

# Using Deep Learning in Lyme Disease Diagnosis

Teja Koduru<sup>1</sup> and Edward Zhang<sup>#</sup>

<sup>1</sup>Thomas Jefferson High School for Science and Technology, Alexandria, VA, USA

<sup>#</sup>Advisor

## ABSTRACT

Untreated Lyme disease can lead to neurological, cardiac, and dermatological complications. Rapid diagnosis of the erythema migrans (EM) rash, a characteristic symptom of Lyme disease, is therefore crucial to early diagnosis and treatment. In this study, we aim to utilize deep learning frameworks including Tensorflow and Keras to create deep convolutional neural networks (DCNN) to detect images of acute Lyme Disease from images of erythema migrans. This study uses a custom database of erythema migrans images of varying quality to train a DCNN capable of classifying images of EM rashes vs non-EM rashes. Images from publicly available sources were mined to create an initial database. Machine based removal of duplicate images was then performed, followed by a thorough examination of all images by a clinician. The resulting database was combined with images of confounding rashes and regular skin, resulting in a total of 683 images. This database was then used to create a DCNN with an accuracy of 93% when classifying images of rashes as EM vs non EM. Finally, this model was converted into a web and mobile application to allow for rapid diagnosis of EM rashes by both patients and clinicians. This tool could be used for patient prescreening prior to treatment and lead to a lower mortality rate from Lyme disease.

## Introduction

Lyme borreliosis, also known as Lyme disease, is a tick-borne disease caused by the bacterium *Borrelia burgdorferi* and is the most common tick-borne disease in the US alone with over 300,000 cases per year [[1], [2]]. The disease is transmitted to humans through the bite of infected blacklegged ticks [1]. In approximately 70-80% of cases, a bullseye shaped rash called *erythema migrans* (EM) marks the site of infection and begins to rapidly spread in a circular fashion [3]. This rash typically appears one to two weeks after infection and can last for approximately 4 weeks after it first appears [4].

The diagnosis of Lyme disease relies on the visual identification of the EM rash or the identification of certain antibodies through serological testing. However, currently available two-tier serological tests are not recommended to be used during the early stages of Lyme disease due to their low specificity (Less than 40%) [5]. In addition, they are not always practical for use by giving the extended processing time for results [6]. The most common form of diagnosis is through a combination of symptoms, probability of exposure to ticks, and serological testing, which minimizes the risk of a false positive [1].

Early diagnosis of Lyme disease via the detection of the Erythema Migrans (EM) rash is critical because antibiotic therapy issued in this phase of the disease is effective in almost 99% of patients. If delayed, the disease may progress to the disseminated stage, affecting other organs including the heart and brain [7]. However, diagnosis of an EM rash remains a challenge because the lesion can take on many different appearances [8]. For example, EM rashes can be commonly confused with drug reactions, insect bites, Ringworm, Urticaria multiforme, or the Pityriasis rosea rash [9]. In one study, general practitioners correctly identified non-target EM lesions 64% of the time and classic target lesions 80% of the time [10]. This inherent difficulty in diagnosing EM rashes highlights the need for new tools to assist clinicians when diagnosing these rashes.

Deep learning (DL) is a set of techniques and algorithms that enable computers to discover complicated patterns in large sets of data. DL algorithms are being used more often in a medical setting. Advancements in DL started around 2012 when deep convolutional neural networks (DCNNs), especially with the introduction of AlexNet, began outperforming other established models on a number of important benchmarks [11]. Deep convolutional neural networks (DCNNs) are solutions to a wide variety of problems in computer vision, natural language processing and robotics. The use of DCNNs in the diagnosis of EM rashes has been investigated before, but these studies focus on the creation of the algorithm rather than its implementation into a usable form [[12], [13]].

This study aims to expand on the work of previous studies by not only creating a DCNN for EM classification, but by implementing the DCNN into a usable form such as an app or website. The creation of such a tool would facilitate the diagnosis of EM rashes by patients, clinicians, and researchers.

## Methods

We aim to create a DCNN capable of Binary Classification of EM versus non-EM rashes in skin images. Non-EM rashes include images of not only normal skin, but also confounding rashes including skin with fixed drug reaction eruptions (FDE), tinea corporis (TC), and pityriasis rosea (PR) [9]. FDE, TC, and PR are all lookalikes to the EM rash, but require different methods of treatment [[14], [15], [16]] .

