DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2019

A comparative study between MLP and CNN for noise reduction on images

The impact of different input dataset sizes and the impact of different types of noise on performance

SANDRO LOCKWALL RHODIN

ERIC KVIST

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

A comparative study between MLP and CNN for noise reduction on images: The impact of different input dataset sizes and the impact of different types of noise on performance

SANDRO LOCKWALL RHODIN, ERIC KVIST

Degree Project in Technology, First Cycle, 15 Credits
Date: June 7, 2019
Supervisor: Pawel Herman
Examiner: Örjan Ekeberg
School of Electrical Engineering and Computer Science
Swedish title: En jämförelsestudie mellan MLP och CNN för brusreducering av bilder: Påverkan av mängd indata och påverkan av val av olika typer av brus på prestanda

Abstract

Images damaged by noise present a problem that can be addressed by performing noise-reduction using neural networks. This thesis analyses the performance of two different neural networks, a Mulilayer Perceptron (MLP) and a Convolutional Neural Network (CNN), when performing noise reduction on images. Specifically focusing on the impact of the size of dataset used to train the two different kinds of neural networks has on the performance, as well as how well these two networks perform when reducing different types of noise. This in an attempt to determine whether the use of the more modern type of network, the CNN, performs better than the older type of network, the MLP, specifically for image noise reduction. The results show as expected that the MLP performs worse than the CNN, also that the impact of the size of the dataset and choice of noise to be reduced is, albeit of great impact on the performance, not as important as the choice of neural network.

Sammanfattning

Bilder som är utsatta för brus är ett problem som kan adresseras genom att utföra brusreduktion med hjälp av neurala nätverk. I denna studie analyseras effekt-skillnader i brusredusering av bilder för två olika typer av neurala nätverk, en Multilayer Perceptron (MLP) och ett konvolutionellt neuralt nätverk (CNN). Fokus ligger specifikt på hur indatans storlek under träningen, är påverkad av två olika typer av neuronnätverk samt hur bra dessa två neurala nätverk presterar när de reducerar olika typer av brus. Detta i ett försök att avgöra om användningen av den modernare typen av nätverk, CNN har högre prestanda än den äldre typen, MLP för brusreducering. Resultaten visar som förväntat att MLP:n fungerar sämre än CNN:n, också att effekten av indatans storlek och valet av brus att reduceras är, trots att de båda har en stor inverkan på prestandan, inte lika viktigt som valet av neuralt nätverk.

Contents

Chapter 1

Introduction

In today's society the presence of photography is recurrent in a vast amount of areas. It encompasses records of medical patients surgeries [1], footage of criminals committing a crime, individuals uploading selfies on social media and pictures of people's childhoods, among other things. The cameras used for taking image vary in quality and are always subject to some degree of noise. Sometimes this is due to the fact that the images are taken in low-light environments or because of the electrical circuitry inside of the camera is interfering with its sensor.[2] Three common types of noise are Gaussian and Poisson, which are both common in cellphone taken photos, and Speckle, which is more common in medical imagery. [2][3][4]

The importance of obtaining a noise-free image can be crucial for a surgeon to proceed in the right way, for the police to obtain solid evidence in order to perhaps apprehend the right people, or for an individual to capture every single detail in their selfie without things becoming distorted. What can then be done with noisy images? The answer is noise reduction. Noise reduction is the removal of noise through various means. Gaussian noise can for example be reduced by adaptive smoothing techniques [5] or by using neural networks, as in this study. [6]

A neural network is a brain-based machine-learning method for computers to simulate the functions of a system of connected neurons.[7] Neural networks are often constructed out of three types of layers: input layer, hidden layer(s) and output layer. Each layer is build on an arbitrary amount of nodes that are connected with other nodes from different layers.

Two commonly used types of neural networks are the Multilayer Perceptron (MLP) and the Convolutional Neural Network (CNN). A MLP is a standard node-based neural network with a minimum of three layers, while a CNN is a network that is constructed without a node-based structure, but rather with a matrix-based structure. CNNs are made to keep a three dimensional perspective in mind. In turn CNNs are often used for image recognition and for image processing tasks.[8] [9]

Although CNNs were designed during the 1980s, they were because of hardware limitations rarely utilized before 2010.[10] On the other hand, neural networks such as MLP have used for noise reduction before 2010, although not necessarily on very complex images.[11] In turn it would be interesting to see how an older type of neural network, an MLP compares to a more modern type of neural network, a CNN.

1.1 Research Question

The aim of this thesis is to investigate if a deep [12] CNN outperforms a shallow [12] MLP when performing noise reduction on images. The specific questions that will be investigated are:

- What impact does the amount of files used for training have on the performance between the neural networks?
- What impact does the type of noise to be reduced have on the performance between the neural networks?

1.2 Scope

- This thesis limits itself to the comparison between two kinds of neural networks, each with a specific configuration. The MLP being constructed as a shallow network with only 3 layers, while the CNN in constructed to be deep with 8 layers in total.
- The training of the neural network is done with dataset sizes no larger than 9000, 100x100 (resolution), images. This due to the lack of powerful hardware and the desire to load all images into RAM.
- The neural networks are trained on three different types of noise, particularly the commonly found Gaussian Noise, the often less trained on Pois-

son noise and the medical imagery noise, Speckle. These three types of noise distributions are primarily chosen because of their commonness in photography and because they can all be applied to images utilizing the scikit-image Python module.

