BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

IMAGE RESTORATION BASED ON CONVOLUTIONAL NEURAL NETWORKS

RESTAURACE OBRAZU KONVOLUČNÍMI NEURONOVÝMI SÍTĚMI

PH.D. THESIS
DISERTAČNÍ PRÁCE

AUTHOR
AUTOR PRÁCE
ING. PAVEL SVOBODA

SUPERVISOR
VEDOUCÍ PRÁCE
PROF. DR. ING. PAVEL ZEMČÍK

BRNO 2016

ABSTRACT

A merit of this thesis is to introduce a unified image restoration approach based on a convolutional neural network which is to some degree degradation type independent. Convolutional neural network models were trained for two different tasks, a motion deblurring of license plate images and a removal of artifacts related to lossy image compression. The capabilities of such models are studied from two main perspectives. Firstly, how well the model can restore an image compared to the state-of-the-art methods. Secondly, what is the model's ability to handle several ranges of the same degradation type.

An idea of the unified end-to-end approach is based on a recent development of neural networks and related deep learning in a field of computer vision. The existing hand-engineered methods of image restoration are often highly specialized for a given degradation type and in fact, define state of the art in several image restoration tasks. The end-to-end approach allows to directly train the required model on specifically corrupted images, and, further, to restore various levels of corruption with a single model.

For motion deblurring, the end-to-end mapping model derived from models used in computer vision is deployed. Compression artifacts are restored with similar end-to-end based model further enhanced using specialized objective functions together with a network skip architecture.

A direct comparison of the convolutional network based models and engineered methods shows that the data-driven approach provides beyond state-of-the-art results with a high ability to generalize over different levels of degradations. Based on the achieved results, this work presents the convolutional neural network based methods suggesting a possibility having the unified approach used for wide range of image restoration tasks.

KEYWORDS

Convolutional neural networks; deep learning; image restoration; motion deblurring; JPEG artifacts

ABSTRAKT

Tématem práce je použití konvolučních neuronových sítí pro obecnou restauraci obrazu. Ta se typicky provádí za pomoci specializovaných metod pro konkrétní typ poškození. Model konvoluční sítě zde představuje jednotný přístup, který je aplikován na dva různé typy degradace obrazu, pohybem rozmazané snímky registračních značek a artefakty vznikající vysokou kompresí. Na modely konvolučních sítí je nahlíženo ze dvou úhlů. A to jak dobře si konvoluční sítě vedou v porovnání se současnými metodami pro restauraci konkrétního typu poškození a jak velký rozsah poškození je právě jeden model ještě schopen zpracovat.

Klasické metody jsou charakteristické svým úzkým zaměřením na konkrétní typ poškození. Díky své specializaci tyto metody dosahují velmi dobrých výsledků a reprezentují tak dosažené poznání v oboru. Naproti tomu je představena myšlenka jednotného přístupu, tedy mapování poškozeného obrazu přímo na restaurovaný obraz. Ta je primárně ovlivněna současným vývojem konvolučních neuronových sítí a jejich hlubokého učení v počítačovém vidění. Právě učením konvoluční sítě lze jednoduše získat model zaměřený na konkrétní typ poškození. Ten je současně nezřídka schopen pokrýt širokou škálu úrovní konkrétního poškození.

V práci je představena metoda přímého mapování z rozmazaného na ostrý obraz pro restauraci pohybem rozmazaných snímků. Ta je odvozena od modelů využívaných v počítačovém vidění pro sémantickou segmentaci obrazu. V případě odstranění kompresních artefaktů je tento přístup rozšířen o specifické učení modelu a různé modifikace samotné architektury sítě.

Modely konvolučních sítí v porovnání s tradičními metodami dosahují kvalitativně lepších výsledků. Zároveň se zde představené modely jednoduše vypořádají s širokým rozsahem konkrétního poškození. Ukazuje se tak, že právě modely konvolučních sítí by mohly reprezentovat jednotný přístup pro restauraci různých typů poškozeni.

