# Detecting Retina Damage from Speckle Noise Polluted OCT-Retinal Images

**Mingzhe Hu, Minhui Yu, Ye Tian**

Department of Electrical and Computer Engineering

Duke University

Durham, NC 27705

mh511@duke.edu, my136@duke.edu, yt149@duke.edu

## Abstract

In this paper, we first study the classification performance of four different neural network models - a custom CNN5 (5 layers convolutional layers CNN), a VGG16, a MobileNet and a custom VGG on the original OCT-retinal images. Both custom VGG and MobileNet achieve an accuracy higher than 95% on test dataset. We then pollute the retinal images with varying degrees of speckle noise which is one of the main physical parameters that may reduce the imaging quality of OCT devices. On the basis of this, we discuss the impact of different levels of noise on the performance of our classifiers. We find that different models have very different sensitivity towards speckle noise. Additionally, we try to use median filter, Lee filter and autoencoder to eliminate the noise, and research how different denoisers may influence the detection of retina damage.

## Conflicts of Interest

There was no requirement of IRB. The author has no ethical conflicts to disclose. The authors have no conflicts of interest to declare.

## 1 Introduction

Retinopathy is a common cause of visual impairment in patients. There are three most common types of retina damage - Choroidal neovascularization (CNV), Diabetic Macular Edema (DME), and Drusen (see Figure 1). Normal vision occurs when light is focused directly on the retina rather than in front or behind it. CNV is the creation of new blood vessels in the choroid layer of the eye. DME is a compilation of diabetes caused by fluid accumulation in the macula that can affect the fovea.[1] Timely detection of the retina damage can enable patients to suffer from a minimum decline of vision.

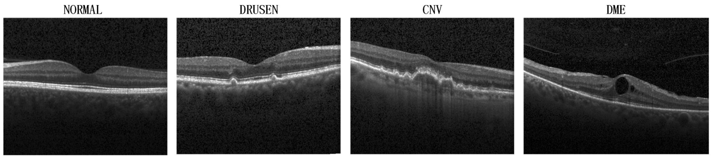


Figure 1: Retinal OCT images.

Optical coherence tomography (OCT) is widely used by ophthalmologists to perform diagnostic imaging on the structure of the anterior eye and retina in which way information about retina damage can be acquired[12].

Up to now, the main method of retina damage detection depends on the individual visual examination of the OCT images, which is time-consuming and requires several years of experience[11]. Many researchers are now developing programs that can automatically detect retina damage from OCT-retinal images. However the process of interpretation of OCT images is restricted to speckle noise which is caused by inevitable multiple forward and backward scattering of light that often obscures subtle but important morphological details[17].

Here, we take a step towards studying the unwanted effect of speckle noise on retina damage detection. We would not simply stop at observing the influence of speckle noise on our classifiers' performance. We also look into possible ways of promoting the detection performance by reducing speckle noise and we will also discuss the impact of denoisers on classification.

## 2 Related Work

Classification of retinal damages using machine learning was attempted by Aleyikov et al. as early as 1998[2]. They trained the system with a modular neural network which achieved a 79% recognition accuracy which is rather low nowadays. After that, with the enhancement of computing power and the development of machine learning, different machine learning algorithms are used to automate the retina damage detection process. Common methods are decision trees, adaBoost, Naive Bayes, Random Forest, SVM and neural networks[3,4,5].

Noise in OCT images is multiplicative in nature and is difficult to suppress due to the fact that in addition to the noise component, OCT speckle also carries structural information about the imaged object[6]. Many researchers began to consider how to remove speckle noise in retinal OCT images to improve usability of some optical noise corrupted datasets. Common ways like nonlinear filters, statistical approach, machine learning methods were tried and their performance varied. Wong et al. used general bayesian estimation to reduce speckle noise in OCT retinal imagery[6]. Lam et al. proposed a speckle reduction method based on contourlet shrinkage[7]. Deep learning methods are also adopted to reduce the noise in OCT[8].