Chapter 2

Background

2.1 Supervised Learning

Supervised learning is one major approach commonly used while training neural networks. It involves having a set of data as well as a representative set of data which matches the other. The way a supervised neural network works is that an input is fed into the neural network, and a result is expected on the other end. The neural network is then essentially altered based on how well the network approximated the result. [13]

2.2 Artificial Neural Network

An artificial neural network is essentially a weighted graph that takes an input and presents some kind of output. This based on the workings of the human brain [7]. A very good example to understand the way in which the simulation of the brain's neurons are related to computer science is by looking at the images below. [14]. We can clearly see that they all represent the number 3, even though they are not identical to each other. But how is this possible? How are we capable of perceiving three different images as the same? The answer is that our brain has previous knowledge on how the number 3 is constructed, due to the fact that we have seen it represented in thousands of different ways and shapes in multiple occasions. We identify two arcs, where one is on top of the other. Or in mathematical terms: two functions of degree 2 that are turned 90°, where one lays on top of the other. This means that every time we see this pattern we identify it as a 3. Thus, the idea behind ANN is to train the neural network with an arbitrary amount of data so that it is then able to identify a valid output.

Figure 2.1: Three representations of the number 3

2.2.1 MLP

A MLP is a widely spread neural network that is used for *supervised learning* as well as for speech and image recognition[8]. A typical MLP architecture (Figure 2) is composed of three main parts: an input layer, an output layer and a couple of hidden layers, that can be seen as a black box. Every node in the figure represents a neuron of the neural network, where every neuron is assigned a specific value. Figure 2 should be seen as a feed-forward network. This means that the information is single directed: the signal goes from the input layer, through the hidden layers, and ending in the output layer, where, for every node n_i in the i^{th} layer there is an edge to all the other nodes n_j in the $(i+1)^{th}$ layer. Once the input signal has arrived to the output layer, the MLP learns by using backpropagation.

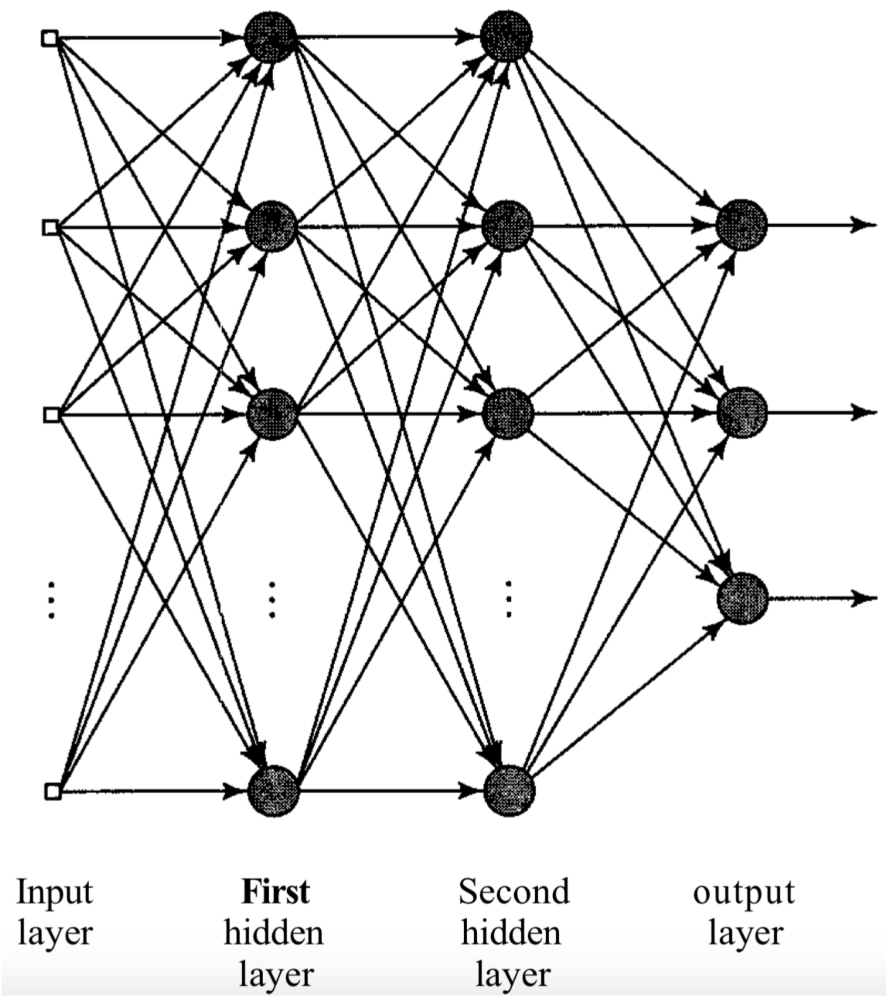

Figure 2.2: Architectural graph of a MLP with two hidden layers[15]

2.2.2 CNN

A CNN is a type of neural network that is primarily used for image-based input.[9] This is due to the fact that CNN performs well when utilizing and recognizing spatial features, such as objects and edges.[16] A CNN will likely be made out of several different layers, including Convolutional-, Pooling-, and the Fully Connected layer types.

In Figure 3, the Fully Connected layer is a layer that behaves much like a standard layer in a MLP. [17] The Fully Connected layer is mostly used for classification tasks, and is in turn not relevant for this thesis. However, the other two types of layers in Figure 3, the Convolutional- and the Pooling layer are. As this thesis focuses on images, the following descriptions will be focused on Convolutional layers for image input.

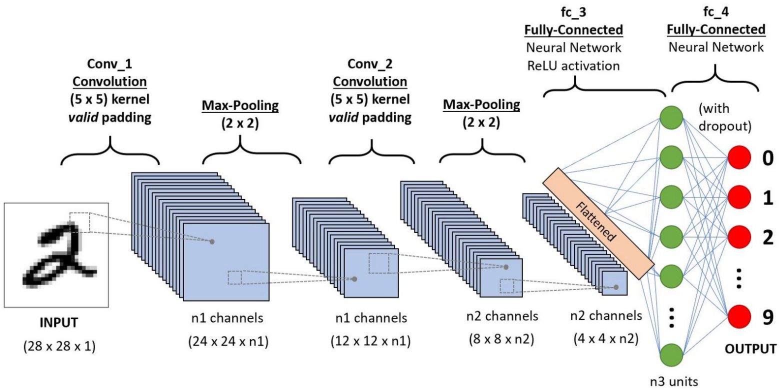

Figure 2.3: Structural representation of a CNN CNNEli5

A Convolutional layer is a layer which takes input and represents it in a different way utilizing filters. Filters are essentially iterators that stride through the input and presents it as a new, smaller, partially merged output. The output may or may not be padded. If the output is padded it will have the same dimensions as the input. The padding is done by filling the parts not existing originally in the output with for example zeroes.[9] Each convolutional layer may have many filter representations per layer, producing an output. When dealing with images the amount of filter representations is equal to the output's depth.[18][19]