KLÍČOVÁ SLOVA

Konvoluční neuronové sítě; hluboké učení; restaurace obrazu; dekonvoluce; JPEG artefakty

BIBLIOGRAPHIC CITATION

Pavel Svoboda: *Image Restoration based on Convolutional Neural Networks*, doctoral thesis Brno, Brno University of Technology, Faculty of Information Technology, 2016.

DECLARATION

I declare that this dissertation thesis is my original work and that I have written it under the guidance of prof. Dr. Ing. Pavel Zemčík. All sources and literature that I have used during my work on the thesis are correctly cited with complete reference to the respective sources.

Brno, 2016

Pavel Svoboda,
August 25, 2016

ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Pavel Zemčík for the great opportunity to study and work under his guidance. Furthermore, I would like to thank Michal Hradiš who introduced me the field of convolutional networks and who steered me directly to image restoration.

I have to definitely thank all my lab mates from L203. Namely, David who initiated the idea of dealing with image artifacts and also who helped me discover the Unix-like systems, Lukáš for all his valuable advice and the opportunity to work with him and learn from him, Šošo for his motivational approach throughout writing this thesis, Ondra who I spent several evenings with talking about programming and drinking the strongest slivovitz in my life.

My thanks go to all my friends and colleagues who I was working with on several projects related to this thesis including *ALMARVI* and *SLAM++ Frontend*. I would like to express my gratitude to my family who supported me and who provided me with the warmth of home and their care. Finally, I would like to thank Kačka for her patience, willingness, and lovely smile.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADALINE	Adaptive Linear Element	17
AE	Auto-Encoder	26
CNN	Convolutional Neural Network	1
DAE	Denoising Auto-Encoder	35
DCT	Discrete Cosine Transform	13
ELU	Exponential Linear Unit	33
EXIF	Exchangeable Image File Format	12
FCN	Fully Convolutional Network	24
FT	Fourier Transform	7
GPU	Graphics Processing Unit	21
IDCT	Inverse Discrete Cosine Transform	14
ILSVRC	ImageNet Large Scale Visual Recognition Challenge	18
JFIF	JPEG File Interchange Format	12
JPEG	Joint Photographic Experts Group	12
JQT	JPEG Quality Transcoder	48
LM-BFGS	Limited Memory Broyden–Fletcher–Goldfarb–Shannon	31
NN	Artificial Neural Network	1
MADALINE	Many Adaline	17
MAP	Maximum A Posteriori	9
MCP	McCulloch Pitts Neuron	17
MSE	Mean Square Error	7
OCR	Optical Character Recognition	36
Pascal VOC	Pascal Visual Object Classes	25
PReLU	Parametrized Rectified Linear Unit	33
PSF	Point Spread Function	1
PSNR	Peak Signal to Noise Ratio	36
ReLU	Rectified Linear Unit	21
SA-DCT	Shape Adaptive Discrete Cosine Transform	15
SR-CNN	Super-Resolution CNN	35
SDAE	Stacked Denoising Auto-Encoder	45
SGD	Stochastic Gradient Descent	30
SPP	Simple Postprocessing	15
SSIM	Structural Similarity	61
TPE	Temporal Propositional Expressions	17

1

INTRODUCTION

In 1943 Warren S. McCulloch, a neurophysiologist, together with Walter Pitts, a mathematician, published their work *A logical calculus of the idea immanent in nervous activity* which is in the field of artificial neural networks considered to be one of the first attempts to define and design a model of a very simplified network reflecting the real neural architecture. During more than 70 years, the neural network based models were developed into more complex and in several aspects more by nature inspired architectures. Nowadays, the most visible Artificial Neural Network (NN) impact is in the tasks of speech recognition and computer vision where the ongoing research develops fast and almost continuously reveals new knowledge. Namely, in the computer vision, the Artificial Neural Networks are found in the state-of-the-art image classification, scene labeling and captioning, object detection and localization. The capabilities which the Artificial Neural Networks demonstrate in the speech recognition and computer vision captured the attention of other computer-based research communities. Naturally, the most visible influence can be found in the speech and vision-related tasks such as speech and image processing which comprise the broad field of signal processing.

