Journal of Research High School

Volume 2022 (1) 86 - 93
ISSN 2688-3651 [online]

The Effect of Retinal Scan Image Resolution on the Performance and Accuracy of Deep Learning Models

Saaketh Madabhushi[1] * and Anish R. Roy[2]

[1]Mission San Jose High School, Fremont, CA USA
[2]Stanford University, Stanford, CA, USA

Received November 18, 2021; Revised April 4, 2022; Accepted, April 13, 2022

Abstract

Deep learning approaches have increasingly been used in the diagnosis of disease and other image recognition problems. Training a neural network requires a high amount of computational power. An assessment of the performance of neural networks on less than optimal image sets and hardware is desirable for these less developed nations. This research seeks to test the performance of neural networks under non-ideal conditions. Examples include fewer resources and training data for the neural network to utilize or outdated hardware. A dataset of images of diabetic retinopathy, with five different levels of severity, was procured. These images were split into four different test resolutions, none of which were the highest possible resolution images of the retinal scans available in the dataset, before being trained across three different neural network architectures: LeNet, AlexNet, and Encoder-Decoder. Images of diabetic retinopathy were utilized specifically as these are more complex images and would allow for the discovery of the limitations of deep learning across different resolutions. Furthermore, after conducting research, it was observed that AlexNet performed the best overall and achieved the highest accuracy of 57.2%. Encoder-Decoder was able to achieve 50.2% and LeNet was able to achieve 51.3%. In addition, the time until convergence for all three neural networks varied depending on the amount of layers and depth of each neural network, as well as the image resolution inputted. In general, neural networks trained with higher resolution images had higher accuracy, but still did not reach optimal performance.

Keywords: Deep Learning, Convolutional Neural Networks, Diabetic Retinopathy

1. Introduction

With the turn of the twenty-first century, the emergence of neural networks (NNs) are increasingly important in day-to-day life. Put simply, a neural network is a computer algorithm that uses multiple data points to try and find a specific pattern amongst the data in order to complete a certain task. The main goal of NNs is to develop algorithms so that machines can complete tasks normally attributed to humans For example, NNs can identify and classify different objects in images. As machines are given access to more data they are able to make more accurate predictions using complex algorithms. The main benefit that an NN provides is increased efficiency and ability to solve complex problems. Examples include autonomous driving and medical image analysis. NNs can be used in driving for geographic mapping and vehicle detection, as well as any other landmarks, potential hazards, or road signals (Huang & Chen, 2020). In terms of medical image analysis, most medical diagnostic and image

* Corresponding Author
saakethraj@gmail.com

Advisor: Anish R. Roy
arroy@stanford.edu

interpretation have been performed by human experts. With the rate of progress of computational medical image analysis, NNs have the potential to be extremely effective in this field, especially in terms of time saved for medical professionals (Shen, et al., 2017).

An array of real world problems can be solved using one specific type of NN: a Convolutional Neural Network (CNN), a subclass of Artificial Neural Networks (ANN). To better understand the layers of a CNN, we must first understand how ANNs are structured. ANNs, which are many neurons (basic units of NNs) connected to each other. A neuron takes the weighted sum of the input and adds a bias to produce one output (Zhou, 2019). All artificial neural networks have an activation function, which turns a bounded input into a predictably formed output. In addition, activation functions modify the input allowing for non-linear problems to be solved. Inputs from neurons are passed down to other neutrons, and this process continues until the desirable output is obtained. Many artificial neural networks have hidden layers, which are any layer between the input and the output (LeCun, et al., 2015). In order to train a neural network, a dataset must be provided for the neural network to “practice” on so that it can fine tune the appropriate weights of the kernels and nodes in the neural network. In order to conduct this training, an adequate hardware setup must be available. However, not everyone has access to such hardware.

When large, detailed images are inputted into the neural network, it is very resource intensive. The main advantage of CNNs is their ability to reduce the complexity of a given model, which often leads to faster training and is generally more efficient, specifically with regards to images. CNNs can be trained to better understand complex images as more images are fed through the system. By reducing any given image into a form which is easier to process without losing any of the main features of that image in order to get an accurate prediction, a CNN can maintain its efficiency and apply what it learns to a wider range of images. Computer vision is vital when tackling complex issues such as image classification. Typically, convolutional neural networks are used to solve these problems efficiently. However, CNNs involve large and complex data sets for training. In this paper, we wanted to determine the effect of image resolution of the training data set on the performance of CNNs for a specific problem: classifying the disease state of retinopathy.

