Tampere University

Joonas Palonen

MACHINE LEARNING FOR COMPUTATIONAL COHERENT NEAR-EYE DISPLAY

End-to-end optimization frameworks with CNNs

Bachelor's Thesis
Faculty of Information Technology and Communication Sciences
Examiner: Jani Mäkinen
November 2021

ABSTRACT

Joonas Palonen: Machine Learning for computational coherent near-eye display
Bachelor's Thesis
Tampere University
Information technology
November 2021

Near eye displays (NEDs) have grown greatly in popularity in a few years and they have been adapted to numerous fields. They still have multiple problems with one of them being vergence-accommodation conflict (VAC). VAC causes visual fatigue and discomfort for the user because the displays are not able to produce accurate enough visual cues for the human visual system (HVS) to interpret.

In this thesis, we study the further development of pre-processing systems for the method proposed by U. Akpinar et.al [1]. It presents an accommodation-invariant computational near-eye display to alleviate the symptoms of VAC. To achieve this, U. Akpinar et.al used U-Net architecture for pre-processing. We implement three alternative end-to-end optimization framework architectures for image pre-processing that are based on Super-Resolution CNN (SRCNN), Very Deep Super-Resolution CNN (VDSR), and Denoising-CNN (DNCNN) respectively. The proposed implementations are compared to the existing U-Net architecture.

The implementations were done with Matlab. In the training phase, the models were given 256x265 RGB image patches from a dataset of 17000 images. At each iteration, the accommodation depth z was randomly selected from 0-3D range. The models were compared by using peak-signal-to-noise-ratio (PSNR) and structural-similarity-index measure (SSIM). Also, the qualitative features of the processed images and the training phase were taken into account. The experiments showed that only the SRCNN based method was able to compete against U-Net. It reached better PSNR values and higher quality images but in a narrower accommodation depth.

Further research on the training process for these architectures is needed. Also, other more complex methods for image-processing have been proposed and those could prove successful for this use case but implementing them is beyond the scope of this thesis.

Keywords: machine learning, neural network, optimization framework, near-eye display, vergence-accommodation conflict

TIIVISTELMÄ

Joonas Palonen: Lähinäyttöjen kuvien esiprosessointi neuroverkoilla
Kandidaatin tutkielma
Tampereen yliopisto
Tietotekniikka
Marraskuu 2021

Lähinäytöt (near-eye display) ovat kasvattaneet suosiotaan valtavasti viime vuosina ja niitä on otettu onnistuneesti käyttöön lukuisilla toimialoilla. Lähinäytöissä esiintyy kuitenkin useita ongelmia, joista yksi on vergenssi-akkommodaatio konflikti (vergence-accommodation conflict). Se aiheuttaa käyttäjälle silmien väsymystä ja epämukavuutta, sillä lähinäyttöjen tuottamat kiintopisteet eivät ole tarpeeksi tarkkoja ihmisissilmän tulkittavaksi.

Tässä tutkimuksessa jatkokehitän esiprosessointisysteemiä, jonka U. Akpinar ym. esittelivät julkaisussaan [1]. Siinä vergenssi-akkommodaatio konfliktin oireita pyritään lieventämään kehittämällä akkommodaatiosta riippumaton lähinäyttö. Tämän saavuttamiseksi systeemin sisääntulodata tulee esiprosessoida sopivasti käyttäen konvolutionaalisia neuroverkkoja. U. Akpinar ym. käyttivät esiprosessointiin U-Net neuroverkkoarkkitehtuuria, jolle pyrin löytämään paremman vaihtoehdon. Toteutan ja vertailen kolme eri arkkitehtuuria, jotka perustuvat super-resoluutio neuroverkkoon (SRCNN), erittäin syvään super-resoluutio neuroverkkoon (VDSR) ja kohinanpoisto neuroverkkoon (DNCNN).

Tutkimuksessa neuroverkot toteutettiin Matlab -ohjelmistolla. Opetusvaiheessa neuroverkoille syötettiin pienissä erissä 17000 kuvan joukosta kooltaan 256x256 värikuvia. Jokaisessa erässä akkommodaatio syvyys, jolle kuva pyrittiin tarkentamaan, arvottiin väliltä 0-3D. Opetustulosten vertailuun käytettiin prosessoitujen kuvien huippusingaali-kohinasuhdetta (PSNR) ja rakenteellisen samankaltaisuuden mittaa (SSIM). Myös kuvien laadulliset ominaisuudet ja neuroverkkojen opetusprosessin ominaisuudet vaikuttivat vertailussa. Tulokset osoittivat ainoastaan SRCNN-verkko kykenevän vastaavaan suorituskykyyn kuin U-Net arkkitehtuuri. SRCNN-verkko saavutti PSNR arvoiltaan ja laadullisesti paremmin prosessoituja kuvia, mutta kapeammalla akkommodaatio syvyydellä.

Jatkokehityksellä kaikkien verkkojen suorituskykyä voitaisiin parantaa entisestään. Kirjallisuudessa on myös esitetty kompleksisempia arkkitehtuureja, jotka voisivat parantaa esiprosessoinnin laatua, mutta näiden toteuttaminen on tämän tutkimuksen laajuuden ulkopuolella.

Avainsanat: koneoppiminen, neuroverkko, optimointisysteemi, lähinäyttö, vergenssi-akkommodaatio konflikti

CONTENTS

1. Introduction 1
2. Background 2
   2.1 Multilayer perceptron 2
   2.2 Convolutional neural network 4
   2.3 Vergence-accommodation conflict 6
   2.4 Architecture of the existing method 6
   2.5 Related work 8
3. Methodology 9
   3.1 UNET 9
   3.2 SRCNN 10
   3.3 VDSR 11
   3.4 DNCNN 12
   3.5 Loss function 13
4. Experiments 14
   4.1 Training 14
   4.2 Results 15
   4.3 Discussion 17
5. Conclusion 18
References 19

1. INTRODUCTION

Near-eye displays (NEDs) have seen a huge jump in popularity in recent years with a growing number of applications in scientific visualization, medical imaging, education, operating remote devices and more. Doctors, engineers and pilots all benefit from having additional information and capabilities right at their visual field. [2][3] However, these displays often cause visual fatigue and discomfort in the user due to distortions in perceived three-dimensional (3D) structure compared to the real scenes [4]. Another cause for these symptoms is the dissociation between vergence and accommodation in the human visual system, caused by the displays not being able to create proper visual cues, known as vergence-accommodation conflict (VAC) [5].