However most of the studies in this area either only focus on using machine learning methods to detect retina damage or only research into reducing speckle noise.Currently almost no paper discusses how denoising speckle noise itself may influence the performance of classifiers on retinal-OCT images.

## 3 Methods

In this project, we first validated that our neural networks approach was feasible so that we can use these models to detect retina damage from original retinal OCT images with an acceptable accuracy higher than 85%. In the second step, we added the physical parameter -- the speckle noise to corrupt retinal images in which way we can simulate the real world situations and test how these models work with the inferior imaging quality. In addition we researched the impact of denoisers on our classifiers.

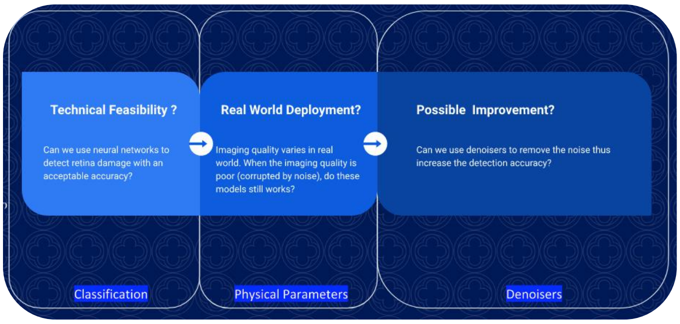


Figure 2: Method section structure

### 3.1 Dataset

We accessed our dataset from kaggle.

(API: kaggle datasets download -d paultimothymooney/kermany2018).

The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (NORMAL, CNV, DME, DRUSEN). There are 84495 OCT images (JPEG) in 4 categories.

### 3.2 Neural Networks

We compared the performance of 4 different models - Custom CNN5, VGG16, MobileNet, Custom VGG. The reasons why we chose these models were simply because these models are popular in image classification tasks and they all had good performance. Also, we focused more on how speckle noise as a physical parameter and the denoiser would impact the performance of models and our main purpose was not picking out a model that can detect the damage best. So we only compared these four models as the basis of next steps.

3.2.1 Custom CNN5

We built our own convolutional neural network model. We applied 5 convolutional layers with 3*3 kernel and relu activations. The number of total trainable parameters was 557636.

3.2.2 VGG16

Researchers from the Oxford Visual Geometry Group or the VGG for short, developed the VGG network, which is characterized by its simplicity, using only 3X3 convolutional layers stacked on top of each other in increasing depth. Reducing the volume size is handled by max pooling. At the end two fully connected layers, each with 4096 nodes, are then followed by a softmax layer[18]. We applied transfer learning and carried out a few changes by removing the top layers and adding our own top layers to fit in our current task (number of output in the last dense layer is 4). The number of total trainable parameters was 6423812.

3.2.3 MobileNet

MobileNet is an architecture which is more suitable for mobile and embedded based vision applications where there is a lack of compute power. This architecture was proposed by Google[19]. We loaded MobileNet with pretrained imagenet weights and excluding the top layers and replacing them with our own top layers. All the convolutional layers are pre-trained, so we froze these layers during training.The number of total trainable parameters was 2626052.

3.2.4 Custom VGG

We customized our own VGG model which also had 16 convolutional layers. There were more trainable parameters compared to the other three models. The number of total trainable parameters was 18267204.

### 3.3 Speckle Noise

Speckle noise is the multiplicative noise that arises due to the effect of environmental conditions on the imaging sensor during the imaging acquisition[20]. Speckle noise is mostly detected in case of medical images, active radar images and synthetic aperture radar (SAR) images. We added the speckle noise to our images by defining our addSpeckle() function following the equation below:

$$Corrupted = \text{Image} + A * N \sim (\mu, \sigma)$$

We set the addSpeckle() function as a preprocessing function in ImageDataGenerator of keras, and retrieved the images by flow_from_directory(). So the speckle noise was added in the preprocessing data before we use different models to predict the test data (see Figure 3). We gradually changed the standard deviation (which is related to the noise power) from 0.02 to 1.28 and the amplitude of noise (which will influence the SNR) gradually from 1 to 8. We then plotted out all the confusion matrices of each model, and also plotted out how their classification accuracies (or loss) change on the same figure to observe how the noise degree will impact the performance of different classifiers.