CNNs generally consist of three main types of layers: convolutional layers, pooling layers and fully-connected (dense) layers (O’Shea & Nash). The first structure in a CNN is the convolutional layer. The first part of this layer is the kernel/filter, which helps to truncate a given input into a form which makes it easier to process a large input of info and to output a more compact, easier-to-understand image/result. The CNN takes the input and convoles it with the kernel and may add a bias. The elements in the kernel are the trainable weights (O’Shea & Nash). The pooling layer is responsible for reducing the size of any given input image. This allows the program to process the image using less computer resources, as it has been scaled down. The pooling layer also makes sure to extract all useful and dominant information from any input, while still allowing the neural network to learn the overall value of any given input (O’Shea & Nash). The most common type of pooling is max pooling. Max pooling takes a given output from the convolutional layer, which includes the key features of a given sector of an image, and records the most dominant part of that sector. Therefore, it cuts out any unnecessary information or “noise” contained in that sector, only returning the most dominant and important feature.

Depending on how complex a CNN is or the inputs are, the number of convolutional and pooling layers’ can vary in any given CNN. The more detail that is needed to be captured from an input, the more convolutional and pooling layers may be added - but at the expense of more computational power being used. A dense layer, or fully-connected layer, is an efficient way to obtain non-image outputs from image datasets. However, fully-connected layers are also very computationally expensive, and therefore, cannot be overused in a CNN model.

Some examples of basic neural network architectures include: LeNet-5 (Figure 1A), AlexNet (also Figure 1A), and Encoder-Decoder (Figure 1B). LeNet-5 is the first major CNN architecture that was developed in 1998. LeNet-5 was a simple architecture that was compact and efficient and was mainly used for letter and number recognition. As it did not have many layers, it was considered a fairly shallow neural network compared to more modern architectures. AlexNet is a more advanced version of LeNet-5. AlexNet consists of eight layers, five of which are convolutional layers and three of which are fully connected (dense) layers. The model won the 2012 ImageNet competition by more than an eleven percent error difference (Gupta, 2020). This was a significant improvement and milestone in the advancement of neural networks. An Encoder-Decoder structure first down-samples images in a dataset to a low resolution, all while using convolutional layers to create a feature map. Then, it up-samples the data back to its original size while running more convolutional layers to add to the creative feature map.

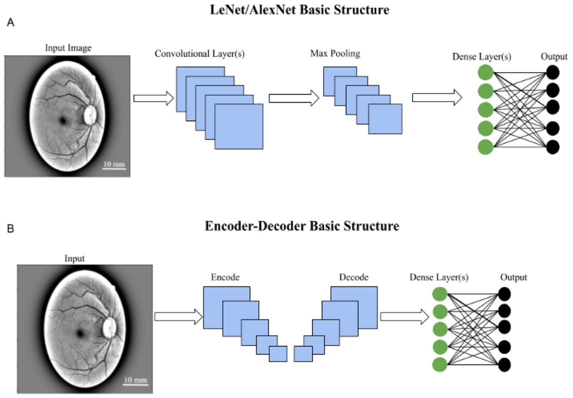

Figure 1. (A) This figure depicts the general schematic of the LeNet and AlexNet CNN architectures. The image is processed through convolutional layers, max pooling layers, and sent to the dense layers before obtaining an output. (B) This figure depicts the general schematic of the Encoder-Decoder CNN architecture. This architecture down-samples and up-samples the image before its information is sent to the dense layer(s) to give an output. The lighter shade of blue highlights the downsampling of the image, while the darker shade of blue highlights the upsampling of the image.

Occasionally, the procurement of good datasets is not feasible. Internationally, datasets are not always easily available, and many datasets have images which are not the best resolution or optimized for machine learning. In order to avoid bias, a dataset should represent the classes that are meant to be predicted (Mestre, 2018). When collecting data of different categories, such as race, gender, or color, it is often difficult to achieve consistency between images in each category, as well as in the dataset overall. Usually, datasets must be stitched together using various online sources and sets (Hooker, 2021). Moreover, adding/finding labels to these images often takes a tremendous amount of time and resources and leads to inaccurate and inefficient datasets. This problem can occur in a dataset of any type when it comes to image classification.Furthermore, not all individuals have access to high performance graphical processing units (GPUs), hardware that drastically decreases time of training, as well as dedicated computers from which to run the neural network. This project simulates the effectiveness of having a less-than-optimal setup for the performance of CNNs.