A Pooling layer is one which takes input and reduces its spatial dimensions to make it easier to process. Some common examples of pooling layers include MaxPooling or AveragePooling. The process of reducing the dimension works much like the iterations of the filters in the convolutional layer. In both cases, it is done by either taking the maximum or average value across one part of the input, reducing the input per dimension of depth based on the specified pooling size.[18][19]

2.3 Hyperparameters

Hyperparameters are the static parameters that can be altered before training a neural network, this includes parameters such as the amount of nodes in the hidden layers of a neural network [20], the specified optimizer or the activation functions used. Hyperparameter tuning (altering) is important to achieve better results[20].

2.4 Noise

Noise is for the purpose of this thesis considered to be *irrelevant* information that obscures *relevant* information.[21] Noise can occur during the capture, storage or retrieval of data because of the used hardware's malfunctioning.[2] In digital imagery there are multiple types of noise. Three of them are Gaussian, Poisson and Speckle. These noises are shown on the pictures below.

Figure 2.4: A clean image extracted from the dataset.

Figure 2.5: The clean image corrupted with Gaussian noise.

Figure 2.6: The clean image corrupted with Poisson noise.

Figure 2.7: The clean image corrupted with Speckle noise.

2.4.1 Gaussian

Gaussian noise is normal distributed noise. In images it is often fore-coming in pictures taken in poor light conditions. However, it may also appear because of electrical interference and overheating hardware[2].

2.4.2 Poisson

Possion noise is one of the more dominating sorts of noise in digital imagery. It is dependent on the amount of photons per pixels and can arise because of poor quality image sensors. [3]

2.4.3 Speckle

Speckle noise in images is noise caused by environmental impacts on the sensor while capturing the image itself. It is commonly found within medical imagery and within active radar images.[22]

2.5 Previous research

The area is not without previous research. The study *A Convolutional Neural Network's Denoising Approach for Salt and Pepper Noise* performs noise reduction on a total of 97 images with different density of Salt and Pepper Noise. This is done by using an own implemented CNN and comparing it to other state-of-the-art neural networks, such as MLP. In this study the authors compare the efficiency of their CNN implementation with other neural networks by visually comparing the results of the noise reduced image. The drawn conclusion from this study is that their own implemented CNN outperforms any other used neural network, inlcuding MLP, when it comes to Salt and Pepper noise-reduction on images. Although Salt and Pepper noise is not quite the same as Gaussian, Poisson or Speckle, it shows that noise reduction can indeed be performed utilizing CNNs.[23]

On the other hand, a study performed at Universidad Politecnica de Valencia showed that MLPs could also be used to perform high quality noise reduction. In the study, this was specifically on scanned images for handwritten recognition tasks. The noise-corrupted images were compared to the clean images by comparing the mean squared error for every pair of images. The mean squared error values obtained in this study varied between 0.0001 and 0.0008, meaning that the images the network put out were almost identical to the originals.

Another study on noise reduction using neural networks was performed by Aoija Zhao at Stanford University. It is another example of the presence of CNNs when noise reducing images. The neural network was trained with different amount of images, all with the same density of Gaussian noise. The maximum dataset that was used to train the neural network was 100 000 images with a resolution of 64x64x3 pixels. The results showed a significant performance increase when the amount of training images were greater than 50 000. [24]

Chapter 3

Methods

The following chapter is divided into six parts where the process for obtaining the results are described. Sections 3.1 and 3.2 explains how the used data for training the neural networks was manipulated and altered. Sections 3.3 and 3.4 describes more in depth about the construction and architecture of the used ANN as well as the used hyperparameters. Finally sections 3.5 and 3.6 reflects how the results were measured and compared.

3.1 Data Utilization

The *Figure Eight Open Images Challenge 2018 Test* set was identified as a space-efficient set of approximately 10 GB of data, containing 99999 images. It was chosen because of hardware limitations, particularly because of the lack of disk-space on the computers this research was conducted on.

The contents of the *Figure Eight Open Images Challenge 2018 Test* set included pictures of different resolutions. In turn the entire set was re-scaled to pictures of resolution 100 x 100 pixels. This increased the space efficiency RAM-wise and made it possible for the neural networks to only have to work on pictures of the same size. Due to the limitations in RAM, only a maximum 10,000 of these 99,999 images were ever used at the same time. With the dataset sizes being chosen as 90, 900, and 9000 paired with a validation set size of 10, 100, and 1000 respectively.
The re-sized images in the datasets were imported as Numpy arrays (essentially matrices) and divided by 255 in order to normalize the array data, representing the values between 0 and 1. The arrays were then reshaped into 3 dimensional structures of the type 100x100x3, this as each 100x100 image contains three

layers, one for each color code, R(ed), G(reen) and B(lue). The arrays were then reshaped to fit the input requirements of the neural networks. At last the arrays were merged and split into train and validation data accordingly.

3.2 Noise Introduction

The training- and test set were duplicated and corrupted with the three different used noises. The noise introduction was performed on these duplicates utilizing the scikit-image Python module. The noise applied to the images was randomly chosen, producing a randomly distributed amount of noise across the training and test sets.[25]

3.3 Neural Network Construction

The neural networks were constructed based on Keras tutorials on how to construct image denoisers as well as tutorials from the Microsoft Cognitive Toolkit, then structured to function based on the limited hardware used. [19] [26]

3.3.1 MLP

The MLP was constructed with one input layer, taking a one dimensional input of size 30,000, covering each pixel value of the 100x100 matrix and including each value for each color layer. The second layer was a compression layer of 1200 nodes, that compressed the input down to 4% of its original size. Finally a reconstruction layer of 30,000 nodes was used to produce a reconstruction of the image fed into the neural network.