In this thesis, we will further develop the method proposed by U. Akpinar et.al. in [1] which presents accommodation-invariant computational NED. It uses extended depth of field imaging with a diffractive optical element together with a conventional refractive eyepiece and a pre-processing convolutional neural network (CNN) to address the VAC problem. The goal of this study is to implement and compare end-to-end optimization frameworks to do the pre-processing and find the most suitable one for optimizing the system proposed in [1]. The frameworks are implemented using Matlab, MatConvNet library and Deep Learning toolbox. All the neural networks are trained to compensate for the effect of the optics, such that the resulting image is perceived as sharp as possible.

The thesis is structured in following way. Chapter 2 presents background and theory for the technologies behind the study and gives insight to other related work related to addressing the VAC problem. In chapter 3, we discuss the machine learning methods used in this study. The fourth chapter reviews the data used in the CNNs, the experiments conducted and the results from the experiments. Conclusion about the results and their is given in the final chapter.

2. BACKGROUND

Machine learning (ML) is and has been a growing trend in information technology for many years now. The development of ML has been accelerated by the increased availability in computational power and data in different forms and a strong open source culture. [6] The ability for algorithms to learn from data and improve their performance in assigned tasks opens numerous applications in fields such as image processing. In image processing, applications can be used, for example, to achieve image super resolution where a higher resolution image can be reconstructed from observed lower-resolution images. [7]

2.1 Multilayer perceptron

The basic structure of any artificial neural network is multilayer perceptron (MLP) which mimics the brains way of doing decisions. There, numerous single units called neurons are linked together to form a neural network. Each neuron consists of three basic components which are also presented in Figure 2.1.

1. A set of i synapses each having a weight w_i. The value of weight w_i may be positive or negative. A positive weight will favor the effect of input x_i in the summation function, while negative will disfavor it.
2. Summation function which adds the weighted input signals. Often, neural networks also include a bias for better a threshold of the specific neurons output. This can be expressed by equation $y_{sum} = \sum_{i=1}^{n} w_i x_i + b$.
3. Activation function σ which results in an output signal only when the sum of the input signals exceeds a specific threshold. The activation function can be expressed as $y_{out} = \sigma(y_{sum})$.

Most common activation functions are logistic sigmoid function and rectified linear unit (ReLU). Sigmoid is defined as

$$\sigma(y) = \frac{1}{1 - e^{-y}}, \tag{2.1}$$

and it moves the input value y to finite range of (0,1). The benefit of using sigmoid is that it is differentiable and hence continuous which is required by many learning algorithms.

ReLU is defined as

$$f(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{2.2}$$

In many applications ReLU can result in faster learning and it can be used for training deeper networks. [8][9]. Both of these activation functions will be used in the CNNs implemented in this thesis.

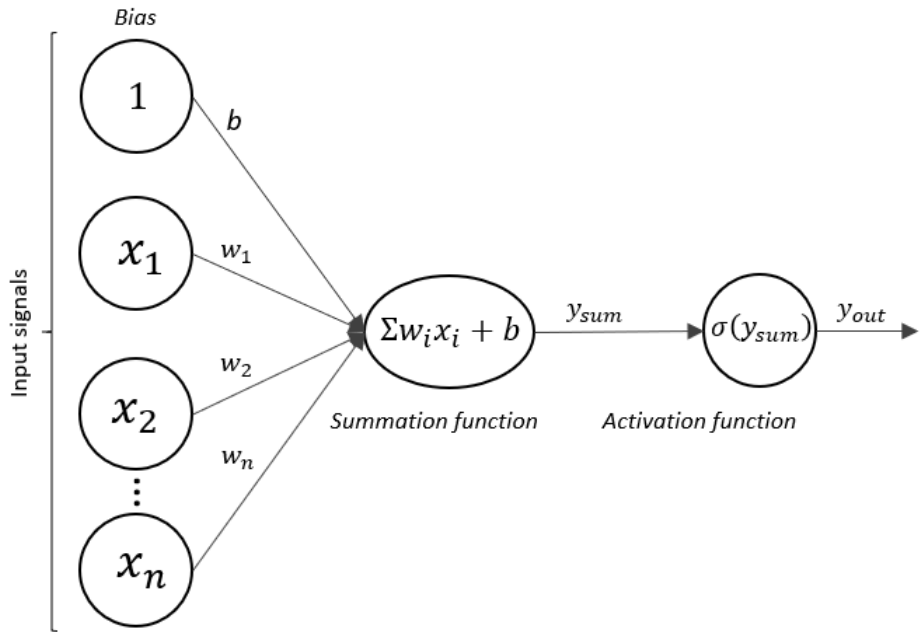

Figure 2.1. *The structure of a neuron with n inputs and bias using ReLU activation function. The activation of this neuron is inserted as an input to subsequent neurons according to MLP connections. Adapted from [8].*

MLP is a feed-forward network, consisting of consecutive layers of neurons linked together. By definition, MLPs consist of an input layer, one or more hidden layers and an output layer. [9] An example, fully connected MLP structure is presented in Figure 2.2.

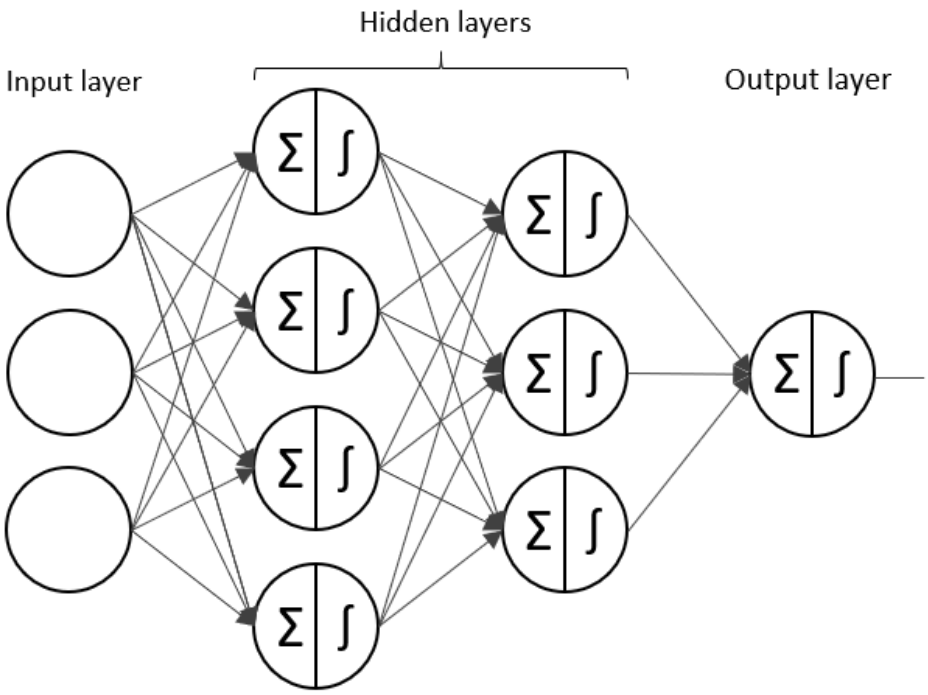

Figure 2.2. *The structure of an example fully connected multilayer perceptron with 3 inputs, 2 hidden layers that have 4 and 3 neurons and one output. Neurons are represented with $\Sigma \int$ labels which represent the summation and activation functions in neurons. Adapted from [8].*

The number of neurons in the output layer depends on the purpose of the model. In a simple classification problem the output may be set to be either 1 or 0. If we want to do multi-class classification, we need the same number of output neurons as we have classes. Each neuron will then correspond to the probability of an input belonging to each class. [8] If the purpose of the model is to do image processing the output layer can have a neuron for every pixel of the input image or output corresponding to a smaller batch of the image [7][10].

