-correlation function in place of convolution as shown in Equation 2.4.

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{2.4}$$

Cross-correlation is not commutative but has similar properties as convolution. A convolution operation on a two dimensional input image is illustrated in Figure 2.3a. Stride and padding are frequently used with convolution operation. Stride specifies how many



(a) Perceptron.



(b) Multilayer perceptron (MLP).

Figure 2.2: Schematic representation of perceptron and multilayer perceptron (MLP). Images modified from [128, 152].

pixels the kernel moves after each convolution operation i.e. the sliding of the kernel between the production of each output element. Padding is the process of adding empty pixels around the border of an input image. Padding is used to maintain image size and also for finding patterns in borders with full convolution on edge pixels. To decrease the dimensionality of the feature maps and boost computing efficiency, CNNs use pooling layers in addition to convolutional layers. Downsampling is normally carried out by pooling layers using the maximum or average value of a set of adjacent pixels in the feature map as illustrated in Figure 2.3b. Pooling also helps for making the model invariant to subtle translational changes in input. Convolutional, pooling, and fully connected layers are some of the layers of interconnected nodes that make up a CNN. The final classification or regression operation is carried out by the fully connected layers. Figure 2.3c shows a schematic representation of a CNN which stacks convolutional layers with

an activation function applied after the convolution operation, pooling layers, and fully connected layers with activation functions. Although, vision transformers [55] are gaining a lot of popularity for various vision related tasks including medical imaging, modern CNN architectures like ConvNext [97] and EfficientNetV2 [160] compete on par with them. Section 3.2.3 contains brief descriptions of the CNN architectures used in this thesis.

2.1.2 Decision Tree

Decision tree is a supervised learning algorithm, which can be utilized for both regression and classification tasks [12, 124]. In this thesis, we are focusing on the decision tree for classification task. Decision tree represents a classifier as a recursive partition of instance space using a set of splitting rules [12, 124]. These rules are easy to visualize and interpret with tree diagrams. Decision tree is a directed tree with no incoming edges at the root node and each of the other nodes has just one incoming edge. A decision or leaf or terminal node is a node without outgoing edges. All other nodes are called test or internal nodes. The instance space is divided into two or more sub-spaces by each test node based on a discrete function of input attribute values. Each decision node is given a class that corresponds to the best suitable target value. Instances are classified according to the test results by navigating from the tree's root to a leaf.

Figure 2.4a shows an example training dataset and Figure 2.4b shows the corresponding decision tree for deciding about playing golf (Yes/No) based on predictors like Outlook (Sunny/Overcast/Rainy), Temperature (Hot/Cool/Mild), Humidity (High/Normal), and Wind (Weak/Strong) [138]. The red, yellow, and green boxes represent root, internal, and decision nodes respectively.

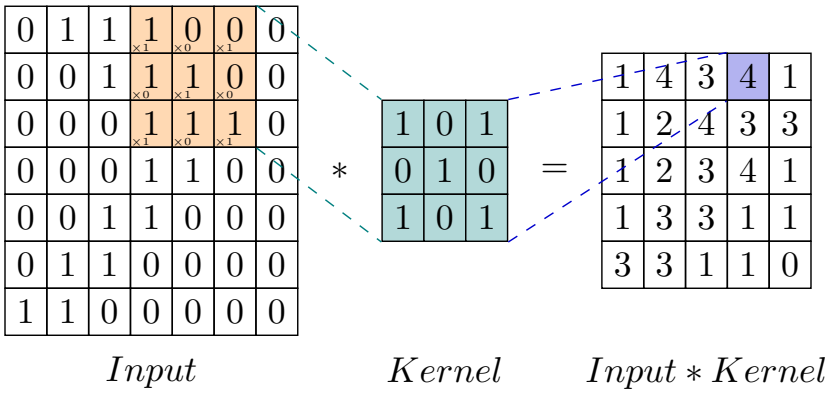
2.1.3 Formal Concept Analysis and Concept Lattice

Formal concept analysis (FCA) is a method of generating a formal concept hierarchy from a set of objects and their properties [174]. FCA has found many applications in machine learning and bioinformatics [107, 108, 109]. In FCA each concept represents objects that share a particular set of attributes. FCA computes concept lattice, a directed, acyclic graph by hierarchically ordering all formal concepts derived from tabular input data.

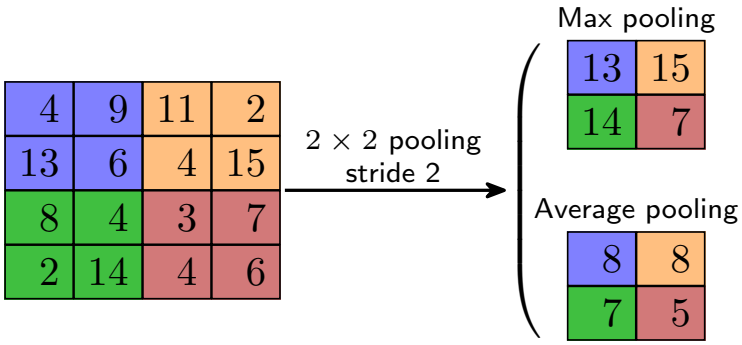
The notion of formal context is central to FCA. Formal context is a triple $\langle O, Y, I \rangle$ where O is a set of objects, Y is a set of attributes, and incidence $I \subseteq O \times Y$ is a binary relation. A pair $\langle A, B \rangle$ is a formal concept of $\langle O, Y, I \rangle$ provided that $A \subseteq O$, $B \subseteq Y$, $A^{\uparrow} = B$, and $B^{\downarrow} = A$ where,

$$A^{\uparrow} = \{y \in Y | for\ each\ o \in A : \langle o, y \rangle \in I\} \tag{2.5}$$

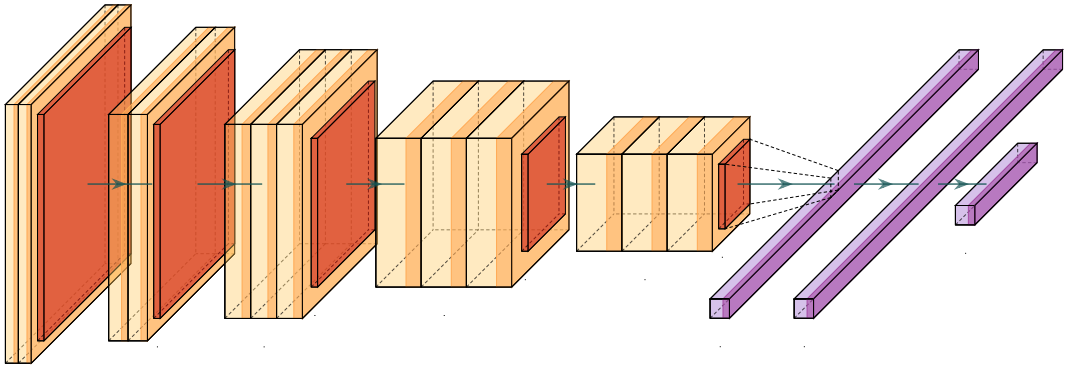
$$B^{\downarrow} = \{o \in O | for\ each\ y \in B : \langle o, y \rangle \in I\} \tag{2.6}$$



(a) Illustration of a convolution operation [168].



(b) Illustration of pooling operation.



(c) Schematic representation of a convolutional neural network. The yellow, and violet boxes with shaded endings represent convolutional and fully connected layers respectively with activation function. The red box represents pooling layer.

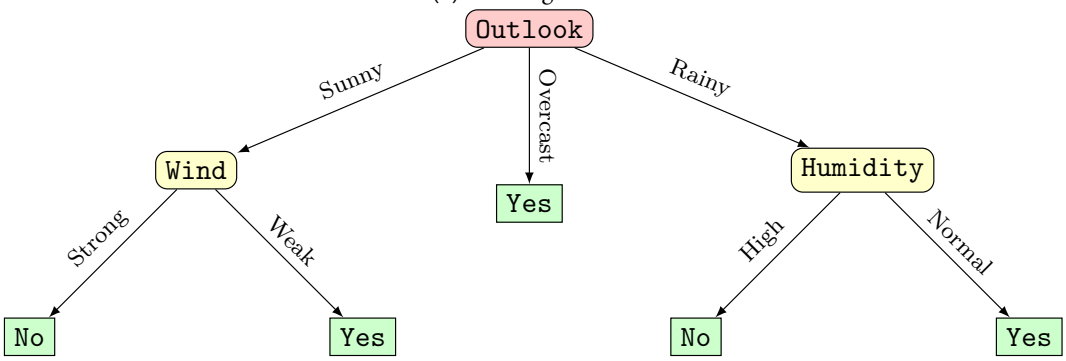
Figure 2.3: Schematic representation of convolution operation, pooling operation and convolutional neural network.

A is called the extent and B is called the intent of a concept $\langle A, B\rangle$. Formal concepts are ordered naturally by subconcept-superconcept relation defined as follows:

$$\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle \iff A_1 \subseteq A_2 (\iff B_2 \subseteq B_1) \tag{2.7}$$

Predictors				Target
Outlook	**Temperature**	**Humidity**	**Wind**	**Play Golf**
Rainy	Hot	High	Weak	No
Rainy	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Sunny	Mild	High	Weak	Yes
Sunny	Cool	Normal	Weak	Yes
Sunny	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Rainy	Mild	High	Weak	No
Rainy	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Weak	Yes
Rainy	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Sunny	Mild	High	Strong	No

(a) Training dataset.



(b) Decision tree.

Figure 2.4: An example of a decision tree for deciding about playing golf (Yes/No) based on predictors like Outlook (Sunny/Overcast/Rainy), Temperature (Hot/Cool/Mild), Humidity (High/Normal), and Wind (Weak/Strong) [138]. The red, yellow, and green boxes represent root, internal, and decision nodes respectively.

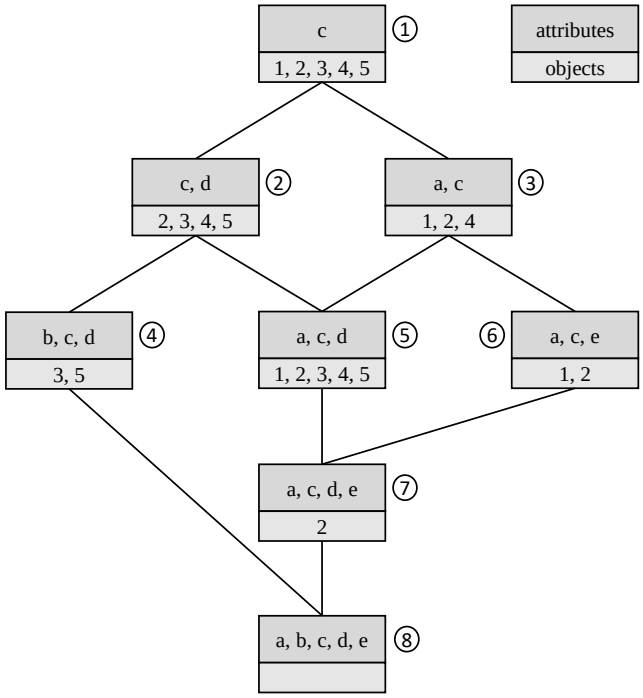
For a formal context $\langle O, Y, I \rangle$ the set $\mathfrak{B}(O, Y, I)$ of all formal concepts with the ordering shown in Equation (2.7) is called the concept lattice.

Figure 2.5a shows a sample formal context in a tabular form called a cross-table. The table rows represent objects and the columns represent attributes. An entry $\times$ in the table represents that the corresponding object has the corresponding attribute. Figure 2.5b shows the concept lattice built from the formal context. Each node represents a formal concept by listing objects that share a set of attributes. The number within a circle beside the node is added to identify a node and for explanation purposes only. Within each node, the bottom box lists the objects i.e. the extent and the top box lists the attributes shared by the objects i.e. the intent of the concept. For example, the intent and extent of node (2) are $\{c, d\}$ and $\{2, 3, 4, 5\}$ respectively. The lines in the lattice represent subconcept-

superconcept relationship. For example, the formal concept represented by node (4) is a subconcept of the formal concept represented by node (2) because the extent of node (4) is a subset of the extent of node (2) and the intent of node (4) is a superset of the intent of node (2).

		Attributes				
		a	b	c	d	e
Objects	1	×		×		×
	2	×		×	×	×
	3		×	×	×	
	4	×		×	×	
	5		×	×	×	

(a) Formal context.



(b) Concept lattice.

Figure 2.5: Example of a formal context in tabular form and corresponding concept lattice. An entry × in the context table represents that the corresponding object has the corresponding attribute.

2.1.4 Gaussian Mixture Model

A Gaussian mixture model (GMM) is a probability density function represented as the weighted sum of component Gaussian densities [130]. The mixture represents a normally distributed overall population whereas the components represent subpopulations within the whole population. For one-dimensional data, a GMM with M components can be defined as:

$$\hat{f}_{GMM}(x) = \sum_{m=1}^{M} \varnothing_m \mathcal{N}(x|\mu_m, \sigma_m) \tag{2.8}$$

where, $\varnothing_m \geq 0$ is the mixture weight i.e. the probability of m-th component κ_m satisfying $\sum_{m=1}^{M} \varnothing_m = 1$ so that the total probability distribution normalizes to 1, and $\mathcal{N}(x|\mu_m, \sigma_m)$ is the distribution of a Gaussian component with mean μ_m and standard deviation σ_m defined as:

$$\mathcal{N}(x|\mu_m, \sigma_m) = \frac{1}{\sigma_m\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_m}{\sigma_m}\right)^2} \tag{2.9}$$

Expectation-Maximization, an iterative unsupervised learning technique can be used to determine the parameters of GMM [32]. Steps involved in Expectation-Maximization for n data points $X = \{x_t|t = 1, \ldots, n\}$ are:

- Guess initial values for GMM parameters denoted by $\hat{\mu}_m, \hat{\sigma}_m$, and $\hat{\varnothing}_m$ respectively.
- Expectation step: calculate $\hat{\gamma}_{t,m}$, the probability of a point x_t being generated by κ_m

$$\hat{\gamma}_{t,m} = \frac{\hat{\varnothing}_m \mathcal{N}(x_t|\hat{\mu}_m, \hat{\sigma}_m)}{\sum_{r=1}^{M} \hat{\varnothing}_r \mathcal{N}(x_t|\hat{\mu}_r, \hat{\sigma}_r)} \tag{2.10}$$

- Maximization step: Update GMM parameters using the following equations:

$$\hat{\mu}_m = \frac{\sum_{t=1}^{n} \hat{\gamma}_{t,m} x_t}{\sum_{t=1}^{n} \hat{\gamma}_{t,m}} \tag{2.11}$$

$$\hat{\sigma}_m = \sqrt{\frac{\sum_{t=1}^{n} \hat{\gamma}_{t,m}(x_t - \hat{\mu}_m)^2}{\sum_{t=1}^{n} \hat{\gamma}_{t,m}}} \tag{2.12}$$

$$\hat{\varnothing}_m = \sum_{t=1}^{n} \frac{\hat{\gamma}_{t,m}}{n} \tag{2.13}$$

- Repeat Expectation and Maximization steps until the total likelihood L converges, where

$$L = \prod_{t=1}^{n} \hat{f}_{GMM}(x_t) \tag{2.14}$$

Information criterion tests like Akaike Information Criteria (AIC) [2] and Bayesian Information Criteria (BIC) [140] can be used to select an appropriate GMM by penalizing the number of free parameters to prevent overfitting. AIC and BIC can be defined as:

$$AIC = 2p + 2\ln L \tag{2.15}$$

$$BIC = p \ln L + 2 \ln L \tag{2.16}$$

where p is the number of free parameters and ln is the natural logarithm. The preferred GMM is the one with minimum AIC and BIC values.

2.1.5 Kernel Density Estimation

Kernel density estimation (KDE) is a non-parametric way of estimating the probability density function of an independent and identically distributed random variable [119, 132]. For n data points $X = \{x_t | t = 1, \ldots, n\}$, KDE is calculated as:

$$\hat{f}_{KDE}(x) = \frac{1}{nh} \sum_{t=1}^{n} K\left(\frac{x - x_t}{h}\right) \tag{2.17}$$

where h is the bandwidth and K is the kernel function. If a Gaussian kernel function is used to estimate the density of univariate data then the bandwidth can be selected using Silverman's rule of thumb [146] as shown in the following equation:

$$h = 0.9min\left(\hat{\sigma}, \frac{IQR}{1.34}\right) n^{\frac{-1}{5}} \tag{2.18}$$

where IQR is the interquartile range and $\hat{\sigma}$ is the standard deviation of the samples.

2.1.6 Transfer Learning and Pre-training strategies

Transfer learning is a collection of techniques to enhance the performance of a model on a target task using the information that a model acquires during training on a source task, even if the two tasks are dissimilar [117]. Transfer learning focuses on knowledge adaptation and it is defined using two concepts: domains and tasks. A domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ has two components: A feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$ where $X = \{x_1, \cdots, x_n\} \in \mathcal{X}$. For a domain $\mathcal{D}$ a task $\mathcal{T} = \{\mathcal{Y}, f(x)\}$ has two parts: A label space $\mathcal{Y}$ and a predictive function $f : \mathcal{X} \rightarrow \mathcal{Y}$, that can be learned from training data of pairs $\{x_i, y_i\}$ where $x_i \in X, y_i \in \mathcal{Y}$. Pan and Yang [117] defined transfer learning as:

Definition 1 (Transfer Learning)**.** *"Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$" [117].*

Plested and Gedeon [122] defined deep transfer learning in the context of image classification as:

Definition 2 (Deep Transfer Learning). *"Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the performance of the target model M on the target task $\mathcal{T}_T$ by initializing it with weights W that are trained on source task $\mathcal{T}_S$ using source dataset D_S (pretraining), where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$" [122].*

Transfer learning can be broadly categorized into three settings based on $\mathcal{T}_S$, $\mathcal{T}_T$ and provided labels as shown in Figure 2.6 [100, 134]. If $\mathcal{T}_S = \mathcal{T}_T$ with only source domain labels provided, it is called transductive transfer learning. When $\mathcal{T}_S \neq \mathcal{T}_T$ with labels available for the target domain, it is called inductive transfer learning. If no labels are provided then it is called unsupervised transfer learning. Inductive transfer learning can be subdivided into multi-task and sequential transfer learning. $\mathcal{T}_S$ and $\mathcal{T}_T$ are simultaneously learned in multi-task learning whereas, in sequential transfer learning $\mathcal{T}_S$ is first learned (pre-training stage), and then $\mathcal{T}_T$ is learned (fine-tuning stage). In this thesis, we are focusing on sequential transfer learning.

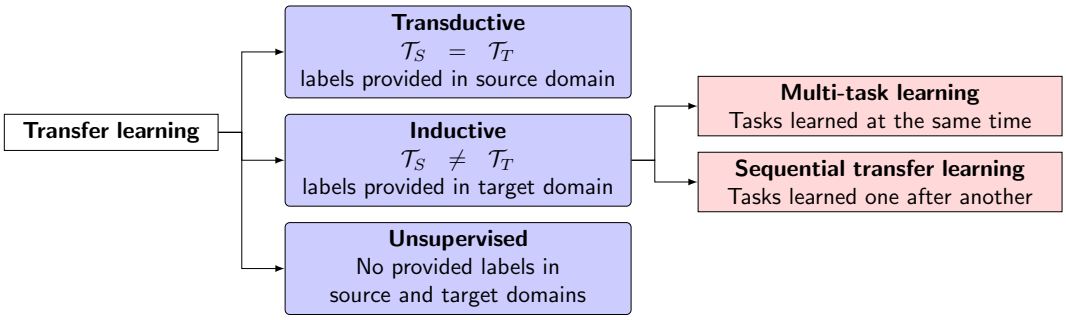


Figure 2.6: Transfer learning scenarios [100, 134]. $\mathcal{T}_S$ and $\mathcal{T}_T$ represent source and target tasks respectively.

Pre-training in the context of transfer learning can be supervised or self-supervised (described in next para) [100]. Supervised transfer learning uses a pre-trained model that was trained on a sizable dataset for a particular task as a starting point for a new task. A smaller dataset is used to fine-tune the pre-trained model on the new task in order to adapt it to the new task. For instance, a model that has already been trained on a huge image classification dataset like ImageNet [135] can be fine-tuned for a particular image classification job, like detecting cats vs dogs, on a smaller dataset. This strategy can save resources and shorten the training period and also increase the new model's accuracy.

Self-supervised pre-training uses a model that has already been trained on a related task that does not require labeled data as a starting point for a new task. In self-supervised pre-training for image classification, the model is first taught to extract important features from images using unsupervised techniques like pretext tasks, generative modeling, or contrastive learning [94]. A pretext task can be predicting a specific feature of the data, like predicting an image's rotation or hue. Generative modeling involves training the model to

create realistic synthetic data. Contrastive learning involves training the model to differentiate between negative and positive image pairs. Using this unsupervised pre-training technique the model learns a useful representation of the data which can be refined for a new downstream task on a labeled dataset. When labeled data is hard to come by or expensive, this strategy can be especially helpful. We can use a source task with a lot of unlabeled examples and transfer the learned knowledge to an interesting target task thanks to the complementing research fields of self-supervised learning and transfer learning [100].

2.1.7 Visual Explanation of CNN Model

Explainability is important for AI tools especially in the case of medical applications [169] to understand how a model is taking its decision. Explainability techniques can be model-based (the model itself is explainable i.e. easy to be understood) vs post hoc (explains a trained model), model-specific (limited to particular types of models) vs model-agnostic (independent of the type of the model), and global (provides general relationships learned by the model in a dataset level) vs local (provides explanations for individual input) [167]. Visual explanation that shows regions of the input image that are significant for predictions from the model is common used for medical image analysis [167]. Grad-CAM [141] is a post hoc, model-specific technique for local visual explanation of CNNs. Grad-CAM uses gradient flowing into the ultimate convolution layer for producing heatmaps, and it is a kind of post-hoc attention that can be applied on an already trained model. Grad-CAM provides similar result to occlusion sensitivity map [185] that works by masking patches of the input image, but Grad-CAM is much faster to calculate compared to image occlusion [141]. Grad-CAM has been used for visual explanation in numerous medical image analysis tasks including brain [79], breast [36], cardiovascular [19], chest [14], dental [170], eye [103], female reproductive system [47], gastrointestinal [73], lymph nodes [75], musculosketal [139], thyroid [84], and skin [192] images. We utilized Grad-CAM technique for visualizing the regions of the input image that are significant for skin lesion class prediction from the CNN models (as shown in Figure 2.7) used in our study.

2.2 Literature Review

The following subsections review the related works on AI for skin lesion diagnosis, a brief overview of Lyme disease with the use of AI for Lyme disease diagnosis, and also the works on the data scarcity problem.

2.2.1 AI for Skin Disorder Diagnosis

Many works have been done utilizing deep learning techniques specifically convolutional neural networks (CNNs) for diagnosing cancerous and other common skin lesions from

(a) Input image.

(b) Gard-CAM visualization.

Figure 2.7: Gard-CAM visualization example.

dermoscopic images. Haenssle et al. [48] used transfer learning from ImageNet [135] pre-trained InceptionV4 CNN architecture for detecting melanoma skin cancer using images from International Skin Imaging Collaboration (ISIC) [26] dermoscopic image archive and compared the model's performance against 58 dermatologists. CNN outperformed most of the expert dermatologists. Brinker et al. [13] also used ImageNet transfer learning with ResNet50 CNN architecture on a dataset of 12,378 dermoscopic images from ISIC archive for the melanoma classification task and compared the CNN's performance against dermatologists. Again deep learning outperformed 136 out of 157 dermatologists. Maron et al. [101]] also used ResNet50 architecture with ImageNet transfer learning and performed a multiclass cancer classification using 11,444 dermoscopic images from ISIC archive where most of the images were taken from HAM10000 dataset [165]. This study also showed the superiority of deep learning models compared to 112 dermatologists.

Liu et al. [95] trained an InceptionV4 based deep learning system using clinical skin lesion images and patient data from 16,114 verified cases for the differential diagnosis of 26 common skin conditions. Esteva et al. [34] trained an InceptionV3 CNN architecture on a dataset of 126,076 clinical skin lesion images and 3,374 dermoscopic images using ImageNet transfer learning for skin cancer classification. The deep learning model performed on par compared with 21 board-certified expert dermatologists. Han et al. [49] trained an ensemble model of ResNet152 and VGG19 CNN architectures using a dataset of 49,567 clinical skin lesion images of onychomycosis that outperformed most of the expert dermatologists.

Results from aforementioned studies confirmed that deep learning-based systems compete on par with expert dermatologists for diagnosing diseases from dermoscopic and clinical skin lesion images. Recent studies showed that incorporating data from multiple modalities in the analysis process significantly improves the artificial intelligence based

models' performance compared to a single modality based analysis for many medical diagnosis tasks [22, 86, 116, 142]. Pacheco et al. [116] proposed an attention based deep learning approach for combining images and patient data for skin cancer classification that resulted in better performance compared to a single modality based analysis. Chen et al. [22] showed a 9 percent improvement in model accuracy with a multimodal fusion of skin image and clinical data compared to image only skin cancer classification.

2.2.2 AI for Lyme Disease Diagnosis

Lyme disease is an infectious disease transmitted by ticks and caused by pathogenic bacteria of the *Borrelia burgdorferi* sensu lato group [144]. It is estimated that around 476,000 people in the United States and more than 200,000 people in western Europe are affected by Lyme disease each year [102]. Most of the time an expanding round or oval red skin lesion known as erythema migrans (EM) becomes visible in the victim's body which is the most common early symptom of Lyme disease [15, 144]. EM usually appears at the site of a tick bite after one to two weeks (range, 3 to 30 days) as a small redness and expands almost a centimeter per day, creating the characteristic bull's-eye pattern as shown in Figure 2.8a [10, 15, 144, 150]. EM generally vanishes within a few weeks or months but the Lyme disease infection advances to affect the nervous system, skin, joints, eyes, and heart [144, 150]. Antibiotics can be used as a medium of effective treatment in the early stage of Lyme disease. So, early recognition of EM is extremely important to avoid long-term complications of Lyme disease.

Most European and North American guidelines recommend a two-tier serology test to detect antibodies against *Borrelia burgdorferi* sensu lato for diagnosing Lyme disease [37, 164]. However, a serology test is only recommended in the absence of EM because early serology has low sensitivity (40% to 60%) and may result in false negatives [37]. Alternatively, direct detection of Borrelia burgdorferi sensu lato can be done using culture, microscopy, or PCR [164]. The gold standard of microbiological diagnosis - the culture of bacteria requires laboratory expertise and special media for *Borrelia burgdorferi* sensu lato [37]. Light microscopy-based detection is not feasible in clinical practice [164]. PCR based diagnosis is also very difficult and shows highly variable sensitivity [164]. Direct detection methods are not always feasible for clinicians because of extended processing time and required expertise [16]. The diagnosis of EM is a challenging task because EM can create different patterns instead of the trademark bull's-eye pattern as shown in Figure 2.8b.

Despite the vast application of AI in the field of skin lesion diagnosis, there are only a few works related to Lyme disease detection from EM skin lesion images. The unavailability of reliable public EM datasets as a result of privacy concerns of medical data may be the reason for the lack of extensive studies in this field. The only publicly available dataset of EM is a small collection of web-scraped unverified images hosted in a kaggle repository [187]. Čuk et al. [29] proposed a visual system for EM recognition on a private EM

dataset using classical machine learning techniques including naïve Bayes, SVM, boosting, and neural nets (not deep learning). They considered ellipse, the common shape of EM, and used eccentricity, small and large axis ratio, ellipse angular, and ellipse focus attributes for classification. Deep learning techniques learn image features from training images via an optimization process and recent studies show that image features extracted by deep learning techniques outperform human-engineered image features for medical image classification tasks [16]. Burlina et al. [15] created a dataset of EM by collecting images from the internet and trained a CNN architecture ResNet50 as a binary classifier to distinguish between EM and other skin conditions. Although their dataset is not public, the trained model is publicly available. Burlina et al. [16] further enriched the dataset with more images from the East Coast and Upper Midwest of the United States and trained six CNNs namely ResNet50, InceptionV3, MobileNetV2, DenseNet121, InceptionResNetV2, and ResNet152 for EM classification. They did not make the dataset or the trained models public for the extended study. Burlina et al. [15] and Burlina et al. [16] used transfer learning from ImageNet pre-trained models and studied the CNNs in terms of predictive performance. Koduru et al. [81] deployed a trained ResNet50 CNN model utilizing a private dataset for a prototype application to identify EM. Jacob et al. [74] used the kaggle Lyme dataset [187] to test state-of-the-art self-supervised learning techniques against supervised transfer learning for different CNN architectures and comparatively self-supervised learning underperformed. Oholtsov et al. [115] suggested using both clean and dirty images for training EM classifier based on experimentation using an unverified dataset of only 106 EM images.

(a) Bull's-eye pattern.

(b) Atypical pattern.

Figure 2.8: Patterns of erythema migrans (EM) [38].

2.2.3 Related Works on Data Scarcity

Transfer learning and expanding data by transforming images with augmentation techniques are frequently used by researchers to improve deep learning model's performance on limited image datasets. Perez et al [120] showed that using data augmentation significantly improves model's performance using InceptionV4, ResNet, and DenseNet architectures for melanoma classification with dermoscopic images from ISIC archive. Pérez et al. [121] showed that combining transfer learning and data augmentation significantly improves model's performance for melanoma diagnosis from images using an extensive experimental study on 11 datasets and 12 CNN architectures.

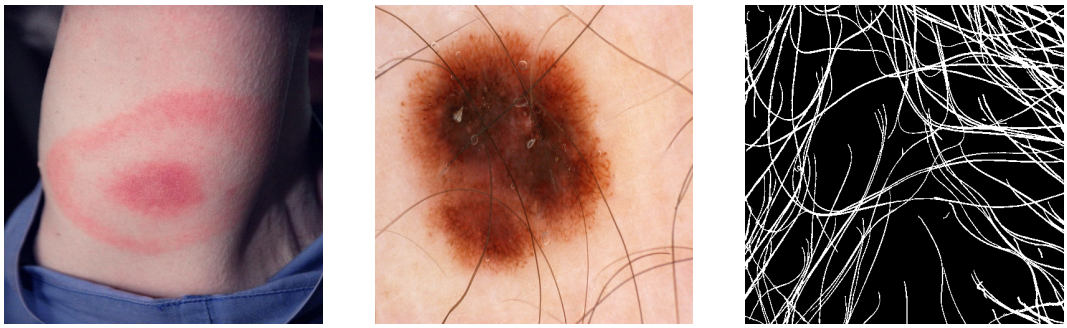
Transfer learning with supervised ImageNet[135] pre-training is frequently used in medical image analysis tasks [44, 53, 95, 105, 106, 176]. Transfer learning from natural images of ImageNet provides performance improvement according to multiple empirical studies [4, 44, 53]. Even if this strategy does not guarantee an improvement in performance Raghu et al. [125] showed using a detailed study that it speeds up convergence and is especially helpful for training with limited data. Gu et al. [45] showed that progressive transfer learning starting from an ImageNet pre-trained model end-to-end fine-tuned on a dataset of similar skin lesions with a slight domain shift increases the classification performance of skin cancer classification task for a smaller dataset. Recently, self-supervised pre-training using unlabeled domain-specific data is gaining popularity in medical image analysis [6, 31, 52, 93, 148, 188]. Azizi et al. [6] showed that training a model on ImageNet in self-supervised fashion followed by self-supervised learning on unlabeled in-domain medical images, and fine-tuning end-to-end for downstream supervised tasks significantly improves model accuracy. They used ResNet architectures on X-ray and dermatology classification tasks for experimentation. Dadsetan et al. [31] also showed that combining ImageNet and domain-specific self-supervised pre-training gives better performance for Alzheimer's disease propagation from brain magnetic resonance imaging.

For multimodal training i.e. training utilizing all the available modalities (for example, image modality and patient data modality), the general assumption is that both the training and test data will have full and paired modalities [113, 184]. However, missing data is common for real-world clinical scenarios [69]. Existing works in the literature generally drop the incomplete samples or impute missing values [98, 186]. Some studies report improvements by dropping incomplete samples [114, 171]. Generative models are used in many studies for imputing missing modalities [18, 87, 118]. Ma et al. [98] approximated missing modality using modality priors learned from dataset. Chen et al. [21] fused multimodal incomplete data using a heterogeneous graph structure. Zhang et al. [186] considered auxiliary data from similar neighbors of a patient to deal with missing modalities. If paired data is missing for the modalities but training data for individual modalities are available then individual classifiers can be trained for each modality and the results can be fused using a weighing scheme [126].

2.3 Research Questions and Challenges

Pre-training with in-domain images is effective for increasing the performance of deep CNN based image classifiers however it is difficult to collect a large number of in-domain images for many skin conditions like EM. The dataset created as part of our study for EM analysis consists of clinical skin lesion images. First, we thought of pre-training a CNN for EM classification using clinical skin lesion images of other skin lesions as we could not collect a large number of in-domain unlabeled samples related to EM. Most of the accessible datasets are concerned with skin cancers. Wen et al. [173] systematically reviewed the available datasets for skin cancers. Out of the open access clinical skin lesion image datasets, SD-198 [154] containing 6,584 clinical skin lesion images and its extended version SD-260 [180] containing 20,600 clinical skin lesion images of skin cancer images seemed promising but these datasets are not easily accessible [1]. On the contrary, dermoscopic image datasets like HAM10000 dataset [165] are easily accessible. The image modality of clinical EM dataset is quite different from dermoscopic images of HAM10000 dataset as shown in Figure 2.9a and 2.9b. Our first research question is:

Research Question 1. *Can we improve the performance of ImageNet pretrained clinical skin lesion image classifier's performance with additional pre-training using dermoscopic images?*



(a) Clinical image. (b) Dermoscopic image. (c) Skin lesion hair mask.

Figure 2.9: Clinical image of erythema migrans [38] vs demoscopic image of skin cancer [165] and a sample of skin lesion hair mask.

Incorporating data from multiple modalities in the analysis process significantly improves the models' performance for skin lesion analysis tasks. For some diseases like Lyme disease, a proper diagnosis based on skin lesions is not effective without considering additional context from patient data. Existing works on early Lyme disease prediction using

[1] The download link mentioned in the paper did not work and we did not get any response from the authors after asking for access.

artificial intelligence techniques only utilize images of EM skin lesions whereas doctors believe corresponding patient data should also be considered to strengthen the predictive performance [16, 61]. Training a multimodal deep learning model utilizing both images and patient data requires a dataset of lesion images with associated patient data. Even though EM image datasets are available, creating a dataset with patient data linked with each lesion image would take much time. Moreover, patient data-only datasets that can be used for creating individual classifiers for Lyme disease are not readily available. So, our second research question is:

Research Question 2. *How to assist deep learning based skin lesion image classifier with patient data in the absence of training data?*

Occlusion of skin lesions in dermoscopic images due to hair artifacts (as shown in Figure 2.9b) affects the performance of computer-assisted lesion analysis algorithms. To tackle this issue, researchers are working on digital hair segmentation, removal, and augmentation techniques [5, 88]. Standard image processing based hair removal is not beneficial for real-time application and removing hair does not give new features to the network. Augmenting images with skin hair can be of interest. Skin hair augmentation techniques require a hair mask to generate hair in given locations as shown in Figure 2.9c [5]. These masks are created either manually, with random curves or lines and segmentation [5]. Generative models can be utilized to automate the creation of hair masks. So, our third research question is:

Research Question 3. *How to efficiently deal with skin lesion hair artifacts for AI-assisted analysis of dermoscopic skin lesion images?*

2.4 Conclusion

In this chapter, we have provided all the theoretical backgrounds needed to understand the rest of the thesis. In addition, we presented a detailed review of existing works in the literature and framed our research questions in the context of existing works. The following chapters describe our contributions by addressing the research questions presented in this chapter.

Key Points (Chapter 2)

In this thesis, we are addressing three research questions:

- Can we improve the performance of ImageNet pretrained clinical skin lesion image classifier's performance with additional pre-training using dermoscopic images?
- How to assist deep learning based skin lesion image classifier with patient data in the absence of training data?
- How to efficiently deal with skin lesion hair artifacts for AI-assisted analysis of dermoscopic skin lesion images?

3 Pre-training Strategy for Improving Clinical Skin Lesion Image Classifier's Performance Using Dermoscopic Images

This chapter addresses research question 1, presents our pre-training strategy for improving clinical skin lesion image classification performance of ImageNet pre-trained convolutional neural networks by utilizing additional pre-training with dermoscopic images. It also contains benchmarking of state-of-the-art convolutional architectures for Lyme disease image classification. Contents from this chapter have been used in the following article:

- S. I. Hossain, J. de Goër de Herve, M. S. Hassan, D. Martineau, E. Petrosyan, V. Corbin, J. Beytout, I. Lebert, J. Durand, I. Carravieri, A. Brun-Jacob, P. Frey-Klett, E. Baux, C. Cazorla, C. Eldin, Y. Hansmann, S. Patrat-Delon, T. Prazuck, A. Raffetin, P. Tattevin, G. Vourc'h, O. Lesens, and E. Mephu Nguifo. "Exploring convolutional neural networks with transfer learning for diagnosing Lyme disease from skin lesion images". *Computer Methods and Programs in Biomedicine* 215, 2022, p. 106624. ISSN: 01692607. DOI: 10.1016/j.cmpb.2022.106624

Chapter Contents

3.1 Introduction

Our pre-training strategy involves fine-tuning some layers from the end of an ImageNet pre-trained convolutional neural network (CNN) architecture using a dermoscopic dataset before training the model on a clinical skin lesion dataset. Our intuition behind the approach is the fact that even though the image modality of dermoscopic and clinical skin lesion images are different, pre-training some layers from the end of the model with dermoscopic images should provide a good feature representation for starting training with clinical skin lesion images. As the layers at the end of a CNN architecture respond to task-specific complex patterns, there should be a similarity between the skin lesion patterns of clinical and dermoscopic skin lesion images. We tested our strategy using dermoscopic images of skin cancer from HAM10000 dataset [165] and clinical skin lesion images related to erythema migrans (EM).

As there is no expert-verified publicly available Lyme dataset of EM images, first, we created a dataset consisting of 866 images of confirmed EM lesions. Images collected from the internet and Clermont-Ferrand University Hospital Center (CF-CHU) of France were carefully labeled into two classes: EM and Confuser, by expert dermatologists and infectiologists from CF-CHU. CF-CHU collected the images from several hospitals in France.

Lightweight CNN-based mobile applications can help people with an initial self-assessment of EM and refer them to expert dermatologist for further diagnosis. Also, resource-intensive CNN-based computer applications can assist non-expert practitioners in identifying EM. In this chapter, besides testing our pre-training strategy, we also studied the performance of state-of-the-art CNNs for diagnosing Lyme disease from EM images and to find out the best architecture based on different criteria. We benchmarked twenty-five well-known CNNs on the prepared dataset in terms of several predictive performance metrics, computational complexity metrics, and statistical significance tests. Alongside our proposed pre-training strategy other best practices for training CNNs on limited data were used. For visualizing the regions of the input image that are significant for predictions from the CNN models we used gradient-weighted class activation mapping (Grad-CAM) [141]. We provided guidelines for model selection based on predictive

performance and computational complexity. Moreover, we made all the trained models publicly available which can be used for transfer learning and building pre-scanners for Lyme disease. Figure 3.1 presents the graphical overview of the study performed in this chapter.

The rest of the chapter is structured as follows: Section 3.2 contains the proposed pre-training strategy, dataset description, a brief overview of CNN architectures, and performance measures; Section 3.3 presents experimental studies, results, discussion, and recommendations for model selections; finally, Section 3.4 presents concluding remarks.