3.3.2 CNN

The CNN was constructed to take a three dimensional input (in practice it was a four dimensional input due to multiple images being presented at once in batches) in the form of 100x100x3. The actual structure of the network is presented below.

- The input layer, taking input in the form 100x100x3 and outputting a 100x100x3 tensor.

- A padded convolutional layer with 128 filters, with the filter size 3x3. (Outputting a 100x100x128 tensor)
- A max pooling layer. (Outputting a 50x50x128 tensor)
- A padded convolutional layer with 64 filters, with the filter size 3x3. (Outputting a 50x50x64 tensor)
- A padded convolutional layer with 64 filters, with the filter size 3x3. (Outputting a 50x50x64 tensor)
- An upsampling layer of two, which simply duplicates all values along the rows and columns of the matrix. (Outputting a 100x100x64 tensor)
- A padded convolutional layer with 128 filters, with the filter size 3x3.(Outputting a 100x100x128 tensor)
- A padded convolutional layer with 3 filters, with the filter size 3x3. (Outputting a 100x100x3 tensor, matching the 100x100x3 output desired)

3.4 Hyperparameters

The networks were trained utilizing the activation function ReLU, except the final layer that utilized the Sigmoid activation function. The reason for the Sigmoid activation function on the final layer was to potentially allow other activation functions to be used in the previous layers, although this was not done.

As for the optimizer, the Adam optimizer was the chosen loss optimizer utilized during the training of the constructed neural networks. This was as the Adam function is one of the more commonly used optimizers, and in turn the results may present information useful to a greater amount of individuals.

As for the rest of the used hyperparameters, the learning rate was set to the default of the Adam optimizer in Keras (0.001). The batch size was set to 1/9th of the size of the training dataset when possible. The CNN could however not be trained with batch sizes above 50.

3.5 Obtaining Results

Once the neural networks had been trained, they were provided with new validation data, this in the form of 1000 new noisy images, never seen before by the network. The same original images were used for all validations, however four different noise profiles were used for each dataset of 1000 corrupted images: Gaussian, Poisson, Speckle and Mixed (a dataset where the images each had been corrupted with one of the three previously mentioned noises). Each of the noise profiles producing 1000 new Mean Squared Error values. These values were then averaged to produce results that enhanced easier reading comprehension. This process was then repeated for each size of datasets that the neural networks were trained with.

3.6 Statistical Analysis

A 3-Way ANOVA statistical test was used for determining the effect of the three independent variables: Artificial Neural Network, Size of Dataset and used Noise, on the performance of the output variable: Mean Squared Error. The effect was quantified by looking at the Partial Eta Squared value and the significance level used was set at 0.05 because of the low size of dataset. As of the impact of the effect size, it was based on Jacob Cohen's values of impact of effect size, with a small effect being a Partial Eta Squared of 0.0099, a medium of 0.0588, and large if above 0.1379.[27]

In order to determine the impact of the independent variables on the noise reduction on images, the following null hypotheses were tested:

- $H_{0_{ANN}}$ = The used artificial neural network has no impact on the performance.
- $H_{0_{SD}}$ = The used size of dataset has no impact on performance.
- $H_{0_{N}}$ = The used noise to validate has no impact on performance.
- $H_{0_{ANN*SD}}$ = The used artificial neural network AND the size of dataset has no impact on the performance.
- $H_{0_{ANN*N}}$ = The used artificial neural network AND the choice of noise has no impact on performance.

- $H_{0_{SD*N}}$ = The used size of dataset AND the choice of noise has no impact on performance.

- $H_{0_{ANN*SD*N}}$ = The used artificial neural network AND the size of the dataset AND the choice of noise has no impact on performance.

Chapter 4

Results

The results are presented in three sections. The first section *4.1 Mean Squared Error Across Validation Data* shows, for both MLP and CNN, the average value of the mean squared error of the validation data for the different sizes of datasets as well as for different noises. The sizes of datasets include 90, 900, and 9000. The different types of noise include Gaussian, Poisson, Speckle, and Mixed, where Mixed refers to a dataset filled with a mixture of images, each corrupted by one of the three different types of noise. The second section *4.2 Statistical Analysis* shows a 3-way Anova, where the mse is used as dependent variable and the type of ANN, Size of Dataset and type of noise as fixed factors. The third section *4.3 Visual Representation* shows visually how well the different ANN performed.

4.1 Mean Squared Error Across Validation Data

4.1.1 MLP

The results from the trained MLP are presented in the form of a table of 3 columns and 12 rows. The first column presents the type of noise that is being addressed, the second what size of dataset has been used to train the neural network and the third, the average mean squared error across 1000 validation images.

Noise	Size of Dataset	MSE
Gaussian	90	0.21645
Poisson	90	0.21622
Speckle	90	0.21676
Mixed	90	0.21649
Gaussian	900	0.13928
Poisson	900	0.13764
Speckle	900	0.14423
Mixed	900	0.14007
Gaussian	9000	0.10642
Poisson	9000	0.10542
Speckle	9000	0.11207
Mixed	9000	0.10805

Figure 4.1: Average mean Squared Error based on Size of Dataset and what type of Noise has been used to corrupt the Dataset.

The table reveals that with an increase in the size of dataset, the average performance increases (as measured in Mean Squared Error). However, the increase in performance diminishes across each new increase in size of dataset.

4.1.2 CNN

The results from the CNN follow the same structure as the MLP: the performance increases every time the dataset increases, regardless of type of noise that is being reduced. However, unlike the MLP, the results are not as similar. Specifically the CNN is a significantly better at reducing Poisson noise compared to the other types of noise. For visual representation please see section *4.3 Visual Representation*.