Neural networks learn by iteratively tuning the weight and biases of the individual neurons. In supervised learning each input is matched with the correct answer and error between the networks output and the correct answer is calculated. In each iteration the weights and biases of neurons are tuned to minimize the errors of the model. [8] [9]

2.2 Convolutional neural network

Images are complex data to use as an input for fully connected MLP architectures. Assume we had a 16×16 grayscale image as an input and a fully connected MLP architecture with one hidden layer consisting of 1000 neurons. The image can be thought of as a $16 \times 16 = 1024$ dimensional vector. All the neurons and elements are connected together, meaning that the fully connected layer would have $1024 \times 1000 = 1024000$

distinct parameters. [11] It would require a lot of computing resources to process this architecture in a meaningful way. This is why most image processing tasks and ML tasks in general are solved by using convolutional neural networks (CNNs). CNNs have much fewer parameters and connections and thus, are easier to train. Pairing this with modern powerful GPUs, highly-optimized convolution implementation and well-designed CNN architectures means that very complex and computationally challenging problems can be solved. [12]

CNNs apply, for example, a series of convolution, pooling and backpropagation operation to extract features from inputs. They learn to adjust the weighting of the filters in such a way that in the last layer of the network, different classes become linearly separable. The architecture of the CNN plays a vital role on its performance and results. [12]

Let us take an example with random filters to understand the role of the convolution better. Figure 2.3 illustrates a simple system with two consecutive convolution layers where the input is an RGB image. The first layer has six $7\times7\times3$ filters. The input has three channels (RGB) and therefore, the third dimension of the first convolutional layer must be equal to 3. Applying the first set of filters to the image produces a six-channel image (each filter produces a single-channel image). The second convolution layer has only one $5 \times 5 \times 6$ filter. Because the output of the first layer is fed directly to the second convolution layer without activation functions, the third dimension of the second convolution layer must be equal to 6. Since the second layer has only one filter, the resulting image will have one channel. [11]

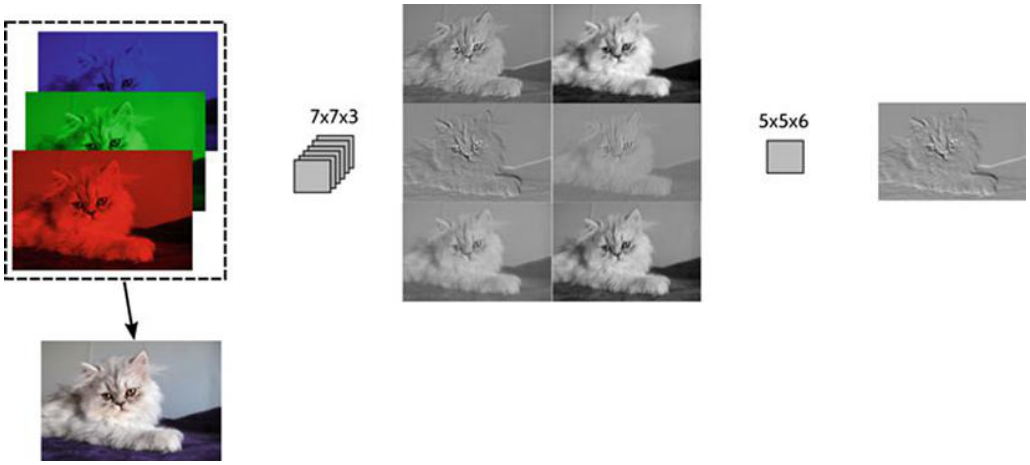

Figure 2.3. *A simple convolution system with three channel input. [11]*

By looking at 2.3 we can see that two of the filter in the first layer worked as low-pass filters (smoothing filters) and the rest worked as high-pass filters (edge-detection filters). The second layer then produces a single channel image by taking a value (m, n) and linearly combining all six channels of the first layer in the 5×5 neighborhood at location (m, n). [11]

2.3 Vergence-accommodation conflict

The human visual system interprets multiple visual cues such as retinal blur and disparity to generate a comfortable 3D viewing experience. Retinal blur is the cue that drives the lens adjustment of the eye to focus on desired depth, known as accommodation, thus minimizing the blur. Retinal disparity is the visual cue that drives rotational movement of the eye in horizontal and vertical directions, called vergence. These two measures also work in parallel to help each other to adjust the eye's focus optimally. [13]

Vergence–accommodation conflict is a well-known problem regarding all stereoscopic displays. In a conventional near-eye display, the optics position the displayed image to a variable depth according to the content of the frame. According to the basic principles of lens imaging in a simple lens system, the image is sharp only at one depth at a time and blurred at all the other depths. [14] Thus, such a system produces focus cues which are inconsistent with a true 3D scene. The inability of traditional NEDs to generate accurate visual cues for the visual system to adjust causes conflicts in the eye's vergence-accommodation feedback loops which, in turn, results in the symptoms of VAC. [13] The basic geometry of VAC is described in Figure 2.4.