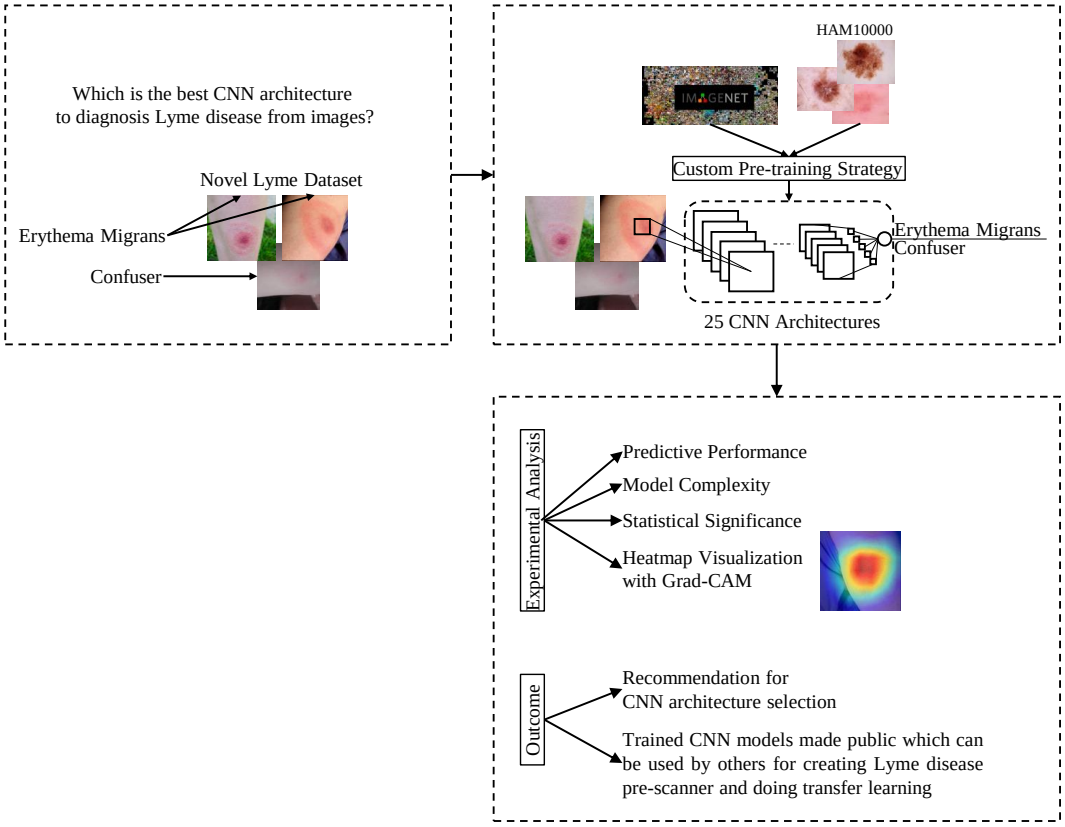


Figure 3.1: Graphical overview of the study on the effectiveness of CNNs utilizing custom pre-training strategy for the diagnosis of Lyme disease from images.

3.2 Materials and Methods

The following subsections describe the pre-training strategy used in this study, the data organization, a short overview of the considered CNN architectures, performance measures, and heatmap visualization method.

3.2.1 Pre-training Strategy

Algorithm 1: Dermoscopic pre-training for clinical lesion image classification

Data :
ImageNet pre-trained CNN model without classfication head: M_I
Total layers in M_I: N_L
Dermoscopic image classification task: $\mathcal{T}_S$
Dermoscopic image dataset: D_S
Clinical skin lesion image classification task: $\mathcal{T}_T$
Clinical skin lesion image dataset: D_T

Output:
CNN model optimized for $\mathcal{T}_T$: M_T

begin
- $M_I \leftarrow$ freeze all the layers of M_I `// make layers non-trainable`
- $U_L \leftarrow$ findLayersToUnfreeze(M_I, N_L, $\mathcal{T}_T$, D_T)
- $M_S \leftarrow$ add classifier head to M_I for $\mathcal{T}_S$
- `/* models are trained and validated using training and validation subsets of the respective dataset */`
- $M_S \leftarrow$ train M_S and fine-tune after unfreezing layers $N_L - U_L + 1$ to N_L on D_S
- $M_S \leftarrow$ freeze all the layers of M_S
- $M_T \leftarrow$ from M_S remove classifier head for $\mathcal{T}_S$ and add classifier head for $\mathcal{T}_T$
- $M_T \leftarrow$ train M_T and fine-tune after unfreezing layers $N_L - U_L + 1$ to N_L on D_T
- **return** M_T

Function *findLayersToUnfreeze (M, N, $\mathcal{T}$, D)*
- $M_T \leftarrow$ add classifier head to M for $\mathcal{T}$
- $\tilde{M_T} \leftarrow$ train $M_\mathcal{T}$ using training data from D
- $U \leftarrow 0$ `// number of layers to unfreeze`
- $max \leftarrow 0$ `// tracks best model accuracy on validation data`
- **for** $i \leftarrow 1$ ***to*** N **do**
  - $M_T \leftarrow$ unfreeze layers $N - i + 1$ to N of M_T `// make layers trainable`
  - $M_T \leftarrow$ fine-tune M_T using training data from D
  - $temp \leftarrow$ measure accuracy of M_T on validation data from D
  - **if** $temp > max$ **then**
    - $max \leftarrow temp$
    - $U \leftarrow i$
  - $M_T \leftarrow \tilde{M_T}$
- **return** U

Algorithm 1 shows the steps of our pre-training strategy. We start with an ImageNet pre-trained CNN model M_I without the original ImageNet classification head. The total

layers in M_I is N_L. The source dermoscopic image classification task and dataset are $\mathcal{T}_S$, and D_S, respectively. The target clinical skin lesion image classification task and dataset are $\mathcal{T}_T$, and D_T, respectively. Initially, all the layers of M_I are frozen to make sure the parameters are not updated during training. Then, we find out the best number of layers U_L of M_I to unfreeze and fine-tune for $\mathcal{T}_T$ using the function *findLayersToUnfreeze*. The function first adds a classification head to M_I for $\mathcal{T}_T$ and trains the model using D_T. Then, the function trains the model by making different frozen layers trainable and the number of unfrozen layers giving the best validation accuracy is returned. M_I is again pre-trained and fine-tuned using D_S based on U_L. For that purpose, we add classification head to M_I for $\mathcal{T}_S$ resulting in a model called M_S. M_S is trained and fine-tuned after unfreezing layers $N_L - U_L + 1$ to N_L using D_S. Then all the layers of M_S are made non-trainable again. After pre-training and fine-tuning the unfrozen part of the model for $\mathcal{T}_S$, the learned feature representation is reused for $\mathcal{T}_T$. We remove the classifier head for $\mathcal{T}_S$ from M_S and add the classifier head for $\mathcal{T}_T$ resulting in a model called M_T. Finally, M_T is trained and fine-tuned after unfreezing layers $N_L - U_L + 1$ to N_L using D_S. We tested the proposed pre-training strategy on EM classification task as shown in Figure 3.2. The source dermoscopic image classification task $\mathcal{T}_S$ and dataset D_S are skin cancer classification and HAM10000 dataset respectively. The target clinical skin lesion image classification task $\mathcal{T}_T$ and dataset D_T are EM classification and Lyme dataset respectively. Our EM classification head consists of global average pooling (GAP) layer [90], dropout layer [149], and a fully connected layer with sigmoid activation for binary classification. Each channel in the feature map is averaged over the whole spatial extent by GAP, and the end result is a single value for each channel that summarizes the spatial information of the feature map. Dropout is a deep learning regularization method used to avoid overfitting. During each training iteration, a portion of the units or neurons in a layer are randomly dropped out (i.e., set to zero) by the dropout layer. As it can no longer rely on any one neuron, this forces the network to learn more reliable and generalizable features.

The intuition behind our approach is the fact that CNN mimics the ventral visual stream process of the human brain [189]. Figure 3.3 shows the outline of the ventral visual stream. The first visual area V1 receives optical input from the retina through the optic nerve and the lateral geniculate nucleus. V1 responds to very simple patterns (edges, lines). As the input traverses through the stream V2 and V4 respond to simple and complex shapes respectively. Further down the path inferotemporal areas respond to complex patterns of semantic entities for object understanding [123]. CNNs replicate the behavior of the ventral visual stream. The initial layers of CNN learn/respond to simple patterns and the later layers respond to complex and semantic patterns. The unfrozen layers at the end of ImageNet pre-trained model giving good performance when fine-tuned on clinical skin lesion image dataset learn task-specific patterns that relate to clinical skin lesions and there should be a similarity between the skin lesion patterns of clinical and dermoscopic skin lesion images. So, fine-tuning the unfrozen part first with a dermoscopic

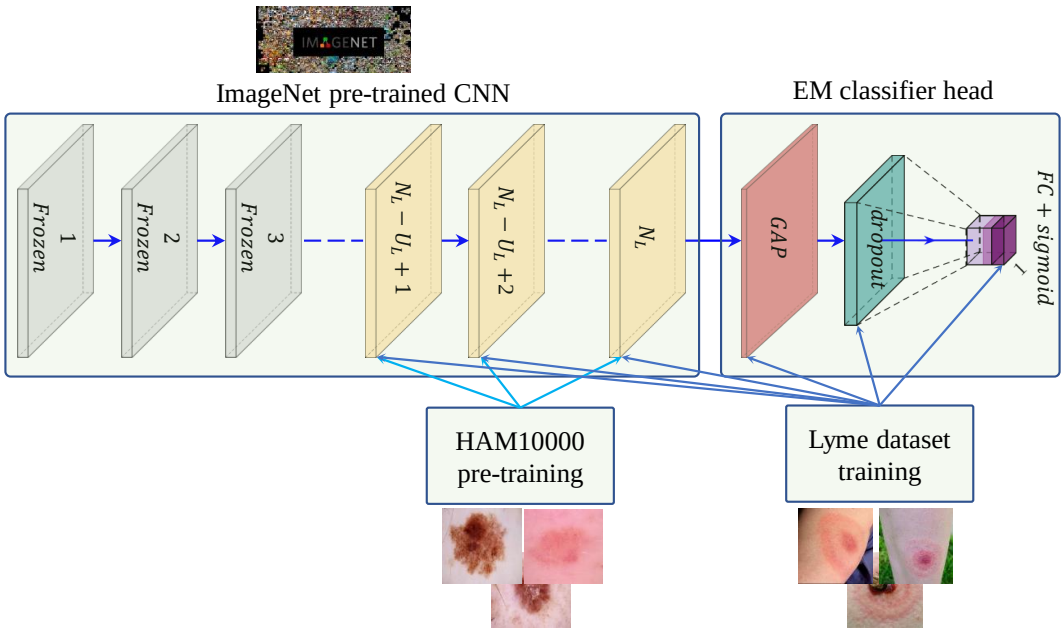


Figure 3.2: Pre-training strategy applied to erythema migrans (EM) classification. GAP and FC stand for global average pooling and fully connected layer respectively. N_L is the number of ImageNet pre-trained layers and U_L represents the number of layers used for fine-tuning.

image dataset should provide a good feature representation and weight initialization of layers for starting training with a clinical skin lesion image dataset.

3.2.2 Dataset Preparation

As a labeled public dataset is not available for Lyme disease prediction from EM images, we created a dataset by collecting clinical skin lesion images from the internet and CF-CHU. CF-CHU collected EM images from several hospitals located in France. The use of images from the internet was inspired by related skin lesion analysis studies [15, 16, 34]. Duplicate images were removed using an image hashing-based duplicate image detector followed by the removal of inappropriate images through human inspection. After the initial curation steps, we got a total of 1672 images. Expert dermatologists and infectiologists from CF-CHU classified the curated images into two categories: EM and Confuser, making it a two-class classification problem. Out of 1672 images, 866 images were assigned to EM class and 806 images were assigned to Confuser class. The images collected from the hospitals are not shareable because of the confidentiality agreement signed with the patients. We can share images collected from the internet and the corresponding la-

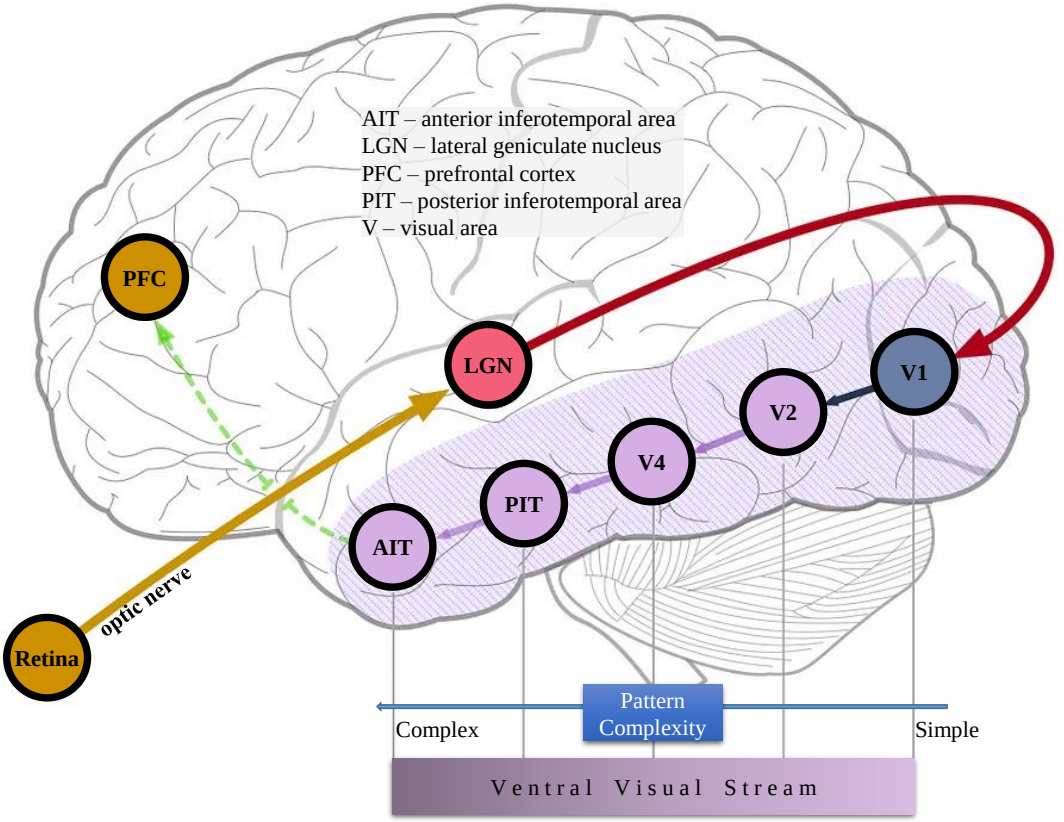


Figure 3.3: Outline of ventral visual stream. Image modified from [28].

bels assigned by the doctors subject to an agreement that the images will not be made public because we do not have permission from the owners of the images [1].

We further subdivided the dataset into five folds using stratified five-fold cross-validation to make sure each of the folds maintains the original class ratio. One of the folds was used as a test set and the remaining four were used as the training set with a rotation of the folds for five runs. Each time, 10% of the training data was assigned to the validation set as shown in Figure 3.4.

Deep CNNs require a considerable amount of data for training and data augmentation can help with expanding the dataset. We applied data augmentation techniques only to the training images. We used flip (vertical or horizontal), rotation, brightness, contrast, and saturation augmentation by considering the best performing augmentations for skin lesions [120]. Besides, we also used perspective skew transformation to cover the case of looking at a picture from different angles. Augmentor [11] an image augmentation library specially built for biomedical image augmentation was used for applying the augmentations. We used 0.5 as the probability of applying each of the augmentation operations.

[1] Interested researchers can send a request to dappem-project@inrae.fr for the data.

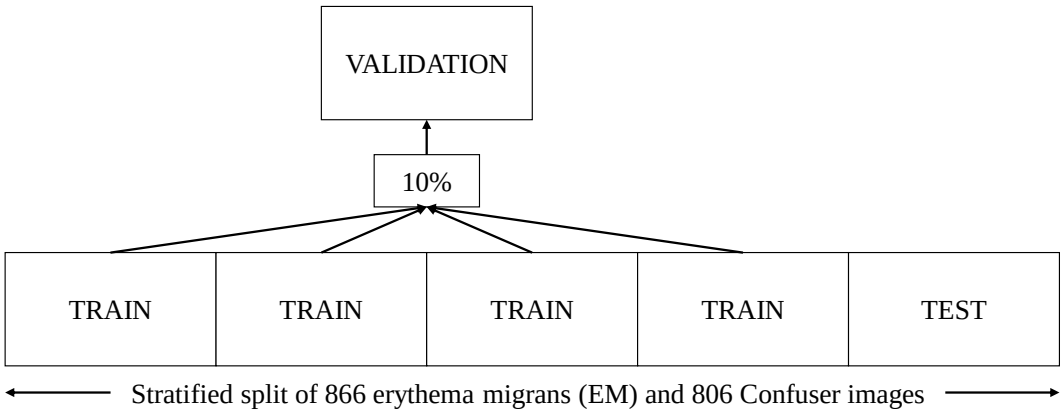


Figure 3.4: Five-fold cross-validation setup.

Rotation operation was performed with a maximum rotation angle of 5 degrees. We also used random rotation by either 90, 180, or 270 degrees. Brightness, contrast, and saturation augmentations were performed with a minimum adjustment factor of 0.7 and a maximum adjustment factor of 1.3. For all the other parameters we used default values provided by Augmentor library. The parameters were adjusted based on the visual inspection of augmented images. Figure 3.5 shows some example images resulting from augmentations applied on a sample image.

Figure 3.5: Data augmentation examples.

3.2.3 Brief Overview of the CNN Architectures Considered in the Study

Starting with LeNet [83] in 1988 the popularity of CNNs increased with AlexNet [82] winning the ImageNet large scale visual recognition challenge (ILSVRC) [135] of 2012. As a result of the effectiveness of CNNs in solving complex problems, several CNN architectures have been introduced over the past few years. The following subsections provide a brief overview of the CNN architectures used in this study.

3.2.3.1 VGG Architecture

VGG architecture [147] is based on the idea of deeper networks with smaller filters (3×3). There are thirteen convolutional layers and three fully connected layers in VGG16 architecture as shown in Figure 3.6. Another variation of VGG architecture called VGG19 has sixteen convolutional layers and three fully connected layers. VGG architecture showed better effectiveness of deeper architectures in terms of predictive performance but requires training a huge number of parameters.

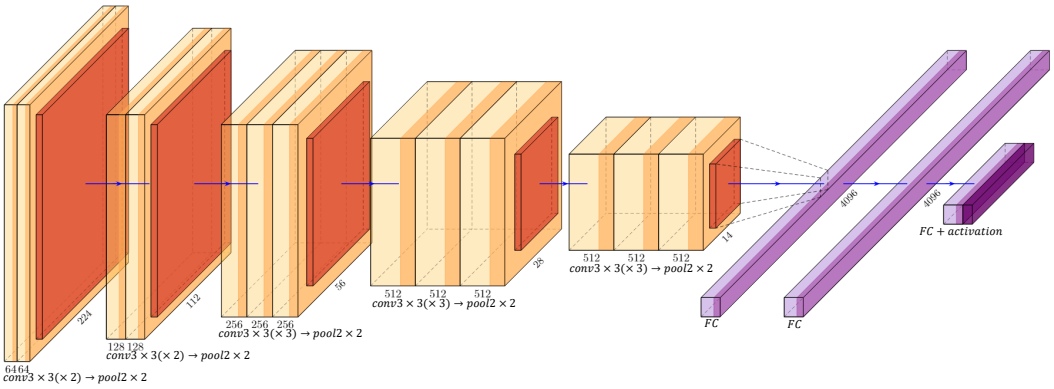


Figure 3.6: VGG16 architecture. Input image is of shape $224 \times 224 \times 3$. *FC* stands for fully connected layer.

3.2.3.2 Inception Architecture

Inception architecture [156] uses inception module as shown in Figure 3.7, which is a combination of several convolution layers with small filters $(1 \times 1,\ 3 \times 3,\ 5 \times 5)$ applied simultaneously on the same input to facilitate the extraction of more information. The output filter banks from the convolution layers of inception module are concatenated into a single vector, which is served as the input for next stage. To reduce learnable parameters and computational complexity inception module uses 1×1 convolution at the beginning of convolution layers. InceptionV1 architecture is the winner of ILSVRC

2014 competition, and it's also known as GoogleNet. Further improvement resulted in the creation of several versions of inception architectures named InceptionV2, InceptionV3, and InceptionV4 [155, 157]. InceptionV2 and InceptionV3 improved the architecture with smart factorized convolution, batch normalized auxiliary classifier, and label smoothing whereas, InceptionV4 focused on the uniformity of the architecture with more inception modules than InceptionV3.

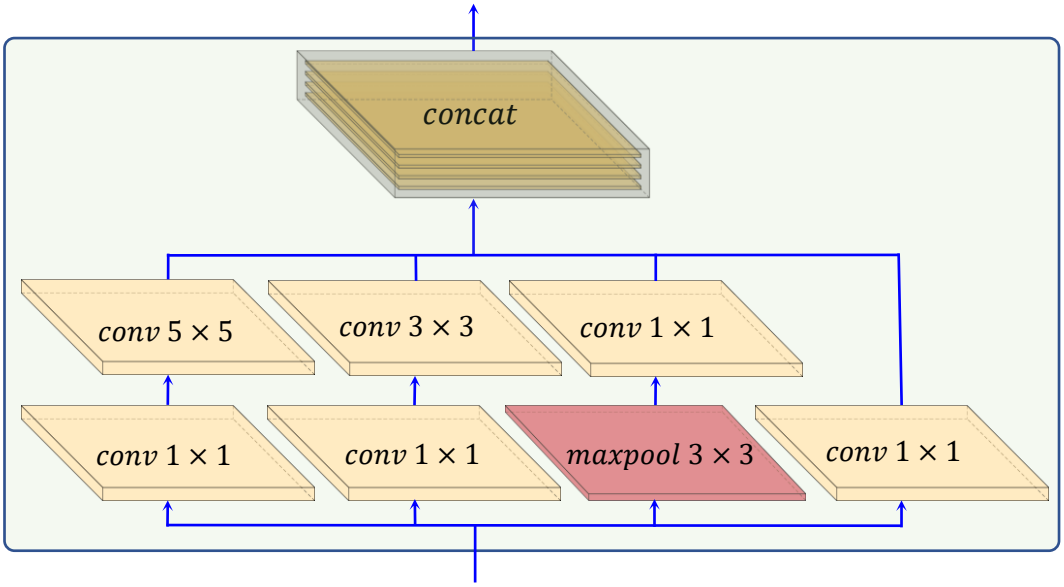


Figure 3.7: Inception module of Inception architecture. "concat" represents the concatenation of feature maps.

3.2.3.3 ResNet Architecture

ResNet architecture [50] tried to solve the vanishing gradient and accuracy degradation problems of deep models by introducing residual block with identity shortcut connection that directly connects the input to the output of the block allowing the gradient to flow through the shortcut path as shown in Figure 3.8. It's the winner of ILSVRC 2015 competition. Depending on the number of weight layers there are many variants of ResNet architecture such as ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, ResNet164, ResNet1202, etc., where the number represents the count of weight layers. Deeper ResNet architectures use bottleneck blocks where 3x3 convolution is sandwiched between 1×1 convolutions, responsible for transitory reduction and expansion of channels. The $wide \rightarrow narrow \rightarrow wide$ architecture of bottlenecks reduces multiplications and the number of parameters and helps the network grow deeper with fewer parameters. He et al. [51] proposed ResNetV2 with pre-activation of the weight layers as opposed to

the post-activation of original ResNet architecture. InceptionResNet is a hybrid of Inception and ResNet architecture having two variations named InceptionResNetV1 and InceptionResNetV2, which differ mainly in terms of the number of used filters [155]. Liu et al. [96] modernized ResNet architecture to match the performance of vision transformers resulting in a new family of architectures called ConvNeXt. ConvNeXt utilizes inverted bottleneck ($narrow \rightarrow wide \rightarrow narrow$), large kernel, depthwise convolution (shown in Figure 3.10), layer normalization [7] instead of batch normalization [72], and GELU activation ReLU as compared to base ResNet models.

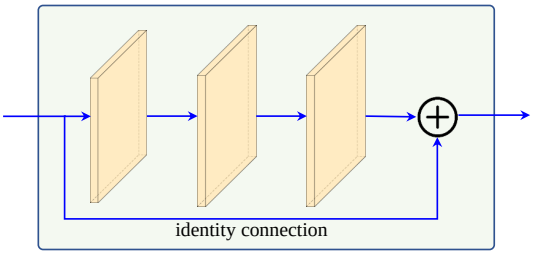


Figure 3.8: Residual block of ResNet architecture.

3.2.3.4 DenseNet Architecture

Dense Convolutional Network (DenseNet) [68] extended ResNet by introducing dense blocks where each layer within a dense block receives inputs from all the previous layers as shown in Figure 3.9 DenseNet concatenates the incoming feature maps of a layer with output feature maps instead of summing them up as done in ResNet. Dense blocks within DenseNet are connected with transition layers consisting of convolution and pooling to perform the required downsampling operation. Depending on the number of weight layers there are several versions of DenseNet like DenseNet121, DenseNet169, DenseNet201, DenseNet264, etc. Besides solving the vanishing gradient problem DenseNet also eases feature propagation and reuse, and a reduction in the number of learnable parameters compared to ResNet.

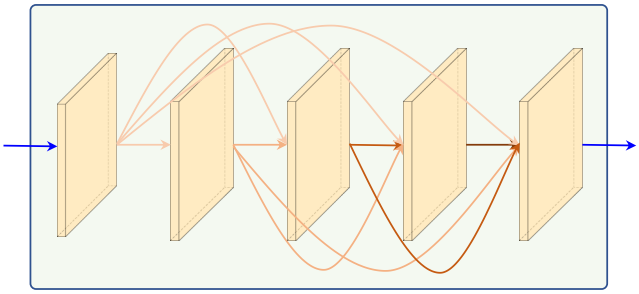


Figure 3.9: Building block of DenseNet architecture.

3.2.3.5 MobileNet Architecture

MobileNetV1 [65] used depthwise separable convolution extensively to reduce the computational cost. Standard convolution performs spatial and channel-wise computations in one step but depthwise separable convolution first applies separate convolutional filter for each input channel and then uses pointwise convolution on concatenated channels to produce required number of output channels as shown in Figure 3.10. MobileNetV1 was designed to run very efficiently on mobile and embedded devices. MobileNetV2 [137] improved upon the concepts of MobileNetV1 by incorporating thin linear bottlenecks with shortcut connections between the bottlenecks as shown in Figure 3.11. This is called inverted residual block as it uses $narrow \rightarrow wide \rightarrow narrow$ as opposed to the $wide \rightarrow narrow \rightarrow wide$ architecture of traditional residual block. MobileNetV3 [64] incorporated squeeze-and-excitation layers [67] in the building block of MobileNetV2 which provides channel-wise attention and used MnasNet [158] to search for a coarse architecture that was further optimized with NetAdapt [181] algorithm.

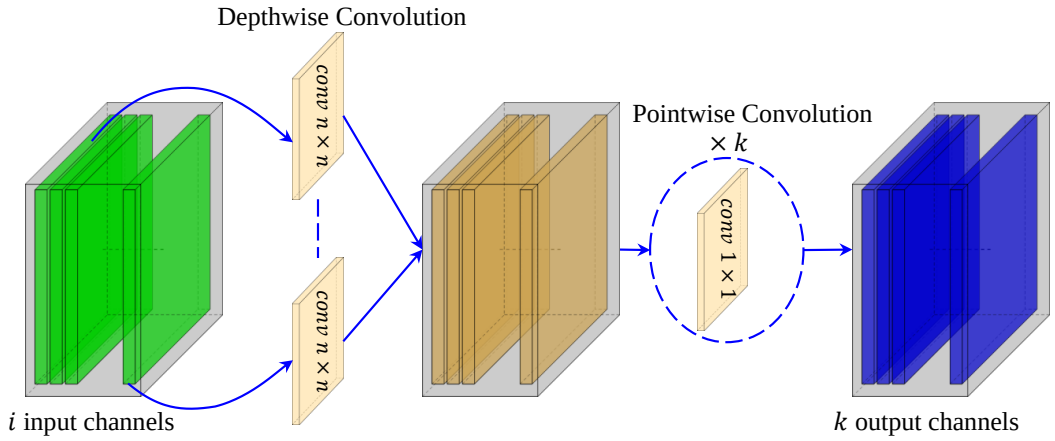


Figure 3.10: Depthwise separable convolution.

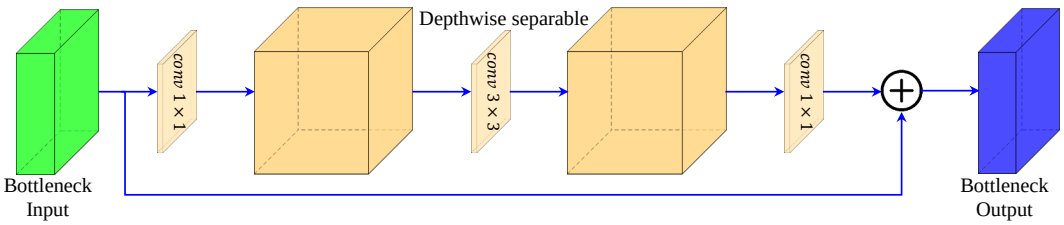


Figure 3.11: Building block of MobileNetV2 architecture.

3.2.3.6 Xception Architecture

Extreme version of Inception the Xception architecture [24] replaced the Inception module with a modified version of depthwise separable convolution where the order of depthwise convolution and pointwise convolutions are reversed as shown in Figure 3.12. Xception also uses shortcut connections like ResNet architecture. On ImageNet dataset Xception performs slightly better than the InceptionV3 architecture.

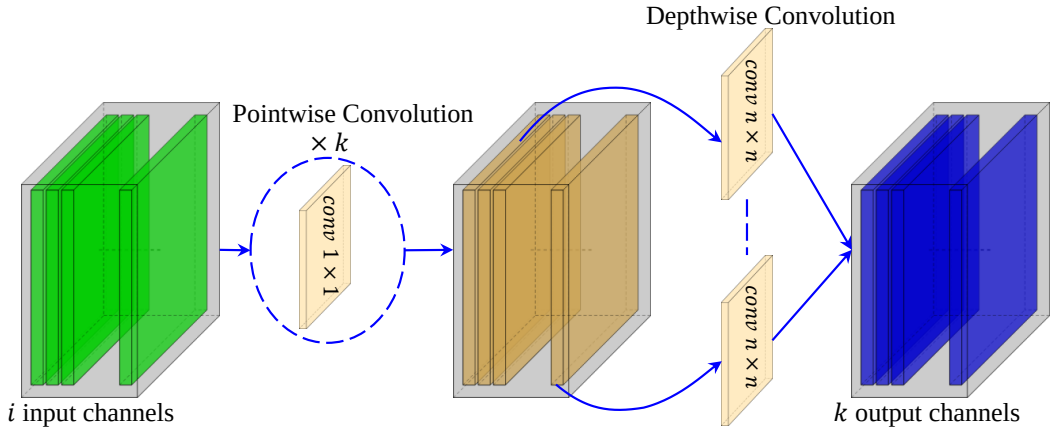


Figure 3.12: Building block of Xception architecture.

3.2.3.7 NASNet Architecture

Neural Architecture Search Netowork [190] from Google Brain utilizes reinforcement learning with a Recurrent Neural Network based controller to search for efficient building blocks for a smaller dataset which is then transferred to a larger dataset by stacking multiple copies of the found building block. NASNet blocks are comprised of normal and reduction cells as shown in Figure 3.13. Normal cells produce feature maps of the same size as input whereas reduction cells reduce the size by a factor of two. NASNet optimized for mobile applications is called NASNetMobile whereas the larger version is called NASNetLarge.

3.2.3.8 EfficientNet Architecture

EfficientNet [159] which is among the most efficient models proposed a scaling method to uniformly scale all dimensions of a network using a compound coefficient. The baseline network of EfficientNet was built with NAS incorporating squeeze-and-excitation in the building block of MobileNetV2. EffcientNet's building block also called MBConv is shown in Figure 3.14a. The scaling method is defined as:

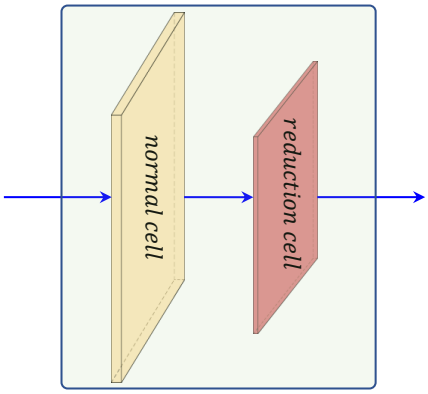
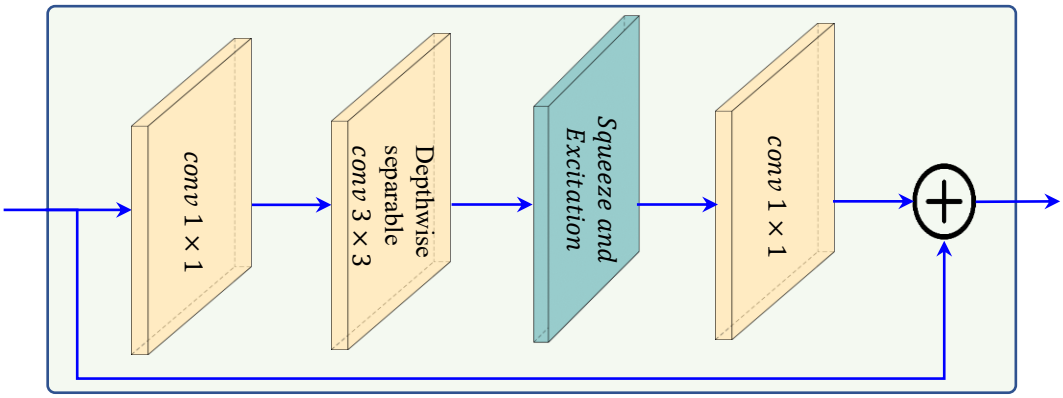
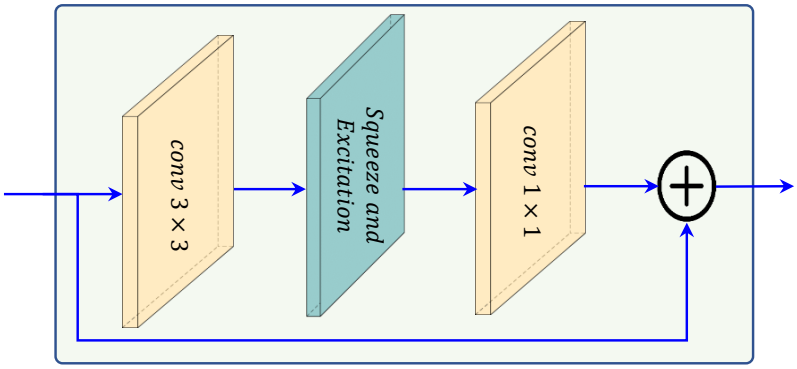


Figure 3.13: Building block of NASNet architecture.



(a) MBConv block.



(b) Fused-MBConv block.

Figure 3.14: EfficientNet building blocks.

$$depth = \zeta_1^{\Theta}$$
$$width = \zeta_2^{\Theta}$$
$$resolution = \zeta_3^{\Theta}$$
$$s.t.\ \zeta_1.\zeta_2^2.\zeta_3^2 \approx 2;\ \zeta_1 \geq 1, \zeta_2 \geq 1, \zeta_1 \geq 1$$

where, the coefficient Θ controls available resources and ζ_1, ζ_2, and ζ_3 are constants obtained by grid search. EfficientNetB0-B7 are a family of architectures scaled up from the baseline network that reflects a good balance of accuracy and efficiency. EfficientV2 was designed to optimize parameter efficiency and training speed. It used an additional Fused-MBConv block. Fused-MBConv uses 3×3 convolution instead of the 3×3 depthwise and 1×1 convolutions of MBConv as shown in Figure 3.14b. Although Fused-MBConv adds a small overhead it improves training speed compared to MBConv. EfficientV2 used training-aware NAS to find the best combination of MBConv and Fused-MBConv blocks.

3.2.4 Predictive Performance Measures

To compare the predictive performance of the trained CNN models we used accuracy, recall/sensitivity/hit rate/true positive rate (TPR), specificity/selectivity/true negative rate (TNR), precision/ positive predictive value (PPV), negative predictive value (NPV), Cohen's kappa coefficient (κ), Matthews correlation coefficient (MCC), positive likelihood ratio (LR+), negative likelihood ratio (LR−), F1-score, confusion matrix and area under the receiver operating characteristic (ROC) curve (AUC) metrics. Confusion matrix is a way of presenting the count of true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP) in a matrix format where the y-axis presents true labels and x-axis presents predicted labels. Accuracy measures the proportion of correctly classified predictions among all the predictions, and it is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Recall/sensitivity/hit rate/TPR measures the proportion of actual positives correctly identified, and it is expressed as:

$$Recall,\ Sensitivity,\ hitrate,\ TPR = \frac{TP}{TP + FN} \tag{3.2}$$

Specificity/selectivity/ TNR measures the proportion of actual negatives correctly identified, and it is expressed as:

$$Specificity,\ Selectivity,\ TNR = \frac{TN}{TN + FP} \tag{3.3}$$

Precision/ PPV measures the proportion of correct positive predictions, and it is calculated as:

$$Precision,\ PPV = \frac{TP}{TP + FP} \tag{3.4}$$

NPV measures the proportion of negative predictions that are correct, and it is calculated as:

$$NPV = \frac{TN}{TN + FN} \tag{3.5}$$

MCC provides a summary of the confusion matrix, and it is calculated as:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.6}$$

MCC value is in the range $[-1, +1]$, where 0 is like random prediction, $+1$ means a perfect prediction, and -1 represents inverse prediction. Cohen's kappa coefficient (κ) metric is used to assess inter-rater agreement which tells us how the model is performing compared to a random classifier, and it is calculated with the formula:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{3.7}$$

where p_o is the relative observed agreement among the raters and p_e is the hypothetical probability of expected agreement which is defined for c categories as:

$$p_e = \frac{1}{N^2} \sum_c n_{c1} n_{c2} \tag{3.8}$$

where, N is the total number of observations, and n_{cr} is the number of predictions of category c by rater r. The value of κ is in the range $[-1, +1]$, where a value of 1 indicates perfect agreement, 0 means agreement only by chance, and a negative value indicates the agreement is worse than the agreement by chance. Likelihood ratio (LR) is used for assessing the potential utility of performing a diagnostic test and it is calculated for both positive test and negative test results called LR+ and LR−, respectively. LR+ is the ratio of the probability of a person having a disease testing positive to the probability of a person without the disease testing positive. LR− is the ratio of the probability of a person having the disease testing negative to the probability of without the disease testing negative. LR+ and LR− are calculated based on sensitivity and specificity values using the following formulas:

$$LR+ = \frac{sensitivity}{1 - specificity} \tag{3.9}$$

$$LR- = \frac{1 - sensitivity}{specificity} \tag{3.10}$$

A value of LR greater than 1 shows increased evidence. F1-Score combines precision and recall, and it is defined as the harmonic mean of precision and recall as follows:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.11}$$

ROC curve is a plot of TPR against false positive rate (FPR) at various threshold settings where FPR is defined as:

$$FPR = \frac{FP}{FP + TN} \tag{3.12}$$

Area under the ROC curve (AUC) is the measure of the classifier's ability to separate between classes and the higher the AUC, the better the ability of the classifier for separating the positive class from the negative class. As our dataset is balanced so, accuracy can be considered a good measure of predictive performance [23] and we did most of the analysis in terms of accuracy but also kept the other metrics to provide insights for experts from different domains as done in relevant studies [15, 16].