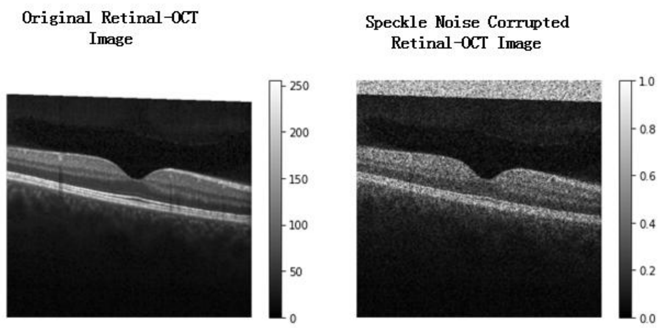


Figure 3: Retinal-OCT Image. Original: left, Corrupted: right

### 3.4 Denoisers

We compared three different methods - median filters, lee filters and auto encoder which represent nonlinear filter approach, statistical approach and machine learning approach respectively. We denoised the speckled noise corrupted images before sending them into our classifiers to predict. And we plotted the resulted classification accuracy (or loss) of those 4 different models at different noise levels on a same figure, we also plotted the classification accuracy (or loss) curve of noise corrupted images on this figure to show how the denoisers may impact the performance of different classifiers at different noise level. The reason why we didn't use SNR (signal to noise ratio) as a direct index is because the different images here had very different signal intensity, it was hard for us to create a dataset that images organized in different SNR groups (each group has the same SNR).

3.4.1 Median filter

The median filtering process is accomplished by sliding a window over the image[21]. The filtered image is obtained by replacing the median of the value of the center of the window. We realized our median filter by calling scipy.ndimage.median_filter().

3.4.2 Lee filter

Lee filter performs noise filtering on an image based on using first order statistics around a prespecified pixel neighborhood. Unlike a typical low-pass smoothing filter, the Lee filter preserves image sharpness and details while suppressing noise[22]. We realized our LeeFilter() function according to the paper, "Digital Image Enhancement and Noise Filtering by Use of local Statistics", written by Lee, Jong-sen[9].

3.4.3 Denoising autoencoder

Intuitively, a denoising autoencoder does two things: try to encode the input (preserve the information about the input), and try to undo the effect of a corruption process stochastically applied to the input of the autoencoder (see Figure 4)[23]. At each noise level ,we used the speckle noise corrupted images as input and the original retinal-OCT images as the label. Then we use this autoencoder to predict the corrupted test data to reduce the corruption process.

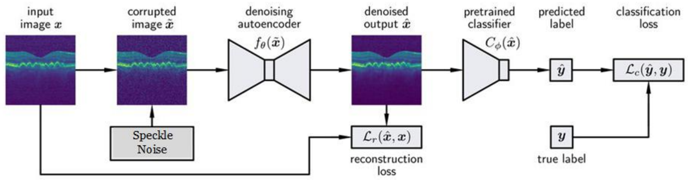


Figure 4: Denoising autoencoder

## 4 Results

We want to stress that we trained the models for multiple times, the curves that we plot here belong to the time the performance of the model was most closely to the average performance.

### 4.1 Retinal damage detection on original images

After training 10 epoches.

4.1.1 Custom CNN5

The training process early stopped at the fourth epoches, and achieved a weighted average 0.94 f1-score. From the confusion matrix, we can see that CNN5 performs better at predicting CNV, DRUSEN and NORMAL compared to DMV (see Figure 5).