The primary objective of this thesis involves the NN deployment in image restoration which, by its nature, is part of the more general image processing field. Such an idea does not evolve for the first time. However, the presented image restoration is framed by a unified approach based on a data-driven Convolutional Neural Network (CNN) model. This idea is introduced in more detail on two examples of common image restoration tasks such as an image deblurring, i. e. restoring the blurred image into its sharp representation, and an image artifacts removal.

A well-established approach exists to restore the degradation caused by blur which consists of several steps. First, the model of the process blurring the image has to be defined. Based on this model, the so-called Point Spread Function (PSF) is derived. Second, having the PSF, the blurred image can be reversed into its sharp representation using deconvolution. The approach differs in the case of image artifacts removal. The degradation process has to be modeled as well; however, the method to remove or at least to suppress the artifacts is diametrically different from the one for deblurring.

Based on the success of various NNs in tasks of computer vision, similar models are deployed in the image restoration comprising a unified data-driven approach. Compared to the traditional engineered methods designed for a particular type of corruption restoration, the NN allows using the same NN based model just trained on different data. A single model used for arbitrary corruption restoration would be the desired outcome, which, considering the capabilities the neural networks have, should not be so much unrealistic. However, this is not the case. This thesis

focuses on the utilization of a NN as the primary approach in image restoration, which may differ in training or particular architecture providing significant and state-of-the-art comparable results. There exist various published methods in image processing which make use of NN. Nevertheless, it is very occasional that the approach solely considers an end-to-end mapping provided by the NN. Usually, NN comprises a part of the more complex processing pipeline, which points out back to the engineered and per task specialized approaches.

A NN deployment in image processing shares with other fields a clearly visible pattern which reflects the interesting history behind the NN itself. The waves of NN interest can be tracked down throughout its history till the 40s. The concept of the Artificial Neural Network model has several times captured a close attention as well as have been several times forsaken. For example, the model considered as the origin of NN was a hardwired architecture without the ability to learn and yet it started the research we build on up to the day. Promising results of several architectures introduced in 50s, note that the models were represented physically, were quickly shadowed by the universal Von Neumann architecture being behind the vast majority of computer architectures. Despite various important discoveries in 60s, 70s[1], and the first half of 80s[2], one of the several NN returns in computer vision is considered to be the work of LeCun et al. [1] who introduced the CNN for handwritten digit recognition. The proposed model achieved the state-of-the-art results and the CNN quickly captured the attention of the computer vision community. However, due to several causes, the general NNs were forsaken by the community only a few years later. Recently, apart from the others, the work of Krizhevsky et al. [2] rehabilitated the artificial neural networks in the computer-vision community again.

It is the approach published by Krizhevsky et al., which this thesis builds on. The CNN designed for the image restoration tasks, namely for the motion deblurring and high compression related image artifacts removal comprise the proposed unified approach. This work aims at providing the image restoration CNN models which, based on the results, are comparable with the traditional state-of-the-art engineered methods or even beyond them, to show that the idea of the end-to-end CNN mapping based approach is mature to be considered in real applications. There are various examples of motion deblurring and compression artifacts removal tasks where the deployment of the CNN based restoration may provide several benefits compared to the engineered methods. In the case of motion deblurring, it includes generally any surveillance system such as a traffic surveillance where the motion blur may worsen the car identification due to poor light conditions or a production line monitoring system, where, considering the possibility to quickly tune the deblurring model or even to have a universal model for several blur lengths and directions, it may allow using low-quality image capture devices. Examples of image artifacts removal utilization reflect the low-quality bandwidth

[1] The backpropagation, i. e. the algorithm used for training the networks, was published.

[2] The introduction of Neocognitron, which became the influential predecessor of Convolutional Neural Network

for a high amount of data transfer, i. e. images may be heavily compressed and restored on the client device. A similar situation occurs with the web images where the required storage capacity may be reduced utilizing the restoration of highly compressed images on the client device as well.