We have used critical difference (CD) diagram [33] to rank the CNN models in terms of accuracy and to show the statistically significant difference in predictive performance. A thick horizontal line connects a group of models in the CD diagram that are not significantly different in terms of predictive performance. We used non-parametric Friedman test [41] to reject the null hypothesis of statistical similarity among all the models followed by Nemenyi post-hoc all-pair comparison test [111] for showing the difference among the models at a significance level, $\alpha = 0.1$. Although deep CNN architectures often do not show any statistically significant differences when tested on large and small image datasets [16, 78, 191], CD diagram is a good way to visualize the multi-fold rank comparisons of the models.

3.2.5 Model Complexity Measures

To compare the trained CNNs in terms of complexity we used the total number of model parameters, the total number of floating-point operations (FLOPs), average training time per epoch, GPU memory usage, and average inference time per image. FLOPs reveal how computationally costly a model is and we counted FLOPs for each of the models using TensorFlow profiler [47] considering a batch size of one. For reporting the average training time per epoch, we calculated the average of the training time of three epochs during transfer learning fine-tuning. We calculated the GPU memory usage of a CNN model by inspecting the memory allocated in the GPU after loading a trained instance of the model. To measure the average inference time per image of a model we took the average of three hundred inferences on the same input image.

3.3 Experimental Studies

The following subsections describe experimental settings including model selection and parameter settings, software and hardware used for the study, the experimental results, and recommendations for model selection.

3.3.1 Experimental Settings

We did an extensive analysis using the ResNet50 architecture to see the effectiveness of our proposed pre-training strategy (described in Section 3.2.1) on the novel Lyme disease dataset. For this purpose, we tested different configurations:

i. Training ResNet50 model on our Lyme dataset from scratch without using transfer learning (called ResNet50-NTL, where, NTL stands for no transfer learning).

ii. Pre-training ResNet50 model with only HAM10000 data followed by fine-tuning all the layers with our Lyme dataset (called ResNet50-HAM-FFT, where HAM means HAM10000 and FFT stands for full fine-tuning).

iii. Training only the EM classifier head of an ImageNet pre-trained ResNet50 model with our Lyme dataset (called ResNet50-IMG-WFT, where IMG means ImageNet and WFT stands for without fine-tuning).

iv. Fine-tuning all the layers of ImageNet pre-trained ResNet50 model with our Lyme dataset (called ResNet50-IMG-FFT).

v. Fine-tuning U_L number of layers of an ImageNet pre-trained ResNet50 model with our Lyme dataset (called ResNet50-IMG-FTU_L, where FTU_L means fine-tuning U_L number of layers).

vi. Pre-training the whole ImgaeNet pre-trained ResNet50 model by HAM10000 data before fine-tuning U_L layers with our Lyme dataset (called ResNet50-IMG-HAMFP-FTU_L, where, HAMFP means full pre-training with HAM10000 dataset).

vii. Pre-training only the unfrozen U_L layers of a ImgaeNet pre-trained ResNet50 model with HAM10000 data before fine-tuning U_L layers with our Lyme dataset (called ResNet50-IMG-HAMPP-FTU_L, where, HAMPP means partial pre-training with HAM10000 dataset). This setting corresponds to **our proposed pre-training strategy** (described in Section 3.2.1).

viii. To see the effect of data augmentation, we trained a ResNet50 model without data augmentation and transfer learning (called ResNet50-NoAug, where NoAug means no data augmentation). All the other configurations were trained with data augmentation as described in Section 3.2.2.

According to the experimental results (discussed in Section 5.3), our proposed pre-training configuration ResNet50-IMG-HAMPP-FTU_L performed best, and we used this configuration for training twenty-five well-known CNNs to find out the effective architecture for diagnosing Lyme disease from EM images. For this benchmarking we trained VGG16[2], VGG19[2], ResNet50[2], ResNet101[2], ResNet50V2[2], ResNet101V2[2], InceptionV3[2], InceptionV4[2], InceptionResNetV2[2], Xception[2], DenseNet121[2], DenseNet169[2], DenseNet201[2], MobileNetV2[2], MobileNetV3Large[2], MobileNetV3Small[2], NASNetMobile[2], EfficientNetB0[2], EfficientNetB1[2], EfficientNetB2[2], EfficientNetB3[2], EfficientNetB4[2], EfficientNetB5[2], EfficientNetV2S[2] and ConvNextTiny[2] architectures. These models were selected to explore a diverse set of CNN models covering various prospects, like different architectures, depths, and complexities. For simplicity, the best performing trained models of each of the architectures are presented in ModelName-UU_L format, where U_L represents the number of unfrozen layers. For example, EfficientNetB0-187 means EfficientNetB0-IMG-HAMPP-FT187 and ResNet50-141 means ResNet50-IMG-HAMPP-FT141. To the best of our knowledge, ResNet50 is the only publicly available trained CNN that was used for Lyme disease identification by Burlina et al. [15]. We are calling this model ResNet50-Burlina which is a collection of five models (trained on five-fold cross-validation data)[3].

For training all the models, we used a dropout rate of 0.2 for the dropout layer in EM classifier head section. Adam optimizer (described in Appendix Section A.2) with author-recommended default values for parameters was used with a learning rate of 0.0001 for training the classifier head and 0.00001 for fine-tuning. We also used early stopping to terminate the training if there was no improvement in validation accuracy for ten epochs. A batch size of 32 was used. For reporting the number of layers to unfreeze during transfer learning, we stated the total number of layers to unfreeze including layers containing both trainable and non-trainable parameters.

We used NVIDIA QUADRO RTX 8000 GPU and a Desktop Computer with Intel Xeon W-2175 processor, 64 GB DDR4 RAM, and Ubuntu 18.04 operating system to perform all the experiments. Python v3.6.9, and TensorFlow v2.4.1 platform [1] were used for all the implementations and experiments of this study.

3.3.2 Results and Discussion

Table 3.1 presents the predictive performance measures of our eight different configurations (explained in Section 3.3.1). ResNet50-NoAug model resulting from training a ResNet50 architecture from scratch without using data augmentation and transfer learning gave an accuracy of 61.42%. ResNet50-NTL model obtained by training ResNet50 architecture with data augmentation and without transfer learning improved the accu-

[2]ImageNet pre-trained model taken from https://www.tensorflow.org/api_docs/python/tf/keras/applications (visited on 02/20/2023).

[3]https://github.com/neil454/lyme-1600-model (visited on 02/20/2023).

racy to 76.35%. So, data augmentation provided large gain in predictive performance (ResNet50-NTL compared to ResNet50-NoAug). ResNet50-HAM-FFT model resulting from pretraining ResNet50 architecture with only HAM10000 data followed by fine-tuning of all the layers with our Lyme dataset showed a degraded accuracy of 72.27%. ResNet50-IMG-WFT, generated by training only the EM classifier head of an ImageNet pre-trained ResNet50 architecture improved the accuracy to 78.94%. ResNet50-IMG-FFT, resulting from fine-tuning all the layers of ImageNet pre-trained ResNet50 architecture, further improved the classification accuracy to 82.22%. Whereas ResNet50-IMG-FT141, model resulting from fine-tuning 141 layers of pre-trained ResNet50 architecture gave an accuracy of 83.24% which is better compared to unfreezing the full architecture. ResNet50-IMG-HAMFP-FT141, model resulting from pretraining the whole ImgaeNet pre-trained ResNet50 model by HAM10000 data before fine-tuning 141 layers with our Lyme dataset reduced the accuracy to 82.35%. Our proposed pre-training strategy(described in Section 3.2.1) i.e. pre-training only the unfrozen 141 layers with HAM10000 data gave us the model ResNet50-IMG-HAMPP-FT141 with the best accuracy of 84.42%. Figure 3.15 shows the CD diagram in terms of accuracy for these ResNet50 based models. The Friedman test null hypothesis was rejected with a p value of 0.00003. From the CD diagram, we can see that ResNet50-IMG-HAMPP-FT141 achieved the best average ranking among the compared models. Although there is no statistically significant difference among ResNet50-IMG-FFT, ResNet50-IMG-FT141, ResNet50-IMG-HAMFP-FT141, and ResNet50-IMG-HAMPP-FT141 in terms of accuracy the ResNet50-IMG-HAMPP-FT141 model performed better in terms of most of the metrics (7 out of 11) as highlighted in Table 3.1. To summarize, our proposed strategy of pre-training only the unfrozen part of an ImageNet pre-trained CNN with HAM10000 data provided the best accuracy according to our experiments. So, for benchmarking all the other CNN architectures, we only reported the performance resulting from this configuration.

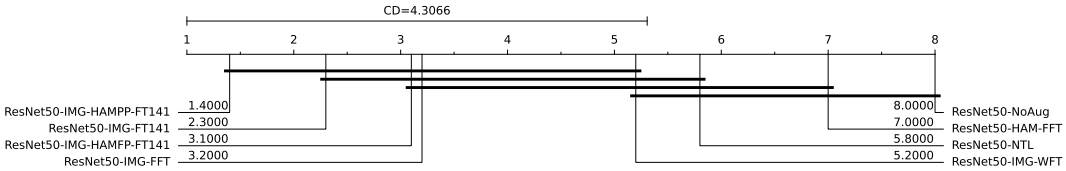


Figure 3.15: Accuracy critical difference diagram for ResNet50 based configurations. The models are ordered by best to worst average ranking from left to right. The number beside a model's name represents the average rank of the model. CD is the critical difference for Nemenyi post-hoc test. Thick horizontal line connects the models that are not statistically significantly different.

In the literature on recognizing EM from images, Čuk et al. [29] reported accuracies in the range of 69.23% to 80.42% using classical machine learning methods, and Burlina et

Table 3.1: Five-fold cross-validation performance metrics of ResNet50 based configurations. Within each cell, the value after ± symbol represents the standard deviation across five folds. Bold indicates the best result for each of the metrics.

	Metric										
Model	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
ResNet50-NoAug	61.42 ±1.29	71.73 ±8.65	50.37 ±8.79	61.0 ±1.5	63.03 ±3.17	0.2302 ±0.0234	0.2224 ±0.0256	1.4592 ±0.0863	0.5497 ±0.0764	0.656 ±0.0325	0.6505 ±0.0216
ResNet50-NTL	76.35 ±2.43	78.49 ±8.47	74.04 ±4.6	76.64 ±1.64	76.92 ±5.22	0.5305 ±0.0431	0.5261 ±0.0464	3.0735 ±0.2867	0.2853 ±0.0906	0.7723 ±0.0398	0.8471 ±0.0185
ResNet50-HAM-FFT	72.27 ±1.69	75.85 ±1.27	68.42 ±4.05	72.18 ±2.55	72.48 ±1.08	0.4447 ±0.0341	0.4435 ±0.0347	2.4434 ±0.3248	0.3536 ±0.0193	0.7393 ±0.0116	0.7979 ±0.0251
ResNet50-IMG-WFT	78.94 ±1.48	82.55 ±2.77	75.06 ±5.11	78.27 ±3.2	80.11 ±1.77	0.5799 ±0.03	0.5772 ±0.0305	3.4636 ±0.7671	0.2316 ±0.0255	0.8025 ±0.0101	0.8666 ±0.0163
ResNet50-IMG-FFT	82.22 ±1.36	85.27 ±2.67	78.93 ±5.26	81.55 ±3.42	83.42 ±1.63	0.6458 ±0.0262	0.6431 ±0.028	4.3127 ±1.0994	0.1854 ±0.0226	0.8326 ±0.0083	0.909 ±0.0092
ResNet50-IMG-FT141	83.24 ±1.04	85.29 ±2.27	**81.04** ±2.28	82.91 ±1.49	83.74 ±1.96	0.6649 ±0.0212	0.6641 ±0.021	4.5575 ±0.493	0.1812 ±0.0255	0.8405 ±0.0104	0.9134 ±0.0091
ResNet50-IMG-HAMFP-FT141	82.35 ±1.62	**89.28** ±2.42	74.91 ±5.11	79.45 ±3.05	**86.81** ±2.03	0.6521 ±0.0295	0.6448 ±0.0333	3.7072 ±0.7368	**0.1421** ±0.0251	0.84 ±0.0111	0.9113 ±0.0091
ResNet50-IMG-HAMPP-FT141	**84.42** ± 1.36	87.93 ± 1.47	80.65 ± 3.59	**83.1** ± 2.49	86.19 ± 1.27	**0.6893** ± 0.0263	**0.6874** ± 0.0277	**4.703** ± 0.8624	0.1493 ± 0.0155	**0.8541** ± 0.0106	**0.9189** ± 0.0115

al. [16] reported the best accuracy of 81.51% using ResNet50 architecture for the case of EM vs all classification problems. There was a common subset of images collected from the internet in both the dataset of Burlina et al. [15] and our Lyme dataset. ResNet50-Burlina model gave an accuracy of 76.05% when tested on our full dataset as shown in Table 3.2. Performance metrics for the best performing configuration of all the CNN

Table 3.2: Performance metrics of ResNet50-Burlina model trained by Burlina et al. [15] tested on the whole dataset of this study. Within each cell, the value after ± symbol represents the standard deviation across five folds. Bold indicates the best result for each of the metrics.

	Metric										
Model	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
ResNet50-Burlina	76.05 ±0.74	70.05 ±3.6	82.51 ±3.31	81.29 ±2.1	72.04 ±1.71	0.5294 ±0.0132	0.5229 ±0.0145	4.1017 ±0.5172	0.362 ±0.0309	0.7515 ±0.0137	0.481 ±0.0509

architectures used in this study are shown in Table 3.4. ResNet50-141 achieved the best accuracy of 84.42%. Most of the models except MobileNetV2-62, MobileNetV3Small-182, and NASNetMobile-617 showed good AUC values of above 90% and good sensitivity suggesting that these CNNs can be a good choice for building pre-scanners for Lyme disease. Figure 3.16 shows the CD diagram in terms of accuracy for these models. The Friedman test null hypothesis was rejected with a p value of 0.09564. From the CD

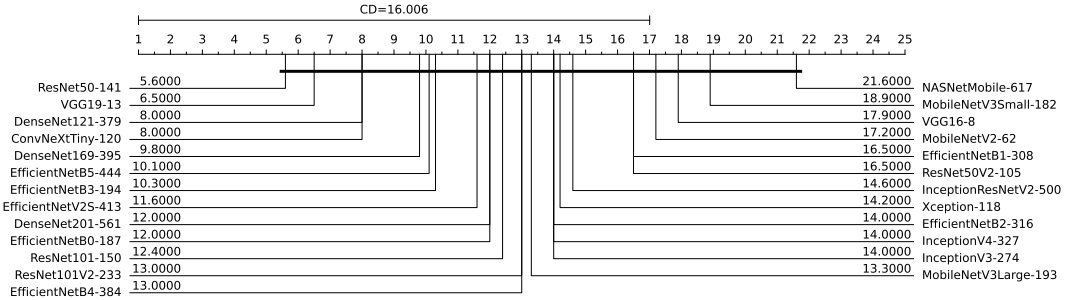


Figure 3.16: Accuracy critical difference diagram for the best performing configurations of the trained CNN models. The models are ordered by best to worst average ranking from left to right. The number beside a model's name represents the average rank of the model. CD is the critical difference for Nemenyi post-hoc test. Thick horizontal line connects the models that are not statistically significantly different.

diagram, we can see that ResNet50- 141 achieved the best average ranking followed by VGG19-13 and DenseNet121-379 respectively. Xception and Inception-based architectures had a similar ranking. NasNetMobile-617 ranked worst among all the models. The accuracy of the models varied from 81.3% to 84.42% and there is no statistically significant difference in terms of accuracy metric among most of the trained models. Overall, ResNet50-141 performed better in terms of various metrics (5 out of 11) as highlighted in Table 3.4. We kept the confusion matrix, ROC curve, and cross-validation fold-wise details of all the trained models in Appendix Section B.2 to make the chapter concise and readable.

Table 3.3 summarizes the complexities of the CNN models used in this study. The most lightweight model with the lowest number of parameters, FLOPs, and memory usage was MobileNetV3Small-182. InceptionResNetV2-500 has the highest number of parameters and memory usage and slowest inference time. Xception-118 was the fastest in terms of inference time. VGG19-13 required the highest number of FLOPs. ResNet50V2-105 required the least amount of time to train on average whereas, EfficientNetB5-444 was the slowest to train.

Table 3.5 shows the Grad-CAM visualizations of the models trained on the same training fold for two test images. From the table, it can be seen that different versions of EfficientNet focused more on the lesion part of the image compared to other models. The squeeze-and-excitation [67] channel attention used in EfficientNet can be the reason behind this behavior.

The experimental results showed that our proposed pre-training strategy utilizing dermoscopic dataset HAM10000 improved the performance of ImageNet pre-trained CNNs for recognizing clinical EM images. The results make it evident that CNNs have great potential to be used for Lyme disease pre-scanner application. Figure 3.17 shows a bubble chart reporting model accuracy vs FLOPs. The size of each bubble represents the

number of parameters of the model. This figure serves as a guideline for selecting models based on complexity and accuracy. It can be seen from the figure that EfficientNetB0-187 is a good choice with reasonable accuracy for resource-constrained mobile platforms. EfficientNetB0-187 also showed good results in Grad-CAM visualization. If resource constraint is not a problem, then RestNet50-141 can be used for the best accuracy.

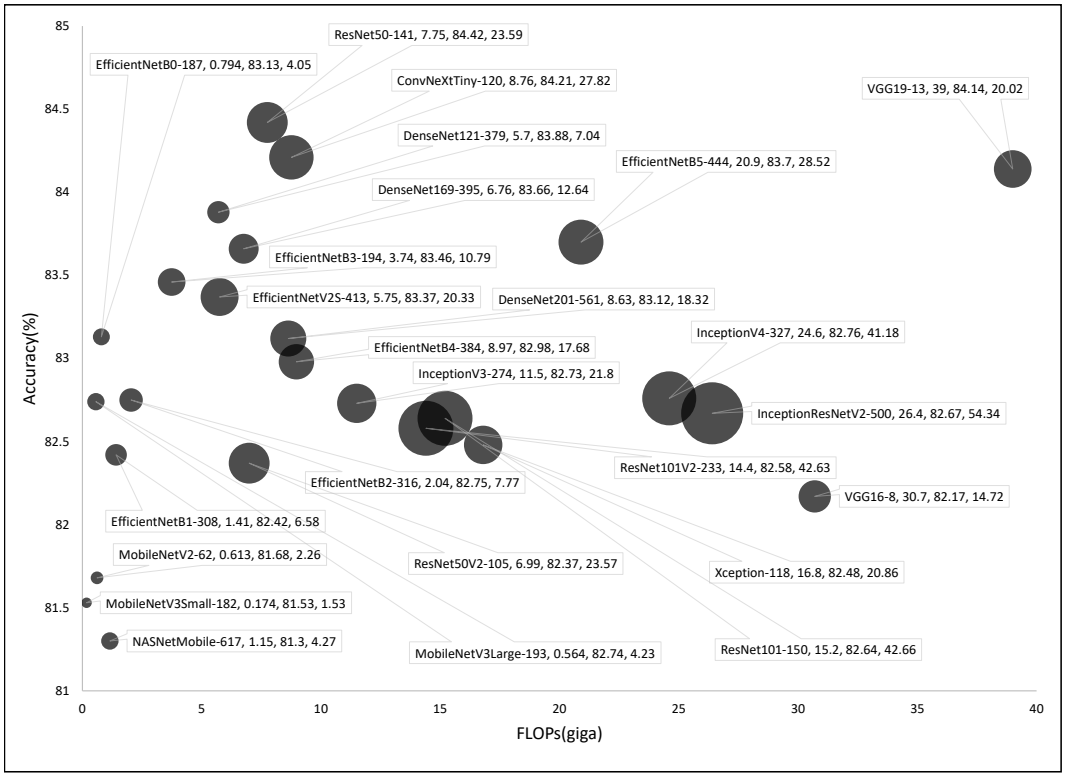


Figure 3.17: Bubble chart reporting model accuracy vs floating-point operations (FLOPs). The size of each bubble represents the number of model parameters measured in millions unit. Beside each model name the three values represent FLOPs, accuracy, and model parameters, respectively.

Even the lightweight EfficientNetB0-187 model showed good performance, and it can be directly deployed in mobile devices without requiring an internet connection for processing the lesion image in a remote server. It can help people living in remote areas without good internet facilities with an initial assessment of the probability of Lyme disease.

For this study, we utilized images from the internet alongside images collected from several hospitals in France. This approach was inspired by related studies on skin lesion analysis.

Although a portion of images in our dataset was collected from the internet the annotation of the dataset is reliable because we ignored the online labels, and all the images were reannotated by expert dermatologists and infectiologists.

We made all the trained models publicly available, which can be utilized by others for transfer learning and building pre-scanners for Lyme disease. The trained CNN models are available at the link stated in Appendix Section B.1.

3.4 Conclusion

In this chapter, a pre-training strategy for improving clinical skin lesion image classification performance of ImageNet pre-trained convolutional neural networks by utilizing additional pre-training with dermoscopic images was proposed. We applied the strategy to benchmark twenty-five well-known CNNs based on predictive performance, complexity, significance tests, and heatmap visualization using a novel Lyme disease dataset to find out the effectiveness of CNNs for Lyme disease diagnosis from EM images. We also provided guidelines for model selection. We found that even the lightweight models like EffiicentNetB0 performed well suggesting the application of CNNs for Lyme disease pre-scanner mobile applications which can help people with an initial assessment of the probability of Lyme disease and referring them to expert dermatologist for further diagnosis. Resource intensive models like ResNet50 can be effective for building computer applications to assist non-expert practitioners with identifying EM.

Key Points (Chapter 3)

- We proposed a pre-training strategy of fine-tuning some layers from the end of an ImageNet pre-trained CNN architecture using a dermoscopic dataset before training the model on a clinical skin lesion dataset.
- The proposed pre-training strategy seemed effective for increasing model performance based on experimentation using a novel Lyme disease dataset.
- We benchmarked several state-of-the-art CNN architectures on the novel Lyme dataset utilizing our pre-training strategy.
- Experimental results suggest that even lightweight CNNs can be effective for Lyme disease pre-scanner mobile applications.

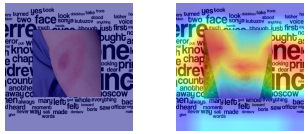
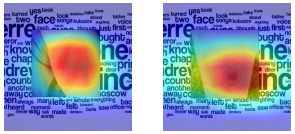
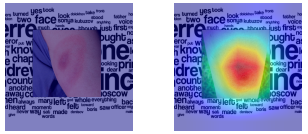
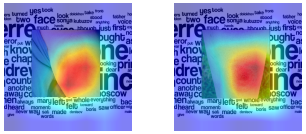
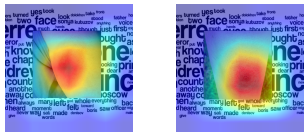
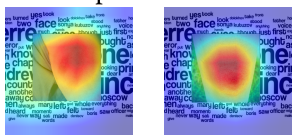
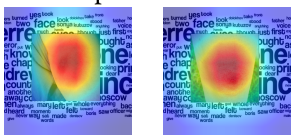
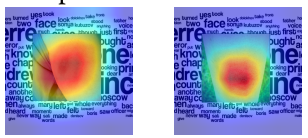
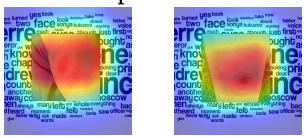
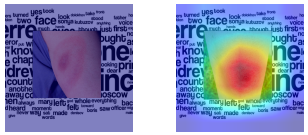
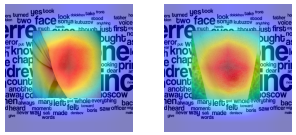
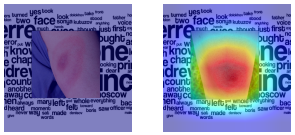
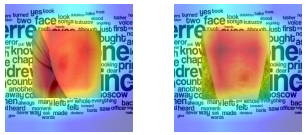
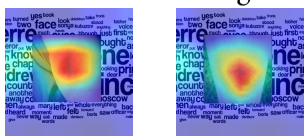
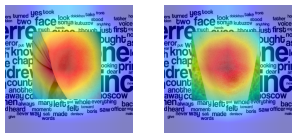
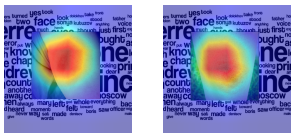
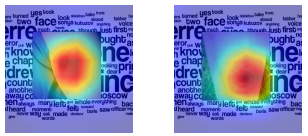
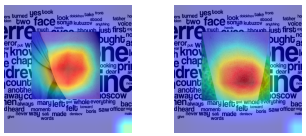
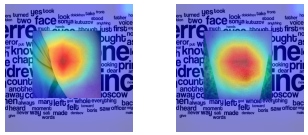
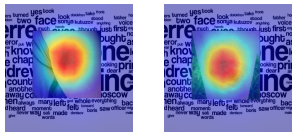
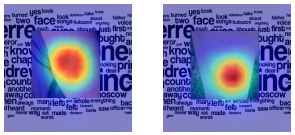
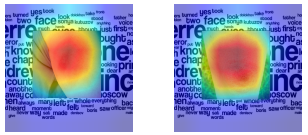
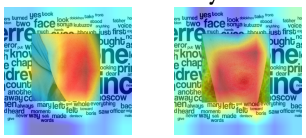
Table 3.3: Complexity metrics of trained CNN models. Bold indicates the best result for each of the metrics.

Model	**Parameters** (million)	**FLOPs** (giga)	**Average training time** (sec per epoch)	**GPU usage** (megabyte)	**Average inference time** (sec per image)
VGG16-8	14.72	30.7	111	565	0.0426
VGG19-13	20.02	39	164	565	0.0431
ResNet50-141	23.59	7.75	113	821	0.0484
ResNet101-150	42.66	15.2	123.33	821	0.0539
ResNet50V2-105	23.57	6.99	76	821	0.0464
ResNet101V2-233	42.63	14.4	152	821	0.0599
InceptionV3-274	21.8	11.5	133	821	0.054
InceptionV4-327	41.18	24.6	223.33	1333	0.0735
InceptionResNetV2-500	54.34	26.4	281.33	1333	0.0958
Xception-118	20.86	16.8	243.33	821	0.0392
DenseNet121-379	7.04	5.7	140.67	437	0.0673
DenseNet169-395	12.64	6.76	130	565	0.0686
DenseNet201-561	18.32	8.63	182.67	565	0.084
MobileNetV2-62	2.26	0.613	78	**341**	0.0429
MobileNetV3Small-182	**1.53**	**0.174**	81	**341**	0.0444
MobileNetV3Large-193	4.23	0.564	86.33	373	0.0444
NASNetMobile -617	4.27	1.15	152	373	0.0741
EfficientNetB0-187	4.05	0.794	87	373	0.0523
EfficientNetB1-308	6.58	1.41	158.33	437	0.0546
EfficientNetB2-316	7.77	2.04	210	437	0.0565
EfficientNetB3-194	10.79	3.74	143	565	0.0648
EfficientNetB4-384	17.68	8.97	431	565	0.0614
EfficientNetB5-444	28.52	20.9	771	821	0.0659
EfficientNetV2S-413	20.33	5.75	**50**	591	0.934
ConvNeXtTiny-120	27.82	8.76	95	847	0.0664

Table 3.4: Five-fold cross-validation performance metrics for the best performing configurations of the trained CNN models. Within each cell, the value after (±) symbol represents the standard deviation across five folds. Bold indicates the best result for each of the metrics.

	Metric										
Model	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
VGG16-8	82.17 ± 1.23	85.77 ± 3.58	78.31 ± 4.36	81.12 ± 2.62	83.88 ± 3.02	0.6453 ± 0.0253	0.6422 ± 0.0249	4.0983 ± 0.7329	0.1802 ± 0.0388	0.8328 ± 0.0116	0.9011 ± 0.0079
VGG19-13	84.14 ± 1.62	85.29 ± 1.69	82.9 ± 2.63	**84.32** ± 1.97	84.0 ± 1.67	0.6826 ± 0.0323	0.6823 ± 0.0326	5.0924 ± 0.6884	0.1777 ± 0.0214	0.8479 ± 0.0146	0.913 ± 0.0074
ResNet50-141	**84.42** ± 1.36	87.93 ± 1.47	80.65 ± 3.59	83.1 ± 2.49	86.19 ± 1.27	**0.6893** ± 0.0263	**0.6874** ± 0.0277	4.703 ± 0.8624	0.1493 ± 0.0155	**0.8541** ± 0.0106	**0.9189** ± 0.0115
ResNet50V2-105	82.37 ± 2.15	85.53 ± 3.35	78.96 ± 6.13	81.66 ± 3.83	83.72 ± 2.63	0.6493 ± 0.0411	0.6461 ± 0.0439	4.3618 ± 1.0495	0.1819 ± 0.0349	0.8343 ± 0.017	0.9013 ± 0.0133
ResNet101V2-233	82.58 ± 2.21	81.9 ± 4.78	**83.32** ± 3.71	84.17 ± 2.55	81.31 ± 3.7	0.6535 ± 0.0429	0.6515 ± 0.0439	**5.104** ± 0.9811	0.2163 ± 0.0541	0.8292 ± 0.0254	0.9118 ± 0.0149
InceptionV3-274	82.73 ± 2.08	86.57 ± 2.42	78.6 ± 2.8	81.33 ± 2.12	84.52 ± 2.52	0.6551 ± 0.0419	0.6533 ± 0.0419	4.1259 ± 0.639	0.1714 ± 0.0328	0.8385 ± 0.0195	0.9052 ± 0.0185
InceptionV4-327	82.76 ± 1.78	85.7 ± 3.96	79.58 ± 2.87	81.92 ± 1.8	84.02 ± 3.41	0.6561 ± 0.0358	0.6541 ± 0.0353	4.2716 ± 0.5734	0.179 ± 0.0465	0.837 ± 0.0197	0.9092 ± 0.019
Inception ResNetV2-500	82.67 ±2.06	83.54 ±3.88	81.74 ±3.16	83.17 ±2.16	82.37 ±3.32	0.6541 ±0.0406	0.653 ±0.041	4.6886 ±0.7264	0.2009 ±0.0456	0.8329 ±0.0218	0.9011 ±0.0133
Xception-118	82.48 ±2.45	83.16 ±5.1	81.75 ±2.76	83.08 ±1.94	82.16 ±4.4	0.6507 ±0.0487	0.6492 ±0.0484	4.6434 ±0.6571	0.2054 ±0.0609	0.8304 ±0.0276	0.9081 ±0.0148
DenseNet121-379	83.88 ±0.92	85.85 ±1.76	81.75 ±0.95	83.49 ±0.69	84.35 ±1.57	0.6773 ±0.0186	0.6768 ±0.0184	4.7169 ±0.254	0.173 ±0.0211	0.8465 ±0.01	0.9158 ±0.0097
DenseNet169-395	83.66 ±1.25	**88.6** ±3.59	78.35 ±2.75	81.54 ±1.53	**86.68** ±3.33	0.6758 ±0.0265	0.6717 ±0.0249	4.1454 ±0.4187	**0.1446** ±0.0414	0.8486 ±0.0138	0.9123 ±0.0129
DenseNet201-561	83.12 ±1.11	85.61 ±1.81	80.45 ±3.92	82.61 ±2.7	83.93 ±1.13	0.663 ±0.0221	0.6615 ±0.0228	4.5729 ±0.9885	0.1783 ±0.0153	0.8403 ±0.0073	0.9125 ±0.0083
MobileNetV2-62	81.68 ±1.99	81.94 ±3.49	81.39 ±1.26	82.55 ±1.21	80.85 ±3.09	0.6337 ±0.0394	0.6332 ±0.0395	4.4256 ±0.3596	0.222 ±0.0441	0.8222 ±0.0218	0.8933 ±0.0135
MobileNetV3 Small-182	81.53 ±1.98	84.93 ±3.29	77.87 ±3.89	80.6 ±2.55	82.91 ±2.85	0.6315 ±0.0398	0.6294 ±0.04	3.9496 ±0.6356	0.1933 ±0.0386	0.8265 ±0.0186	0.896 ±0.013
MobileNetV3 Large-193	82.74 ±2.17	83.69 ±0.43	81.71 ±4.6	83.26 ±3.39	82.3 ±0.89	0.6548 ±0.0437	0.6542 ±0.0442	4.8573 ±1.1585	0.2002 ±0.0117	0.8344 ±0.017	0.9034 ±0.0094
NASNet Mobile-617	81.3 ±1.45	83.2 ±1.66	79.25 ±3.98	81.29 ±2.65	81.48 ±1.07	0.6261 ±0.0287	0.6251 ±0.0297	4.1452 ±0.7283	0.2117 ±0.0156	0.8219 ±0.0108	0.8897 ±0.0152
EfficientNet B0-187	83.13 ±1.2	85.21 ±3.91	80.89 ±2.95	82.83 ±1.75	83.79 ±3.19	0.6636 ±0.0244	0.6618 ±0.0237	4.5522 ±0.6116	0.1817 ±0.0427	0.8392 ±0.0147	0.9094 ±0.0129
EfficientNet B1-308	82.42 ±1.04	85.85 ±2.14	78.71 ±3.75	81.37 ±2.34	83.9 ±1.59	0.6492 ±0.0202	0.647 ±0.0214	4.1494 ±0.6707	0.179 ±0.0209	0.835 ±0.0074	0.9088 ±0.0134
EfficientNet B2-316	82.75 ±1.4	84.95 ±3.41	80.39 ±3.02	82.39 ±1.91	83.4 ±2.69	0.6556 ±0.0276	0.6542 ±0.0279	4.4211 ±0.6202	0.1865 ±0.0379	0.8359 ±0.0158	0.9075 ±0.0082
EfficientNet B3-194	83.46 ±0.87	85.15 ±4.28	81.64 ±2.9	83.4 ±1.6	83.9 ±3.14	0.6704 ±0.0157	0.6685 ±0.0167	4.7361 ±0.6283	0.1803 ±0.0443	0.8416 ±0.0144	0.9163 ±0.0074
EfficientNet B4-384	82.98 ±1.31	87.55 ±2.2	78.06 ±3.76	81.2 ±2.33	85.46 ±1.76	0.6613 ±0.0249	0.6581 ±0.0268	4.0946 ±0.6159	0.1589 ±0.0233	0.842 ±0.0107	0.9138 ±0.0074
EfficientNet B5-444	83.7 ±1.21	86.85 ±2.89	80.32 ±3.73	82.71 ±2.39	85.17 ±2.38	0.6752 ±0.024	0.6729 ±0.0245	4.5562 ±0.7645	0.1629 ±0.0303	0.8466 ±0.0108	0.9138 ±0.0161
EfficientNet V2S-413	83.37 ±2.26	84.53 ±3.11	82.13 ±4.05	83.68 ±3.02	83.26 ±2.76	0.6679 ±0.0451	0.6669 ±0.0453	4.9859 ±1.1671	0.1884 ±0.0365	0.8404 ±0.0212	0.9144 ±0.0145
ConvNeXt Tiny-120	84.21 ±2.07	86.72 ±3.5	81.51 ±2.0	83.45 ±1.6	85.23 ±3.31	0.6845 ±0.0418	0.6834 ±0.0412	4.7452 ±0.5401	0.1631 ±0.0436	0.8502 ±0.0215	0.9183 ±0.015

Table 3.5: Grad-CAM heatmap visualization of the trained models.

VGG16-8	VGG19-13	ResNet50-141	ResNet101-150	ResNet50V2-105
				
ResNet101V2-233	InceptionV3-274	InceptionV4-327	InceptionResNetV2-500	Xception-118
				
DenseNet121-379	DenseNet169-395	DenseNet201-561	MobileNetV2-62	MobileNetV3Small-182
				
MobileNetV3Large-193	NASNetMobile-617	EfficientNetB0-187	EfficientNetB1-308	EfficientNetB2-316
				
EfficientNetB3-194	EfficientNetB4-384	EfficientNetB5-444	EfficientNetV2S-413	ConvNeXtTiny-120
				

4 Expert Opinion Elicitation for Assisting Lesion Image Classifier With Patient Data

This chapter addresses research question 2, presents our questionnaire based expert opinion elicitation method for calculating disease probability from patient data and an approach for combining independent probability estimates from multiple modalities. Contents from this chapter have been used in the following article:

- S. I. Hossain, J. de Goër de Herve, D. Abrial, R. Emilion, I. Lebertb, Y. Frendo, D. Martineau, O. Lesens, and E. Mephu Nguifo. "Expert Opinion Elicitation for Assisting Deep Learning based Lyme Disease Classifier with Patient Data", 2022. arXiv: 2208.14384

Chapter Contents

4.1 Introduction

When the dermatologists rechecked the annotations of the Lyme image dataset misclassified by most of the convolutional neural networks (CNNs) (discussed in Chapter 3) they found mistakes in some of the initial annotations. This suggests that some images are too confusing to classify even for the experts without additional context from patient data.

Expert opinion elicitation can be effective when high quality data is difficult to collect [175]. Point estimates (such as medians or means), intervals of uncertainty (such as confidence intervals or quartiles), or probability distributions can all be included in the measurements elicited [17]. Expert opinion elicitation and aggregation processes can be classified into two categories: behavioral and mathematical approaches [17, 25]. The behavioral approach tries to produce group consensus among experts whereas, the mathematical approach combines subjective probabilities from experts using mathematical methods (some form of averaging) [17].

Expert elicitation proved effective for medical diagnosis and decision making. For example, Van Der Gaag et al. [166] created a probabilistic network to describe the oesophageal cancer presentation characteristics and the pathophysiological mechanisms of invasion and metastasis by eliciting opinions from two experts. Saegerman et al. [136] elicited opinions from eleven European experts to rank the drivers of the emergence of bovine besnoitiosis a chronic disease in cattle. Wilson et al. [175] elicited opinions from sixteen experts on the disease progression probability in patients with untreated melanoma. The article by Cadham et al. [17] contains a detailed review of the application of expert elicitation in health research computational modeling studies.

In our study, for the first time, we elicited opinions from fifteen expert dermatologists to create a model for calculating erythema migrans (EM) probability from patient data as an early symptom of Lyme disease. First, with the help of the experts, a questionnaire was prepared based on questions that the doctors ask during EM diagnosis. The traditional expert elicitation process of collecting probability estimates for cases based on the questionnaire is time consuming and it is difficult for doctors to provide probability estimates for cases or distribution parameters. Therefore, we opted for a more relaxed approach of relative weight assignment to different answers to the questions and converted the doctor's evaluations to EM probabilities utilizing Gaussian mixture model (GMM) based density estimation (described in Section 2.1.4). To validate the elicited probability model and explain its behavior to the experts we utilized formal concept analysis (described in Section 2.1.3) and decision tree (described in Section 2.1.2). The elicited patient data based EM probability model will be useful for assisting image-based EM classifiers with additional context from patient data. We also proposed an algorithm for combining the EM probability score from a deep learning image classifier with the elicited probability score from patient data. The proposed algorithm ensures veto power for the patient data. The elicited probability score and the proposed algorithm can be utilized to make image based deep learning Lyme disease pre-scanners robust and the techniques will be useful for questionnaire based opinion elicitation of other diseases.

The rest of the chapter is structured as follows: Section 4.2 describes the expert elicitation process and elicitation result; Section 4.3 contains the strategy for combining probabilities; finally, Section 4.4 provides concluding remarks.

4.2 Elicitation Method

The details of our expert elicitation process like expert recruitment, questionnaire preparation, experts' opinion collection, elicitation methods, result, and analysis are presented in the following subsections.

4.2.1 Expert Selection

The recruited experts are hospital practitioners who are infectious disease specialists or dermatologists working in reference centers for tick-borne diseases of France - Centres de Référence des Maladies Vectorielles liées aux Tiques (CRMVT) [27]. At a CRMVT steering committee meeting held in June 2021 with participants from all the reference centers, Professor Olivier Lesens (Infectious and Tropical Diseases Department, CRIOA, CHU Clermont-Ferrand, France) explained the importance of expert elicitation for calculating EM probability based on patient data and requested the interested experts to participate in the elicitation process. Fifteen experts agreed to participate[1]. Table 4.1 lists the reference centers and the corresponding number of experts participating in the elicitation process.