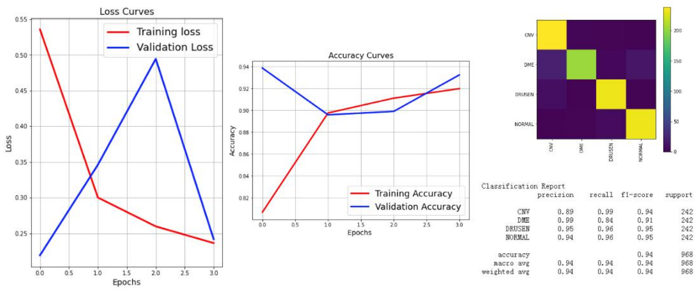


Figure 5: Training history, Confusion Matrix and Classification Report of CNN5

4.1.2 VGG16

After training 10 epoches, and achieved a weighted average 0.94 f1-score. From the confusion matrix, we can see that CNN5 performs better at predicting CNV and NORMAL compared to DMV and DRUSEN (see Figure 6).

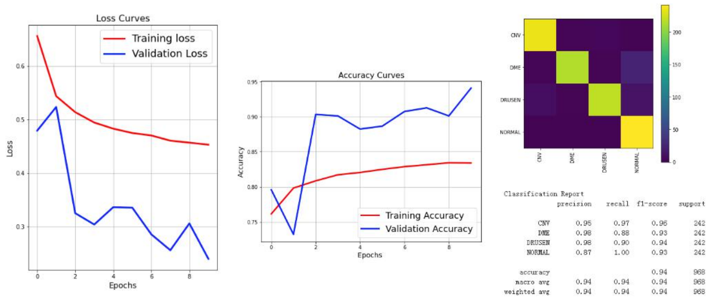


Figure 6: Training history, Confusion Matrix and Classification Report of VGG16

4.1.3 MobileNet

The training process early stopped at the fifth epoches, and achieved a weight average 0.94 f1-score. From the confusion matrix, we can see that CNN5 performs a little bit better at predicting CNV and NORMAL (see Figure 7).

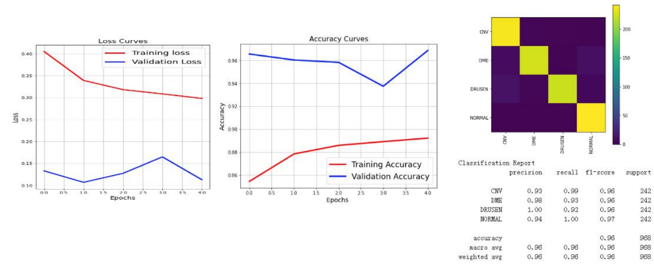


Figure 7: Training history, Confusion Matrix and Classification Report of MobileNet

4.1.4 Custom VGG

The training process early stopped at the ninth epoch, and achieved a weight average 0.99 f1-score. The confusion matrix showed that Custom VGG performs very well at all the four classes when no noise is added (see Figure 8).

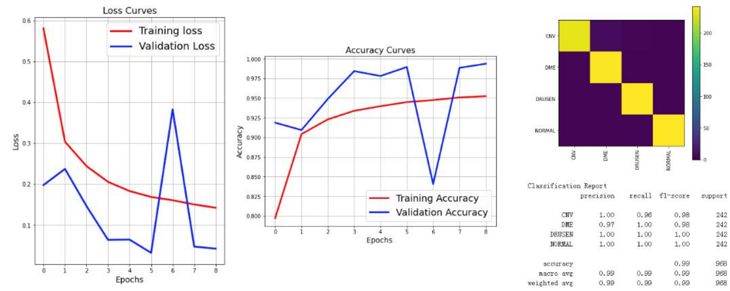


Figure 8: Training history, Confusion Matrix and Classification Report of Custom VGG

Compared to other models, we can see that the custom VGG got the best performance (see Figure 9).