THE OBJECTIVE Based on the NN exploitation in the field of computer vision and the actual state of the art in image restoration, the main hypothesis of this thesis and the related objectives can be summarized as follows. *Most of the different image restoration methods is replaceable by a unified approach represented by CNN models which are end-to-end trained and often achieves state-of-the-art or even beyond results.* These models may differ in particular architecture or in the objectives they are trained for. The term unified covers the data-driven approach which adapts to a certain type of degradation, it does not inherently mean a single model. Different training objectives provide various speeds of convergence and rarely better models as well. The end-to-end mapping considers the direct transformation from a corrupted representation of a restored image. To provide the evidence showing the validity of such a hypothesis, two various image restoration tasks are selected. Firstly, the deblurring, namely the motion deblurring, is evaluated on the specific text images including the license plates captured by the surveillance system. In this task, the primary attention will be given on the capability of deblurring itself under the assumption of not known blur parameters, i. e. the model will provide a blind deconvolution. CNN deblurring model will be examined to reveal its capacity which may allow using a single model for a large range of possible blurs.

Secondly, it is the high compression related image artifacts removing, which substantially differs from deblurring methods. These artifacts relate to a missing image information lost by a high lossy compression compared to the blurred image where the information persists just hidden in the transformed data. Besides other aspects, the artifacts comprise a non-linear corruption compared to the linear blur degradation. In this task, the same approach of CNN as in motion deblurring comprise the unified approach. Next to the simple architecture used in the direct end-to-end mapping approach, several different objectives the model is trained for together with an architecture extension are studied. Finally, the CNN based image restoration applied directly on the JPEG coefficients instead of pixels is proposed and described. Considering the deployment of a CNN model in different data domain, the achieved results may support the idea of a single CNN based approach for different tasks of image restoration.

THESIS OUTLINE The structure of this thesis consists of five main chapters with introduction and conclusion. ENGINEERED IMAGE RESTORATION briefly introduces the motion blur and the high compression artifacts together with several engineered restoration methods. CONVOLUTIONAL NEURAL NETWORKS provides the formal definition of NN, the novel techniques used for training the models and also the description of the most important architectures that the models presented in this thesis are based on. IMAGE PROCESSING BASED ON NEURAL NETWORKS

consists of various NN and CNN based approaches used in the image processing during last 20 years with a more or less direct relation to the presented restoration approach. CNN IMAGE RESTORATION comprises the core hypothesis of this thesis with the detailed description of the objectives framed by the principle idea of a single unified approach. EXPERIMENTS provides the evaluations and results showing the validity of the presented hypothesis and also offers the possible extensions to the introduced models with the hints for further research. CONCLUSION summarizes the whole work, highlights achievements, and with a conclusion based on the results closes this thesis.

2

ENGINEERED IMAGE RESTORATION

Image restoration is generally a transformation of a damaged image on an undistorted image. This chapter introduces the selected degradations and describes various hand engineered widely used methods for their restoration. Image restoration[1], in the scope of this thesis, consists of two different inverse problems. Motion deblurring can be understood as a linear inverse transformation described as deconvolution. In contrary, a restoration of lossy JPEG compression represents the nonlinear inverse transformation which, generally, is an ill-posed[2] problem because the transformation can be non-invertible. JPEG compression, with a low-quality setting, produces the blocking artifact and the ringing — Gibbs phenomenon.

In this thesis, an image is understood as a finite matrix. Precisely, an image expressed as a continues function $f(x, y)$ of two coordinates in the plane is sampled into a matrix $M \times N$, where each sample is quantized to an integer value of K intervals [3]. Three types of images are considered, latent[3] image represents an ideal image which does not suffer from any corruption and it is denoted as x. An undistorted image represents the estimation of the latent image and is denoted as $\hat{x}$. Finally, a distorted image is the result of the process modifying a latent image and is denoted as y. In this work, the terms like degradation, corruption, damage, etc., are understood as synonyms for a general process modifying the latent images.

Both types of degradation can be decomposed into an operator applied to a discrete image and additive noise. An approximated model of degradation [4] considering the discrete property and additive noise can be written

$$y = Ux + W \,, \tag{2.1}$$

where y is the degraded image, x is the latent image, U represents the discrete operator, motion blur or JPEG artifacts, and W is an additive noise. The discrete operator U can be represented as a linear operation, i. e. convolution, or a nonlinear operation, the discrete cosine transform with quantization.