Noise	Size of Dataset	MSE
Gaussian	90	0.02992
Poisson	90	0.01185
Speckle	90	0.03465
Mixed	90	0.02360
Gaussian	900	0.02699
Poisson	900	0.00729
Speckle	900	0.02957
Mixed	900	0.01955
Gaussian	9000	0.02307
Poisson	9000	0.00293
Speckle	9000	0.02286
Mixed	9000	0.01487

Figure 4.2: Average Mean Squared Error based on Size of Dataset and what type of Noise has been used to corrupt the Dataset.

4.2 Statistical Analysis

The results below (Figure 4.3) show the relevant values of the 3-way ANOVA performed. The values include the Source, detailing what kind of Null-Hypothesis is being tested by the 3-way ANOVA. The Significance, detailing whether the Null-Hypothesis is confirmed or rejected. Finally, the Partial Eta Squared, detailing what kind of impact the stated source has on the performance. The values presented are rounded off to four digits.

Source	Significance	Partial Eta Squared
ANN	0.000	0.632
Size of Dataset	0.000	0.187
Noise	0.000	0.012
ANN*Size of Dataset	0.000	0.144
ANN*Noise	0.000	0.007
Size of Dataset*Noise	0.813	0.000
ANN*Size of Dataset*Noise	0.178	0.000

Figure 4.3: Results from 3-way ANOVA performed on the validation data for both neural networks, all dataset sizes and the three different types of noise.

Looking at the second column of Figure 4.3, the significance level, the Null-Hypotheses are rejected for all sources, excluding Size of Dataset*Noise and ANN*Size of Dataset*Noise. This indicates that the combination of Size of Dataset and Noise, as well as the combination of ANN, Size of Dataset and chosen Noise, do not have an impact on the performance when dealing with the ability to reduce noise.

Observing at the other sources, the strongest impact on the performance is the choice of ANN, having a great impact on the performance, based on the value 0.632 being greater than 0.1379. According to Jacob Cohen's indicator of impact,a value with a Partial Eta Squared greater than 0.1379 has a great impact on the performance. When it comes to the choice of Size of Dataset the value for its Partial Eta Squared is greater than 0.1379, although it is only about a third of the Partial Eta Squared of the choice of ANN. The choice of ANN together with the choice of Size of Dataset has a Partial Eta Squared of about 0.144, also indicating a great impact. The choice of noise, although not having a great impact on the performance, does achieve a low impact, as the Partial Eta Squared value of 0.012 is greater than Cohen's limit for something having a low impact (0.099). The remaining source with a rejected null hypothesis, ANN*Noise, does seemingly have an impact on performance, although it is lower than Cohen's limit for low impact performance.

4.3 Visual Representation

This section shows a visual representation of the noise-reduced images after trained by the neural network. The first image, Figure 4.4, is the original image and the second image, Figure 4.5, is one corrupted by Speckle noise. The third and fourth images, Figure 4.6 and Figure 4.7, show the noise reduction performed by the MLP and CNN, respectively, after being trained on 9000 images. For other visual representations of different types of noise or other sizes of the image dataset, please see Appendix A, Figures A.1 to A.24.

Figure 4.4: A clean image extracted from the dataset.

Figure 4.5: The clean image corrupted with Speckle noise.

Figure 4.6: The corrupted image noise-reduced using the MLP trained on 9000 files.

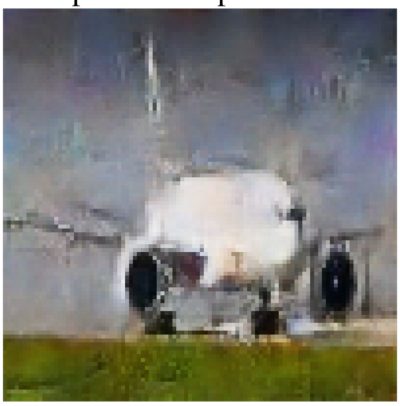

Figure 4.7: The corrupted image noise-reduced using the CNN trained on 9000 files.

Chapter 5

Discussion

5.1 Summary of results

The results show, as expected, that the MLP structure performs worse than the CNN. The increase of the size of the dataset used, decreases the mean squared error for both kinds of networks. As for the choice of noise it seems that there is no bigger impact on what noise is reduced. However the CNN seems to be a lot better at removing Poisson noise than the MLP. The statistical analysis, shows a large impact from both the usage of ANN, the size of the dataset used, as well as the two in combination. This together with a low impact from the choice of noise.

5.2 Size of Dataset

So what is the impact on performance based on the choice of size of dataset? The Partial Eta Squared reveals that the impact is indeed great. Both tables (Figure 4.2 and Figure 4.3) of the results gathered from the model based on a shallow MLP and a deep CNN reveals similar results.

Looking at the statistical analysis it shows that the choice of size of dataset, albeit important, has a lower impact than the choice of neural network. At the same time the source combination of size of dataset and type of neural network has a great impact on the performance. These are expected results, as the more nowadays used type of network, the CNN, is both deeper and made for image processing compared to the MLP.

Furthermore, it makes sense, as seen in the research by Aoija Zhao, that more images being trained on should allow the network to perform better. This explains the increase in validation performance, as shown by the decrease in mean squared errors. Although the performance difference measured in mean squared error is greater with an increase in the size of the dataset, the difference between using 900 and 9000 images may be considered negligible. This as the decrease in Mean Squared Error isn't as drastic between 900 and 9000 images compared to the decrease between 90 and 900 images. This is mostly notable with the MLP, however there are some hints of it within the results based on the CNN as well.

5.3 Impact of Noise

What impact does the type of noise to be reduced have on the performance between the neural networks? The choice of noise does not seem to matter when performing noise reduction with the shallow MLP. This, as the mean squared errors per choice of size of dataset, regardless of choice of noise, is contained within a small range of values (e.g 0.13764-0.14007 and 0.21622-0.21676).