4.2.2 Questionnaire and Experts' Evaluation

For the EM probability elicitation, a questionnaire was prepared based on questions about the context of onset and progression of the skin lesion that a physician usually asks when diagnosing EM. The questionnaire is based on a previous study concerning the collection of EM related data from rural areas of France [85]. The questionnaire was finalized through several meetings held in April 2020 among the doctors of CRMVT in Clermont-Ferrand and experts in tick ecology from the French national research institute for agriculture, food and the environment - Institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAE) [71]. Experts who volunteered to participate in the elicitation process at the meeting in June 2021 agreed that there were many possible cases from the combination of the questions and answers, and it was time consuming and difficult for them to provide probability estimates for all those different cases. Therefore, experts agreed to independently assign relative weights to different possible answers associated with each question. The assigned weight values are in the range -1 to $+3$ (a higher value represents a higher contribution of the answer towards the possibility of the EM). The experts were contacted via email with detailed instructions to provide their weight attributions independently. Table 4.2 lists the questions, answers, and weight attribution from the doctors. After receiving the weight attributions from all the experts, they participated in a meeting in November 2021 and agreed that fever, fatigue, faintness, and headache should contribute equally if one or more of these answers

[1] The experts did not receive any monetary benefits for participating in the elicitation process.

Table 4.1: Experts recruited for erythema migrans probability elicitation.

Center	Number of Participating Experts
CRMVT du Grand Ouest Hôpital Pontchaillou Centre Urgences-Réanimations 2 rue Henri Le Guilloux 35033 Rennes Cedex 09	2
CRMVT Ile-de-France et Hauts-de-France Hôpital de Villeneuve-Saint-Georges Secrétariat du centre Lyme 40 allée de la Source 94195 Villeneuve-Saint-Georges Cedex	2
CRMVT de Strasbourg Nouvel Hôpital Civil - Hôpitaux Universitaires de Strasbourg (HUS) 1 place de l'Hôpital BP461 67091 STRASBOURG Cedex	1
CRMVT de Nancy Centre Hospitalier Régional et Universitaire (CHRU) de Nancy Rue du Morvan 54500 Vandœuvre-lès-Nancy	1
CRMVT de Clermont-Ferrand Service des Maladies Infectieuses et Tropicales CHU Gabriel Montpied 58, Rue Montalembert 63003 Clermont-Ferrand Cedex 1	8
CRMVT de Saint-Etienne Service des Maladies Infectieuses et Tropicales Hôpital Nord Avenue Albert Raimond 42270 Saint-Priest-en-Jarez	1

were present and the contribution should be the average of these four answers. Therefore, the four answers were replaced with one, and the possible cases reduced to 1,536 from 12,288 cases. This modification is shown in Table 4.3.

4.2.3 Opinion Elicitation

Following are some notations used in the rest of the manuscript:

- Set of doctors, $D = \{d_e | e = 1, \ldots, n_e\}$.
- Set of questions, $Q = \{q_i | i = 1, \ldots, n_i\}$.
- Set of possible cases, $C = \{c_l | l = 1, \ldots, n_l\}$.

Table 4.2: Questionnaire and doctors' weight attribution for erythema migrans. The assigned weight values are in the range -1 to $+3$ (a higher value represents a higher contribution of the answer towards the possibility of the erythema migrans). d_1 to d_{15} represents the doctors.

Question	**Answer**	**Weight Assigned by Doctors (Doctor's Evaluation)**															
		d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	Average
Other symptoms observed alongside the skin lesion	No	0	0	3	0	0	1	2	1	2	1	2	1	1	2	3	1.27
	Fever	-1	0	-1	1	1	1	1	1	1	1	1	1	1	1	0	0.6
	Fatigue	0	1	1	2	0	0	1	1	1	1	1	0	1	0	0	0.67
	Faintness	0	0	1	0	0	0	0	0	1	0	1	-1	1	0	0	0.2
	Joint pain	0	0	-1	0	2	0	1	1	0	2	1	1	1	0	0	0.53
	Headache	1	1	-1	2	2	1	1	1	1	1	1	1	1	0	0	0.87
	Itching	-1	-1	-1	-1	1	-1	0	0	0	1	-0.5	-1	-1	1	0	-0.3
What was the maximum size of the red rash	<1 cm	-1	-1	0	-1	-1	-1	-1	0	0	-1	-1	-1	-1	1	-1	-0.67
	1 to 5 cm	1	1	1	0	1	0	1	1	2	1	1	1	1	2	1	1
	>5 cm	3	2	2	2	3	2	3	2	1	3	2	2	3	3	3	2.4
	I do not know	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0
Is the size of the red rash increasing or has it gradually increased	Yes	3	1	3	3	3	3	3	3	2	3	3	3	3	3	3	2.8
	No	0	-1	-1	-1	-1	-1	-1	0	-1	0	-1	-1	-1	1	-1	-0.67
	I do not know	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.07
Have you seen a tick bite on this red rash in the past 30 days	Yes	3	2	3	2	3	1	3	3	2	3	2	3	1	3	3	2.47
	No	0	0	0	0	0	1	0	1	0	0	-0.5	-1	0	1	0	0.1
Frequency of tick bites in the last 30 days before the appearance of the red rash	Never	-1	-1	0	0	-1	0	0	0	0	0	-1	-1	0	-1	0	-0.4
	1 time	0	0	2	1	1	1	1	1	2	1	1	1	1	2	1	1.07
	2 to 5 times	1	1	3	1	1	1	1	2	2	1	2	1	1	3	1	1.47
	>5 times	2	2	1	2	2	1	1	2	2	2	3	1	1	3	2	1.8
Outdoor activities in the last 30 days before the onset of the red rash	Yes	1	1	2	2	1	1	2	2	2	2	2	2	1	3	2	1.73
	No	-1	-1	-1	-1	-1	0	-1	1	-1	-1	-1	-1	0	-1	0	-0.67

- Total number of answers corresponding to q_i question $= n_{q_i}$.
  For example, $n_{q_2} = 4$ because question q_2 has four possible answers (refer to Table 4.3).

- j^{th} answer corresponding to q_i question,

$$a_{j,q_i} = \begin{cases} 1, & \text{if the answer is true} \\ 0, & \text{otherwise} \end{cases}, \text{where } j = 1, \ldots, n_{q_i}$$

Table 4.3: Weight modified questionnaire and doctors' weight attribution for erythema migrans. The assigned weight values are in the range -1 to $+3$ (a higher value represents a higher contribution of the answer towards the possibility of the erythema migrans). d_1 to d_{15} represents the doctors.

Question	Answer	Weight Assigned by Doctors (Doctor's Evaluation)															
		d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	Average
Other symptoms observed alongside the skin lesion (q_1)	No (a_{1,q_1})	0	0	3	0	0	1	2	1	2	1	2	1	1	2	3	1.27
	Fever/ Fatigue/ Faintness/ Headache (a_{2,q_1})	-0.25	0.25	0	0.75	0.75	0.25	0.75	0.75	0.75	1	1	0.25	1	0.25	0	0.5
	Joint pain (a_{3,q_1})	1	1	-1	2	2	1	1	1	1	1	1	1	1	0	0	0.87
	Itching (a_{4,q_1})	-1	-1	-1	-1	1	-1	0	0	0	1	-0.5	-1	-1	1	0	-0.3
What was the maximum size of the red rash (q_2)	<1 cm (a_{1,q_2})	-1	-1	0	-1	-1	-1	-1	0	0	-1	-1	-1	-1	1	-1	-0.67
	1 to 5 cm (a_{2,q_2})	1	1	1	0	1	0	1	1	2	1	1	1	1	2	1	1
	>5 cm (a_{3,q_2})	3	2	2	2	3	2	3	2	1	3	2	2	3	3	3	2.4
	I do not know (a_{4,q_2})	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0
Is the size of the red rash increasing or has it gradually increased (q_3)	Yes (a_{1,q_3})	3	1	3	3	3	3	3	3	2	3	3	3	3	3	3	2.8
	No (a_{2,q_3})	0	-1	-1	-1	-1	-1	-1	0	-1	0	-1	-1	-1	1	-1	-0.67
	I do not know (a_{3,q_3})	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.07
Have you seen a tick bite on this red rash in the past 30 days (q_4)	Yes (a_{1,q_4})	3	2	3	2	3	1	3	3	2	3	2	3	1	3	3	2.47
	No (a_{2,q_4})	0	0	0	0	0	1	0	1	0	0	-0.5	-1	0	1	0	0.1
Frequency of tick bites in the last 30 days before the appearance of the red rash (q_5)	Never (a_{1,q_5})	-1	-1	0	0	-1	0	0	0	0	0	-1	-1	0	-1	0	-0.4
	1 time (a_{2,q_5})	0	0	2	1	1	1	1	1	2	1	1	1	1	2	1	1.07
	2 to 5 times (a_{3,q_5})	1	1	3	1	1	1	1	2	2	1	2	1	1	3	1	1.47
	>5 times (a_{4,q_5})	2	2	1	2	2	1	1	2	2	2	3	1	1	3	2	1.8
Outdoor activities in the last 30 days before the onset of the red rash (q_6)	Yes (a_{1,q_6})	1	1	2	2	1	1	2	2	2	2	2	2	1	3	2	1.73
	No (a_{2,q_6})	-1	-1	-1	-1	-1	0	-1	1	-1	-1	-1	-1	0	-1	0	-0.67

- Weight assigned by doctor d_e to a_{j,q_i} answer $= w_{d_e,a_{j,q_i}}$.
  For example, $w_{d_1,a_{3,q_2}} = 3$ because the third answer to second question has a weight of 3 assigned by the first doctor (refer to Table 4.3).

Our opinion elicitation task for Lyme disease involved fifteen experts, the prepared questionnaire contains six questions and the possible cases from the combination of questions and answers is 1,536. So, for the Lyme disease task $n_e = 15$, $n_i = 6$, and $n_l = 1536$. First, we summarized each of the n_l possible cases as a weight sum s_{c_l} as shown in Equation (4.1).

$$s_{c_l} = \sum_{i=1}^{|Q|} \sum_{j=1}^{n_{q_i}} a_{j,q_i} \times \left(\frac{1}{|D|} \sum_{d=1}^{|D|} w_{d_e,a_{j,q_i}} \right) \tag{4.1}$$

The set of case weight sum is defined as $S = \{s_{c_l} | l = 1, \ldots, n_l\}$. Then, we normalized each case weight sum with min-max normalization as shown in Equation (4.2).

$$\tilde{s}_{c_l} = \frac{s_{c_l} - \min(S)}{\max(S) - \min(S)} \tag{4.2}$$

The set of min-max normalized case weight sum is defined as $\tilde{S} = \{\tilde{s}_{c_l} | l = 1, \ldots, n_l\}$. We proposed three approaches to the experts to convert the normalized case weight sum to a probability score for EM. The following subsections explain the three approaches.

4.2.3.1 Cumulative Probability from Density Estimate Based on GMM

We modeled our normalized weight sum data density using a GMM with two components. The number of components was selected based on the intuition that there are two sub populations within the data: one is the ill sub population and the other one is not ill sub population. The number of components was also supported by AIC and BIC values. Table 4.4 lists the selected parameters for the GMM. The blue curve in Figure 4.1 shows

Table 4.4: Parameters of Gaussian mixture model used to model the density of min-max normalized weight sum of erythema migrans cases. $\varnothing$, μ ,and σ represent mixture weight, mean and standard deviation respectively.

Parameter Name	**Value**
Components	2
$\varnothing_1$	0.364801
$\varnothing_2$	0.635199
μ_1	0.359548
μ_2	0.572878
σ_1	0.128782
σ_2	0.156241

the estimated density function using GMM. We defined the cumulative probability [20] of a normalized case weight sum from the GMM density estimate as the probability of EM as shown in Equation (4.3).

$$\hat{F}_{GMM}(x) = \int_{-\infty}^{x} \left(\sum_{m=1}^{2} \varnothing_m \mathcal{N}(x | \mu_m, \sigma_m) \right) dx \tag{4.3}$$

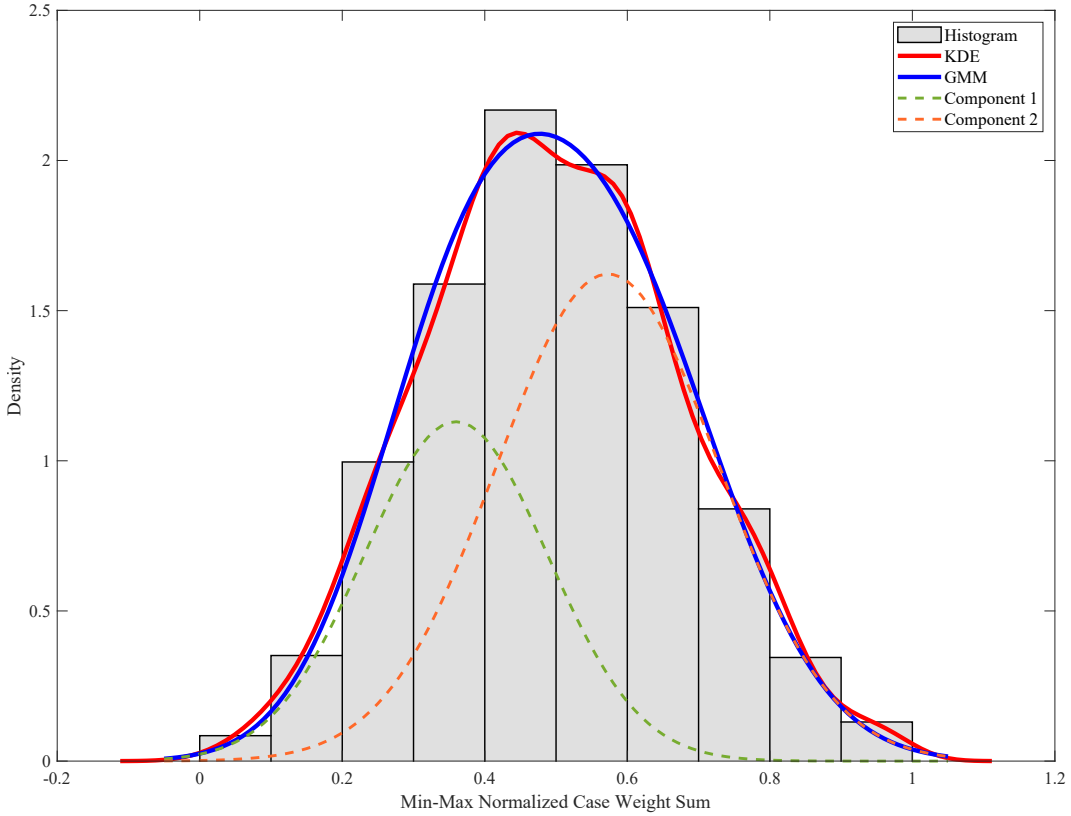


Figure 4.1: Proposed approaches for expert opinion elicitation. GMM and KDE stand for Gaussian mixture model and kernel density estimation respectively.

4.2.3.2 Posterior Probability of a Case Belonging to the Ill Subpopulation of GMM

The first and second components of our GMM are shown in Figure 4.1 with green and orange dotted lines respectively. If we assume that the second component represents the ill subpopulation then the posterior probability of a normalized case weight sum belonging to the second component [130] can be defined as the EM probability as shown in Equation (4.4).

$$p(\kappa_2|x) = \frac{\varnothing_2 \mathcal{N}(x|\mu_2, \sigma_2)}{\sum_{m=1}^{2} \varnothing_m \mathcal{N}(x|\mu_m, \sigma_m)} \tag{4.4}$$

4.2.3.3 Cumulative Probability from Density Estimate Based on Kernel Density Estimation

We used a Gaussian kernel with bandwidth, $h = 0.03676$ on our $n_l = 1,536$ data points for the probability density estimation of the normalized weight sum variable as shown in Equation (4.5).

$$\hat{f}_{KDE}(x) = \frac{1}{n_l \times h} \sum_{l=1}^{n_l} \frac{1}{2\pi} e^{-0.5\left(\frac{x-\tilde{s}_{c_l}}{h}\right)^2} \tag{4.5}$$

The red curve in Figure 4.1 shows the estimated density function. We defined the cumulative probability of a normalized case weight sum as the probability of having EM as shown in Equation (4.6).

$$\hat{F}_{KDE}(x) = \int_{-\infty}^{x} \hat{f}_{KDE}(x)dx \tag{4.6}$$

4.2.3.4 Elicitation Result and Analysis

We calculated EM probability score for all possible cases using the three approaches described in Section 4.2.3.1, 4.2.3.2, and 4.2.3.3 and presented the results with explanations to the experts in a meeting held in May 2022. Figure 4.2 shows the EM probability plot for all the cases using the three approaches. In the figure blue and red lines represent the probability scores based on density estimates from the Gaussian mixture model (approach 1) and kernel density estimate (approach 2) respectively. The orange line represents probability scores based on the posterior probability of a case belonging to the second component i.e. the ill subpopulation of the Gaussian mixture model (approach 3). Results obtained from approach 1 and approach 2 are close because both of them are based on density estimates whereas, probability scores obtained from approach 3 are always higher than the other two approaches. Based on the results and explanations the experts came to a consensus on the use of approach 1 (described in Section 4.2.3.1) mainly because the density estimate in approach 1 is smoother compared to approach 2 (described in Section 4.2.3.3).

To validate elicited model and explain its behavior to the experts first we used decision trees. For building the decision tree, we divided calculated EM probability scores into three categories: LOW (scores in the range $[0, 0.33)$), MEDIUM (scores in the range $[0.33, 0.68)$), and HIGH (scores in the range $[0.68, 1]$) Figure 4.3 shows a pruned version of the decision tree for approach 1. In the figure, each node shows the majority category along with the percentage and number of cases belonging to each category. From the tree, we can see that the model assigns HIGH EM probability to cases whenever the first answer, "*yes*" to the third question "*Is the size of the spot increasing or has it gradually*

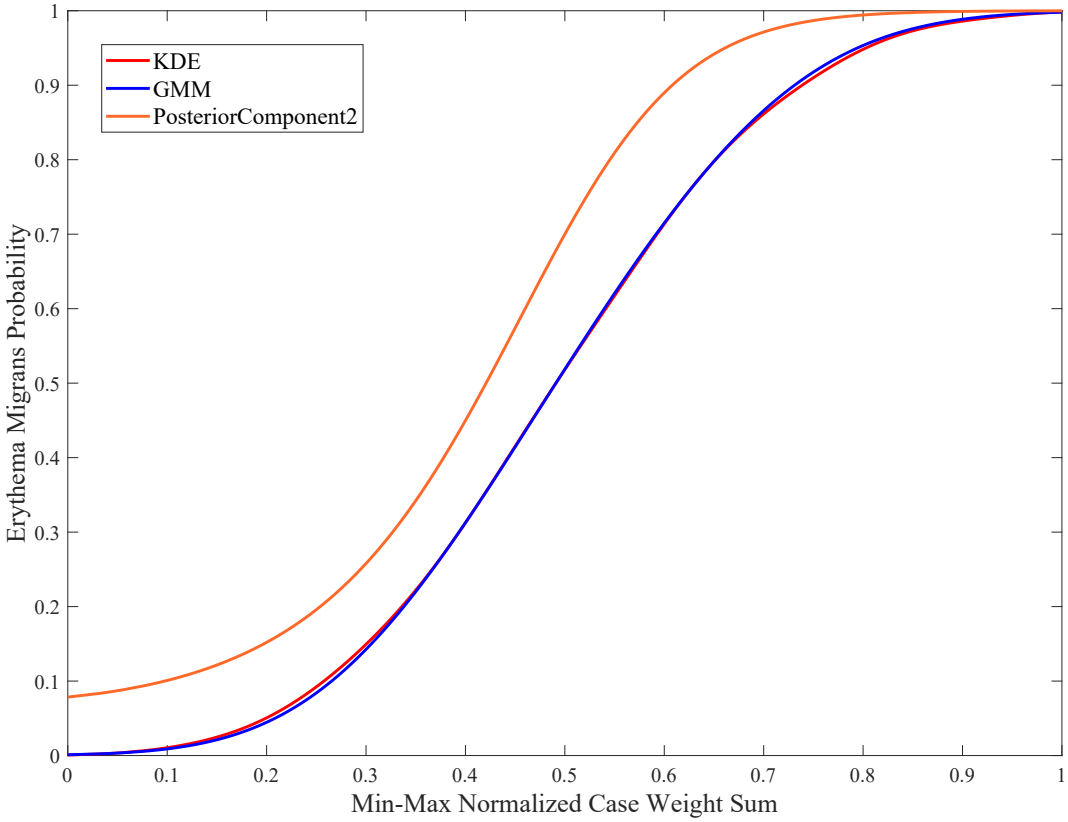


Figure 4.2: Elicited erythema migrans probability plot. Blue and red lines represent the probability scores based on density estimates from Gaussian mixture model and kernel density estimate respectively. Orange line represents probability scores based on the posterior probability of a case belonging to the second component i.e. the ill subpopulation of the Gaussian mixture model.

increased", a_{1,q_3} is true. This behavior supports the doctors' opinion because the first answer to the third question has the highest weight given by most of the doctors.

To further explain the behavior of the model we utilized formal concept analysis (FCA) to find out questions and answers important for different probability groups. Figure 4.4 shows a simplified FCA lattice view for the 162 cases belonging to the lowest probability score group in the range $[0, 0.1)$ obtained from approach 1. In the figure, the top box of a node represents an attribute (answer) or a number of attributes, which are connected by lines, and the bottom box represents how many objects (cases) contain the corresponding attribute shown in the top box. In Figure 4.4, we start with 162 cases in the root node. At the first level, the number inside the bottom box of a node represents how many cases out of 162 cases contain the corresponding answer shown in the top box. For example, the "*no*" answer to the question "*Outdoor activities in the last 30 days before the onset of the red spot*", a_{2,q_6} is present in 145 cases. At the second level, each node represents how many

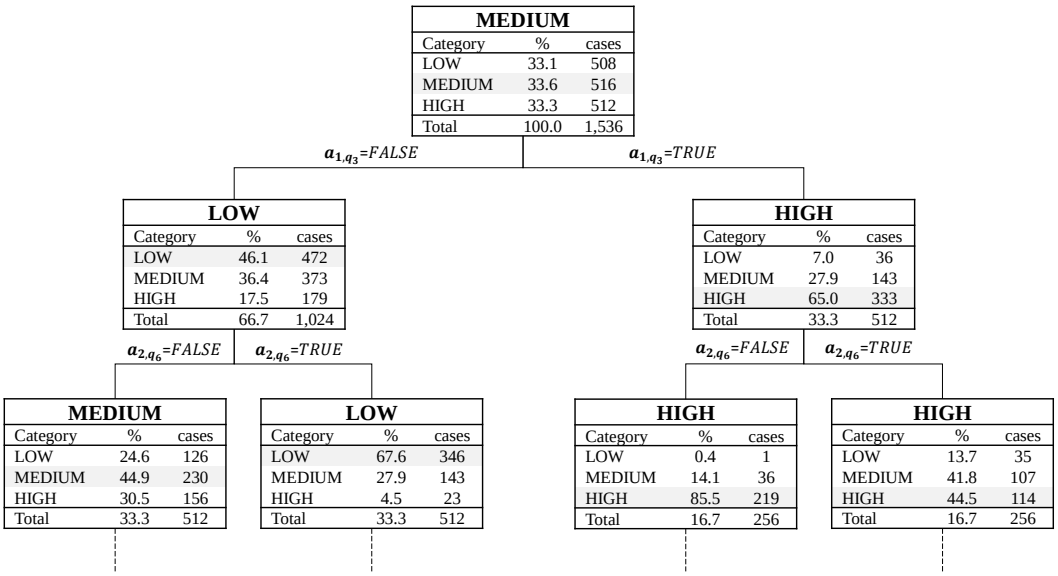


Figure 4.3: Pruned decision tree explaining elicited erythema migrans probability model behavior. Each node shows the majority category along with percentage and number of cases belonging to each category. Refer to Table 4.3 for details about the questions and answers. The full tree is available at the link stated in Appendix Section C.1.

cases contain two answers connected by a line. For example, a_{2,q_4} and a_{2,q_6} are jointly true in 128 cases. The rest of the FCA lattice is organized similarly. We can see from the figure that the answers common to most of these cases are the ones having lowest assigned weights or the opposites of the answers having highest assigned weights by most of the doctors.

The elicited EM probability scores for all possible cases, detailed decision tree, and FCA context files for different probability score groups are available at the link stated in Appendix Section C.1.

4.3 Combining Probabilities from Image and Patient Data

Our experiments showed that some images are too confusing to classify even for experts. Based on this evidence, experts suggest that EM probability obtained from the image data should not be prioritized and probability from patient data should have veto power over image data. If the EM probability obtained from image data and patient data are p_{image} and p_{data} respectively then the combined probability, $p_{combined}$ using the geometric mean as $\sqrt{p_{image} \times p_{data}}$ ensures veto power both for image and patient data as shown in Figure 4.5a. But according to the experts' suggestion, we want to keep the veto power only

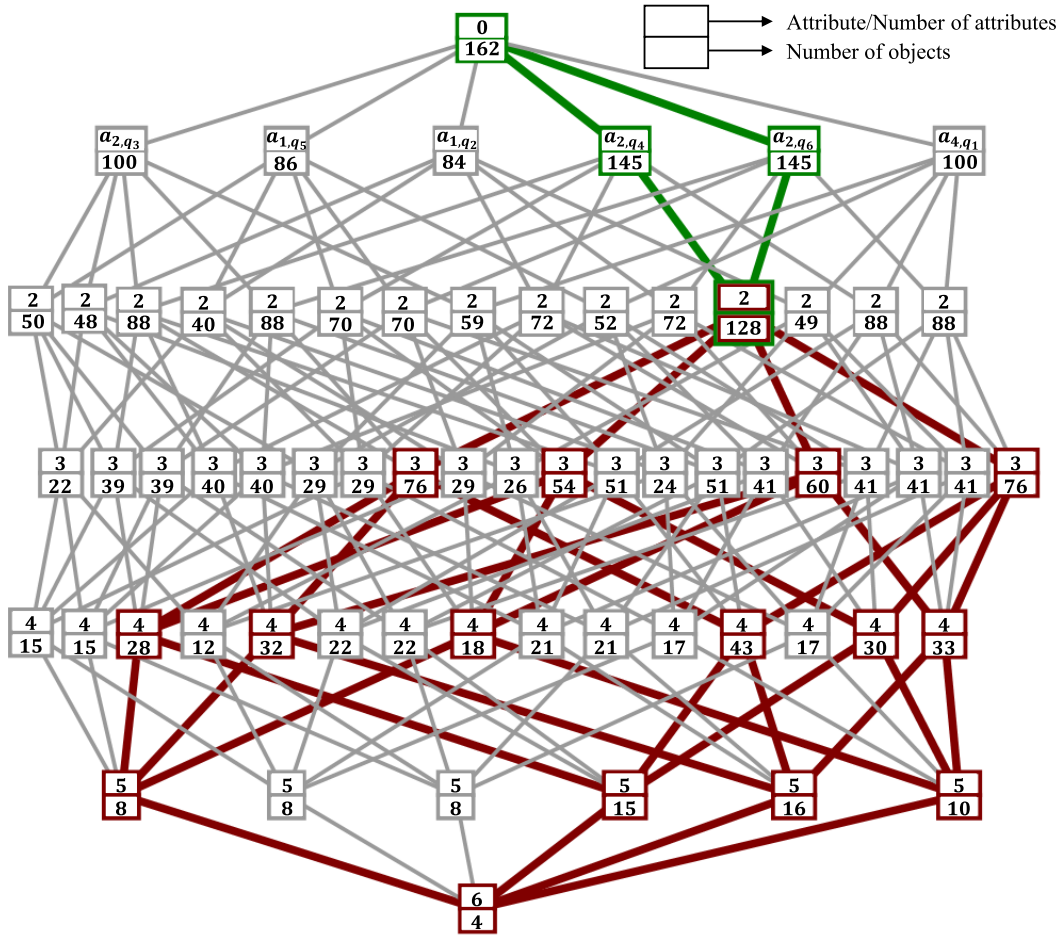
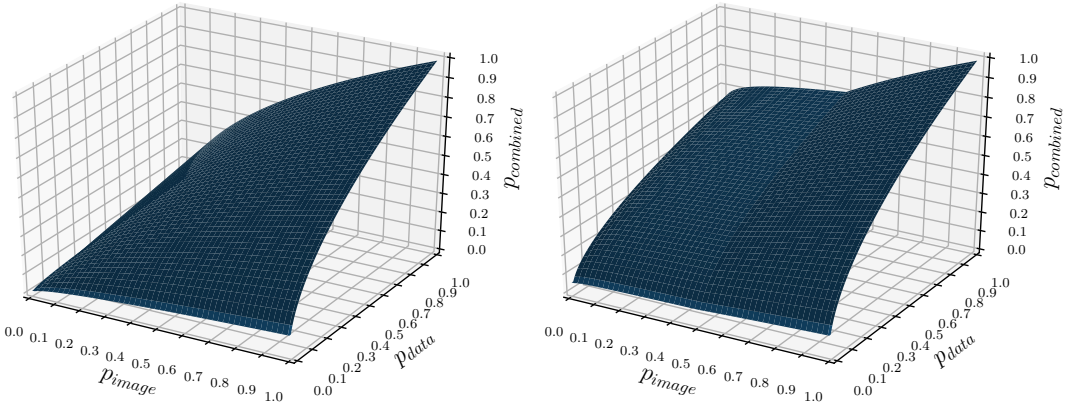


Figure 4.4: Concept lattice view for 162 very low probability score cases in the range $[0, 0.1)$. The top box of a node represents an attribute (answer) or a number of attributes, which are connected by lines, and the bottom box represents how many objects (cases) contain the corresponding attribute shown in the top box. Refer to Table 4.3 for details about the questions and answers.

for the patient data. To achieve this, we made p_{image} less extreme in the lower half probability range using the transformation shown in Equation 4.7. This transformation is popular in the literature of forecast probability aggregation for making the forecasts less or more extreme [8, 77, 145].

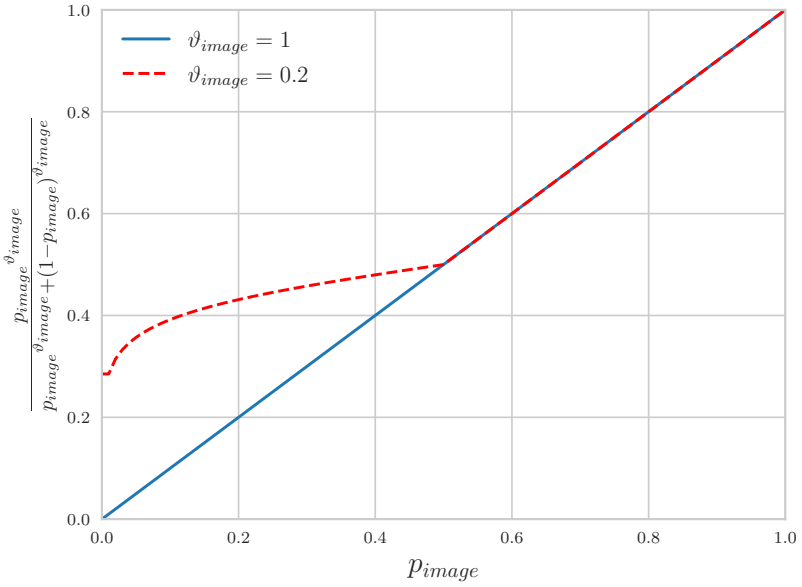
$$\tilde{p}_{image} = \frac{{p_{image}}^{\vartheta_{image}}}{{p_{image}}^{\vartheta_{image}} + (1 - p_{image})^{\vartheta_{image}}} \tag{4.7}$$

The adjustment factor ϑ_{image} was set to 0.2 so that a very low value of p_{image} does not pull down $p_{combined}$ too much. This value was selected based on expert's suggestion to



(a) Geometric mean.

(b) Probability adjusted geometric mean.



(c) Transformed image probability. ϑ_{image} is the transformation factor.

Figure 4.5: Combining erythema migrans probabilities from image and patient data. p_{image} and p_{data} represent probabilities from image and patient data.

ensure that $p_{combined}$ will be at least 50% if $p_{data} \geq 90\%$. The adjustment of p_{image} is shown in Figure 4.5c. The plot of $p_{combined}$ after the adjustment of p_{image} is shown in Figure 4.5b. From the figure, we can see that the veto power was retained for p_{data} while effectively revoking it from p_{image}. As geometric mean uses multiplication we replaced a zero value of p_{image} or p_{data} with a small value of 0.1 to avoid a zero value of $p_{combined}$.

The generalized steps involved in our strategy for combining probabilities from image and patient data are shown in Algorithm 2. The notations, inputs, and outputs are listed at the beginning of the algorithm. First, a zero value of probability from image p_{image} or patient data p_{data} is replaced by a small value ϵ to make sure the combined probability

$p_{combined}$ does not become zero because of the geometric mean. Then, p_{image} and p_{data} are transformed using the transform function. The transform function uses Equation 4.7 to make the input probability less or more extreme based on the transforming factor if the input probability falls within the user-defined range. Finally, the combined probability $p_{combined}$ is calculated using the geometric mean of transformed probabilities from image $\tilde{p}_{image}$ and patient data $\tilde{p}_{data}$. Geometric mean ensures veto power for the modalities which can be adjusted using the transformation with suggestions from domain experts.

Algorithm 2: Combining probabilities from image and patient data

Input :
Probability estimate from lesion image: $p_{image} \in [0, 1]$
Probability estimate from patient data: $p_{data} \in [0, 1]$
Factor for transforming p_{image}: ϑ_{image}
Factor for transforming p_{data}: ϑ_{data}
Value used to avoid zero probability: $\epsilon \in (0, 1]$
Range beginning for transforming p_{image}: $b_{image} \in [0, 1]$
Range end for transforming p_{image}: $e_{image} \in [0, 1]$
Range beginning for transforming p_{data}: $b_{data} \in [0, 1]$
Range end for transforming p_{data}: $e_{data} \in [0, 1]$

Output:
Combined probability: $p_{combined} \in [0, 1]$

begin
- **if** $p_{image} = 0$ **then**
  - $p_{image} \leftarrow \epsilon$ `// avoiding zero probability from image modality`
- **if** $p_{data} = 0$ **then**
  - $p_{data} \leftarrow \epsilon$ `// avoiding zero probability from patient data modality`
- $\tilde{p}_{image} \leftarrow$ `transform` $(p_{image}, \vartheta_{image}, b_{image}, e_{image})$ `// transform` p_{image}
- $\tilde{p}_{data} \leftarrow$ `transform` $(p_{data}, \vartheta_{data}, b_{data}, e_{data})$ `// transform` p_{data}
- $p_{combined} \leftarrow \sqrt{\tilde{p}_{image} \times \tilde{p}_{data}}$ `// geometric mean`
- **return** $p_{combined}$

Function *transform* (p, ϑ, b, e)
- **if** $p >= b$ *and* $p <= e$ **then**
  - $p \leftarrow \frac{p^{\vartheta}}{p^{\vartheta}+(1-p)^{\vartheta}}$ `// transformation in specified range`
- **return** p

4.4 Conclusion

In this chapter, we successfully elicited opinions from fifteen expert doctors to create a model for obtaining EM probability score from patient data. The elicited probability

model will help address the data scarcity problem towards building an effective Lyme disease pre-scanner system. We also proposed a strategy to jointly utilize EM probabilities from both image and patient data. Image-only EM analysis is not robust enough and dark skin is underrepresented in existing EM image datasets. Therefore, image-only analysis is not appropriate for a proper diagnosis of EM. We believe that combining the elicited probability score from patient data with image-based analysis can partially address these issues. The proposed techniques of questionnaire based opinion elicitation and combining probabilities from image and patient data will be useful for other diseases with similar requirements.

Key Points (Chapter 4)

- We proposed a questionnaire based expert opinion elicitation approach that utilizes Gaussian mixture model based density estimation.
- We opted for relative weight assignment to different answers to the questions which is easier for the experts compared to traditional approach of collecting probability estimates.
- We exploited decision tree and formal concept analysis for intuitive validation of the elicited model.
- We proposed an approach for combining the probability score from a deep learning image classifier with the elicited probability score from patient data. The proposed algorithm ensures veto power for the chosen modality based on expert's decision.
- For the first time, we elicited opinions from fifteen expert dermatologists to create a model for calculating erythema migrans probability from patient data as an early symptom of Lyme disease.

5 Miscellaneous

This chapter addresses research question 3, presents our ongoing works on efficiently dealing with dermoscopic skin lesion hair artifact, custom architecture for Lyme disease image classifier, and an application utilizing our research findings. Contents from this chapter have been used in the following publications:

- S. I. Hossain, S. S. Roy, J. de Goër de Herve, R. E. Mercer, and E. Mephu Nguifo. "A skin lesion hair mask dataset with fine-grained annotations". 1, 2023. DOI: 10.17632/J5YWPD2P27.1 (accepted, Data in Brief journal)
- S. I. Hossain, J. de Goër de Herve, Y. Frendo, D. Martineau, I. Lebert, O. Lesens, and E. Mephu Nguifo. "EMScan : une application mobile pour l'assistance au diagnostic des formes précoces de la maladie de Lyme". *Revue des Nouvelles Technologies de l'Information* Extraction et Gestion des Connaissances, RNTI-E-39, 2023, pp. 613–620. URL: https://editions-rnti.fr/?inprocid=1002869
- Y. Frendo, J. de Goër de Herve, S. I. Hossain, D. Martineau, I. Lebert, O. Lesens, and E. Mephu Nguifo. "EMScan: A Mobile Application for Early Lyme Disease Diagnosis". In: *European Conference on Computer Vision ECCV*. 2022. Project Demo. URL: https://eccv2022.ecva.net/program/demo-list/

Chapter Contents

5.1 Introduction

Artificial intelligence-assisted skin lesion analysis is becoming popular nowadays thanks to the advancement in deep learning techniques. However, their performances may be affected by skin hair artifacts. Lesion analysis can benefit from digital hair removal or realistic hair simulation techniques as discussed in Section 2.3. An accurate hair mask segmentation dataset is needed to properly benchmark the segmentation algorithms. Moreover, existing researches on skin hair augmentation require a hair mask to generate hair in specified locations. These masks are created using pre-segmented hair masks or random lines or curves [5]. A well-annotated hair mask dataset will be effective for training generative models to automate the mask generation process.

We have created the largest publicly available skin lesion hair segmentation mask dataset by carefully annotating 500 dermoscopic images. Compared to the existing datasets, our dataset is free of non-hair artifacts like ruler markers, bubbles, and ink marks. The dataset is also less prone to over and under segmentations because of fine-grained annotations and quality checks from multiple independent annotators. To create the dataset, first, we collected five hundred copyright-free CC0 licensed dermoscopic images covering different hair patterns. Second, we trained a deep learning hair segmentation model on a publicly available weakly annotated dataset. Third, we extracted hair masks for the selected five hundred images using the segmentation model. Finally, we manually corrected all the segmentation errors and verified the annotations by superimposing the annotated masks on top of the dermoscopic images. Multiple annotators were involved in the annotation and verification process to make the annotations as error-free as possible.

The prepared dataset will be useful for benchmarking and training hair segmentation algorithms as well as creating realistic hair augmentation systems. Our first plan is to train an accurate hair segmentation model utilizing the prepared dataset to extract lots of hair masks from dermoscopic datasets. Then, these masks can be utilized to train a generative model which can automate the process of realistic hair mask generation for the hair augmentation process. Finally, we want to test if hair augmentation can be an effective replacement for hair removal or not.

We are working on a custom convolutional neural network (CNN) architecture targeting the task of classifying erythema migrans (EM) from images. The architecture utilizes findings from our analysis of existing architectures as described in Chapter 3. The initial results look promising and we are planning to further improve the architecture with neural architecture search (NAS) and our proposed pre-training strategy.