Both of degradations and the methods of its restoration are introduced. Motion blur is described with examples of some simple yet typical linear operators and its outcomes. Next, a basic motion blur Point Spread Function (PSF) estimation is introduced to compute the Wiener filter and produce the estimated sharp image. A state-of-the-art text-oriented deconvolution method L0-regularized intensity and gradient prior [5] is described to be later on compared with the introduced data-driven learned CNN based approach.

[1] As both degradations can be well modeled, the inverse transformation is therefore referred as restoration. Image enhancement, on the other hand, does not suppose a strong model.

[2] An incorrectly or improperly posed problem.

[3] The original meaning is related to exposed photosensitive material — photographic film.

JPEG compression based degradation is mentioned with the emphasize on stages of transformation pipeline where the compression artifacts come from. Methods dealing with these artifacts are mentioned with a description of the current state-of-the-art Shape Adaptive Discrete Cosine Transform method [6, 7]. The majority of hand engineered methods usually consist of several steps based on an analytical solution. This chapter briefly introduces several of such methods to highlight the difference between data-driven methods which a CNN is a part of.

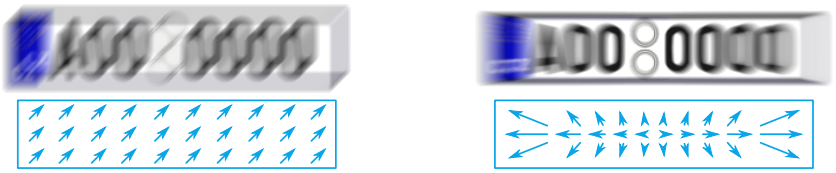
2.1 MOTION BLUR

Digital image restoration related, beside others, to the motion blur massively appeared with the space programs in 1950s. The rising amount of aerial pictures taken during the missions were often subject to many photographic degradations including the motion blur [8]. This is often caused by a shake of a camera or a moving object in the scene. Degraded images can be uniformly or non-uniformly blurred. A convenient example of the easier case, uniform blur, can be found in surveillance systems where the camera is fixed and a moving object appears captured with longer exposure. A uniform blur is represented solely by a single PSF applied on the entire image. Non-uniform blur may often be related to an optics distortion, camera rotation, or objects moving in the scene with different speed or in various distances and consists of several PSF describing the blur in a particular part in the image. Both types of blur, uniform and non-uniform is shown in Figure 2.1. Direct solution of (2.2) leads to the inverse filter with all the drawbacks mentioned further. In a case of considering the noise and keeping the assumption of linearity, the Wiener filter is usually used. This may be based on known or unknown PSF. In such a case the transformation called a non-blind or blind deconvolution. Often the existing methods of blind deconvolution concentrated in estimating the single blur PSF for the entire image. This is valid for a restricted set of applications but generally, such an assumption is far being satisfied in the case of objects which in the scene move independently.

In case of an uniform motion blur, the equation (2.1) can be derived into a model described as

$$y = x * g + w \,, \tag{2.2}$$

where y is the captured motion blurred image, x is the sharp latent image. The operator U (2.1) becomes the convolution $*$ with a shift invariant PSF g representing a degradation due to motion and optics imperfections, and, finally, w is an additive random noise with zero mean Gaussian distribution. Figure 2.2 shows the example of motion blurred license plate image with the corresponding PSF. The presented model (2.2) rarely, if any, match the realistic conditions, e. g. optics is not exactly shift-invariant, digital imaging sensors do not have the precise Gaussian distribution of noise etc.



(a) (b)

Figure 2.1: An uniform motion blur (a) with a vector field representing the spatial blur and a non-uniform blur (b).