Figure 9: Compare the performance of CNN5, VGG16, MobileNet, CustomVGG with F1-scores

### 4.2 Impact of speckle noise

We can observe that different models have different sensitivity to noise. When speckle noise was added, the classification accuracy of MobileNet dropped most rapidly while custom VGG was the least sensitive to noise. When we added the highest level noise, it still had a rather good performance (0.78). Similar performance on noise free images doesn't mean similar performance at the same noise level. We can find that when sigma (standard deviation) is lower than 0.1,VGG16 performed better than CNN5, with the rising of noise level, it was surpassed by CNN5 and finally performed better than CNN5 again when noise level rose above 0.6 (see Figure 10).

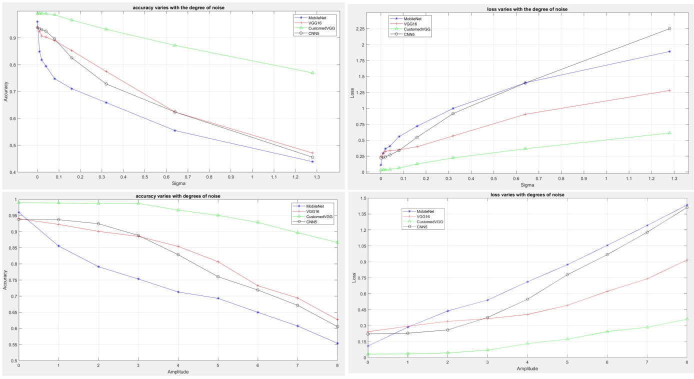

Figure 10: The performance of different model at different speckle noise level (Upper left: accuracy, standard deviation Upper right: loss, standard deviation Bottom left: accuracy, amplitude of noise Bottom right: loss, amplitude of noise)

Speckle noise also impacts the same model's performance at DRUSEN, DME, NORMAL, CNV differently (see Figure 11). We plotted out all the confusion matrices to provide a more direct view of how the increasing noise level is impacting the performance of each model at each category. In VGG16, the performance at DME and DRUSEN are very sensitive to speckle noise. In MobileNet however, the performance at Drusen is most sensitive to speckle noise followed by DME. In CNN5, the performance at DME is most sensitive to noise. In Custom VGG, the performance at CNV is most sensitive to noise.

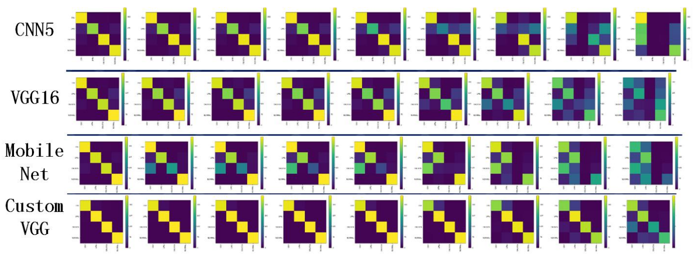


Figure 11: Confusion matrices of models when noise standard deviation rise. $\sigma$ rom left to right: 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.28

### 4.3 Impact of Denoisers

We picked out the model that had the best performance on both original images and speckle noise corrupted images - Custom VGG. We plotted out the accuracy curve of Custom VGG at different noise levels after being denoised by 3 different denoisers (see Figure 12). We can observe that denoisers do not necessarily promote the performance of our model. After being denoised by the median filter, the classification performance is even worse. Lee filter and autoencoder, similarly, lead to a worse performance when noise level is low (below 0.1 aroundfor autoencoder, below around 0.5 for Lee Filter), then slowly promote the performance over the classification performance on noise corrupted images.