The techniques proposed in this thesis have been utilized in a prototype mobile application for assisting with the early diagnosis of Lyme disease. Initial trials with the application are getting positive feedback from the community.

The rest of the chapter is structured as follows: Section 5.2 describes our ongoing work on efficiently dealing with skin lesion hair artifact; Section 5.3 presents the custom architecture for EM image classifier, and the work plan to further improve it; Section 5.4 briefly describes the application utilizing findings from the thesis; finally, Section 5.5 presents concluding remarks.

5.2 Efficiently Dealing With Dermoscopic Skin Lesion Hair Artifact

The following subsections describe our prepared dermoscopic skin lesion hair mask dataset and work plan to effectively handle the hair artifact.

5.2.1 A Skin Lesion Hair Mask Dataset With Fine-grained Annotations

The following subsections present our motivation behind creating the dataset, value of the data, data description, and methods of dataset preparation.

5.2.1.1 Motivation

According to our study, the largest publicly available skin lesion hair mask dataset [89] contains annotations for 306 images but with 18 duplicates and suffers from under-segmentation, and non-hair artifacts. Gallucci [42] created a dataset of 75 images only, which lacks complex patterns and is not a well-representative of the broader skin hair distribution. Akyel et al. [3] prepared a non-public dataset of 2500 images. However, it contains rulers, ink spots, and other noises alongside skin hair. Our motivation for creating the dataset was to resolve the issues in available datasets.

5.2.1.2 Value of the Data

- This is the largest publicly available fine-grained skin lesion hair segmentation mask dataset. High-quality hand-annotated segmentation masks are costly and time-consuming to produce.
- This is the only dataset free of non-hair artifacts.
- This dataset will be useful for proper benchmarking of hair segmentation algorithms, as it is free of non-hair artifacts and segmentation errors.

- Our dataset can be used to train a generative model for automating the task of realistic skin hair mask generation.
- The dataset will contribute to skin lesion research by allowing researchers to train robust skin lesion hair segmentation algorithms.

5.2.1.3 Data Description

Our dataset is publicly available in an online repository[1] [63]. It contains skin hair annotation masks for 500 dermoscopic images collected from ISIC 2018 dataset [26]. The dataset is organized into three folders namely dermoscopic_image, hair_mask, and overlay. Table 5.1 shows some example images from each of the folders. The dermoscopic_image folder contains 500 dermoscopic images handpicked from the primary image source covering different hair patterns. We retained the original names of the image files from the primary image source. The hair_mask folder contains a binary segmentation mask for each of the images of the dermoscopic_image folder. In a segmentation mask image, white pixels represent skin hair and black pixels represent background. The overlay folder contains hair mask images superimposed on the original dermoscopic images. We provided the superimposed images for easy public verification so that, other people can report any annotation mistakes and contribute to improving the dataset. Images in the hair_mask and overlay folders share the same names as the primary images in the dermoscopic_image folder.

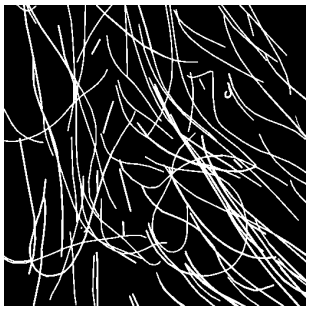
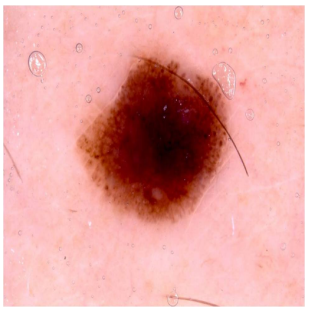
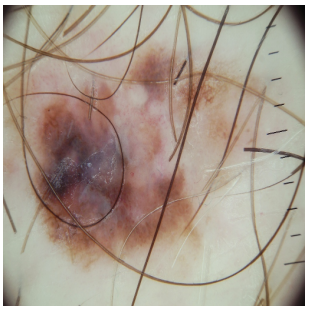
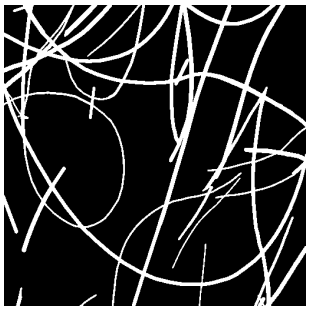
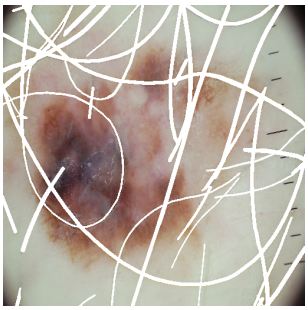
5.2.1.4 Dataset Design, Materials and Methods

Annotating skin lesion hair from scratch is a tedious task. To ease the process, we trained a U-Net [131] deep segmentation model using a weakly annotated dataset provided by Li et al. [89]. U-Net is a popular CNN architecture for image segmentation tasks. It is made up of a contracting path that captures the image's context and an expansive path that creates the segmented output. In order to maintain spatial information, the network uses skip connections, which enables accurate segmentation even when the target objects are small. The U-Net architecture is illustrated in Figure 5.1 The codes used for the process are publicly available in the unet folder in a github repository[2]. Inside the unet folder the U-Net model is defined in model.py file, unet training is performed using the unet_training.ipynb python notebook file and the task of predicting initial masks for the dermoscopic images are done using the predict_mask.ipynb file.

Using the trained U-Net we extracted the initial hair mask for 500 handpicked copyright-free dermoscopic skin lesion images from ISIC 2018 dataset [26] to cover different hair patterns. The resulting masks suffer from various segmentation errors like

[1] https://data.mendeley.com/datasets/j5ywpd2p27 (visited on 02/20/2023).
[2] https://github.com/imranrana/Skin-Lesion-Hair-Mask-Dataset (visited on 02/20/2023).

Table 5.1: Samples from the prepared skin lesion hair mask dataset.

File	Folder		
	dermoscopic_image	hair_mask	overlay
ISIC_0000115.png			
ISIC_0000200.png			
ISIC_0009992.png			
Total: 1500 images (500 images per folder)			

under-segmentation, over-segmentation, and non-hair artifacts. We involved three independent annotators for the correction of the segmentation errors.

The first annotator manually corrected all the found segmentation errors with Adobe Photoshop software [70]. A video demonstration of the segmentation mask editing process using Adobe Photoshop software is available in the mask_editing_process.mp4 file of our GitHub repository. The steps involved are as follows:

- Open the dermoscopic image in photoshop.

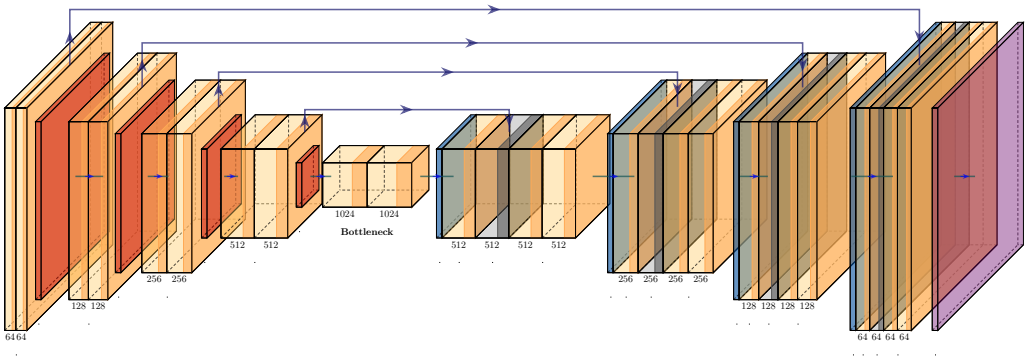


Figure 5.1: U-Net architecture. The final output layer uses sigmoid activation.

- Open the initial segmentation mask image in photoshop and copy it on top of the dermoscopic image.
- Change the blending mode of the mask image to “screen”.
- Select the brush type as hard brush (hardness of the brush set to 100 percent).
- Remove unwanted segmentation marks from the mask image by painting with a black brush.
- Adjust the brush size according to the width of the skin hair and add missing segmentation marks to the mask image by painting with a white brush.
- Change back the blending mode of the mask image to “normal” mode.
- Make additional adjustments to the segmentation mask if required.
- Save the finalized segmentation mask image in the desired format.

To verify the quality of the annotation first we binarized each corrected masks to make sure every pixel is either black or white. Then, we made the black pixels of the mask image fully transparent and superimposed it on the original dermoscopic image. Finally, we created a collage of three types of images: dermoscopic image, corrected mask, and the superimposed image for easy verification. Each collage looks like a row from Table 5.1. The code used for these operations is available in the check_annotation.ipynb file in the github repository[2]. Using the image collage a second annotator marked errors missed by the first annotator. A third annotator corrected the mistakes identified by the second annotator, which was finally reverified by the first annotator. We tried to make the annotations as error-free as possible. The overall dataset creation workflow is shown in Figure 5.2.

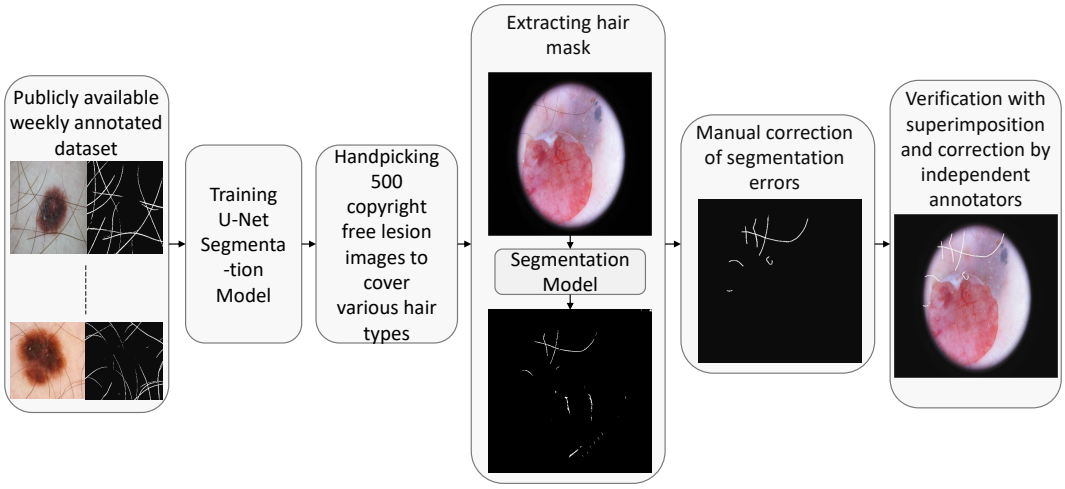


Figure 5.2: Skin hair mask dataset creation workflow.

5.2.2 Work Plan

To address the research question 3 we want to investigate how augmenting training data with skin hair impacts model performance compared to hair removal. First, we will try to train a generative model to automate the skin hair mask generation process for the augmentation algorithms. If we do not get satisfactory results from the generative model with our created 500 images then, we will train another segmentation model with our dataset and use that model to extract more samples of hair masks from dermoscopic datasets. These images will be useful for the training of the generative model.

The skin hair augmentation pipeline will work as follows:

i. Remove skin hair from input lesion image with a hair removal algorithm [89].

ii. Generate a hair mask with the trained generative model.

iii. Augment hair on the dermoscopic image using the generated mask and a realistic hair simulator [5].

We will train a model for dermoscopic image classification using the augmentation pipeline and compare the performance with a model that does not use hair augmentation but uses a hair removal pre-processing step. This study will help to decide if training time hair augmentation can effectively replace test time hair removal pre-processing or not.

5.3 Custom Architecture for Lyme Disease Image Classifier

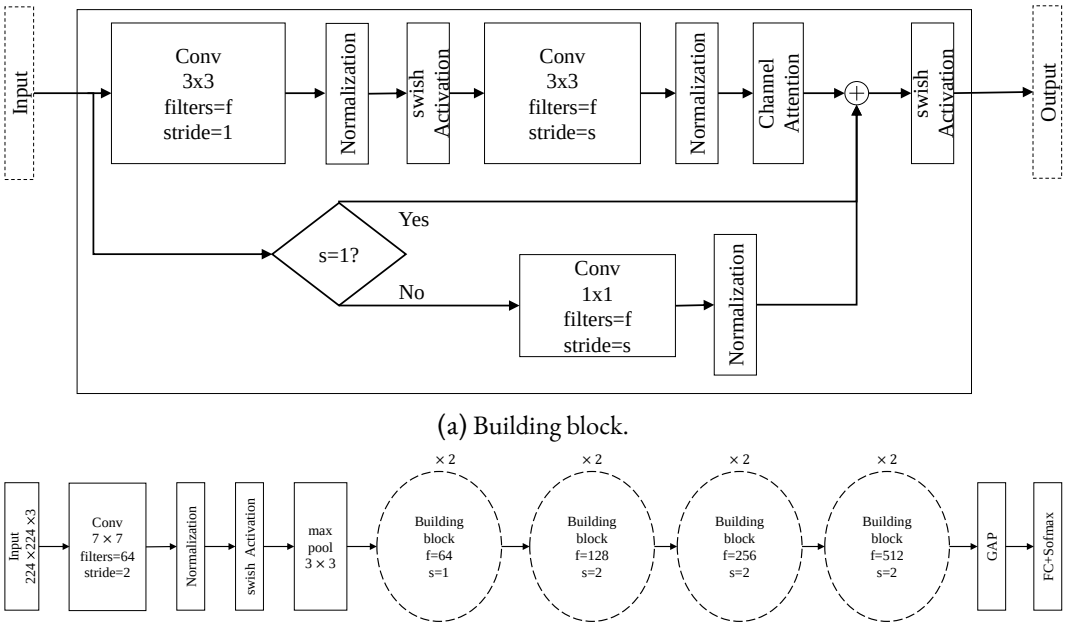
From the experimental results discussed in Section , we saw that ResNet50-141 model performed the best in terms of accuracy for the Lyme disease image classification task and also EfficientNet variations showed good results in heatmap visualization. Taking these factors into consideration we tested a custom architecture as shown in Figure 5.3. We opted for a residual block incorporating efficient channel attention [172] and swish activation as shown in Figure 5.3a. The architecture design is like a ResNet18 model as shown in Figure 5.3b. The source code for the custom architecture is available in a github repository[3]. We tested the architecture on the second version of the dataset that includes some label corrections and additional images. The architecture was trained from scratch without any pre-training. The result looks promising compared to the best performing ResNet50-141 model as shown in Table 5.2. The confusion matrix, ROC curve, and cross-validation fold-wise details are available in Appendix Section D.2. The custom architecture has 11.19 million parameters compared to 23.59 million of ResNet50-141. Depthwise separable convolution can be effective in further reducing the parameters but it decreases accuracy. In dilated convolution [183], a larger effective kernel size is achieved by introducing gaps, or dilations, between the kernel values as shown in Figure 5.4a. Without increasing the number of parameters or the computational cost, it enables the network to have a bigger receptive field. Combining dilated and depthwise convolutions [153] in a residual block (as shown in Figure 5.4b) and optimally placing them with the block of Figure 5.3a utilizing NAS [129] can produce an effective architecture. Particularly, we plan to use NAS utilizing my previously proposed particle swarm optimization with selective search[4] for finding the optimized arrangement of these building blocks. Particle swarm optimization with selective search retains the intermediate best result during the particle update process and performs better than vanilla particle swarm optimization. Also, using our proposed pre-training strategy with the architecture may further increase performance.

5.4 Application From the Thesis

The techniques proposed in this thesis have been utilized in a mobile application called EMScan. The application was developed as part of the DAPPEM (Développement d'une APPlication d'identification des Erythèmes Migrants à partir de photographies) project funded by European Regional Development Fund. The EMScan mobile application was

[3] https://github.com/imranrana/Lyme-Disease (visited on 03/15/2023).

[4] S. I. Hossain, M. A. Akhand, M. I. Shuvo, N. Siddique, and H. Adeli. "Optimization of University Course Scheduling Problem using Particle Swarm Optimization with Selective Search". *Expert Systems with Applications* 127, 2019, pp. 9–24. ISSN: 09574174. DOI: 10.1016/j.eswa.2019.02.026.



(a) Building block.

(b) Architecture.

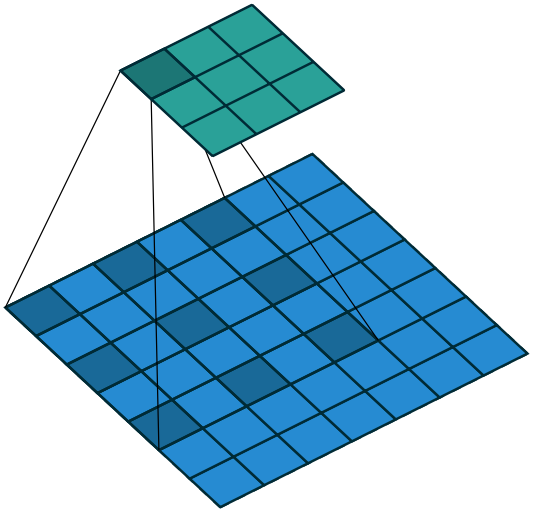
Figure 5.3: Custom architecture design for Lyme image classifier.

Table 5.2: Experimental results with custom architecture. Within each cell, the value after (±) symbol represents the standard deviation across five folds. Second version of the prepared Lyme dataset was used for the experiments.

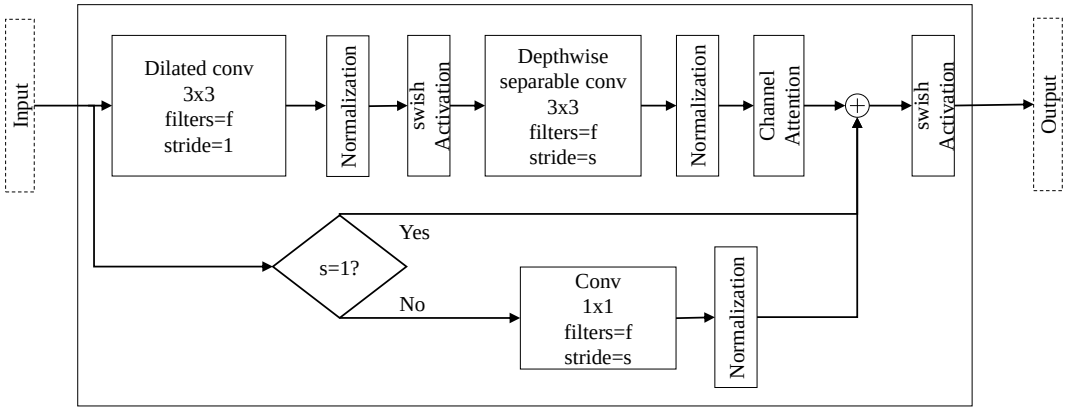
	Metric										
Model	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
ResNet50-141	84.85 ±1.23	87.49 ±1.83	81.93 ±1.42	84.29 ±1.09	85.57 ±1.89	0.6963 ±0.0249	0.6956 ±0.0246	4.8707 ±0.3856	0.1528 ±0.0227	0.8585 ±0.0118	0.9231 ±0.0069
Custom	84.68 ±2.62	87.05 ±3.4	82.05 ±4.0	84.38 ±3.03	85.24 ±3.44	0.6936 ±0.0526	0.6922 ±0.0525	5.1201 ±1.2442	0.1582 ±0.0434	0.8565 ±0.0248	0.9151 ±0.0214

created with the goal of assisting doctors and general people with early diagnosis and suggestion, and also to advance artificial intelligence assisted Lyme disease diagnosis research. The goals of the application are shown in Figure 5.5. Initial trials with the prototype showed promising results for real-life applications. A video demonstration of the application is available on DAPPEM project website[5]. The overall workflow of the EMScan mobile application is described in Appendix Section D.1.

[5] https://dappem.limos.fr (visited on 02/20/2023).



(a) Illustration of dilated convolution [35].



(b) Custom building block.

Figure 5.4: Custom building block utilizing dilated and depthwise separable convolutions.

5.5 Conclusion

In this chapter, we have discussed our ongoing research works. We have created the largest publicly available dermoscopic skin lesion hair mask annotation dataset which can be utilized for training accurate segmentation algorithms and also to enhance the hair augmentation process. Further study is required to see if training time hair augmentation can be an effective replacement for test time hair removal or not. We are also working on a custom architecture for Lyme image classifier. NAS and our proposed pre-training strategy can be utilized to enhance the architecture and its performance. Our proposed techniques were utilized in a mobile application that looks promising for assisting with early Lyme disease diagnosis.

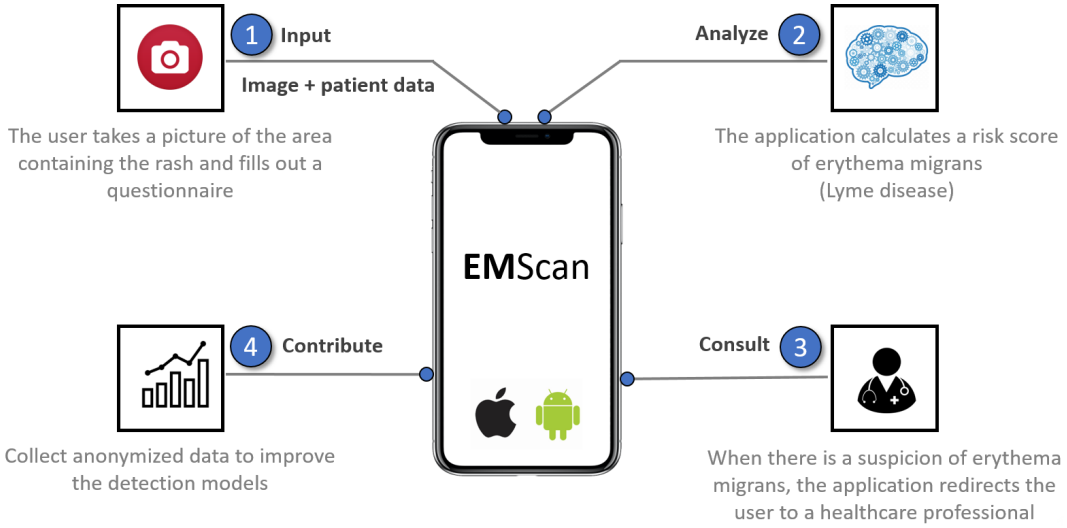


Figure 5.5: EMScan application goals.

Key Points (Chapter 5)

- We have created the largest publicly available dermoscopic skin lesion hair mask annotation dataset by carefully annotating 500 images.
- The prepared dataset will be useful for benchmarking and training hair segmentation algorithms as well as creating realistic hair augmentation systems.
- A hand-designed CNN architecture showed promising results for EM classification from images. The architecture can be further optimized with NAS.
- The techniques proposed in this thesis were used in a prototype mobile application for assisting with early Lyme disease diagnosis. Initial trials seem effective for real-life applications.

6 Conclusions

This chapter presents a short summary of the key findings of this thesis, possible future research directions, and publications resulting from the thesis.

Chapter Contents

6.1 General Conclusion and Research Findings

In this thesis, we tried to tackle the data scarcity problem of artificial intelligence based multimodal skin lesion analysis. We addressed the challenges of a small clinical skin lesion image dataset and also the lack of training data for patient modality.

First, to deal with image data scarcity of clinical skin lesion images we proposed a pre-training strategy that involves fine-tuning some layers from the end of an ImageNet pre-trained convolutional neural network (CNN) architecture using a dermoscopic dataset before training the model on a clinical skin lesion dataset. Experimental results using a novel Lyme disease dataset built as part of the thesis showed the effectiveness of the proposed approach for improving CNN performance. In order to evaluate the efficacy of CNNs for Lyme disease diagnosis using erythema migrans (EM) pictures, we used the proposed strategy to compare well-known CNNs and the results suggest that even lightweight models, such as EffiicentNetB0, performed admirably, pointing to the potential use of CNNs in Lyme disease pre-scanner mobile applications that can assist people with a preliminary assessment.

Second, to address the scarcity problem of patient data we have proposed a questionnaire-oriented expert opinion elicitation approach that can provide disease probability in the absence of training data. As it is difficult and time consuming for doctors to provide probability estimates for all possible cases or distribution parameters we collected relative weight assignments to different answers to the questions and converted

the doctor's evaluations to probabilities utilizing Gaussian mixture model based density estimation. We also proposed the use of formal concept analysis and decision trees for easy model validation. The proposed approach is easy for doctors to follow. We also proposed an approach for combining the probability estimates from CNN image classifier and opinion elicited disease probability by considering the expert's choice. The proposed techniques proved effective when applied to a Lyme disease diagnosis scenario.

Third, to address the problem of skin hair artifact on dermoscopic images we have created the largest publicly available skin lesion hair mask annotation dataset by carefully annotating five hundred dermoscopic images. The dataset can be utilized for training accurate segmentation algorithms and also to enhance the hair augmentation process. Currently, we are working on hair augmentation utilizing the prepared dataset and also on a lightweight architecture for EM image classification.

The techniques proposed in this thesis were applied to a mobile application for assisting with the early diagnosis of Lyme disease. Initial trials with the prototype showed promising results for real-life application and we believe that these techniques will be useful for addressing data scarcity issues in similar diseases.

6.2 Limitations and Future Research Directions

We have used supervised learning for our proposed pre-training strategy. Using self-supervised learning for the pre-training can be an interesting study. The search for the number of layers to unfreeze from the pre-trained model takes time as it requires training different versions of the model on the target dataset. Although, it takes does not take very long as most of the layers are frozen the search time can be improved by utilizing techniques used to reduce candidate architecture evaluation time from neural architecture search (NAS) literature[177]. Also, the pre-training approach can be tested with clinical skin lesion image pre-training in place of dermoscopic images.

Our work on combining probability estimates from CNN and the elicited model needs to be validated using real case scenarios. We plan to collect real scenarios of Lyme disease cases by deploying the mobile application. After collecting sufficient data the performance of the elicited model and approach of combining probabilities can be compared with a multimodal model jointly trained using multimodal training data. It would be interesting to see if calibrating CNN [46] to make its predicted confidence score more accurate representative of the true probability estimate provides better performance or not.

Our prepared skin lesion hair mask dataset can be utilized for training generative models to automate the task of hair mask generation process of realistic hair augmentation techniques. Another interesting study would be to see if training time hair augmentation can be an effective replacement for test time hair removal or not.

The custom architecture for EM image classification can be optimized using NAS with the building blocks described in Section 5.3. My previously proposed particle swarm optimization with selective search technique[1] that retains the intermediate best solution during the search process can be an interesting choice for performing the NAS. Also, utilizing our proposed pre-training strategy may improve the model's performance. The optimized model can be a good option for deploying in a mobile application.

In this study, we have considered the case of differential diagnosis for Lyme disease i.e. differentiating between EM and similar skin lesions. As the used CNN architectures are not distance aware by default they classify out-of-distribution data with high confidence. For real-life applications, there will be trust issues among users if the model classifies unrelated images as EM. So, out-of-distribution image detection [66, 92, 179] needs to be studied and utilized for improving the application.

6.3 Data Statement

All the research data associated with this thesis are available on the DAPPEM website[2].

6.4 Research Publications

The following publications resulted from the findings of the thesis:

Research Article

- S. I. Hossain, J. de Goër de Herve, M. S. Hassan, D. Martineau, E. Petrosyan, V. Corbin, J. Beytout, I. Lebert, J. Durand, I. Carravieri, A. Brun-Jacob, P. Frey-Klett, E. Baux, C. Cazorla, C. Eldin, Y. Hansmann, S. Patrat-Delon, T. Prazuck, A. Raffetin, P. Tattevin, G. Vourc'h, O. Lesens, and E. Mephu Nguifo. "Exploring convolutional neural networks with transfer learning for diagnosing Lyme disease from skin lesion images". *Computer Methods and Programs in Biomedicine* 215, 2022, p. 106624. ISSN: 01692607. DOI: 10.1016/j.cmpb.2022.106624

- S. I. Hossain, J. de Goër de Herve, D. Abrial, R. Emilion, I. Lebertb, Y. Frendo, D. Martineau, O. Lesens, and E. Mephu Nguifo. "Expert Opinion Elicitation for Assisting Deep Learning based Lyme Disease Classifier with Patient Data", 2022. arXiv: 2208.14384 (submitted)

[1] S. I. Hossain, M. A. Akhand, M. I. Shuvo, N. Siddique, and H. Adeli. "Optimization of University Course Scheduling Problem using Particle Swarm Optimization with Selective Search". *Expert Systems with Applications* 127, 2019, pp. 9–24. ISSN: 09574174. DOI: 10.1016/j.eswa.2019.02.026.

[2] https://dappem.limos.fr (visited on 02/20/2023).

- S. I. Hossain, S. S. Roy, J. de Goër de Herve, R. E. Mercer, and E. Mephu Nguifo. "A skin lesion hair mask dataset with fine-grained annotations". 1, 2023. DOI: 10.17632/J5YWPD2P27.1 (accepted, Data in Brief journal)

Research Demonstration

- S. I. Hossain, J. de Goër de Herve, Y. Frendo, D. Martineau, I. Lebert, O. Lesens, and E. Mephu Nguifo. "EMScan : une application mobile pour l'assistance au diagnostic des formes précoces de la maladie de Lyme". *Revue des Nouvelles Technologies de l'Information* Extraction et Gestion des Connaissances, RNTI-E-39, 2023, pp. 613–620. URL: https://editions-rnti.fr/?inprocid=1002869

- Y. Frendo, J. de Goër de Herve, S. I. Hossain, D. Martineau, I. Lebert, O. Lesens, and E. Mephu Nguifo. "EMScan: A Mobile Application for Early Lyme Disease Diagnosis". In: *European Conference on Computer Vision ECCV*. 2022. Project Demo. URL: https://eccv2022.ecva.net/program/demo-list/

Doctoral Consortium

- S. I. Hossain. "Early Diagnosis of Lyme Disease by Recognizing Erythema Migrans Skin Lesion from Images Utilizing Deep Learning Techniques". In: *IJCAI International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Vienna, 2022, pp. 5855–5856. ISBN: 9781956792003. DOI: 10.24963/ijcai.2022/830.

Research Talk

- S. I. Hossain, J. de Goër de Herve, D. Abrial, R. Emilion, I. Lebertb, Y. Frendo, D. Martineau, O. Lesens, and E. Mephu Nguifo. "Assisting Deep Learning based Lyme Disease Classifier with Patient Data". In: *Apprentissage automatique multimodal et fusion d'informations (3ième édition)*. 2022. URL: https://www.gdr-isis.fr/index.php/reunion/485/

- S. I. Hossain, E. Mephu Nguifo, and J. de Goër de Herve. "Early Diagnosis of Lyme Disease by Recognizing Erythema Migrans Skin Lesion from Images Utilizing Deep Learning Techniques". In: *Deep learning with weak or few labels in medical image analysis*. 2022. URL: https://www.gdr-isis.fr/index.php/reunion/468/

A Appendices for Chapter 2

A.1 Activation Functions

Table A.1 lists some of the most commonly used activation functions in deep learning.

Table A.1: Activation functions.

Activation function	**Equation**
Identity	$f(x) = x$
Sigmoid [110]	$f(x) = \frac{1}{1+e^{-x}}$
Swish [127]	$f(x) = \frac{x}{1+e^{-x}}$
Hyperbolic tangent (tanh) [99]	$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
Rectified Linear Unit (ReLU) [99]	$f(x) = max(0, x)$
Gaussian Error Linear Unit (GELU) [54]	$f(x) = x\Phi(x)$ where $\Phi(x)$ is the Gaussian cumulative distribution function.
Softmax [43]	Normalizes an input vector z of n real numbers into a probability distribution $f(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$ for $i = 1, \ldots, n$ and $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$

A.2 Gradient Descent and Adam Optimizer

Gradient descent also known as batch gradient descent is an optimization algorithm for locating a differentiable objective function's local minimum. It iteratively reduces the value of the objective function by adapting model parameters. The iterative parameter update equation of gradient descent is shown in Equation A.1.

$$\theta = \theta - \eta \cdot \nabla_\theta L(\theta) \tag{A.1}$$

where θ represents model parameters, $L(\theta)$ is the objective function, η is the learning rate, and $\nabla_\theta L(\theta)$ is the gradient of the objective function w.r.t the parameters.

Adaptive moment estimation (Adam) optimizer [80] calculates adaptive learning rate for each individual parameter and uses exponentially decaying average of past gradients (m_t) and squared gradients(v_t) as shown in the following equation:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \end{aligned} \tag{A.2}$$

where $g_t = \nabla_\theta L_t(\theta_{t-1})$ is the gradient of the objective function at timestep t, and $\beta 1, \beta 2 \in [0, 1)$ are hyper-parameters to control the exponential decay rates of moving averages. m_t and v_t are bias corrected as follows:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2^t} \end{aligned} \tag{A.3}$$

The parameter update rule of Adam optimizer is shown in Equation A.4.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{A.4}$$

where ϵ is a small number for preventing division by zero. The author proposed default values for $\beta_1, \beta_2,$ and ϵ are $0.9, 0.999$ and 10^{-8} respectively.

B Appendices for Chapter 3

B.1 Online Resources

Trained convolutional neural network models can be downloaded from becnhmarking page of DAPPEM website[1] by clicking on "*Downloads*" link.

[1] https://dappem.limos.fr/benchmarking.html (visited on 02/20/2023)

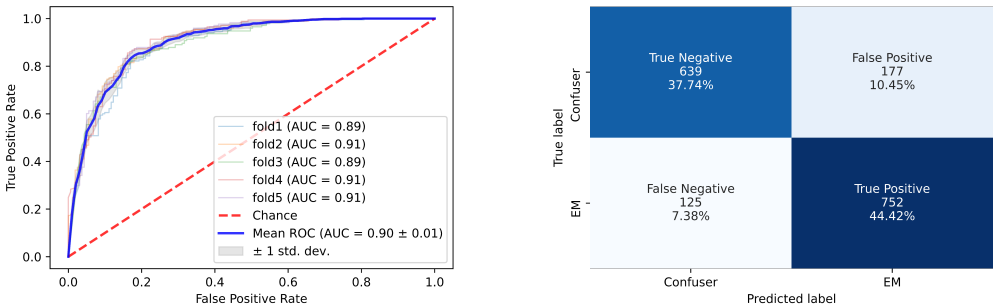
B.2 Supplementary Data for Trained CNN Models

This section provides detailed five-fold cross validation results of all the trained models.

B.2.1 VGG16-8

Table B.1: Five-fold cross-validation performance metrics of VGG16-8 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	81.18	84.32	77.78	80.41	82.1	0.6231	0.6223	3.7946	0.2015	0.8232	0.8891
fold2	82.09	81.5	82.72	83.43	80.72	0.6419	0.6417	4.7155	0.2236	0.8246	0.9052
fold3	80.54	87.28	73.29	77.84	84.29	0.6134	0.6085	3.268	0.1735	0.8229	0.8943
fold4	83.23	91.91	73.91	79.1	89.47	0.6719	0.6622	3.5231	0.1095	0.8503	0.9087
fold5	83.83	83.82	83.85	84.8	82.82	0.6764	0.6764	5.1901	0.193	0.843	0.9081
average	82.17	85.77	78.31	81.12	83.88	0.6453	0.6422	4.0983	0.1802	0.8328	0.9011
std. deviation	1.23	3.58	4.36	2.62	3.02	0.0253	0.0249	0.7329	0.0388	0.0116	0.0079



(a) ROC curve.

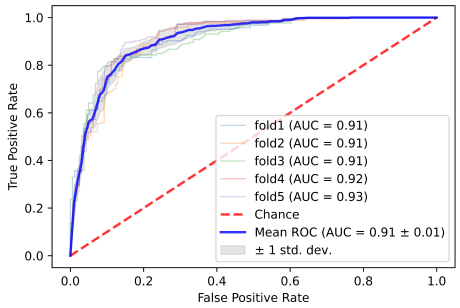
(b) Confusion matrix.

Figure B.1: Five-fold cross-validation ROC curve and confusion matrix of VGG16-8 model.

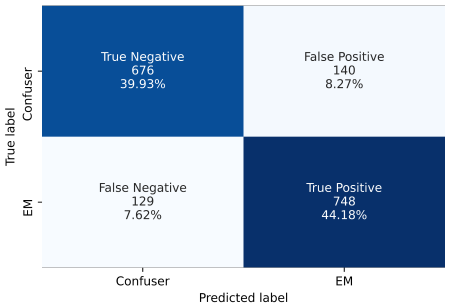
B.2.2 VGG19-13

Table B.2: Five-fold cross-validation performance metrics of VGG19-13 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	81.74	85.41	77.78	80.61	83.12	0.6346	0.6334	3.8432	0.1876	0.8294	0.907
fold2	84.18	84.97	83.33	84.48	83.85	0.6832	0.6832	5.0983	0.1803	0.8473	0.9079
fold3	83.83	83.82	83.85	84.8	82.82	0.6764	0.6764	5.1901	0.193	0.843	0.9069
fold4	84.13	83.82	84.47	85.29	82.93	0.6825	0.6824	5.3977	0.1916	0.8455	0.9176
fold5	86.83	88.44	85.09	86.44	87.26	0.7362	0.736	5.9328	0.1359	0.8743	0.9254
average	84.14	85.29	82.9	84.32	84	0.6826	0.6823	5.0924	0.1777	0.8479	0.913
std. deviation	1.62	1.69	2.63	1.97	1.67	0.0323	0.0326	0.6884	0.0214	0.0146	0.0074



(a) ROC curve.



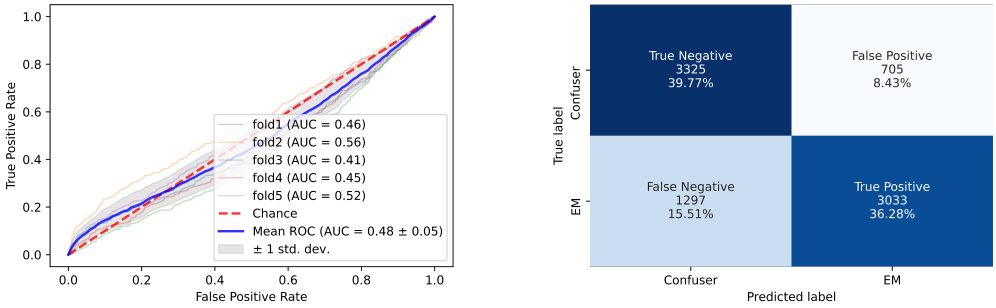
(b) Confusion matrix.

Figure B.2: Five-fold cross-validation ROC curve and confusion matrix of VGG19-13 model.

B.2.3 ResNet50-Burlina

Table B.3: Performance metrics of ResNet50-Burlina models[2] trained by Burlina et al. [15] tested on the whole dataset of this study.

	Metric										
Model	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	75.12	65.7	85.24	82.7	69.82	0.5172	0.5055	4.4502	0.4024	0.7323	0.4646
fold2	75.6	74.48	76.8	77.52	73.69	0.5125	0.512	3.2102	0.3323	0.7597	0.5589
fold3	75.72	66.4	85.73	83.33	70.37	0.5291	0.5174	4.6536	0.392	0.7391	0.4143
fold4	76.67	69.98	83.87	82.34	72.22	0.542	0.5355	4.3386	0.358	0.7566	0.4509
fold5	77.15	73.67	80.89	80.56	74.09	0.5461	0.5439	3.8558	0.3255	0.7696	0.5162
average	76.05	70.05	82.51	81.29	72.04	0.5294	0.5229	4.1017	0.362	0.7515	0.481
std. deviation	0.74	3.6	3.31	2.1	1.71	0.0132	0.0145	0.5172	0.0309	0.0137	0.0509



(a) ROC curve.

(b) Confusion matrix.