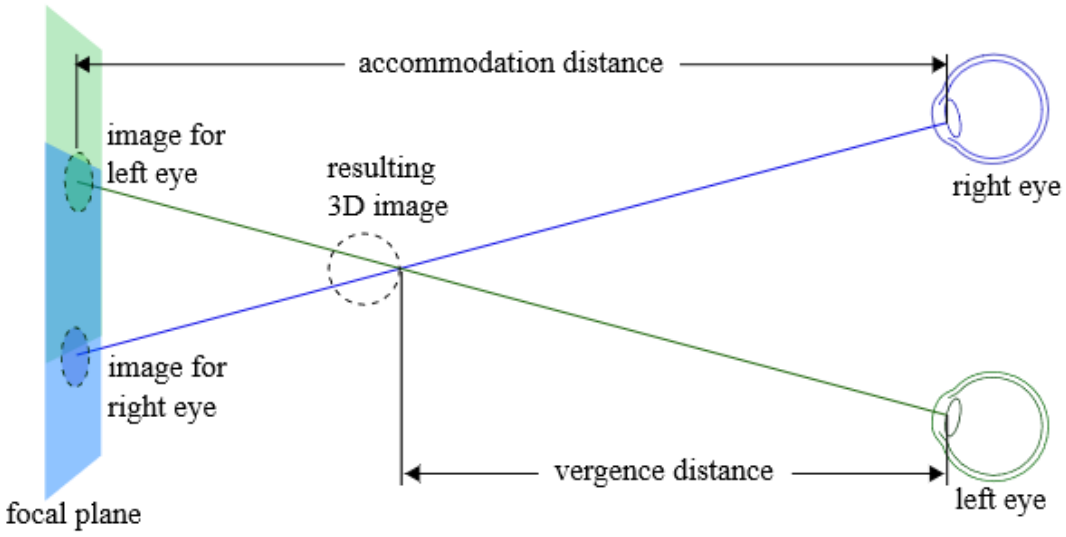

Figure 2.4. *The basic geometry behind the vergence-accommodation conflict [13].*

By processing the displayed images in a way that removes the focus cues and sharpens them at any given depth, the eye does not have to change its accommodation. This has the potential to mitigate the symptoms of VAC. [15]

2.4 Architecture of the existing method

The architecture of the method proposed by U. Akpinar et.al. in [1] is illustrated in Figure 2.5. The system consist of two main parts: the pre-processing and the display. First,

an all-in-focus image I is given as an input to the pre-processing network. The output of this is I^d and it is given as an input to the display model. The output of the whole network is I^p which is the simulated perceived image seen by the eye. As a last step, the loss between the initial input I and I^p is calculated which is used to co-optimize the display design and the pre-processing network. The point spread function (PSF) is used to calculate the response of the imaging system for the desired accommodation distance z to display images.

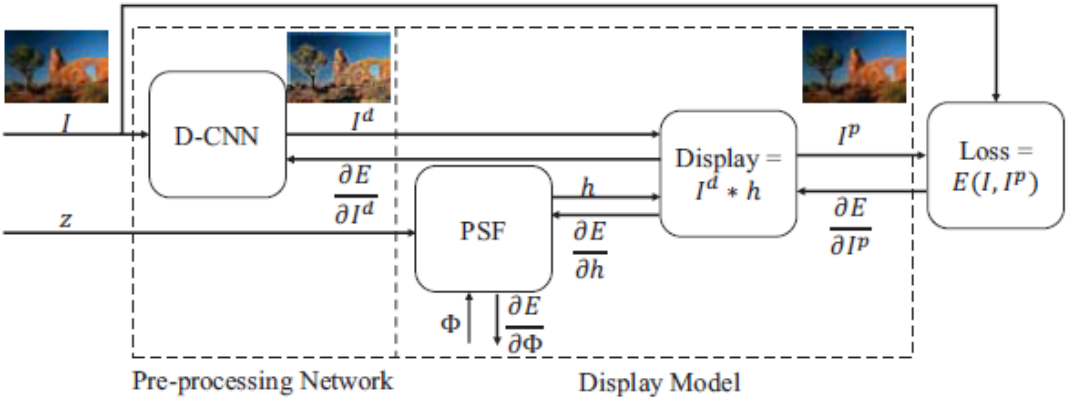

Figure 2.5. *Representation of the method proposed in [1].*

The computational near-eye display combines a refractive lens and a diffractive optical element (DOE). In forward model the perceived images are simulated on the reference plane at which an assumed aberration-free eye is supposed to be accommodated. This is shown in Figure 2.6. The system PSF needed for the model can be derived using the generalized pupil function.

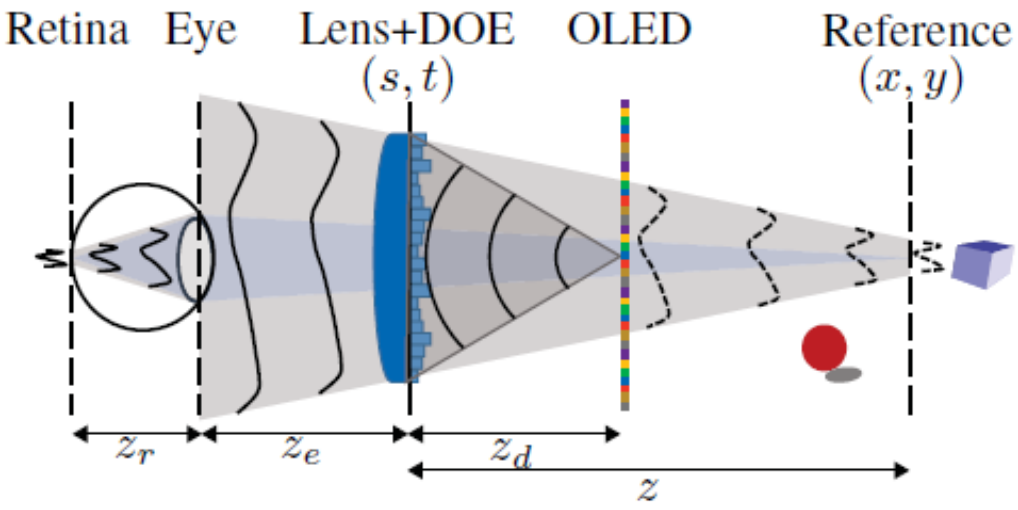

Figure 2.6. *Architecture the NED setup proposed in [1].*

The main difference in this architecture compared to others proposed in literature is the use of static optics, which is easy to integrate into conventional headsets. Also, it is compatible with any 2D display unit.

2.5 Related work

Numerous other methods have been proposed to alleviate and fix the problems of VAC, mainly by either making the display accommodation-invariant (AI) [15] or by providing the depth cues accurately (i.e. using accommodation enabling displays). AI displays have been achieved, for example, by using light field displays [16][17] or with varifocal [18] or multifocal plane [19] displays.

Some network architectures that achieve image super-resolution and deblurring include, for example, generative adversial network (GAN) [20] and residual dense network (RDN) [21]. In a GAN two networks are working in a feed-forward configuration. The first one generates a high-resolution image I^{SR} and the other tries to compare and distinguish the I^{SR} from real high-resolution images. Thus, the first network can learn to create solutions that are highly similar to real images. [20] RDN generates high-resolution images I^{SR} in more straightforward manner by regular learning-based model. This can lead to higher PSNR values compared to GAN, but the output images can contain unpleasing artefacts or other visual imperfections. [21].