For the CNN however, the results are quite different. The performance varies in regards to what noise the network is reducing. Something unexpected was how well the network performed in the reduction of Poisson noise. Especially with the uniformity of capability of reducing noise as observed with the MLP. With a size of dataset of 90 images, perhaps one could assume, as the assignation of noise was random between each training run (for each network and for each size of dataset), that more images with Poisson noise might have been provided to the network during the training process. However as the results repeat for both 900 and 9000 images, and the fact that it is quite unlikely that Poisson noise was favoured, in the datasets for training, three times in a row. In turn, it seems just to state that Poisson noise is easier to remove than Speckle or Gaussian noise. Perhaps Poisson noise isn't as damaging to the images as the other two types of noise. Something that might seem reasonable when looking at the corrupted images in Appendix A, Figure A.18 (an image corrupted with Poisson noise) compared to Figure A.2 (an image corrupted with Speckle noise). In the image with Speckle noise, the plane can barely be seen, compared to in the image with Poisson noise where the plane can be seen even though the image is noisy.

Taking it to a greater perspective, if one were to utilize this network for noise reduction, it could be great for people with cameras with low quality image sensors. This as they would likely produce images with more Poisson distributed noise, and in turn would have the noise reduced to a higher quality.

5.4 Impact of Neural Network

The choice of neural network does have a great impact on the performance of the noise reduction. There is nothing surprising about this. The MLP, although having been used in the past for noise reduction tasks, is outclassed by the newer generation of neural networks, the CNN. The MLP used was shallow and the CNN used was deep, in turn something that might perhaps be interesting to see is how a shallow CNN performs against a shallow MLP or how a deep MLP performs against a deep CNN. Especially the latter once, since this is something that could not be done on the limited hardware this study was performed on.

Something to point out though, was that the shallow neural network went through quite the compression, nodewise, starting out with 30000 nodes then going to 1200 before returning to 30000. This was as stated before due to hardware limitations, but perhaps the compression from 30000 to 1200 is a bit too much for the network to be able to reduce the noise and still produce a legible image. Although this could also have been solved by allowing the network to train on a greater amount of files. This based on what can be seen in for example Appendix A, Figure A.5 and A.7, where the MLP goes from just showing a somewhat correctly colored blur to the very blurry shape of a plane.

5.5 Limitations

This study has faced some limitations. As for the chosen neural networks, there were only two types of configurations used, MLP and CNN, leaving out other neural networks such as RNN, Spiking NN etc. At the same time, the construction of the ANNs were purposly perfomed quite differently, with the CNN being deep and the MLP being shallow. This makes it hard to generalize the results to all types of MLPs and all types of CNNs.

Furthermore, the some of the used hyperparameters of the networks were not altered whatsoever, including the learning rate of the optimizer Adam, the activation function and the batch size. Perhaps better results could be obtained by tuning the hyperparameters, as this is a step in neural network construction that was left out in this study.

Moreover, something that limits our findings is the fact that the neural networks are only capable of performing noise reduction on 100x100 squares, something that may not be directly useful to the general public when camera phones with an ever increasing megapixel count. Although there is potential for targeted noise reduction, this study only shows a proof-of-concept of the effectiveness on how noise reduction can be performed using neural networks.

There is also the fact that the amount of files used in the training of the dataset can be considered low from a big data perspective. Compared to the study by Zhao at Stanford, the maximum size of dataset used in this thesis is only about a tenth of the maximum used in her study. Thus, a thesis that investigates the impact of size of dataset, a greater amount of configurations of images in the dataset trained on would potentially provide more interesting results. Even though the Mean Squared Error decreased between 900 and 9000 images, there is no solid evidence that the Mean Squared Error would keep decreasing for greater sizes of datasets.

5.6 Future research

As stated in the Limitation section, this thesis is a proof-of-concept on how neural networks can be used to perform noise reduction on images. By utilizing greater hardware and cloud based computing, the amount of images trained and validated on could be increased. The resolution of the images could as well be increased in that way. Apart from this, the networks constructed for this thesis were rather shallow, creating potential for utilizing deeper networks to see if any greater performance improvements could be gained.

Additionally, trying out noise reduction on more types of noises as well as utilizing more types of neural networks to try to perform noise reduction could potentially be an area of interest for future studies. As mentioned before, so could the comparison between two (or more) shallow or deep neural networks, of different types, in the process of performing noise reduction.

Lastly, investigating whether the 1200 nodes in the hidden layer of the MLP is the bottleneck in the performance of the MLP would also be rather interesting. This as it is a hyperparameter that could not be altered due to hardware limitations.

Chapter 6

Conclusions

This thesis is structured with the goal to investigate whether a shallow MLP is outperformed by a deep CNN in the process of noise reduction, particularly focusing on the impact the size of dataset has on the performance of the neural networks, as well as if any different type of noise is easier to reduce.

From the presented results it is possible to showcase that performance-wise the shallow MLP is outclassed by the deep CNN, even when comparing the MLPs that have been trained on 9000 images to the CNNs that have only been trained on 90.

For the impact of the choice of the size of dataset we can see that it is also significant, regardless of noise or neural network used. This can be seen in how the average Mean Squared Error values for the both of the trained network models decrease together with the increase in the size of the dataset used for training.

Lastly, the impact of the choice of noise is especially notable when it comes to validation data obtained from the CNN-based model. Whereas it is not when dealing with the MLP-based one. This can especially be seen in the Mean Squared Error dealing with the reduction of Poisson noise for the CNN-based model. For the validation data on the CNN trained with 9000 files, the reduction of Gaussian noise performs about ten times worse than the reduction of Poisson noise. While the MLP simply keeps the reduction of every type of noise within a similar range, within each section of size of dataset used to train the network.

Bibliography

[1] Bhawna Goyal et al. “Noise Issues Prevailing in Various Types of Medical Images”. In: *Biomedical and Pharmacology Journal* 11.3 (2018), pp. 1227–1237. DOI: 10.13005/bpj/1484. URL: https://biomedpharmajournal.org/vol11no3/noise-issues-prevailing-in-various-types-of-medical-images/.