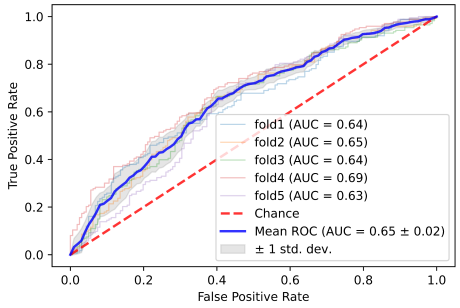
Figure B.3: ROC curve and confusion matrix of ResNet50-Burlina models trained by Burlina et al. [15] tested on the whole dataset of this study.

[2] https://github.com/neil454/lyme-1600-model (visited on 02/20/2023).

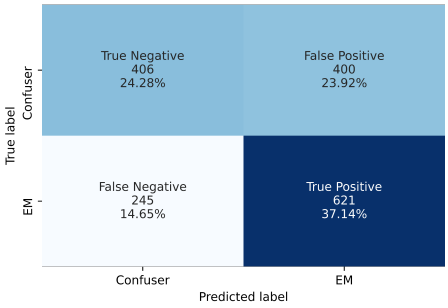
B.2.4 ResNet50-NoAug

Table B.4: Five-fold cross-validation performance metrics of ResNet50-NoAug model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR_+	LR_-	F1-Score	AUC
fold1	59.7	56.9	62.73	62.26	57.39	0.1964	0.1956	1.5267	0.6871	0.5946	0.6429
fold2	62.09	71.1	52.47	61.5	62.96	0.2401	0.2369	1.4958	0.5508	0.6595	0.6491
fold3	61.98	71.1	52.17	61.5	62.69	0.2372	0.2341	1.4866	0.5539	0.6595	0.6405
fold4	63.17	76.3	49.07	61.68	65.83	0.2642	0.2559	1.4981	0.483	0.6822	0.6915
fold5	60.18	83.24	35.4	58.06	66.28	0.213	0.1895	1.2886	0.4735	0.6841	0.6285
average	61.42	71.73	50.37	61	63.03	0.2302	0.2224	1.4592	0.5497	0.656	0.6505
std. deviation	1.29	8.65	8.79	1.5	3.17	0.0234	0.0256	0.0863	0.0764	0.0325	0.0216



(a) ROC curve.



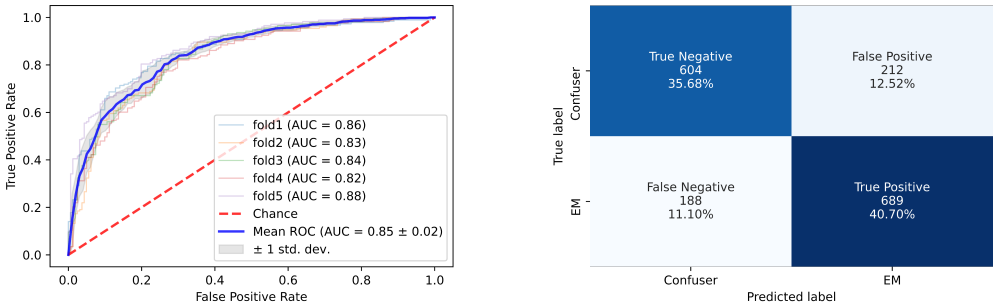
(b) Confusion matrix.

Figure B.4: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-NoAug model.

B.2.5 ResNet50-NTL

Table B.5: Five-fold cross-validation performance metrics of ResNet50-NTL model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	78.09	83.78	71.93	76.35	80.39	0.5623	0.5594	2.9848	0.2254	0.799	0.8565
fold2	77.01	79.77	74.07	76.67	77.42	0.5396	0.5392	3.0768	0.2731	0.7819	0.8342
fold3	76.05	82.66	68.94	74.09	78.72	0.5221	0.5183	2.6616	0.2515	0.7814	0.8423
fold4	71.86	61.85	82.61	79.26	66.83	0.4527	0.441	3.5564	0.4618	0.6948	0.8248
fold5	78.74	84.39	72.67	76.84	81.25	0.5758	0.5727	3.088	0.2148	0.8044	0.8776
average	76.35	78.49	74.04	76.64	76.92	0.5305	0.5261	3.0735	0.2853	0.7723	0.8471
std. deviation	2.43	8.47	4.6	1.64	5.22	0.0431	0.0464	0.2867	0.0906	0.0398	0.0185



(a) ROC curve.

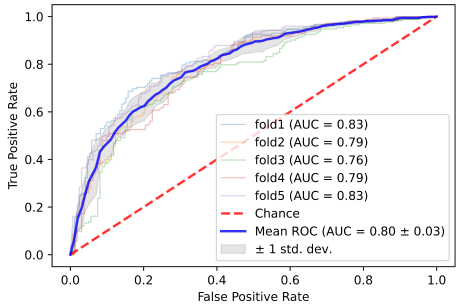
(b) Confusion matrix.

Figure B.5: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-NTL model.

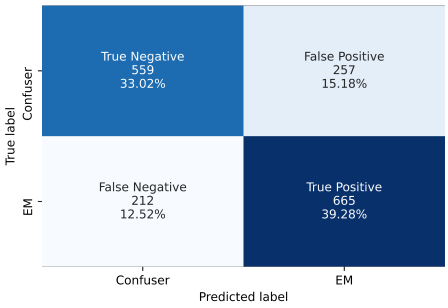
B.2.6 ResNet50-HAM-FFT

Table B.6: Five-fold cross-validation performance metrics of ResNet50-HAM-FFT model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR_+	LR_-	F1-Score	AUC
fold1	74.72	74.05	75.44	76.54	72.88	0.4946	0.4943	3.0151	0.3439	0.7527	0.8255
fold2	71.94	76.88	66.67	71.12	72.97	0.4382	0.4367	2.3064	0.3468	0.7389	0.7927
fold3	70.66	76.88	63.98	69.63	72.03	0.4126	0.4101	2.134	0.3614	0.7308	0.7596
fold4	70.36	74.57	65.84	70.11	70.67	0.4059	0.405	2.1828	0.3863	0.7227	0.7861
fold5	73.65	76.88	70.19	73.48	73.86	0.472	0.4715	2.5786	0.3294	0.7514	0.8254
average	72.27	75.85	68.42	72.18	72.48	0.4447	0.4435	2.4434	0.3536	0.7393	0.7979
std. deviation	1.69	1.27	4.05	2.55	1.08	0.0341	0.0347	0.3248	0.0193	0.0116	0.0251



(a) ROC curve.



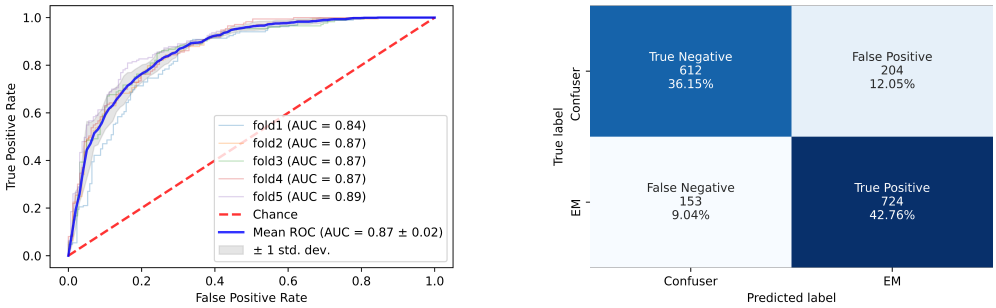
(b) Confusion matrix.

Figure B.6: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-HAM-FFT model.

B.2.7 ResNet50-IMG-WFT

Table B.7: Five-fold cross-validation performance metrics of ResNet50-IMG-WFT model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	76.97	83.24	70.18	75.12	79.47	0.54	0.5366	2.7911	0.2388	0.7897	0.8379
fold2	78.81	79.77	77.78	79.31	78.26	0.5756	0.5756	3.5896	0.2601	0.7954	0.8663
fold3	79.34	86.71	71.43	76.53	83.33	0.5899	0.5842	3.0347	0.1861	0.813	0.8667
fold4	78.14	83.82	72.05	76.32	80.56	0.5637	0.5607	2.9987	0.2246	0.7989	0.8744
fold5	81.44	79.19	83.85	84.05	78.95	0.6302	0.6291	4.9037	0.2482	0.8155	0.8875
average	78.94	82.55	75.06	78.27	80.11	0.5799	0.5772	3.4636	0.2316	0.8025	0.8666
std. deviation	1.48	2.77	5.11	3.2	1.77	0.03	0.0305	0.7671	0.0255	0.0101	0.0163



(a) ROC curve.

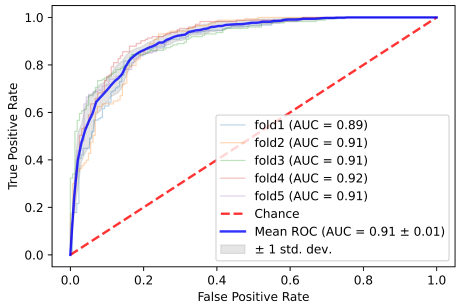
(b) Confusion matrix.

Figure B.7: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-IMG-WFT model.

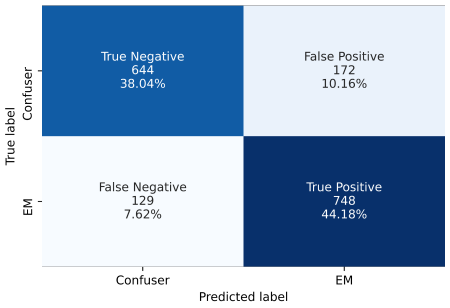
B.2.8 ResNet50-IMG-FFT

Table B.8: Five-fold cross-validation performance metrics of ResNet50-IMG-FFT model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	82.58	87.03	77.78	80.9	84.71	0.6521	0.6501	3.9162	0.1668	0.8385	0.895
fold2	82.09	82.08	82.1	83.04	81.1	0.6416	0.6415	4.5852	0.2183	0.8256	0.9051
fold3	80.24	88.44	71.43	76.88	85.19	0.6096	0.6021	3.0954	0.1618	0.8226	0.9114
fold4	84.43	82.08	86.96	87.12	81.87	0.6901	0.6889	6.2929	0.2061	0.8452	0.9234
fold5	81.74	86.71	76.4	79.79	84.25	0.6357	0.6331	3.6736	0.174	0.831	0.9101
average	82.22	85.27	78.93	81.55	83.42	0.6458	0.6431	4.3127	0.1854	0.8326	0.909
std. deviation	1.36	2.67	5.26	3.42	1.63	0.0262	0.028	1.0994	0.0226	0.0083	0.0092



(a) ROC curve.



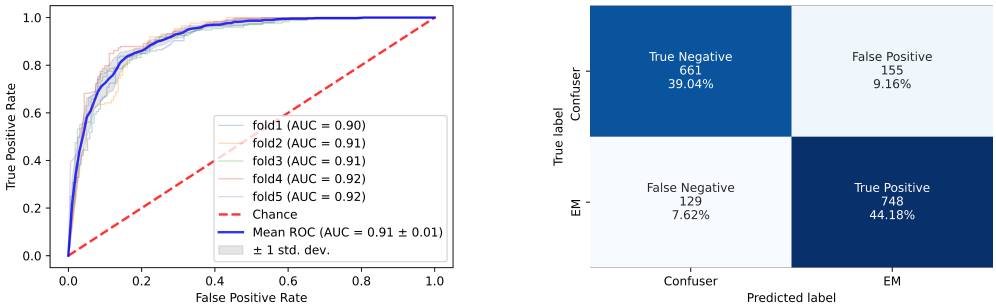
(b) Confusion matrix.

Figure B.8: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-IMG-FFT model.

B.2.9 ResNet50-IMG-FT141

Table B.9: Five-fold cross-validation performance metrics of ResNet50-IMG-FT141 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	81.74	85.41	77.78	80.61	83.12	0.6346	0.6334	3.8432	0.1876	0.8294	0.9003
fold2	84.18	86.13	82.1	83.71	84.71	0.6832	0.6829	4.8112	0.169	0.849	0.9102
fold3	82.34	83.24	81.37	82.76	81.88	0.6462	0.6462	4.4671	0.206	0.83	0.9091
fold4	84.43	89.02	79.5	82.35	87.07	0.6897	0.6873	4.343	0.1381	0.8556	0.9246
fold5	83.53	82.66	84.47	85.12	81.93	0.6709	0.6706	5.3232	0.2053	0.8387	0.9228
average	83.24	85.29	81.04	82.91	83.74	0.6649	0.6641	4.5575	0.1812	0.8405	0.9134
std. deviation	1.04	2.27	2.28	1.49	1.96	0.0212	0.021	0.493	0.0255	0.0104	0.0091



(a) ROC curve.

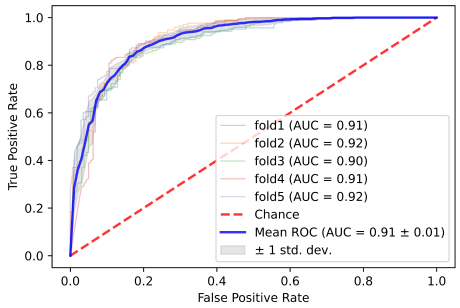
(b) Confusion matrix.

Figure B.9: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-IMG-FT141 model.

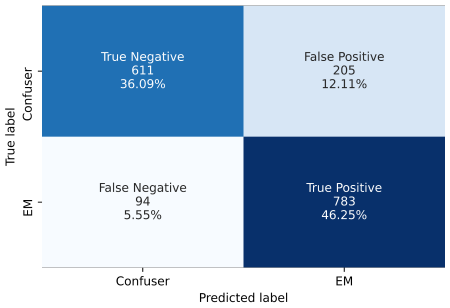
B.2.10 ResNet50-IMG-HAMFP-FT141

Table B.10: Five-fold cross-validation performance metrics of ResNet50-IMG-HAMFP-FT141 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	81.18	89.73	71.93	77.57	86.62	0.6291	0.6206	3.1966	0.1428	0.8321	0.9061
fold2	83.58	85.55	81.48	83.15	84.08	0.6713	0.671	4.6197	0.1774	0.8433	0.9189
fold3	80.24	89.6	70.19	76.35	86.26	0.6118	0.6017	3.0052	0.1482	0.8245	0.8982
fold4	82.04	93.06	70.19	77.03	90.4	0.6531	0.6374	3.1215	0.0988	0.8429	0.9096
fold5	84.73	88.44	80.75	83.15	86.67	0.695	0.6935	4.5931	0.1432	0.8571	0.9236
average	82.35	89.28	74.91	79.45	86.81	0.6521	0.6448	3.7072	0.1421	0.84	0.9113
std. deviation	1.62	2.42	5.11	3.05	2.03	0.0295	0.0333	0.7368	0.0251	0.0111	0.0091



(a) ROC curve.



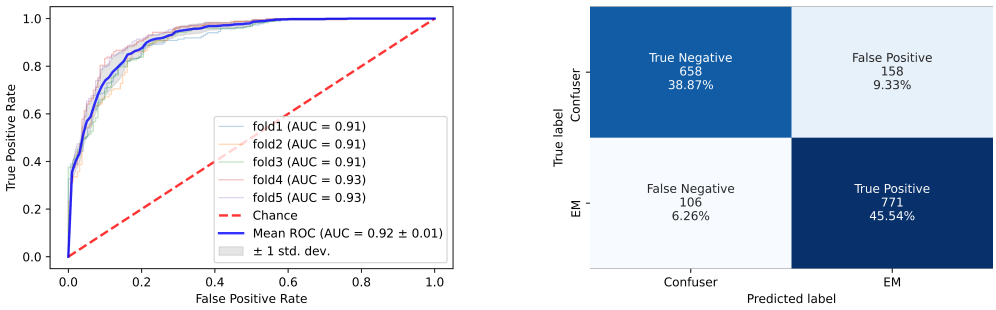
(b) Confusion matrix.

Figure B.10: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-IMG-HAMFP-FT141 model.

B.2.11 ResNet50-IMG-HAMPP-FT141/ ResNet50-141

Table B.11: Five-fold cross-validation performance metrics of ResNet50-IMG-HAMPP-FT141/ ResNet50-141 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	83.15	86.49	79.53	82.05	84.47	0.6627	0.6617	4.2255	0.1699	0.8421	0.9072
fold2	83.28	87.86	78.4	81.28	85.81	0.6667	0.6644	4.0667	0.1548	0.8444	0.9109
fold3	83.53	90.75	75.78	80.1	88.41	0.6751	0.6686	3.7464	0.1221	0.8509	0.9107
fold4	85.93	87.28	84.47	85.8	86.08	0.7181	0.718	5.621	0.1505	0.8653	0.9323
fold5	86.23	87.28	85.09	86.29	86.16	0.7241	0.7241	5.8553	0.1494	0.8678	0.9335
average	84.42	87.93	80.65	83.1	86.19	0.6893	0.6874	4.703	0.1493	0.8541	0.9189
std. deviation	1.36	1.47	3.59	2.49	1.27	0.0263	0.0277	0.8624	0.0155	0.0106	0.0115



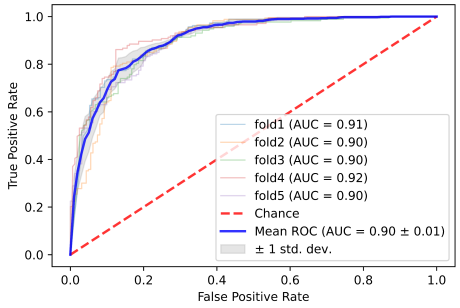
(a) ROC curve. (b) Confusion matrix.

Figure B.11: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-IMG-HAMPP-FT141/ ResNet50-141 model.

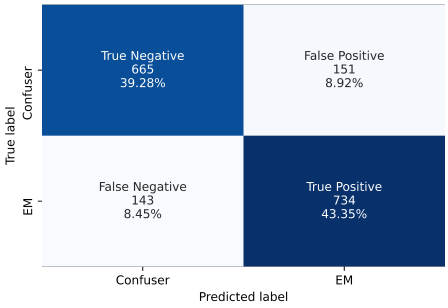
B.2.12 ResNet101-150

Table B.12: Five-fold cross-validation performance metrics of ResNet101-150 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	82.3	84.86	79.53	81.77	82.93	0.6455	0.645	4.1463	0.1903	0.8329	0.9076
fold2	82.69	80.92	84.57	84.85	80.59	0.6546	0.6539	5.2439	0.2256	0.8284	0.8982
fold3	82.34	85.55	78.88	81.32	83.55	0.6465	0.6456	4.051	0.1832	0.8338	0.8958
fold4	86.23	88.44	83.85	85.47	87.1	0.7243	0.7238	5.4764	0.1379	0.8693	0.9214
fold5	79.64	78.61	80.75	81.44	77.84	0.5932	0.5928	4.0828	0.2649	0.8	0.8992
average	82.64	83.68	81.52	82.97	82.4	0.6528	0.6522	4.6001	0.2004	0.8329	0.9044
std. deviation	2.1	3.49	2.29	1.8	3.09	0.0419	0.0418	0.6257	0.0427	0.022	0.0094



(a) ROC curve.



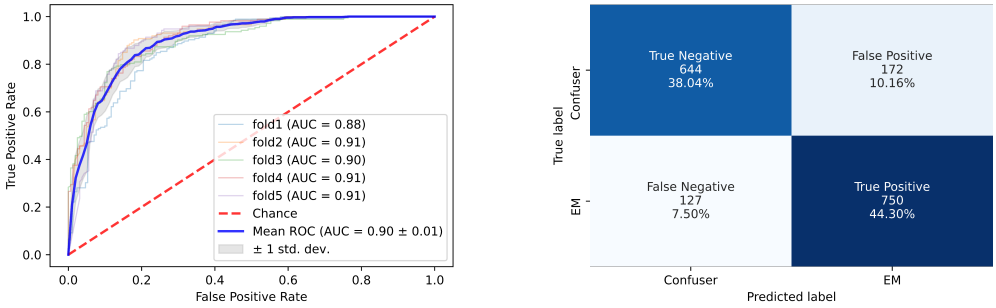
(b) Confusion matrix.

Figure B.12: Five-fold cross-validation ROC curve and confusion matrix of ResNet101-150 model.

B.2.13 ResNet50V2-105

Table B.13: Five-fold cross-validation performance metrics of ResNet50V2-105 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	80.06	84.86	74.85	78.5	82.05	0.6013	0.5992	3.3749	0.2022	0.8156	0.8761
fold2	85.07	84.97	85.19	85.96	84.15	0.7013	0.7013	5.7355	0.1764	0.8547	0.9103
fold3	80.24	90.75	68.94	75.85	87.4	0.6145	0.6014	2.9222	0.1341	0.8263	0.8999
fold4	81.74	80.35	83.23	83.73	79.76	0.6354	0.6348	4.7911	0.2361	0.8201	0.9128
fold5	84.73	86.71	82.61	84.27	85.26	0.6942	0.6939	4.9855	0.1609	0.8547	0.9076
average	82.37	85.53	78.96	81.66	83.72	0.6493	0.6461	4.3618	0.1819	0.8343	0.9013
std. deviation	2.15	3.35	6.13	3.83	2.63	0.0411	0.0439	1.0495	0.0349	0.017	0.0133



(a) ROC curve.

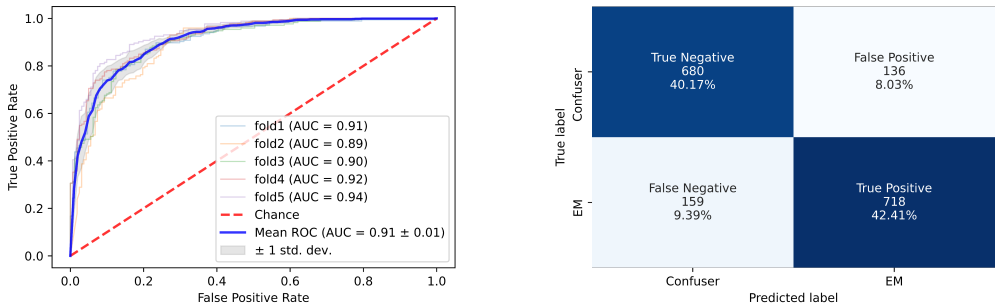
(b) Confusion matrix.

Figure B.13: Five-fold cross-validation ROC curve and confusion matrix of ResNet50V2-105 model.

B.2.14 ResNet101V2-233

Table B.14: Five-fold cross-validation performance metrics of ResNet101V2-233 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	82.3	80	84.8	85.06	79.67	0.6476	0.6464	5.2615	0.2359	0.8245	0.9067
fold2	78.81	73.99	83.95	83.12	75.14	0.581	0.5772	4.61	0.3098	0.7829	0.8927
fold3	82.63	88.44	76.4	80.1	86.01	0.6547	0.6509	3.747	0.1513	0.8407	0.9032
fold4	83.53	83.24	83.85	84.71	82.32	0.6706	0.6704	5.1543	0.1999	0.8397	0.9212
fold5	85.63	83.82	87.58	87.88	83.43	0.7135	0.7127	6.7471	0.1848	0.858	0.9354
average	82.58	81.9	83.32	84.17	81.31	0.6535	0.6515	5.104	0.2163	0.8292	0.9118
std. deviation	2.21	4.78	3.71	2.55	3.7	0.0429	0.0439	0.9811	0.0541	0.0254	0.0149



(a) ROC curve.

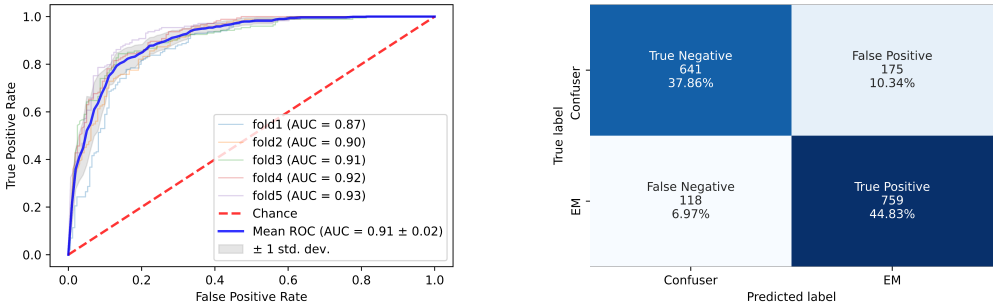
(b) Confusion matrix.

Figure B.14: Five-fold cross-validation ROC curve and confusion matrix of ResNet101V2-233 model.

B.2.15 InceptionV3-274

Table B.15: Five-fold cross-validation performance metrics of InceptionV3-274 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	80.06	84.86	74.85	78.5	82.05	0.6013	0.5992	3.3749	0.2022	0.8156	0.875
fold2	81.49	83.24	79.63	81.36	81.65	0.6293	0.6292	4.0862	0.2105	0.8229	0.8963
fold3	82.34	86.13	78.26	80.98	84	0.6468	0.6454	3.9618	0.1773	0.8347	0.907
fold4	83.53	89.6	77.02	80.73	87.32	0.6733	0.6689	3.8986	0.1351	0.8493	0.921
fold5	86.23	89.02	83.23	85.08	87.58	0.7246	0.7237	5.3081	0.132	0.8701	0.9268
average	82.73	86.57	78.6	81.33	84.52	0.6551	0.6533	4.1259	0.1714	0.8385	0.9052
std. deviation	2.08	2.42	2.8	2.12	2.52	0.0419	0.0419	0.639	0.0328	0.0195	0.0185



(a) ROC curve.

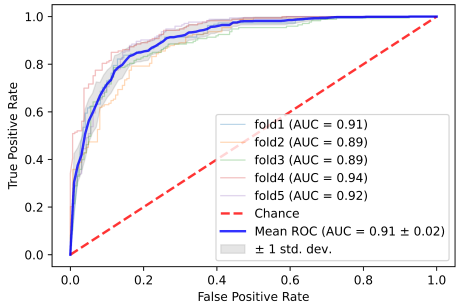
(b) Confusion matrix.

Figure B.15: Five-fold cross-validation ROC curve and confusion matrix of InceptionV3-102 model.

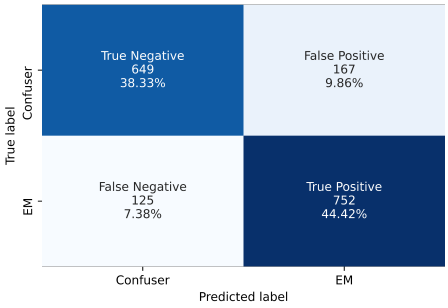
B.2.16 InceptionV4-327

Table B.16: Five-fold cross-validation performance metrics of InceptionV4-327 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	82.58	89.19	75.44	79.71	86.58	0.6545	0.6494	3.6313	0.1433	0.8418	0.9074
fold2	80.3	79.19	81.48	82.04	78.57	0.6064	0.606	4.2763	0.2554	0.8059	0.8907
fold3	81.44	83.82	78.88	81.01	81.94	0.6282	0.6278	3.9689	0.2052	0.8239	0.8884
fold4	85.03	86.13	83.85	85.14	84.91	0.7001	0.7001	5.3333	0.1654	0.8563	0.9393
fold5	84.43	90.17	78.26	81.68	88.11	0.6911	0.687	4.148	0.1256	0.8571	0.9202
average	82.76	85.7	79.58	81.92	84.02	0.6561	0.6541	4.2716	0.179	0.837	0.9092
std. deviation	1.78	3.96	2.87	1.8	3.41	0.0358	0.0353	0.5734	0.0465	0.0197	0.019



(a) ROC curve.



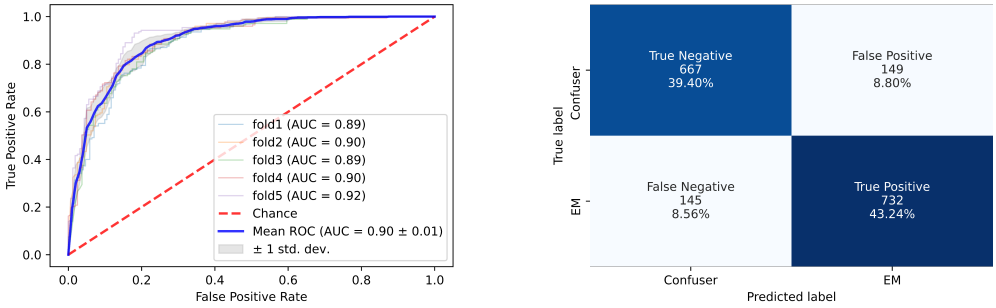
(b) Confusion matrix.

Figure B.16: Five-fold cross-validation ROC curve and confusion matrix of InceptionV4-327 model.

B.2.17 InceptionResNetV2-500

Table B.17: Five-fold cross-validation performance metrics of InceptionResNetV2-500 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	79.78	77.84	81.87	82.29	77.35	0.5967	0.5958	4.2936	0.2707	0.8	0.8868
fold2	82.39	81.5	83.33	83.93	80.84	0.648	0.6477	4.8902	0.222	0.827	0.9022
fold3	82.34	88.44	75.78	79.69	85.92	0.6491	0.6448	3.651	0.1526	0.8384	0.889
fold4	82.63	82.66	82.61	83.63	81.6	0.6524	0.6524	4.7529	0.2099	0.8314	0.9036
fold5	86.23	87.28	85.09	86.29	86.16	0.7241	0.7241	5.8553	0.1494	0.8678	0.9241
average	82.67	83.54	81.74	83.17	82.37	0.6541	0.653	4.6886	0.2009	0.8329	0.9011
std. deviation	2.06	3.88	3.16	2.16	3.32	0.0406	0.041	0.7264	0.0456	0.0218	0.0133



(a) ROC curve.

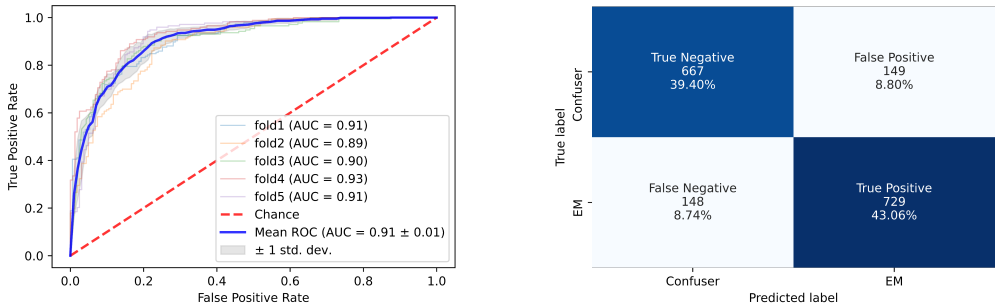
(b) Confusion matrix.

Figure B.17: Five-fold cross-validation ROC curve and confusion matrix of InceptionResNetV2-500 model.

B.2.18 Xception-118

Table B.18: Five-fold cross-validation performance metrics of Xception-118 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	80.9	80.54	81.29	82.32	79.43	0.6179	0.6177	4.3039	0.2394	0.8142	0.9061
fold2	78.81	76.3	81.48	81.48	76.3	0.5778	0.5766	4.1202	0.2909	0.7881	0.8861
fold3	83.83	90.17	77.02	80.83	87.94	0.6798	0.6748	3.9238	0.1276	0.8525	0.9029
fold4	85.93	87.86	83.85	85.39	86.54	0.7182	0.7179	5.4406	0.1448	0.8661	0.9314
fold5	82.93	80.92	85.09	85.37	80.59	0.6599	0.6589	5.4287	0.2242	0.8309	0.9139
average	82.48	83.16	81.75	83.08	82.16	0.6507	0.6492	4.6434	0.2054	0.8304	0.9081
std. deviation	2.45	5.1	2.76	1.94	4.4	0.0487	0.0484	0.6571	0.0609	0.0276	0.0148



(a) ROC curve.

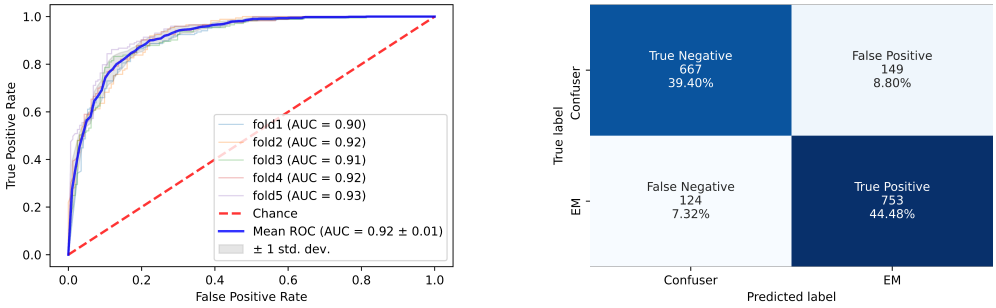
(b) Confusion matrix.

Figure B.18: Five-fold cross-validation ROC curve and confusion matrix of Xception-118 model.

B.2.19 DenseNet121-379

Table B.19: Five-fold cross-validation performance metrics of DenseNet121-379 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	83.71	86.49	80.7	82.9	84.66	0.6738	0.6731	4.4816	0.1675	0.8466	0.9035
fold2	82.69	83.24	82.1	83.24	82.1	0.6534	0.6534	4.6498	0.2042	0.8324	0.9159
fold3	84.43	87.86	80.75	83.06	86.09	0.6888	0.6875	4.5631	0.1503	0.8539	0.9094
fold4	83.23	84.39	81.99	83.43	83.02	0.6641	0.6641	4.6853	0.1904	0.8391	0.9178
fold5	85.33	87.28	83.23	84.83	85.9	0.7062	0.7059	5.2047	0.1528	0.8604	0.9325
average	83.88	85.85	81.75	83.49	84.35	0.6773	0.6768	4.7169	0.173	0.8465	0.9158
std. deviation	0.92	1.76	0.95	0.69	1.57	0.0186	0.0184	0.254	0.0211	0.01	0.0097



(a) ROC curve.

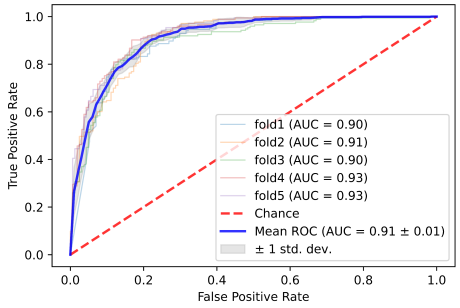
(b) Confusion matrix.

Figure B.19: Five-fold cross-validation ROC curve and confusion matrix of DenseNet121-379 model.

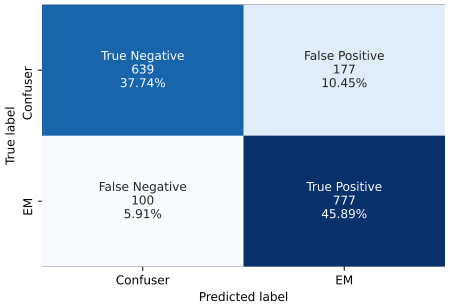
B.2.20 DenseNet169-395

Table B.20: Five-fold cross-validation performance metrics of DenseNet169-395 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	82.02	88.65	74.85	79.23	85.91	0.6431	0.6381	3.5253	0.1516	0.8367	0.8996
fold2	84.78	93.06	75.93	80.5	91.11	0.7029	0.6936	3.8657	0.0914	0.8633	0.9124
fold3	83.53	87.28	79.5	82.07	85.33	0.6709	0.6694	4.2584	0.16	0.8459	0.8964
fold4	85.33	91.33	78.88	82.29	89.44	0.7097	0.705	4.3247	0.1099	0.8658	0.9269
fold5	82.63	82.66	82.61	83.63	81.6	0.6524	0.6524	4.7529	0.2099	0.8314	0.9264
average	83.66	88.6	78.35	81.54	86.68	0.6758	0.6717	4.1454	0.1446	0.8486	0.9123
std. deviation	1.25	3.59	2.75	1.53	3.33	0.0265	0.0249	0.4187	0.0414	0.0138	0.0129



(a) ROC curve.



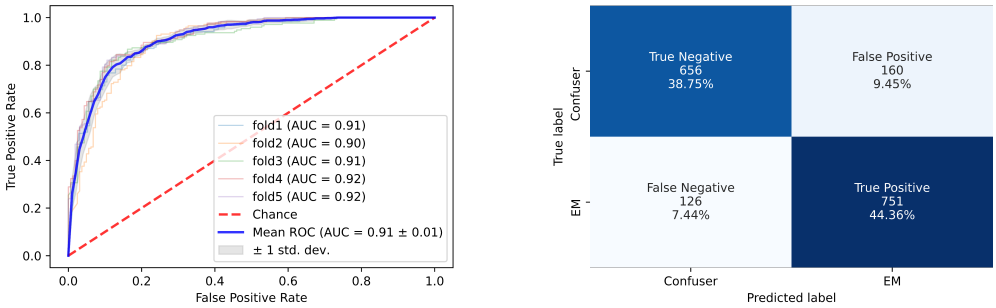
(b) Confusion matrix.

Figure B.20: Five-fold cross-validation ROC curve and confusion matrix of DenseNet169-395 model.

B.2.21 DenseNet201-561

Table B.21: Five-fold cross-validation performance metrics of DenseNet201-561 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	$LR+$	$LR-$	F1-Score	AUC
fold1	82.02	87.57	76.02	79.8	84.97	0.6418	0.6385	3.6522	0.1635	0.8351	0.9073
fold2	82.09	84.97	79.01	81.22	83.12	0.6416	0.6408	4.0486	0.1902	0.8305	0.9016
fold3	82.93	87.86	77.64	80.85	85.62	0.6598	0.6571	3.9294	0.1563	0.8421	0.9093
fold4	83.53	84.39	82.61	83.91	83.12	0.6702	0.6702	4.8526	0.1889	0.8415	0.9223
fold5	85.03	83.24	86.96	87.27	82.84	0.7015	0.7007	6.3815	0.1928	0.8521	0.922
average	83.12	85.61	80.45	82.61	83.93	0.663	0.6615	4.5729	0.1783	0.8403	0.9125
std. deviation	1.11	1.81	3.92	2.7	1.13	0.0221	0.0228	0.9885	0.0153	0.0073	0.0083



(a) ROC curve.

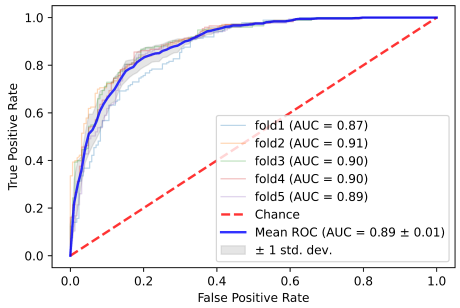
(b) Confusion matrix.

Figure B.21: Five-fold cross-validation ROC curve and confusion matrix of DenseNet201-561 model.

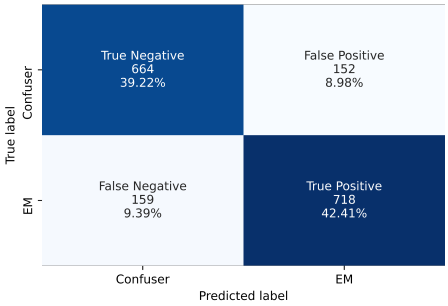
B.2.22 MobileNetV2-62

Table B.22: Five-fold cross-validation performance metrics of MobileNetV2-62 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR_+	LR_-	F1-Score	AUC
fold1	78.09	76.76	79.53	80.23	75.98	0.5625	0.5619	3.7501	0.2922	0.7845	0.8705
fold2	81.79	80.35	83.33	83.73	79.88	0.6365	0.6359	4.8208	0.2358	0.8201	0.9074
fold3	83.53	86.13	80.75	82.78	84.42	0.6703	0.6697	4.4731	0.1718	0.8442	0.9025
fold4	83.53	85.55	81.37	83.15	83.97	0.6702	0.6699	4.5911	0.1776	0.8433	0.9006
fold5	81.44	80.92	81.99	82.84	80	0.6288	0.6286	4.4927	0.2327	0.8187	0.8856
average	81.68	81.94	81.39	82.55	80.85	0.6337	0.6332	4.4256	0.222	0.8222	0.8933
std. deviation	1.99	3.49	1.26	1.21	3.09	0.0394	0.0395	0.3596	0.0441	0.0218	0.0135



(a) ROC curve.