3. METHODOLOGY

We implement three deblurring convolutional neural networks (D-CNNs) and integrate them into the end-to-end optimization and display system shown in Figure 2.5. These methods are then compared to the existing D-CNN that uses U-Net [22] architecture. Comparison is done by giving the trained networks testing images and then calculating their peak-signal-to-noise-ratios (PSNR) and structural similarity index measures (SSIM). Also, the qualitative features of the test outputs are taken into account.

The three D-CNNs implemented are based on Super-Resolution CNN (SRCNN) proposed by C. Dong et.al in [7], Very Deep Super-Resolution CNN (VDSR) proposed by J. Kim et.al in [23] and Denoising-CNN (DNCNN) proposed by K. Zhang et.al in [10]. All of these methods are discussed in more detail in the following section.

3.1 UNET

The existing implementation uses U-Net architecture [22], which has been shown to be a performant way to tackle various image restoration problems. U-Net is a multi-level network that consists of two main parts; a contraction (encoding) path and an extension (decoding) path which are connected via skip connections at every level. The architecture as a whole is illustrated in Figure 3.1. [22]

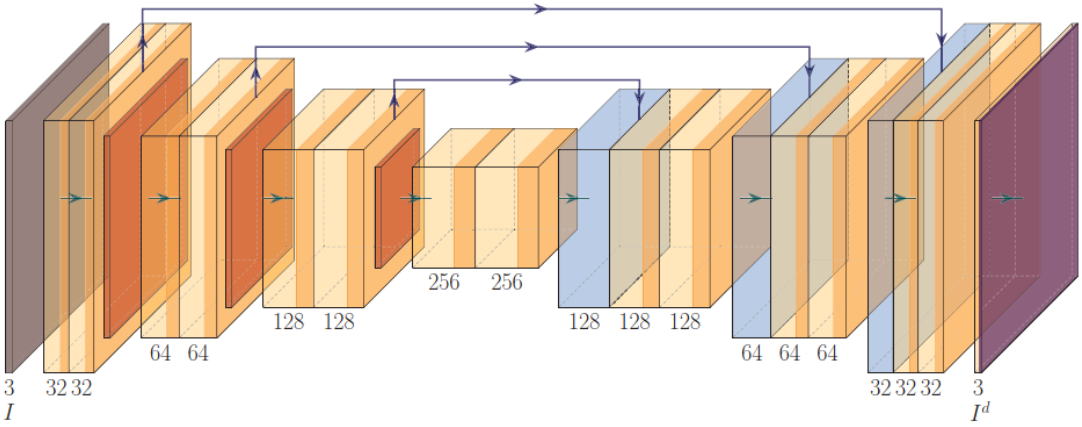

Figure 3.1. *Architeture of U-Net CNN where convolutions are light yellow block, ReLUs are graded yellow, max poolings are red, transposed convolutions are blue and clamping is purple. The channel sizes after each convolution are given below the blocks. Adapted from [22].*

The encoding part consists of 3x3 convolutions, shown as light yellow in Figure 3.1, with ReLU activation functions (graded yellow blocks). The feature maps' channel size after each convolution is given under each block. The channel size doubles after every downsampling. Downsampling is done by 2x2 max pooling layers with stride 2, shown as red blocks in Figure 3.1. [22] [1]

In the decoding part, feature maps are upsampled via 2x2 transposed convolutions (blue blocks) with an upsampling parameter of 2. The output feature maps from the encoding part are concatenated to corresponding feature maps after each upsampling via skip connections, illustrated as arrows in Figure 3.1. Lastly, a 1x1 convolution layer is used to map the network output to the original image size of 3 channels. [22] The output is clamped to be between 0 and 1 in order to account for the dynamic range of the physical display (purple block). The clamped output I^d then goes to the display model. [1]

3.2 SRCNN

The first and the simplest of the three implemented D-CNNs uses the architecture of Super Resolution CNN (SRCNN) illustrated in Figure 3.2 [7]. It consists of the following three parts:

1. patch extraction and representation,
2. non-linear mapping and
3. reconstruction.

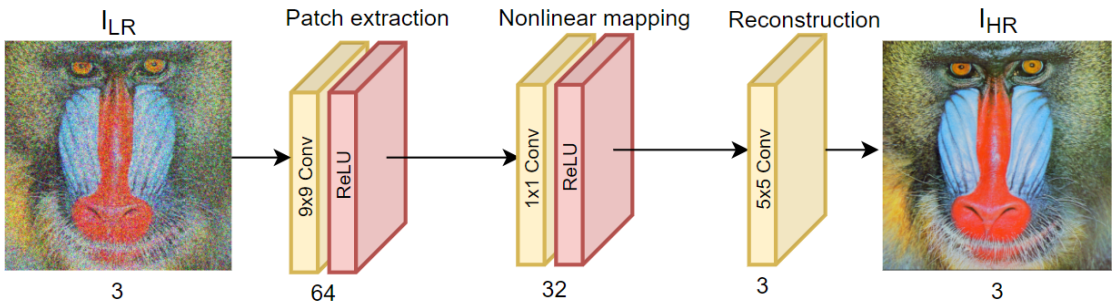

Figure 3.2. *Architeture of SRCNN. Number of feature channels is given below each step. Adapted from [7]*

The first step extracts (overlapping) patches from the input image I_{LR} and represents those patches as high-dimensional vectors. These vectors comprise a set of feature maps. The number of feature maps determines the dimensionality of each vector. Formally, the first step applies a 9x9 convolution with ReLU activation. Then, a 1x1 convolution with ReLU is applied to map each high-dimensional vector onto another high-dimensional vector to conceptually represent a high-resolution patch. It would be possible to add more similar convolutional layers to increase non-linearity with the cost of complexity and thus, more training time. Finally, the high-resolution patch-wise representations are aggregated to generate the final high-resolution image I_{HR}. This is done by applying a 5x5 convolution to the high-dimensional vectors from the second step. Throughout the three steps, the number of feature channels is reduced, in order to have the output in the original image size of 3 channels. [7]

3.3 VDSR

Similarily to SRCNN, the Very Deep Super Resolution CNN (VDSR) [23] consists of the same three parts. However, VDSR uses n number of 3x3 convolutional non-linearity layers, rather than 9-1-5 convolutions. The output of the n 3x3 convolutions with ReLU activations is a residual image which is then summed to the input I_{LR} to get the final output image I_{HR}. Again, the I_{HR} is then given to the display model. [23]