[2] Philippe Cattin. *Image Restoration: Introduction to Signal and Image Processing*. MIAC, University of Basel. Apr. 24, 2012.

[3] Hugues Talbot et al. “Efficient Poisson Denoising for Photography”. In: Dec. 2009, pp. 3881–3884. DOI: 10.1109/ICIP.2009.5414042.

[4] Shan Gai et al. “Speckle noise reduction in medical ultrasound image using monogenic wavelet and Laplace mixture distribution”. In: *Digital Signal Processing* 72 (2018), pp. 192–207.

[5] MathWorks. *Noise Removal - MATLAB Simulink - MathWorks Nordic*. URL: https://se.mathworks.com/help/images/noise-removal.html.

[6] Lubna Badri. “Development of neural networks for noise reduction.” In: *Int. Arab J. Inf. Technol.* 7.3 (2010), pp. 289–294.

[7] Cambridge University. *NEURAL NETWORK | meaning in the Cambridge English Dictionary*. Cambridge University Press, 2017. URL: https://dictionary.cambridge.org/dictionary/english/neural-network.

[8] Techopedia. *What is a Multilayer Perceptron (MLP)? - Definition from Techopedia*. Techopedia.com, 2017. URL: https://www.techopedia.com/definition/20879/multilayer-perceptron-mlp.

[9] Hamed Habibi Aghdam and Elnaz Jahani Heravi. *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*. Gewerbestrasse 11, 6330 Cham, Switzerland, 2017.

[10] Alom Zahangir et al. *The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches*. 300 College Park, Dayton, OH 45469 USA, 2018.

[11] Jose Luis Hidalgo et al. *Enhancement and Cleaning of Handwritten Data by Using Neural Networks*. Camí de Vera, s/n, 46022 València, Spain, 2005.

[12] Daniel S Berman et al. "A Survey of Deep Learning Methods for Cyber Security". In: *Information* 10.4 (2019), p. 122.

[13] MathWorks. *Supervised Learning Workflow and Algorithms MATLAB Simulink*. URL: `https://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html`.

[14] Grant Sanderson. *But What is a Neural Network | Deep learning, chapter 1*. 3Blue1Brown. 2015. URL: `https://www.youtube.com/watch?v=aircAruvnKk`.

[15] Simon Haykin. "Neural Networks: A comprehensive foundation 2nd edition". In: *The Oxford Handbook of Innovation*. Ed. by Prentice Hall. Upper Saddle River, New Jersey, USA: Prentice Hall, 1998. Chap. 6, p. 138.

[16] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2018. URL: `https://arxiv.org/pdf/1511.08458.pdf`.

[17] Le Lu et al. *Deep Learning and Convolutional Neural Networks for Medical Image Computing: Precision Medicine, High Performance and Large-Scale Datasets*. Gewerbestrasse 11, 6330 Cham, Switzerland, 2017.

[18] Harsh Pokharna. *The best explanation of Convolutional Neural Networks on the Internet!* 2018. URL: `https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8`.

[19] François Chollet et al. *Keras*. `https://keras.io`. 2015.

[20] Google. *Overview of Hyperparameter Tuning | AI Platform for TensorFlow | Google Cloud*. URL: `https://cloud.google.com/ml-engine/docs/tensorflow/hyperparameter-tuning-overview` (visited on 06/07/2019).

[21] *Noise*. Oxford Dictionaries. URL: `https://en.oxforddictionaries.com/definition/noise`.

[22] A. Maity et al. “A Comparative Study on Approaches to Speckle Noise Reduction in Images”. In: *2015 International Conference on Computational Intelligence and Networks*. Jan. 2015, pp. 148–155. DOI: 10.1109/CINE.2015.36.

[23] Bo Fu et al. *A Convolutional Neural Networks Denoising Approach fot Salt and Pepper Noise*. School of Computer and Information Technology, Liaoning Normal University, China, 2018. URL: https://arxiv.org/pdf/1807.08176.pdf/.

[24] Aojia Zhao. *Image Denoising with Deep Convolutional Neural Networks*. Stanford University, 2016. URL: https://web.stanford.edu/class/cs331b/2016/projects/zhao.pdf.

[25] Josh Warner, Juan Nunez-Iglesias, and Steven Silvester. *scikit-image*. https://github.com/scikit-image/scikit-image/blob/master/skimage/util/noise.py#L8. 2018.

[26] Yuqing Tang. *CNTK 105: Basic autoencoder (AE) with MNIST data*. https://github.com/microsoft/CNTK/blob/master/Tutorials/CNTK_105_Basic_Autoencoder_for_Dimensionality_Reduction.ipynb?fbclid=IwAR1x5zFcbkXcejcthm_YG-pxyDHSFKl79-X60eoHZbKi2WBDZNkBJnblr74. 2017.

[27] John TE Richardson. “Eta squared and partial eta squared as measures of effect size in educational research”. In: *Educational Research Review* 6.2 (2011), pp. 135–147.

Appendix A

Visual Representations

Figure A.1: A clean image extracted from the dataset.

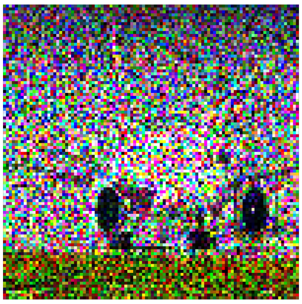

Figure A.2: The clean image corrupted with Speckle noise.

Figure A.3: The Speckle noise corrupted image noise-reduced using the MLP trained on 90 files.

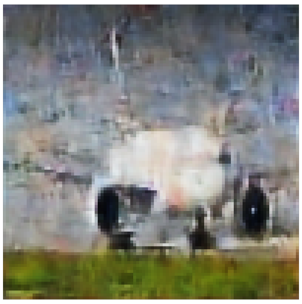

Figure A.4: The Speckle noise corrupted image noise-reduced using the CNN trained on 90 files.