### Data Collection

The lack of a publicly available dataset of EM rashes poses the first challenge in the creation of the DCNN. To create a database with a sufficient number of images, we first created a python-based web scraper using Selenium WebDriver. This web scraper was given a set of keywords including “Erythema migrans”, “bullseye rash”, and “ringworm rash”. Using these keywords, the web scraper downloaded images from Google Images. These images were then compiled into an initial database with 1194 images. Machine based removal of duplicate images was then performed. Finally, the dataset was curated by a clinician. Inappropriate or irrelevant images were removed, alongside images of low quality and low probability of accurate group classification.

### DCNN Training

In this study, we utilized a DCNN for our classification model which generates features from images at varying levels of detail. The preprocessing required in a DCNN is much lower compared to other traditional classification algorithms where filters are hand-engineered [17]. DCNNs are designed to adaptively learn spatial hierarchies of features and directly learn and extrapolate from the image data itself [17].

The DCNN takes a skin image as input and processes the images through many different layers (building blocks) to produce an output probability for each class. The three main layers include convolution, pooling, and fully connected layers.

The convolution and pooling layers use multiple layers of processing which apply linear operations between the pixel data to perform feature extraction from the given image. [18]. Next nonlinear operations such as the rectified linear unit (ReLU) activation function are applied to generate low, mid, and high-level feature representations of the input image.

The last layer is then flattened into a one dimensional vector and is further processed with a fully connected layer. Finally, the DCNN outputs a probability value via SoftMax for each class label. All weights and biases from the network are learned from backpropagation during training, using several loss optimization strategies such as Adam, RMSProp and F1.

## Data Preparation

Before training the model, we split our data into training and validation sets. In machine learning, it is common practice to split a dataset with 80% of the data used for training while the remaining portion is used for validation. We followed this convention with our data, taking 80% of EM and non-EM images for training and leaving 20 percent of the data for the validation set. Before training the model, we applied data augmentation techniques to our images, so we could generate more images without blatant repetition. Images were rotated, flipped, shifted, zoomed in and out, and various types of preprocessing was applied. The result of these techniques was an increase in the amount of data available to train the model.

## Modeling

Deep learning frameworks such as Tensorflow and Keras were used alongside transfer learning in this study. We utilized a variety of pre trained DCNNs including ResNet50, NasNetLarge, and Mobilenet [[21], [22], [23]]. All of these models were trained on the ImageNet dataset to classify thousands of different images. We used transfer learning to train each of these models with our data, to fine tune them with the task of classifying EM rashes. We used the loss optimizer Adam (with an initial learning rate of 0.00001), along with the binary cross entropy loss function for all of our DCNNs. During training, we also used an early stopping approach which would stop the training process after 20 epochs if the validation set accuracy did not improve. Additionally, the learning rate would be decreased by a factor of 0.8 if the loss loss did not reduce in 5 epochs. Moreover, a dropout of 0.5 was initiated in the compilation of the model to prevent overfitting.

## Implementation

We aim to convert the trained model into a format which can be stored locally on a mobile device, allowing users to diagnose rashes without a stable internet connection. We first converted the h5 file produced after training into a tflite file. To do this, we used the TensorFlow Lite Converter class in the Keras package. The tflite file has many advantages over the h5 file; namely, the tflite file is much smaller than the h5 file which allows it to be stored on a mobile device. This is necessary as the Google Play Store enforces a maximum app size of 100 MB. However, decreasing the storage space for a model results in accuracy of the model being degraded. To circumvent this issue, we utilized post-training quantization, specifically dynamic range quantization, to decrease the size of the model while simultaneously retaining the accuracy.

Using the tflite file produced earlier, we then created both a website and an app which used the model. Repurposed colab labs were used to create both the app and the website [[19], [20]]. By substituting our .tflite file with those found in the shell code, we were able to create a simple yet effective implementation of the model.