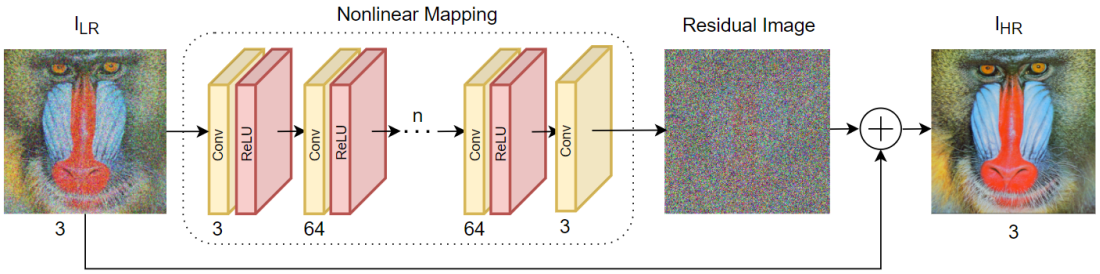

Figure 3.3. *Architeture of VDSR with n number of hidden layers. Number of feature channels after every convolution is given below each block. Adapted from [23]*

A high-resolution image consists of low frequency (corresponding to low-resolution image) and high frequency information (residual image). Because VDSR models residual images rather than high-resolution images directly, like SRCNN does, the training can be focused on the details of the image, rather than carrying the input through the model. This means that VDSR can have faster convergence with even better accuracy. [23]

3.4 DNCNN

The third and final denoising CNN implemented in this study follows the architecture proposed by K. Zhang et. al. in [10]. While the architecture of the model is very similar to the previous two, one key difference could make this model perform better; DNCNN utilizes batch normalization in the hidden layers between 3x3 convolutions and ReLUs. These batch normalization blocks are illustrated light blue in Figure 3.4. From the figure we can see that other than the batch normalization, the DNCNN architecture is indentical to VDSR.

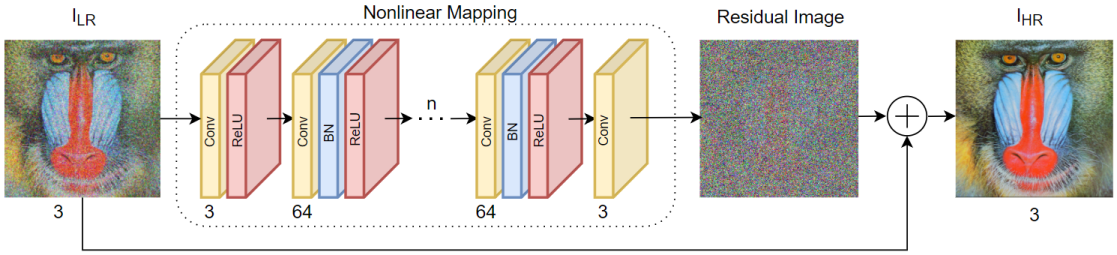

Figure 3.4. *Architeture of DNCNN with n number of hidden layers. Number of feature channels is given below each step. Adapted from [10]*

In their paper, K. Zhang et. al. argue that by using the batch normalization together with residual learning, the training process is faster. This is due to residual learning and batch normalization benefiting each other in the training phase to, not only speed up training, but also boost the denoising process. [10]

3.5 Loss function

In neural networks, the loss layer is a vital part of the training process. Comparing the outputs of the network and the ground truths during training and then shifting the weight of the training to more desirable direction is essential. Thus, a loss metric that favours relevat properties is necessary. [24]

The loss between the output I^p, given by the display model (Figure 2.5), and the all-in-focus image I is calculated using both, a regularized L1-loss, and Structural Similarity (SSIM) index. For a batch of N ground-truth images I_N and corresponding outputs I_N^p the reconstruction L1 loss is calculated by

$$E_{L1}(I, I^p) = \frac{1}{N}\sum_{n=1}^{N}(||I_n - I_n^p|| + \alpha R(I_n, I_n^p)), \tag{3.1}$$

where α is the regularization weight and $R(I_n, I_n^p)$ is the regularization term. A dark channel prior is utilized as the regularization term, as it has been shown to be a powerful prior for image restoration problems. [1][24]

For the same batch of N images the SSIM index is calculated on various windows of the images. SSIM between two windows x and y of same size is

$$E_{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.2}$$

The use of L1-loss is desirable because it does not over-penalize large errors. While on the other hand, SSIM aims to produce visually pleasing images.[24]

4. EXPERIMENTS

In this section, we go over the training process, some parameters necessary for training and the training setup. After that, we compare the results between models and discuss their quality and usability.

4.1 Training

In the training process, the models were given 256x265 RGB image patches from a dataset of 17000 images. At each iteration, the accommodation depth z was randomly selected from {0D, 0.5D, 1D, 1.5D, 2D, 2.5D, 3D} range. Adam optimization function, with initial learning rate of 1e-3 and weight decay of 1e-4, was used. Each model was trained for 20 epochs with a system equipped with a high-end GPU, which resulted in average training times of two weeks per model.

Figure 4.1 illustrates the L1 and SSIM loss functions (see 3.5) of each model during training (blue) and validation (orange) processes. Because the results vary a lot between models and epochs, the "best" epoch was hand-chosen for every model to ensure best results. These epochs were as follows: U-Net 20, SRCNN 17, VDSR 19 and DNCNN 19. From these figures we can see that all the models converge similarily and the same training setup works quite well for all of them. But clearly, the complexity also adds uncertainty. If the images in a given epoch are not favorable to the model, the more complex models experience larger jumps in loss values.