Figure A.5: The Speckle noise corrupted image noise-reduced using the MLP trained on 900 files.

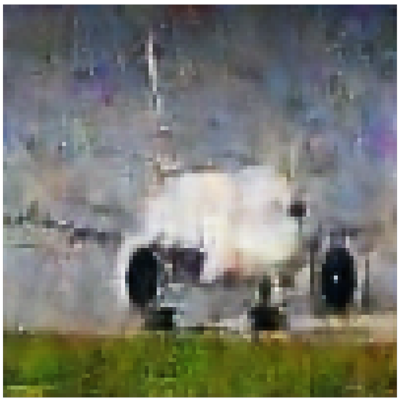

Figure A.6: The Speckle noise corrupted image noise-reduced using the CNN trained on 900 files.

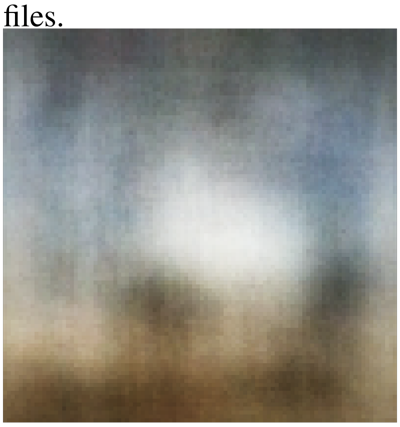

Figure A.7: The Speckle noise corrupted image noise-reduced using the MLP trained on 9000 files.

Figure A.8: The Speckle noise corrupted image noise-reduced using the CNN trained on 9000 files.

Figure A.9: A clean image extracted from the dataset.

Figure A.10: The clean image corrupted with Gaussian noise.

Figure A.11: The corrupted image noise-reduced using the MLP trained on 90 files.

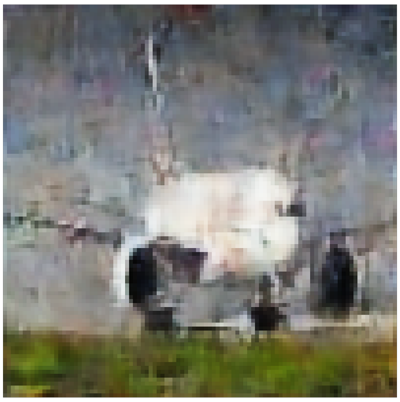

Figure A.12: The Gaussian noise corrupted image noise-reduced using the CNN trained on 90 files.

Figure A.13: The Gaussian noise corrupted image noise-reduced using the MLP trained on 900 files.

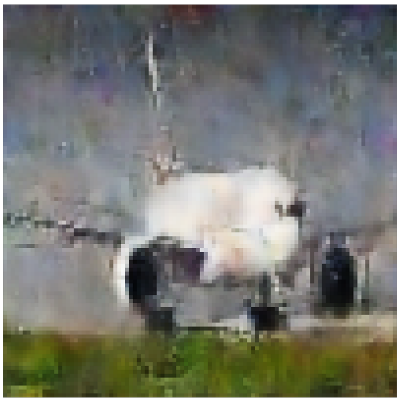

Figure A.14: The Gaussian noise corrupted image noise-reduced using the CNN trained on 900 files.

Figure A.15: The corrupted image noise-reduced using the MLP trained on 9000 files.

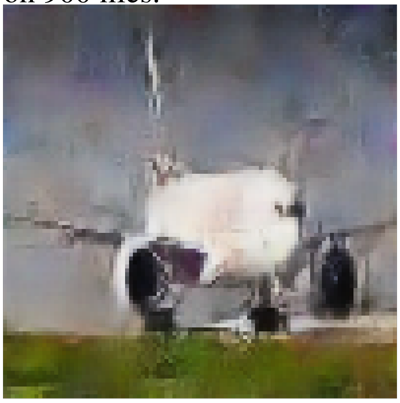

Figure A.16: The Gaussian noise corrupted image noise-reduced using the CNN trained on 9000 files.

Figure A.17: A clean image extracted from the dataset.

Figure A.18: The clean image corrupted with Poisson noise.

Figure A.19: The Poisson noise corrupted image noise-reduced using the MLP trained on 90 files.

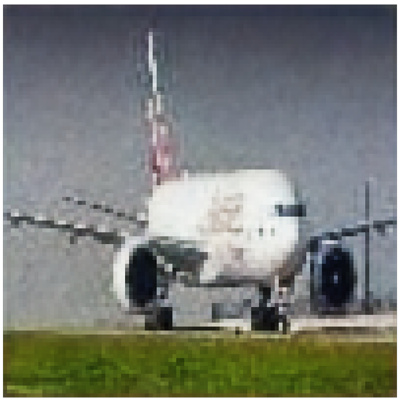

Figure A.20: The Poisson noise corrupted image noise-reduced using the CNN trained on 90 files.

Figure A.21: The Poisson noise corrupted image noise-reduced using the MLP trained on 900 files.

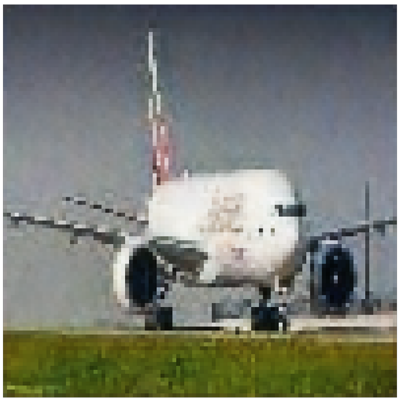

Figure A.22: The Poisson noise corrupted image noise-reduced using the CNN trained on 900 files.

Figure A.23: The Poisson noise corrupted image noise-reduced using the MLP trained on 9000 files.

Figure A.24: The Poisson noise corrupted image noise-reduced using the CNN trained on 9000 files.

TRITA-EECS-EX-2019:347