## Results

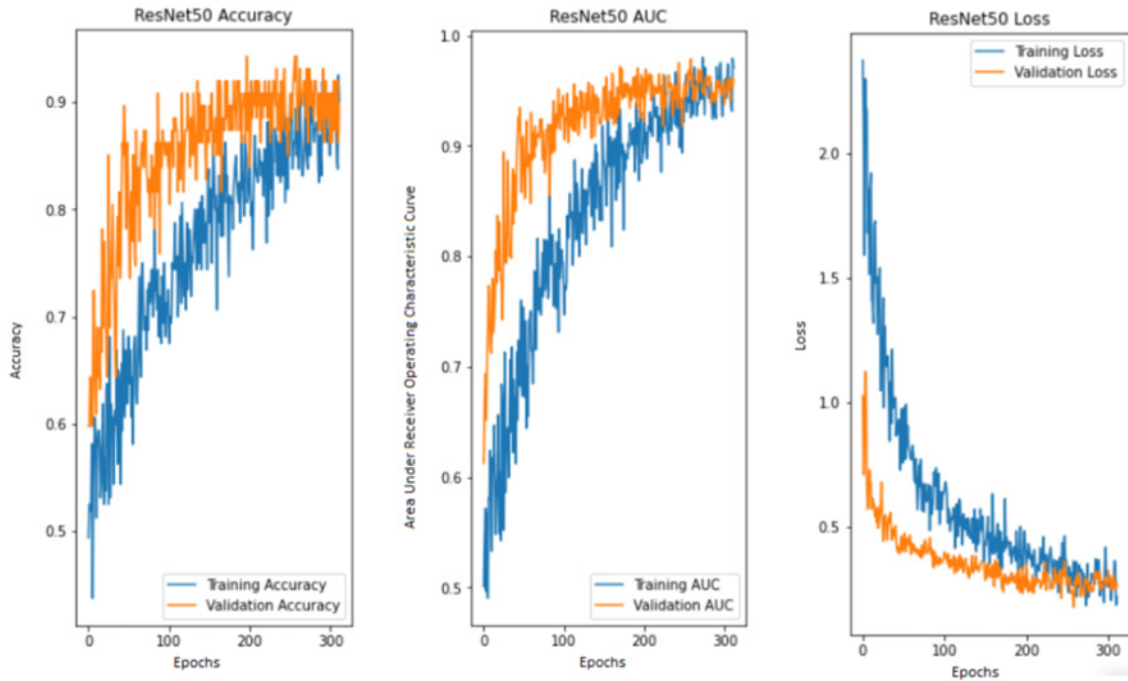


Fig. 1. Accuracy plot for ResNet50 Model

Fig. 2. Area Under Curve plot for the ResNet50 model.

Fig. 3. Loss plot for the ResNet50 model.

Figure 1 plots the training and validation accuracy of the model. For the two-class classification problem, the model had a peak training accuracy of 92 percent and a peak validation accuracy of 95 percent, but 93 was the most consistent high. One interesting point to note is how validation accuracy appears to be equal to, if not better, than the training cases at most points. This is most likely caused due to the use of a dropout of 0.5 when training the model. The model nearly always outperforms traditional clinician accuracy, which was found to be approximately 80% [13].

Figure 2 shows the AUC value of the model as a function of the number of epochs. AUC is a measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes. An AUC value of approximately 0.5 means that the model has no discriminatory ability, while a value above 0.5 indicates that the model is able to distinguish between the classes effectively. Our model has a maximum AUC of approximately 0.93, which is considered excellent [26].

The validation and training loss of the model is shown in Figure 3. Both the validation and training loss of the model fell to around 0.25 at the maximum number of epochs. This value was used to determine when to stop the training of the model. In this case, we stopped training the model at around 300 epochs as accuracy failed to significantly improve at that point.