2.1.1 *Motion Blur Restoration*

A straightforward solution based on simple deconvolution with known blur PSF yields to the *inverse filter* [9] with, in practice, straight drawbacks. The motion blur equation (2.2) can be represented in the frequency domain using the $\mathcal{F}$ Fourier Transform (FT). Regards to the convolution theorem[4], the motion blur model (2.2) is expressed as $Y = XG + W$, where $Y = \mathcal{F}(y)$, $X = \mathcal{F}(x)$, $G = \mathcal{F}(g)$, and $W = \mathcal{F}(w)$. In case the blurred image is noise free, the estimated restored image $\hat{x}$ is expressed as

$$\hat{x} = \mathcal{F}^{-1}\left(YG^{-1}\right), \tag{2.3}$$

which is simply the inverse Fourier transform $\mathcal{F}^{-1}$ of the blurred image Y divided by the inverse filter G^{-1} in the frequency domain. A drawback of such an approach is the assumption of no noise W which, in contrary, is practically always a part of the model. Rewriting (2.3) to include the added noise yields to $\hat{x} = \mathcal{F}^{-1}\left((Y - W)G^{-1}\right)$. Usually, the G is a low-pass filter which produces to near zero outcomes of high spatial frequencies. These are on the other hand strongly amplified in the case of the inverse filter.

The deconvolution based on optimizing the Mean Square Error (MSE) between a latent clean image x and the estimated restored image $\hat{x}$ and considering additive random noise w is called the *Wiener filter* [9]. It is an optimal linear filter based on the assumptions of the stationary signals and zero mean noise with Gaussian distribution. The stationarity assumption yields to a known autocorrelation in the sense it is expected to be not dependent on the spatial position in the image. The estimated restored image $\hat{x}$ and the loss function L which the Wiener filter h minimizes is written

$$\hat{x} = h * (y + w) \tag{2.4}$$

$$L = E\,(x - \hat{x})^2. \tag{2.5}$$

Optimizing the MSE of the loss function between the latent and estimated image according to h is the Wiener filter in Fourier domain described as

$$H = G^{-1}\frac{|G|^2}{|G|^2 + \frac{W}{S}}, \tag{2.6}$$

[4] Under suitable conditions the convolution in spatial domain transformed by FT is element-wise multiplication in frequency domain.

(a) (b) (c)

Figure 2.2: The sharp image x (a) blurred with the motion blur PSF g (b). The result is an uniformly blurred image y (c).

where H is the Wiener filter, G is the Fourier spectrum of a PSF, W is the mean power spectral density of noise w, and S is the mean power spectral density of the original image x. The S is usually unknown and therefore the signal-to-noise ratio $1/SNR$ is estimated instead of W/S. Having the definition of the linear optimal filter the motion blur PSF estimation is needed to compute the Wiener restoration.

The Motion Blur PSF Estimation

There exist several methods for PSF estimation and the following deconvolution. The focus of this thesis is not in these methods directly but in the machine learning based NN methods for several tasks of image processing. Nevertheless, the concepts of these engineered methods reflect into some NN based approaches as well. Yet the image processing is dominated by the engineered methods, the NN based approaches are usually compared with them.

CEPSTRUM AND RADON TRANSFORM BASED APPROACHES The straightforward approach estimating the blur PSF is to represent the blurred image so that the blur, specifically motion blur, becomes easily to estimate. There are two simple yet well working under specific assumptions approaches based on the image *cepstrum* $\mathcal{C}$ or the *Radon transform* $\mathcal{R}$. The cepstrum $\mathcal{C}$ is defined as the logarithm of the Fourier domain transformed back to the spatial representation based on the inverse FT, and is written

$$\mathcal{C}(g) = \mathcal{F}^{-1}\Big(\log\big|\mathcal{F}(g)\big|\Big), \tag{2.7}$$

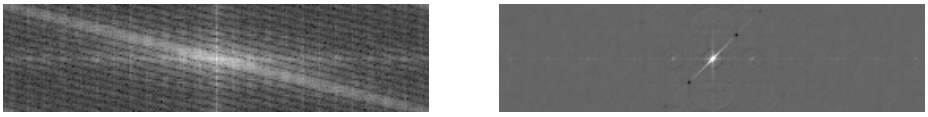
where g is the motion blur PSF. An estimation of the blur PSF parameters based on a blurred image cepstrum is described in [10].