As the resolution of an image set increases, the computational power required to process the image set also increases. The resolution of the image set is critical as the higher the resolution, the more detail and information the neural network can extract from each given image. To test the effect of non-ideal resolution and hardware on neural networks, we assessed the ability CNNs to predict the severity of diabetic retinopathy when given a retinal scan. Diabetic retinopathy is a diabetes complication that affects the eyes. It is mainly caused by damage to certain blood vessels in the retina. Usually, diabetic retinopathy starts with little to no symptoms which increasingly worsen over time. It is a chronic condition and can last for years or be lifelong. This disease typically affects around 200,000 people in the U.S. per year. Some problems related to diabetic retinopathy include difficulty with vision and perception of different colors (Mayo Clinic, 2021). As stated above, CNNs can be very useful in diagnosis of diseases and other image recognition type situations that may arise. Since an image set of varying eye disease cases is more complex when compared to simple problems such as classifying numbers, it will expose the limitations of neural networks when the resolutions of the datasets are reduced for medically relevant data.

2. Materials and Methods

The data inputted to each neural network was normalized and shuffled. Tensorflow was used in the creation of the three neural networks. The parameters among the three CNNs are similar. All of the nets use an increasing number of convolutions starting with 16 up to 128 in powers of 2. All max pooling layers are of size (2,2) and a stride of 2. Each network had 1024 dense layers with a dropout layer parameterized to 0.2. All CNNs used the Relu activation function and the SoftMax classification function was used in the last layer. We used the Adam optimizer to train. The batch size used was 32. A callback function was used to reduce the learning rate if the validation loss plateaued, with a patience of 2 epochs. Each reduction reduced the learning rate by a factor of 0.2. The initial learning rate used was 0.01. LeNet consists of three convolutional layers and two dense layers. AlexNet consists of five convolutional layers, with two dense layers. Encoder-Decoder consists of three downsampling layers and three upsampling layers, with two dense layers.

To achieve the four different resolutions of our imageset, we used the *resize* function from the CV2 library of Python. This function changes the dimensions of an image and preserves the aspect ratio. As we decrease the resolutions of the images, the pixel size increases. This will help to keep the image set relatively uniform in terms of size while allowing simulation of different levels in dataset image quality. We made sure to label each category of severity in increasing order, with zero being no eye disease and four being severe eye disease.

Our computer is homebuilt and only equipped with 16 gigabytes of random access memory (RAM), as well as 6 gigabytes of GPU memory. The computer setup overall is not specialized for computationally demanding applications. So, our computer was not able to handle image resolution sizes up to 164, as it was not able to allocate enough memory during training for larger image sizes.

3. Results

We chose these three convolutional neural network architectures (LeNet-5, AlexNet, and Encoder-Decoder). The performance of each architecture can be tested when given the same dataset with varying resolutions. (See materials and methods for further details about the nets).

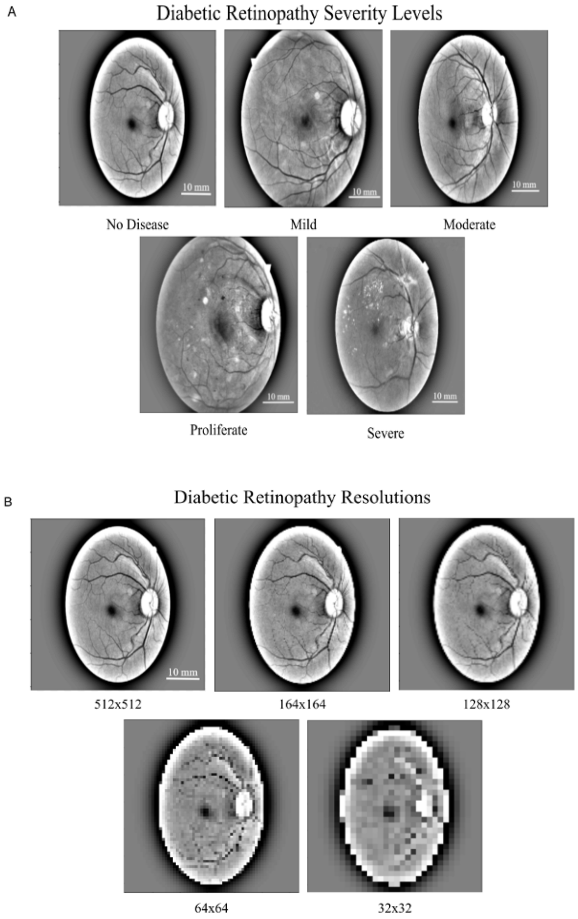

Figure 2. (A) This figure depicts some sample images from the dataset that was used. Each of the images have a 512x512 resolution and depict the different categories of eye disease present in the dataset. (B) This figure depicts some sample images from the dataset that was used. This figure depicts the different resolutions tested in this project using the same sample eye image which is from the same scan of an eye that does not have disease.