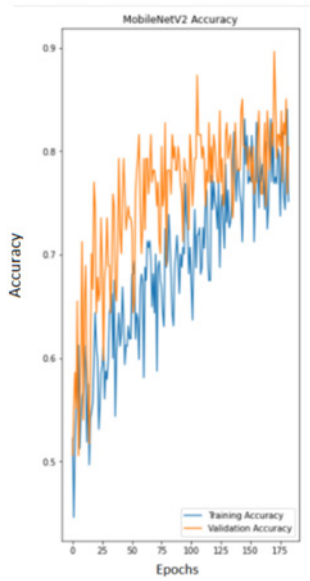


Fig. 4. Accuracy plot for MobileNet

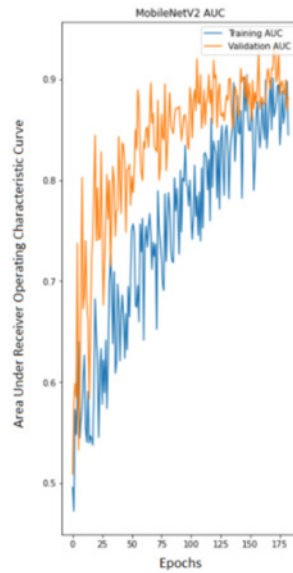


Fig. 5. AUC plot for MobileNet.

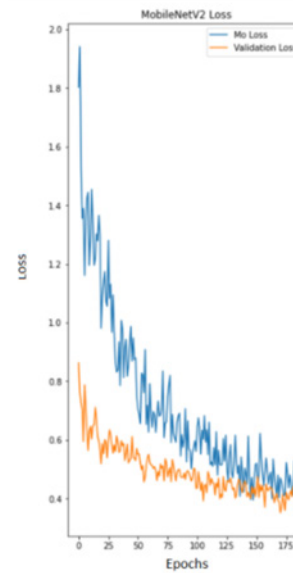


Fig. 6. Loss plot for MobileNet.

Figures 4, 5, and 6 plot the accuracy, loss, and the area under the curve values for the mobileNet model. As you can see, the ResNet50 model nearly always outperforms the mobileNet model in every measure.

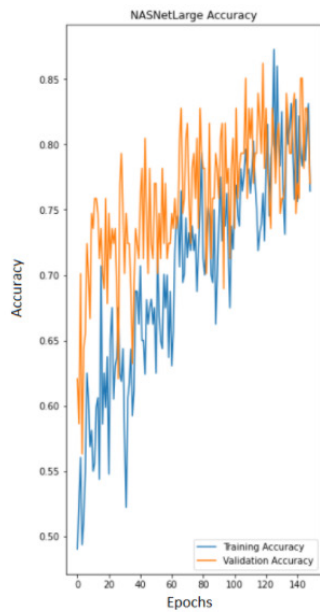


Fig. 7. Accuracy plot for NASNetLarge

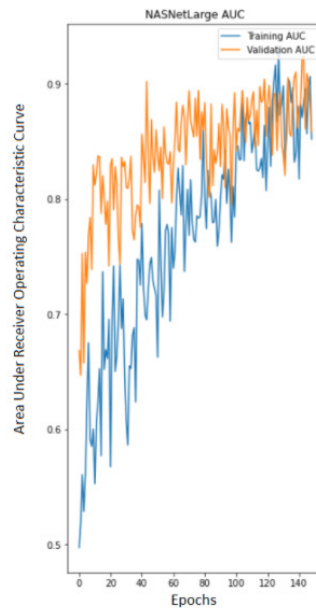


Fig. 8. AUC plot for NASNetLarge.