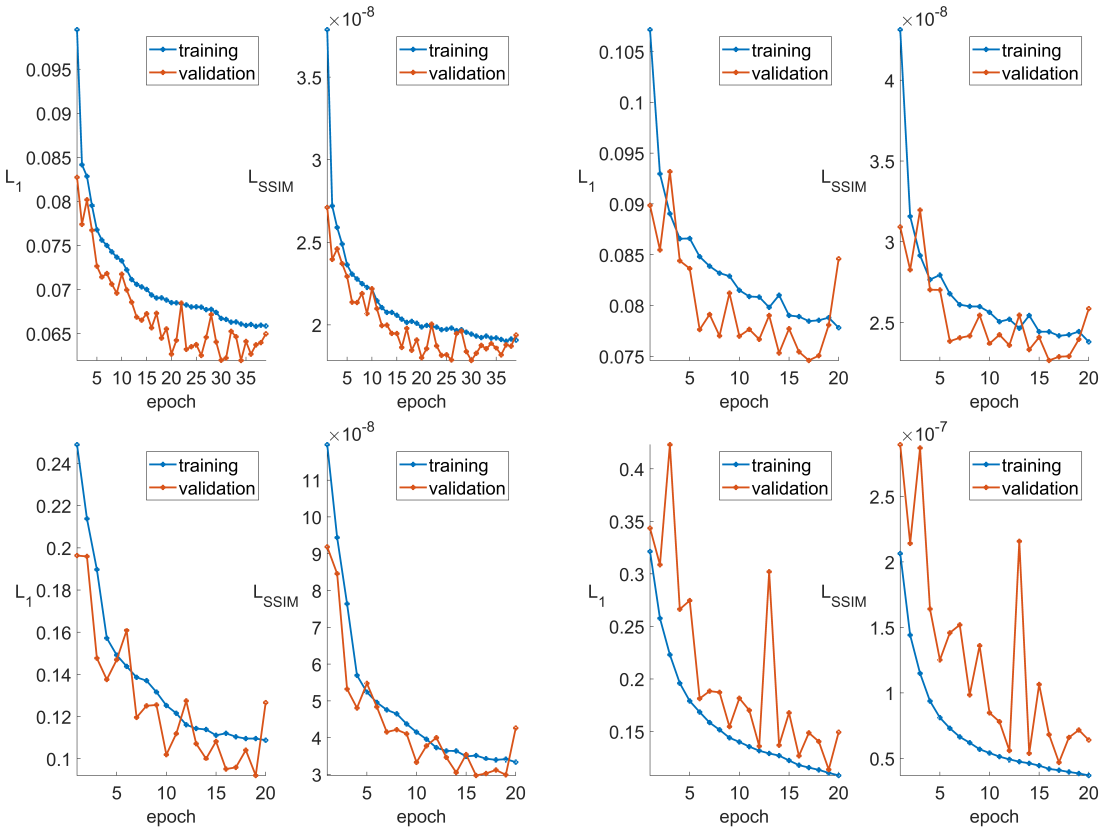

Figure 4.1. *Graphs of loss functions during training and validation processes. From top-left: UNET, SRCNN, VDSR, DNCNN*

4.2 Results

To validate the training results, we compare each model using Peak-Signal-to-Noise-Ratio (PSNR) and also take a look into resulting images and compare their perceived quality. Images used as reference are shown in Figure 4.2. The PSNR values of those images after going through the model pipelines, are shown in Figure 4.3.

If we analyze the PSNR plots further, we see that VDSR and DNCNN are clearly behind U-Net and SRCNN in all accommodation depths. This might be due to the fact that they were initially designed for image denoising and super-resolution tasks, and the high number of non-linearity in hidden layers hurts their performance in this task. With better fine-tuning and tweaking they might be able to compete with the previous U-Net architecture but due to the very long training process this is not viable in this study.

SRCNN on the other hand is almost on-par with U-Net. It out-performs U-Net in a narrower depth-of-field. And comparing the complexity of these two models, SRCNN is much simpler and faster to train that U-Net. So with more tuning and double the training time, SRCNN might out-perform U-Net in the desired depth range.

Figure 4.2. *Reference test images. From top-left: test images a, b, c, d*

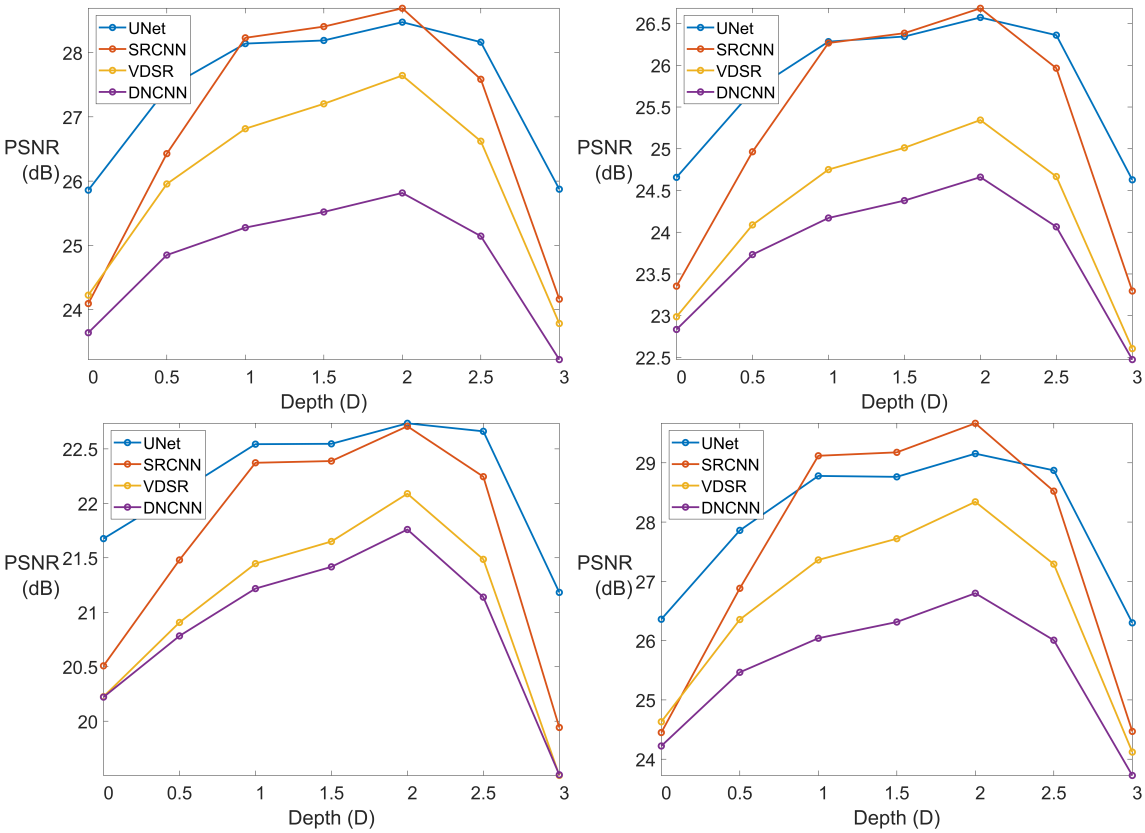

Figure 4.3. *Comparison of PSNR values between networks. Each figure corresponds to one reference image. From top-left: test images a, b, c, d*

Focusing more on the perceived quality of the output images shown in Figure 4.4, we can see that here also VDSR and DNCNN are clearly not competing with U-NET and SRCNN. The output images suffer from heavy color oversaturation, where especially the reds and yellows suffer the most. This effect is not visible on the other two models, where colors look similar to the reference image in 4.2.

Figure 4.4. *Comparison of the outputs given by different networks at varying depths for test image a. The PSNR values are given under each image in dB*

4.3 Discussion

The results given here are definitely not the best that these models can achieve. They lack the latest optimization tricks that are not implemented in Matlab. Also, additional training could still give better results, but is not viable in this study due to time constraints. Multiple publications address more complex models to solve this problem. For example, Generative Adversial Network (GAN) based model proposed by C. Ledig et. al. in [20] or EnhanceNet proposed by M. Sajjadi, B. Schölkopf and M. Hirsch in [25] could give far better results. However, as this study is limited to using Matlab for model coding, implementing these models is outside the scope of this thesis.