We procured our dataset of various eye scans with variations in severity of diabetic retinopathy to train the nets from Kaggle, a popular website where many datasets from a range of genres can be found (Tandon, 2021). This dataset has roughly 35,000 images, as well as individual training sets, testing sets, and validation sets within the superset. All images are 512x512 base resolution (Figure 2A). All images are formatted in gray-scale and are organized into five different categories based on the severity of diabetic retinopathy present in the given eye. These categories are no diabetic retinopathy, mild, moderate, proliferate, and severe. Each of the categories in the training set has 7,000 images. Each of the categories in the validation set has 2,000 images that were not used in training. We worked with four different resolutions: 32x32, 64x64, 128x128, and 164x164 (Figure 2B).

In terms of LeNet, as the image resolution increased, the validation accuracy tended to increase. However, the validation accuracy plateaued around roughly fifty-one percent. It should be noted that LeNet architecture is the oldest of the three architectures on which we chose to base our models. Given the shallow nature of LeNet and the fact that it was created for simpler datasets with less complexity in each image, it was expected that LeNet would not perform up to the same standards as the other two models. Even if the image size was increased up to 512x512, it is not clear whether this model would perform with a similar accuracy to AlexNet and Encoder-Decoder. As shown in Figure 3A and Table 1, the accuracy only gets marginally better (0.5%) when given image inputs 164x164 resolution. It is also noted in the table that as the resolution of the image increases, the time until convergence of each test run of the neural network also increases.

In terms of AlexNet, the validation accuracy increased as the image resolution was increased. As AlexNet is more complex than LeNet in terms of its layers, it is possible that the accuracy could have kept on increasing at even higher resolutions tested in this paper. AlexNet gave the best results for this experiment. As AlexNet is a deeper network with more convolutional and pooling layers compared to LeNet, the higher accuracy is to be expected with training with the base image resolution of 512x512. As mentioned above and shown in Figure 3A and Table 2, the time until convergence for AlexNet increases as the image resolution increases. Since AlexNet is a more complex neural network compared to LeNet, we can see that there is a much longer time until convergence for the two upper resolutions of the images.

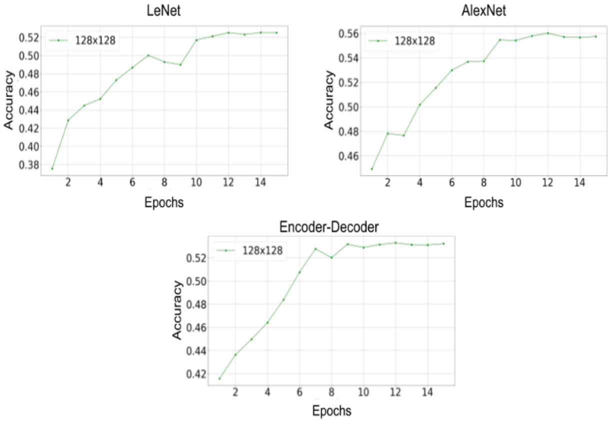

Figure 3A. This figure depicts a line graph of Accuracy v.s. Epochs for each CNN at 128x128 image resolution.

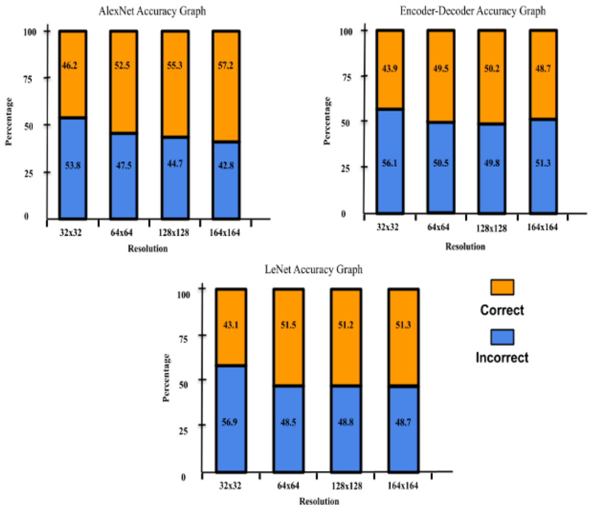

Figure 3B. This figure depicts three segmented bar graphs that each pertain to one CNN architecture. The graphs show how each network performed for each given image resolution. Orange represents the percentage of correctly classified images, and blue represents the percentage of incorrectly classified images. For each network, accuracy tended to increase as the image resolution increased. Accuracy is defined as (number of images classified correctly) / (total number of images in the dataset).