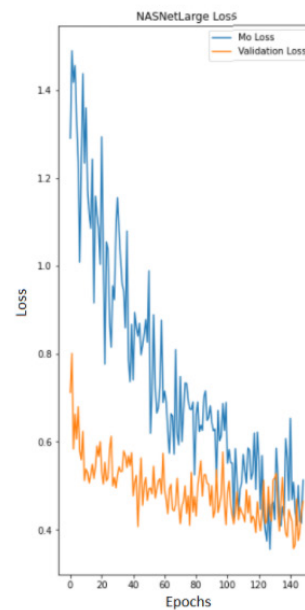
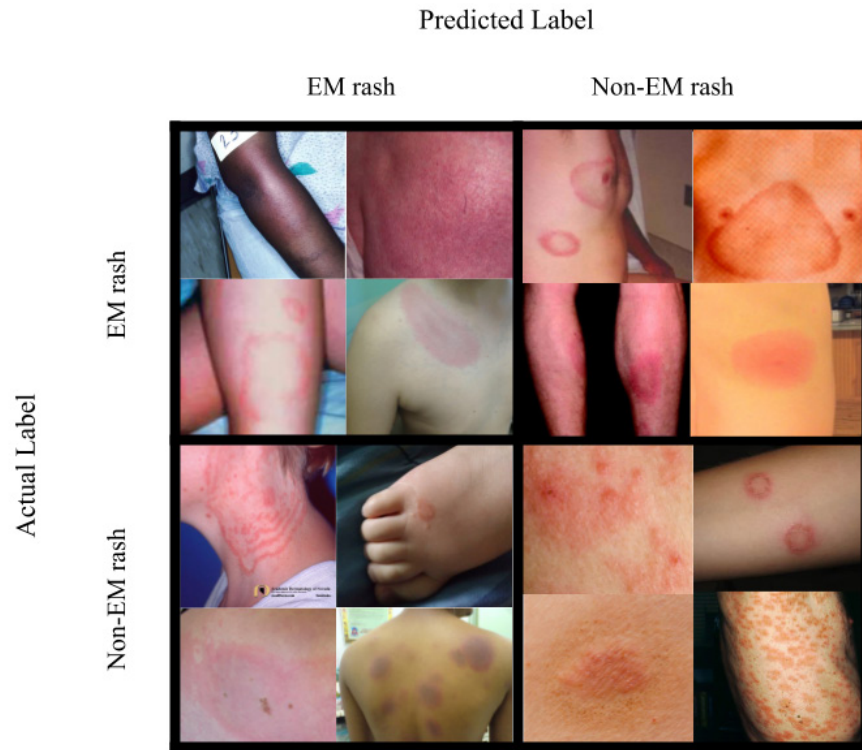


Fig. 9. Loss plot for NASNetLarge

Finally, figures 7, 8, and 9 plot the metrics for the NASNetLarge models. Unlike the other two models, the NASNetLarge model peaks at only an 85% accuracy. Training of the NASNetLarge model also ended much earlier than the other models, with only 140 epochs compared to the 175 of the mobileNet model and the 300+ of the

ResNet50 model. As both the mobileNet and NASNetLarge models fail to outperform the ResNet50 models on these benchmarks, the ResNet50 model was chosen to be implemented into an app.



The confusion matrix for the ResNet50 model is displayed above. A majority of the classifications were true positives or true negatives. Accuracy was 91.95% and the Matthews Correlation Coefficient was 0.8391. These results show that the model matches or outperforms clinicians when diagnosing EM rashes.

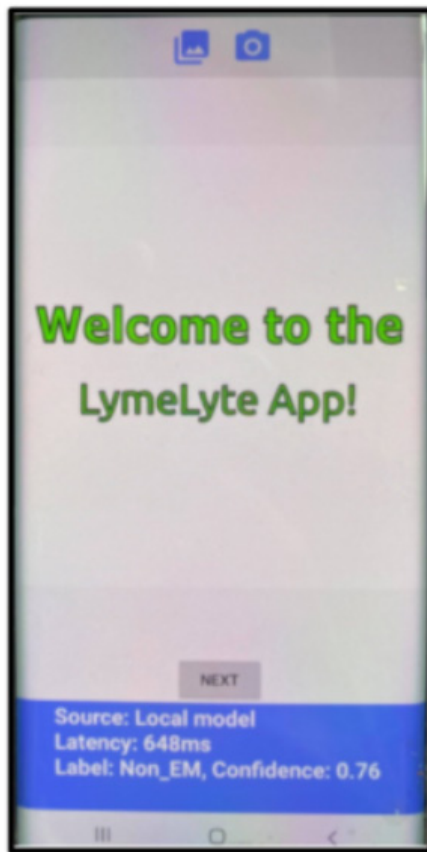


Fig. 10. Home screen of the app



Fig. 11. App being used to diagnose an EM rash

Figures 10 and 11 show the home screen of the app implementation of the model. The app itself is quite simple, and is designed this way to make it as easy as possible to diagnose a rash. The two buttons at the top allow users to either take a photo of the rash using their camera, or upload an image from the gallery. After selecting an image, the tflite model within the device classifies the image. Results are displayed at the bottom, alongside a measure of how long it took for the model to classify the image. By clicking the “next” button, users are shown a variety of screens informing them regarding EM rashes and how the app works. Once again, this app was built as a testing platform. We are currently working on ways to improve the user experience and the accuracy of the model within the device.