5. CONCLUSION

Most of the near-eye displays suffer from vergence-accommodation conflict that leads to user discomfort. By mitigating the symptoms of VAC, NEDs could become available for broader set of professional and entertainment users.

This thesis attempted to implement the best end-to-end optimization frameworks for image pre-processing for accommodation-invariant computational NED. Three convolutional neural networks implemented were based on Super-Resolution CNN (SRCNN), Very Deep Super-Resolution CNN (VDSR) and Denoising-CNN (DNCNN). The proposed implementations were compared to previous related work where the pre-processing was done by using U-Net architecture. Different architectures were compared by training characteristics, using PSNR and perceived quality.

In the experiments all models were trained to pre-process images to a random accommodation depth z from 0D-3D range at each iteration. The results show, that only SRCNN model was able to achieve the same level results as previously proposed U-Net. SRCNN performed measurably better in 2D-1D range. This means that U-Net has the advantage of being performant in a wider depth range and it might altogether be a better model for this task with further tuning.

Experiments with state-of-the-art complex architectures such as Generative Adversial Networks (GANs) or residual dense networks (RDNs) should be conducted in order to achieve more significant results. Also, doing the development in another programming language might be advantageous to have access to state-of-the-art machine learning libraries and optimization tools.

REFERENCES

[1] Akpinar, U., Sahin, E. and Gotchev, A. Phase-Coded Computational Imaging For Accommodation-Invariant Near-Eye Displays. *2020 IEEE International Conference on Image Processing (ICIP)*. 2020, pp. 3159–3163. DOI: 10.1109/ICIP40778.2020.9191236.

[2] Hua, H., Cheng, D., Wang, Y. and Liu, S. Near-eye displays: State-of-the-art and emerging technologies. *Proceedings of SPIE - The International Society for Optical Engineering* 7690 (Apr. 2010). DOI: 10.1117/12.852504.

[3] Vorraber, W., Voessner, S., Stark, G., Neubacher, D., DeMello, S. and Bair, A. Medical applications of near-eye display devices: An exploratory study. eng. *International journal of surgery (London, England)* 12.12 (2014), pp. 1266–1272. ISSN: 1743-9191. DOI: https://doi.org/10.1016/j.ijsu.2014.09.014.

[4] Zhang, L., Zhang, Y.-Q., Zhang, J.-S., Xu, L. and Jonas, J. B. Visual fatigue and discomfort after stereoscopic display viewing. eng. *Acta ophthalmologica (Oxford, England)* 91.2 (2013), e149–e153. ISSN: 1755-375X. DOI: 10.1111/aos.12006.

[5] Hoffman, D. M., Girshick, A. R., Akeley, K. and Banks, M. S. Vergence-accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of vision (Charlottesville, Va.)* 8.3 (2008), pp. 33.1–3330. DOI: 10.1167/8.3.33.

[6] Sosnovshchenko, A. *Machine learning with swift: artificial intelligence for iOS*. eng. Birmingham: Packt Publishing, 2018. ISBN: 1787121518.

[7] Dong, C., Loy, C. C., He, K. and Tang, X. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), pp. 295–307. URL: https://arxiv.org/abs/1501.00092.

[8] Chandramouli, S., Das, A. K. and Dutt, S. *Machine Learning*. eng. Pearson Education India, 2018. ISBN: 9389588138.

[9] Bishop, C. M. *Pattern recognition and machine learning*. eng. Information science and statistics. New York: Springer, 2006. ISBN: 0-387-31073-8.

[10] Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155. DOI: 10.1109/TIP.2017.2662206.

[11] Habibi Aghdam, H. *Guide to Convolutional Neural Networks A Practical Application to Traffic-Sign Detection and Classification*. eng. Cham, 2017.

[12] Krizhevsky, A., Sutskever, I. and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386.

[13] Kramida, G. Resolving the Vergence-Accommodation Conflict in Head-Mounted Displays. eng. *IEEE transactions on visualization and computer graphics* 22.7 (2016), pp. 1912–1931. ISSN: 1077-2626.

[14] Hainich, R. *Displays, 2nd Edition*. eng. 2016. URL: https://learning.oreilly.com/library/view/displays-2nd-edition/9781315350363/.

[15] Konrad, R., Padmanaban, N., Molner, K., Cooper, E. and Wetzstein, G. Accommodation-invariant computational near-eye displays. eng. *ACM transactions on graphics* 36.4 (2017), pp. 1–12. ISSN: 0730-0301. DOI: 10.1145/3072959.3073594.

[16] Ueno, T. and Takaki, Y. Super multi-view near-eye display to solve vergence-accommodation conflict. *Opt. Express* 26.23 (Nov. 2018), pp. 30703–30715. DOI: 10.1364/OE.26.030703.

[17] Hua, H. and Javidi, B. A 3D integral imaging optical see-through head-mounted display. *Opt. Express* 22.11 (June 2014), pp. 13484–13491. DOI: 10.1364/OE.22.013484.

[18] Padmanaban, N., Konrad, R., Stramer, T., Cooper, E. A. and Wetzstein, G. Optimizing virtual reality for all users through gaze-contingent and adaptive focus displays. eng. *Proceedings of the National Academy of Sciences - PNAS* 114.9 (2017), pp. 2183–2188. ISSN: 0027-8424.

[19] Akeley, K., Watt, S. J., Girshick, A. R. and Banks, M. S. A Stereo Display Prototype with Multiple Focal Distances. *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 804–813. DOI: 10.1145/1015706.1015804.

[20] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. and Shi, W. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 105–114. DOI: 10.1109/CVPR.2017.19.

[21] Zhang, Y., Tian, Y., Kong, Y., Zhong, B. and Fu, Y. Residual Dense Network for Image Restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. DOI: 10.1109/TPAMI.2020.2968521.

[22] Ronneberger, O., Fischer, P. and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells and A. F. Frangi. Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[23] Kim, J., Lee, J. K. and Lee, K. M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1646–1654. DOI: 10.1109/CVPR.2016.182.

[24] Zhao, H., Gallo, O., Frosio, I. and Kautz, J. Loss Functions for Image Restoration With Neural Networks. *IEEE Transactions on Computational Imaging* 3.1 (2017), pp. 47–57. DOI: 10.1109/TCI.2016.2644865.

[25] Sajjadi, M. S. M., Schölkopf, B. and Hirsch, M. EnhanceNet: Single Image Super-Resolution through Automated Texture Synthesis. *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE. 2017, pp. 4501–4510. URL: https://arxiv.org/abs/1612.07919/.