In terms of Encoder-Decoder, the validation accuracy increased up to the 128x128 image and then strikingly slightly regressed when given an input image of 164x164. Likely, this was caused by such a drastic reduction in the image size, as well as only inputting images that are less than half of the original image size. Therefore, the amount of detail that is captured by the network is lowered due to the extremely low resolution of the image. Since the 164x164 image resolution is not a significant difference in image resolution compared to 128x128, the amount of information that the network was able to extract from 164x164 was likely very similar to the 128x128 resolution image. Accordingly, as both image resolutions were fairly similar as inputs for the neural network, the fact that 128x128 performed better is likely due to chance. This likely explains the cause of some of the inefficiencies that occured in this network and why it did not improve as much as AlexNet when provided with a marginally higher resolution image. As mentioned above and shown in Figure 3A and Table 3, the time until convergence for Encoder-Decoder increases as the image resolution increases. We can see that Encoder-Decoder takes the longest time until convergence at all given image resolutions, as it is the most complex of the three neural networks.

Table 1: This datatable gives a context for the time taken by the LeNet CNN architecture to converge with each given image resolution. This table helps to contextualize the line graph measuring the validation accuracy of LeNet over 15 epochs by informing how long in seconds LeNet took to train with each resolution.

Resolution	32x32	64x64	128x128	164x164
Time until convergence (seconds)	75	75	120	195

Table 2: This datatable gives a context for the time taken by the AlexNet CNN architecture to converge with each given image resolution. This table helps to contextualize the line graph measuring the validation accuracy of AlexNet over 15 epochs by informing how long in seconds AlexNet took to train with each resolution.

Resolution	32x32	64x64	128x128	164x164
Time until convergence (seconds)	75	105	255	360

Table 3: This datatable gives a context for the time taken by the Encoder-Decoder CNN architecture to converge with each given image resolution.This table helps to contextualize the line graph measuring the validation accuracy of Encoder-Decoder over 15 epochs by informing how long in seconds Encoder-Decoder took to train with each resolution.

Resolution	32x32	64x64	128x128	164x164
Time until convergence (seconds)	90	135	375	540

4. Discussion

These results all show that when inputting a complex image in a significantly downscaled form, the neural network overall performs worse. In our specific case, the neural networks performed around 10% worse with the smallest resolution compared to the largest resolution. As shown by Figure 3A and 3B, AlexNet performed the best overall. AlexNet was able to achieve the highest accuracy of 57.2%, compared to 50.2% for Encoder-Decoder and 51.3% for LeNet. One problem that we encountered when performing this research is that our computer, with our Nvidia GTX 1660 GPU, was not able to handle image sizes larger than 164x164 resolution. Given that the original image dataset consisted of images of resolution 512x512, even our highest resolution image at 164x164 was significantly downscaled. Therefore, some of our results may be different if this experiment had been performed with a stronger computer that was able to handle image resolutions up to 512x512. As image size increases up to 512x512, it is very likely that the accuracy of these convolutional neural networks also increases.

Additionally, we have considered that, for countries of lower than average socioeconomic status, such as developing countries, there are serious developmental challenges in terms of technology due to a range of factors. These include low levels of investment and low rates of education and skilled labor. This leads to lower levels of diffusion of technology and innovation (Utoikamanu, 2019). This, coupled with low income, results in setbacks similar to those that were exhibited during our research. Based on our data and the findings above, it is seen that a certain level of computer specifications, as well as dataset image quality, must be met in order to gain accurate, meaningful results for neural networks. When using a midrange specification computer with detailed images of resolutions ranging from 32x32 up to 164x164, it is not possible to accurately train a neural network to modern standards (greater than 90% accuracy). As shown by our research, one would need to procure a much more powerful computer with more RAM, a state of the art CPU, and a GPU with more memory and more parallel processing cores (Christensson, 2006; Dettmers, 2020). In addition to high end computer specifications, a large dataset with detailed, colored, high resolution images is also required to achieve high end accuracy.