## Discussion & Conclusion

This study yielded promising results for the implementation of machine learning models into a mobile app for the early diagnosis of Lyme disease. Results indicated that the model has a high accuracy when diagnosing between EM and confounding rashes. However, the model had trouble properly classifying certain images of bare skin, which most likely resulted from a disproportionate ratio of skin data. Furthermore, like other studies, our model faced issues when

identifying EM rashes in darker skin tones. Once again, this is most likely due to the lack of sufficient training data containing images of EM rashes on dark skinned individuals.

We believe that this study lays the groundwork for future studies examining the implementation of DCNNs into mobile applications. There exist several avenues for further research into the field. Firstly, a publicly available dataset of EM rash images needs to be created. While this study and other similar studies use data collected from online sources, research into this field would be greatly benefitted from a large, central database of curated EM rash images available for all researchers. Next, future studies could examine debiasing the data to create a more generalizable dataset, given the problem of low accuracy on darker skinned images. This could potentially be achieved by using variational autoencoders (VAE) or simply having more data of the sort. Finally, while we have been able to implement our DCNN into a mobile app, this process could be done in a more efficient manner which would preserve the accuracy of the DCNN. We utilized post training quantization as a method to reduce the size of the model so that it could fit better into an app. Future studies may examine other methods of reducing the size of the model, so that more accuracy is retained.

### 5.1 Conclusion

We created a DCNN for EM rash classification and implemented it into a mobile application. Our results show that this DCNN is highly accurate when diagnosing between EM and non-EM rashes, but struggles when diagnosing dark skin. With more training data, this application could have many uses in the medical field. For example, patients could be prescreened by this application prior to seeing a clinician. While future research into the field is necessary, this study and other similar studies lay the groundwork for the use of machine learning in Lyme disease diagnosis.

## Acknowledgments

I would like to thank my advisor Edward Zhang for his guidance in this project.

## References

- [1] *Lyme Disease*. (n.d.). Center for Disease Control. Retrieved December 26, 2020, from <https://www.cdc.gov/lyme/index.html>
- [2] *Lyme Disease Data and Surveillance*. (n.d.). Center for Disease Control. Retrieved December 26, 2020, from <https://www.cdc.gov/lyme/stats/humancases.html>
- [3] Steere, A. C., & Sikland, V. K. (2003, June 12). *The Presenting Manifestations of Lyme Disease and the Outcomes of Treatment*. *The New England Journal of Medicine*. Retrieved December 26, 2020, from <https://www.nejm.org/doi/pdf/10.1056/NEJM200306123482423>
- [4] Robert B. Nadelman, John Nowakowski, Gilda Forseter, Neil S. Goldberg, Susan Bittker, Denise Cooper, Maria Agueró-Rosenfeld, Gary P. Wormser, The clinical spectrum of early lyme borreliosis in patients with culture-confirmed erythema migrans, *The American Journal of Medicine*, Volume 100, Issue 5, 1996, Pages 502-508, ISSN 0002-9343, [https://doi.org/10.1016/S0002-9343\(95\)99915-9](https://doi.org/10.1016/S0002-9343(95)99915-9).
- [5] Martin E. Schriefer, Lyme Disease Diagnosis: Serology, *Clinics in Laboratory Medicine*, Volume 35, Issue 4, 2015, Pages 797-814, ISSN 0272-2712, ISBN 9780323402545, <https://doi.org/10.1016/j.cll.2015.08.001>.