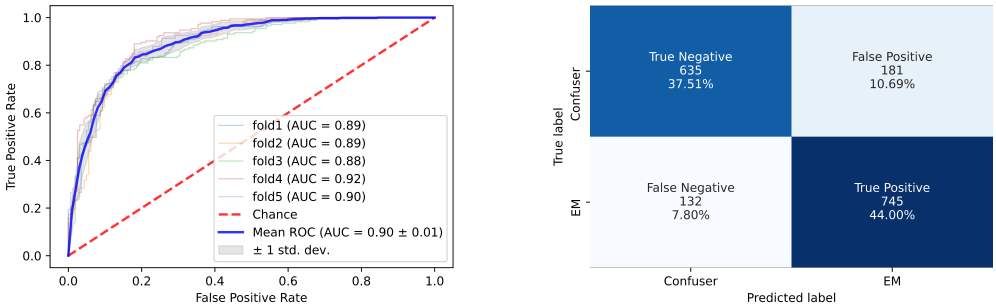
(b) Confusion matrix.

Figure B.22: Five-fold cross-validation ROC curve and confusion matrix of MobileNetV2-62 model.

B.2.23 MobileNetV3Small-182

Table B.23: Five-fold cross-validation performance metrics of MobileNetV3Small-182 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	79.78	86.49	72.51	77.29	83.22	0.5975	0.5929	3.1466	0.1864	0.8163	0.8898
fold2	81.49	79.77	83.33	83.64	79.41	0.6308	0.63	4.7861	0.2428	0.8166	0.8887
fold3	79.04	83.24	74.53	77.84	80.54	0.5807	0.5792	3.2686	0.2249	0.8045	0.8805
fold4	84.43	89.6	78.88	82.01	87.59	0.6903	0.6871	4.2426	0.1319	0.8564	0.917
fold5	82.93	85.55	80.12	82.22	83.77	0.6583	0.6577	4.3042	0.1804	0.8385	0.9041
average	81.53	84.93	77.87	80.6	82.91	0.6315	0.6294	3.9496	0.1933	0.8265	0.896
std. deviation	1.98	3.29	3.89	2.55	2.85	0.0398	0.04	0.6356	0.0386	0.0186	0.013



(a) ROC curve.

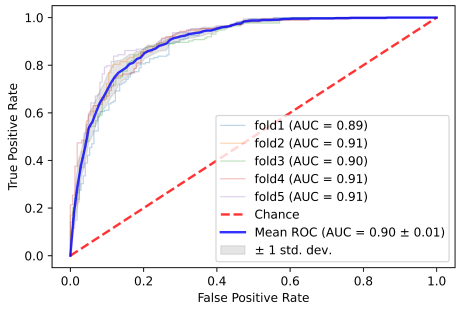
(b) Confusion matrix.

Figure B.23: Five-fold cross-validation ROC curve and confusion matrix of MobileNetV3Small-182 model.

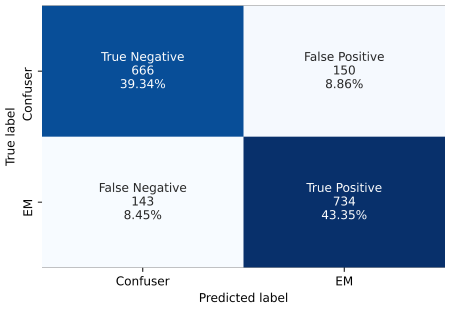
B.2.24 MobileNetV3Large-193

Table B.24: Five-fold cross-validation performance metrics of MobileNetV3Large-193 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	78.93	83.78	73.68	77.5	80.77	0.5787	0.5766	3.1838	0.2201	0.8052	0.8884
fold2	83.58	83.24	83.95	84.71	82.42	0.6716	0.6715	5.1863	0.1997	0.8397	0.9092
fold3	82.93	83.24	82.61	83.72	82.1	0.6583	0.6583	4.7861	0.2029	0.8348	0.8973
fold4	82.63	84.39	80.75	82.49	82.8	0.6521	0.6519	4.383	0.1933	0.8343	0.9073
fold5	85.63	83.82	87.58	87.88	83.43	0.7135	0.7127	6.7471	0.1848	0.858	0.915
average	82.74	83.69	81.71	83.26	82.3	0.6548	0.6542	4.8573	0.2002	0.8344	0.9034
std. deviation	2.17	0.43	4.6	3.39	0.89	0.0437	0.0442	1.1585	0.0117	0.017	0.0094



(a) ROC curve.



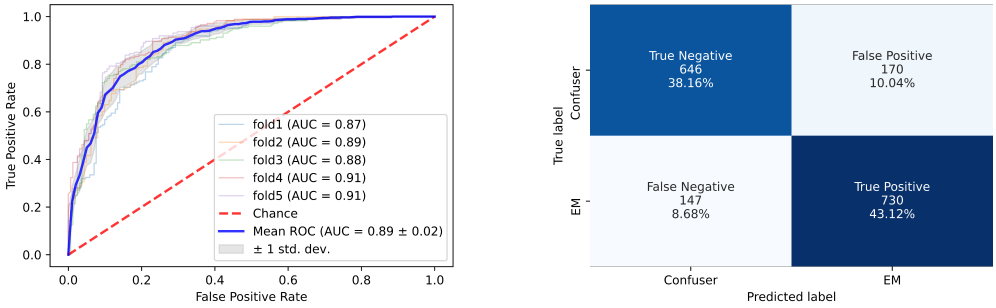
(b) Confusion matrix.

Figure B.24: Five-fold cross-validation ROC curve and confusion matrix of MobileNetV3Large-193 model.

B.2.25 NASNetMobile-617

Table B.25: Five-fold cross-validation performance metrics of NASNetMobile-617 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	79.49	85.95	72.51	77.18	82.67	0.5915	0.5873	3.127	0.1938	0.8133	0.8665
fold2	80.3	82.66	77.78	79.89	80.77	0.6055	0.6051	3.7197	0.223	0.8125	0.8855
fold3	80.84	80.92	80.75	81.87	79.75	0.6165	0.6164	4.2029	0.2362	0.814	0.8836
fold4	82.34	83.82	80.75	82.39	82.28	0.6461	0.646	4.353	0.2004	0.8309	0.9055
fold5	83.53	82.66	84.47	85.12	81.93	0.6709	0.6706	5.3232	0.2053	0.8387	0.9072
average	81.3	83.2	79.25	81.29	81.48	0.6261	0.6251	4.1452	0.2117	0.8219	0.8897
std. deviation	1.45	1.66	3.98	2.65	1.07	0.0287	0.0297	0.7283	0.0156	0.0108	0.0152



(a) ROC curve.

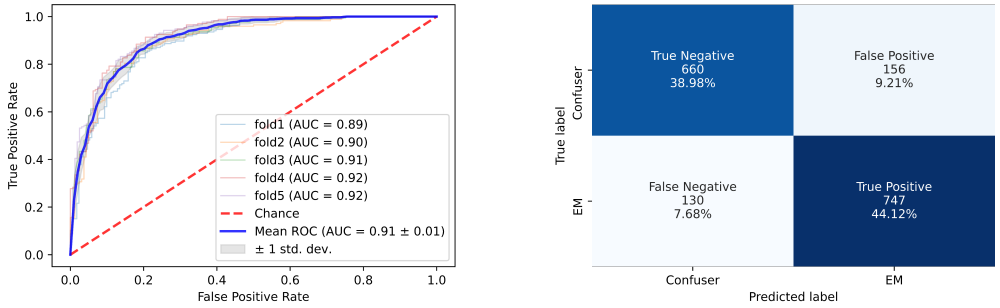
(b) Confusion matrix.

Figure B.25: Five-fold cross-validation ROC curve and confusion matrix of NASNetMobile-617 model.

B.2.26 EfficientNetB0-187

Table B.26: Five-fold cross-validation performance metrics of EfficientNetB0-187 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	81.18	82.7	79.53	81.38	80.95	0.6229	0.6228	4.0406	0.2175	0.8204	0.8927
fold2	82.39	79.19	85.8	85.62	79.43	0.6502	0.6483	5.5778	0.2425	0.8228	0.8984
fold3	83.53	89.02	77.64	81.05	86.81	0.6725	0.669	3.9811	0.1415	0.8485	0.9075
fold4	84.13	85.55	82.61	84.09	84.18	0.6821	0.682	4.9191	0.1749	0.8481	0.9239
fold5	84.43	89.6	78.88	82.01	87.59	0.6903	0.6871	4.2426	0.1319	0.8564	0.9243
average	83.13	85.21	80.89	82.83	83.79	0.6636	0.6618	4.5522	0.1817	0.8392	0.9094
std. deviation	1.2	3.91	2.95	1.75	3.19	0.0244	0.0237	0.6116	0.0427	0.0147	0.0129



(a) ROC curve.

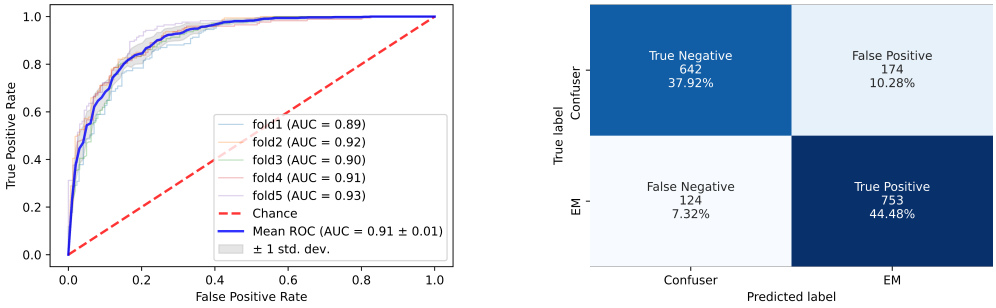
(b) Confusion matrix.

Figure B.26: Five-fold cross-validation ROC curve and confusion matrix of EfficientNetB0-187 model.

B.2.27 EfficientNetB1-308

Table B.27: Five-fold cross-validation performance metrics of EfficientNetB1-308 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	81.18	86.49	75.44	79.21	83.77	0.6245	0.6216	3.5212	0.1791	0.8269	0.8899
fold2	82.99	83.24	82.72	83.72	82.21	0.6594	0.6594	4.8159	0.2027	0.8348	0.9193
fold3	82.04	89.6	73.91	78.68	86.86	0.6452	0.6384	3.4345	0.1408	0.8378	0.9006
fold4	81.74	84.97	78.26	80.77	82.89	0.6345	0.6335	3.9087	0.192	0.8282	0.9063
fold5	84.13	84.97	83.23	84.48	83.75	0.6822	0.6822	5.0668	0.1806	0.8473	0.9278
average	82.42	85.85	78.71	81.37	83.9	0.6492	0.647	4.1494	0.179	0.835	0.9088
std. deviation	1.04	2.14	3.75	2.34	1.59	0.0202	0.0214	0.6707	0.0209	0.0074	0.0134



(a) ROC curve.

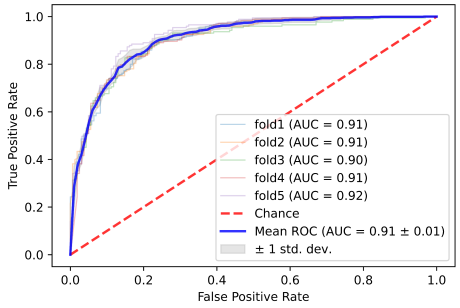
(b) Confusion matrix.

Figure B.27: Five-fold cross-validation ROC curve and confusion matrix of EfficientNetB1-308 model.

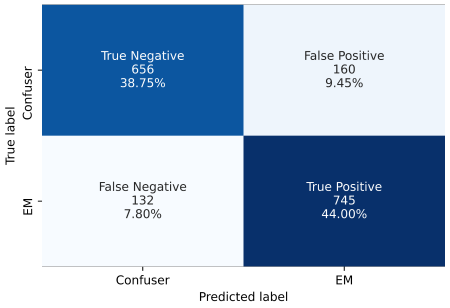
B.2.28 EfficientNetB2-316

Table B.28: Five-fold cross-validation performance metrics of EfficientNetB2-316 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	82.87	84.86	80.7	82.63	83.13	0.6567	0.6564	4.3975	0.1875	0.8373	0.9059
fold2	80.6	78.61	82.72	82.93	78.36	0.6131	0.6122	4.5483	0.2586	0.8071	0.906
fold3	82.63	87.28	77.64	80.75	85.03	0.6535	0.6512	3.9035	0.1638	0.8389	0.897
fold4	82.63	88.44	76.4	80.1	86.01	0.6547	0.6509	3.747	0.1513	0.8407	0.9062
fold5	85.03	85.55	84.47	85.55	84.47	0.7002	0.7002	5.5094	0.1711	0.8555	0.9224
average	82.75	84.95	80.39	82.39	83.4	0.6556	0.6542	4.4211	0.1865	0.8359	0.9075
std. deviation	1.4	3.41	3.02	1.91	2.69	0.0276	0.0279	0.6202	0.0379	0.0158	0.0082



(a) ROC curve.



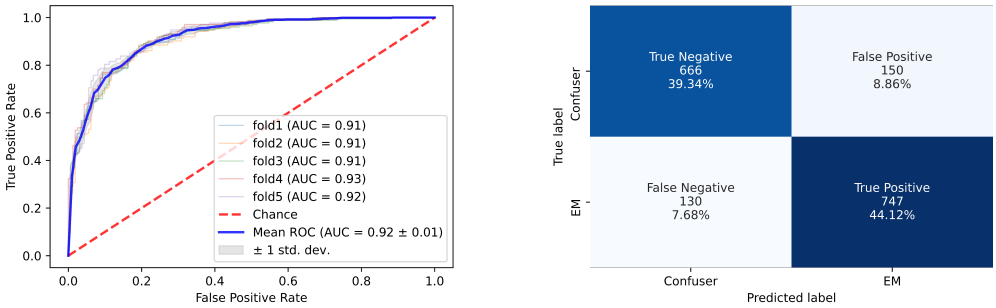
(b) Confusion matrix.

Figure B.28: Five-fold cross-validation ROC curve and confusion matrix of EfficientNetB2-316 model.

B.2.29 EfficientNetB3-194

Table B.29: Five-fold cross-validation performance metrics of EfficientNetB3-194 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	83.43	87.03	79.53	82.14	85	0.6685	0.6672	4.2519	0.1631	0.8451	0.9149
fold2	81.79	76.88	87.04	86.36	77.9	0.6409	0.6368	5.9306	0.2656	0.8135	0.9059
fold3	84.13	87.28	80.75	82.97	85.53	0.6826	0.6816	4.5331	0.1575	0.8507	0.9113
fold4	84.13	89.02	78.88	81.91	86.99	0.684	0.6812	4.2152	0.1392	0.8532	0.9253
fold5	83.83	85.55	81.99	83.62	84.08	0.6761	0.6759	4.7495	0.1763	0.8457	0.9239
average	83.46	85.15	81.64	83.4	83.9	0.6704	0.6685	4.7361	0.1803	0.8416	0.9163
std. deviation	0.87	4.28	2.9	1.6	3.14	0.0157	0.0167	0.6283	0.0443	0.0144	0.0074



(a) ROC curve.

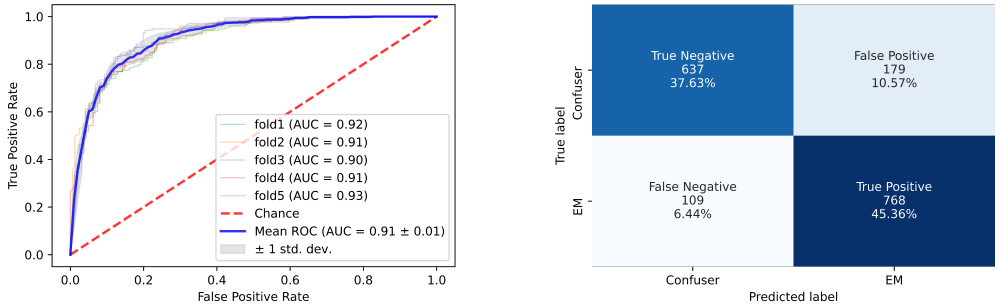
(b) Confusion matrix.

Figure B.29: Five-fold cross-validation ROC curve and confusion matrix of EfficientNetB3-194 model.

B.2.30 EfficientNetB5-444

Table B.30: Five-fold cross-validation performance metrics of EfficientNetB5-444 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	83.43	89.73	76.61	80.58	87.33	0.6712	0.6665	3.8359	0.1341	0.8491	0.9119
fold2	85.37	90.17	80.25	82.98	88.44	0.7092	0.7063	4.565	0.1225	0.8643	0.9261
fold3	81.74	87.28	75.78	79.47	84.72	0.6363	0.6329	3.6032	0.1678	0.832	0.8833
fold4	83.53	83.24	83.85	84.71	82.32	0.6706	0.6704	5.1543	0.1999	0.8397	0.9236
fold5	84.43	83.82	85.09	85.8	83.03	0.6887	0.6885	5.6226	0.1902	0.848	0.9242
average	83.7	86.85	80.32	82.71	85.17	0.6752	0.6729	4.5562	0.1629	0.8466	0.9138
std. deviation	1.21	2.89	3.73	2.39	2.38	0.024	0.0245	0.7645	0.0303	0.0108	0.0161



(a) ROC curve.

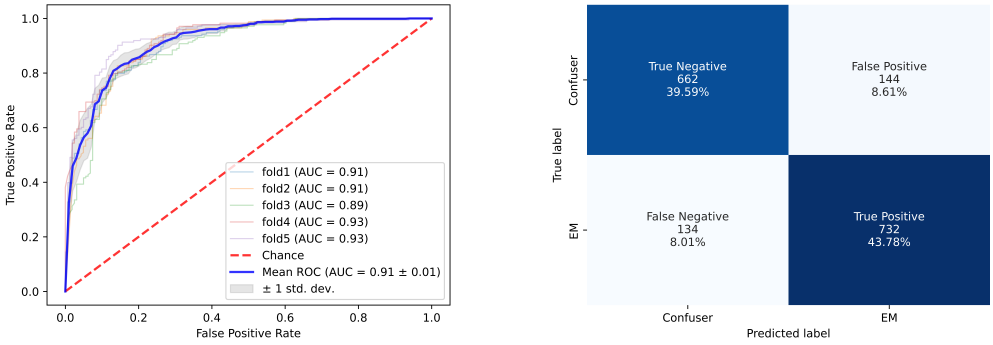
(b) Confusion matrix.

Figure B.30: Five-fold cross-validation ROC curve and confusion matrix of EfficientNetB5-444 model.

B.2.31 EfficientNetV2S-413

Table B.31: Five-fold cross-validation performance metrics of EfficientNetV2S-413 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	81.79	81.61	81.99	83.04	80.49	0.6356	0.6355	4.5307	0.2243	0.8232	0.9097
fold2	83.58	80.92	86.42	86.42	80.92	0.6734	0.672	5.959	0.2207	0.8358	0.9106
fold3	80.84	83.82	77.64	80.11	81.7	0.6163	0.6156	3.7484	0.2085	0.8192	0.8917
fold4	83.23	88.44	77.64	80.95	86.21	0.6662	0.6631	3.9552	0.1489	0.8453	0.927
fold5	87.43	87.86	86.96	87.86	86.96	0.7482	0.7482	6.736	0.1396	0.8786	0.9328
average	83.37	84.53	82.13	83.68	83.26	0.6679	0.6669	4.9859	0.1884	0.8404	0.9144
std. deviation	2.26	3.11	4.05	3.02	2.76	0.0451	0.0453	1.1671	0.0365	0.0212	0.0145



(a) ROC curve.

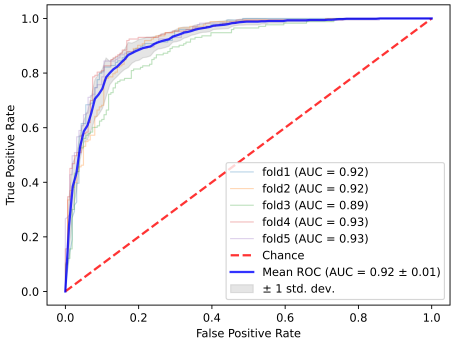
(b) Confusion matrix.

Figure B.31: Five-fold cross-validation ROC curve and confusion matrix of EfficientNetV2S-413 model.

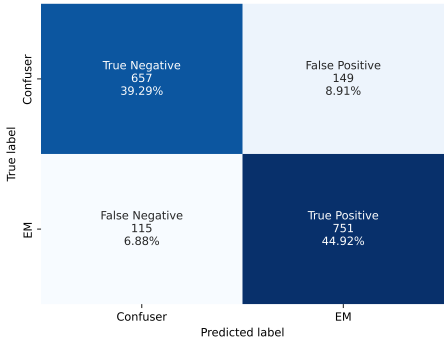
B.2.32 ConvNeXtTiny-120

Table B.32: Five-fold cross-validation performance metrics of ConvNeXtTiny-120 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	84.78	90.23	78.88	82.2	88.19	0.6975	0.6939	4.2727	0.1239	0.8603	0.9227
fold2	83.88	83.82	83.95	84.8	82.93	0.6774	0.6774	5.2223	0.1928	0.843	0.9188
fold3	80.54	81.5	79.5	81.03	80	0.6102	0.6102	3.9764	0.2327	0.8127	0.8897
fold4	86.83	90.17	83.23	85.25	88.74	0.7369	0.7356	5.377	0.1181	0.8764	0.9319
fold5	85.03	87.86	81.99	83.98	86.27	0.7005	0.6997	4.8778	0.1481	0.8588	0.9283
average	84.21	86.72	81.51	83.45	85.23	0.6845	0.6834	4.7452	0.1631	0.8502	0.9183
std. deviation	2.07	3.5	2	1.6	3.31	0.0418	0.0412	0.5401	0.0436	0.0215	0.015



(a) ROC curve.



(b) Confusion matrix.

Figure B.32: Five-fold cross-validation ROC curve and confusion matrix of ConvNeXtTiny-120 model.

C Appendices for Chapter 4

C.1 Online Resources

The data that support the findings of this study are openly available at the elicitation page of DAPPEM website[1]. The data can be accessed by clicking on "*Download Research Data*" link. Available files are listed below:

- *Elicited Probability.xlxs* : contains the 1536 possible cases and the corresponding elicited EM probability estimates.
- *EM Decision Tree.png* : detailed version of the pruned decision tree described in Section 4.2.3.4.
- *FCA Context Files*: this directory contains FCA lattice context files for different probability groups of EM cases. The context files can be explored using FCA software like The Concept Explorer [182] or Formal Concept Analysis Research Toolbox (FCART)[112]. Following is the list of files inside this directory:
  - *Group (1).cxt* : cases with probability $[0, 0.1)$.
  - *Group (2).cxt* : cases with probability $[0.1, 0.2)$.
  - *Group (4).cxt* : cases with probability $[0.3, 0.4)$.
  - *Group (5).cxt* : cases with probability $[0.4, 0.5)$.
  - *Group (6).cxt* : cases with probability $[0.5, 0.6)$.
  - *Group (7).cxt* : cases with probability $[0.6, 0.7)$.
  - *Group (8).cxt* : cases with probability $[0.7, 0.8)$.
  - *Group (9).cxt* : cases with probability $[0.8, 0.9)$.
  - *Group (10).cxt* : cases with probability $[0.9, 1]$.

[1] https://dappem.limos.fr/elicitation.html (visited on 02/20/2023)

D Appendices for Chapter 5

D.1 EMScan: A Mobile Application for Assisting With Early Lyme Disease Diagnosis

Figure D.1 shows the overall workflow of the EMScan mobile application. First, the user takes a photo of the skin lesion using the mobile camera. Second, the application automatically detects and crops the skin lesion which can be also manually adjusted by the user. Third, patient data related to the skin lesion is acquired through a series of questions and answers. Fourth, the lesion image is analyzed by a CNN image classifier, and the patient data is analyzed by the elicited statistical model to obtain two probabilities. Finally, a disease prediction with suggestion is provided to the user based on the analysis of skin lesion image and patient data. The application also provides the user with necessary details about Lyme disease and collects data with the patient's consent for research purposes.

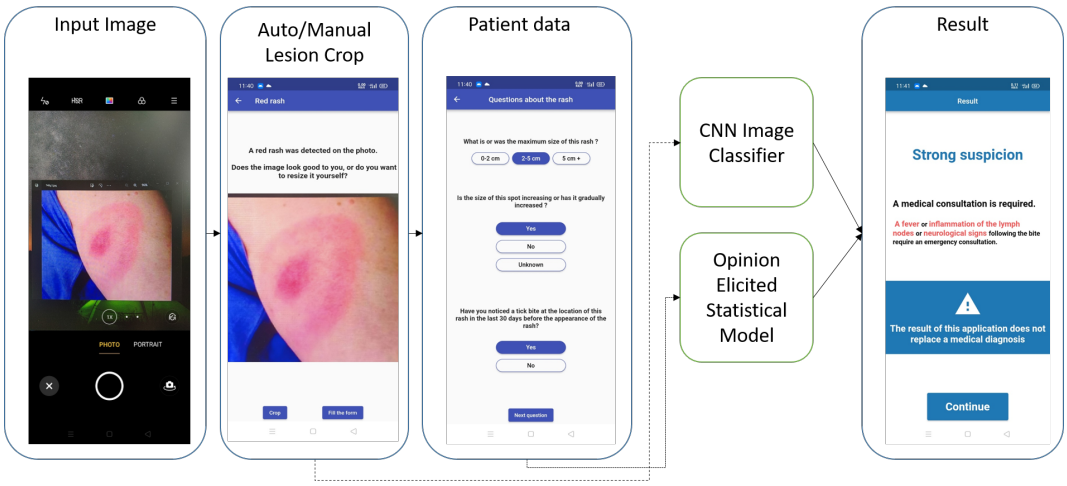


Figure D.1: EMScan application workflow.

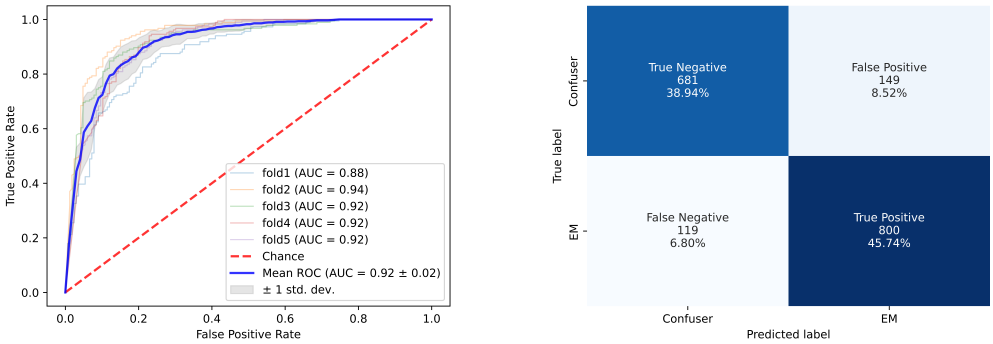
D.2 Supplementary Data for Custom Architecture

This section provides detailed five-fold cross validation results of the custom architecture and the ResNet50-141 model for the updated Lyme dataset.

D.2.1 Custom Architecture

Table D.1: Five-fold cross-validation performance metrics of custom model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR+	LR−	F1-Score	AUC
fold1	80.57	82.07	78.92	81.18	79.88	0.6102	0.6102	3.8922	0.2273	0.8162	0.8783
fold2	88.29	89.67	86.75	88.24	88.34	0.765	0.7649	6.7663	0.119	0.8895	0.9447
fold3	86.29	85.87	86.75	87.78	84.71	0.7255	0.7253	6.4792	0.1629	0.8681	0.9221
fold4	84.86	91.85	77.11	81.64	89.51	0.7005	0.6943	4.0123	0.1057	0.8645	0.9151
fold5	83.38	85.79	80.72	83.07	83.75	0.6667	0.6663	4.4505	0.176	0.8441	0.9152
average	84.68	87.05	82.05	84.38	85.24	0.6936	0.6922	5.1201	0.1582	0.8565	0.9151
std. deviation	2.62	3.4	4	3.03	3.44	0.0526	0.0525	1.2442	0.0434	0.0248	0.0214



(a) ROC curve.

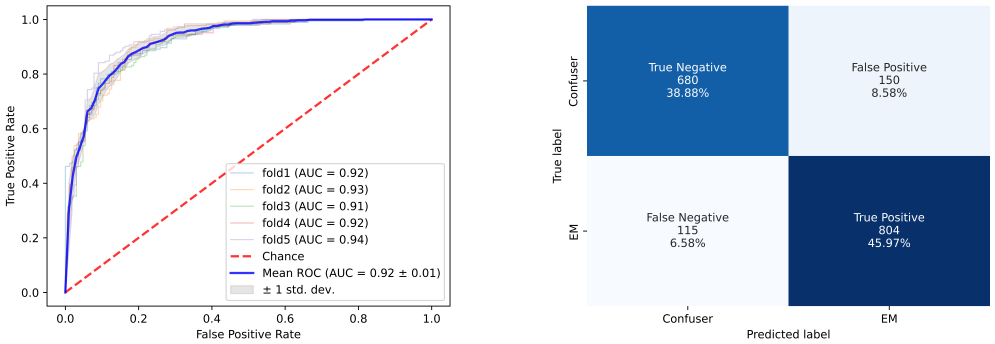
(b) Confusion matrix.

Figure D.2: Five-fold cross-validation ROC curve and confusion matrix of custom model.

D.2.2 ResNet50-141

Table D.2: Five-fold cross-validation performance metrics of VGG19-13 model.

	Metric										
Fold	Accuracy	Sensitivity	Specificity	Precision	NPV	MCC	Kappa	LR^+	LR^-	F1-Score	AUC
fold1	84.57	85.33	83.73	85.33	83.73	0.6906	0.6906	5.246	0.1752	0.8533	0.9203
fold2	85.43	88.04	82.53	84.82	86.16	0.7078	0.7072	5.0397	0.1449	0.864	0.9259
fold3	83.14	86.41	79.52	82.38	84.08	0.6619	0.6611	4.219	0.1709	0.8435	0.9148
fold4	84.29	86.96	81.33	83.77	84.91	0.6848	0.6842	4.6564	0.1604	0.8533	0.9194
fold5	86.82	90.71	82.53	85.13	88.96	0.7366	0.7349	5.1924	0.1126	0.8783	0.935
average	84.85	87.49	81.93	84.29	85.57	0.6963	0.6956	4.8707	0.1528	0.8585	0.9231
std. deviation	1.23	1.83	1.42	1.09	1.89	0.0249	0.0246	0.3856	0.0227	0.0118	0.0069



(a) ROC curve.

(b) Confusion matrix.

Figure D.3: Five-fold cross-validation ROC curve and confusion matrix of ResNet50-141 model.

Acronyms

AI	Artificial Intelligence
AUC	Area Under the Receiver Operating Characteristic (ROC) Curve
CD	Critical Difference
CF-CHU	Clermont-Ferrand University Hospital Center
CNN	Convolutional Neural Network
CRMVT	Centres de Référence des Maladies Vectorielles liées aux Tiques
DAPPEM	Développement d'une APPlication d'identification des Erythèmes Migrants à partir de photographies
EM	Erythema Migrans
FCA	Formal Concept Analysis
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GAP	Global Average Pooling
GELU	Gaussian Error Linear Unit
Grad-CAM	Gradient-weighted Class Activation Mapping
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
INRAE	Institut national de recherche pour l'agriculture, l'alimentation et l'environnement
ISIC	International Skin Imaging Collaboration
KDE	Kernel Density Estimation
LR	Likelihood Ratio
MCC	Matthews Correlation Coefficient
MLP	Multilayer Perceptron
NAS	Neural Architecture Search
NPV	Negative Predictive Value
PPV	Positive Predictive Value
ReLU	Rectified Linear Unit
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

List of Figures

List of Tables

List of Algorithms

Bibliography

1. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. "TensorFlow: A system for large-scale machine learning". *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, 2016, pp. 265–283. arXiv: 1605.08695.
2. H. Akaike. "A New Look at the Statistical Model Identification". *IEEE Transactions on Automatic Control* 19:6, 1974, pp. 716–723. ISSN: 15582523. DOI: 10.1109/TAC.1974.1100705.
3. C. Akyel and N. Arıcı. "LinkNet-B7: Noise Removal and Lesion Segmentation in Images of Skin Cancer". *Mathematics* 10:5, 2022, p. 736. ISSN: 2227-7390. DOI: 10.3390/math10050736.
4. L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, J. Santamaría, Y. Duan, and S. R. Oleiwi. "Towards a Better Understanding of Transfer Learning for Medical Imaging: A Case Study". *Applied Sciences* 10:13, 2020. ISSN: 2076-3417. DOI: 10.3390/app10134523.
5. M. Attia, M. Hossny, H. Zhou, S. Nahavandi, H. Asadi, and A. Yazdabadi. "Realistic hair simulator for skin lesion images: A novel benchemarking tool". *Artificial Intelligence in Medicine* 108, 2020, p. 101933. ISSN: 18732860. DOI: 10.1016/j.artmed.2020.101933.
6. S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, V. Natarajan, and M. Norouzi. "Big Self-Supervised Models Advance Medical Image Classification". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 3458–3468. DOI: 10.1109/ICCV48922.2021.00346.
7. J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer Normalization", 2016. DOI: 10.48550/arxiv.1607.06450. arXiv: 1607.06450.
8. J. Baron, B. A. Mellers, P. E. Tetlock, E. Stone, and L. H. Ungar. "Two reasons to make aggregated probability forecasts more extreme". *Decision Analysis* 11:2, 2014, pp. 133–145. DOI: 10.1287/deca.2014.0293.

9. A. Ben Hamida, M. Devanne, J. Weber, C. Truntzer, V. Derangère, F. Ghiringhelli, G. Forestier, and C. Wemmert. "Deep learning for colon cancer histopathological images analysis". *Computers in Biology and Medicine* 136, 2021, p. 104730. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2021.104730.

10. J. Berglund, R. Eitrem, K. Ornstein, A. Lindberg, Å. Ringnér, H. Elmrud, M. Carlsson, A. Runehagen, C. Svanborg, and R. Norrby. "An Epidemiologic Study of Lyme Disease in Southern Sweden". *New England Journal of Medicine* 333:20, 1995, pp. 1319–1324. ISSN: 0028-4793. DOI: 10.1056/nejm199511163332004.

11. M. D. Bloice, P. M. Roth, and A. Holzinger. "Biomedical image augmentation using Augmentor". *Bioinformatics* 35:21, 2019. Ed. by R. Murphy, pp. 4522–4524. ISSN: 14602059. DOI: 10.1093/bioinformatics/btz259.

12. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984. ISBN: 978-0-412-04841-8.

13. T. J. Brinker et al. "Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task". *European Journal of Cancer* 113, 2019, pp. 47–54. ISSN: 18790852. DOI: 10.1016/j.ejca.2019.04.001.

14. L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone. "Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays". *Computer Methods and Programs in Biomedicine* 196, 2020, p. 105608. ISSN: 18727565. DOI: 10.1016/j.cmpb.2020.105608.

15. P. Burlina, N. Joshi, E. Ng, S. Billings, A. Rebman, and J. Aucott. "Skin Image Analysis for Erythema Migrans Detection and Automated Lyme Disease Referral". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11041 LNCS. 2018, pp. 244–251. ISBN: 9783030012007. DOI: 10.1007/978-3-030-01201-4_26.

16. P. M. Burlina, N. J. Joshi, P. A. Mathew, W. Paul, A. W. Rebman, and J. N. Aucott. "AI-based detection of erythema migrans and disambiguation against other skin lesions". *Computers in Biology and Medicine* 125, 2020, p. 103977. ISSN: 18790534. DOI: 10.1016/j.compbiomed.2020.103977.

17. C. J. Cadham, M. Knoll, L. M. Sánchez-Romero, K. M. Cummings, C. E. Douglas, A. Liber, D. Mendez, R. Meza, R. Mistry, A. Sertkaya, N. Travis, and D. T. Levy. "The Use of Expert Elicitation among Computational Modeling Studies in Health Research: A Systematic Review". *Medical Decision Making* 42:5, 2022, pp. 684–703. DOI: 10.1177/0272989X211053794.

18. L. Cai, Z. Wang, H. Gao, D. Shen, and S. Ji. "Deep Adversarial Learning for Multi-Modality Missing Data Completion". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. Association for Computing Machinery, London, United Kingdom, 2018, pp. 1158–1166. ISBN: 9781450355520. DOI: 10.1145/3219819.3219963.

19. S. Candemir, R. D. White, M. Demirer, V. Gupta, M. T. Bigelow, L. M. Prevedello, and B. S. Erdal. "Automated coronary artery atherosclerosis detection and weakly supervised localization on coronary CT angiography with a deep 3-dimensional convolutional neural network". *Computerized Medical Imaging and Graphics* 83, 2020, p. 101721. ISSN: 18790771. DOI: 10.1016/j.compmedimag.2020.101721.

20. D. B. Carr. "Graphics in the Physical Sciences". *Encyclopedia of Physical Science and Technology*, 2003, pp. 1–14. DOI: 10.1016/B0-12-227410-5/00297-0.

21. J. Chen and A. Zhang. "HGMF: Heterogeneous Graph-Based Fusion for Multimodal Data with Incompleteness". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '20. Association for Computing Machinery, Virtual Event, CA, USA, 2020, pp. 1295–1305. ISBN: 9781450379984. DOI: 10.1145/3394486.3403182.

22. Q. Chen, M. Li, C. Chen, P. Zhou, X. Lv, and C. Chen. "MDFNet: application of multimodal fusion method based on skin image and clinical data to skin cancer classification". *Journal of Cancer Research and Clinical Oncology*, 2022. ISSN: 14321335. DOI: 10.1007/s00432-022-04180-1.

23. D. Chicco and G. Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". *BMC Genomics* 21:1, 2020, pp. 1–13. ISSN: 1471-2164. DOI: 10.1186/s12864-019-6413-7.

24. F. Chollet. "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Vol. 2017-Janua. Institute of Electrical and Electronics Engineers Inc., 2017, pp. 1800–1807. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.195. arXiv: 1610.02357v3.

25. R. T. Clemen and R. L. Winkler. "Combining Probability Distributions From Experts in Risk Analysis". *Risk Analysis* 19:2, 1999, pp. 187–203. ISSN: 0272-4332. DOI: 10.1111/j.1539-6924.1999.tb00399.x.

26. N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, H. Kittler, and A. Halpern. "Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the

International Skin Imaging Collaboration (ISIC)", 2019. DOI: 10.48550/arxiv.1902.03368. arXiv: 1902.03368.

27. CRMVT. *Centre de Référence des Maladies Vectorielles liées aux Tiques - CRMVT*. URL: https://crmvt.fr/ (visited on 02/20/2023).

28. C. v. Csefalvay. *Ventral Visual Stream*. URL: https://medium.com/starschema-blog/growing-neural-gas-models-theory-and-practice-b63e5bbe058d (visited on 02/20/2023).

29. E. Čuk, M. Gams, M. Možek, F. Strle, V. Maraspin Čarman, and J. F. Tasič. "Supervised visual system for recognition of erythema migrans, an early skin manifestation of lyme borreliosis". *Strojniski Vestnik/Journal of Mechanical Engineering* 60:2, 2014, pp. 115–123. ISSN: 00392480. DOI: 10.5545/sv-jme.2013.1046.

30. M. Czepita and A. Fabijańska. "Image processing pipeline for the detection of blood flow through retinal vessels with subpixel accuracy in fundus images". *Computer Methods and Programs in Biomedicine* 208, 2021, p. 106240. ISSN: 18727565. DOI: 10.1016/j.cmpb.2021.106240.