If, for any reason, both of these necessities are not adequately met, the results may be very subpar. When a dataset is used for a medical or some other professional, important purpose, image resolution and image quality should be as high as possible to ensure the greatest accuracy. Downscaling the image to anything below 50% of the image size will result in accuracy that, at most, can reach up to 60%.

5. Conclusion

Neural networks are useful in a variety of circumstances, from medical disease detection to autonomous driving. In order to tackle more complex issues, computer vision is needed, which requires the use of convolutional neural networks, which use image sets as the input values. There are different architectures of neural networks that were created for different purposes, including the three which we used in our project: LeNet, AlexNet, and Encoder-Decoder. The main purpose of our project is to simulate the effectiveness of having a less than optimal setup when trying to train a convolutional neural network, as we wanted to show the limitations of our hardware and the difficulties of finding adequate datasets. We chose a more complex topic, which is the detection of diabetic retinopathy from a given retinal scan. We specifically chose this topic as classifying disease states from images is highly relevant for application in the medical field. Benchmarking the performance of CNNs on medical images would be very beneficial to doctors and patients. We measured the accuracy of the three neural networks' detection capabilities when given a range of lowered resolutions of images: 32x32, 64x64, 128x128, and 164x164. Based on our findings, as the image resolution increased, so did the accuracy of the neural networks. However, all the neural networks struggled, with the maximum accuracy being only 57.2%. Some issues that we faced included our computer specifications not being powerful enough to handle larger image resolutions, as well as the use of simple architectures, which likely led to underperformance. These results imply that, if both high specification of computer and quality image datasets are not utilized, the neural network will not be properly trained, causing large inaccuracies when compared to modern standards. For countries which do not have the required infrastructure or funding to provide these resources, it will be difficult to achieve good results when using convolutional neural networks. Further experiments need to be conducted to research the lowest possible resolution and least complex hardware required to sufficiently train an accurate network.

Acknowledgements

We would like to acknowledge Pramod Madabhushi and Madhuri Madabhushi for their help editing and looking over this paper through its multiple revisions. They also aided us in the organization and logistics of the project, and the funding for this project.

References

Christensson, P. (2006). *VRAM Definition*. TechTerms. https://techterms.com

Dettmers, T. (2020, September 7). *Which GPU(s) to Get for Deep Learning: My Experience and Advice for Using GPUs in Deep Learning*. TimDettmers.com. https://timdettmers.com/2020/09/07/which-gpu-for-deep-learning/

Gupta, A. (2020, May 17). *Evolution of Convolutional Neural Network Architectures*. Medium. https://medium.com/the-pen-point/evolution-of-convolutional-neural-network-architectures-6b90d067e403

Hooker, S. (2021). “Moving Beyond 'Algorithmic Bias is a Data Problem.’” *Patterns*, 2 (4). https://doi.org/10.1016/j.patter.2021.10024

Huang, Y. & Chen, Y. (2020). “Autonomous Driving with Deep Learning: A Survey of State-of-Art Technologies.” *ArXiv*. https://arxiv.org/abs/2006.06091

LeCun, Y., et al. (2015). “Deep Learning.” *Nature*, 521. doi:10.1038/nature14539

Mayo Clinic. (2021). *Diabetic Retinopathy*. The Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611

Mestre, D. (2018, September 14). *How to Solve the Common Problems in Image Recognition*. Medium. https://medium.com/empathyco/how-to-solve-the-common-problems-in-image-recognition-d519af322bea

O’Shea, K. & Nash, R. (2015). “An Introduction to Convolutional Neural Networks.” *ArXiv*. https://drive.google.com/file/d/1UBBLcqWO9zvXujNuVCaN3y9vUTb3L-M-/view?ts=6122a775

Shen, D., Wu, G., & Suk H. (2017). “Deep Learning in Medical Image Analysis.” *The Annual Review of Biomedical Engineering*, 19. https://doi.org/10.1146/annurev-bioeng-071516-044442

Tandon, K. (2021). *Diabetic Retinopathy Balanced* (Version 1). [Data set]. Kaggle. https://www.kaggle.com/kushagratandon12/diabetic-retinopathy-balanced

Utoikamanu, F. (2019). “Closing the Technology Gap in Least Developed Countries.” *UN Chronicle*, 55 (4). https://www.un.org/en/chronicle/article/closing-technology-gap-least-developed-countries

Zhou, V. (2019, March 5). *Machine Learning for Beginners: An Introduction to Neural Networks*. Towards Data Science. https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9