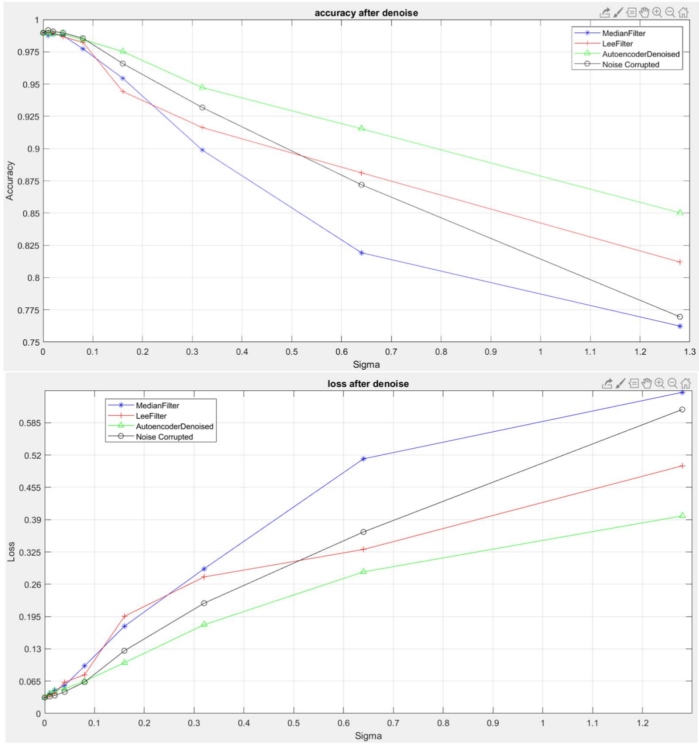


Figure 12: Testing the model on denoised images. Noise standard deviation rise from 0.02 to 1.28) Top: accuracy as metric. Bottom: loss as metric.

## 5 Discussion

The first thing we want to clarify here is that, in section3.1 (classification on original images), as you may observe that the testing accuracy was starting at a very high value even in the first epoch and was even higher than the training accuracy at last.That is caused by unevenly distribution of the number of images in this dataset. There are 83484 images in the training set and only 968 images in the testing set. And the large number of training data made our models achieve impressive performances even in the first epoch. We directly access this dataset through the kaggle api and there are too many images that it was even very hard to call the keras function to resplit the images. We strongly suggest the author of this dataset to resplit the images in a 0.7(train) to 0.3(test) manner.

From the impact of speckle noise on classification performance, we can see the importance of physical parameters (speckle noise here) in an imaging system. When the noise level is high, the deterioration of the classification performance is obvious and the images generated by these imaging systems are useless without processing. We also see that the sensitivities of classifiers are different towards speckle noise. Even for the same model, the sensitivity at different retina damage categories also varies. In real world application, if the speckle noise level is estimable, we suggest users to choose the model that has the best performance at this certain noise level.

Besides using denoisers, there are multiple ways to tackle corruption of speckle noise. We can add noise layers to our models which may lead to better classification performances on noise corrupted images. We can also detect the images with multiple models at the same time and pick out the result that shows the most of the time as predicted.

Among all the three denoisers, the autoencoder got the best performance, however you have to predict the noise level of images you want to detect because every autoencoder is trained with a specific noise level.

In our original recognitions, classification performance of images after denoising should always have a better performance than those images that are corrupted by noise. However we found that we neglect one subtle point. The process of denoising itself will also influence the image quality. That's to say, we can't design any denoiser that only removes the noise pixels and keeps all the image pixels the same. Blurring and distortions are inevitable in this process. When the noise level is low, compared to the noise itself, the denoising process may have a more undesirable effect on the details of images leading to a deterioration of performance. When the noise level is high, the noise is obviously more harmful to the images.

So we suggest that when the imaging quality is fair we can directly use our neural networks to perform the detection. Only when the noise level is high, we should use the denoisers to promote the performance.

There are still many deficiencies in this paper. We have not been able to compare more denoisers which may lead to better denoising performance, nor have we quantitatively analyzed the noise level. We also want to deploy the models to apps so that everyone can use it. We hope to advance it in future research.