31. S. Dadsetan, M. Hejrati, S. Wu, and S. Hashemifar. *Cross-Domain Self-Supervised Deep Learning for Robust Alzheimer's Disease Progression Modeling*. 2022. DOI: 10.48550/ARXIV.2211.08559.

32. A. P. Dempster, N. M. Laird, and D. B. Rubin. " Maximum Likelihood from Incomplete Data Via the EM Algorithm ". *Journal of the Royal Statistical Society: Series B (Methodological)* 39:1, 1977, pp. 1–22. ISSN: 00359246. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

33. J. Demšar. *Statistical comparisons of classifiers over multiple data sets*. Technical report. 2006, pp. 1–30.

34. "Dermatologist-level classification of skin cancer with deep neural networks". *Nature* 542:7639, 2017, pp. 115–118. ISSN: 14764687. DOI: 10.1038/nature21056.

35. V. Dumoulin. *Dilated Convolution*. URL: https://github.com/vdumoulin (visited on 02/20/2023).

36. M. El Adoui, S. Drisis, and M. Benjelloun. "Multi-input deep learning architecture for predicting breast tumor response to chemotherapy using quantitative MR images". *International Journal of Computer Assisted Radiology and Surgery* 15:9, 2020, pp. 1491–1500. ISSN: 18616429. DOI: 10.1007/s11548-020-02209-9.

37. C. Eldin, A. Raffetin, K. Bouiller, Y. Hansmann, F. Roblot, D. Raoult, and P. Parola. "Review of European and American guidelines for the diagnosis of Lyme borreliosis". *Medecine et Maladies Infectieuses* 49:2, 2019, pp. 121–132. ISSN: 17696690. DOI: 10.1016/j.medmal.2018.11.011.

38. *Erythema migrans*. URL: https://commons.wikimedia.org/wiki/Category:Erythema_migrans (visited on 02/20/2023).

39. H. Feng, J. Berk-Krauss, P. W. Feng, and J. A. Stein. "Comparison of dermatologist density between urban and rural counties in the United States". eng. *JAMA Dermatology* 154:11, 2018, pp. 1265–1271. ISSN: 21686068. DOI: 10.1001/jamadermatol.2018.3022.

40. Y. Frendo, J. de Goër de Herve, S. I. Hossain, D. Martineau, I. Lebert, O. Lesens, and E. Mephu Nguifo. "EMScan: A Mobile Application for Early Lyme Disease Diagnosis". In: *European Conference on Computer Vision ECCV*. 2022.

41. M. Friedman. "A Comparison of Alternative Tests of Significance for the Problem of m Rankings". *The Annals of Mathematical Statistics* 11:1, 1940, pp. 86–92. ISSN: 0003-4851. DOI: 10.1214/aoms/1177731944.

42. A. Gallucci. "Data set of multi-source dermatoscopic images of skin hair for skin lesions", 2020. DOI: 10.4121/uuid:9ed94e25-8b74-4807-b84a-2c54ec9d96f0.

43. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

44. M. Graziani, V. Andrearczyk, and H. Müller. "Visualizing and Interpreting Feature Reuse of Pretrained CNNs for Histopathology". In: *IMVIP 2019 : Irish Machine Vision and Image Processing Conference Proceedings*. Irish Pattern Recognition and Classification Society. 2019, pp. 1–4.

45. Y. Gu, Z. Ge, C. P. Bonnington, and J. Zhou. "Progressive Transfer Learning and Adversarial Domain Adaptation for Cross-Domain Skin Disease Classification". *IEEE Journal of Biomedical and Health Informatics* 24:5, 2020, pp. 1379–1393. DOI: 10.1109/JBHI.2019.2942429.

46. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. "On calibration of modern neural networks". *34th International Conference on Machine Learning, ICML 2017* 3, 2017, pp. 2130–2143. arXiv: 1706.04599.

47. M. Gupta, C. Das, A. Roy, P. Gupta, G. R. Pillai, and K. Patole. "Region of Interest Identification for Cervical Cancer Images". In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. 2020, pp. 1293–1296. DOI: 10.1109/ISBI45749.2020.9098587.

48. H. A. Haenssle, C. Fink, R. Schneiderbauer, F. Toberer, T. Buhl, A. Blum, A. Kalloo, A. Ben Hadj Hassen, L. Thomas, A. Enk, L. Uhlmann, C. Alt, M. Arenbergerova, R. Bakos, A. Baltzer, I. Bertlich, A. Blum, T. Bokor-Billmann, J. Bowling, N. Braghiroli, R. Braun, K. Buder-Bakhaya, T. Buhl, H. Cabo, L. Cabrijan, N. Cevic, A. Classen, D. Deltgen, C. Fink, I. Georgieva, L. E. Hakim-Meibodi, S. Hanner, F. Hartmann, J. Hartmann, G. Haus, E. Hoxha, R. Karls, H. Koga,

J. Kreusch, A. Lallas, P. Majenka, A. Marghoob, C. Massone, L. Mekokishvili, D. Mestel, V. Meyer, A. Neuberger, K. Nielsen, M. Oliviero, R. Pampena, J. Paoli, E. Pawlik, B. Rao, A. Rendon, T. Russo, A. Sadek, K. Samhaber, R. Schneiderbauer, A. Schweizer, F. Toberer, L. Trennheuser, L. Vlahova, A. Wald, J. Winkler, P. Wo¨lbing, and I. Zalaudek. "Man against Machine: Diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists". *Annals of Oncology* 29:8, 2018, pp. 1836–1842. ISSN: 15698041. DOI: 10.1093/annonc/mdy166.

49. S. S. Han, G. H. Park, W. Lim, M. S. Kim, J. I. Na, I. Park, and g. E. Chang. "Deep neural networks show an equivalent and often superior performance to dermatologists in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network". *PLoS ONE* 13:1, 2018. ISSN: 19326203. DOI: 10.1371/journal.pone.0191493.

50. K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2016-Decem. IEEE, 2016, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

51. K. He, X. Zhang, S. Ren, and J. Sun. "Identity mappings in deep residual networks". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9908 LNCS, 2016, pp. 630–645. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46493-0_38. arXiv: 1603.05027.

52. X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P. Xie. "Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans". *medRxiv*, 2020. DOI: 10.1101/2020.04.13.20063941. eprint: https://www.medrxiv.org/content/early/2020/04/17/2020.04.13.20063941.full.pdf.

53. M. Heker and H. Greenspan. "Joint Liver Lesion Segmentation and Classification via Transfer Learning", 2020. arXiv: 2004.12352.

54. D. Hendrycks and K. Gimpel. "Gaussian Error Linear Units (GELUs)", 2016. arXiv: 1606.08415.

55. E. U. Henry, O. Emebob, and C. A. Omonhinmin. "Vision Transformers in Medical Imaging: A Review", 2022. arXiv: 2211.10043.

56. S. I. Hossain. "Early Diagnosis of Lyme Disease by Recognizing Erythema Migrans Skin Lesion from Images Utilizing Deep Learning Techniques". In: *IJCAI International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Vienna, 2022, pp. 5855–5856. ISBN: 9781956792003. DOI: 10.24963/ijcai.2022/830.

57. S. I. Hossain, M. A. Akhand, M. I. Shuvo, N. Siddique, and H. Adeli. "Optimization of University Course Scheduling Problem using Particle Swarm Optimization with Selective Search". *Expert Systems with Applications* 127, 2019, pp. 9–24. ISSN: 09574174. DOI: 10.1016/j.eswa.2019.02.026.

58. S. I. Hossain, J. de Goër de Herve, D. Abrial, R. Emilion, I. Lebertb, Y. Frendo, D. Martineau, O. Lesens, and E. Mephu Nguifo. "Assisting Deep Learning based Lyme Disease Classifier with Patient Data". In: *Apprentissage automatique multimodal et fusion d'informations (3ième édition)*. 2022.

59. S. I. Hossain, J. de Goër de Herve, D. Abrial, R. Emilion, I. Lebertb, Y. Frendo, D. Martineau, O. Lesens, and E. Mephu Nguifo. "Expert Opinion Elicitation for Assisting Deep Learning based Lyme Disease Classifier with Patient Data", 2022. arXiv: 2208.14384.

60. S. I. Hossain, J. de Goër de Herve, Y. Frendo, D. Martineau, I. Lebert, O. Lesens, and E. Mephu Nguifo. "EMScan : une application mobile pour l'assistance au diagnostic des formes précoces de la maladie de Lyme". *Revue des Nouvelles Technologies de l'Information* Extraction et Gestion des Connaissances, RNTI-E-39, 2023, pp. 613–620.

61. S. I. Hossain, J. de Goër de Herve, M. S. Hassan, D. Martineau, E. Petrosyan, V. Corbin, J. Beytout, I. Lebert, J. Durand, I. Carravieri, A. Brun-Jacob, P. Frey-Klett, E. Baux, C. Cazorla, C. Eldin, Y. Hansmann, S. Patrat-Delon, T. Prazuck, A. Raffetin, P. Tattevin, G. Vourc'h, O. Lesens, and E. Mephu Nguifo. "Exploring convolutional neural networks with transfer learning for diagnosing Lyme disease from skin lesion images". *Computer Methods and Programs in Biomedicine* 215, 2022, p. 106624. ISSN: 01692607. DOI: 10.1016/j.cmpb.2022.106624.

62. S. I. Hossain, E. Mephu Nguifo, and J. de Goër de Herve. "Early Diagnosis of Lyme Disease by Recognizing Erythema Migrans Skin Lesion from Images Utilizing Deep Learning Techniques". In: *Deep learning with weak or few labels in medical image analysis*. 2022.

63. S. I. Hossain, S. S. Roy, J. de Goër de Herve, R. E. Mercer, and E. Mephu Nguifo. "A skin lesion hair mask dataset with fine-grained annotations". 1, 2023. DOI: 10.17632/J5YWPD2P27.1.

64. A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. "Searching for MobileNetV3". *Proceedings of the IEEE International Conference on Computer Vision* 2019-October, 2019, pp. 1314–1324. arXiv: 1905.02244.

65. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". *arXiv*, 2017. arXiv: 1704.04861.

66. Y. C. Hsu, Y. Shen, H. Jin, and Z. Kira. "Generalized ODIN: Detecting Out-of-Distribution Image without Learning from Out-of-Distribution Data". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 10948–10957. ISBN: 978-1-7281-7168-5. DOI: 10.1109/CVPR42600.2020.01096. arXiv: 2002.11297.

67. J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu. "Squeeze-and-Excitation Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42:8, 2020, pp. 2011–2023. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2019.2913372. arXiv: 1709.01507.

68. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. "Densely Connected Convolutional Networks". *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January, 2016, pp. 2261–2269. arXiv: 1608.06993.

69. S.-C. Huang, A. Pareek, S. Seyyedi, I. Banerjee, and M. P. Lungren. "Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines". *npj Digital Medicine* 3:1, 2020, p. 136. ISSN: 2398-6352. DOI: 10.1038/s41746-020-00341-z.

70. A. Inc. *Adobe Photoshop*. URL: https://www.adobe.com/products/photoshop.html (visited on 01/08/2023).

71. INRAE. *Institut national de recherche pour l'agriculture, l'alimentation et l'environnement - INRAE*. URL: https://www.inrae.fr/en (visited on 02/20/2023).

72. S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". *32nd International Conference on Machine Learning, ICML 2015* 1, 2015, pp. 448–456. DOI: 10.48550/arxiv.1502.03167. arXiv: 1502.03167.

73. H. Itoh, Z. Lu, Y. Mori, M. Misawa, M. Oda, S.-e. Kudo, and K. Mori. "Visualising decision-reasoning regions in computer-aided pathological pattern diagnosis of endoscytoscopic images based on CNN weights analysis". In: *Medical Imaging 2020: Computer-Aided Diagnosis*. Ed. by H. K. Hahn and M. A. Mazurowski. SPIE, 2020, p. 113. ISBN: 9781510633957. DOI: 10.1117/12.2549532.

74. D. Jacob, O. Nankar, S. Gite, S. Patil, and K. Kotecha. "Lyme Disease Detection Using Progressive Resizing and Self-Supervised Learning Algorithmslyme Disease Detection Using Progressive Resizing and Self-Supervised Learning Algorithms". *SSRN Electronic Journal*, 2022. ISSN: 1556-5068. DOI: 10.2139/ssrn.4059738.

75. J. Ji. "Gradient-based Interpretation on Convolutional Neural Network for Classification of Pathological Images". In: *Proceedings - 2019 International Conference on Information Technology and Computer Application, ITCA 2019*. IEEE, 2019, pp. 83–86. ISBN: 9781728164946. DOI: 10.1109/ITCA49981.2019.00026.

76. A. E. Kaid, K. Baïna, J. Baïna, and V. Barra. "Real-World Case Study of a Deep Learning Enhanced Elderly Person Fall Video-Detection System". In: *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,* INSTICC. SciTePress, 2023, pp. 575–582. ISBN: 978-989-758-634-7. DOI: 10.5220/0011674800003417.

77. U. S. Karmarkar. "Subjectively weighted utility: A descriptive extension of the expected utility model". *Organizational behavior and human performance* 21:1, 1978, pp. 61–72. DOI: 10.1016/0030-5073(78)90039-9.

78. A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi. "A survey of the recent architectures of deep convolutional neural networks". *Artificial Intelligence Review* 53:8, 2020, pp. 5455–5516. DOI: 10.1007/s10462-020-09825-6.

79. B.-H. Kim and J. C. Ye. "Understanding Graph Isomorphism Network for rs-fMRI Functional Connectivity Analysis". *Frontiers in Neuroscience* 14, 2020. ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00630.

80. D. P. Kingma and J. L. Ba. "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2015. arXiv: 1412.6980v9.

81. T. Koduru and E. Zhang. "Using Deep Learning in Lyme Disease Diagnosis". *Journal of Student Research* 10:4, 2021. ISSN: 2167-1907. DOI: 10.47611/jsrhs.v10i4.2389.

82. A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks". *Communications of the ACM* 60:6, 2017, pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386.

83. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86:11, 1998, pp. 2278–2324. DOI: 10.1109/5.726791.

84. J. H. Lee, E. J. Ha, D. Y. Kim, Y. J. Jung, S. Heo, Y. ho Jang, S. H. An, and K. Lee. "Application of deep learning to the diagnosis of cervical lymph node metastasis from thyroid cancer with CT: external validation and clinical utility for resident training". *European Radiology* 30:6, 2020, pp. 3066–3072. ISSN: 14321084. DOI: 10.1007/s00330-019-06652-4.

85. P. Letertre-Gibert, G. Vourc'h, I. Lebert, M. Rene-Martellet, V. Corbin-Valdenaire, D. Portal-Martineau, J. Beytout, and O. Lesens. "Lyme snap: A feasibility study of on-line declarations of erythema migrans in a rural area of France". *Ticks and Tick-borne Diseases* 11:1, 2020. ISSN: 18779603. DOI: 10.1016/j.ttbdis.2019.101301.

86. D. Li, J. Vaidya, M. Wang, B. Bush, C. Lu, M. Kollef, and T. Bailey. "Feasibility Study of Monitoring Deterioration of Outpatients Using Multimodal Data Collected by Wearables". *ACM Trans. Comput. Healthcare* 1:1, 2020. ISSN: 2691-1957. DOI: 10.1145/3344256.

87. L. Li, B. Du, Y. Wang, L. Qin, and H. Tan. "Estimation of missing values in heterogeneous traffic data: Application of multimodal deep learning model". *Knowledge-Based Systems* 194, 2020, p. 105592. ISSN: 09507051. DOI: 10.1016/j.knosys.2020.105592.

88. W. Li, A. N. Joseph Raj, T. Tjahjadi, and Z. Zhuang. "Digital hair removal by deep learning for skin lesion segmentation". *Pattern Recognition* 117, 2021, p. 107994. ISSN: 00313203. DOI: 10.1016/j.patcog.2021.107994.

89. W. Li, A. N. Joseph Raj, T. Tjahjadi, and Z. Zhuang. "Digital hair removal by deep learning for skin lesion segmentation". *Pattern Recognition* 117, 2021, p. 107994. ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2021.107994.

90. M. Lin, Q. Chen, and S. Yan. "Network in network". *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014. arXiv: 1312.4400.

91. G. Liu and I. Bichindaritz. "An Explainable Deep Network Framework with Case-based Reasoning Strategies for Survival Analysis in Cancer", 2022. DOI: 10.21203/rs.3.rs-2184342/v1.

92. J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan. "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 7498–7512.

93. J. Liu, G. Zhao, Y. Fei, M. Zhang, Y. Wang, and Y. Yu. "Align, Attend and Locate: Chest X-Ray Diagnosis via Contrast Induced Attention Network With Limited Supervision". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 10631–10640. DOI: 10.1109/ICCV.2019.01073.

94. X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. "Self-Supervised Learning: Generative or Contrastive". *IEEE Transactions on Knowledge and Data Engineering* 35:1, 2023, pp. 857–876. DOI: 10.1109/TKDE.2021.3090866.

95. Y. Liu, A. Jain, C. Eng, D. H. Way, K. Lee, P. Bui, K. Kanada, G. de Oliveira Marinho, J. Gallegos, S. Gabriele, V. Gupta, N. Singh, V. Natarajan, R. Hofmann-Wellenhof, G. S. Corrado, L. H. Peng, D. R. Webster, D. Ai, S. J. Huang, Y. Liu, R. C. Dunn, and D. Coz. "A deep learning system for differential diagnosis of skin diseases". *Nature Medicine* 26:6, 2020, pp. 900–908. ISSN: 1546170X. DOI: 10.1038/s41591-020-0842-3. arXiv: 1909.05382.

96. Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. "A ConvNet for the 2020s". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2022-June, 2022, pp. 11966–11976. ISSN: 10636919. DOI: 10.1109/CVPR52688.2022.01167. arXiv: 2201.03545.

97. Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11976–11986.

98. M. Ma, J. Ren, L. Zhao, S. Tulyakov, C. Wu, and X. Peng. "Smil: Multimodal learning with severely missing modality". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 3. 2021, pp. 2302–2310.

99. A. L. Maas, A. Y. Hannun, A. Y. Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. ICML*. Vol. 30. 1. Atlanta, Georgia, USA. 2013, p. 3.

100. H. H. Mao. "A Survey on Self-supervised Pre-training for Sequential Transfer Learning in Neural Networks", 2020. DOI: 10.48550/arXiv.2007.00800. arXiv: 2007.00800.

101. R. C. Maron et al. "Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks". *European Journal of Cancer* 119, 2019, pp. 57–65. ISSN: 18790852. DOI: 10.1016/j.ejca.2019.06.013.

102. A. R. Marques, F. Strle, and G. P. Wormser. "Comparison of lyme disease in the United States and Europe". *Emerging Infectious Diseases* 27:8, 2021, pp. 2017–2024. ISSN: 10806059. DOI: 10.3201/eid2708.204763.

103. J. Martins, J. S. Cardoso, and F. Soares. "Offline computer-aided diagnosis for Glaucoma detection using fundus images targeted at mobile devices". *Computer Methods and Programs in Biomedicine* 192, 2020, p. 105341. ISSN: 18727565. DOI: 10.1016/j.cmpb.2020.105341.

104. W. S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent in nervous activity". *The bulletin of mathematical biophysics* 5, 1943, pp. 115–133.

105. S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. C. Corrado, A. Darzi, M. Etemadi, F. Garcia-Vicente, F. J. Gilbert, M. Halling-Brown, D. Hassabis, S. Jansen, A. Karthikesalingam, C. J. Kelly, D. King, J. R. Ledsam, D. Melnick, H. Mostofi, L. Peng, J. J. Reicher, B. Romera-Paredes, R. Sidebottom, M. Suleyman, D. Tse, K. C. Young, J. De Fauw, and S. Shetty. "International evaluation of an AI system for breast cancer screening". *Nature* 577:7788, 2020, pp. 89–94. ISSN: 14764687. DOI: 10.1038/s41586-019-1799-6.

106. A. Menegola, M. Fornaciali, R. Pires, F. V. Bittencourt, S. Avila, and E. Valle. "Knowledge transfer for melanoma screening with deep learning". In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. 2017, pp. 297–300. DOI: 10.1109/ISBI.2017.7950523.

107. E. Mephu Nguifo and P. Njiwoua. "Using lattice-based framework as a tool for feature extraction". In: *Machine Learning: ECML-98*. Ed. by C. Nédellec and C. Rouveirol. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 304–309. ISBN: 978-3-540-69781-7.

108. E. Mephu Nguifo and J. Sallantin. "Prediction of primate splice junction gene sequences with a cooperative knowledge acquisition system." In: *ISMB*. 1993, pp. 292–300.

109. S. Motameny, B. Versmold, and R. Schmutzler. "Formal Concept Analysis for the Identification of Combinatorial Biomarkers in Breast Cancer". In: *Formal Concept Analysis*. Ed. by R. Medina and S. Obiedkov. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 229–240.

110. S. Narayan. "The Generalized Sigmoid Activation Function: Competitive Supervised Learning". *Information Sciences* 99:1-2, 1997, pp. 69–82. ISSN: 00200255. DOI: 10.1016/S0020-0255(96)00200-9.

111. P. B. Nemenyi. "Distribution-free multiple comparisons (PhD thesis)". PhD thesis. Princeton University, 1963.

112. A. Neznanov and A. Parinov. *Formal Concept Analysis Research Toolbox (FCART)*. 2016. URL: https://cs.hse.ru/en/ai/issa/proj_fcart (visited on 01/08/2023).

113. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. "Multimodal deep learning". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 689–696.

114. J. Ni, L. Muhlstein, and J. McAuley. "Modeling Heart Rate and Activity Data for Personalized Fitness Recommendation". In: *The World Wide Web Conference*. WWW '19. Association for Computing Machinery, San Francisco, CA, USA, 2019, pp. 1343–1353. ISBN: 9781450366748. DOI: 10.1145/3308558.3313643.

115. I. Oholtsov, Y. Gordienko, and S. Stirenko. "Effect of Small Dataset Quality on Deep Neural Network Performance for Lyme Disease Classification". In: 2023, pp. 561–573. DOI: 10.1007/978-981-19-3590-9_44.

116. A. G. C. Pacheco and R. A. Krohling. "An Attention-Based Mechanism to Combine Images and Metadata in Deep Learning Models Applied to Skin Cancer Classification". *IEEE Journal of Biomedical and Health Informatics* 25:9, 2021, pp. 3554–3563. DOI: 10.1109/JBHI.2021.3062002.

117. S. J. Pan and Q. Yang. "A Survey on Transfer Learning". *IEEE Transactions on Knowledge and Data Engineering* 22:10, 2010, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

118. Y. Pan, M. Liu, Y. Xia, and D. Shen. "Disease-image-specific learning for diagnosis-oriented neuroimage synthesis with incomplete multi-modality data". *IEEE transactions on pattern analysis and machine intelligence* 44:10, 2021, pp. 6839–6853.

119. E. Parzen. "On Estimation of a Probability Density Function and Mode". *The Annals of Mathematical Statistics* 33:3, 1962, pp. 1065–1076. ISSN: 0003-4851. DOI: 10.1214/aoms/1177704472.

120. F. Perez, C. Vasconcelos, S. Avila, and E. Valle. "Data augmentation for skin lesion analysis". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11041 LNCS. Springer Verlag, 2018, pp. 303–311. ISBN: 9783030012007. DOI: 10.1007/978-3-030-01201-4_33. arXiv: 1809.01442.

121. E. Pérez, O. Reyes, and S. Ventura. "Convolutional neural networks for the automatic diagnosis of melanoma: An extensive experimental study". *Medical Image Analysis* 67, 2021, p. 101858. ISSN: 13618423. DOI: 10.1016/j.media.2020.101858.

122. J. Plested and T. Gedeon. "Deep transfer learning for image classification: a survey", 2022. DOI: 10.48550/arxiv.2205.09904. arXiv: 2205.09904.

123. Z. Qin, F. Yu, C. Liu, and X. Chen. "How convolutional neural networks see the world — A survey of convolutional neural network visualization methods". *Mathematical Foundations of Computing* 1:2, 2018, pp. 149–180. ISSN: 2577-8838. DOI: 10.3934/mfc.2018008. arXiv: 1804.11191.

124. J. R. Quinlan. "Induction of decision trees". *Machine learning* 1, 1986, pp. 81–106. DOI: 10.1007/BF00116251.

125. M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. "Transfusion: Understanding transfer learning for medical imaging". *Advances in Neural Information Processing Systems* 32, 2019. ISSN: 10495258. arXiv: 1902.07208.

126. A. Rahate, R. Walambe, S. Ramanna, and K. Kotecha. "Multimodal co-learning: challenges, applications with datasets, recent advances and future directions". *Information Fusion* 81, 2022, pp. 203–239.

127. P. Ramachandran, B. Zoph, and Q. V. Le Google Brain. "Searching for activation functions". *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 2018. DOI: 10.48550/arxiv.1710.05941. arXiv: 1710.05941.

128. D. Reading. *Perceptron*. URL: https://github.com/dreading/tex-neural-network (visited on 02/20/2023).

129. P. Ren, Y. Xiao, X. Chang, P. Y. Huang, Z. Li, X. Chen, and X. Wang. "A comprehensive survey of neural architecture search: Challenges and solutions". *ACM Computing Surveys* 54:4, 2021, pp. 1–34. ISSN: 15577341. DOI: 10.1145/3447582. arXiv: 2006.02903.

130. D. Reynolds. "Gaussian Mixture Models". In: *Encyclopedia of Biometrics*. Ed. by S. Z. Li and A. Jain. Springer US, Boston, MA, 2009, pp. 659–663. ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_196.

131. O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Vol. 9351. Springer International Publishing, Cham, 2015, pp. 234–241. ISBN: 9783319245737. DOI: 10.1007/978-3-319-24574-4_28.

132. M. Rosenblatt. "Remarks on Some Nonparametric Estimates of a Density Function". *The Annals of Mathematical Statistics* 27:3, 1956, pp. 832–837. ISSN: 0003-4851. DOI: 10.1214/aoms/1177728190.

133. S. Ruder. "An overview of gradient descent optimization algorithms", 2016. arXiv: 1609.04747.

134. S. Ruder. "Neural Transfer Learning for Natural Language Processing". PhD thesis. National University of Ireland, Galway, 2019.

135. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". *International Journal of Computer Vision* 115:3, 2015, pp. 211–252. ISSN: 15731405. DOI: 10.1007/s11263-015-0816-y. arXiv: 1409.0575.

136. C. Saegerman, J. Evrard, J.-Y. Houtain, J.-P. Alzieu, J. Bianchini, S. E. Mpouam, G. Schares, E. Liénard, P. Jacquiet, L. Villa, G. Álvarez-García, A. L. Gazzonis, A. Gentile, and L. Delooz. "First Expert Elicitation of Knowledge on Drivers of Emergence of Bovine Besnoitiosis in Europe." *Pathogens (Basel, Switzerland)* 11:7, 2022. ISSN: 2076-0817. DOI: 10.3390/pathogens11070753.

137. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. arXiv: 1801.04381.

138. S. Sayad. *Decision Tree - Classification*. URL: https://www.saedsayad.com/decision_tree.htm (visited on 02/20/2023).

139. C. E. von Schacky, J. H. Sohn, F. Liu, E. Ozhinsky, P. M. Jungmann, L. Nardo, M. Posadzy, S. C. Foreman, M. C. Nevitt, T. M. Link, and V. Pedoia. "Development and validation of a multitask deep learning model for severity grading of hip osteoarthritis features on radiographs". *Radiology* 295:1, 2020, pp. 139–145. ISSN: 15271315. DOI: 10.1148/radiol.2020190925.

140. G. Schwarz. "Estimating the Dimension of a Model". *The Annals of Statistics* 6:2, 2007. ISSN: 0090-5364. DOI: 10.1214/aos/1176344136.

141. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". *International Journal of Computer Vision* 128:2, 2020, pp. 336–359. ISSN: 15731405. DOI: 10.1007/s11263-019-01228-7. arXiv: 1610.02391.

142. H. Senaratne, S. Oviatt, K. Ellis, and G. Melvin. "A Critical Review of Multimodal-Multisensor Analytics for Anxiety Assessment". *ACM Trans. Comput. Healthcare* 3:4, 2022. ISSN: 2691-1957. DOI: 10.1145/3556980.

143. D. Seth, K. Cheldize, D. Brown, and E. E. Freeman. "Global Burden of Skin Disease: Inequities and Innovations". *Current Dermatology Reports* 6:3, 2017, pp. 204–210. ISSN: 21624933. DOI: 10.1007/s13671-017-0192-7.

144. E. D. Shapiro. "Clinical practice. Lyme disease." *The New England journal of medicine* 370:18, 2014. Ed. by C. G. Solomon, pp. 1724–31. ISSN: 1533-4406. DOI: 10.1056/NEJMcp1314325.

145. Y. Shlomi and T. S. Wallsten. "Subjective recalibration of advisors' probability estimates". *Psychonomic bulletin & review* 17, 2010, pp. 492–498. DOI: 10.3758/PBR.17.4.492.

146. B. W. Silverman. *Density estimation: For statistics and data analysis*. Chapman & Hall, London, 2018, pp. 1–175. ISBN: 9781351456173. DOI: 10.1201/9781315140919.

147. K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. Technical report. 2015. arXiv: 1409.1556.

148. H. Sowrirajan, J. Yang, A. Y. Ng, and P. Rajpurkar. *MoCo-CXR: MoCo Pretraining Improves Representation and Transferability of Chest X-ray Models*. 2020. DOI: 10.48550/ARXIV.2010.05352.

149. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting". *Journal of Machine Learning Research* 15:56, 2014, pp. 1929–1958. ISSN: 15337928.

150. F. Strle and G. Stanek. "Clinical manifestations and diagnosis of lyme borreliosis". In: *Current Problems in Dermatology*. Vol. 37. KARGER, Basel, 2009, pp. 51–110. ISBN: 9783805591140. DOI: 10.1159/000213070.

151. *Structure of Neuron*. URL: https://commons.wikimedia.org/wiki/File:Structure_of_Neuron.png (visited on 02/20/2023).

152. D. Stutz. *Multilayer perceptron*. URL: https://github.com/davidstutz/latex-resources/tree/master/tikz-multilayer-perceptron (visited on 02/20/2023).

153. W. Sun, X. Zhang, and X. He. "Lightweight image classifier using dilated and depthwise separable convolutions". *Journal of Cloud Computing* 9:1, 2020, p. 55. ISSN: 2192113X. DOI: 10.1186/s13677-020-00203-9.

154. X. Sun, J. Yang, M. Sun, and K. Wang. "A benchmark for automatic visual classification of clinical skin disease images". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*. Springer. 2016, pp. 206–222.

155. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. "Inception-v4, inception-ResNet and the impact of residual connections on learning". In: *31st AAAI Conference on Artificial Intelligence, AAAI 2017*. AAAI press, 2017, pp. 4278–4284. arXiv: 1602.07261v2.

156. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 07-12-June. IEEE Computer Society, 2015, pp. 1–9. ISBN: 9781467369640. DOI: 10.1109/CVPR.2015.7298594. arXiv: 1409.4842v1.

157. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-Decem. IEEE Computer Society, 2016, pp. 2818–2826. ISBN: 9781467388504. DOI: 10.1109/CVPR.2016.308. arXiv: 1512.00567v3.

158. M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le. "Mnasnet: Platform-aware neural architecture search for mobile". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June, 2019, pp. 2815–2823. ISSN: 1063-6919. DOI: 10.1109/CVPR.2019.00293. arXiv: 1807.11626.

159. M. Tan and Q. V. Le. "EfficientNet: Rethinking model scaling for convolutional neural networks". In: *36th International Conference on Machine Learning, ICML 2019*. Vol. 2019-June. International Machine Learning Society (IMLS), 2019, pp. 10691–10700. ISBN: 9781510886988. arXiv: 1905.11946v5.

160. M. Tan and Q. V. Le. "EfficientNetV2: Smaller Models and Faster Training", 2021. arXiv: 2104.00298.

161. T. Tieleman, G. Hinton, et al. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude". *COURSERA: Neural networks for machine learning* 4:2, 2012, pp. 26–31.

162. H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, et al. "Resmlp: Feedforward networks for image classification with data-efficient training". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

163. H. Tran, K. Chen, A. C. Lim, J. Jabbour, and S. Shumack. "Assessing diagnostic skill in dermatology: A comparison between general practitioners and dermatologists". *Australasian Journal of Dermatology* 46:4, 2005, pp. 230–234. ISSN: 00048380. DOI: 10.1111/j.1440-0960.2005.00189.x.

164. G. Trevisan, S. Bonin, and M. Ruscio. "A Practical Approach to the Diagnosis of Lyme Borreliosis: From Clinical Heterogeneity to Laboratory Methods". *Frontiers in Medicine* 7, 2020, p. 265. ISSN: 2296858X. DOI: 10.3389/fmed.2020.00265.

165. P. Tschandl, C. Rosendahl, and H. Kittler. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". *Scientific Data* 5:1, 2018, p. 180161. ISSN: 2052-4463. DOI: 10.1038/sdata.2018.161. arXiv: 1803.10417.

166. L. C. Van Der Gaag, S. Renooij, C. L. Witteman, B. M. Aleman, and B. G. Taal. "Probabilities for a probabilistic network: A case study in oesophageal cancer". *Artificial Intelligence in Medicine* 25:2, 2002, pp. 123–148. ISSN: 09333657. DOI: 10.1016/S0933-3657(02)00012-X.

167. B. H. van der Velden, H. J. Kuijf, K. G. Gilhuijs, and M. A. Viergever. "Explainable artificial intelligence (XAI) in deep learning-based medical image analysis". *Medical Image Analysis* 79, 2022, p. 102470. ISSN: 13618415. DOI: 10.1016/j.media.2022.102470.

168. P. Veličković. *TikZ*. URL: https://github.com/PetarV-/TikZ/ (visited on 02/20/2023).

169. A. Vellido. "The importance of interpretability and visualization in machine learning for applications in medicine and health care". *Neural Computing and Applications* 32:24, 2020, pp. 18069–18083. ISSN: 1433-3058. DOI: 10.1007/s00521-019-04051-w.

170. N. Vila-Blanco, M. J. Carreira, P. Varas-Quintana, C. Balsa-Castro, and I. Tomás. "Deep Neural Networks for Chronological Age Estimation From OPG Images". *IEEE Transactions on Medical Imaging* 39:7, 2020, pp. 2374–2384. DOI: 10.1109/TMI.2020.2968765.

171. Q. Wang, L. Zhan, P. Thompson, and J. Zhou. "Multimodal learning with incomplete modalities by knowledge distillation". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 1828–1838.

172. Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu. "ECA-Net: Efficient channel attention for deep convolutional neural networks". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11531–11539. ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.01155. arXiv: 1910.03151.

173. D. Wen, S. M. Khan, A. J. Xu, H. Ibrahim, L. Smith, J. Caballero, L. Zepeda, C. de Blas Perez, A. K. Denniston, X. Liu, and R. N. Matin. "Characteristics of publicly available skin cancer image datasets: a systematic review". *The Lancet Digital Health* 4:1, 2022, e64–e74. ISSN: 25897500. DOI: 10.1016/S2589-7500(21)00252-1.

174. R. Wille. "Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts". In: *Ordered Sets*. Ed. by I. Rival. Springer Netherlands, Dordrecht, 1982, pp. 445–470.

175. E. C. Wilson, J. A. Usher-Smith, J. Emery, P. G. Corrie, and F. M. Walter. "Expert Elicitation of Multinomial Probabilities for Decision-Analytic Modeling: An Application to Rates of Disease Progression in Undiagnosed and Untreated Melanoma". *Value in Health* 21:6, 2018, pp. 669–676. ISSN: 15244733. DOI: 10.1016/j.jval.2017.10.009.

176. H. Xie, H. Shan, W. Cong, X. Zhang, S. Liu, R. Ning, and G. Wang. "Dual network architecture for few-view CT - trained on ImageNet data and transferred for medical imaging". In: *Developments in X-Ray Tomography XII*. Ed. by B. Müller and G. Wang. Vol. 11113. International Society for Optics and Photonics. SPIE, 2019, p. 111130V. DOI: 10.1117/12.2531198.

177. X. Xie, X. Song, Z. Lv, G. G. Yen, W. Ding, and Y. Sun. "Efficient Evaluation Methods for Neural Architecture Search: A Survey", 2023. arXiv: 2301.05919.

178. R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi. "Convolutional neural networks: an overview and application in radiology". *Insights into imaging* 9, 2018, pp. 611–629.

179. J. Yang, K. Zhou, Y. Li, and Z. Liu. "Generalized Out-of-Distribution Detection: A Survey", 2021. arXiv: 2110.11334.

180. J. Yang, X. Wu, J. Liang, X. Sun, M.-M. Cheng, P. L. Rosin, and L. Wang. "Self-Paced Balance Learning for Clinical Skin Disease Recognition". *IEEE Transactions on Neural Networks and Learning Systems* 31:8, 2020, pp. 2832–2846. DOI: 10.1109/TNNLS.2019.2917524.

181. T. J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam. "NetAdapt: Platform-aware neural network adaptation for mobile applications". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11214 LNCS, 2018, pp. 289–304. ISSN: 1611-3349. DOI: 10.1007/978-3-030-01249-6_18. arXiv: 1804.03230.

182. S. Yevtushenko, J. Tane, T. B. Kaiser, S. Objedkov, J. H. Correia, and H. Reppe. *The Concept Explorer*. URL: https://conexp.sourceforge.net/ (visited on 01/08/2023).

183. F. Yu and V. Koltun. "Multi-scale context aggregation by dilated convolutions". *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016. arXiv: 1511.07122.

184. A. Zadeh, M. Chen, E. Cambria, S. Poria, and L. P. Morency. "Tensor fusion network for multimodal sentiment analysis". *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017, pp. 1103–1114. DOI: 10.18653/v1/d17-1115. arXiv: 1707.07250.

185. M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Vol. 8689 LNCS. PART 1. Springer International Publishing, Cham, 2014, pp. 818–833. ISBN: 9783319105895. DOI: 10.1007/978-3-319-10590-1_53. arXiv: 1311.2901.

186. C. Zhang, X. Chu, L. Ma, Y. Zhu, Y. Wang, J. Wang, and J. Zhao. "M3Care: Learning with Missing Modalities in Multimodal Healthcare Data". In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '22. Association for Computing Machinery, Washington DC, USA, 2022, pp. 2418–2428. ISBN: 9781450393850. DOI: 10.1145/3534678.3539388.

187. E. Zhang. *Lyme Disease Erythema Migrans Rashes*. URL: https://www.kaggle.com/datasets/sshikamaru/lyme-disease-rashes (visited on 02/20/2023).

188. H.-Y. Zhou, S. Yu, C. Bian, Y. Hu, K. Ma, and Y. Zheng. "Comparing to Learn: Surpassing ImageNet Pretraining on Radiographs by Comparing Image Representations". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Ed. by A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz. Springer International Publishing, Cham, 2020, pp. 398–407. ISBN: 978-3-030-59710-8.

189. C. Zhuang, S. Yan, A. Nayebi, M. Schrimpf, M. C. Frank, J. J. DiCarlo, and D. L. Yamins. "Unsupervised neural network models of the ventral visual stream". *Proceedings of the National Academy of Sciences of the United States of America* 118:3, 2021. ISSN: 10916490. DOI: 10.1073/pnas.2014196118.

190. B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. "Learning Transferable Architectures for Scalable Image Recognition". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710. ISSN: 1063-6919. DOI: 10.1109/CVPR.2018.00907. arXiv: 1707.07012.

191. B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. "Learning Transferable Architectures for Scalable Image Recognition". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710. ISSN: 10636919. DOI: 10.1109/CVPR.2018.00907. arXiv: 1707.07012.

192. H. Zunair and A. Ben Hamza. "Melanoma detection using adversarial training and deep transfer learning". *Physics in Medicine and Biology* 65:13, 2020, p. 135005. ISSN: 13616560. DOI: 10.1088/1361-6560/ab86d3. arXiv: 2004.06824.