- [6] Shapiro, E. D. (2014, May 1). *Lyme Disease*. The New England Journal of Medicine. Retrieved December 26, 2020, from <https://www.nejm.org/doi/full/10.1056/nejmcp1314325>
- [7] Biesiada, G., Czepiel, J., Leśniak, M. R., Garlicki, A., & Mach, T. (2012). Lyme disease: review. *Archives of medical science : AMS*, 8(6), 978–982. <https://doi.org/10.5114/aoms.2012.30948>
- [8] Chinmoy Bhate, Robert A. Schwartz, Lyme disease: Part I. Advances and perspectives, Journal of the American Academy of Dermatology, Volume 64, Issue 4, 2011, Pages 619-636, ISSN 0190-9622, <https://doi.org/10.1016/j.jaad.2010.03.046>.
- [9] *Lyme Disease Rashes and Look-alikes*. (n.d.). Center for Disease Control. Retrieved December 26, 2020, from [https://www.cdc.gov/lyme/signs\\_symptoms/rashes.html](https://www.cdc.gov/lyme/signs_symptoms/rashes.html)
- [10] Lipsker D, Lieber-Mbomeyo A, Hedelin G. How Accurate Is a Clinical Diagnosis of Erythema Chronicum Migrans? Prospective Study Comparing the Diagnostic Accuracy of General Practitioners and Dermatologists in an Area Where Lyme Borreliosis Is Endemic. *Arch Dermatol*. 2004;140(5):620–621. doi:10.1001/archderm.140.5.620
- [11] Draelos, R., Ph.D. (2019, April 13). *The History of Convolutional Neural Networks*. Glassboxmedicine. Retrieved December 26, 2020, from <https://glassboxmedicine.com/2019/04/13/a-short-history-of-convolutional-neural-networks/>
- [12] Čuk, E., Gams, M., Možek, M., Strle, F., Čarman, V. M., & Tasič, J. F. (2014). Supervised visual system for recognition of erythema migrans, an early skin manifestation of lyme borreliosis. *Strojniški vestnik-Journal of Mechanical Engineering*, 60(2), 115-123.
- [13] Philippe M. Burlina, Neil J. Joshi, Elise Ng, Seth D. Billings, Alison W. Rebman, John N. Aucott, Automated detection of erythema migrans and other confounding skin lesions via deep learning, *Computers in Biology and Medicine*, Volume 105, 2019, Pages 151-156, ISSN 0010-4825, <https://doi.org/10.1016/j.combiomed.2018.12.007>.
- [14] *Treatment for erythema migrans*. (n.d.). Centers for Disease Control and Prevention. Retrieved November 3, 2020, from <https://www.cdc.gov/lyme/treatment/index.html>
- [15] *Treatment for Ringworm*. (n.d.). Centers for Disease Control and Prevention. Retrieved November 13, 2020, from <https://www.cdc.gov/fungal/diseases/ringworm/treatment.html>
- [16] *Pityriasis rosea*. (2020, August 5). Mayo Clinic. <https://www.mayoclinic.org/diseases-conditions/pityriasis-rosea/diagnosis-treatment/drc-20376410>
- [17] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9, 611–629 (2018). <https://doi.org/10.1007/s13244-018-0639-9>
- [18] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

- [19] *Progressive Web App for Offline Image Classification with TensorFlow.js* [Video]. (2019, October 4). Youtube. <https://www.youtube.com/watch?v=DmII0Dlr6iQ&t>
- [20] *Train and deploy on-device image classification model with AutoML Vision in ML Kit*.(n.d.). codelabs.<https://codelabs.developers.google.com/codelabs/automl-vision-edge-in-mlkit#3>
- [21] Zoph, B., Vasudevan, V., & Le, Q. V. (2018). *Learning Transferable Architectures for Scalable Image Recognition* (J. Shlens, Ed.). Cornell University. <https://arxiv.org/abs/1707.07012>
- [22] *Wolfram neural net repository*. (2019, July 17). <https://resources.wolframcloud.com/NeuralNetRepository/resources/MobileNet-V2-Trained-on-ImageNet-Competition-Data#:~:text=Released%20in%202018%20by%20researchers,between%20the%20thin%20bottleneck%20layers>
- [23] He, K., hang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. CVPR 2016. [https://openaccess.thecvf.com/content\\_cvpr\\_2016/html/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html)
- [24] *About ImageNet*. (n.d.). ImageNet. <http://image-net.org/about-overview>
- [25] M. I. Ul Haque and D. Valles, "Facial Expression Recognition Using DCNN and Development of an iOS App for Children with ASD to Enhance Communication Abilities," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 2019, pp. 0476-0482, doi: 10.1109/UEMCON47517.2019.8993051.
- [26] Jayawant N. Mandrekar, Receiver Operating Characteristic Curve in Diagnostic Test Assessment, *Journal of Thoracic Oncology*, Volume 5, Issue 9, 2010, Pages 1315-1316, ISSN 1556-0864, <https://doi.org/10.1097/JTO.0b013e3181ec173d>.