The identification of motion blur direction and length is based on the assumption of additivity under the logarithm of convolution in the Fourier domain. The cepstrum of the motion blurred image is written

$$\mathcal{C}(x * g) = \mathcal{F}^{-1}\Big(\log\big|\mathcal{F}(x * g)\big|\Big) \tag{2.8}$$

$$\mathcal{C}(x * g) = \mathcal{F}^{-1}\Big(\log\big|\mathcal{F}(x)\big| + \log\big|\mathcal{F}(g)\big|\Big), \tag{2.9}$$

which preserves the added negative spikes of $\mathcal{C}(g)$ to the latent non-distorted image $\mathcal{C}(x)$. The angle of motion blur is approximated by the inverse tangent of the straight line slope connecting the origin with the negative peak. The length is equal to the distance between the negative peak and the origin. The cepstrum $\mathcal{C}$ of the uniformly motion blurred license plate image is shown in Figure 2.3b with two visible black dots representing the negative peaks. Problems occur in case the motion



(a) (b)

Figure 2.3: On the left (a) is the logarithm spectrum of uniformly motion blurred image Figure 2.2c with clearly visible parallel lines related to the angle of the motion blur. On the right (b) is the cepstrum of the identical motion blur image with two black dots representing the negative pikes related to blur direction and length.

is not linear and the blurred image contains a high amount of noise. According to the [10], the method based on PSF estimation from the blurred image cepstrum is accurate for various lengths till the level of noise is small.

Radon transform is, next to the cepstrum, another simple projection approach to estimate the motion blur parameters, i. e. its length and direction. The Radon transform is written

$$\mathcal{R}_{\rho,\theta}(x) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_{i,j}\, \delta\big(\rho - i\cos(\theta) - j\sin(\theta)\big)\, di\, dj\,, \tag{2.10}$$

where x is the input image integrated along the line determined by the angle θ of its normal and the distance ρ from the origin. The δ function simply returns one in case the projection line defined by the ρ, θ lies on the i, j coordinates and zero otherwise. Radon transform is usually computed in the maximum inscribed square of the input image x.

The Radon transform method based approaches [10, 11, 12] usually expect the $\log|\mathcal{F}(y)|$ input, i. e. the logarithm of Fourier transform of a blurred image. The motion blur parameters can be derived from the specific parallel lines present in $\log\big(\mathcal{F}(y)\big)$ Figure 2.3a, where the angle between the parallel line normal and the horizontal axis is equal to the direction angle θ of the motion blur [12].

The Radon transform is used to reveal these parallel lines in the logarithm spectrum of a degraded image Figure 2.4. The length is estimated as N/d, where N is the image dimension and d is the distance between two successive parallel lines. Another way to identify the motion blur parameters, e. g. the direction is to compute the $\arg\max_\theta \mathrm{Var}\Big(R_{\rho,\theta}\big(\log\mathcal{F}(y)\big)\Big)$, where the Var is the variance of the set of values obtained by varying ρ. The $\arg\max_\theta$ depends fundamentally on the orientation of the blur kernel [11].

THE IMAGE PRIOR-BASED METHODS The frequency domain-based methods, including cepstrum or Radon transform, are not used in case of complex non-uniform blur degradation. In regards, there are various methods based on the natural image statistics, i. e. natural image priors. These methods often differ in the specific image prior definition while the optimization based on the Maximum A Posteriori (MAP) is usually common.

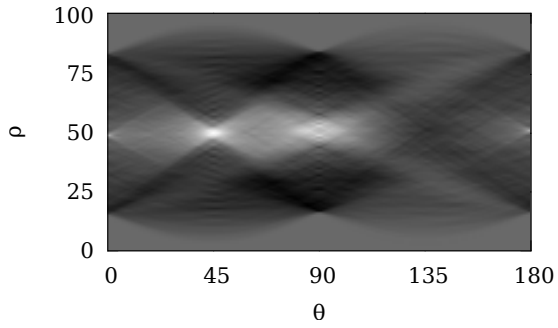


Figure 2.4: The radon transform of the logarithm spectrum Figure 2.3a of uniformly blurred image presents the spikes corresponding to the parameterization of the lines in the logarithm spectrum.