## References


[1] Drexler, Wolfgang, and James G. Fujimoto. "State-of-the-art retinal optical coherence tomography." *Progress in retinal and eye research* 27.1 (2008): 45-88.

[2] Aleynikov, Sergey, and Evangelia Micheli-Tzanakou. "Classification of retinal damage by a neural network based system." *Journal of medical systems* 22.3 (1998): 129-136.

[3] Bhatia, Karan, Shikhar Arora, and Ravi Tomar. "Diagnosis of diabetic retinopathy using machine learning classification algorithm." *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. IEEE, 2016.

[4] Lachure, Jaykumar, et al. "Diabetic Retinopathy using morphological operations and machine learning." *2015 IEEE International Advance Computing Conference (IACC)*. IEEE, 2015.

[5] Priya, R., and P. Aruna. "Diagnosis of diabetic retinopathy using machine learning techniques." *ICTACT Journal on soft computing* 3.4 (2013): 563-575.

[6] Wong, Alexander, et al. "General Bayesian estimation for speckle noise reduction in optical coherence tomography retinal imagery." *Optics express* 18.8 (2010): 8338-8352.

[7] Xu, Jianbing, et al. "Speckle reduction of retinal optical coherence tomography based on contourlet shrinkage." *Optics letters* 38.15 (2013): 2900-2903.

[8] Ma, Yuhui, et al. "Speckle noise reduction in optical coherence tomography images based on edge-sensitive cGAN." *Biomedical optics express* 9.11 (2018): 5129-5146.

[9] Lee, Jong-Sen. "Digital image enhancement and noise filtering by use of local statistics." *IEEE transactions on pattern analysis and machine intelligence* 2 (1980): 165-168.

[10] Howard AG,Zhu M,Chen B, et al. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. Arxiv Preprint Arxiv:1704.04861, 2017.

[11] Kermany DS,Goldbaum M,Cai W, et al. Identifying Medical Diagnoses and Treatable Diseases By Image-based Deep Learning[J]. Cell, 2018, 172(5): 1122-1131.

[12] Podoleanu AG,Rogers JA,Jackson DA, et al. Three Dimensional Oct Images From Retina and Skin[J]. Optics Express, 2000, 7(9): 292-298.

[13] Krizhevsky A,Sutskever I,Hinton GE. Imagenet Classification with Deep Convolutional Neural Networks[C]//Advances in Neural Information Processing Systems, [S.l.]: [s.n.], 2012: 1097-1105.

[14] Pan SJ,Yang Q. A Survey on Transfer Learning[J]. Ieee Transactions on Knowledge and Data Engineering, 2009, 22(10): 1345-1359.

[15] Simonyan K,Zisserman A. Very Deep Convolutional Networks for Large-scale Image Recognition[J]. Arxiv Preprint Arxiv:1409.1556, 2014.
[16]https://www.tensorflow.org/js/tutorials/conversion/import_keras
[17]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5437767/
[18] Dodge, Samuel, and Lina Karam. "Understanding how image quality affects deep neural networks." *2016 eighth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2016.
[19] Su, Jiang, et al. "Redundancy-reduced MobileNet acceleration on reconfigurable logic for ImageNet classification." *International Symposium on Applied Reconfigurable Computing*. Springer, Cham, 2018.
[20] Kuan, D. A. R. W. I. N. T., et al. "Adaptive restoration of images with speckle." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35.3 (1987): 373-383.
[21] Brownrigg, David RK. "The weighted median filter." *Communications of the ACM* 27.8 (1984): 807-818.
[22] Baraldi, Andrea, and Flavio Parmiggiani. "An alternative form of the Lee filter for speckle suppression in SAR images." *Graphical models and image processing* 57.1 (1995): 75-78.
[23] Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." *Proceedings of the 25th international conference on Machine learning*. 2008.