In MAP estimation, the most likely estimate of the blur kernel g is searched given the sharp image $\hat{x}$ and the observed blurred image y, using the known image formation model and noise. The general MAP estimation is written

$$\begin{aligned} P(g \mid y) &= \frac{P(y \mid g)\ P(g)}{P(y)} \\ \hat{g}_{\text{MAP}} &= \arg\max_{g} P(g \mid y)\ , \end{aligned} \tag{2.11}$$

where $\hat{g}_{\text{MAP}}$ is the estimated blur kernel, y the blurred image based on the equation (2.2), and q the unknown latent PSF.

An image prior based on the assumption the blurred edges are degraded sharp step edges in the latent sharp image was proposed by Joshi et al. [13]. The method utilizes the sub-pixel difference of Gaussian edge detector to find the location and orientation of blurred edges. The sharp edges are predicted from their blur appearance based on the sub-pixel accuracy of the edges pose. Based on the predicted sharp and given blurred image, the uniform or spatially-varied PSF is estimated using the MAP estimation. Such an approach seems to work quite well till the motion blur is in some reasonable length. In a case of longer motion blur, the methods are prone to fail. This is related to the assumption of localizing the not degraded sharp edges which can not be hold [13].

The prior based on the histogram of derivatives was evaluated by Levin [14]. The prior assumption is based on the observation that the histogram of several artificially blurred images derivatives significantly differs. The PSF can be most likelihood estimated according to change of the histogram shape. However, the presented evaluation is based on vertical blur direction only and the assumption the image contains blurred foreground and sharp background. The estimated PSF is then used for deconvolution of the blurred segment. The approach has several limitations based on the segment identification, blur direction and length estimation, which highly depend on the image statistics the model is computed from and the assumption of a simple blur model.

The approach combining the image prior based on the edges and several different yet ideally consecutive observations of the same scene introduced Cho et al.

[15]. The presented method come out from estimating the motion model based on various images of the same scene. First, the images are segmented into to several corresponding segments. According to the number of segments, the number of spatially different PSF can be estimated. Second, based on the correspondences, an affine transformation is estimated from a segment in the first image to the corresponding segment in a second image. Since the motion blurs represented by PSF are commutative [15], their application on corresponding segments with known affine transformation between can define an equation to be optimized which is regularized by the edge preserving based term. Naturally, the approach is prone to texture fewer images implying the problem with segmentation and consequently the correspondence.

Another gradient based image prior was used in [16] to estimate the camera shake PSF to consequently deblur the image. Specifically, the prior based on the histogram of derivatives of naturally sharp images was learned. The goal was to model a non-uniform blur on the assumption that the cause is based on the camera rotation with one global blur operator. Strictly written, in this case, the PSF is not the convolution kernel but the global operator. Blur kernel is estimated based on marginalizing the posterior distribution $p(x, g \mid y)$ of latent sharp image x and blur kernel g conditioned by the observed blur image y.

In case the method is focused on a specific domain, e. g. the text deblurring by Pan et al. [5], the image prior differs from the one used in natural images. The state-of-the-art considered text image deblurring method, *L0-regularized*, is based on the assumption that clean and blurred images can be differentiated based on the pixel intensities and related gradients. The pixel intensity $P_t(x) = ||x||_0$ which is a number of non-zero values, and gradients of non-zero values observed from document images $P_t(\nabla x)$ were used as the prior to estimate the PSF. The prior for text document images is thus

$$P(x) = \sigma P_t\left(x\right) + P_t\left(\nabla x\right) \tag{2.12}$$

where σ is the hyper-parameter representing the pixel intensity weight. The main idea of $P(x)$ is developed based on the assumption that text and background in the gray-scale image document without blur have near uniform intensity values. The loss function which is minimized to estimate the PSF based on the defined prior is

$$L = ||x * g - y||_2^2 + \gamma||g||_2^2 + \lambda P\left(x\right) \tag{2.13}$$

where x is the latent sharp text image, g is the blur PSF, y is the blurred observed image, γ and λ are the weights. The proposed method simply yet effectively restore the blurred text images and is considered to be the state-of-the-art for text image deblurring.

2.1.2 *Summary on Motion Deblurring*

Deblurring may provide unsatisfactory results even when the precise blur kernel is known, non-blind deconvolution, due to image noise or aspects of the image

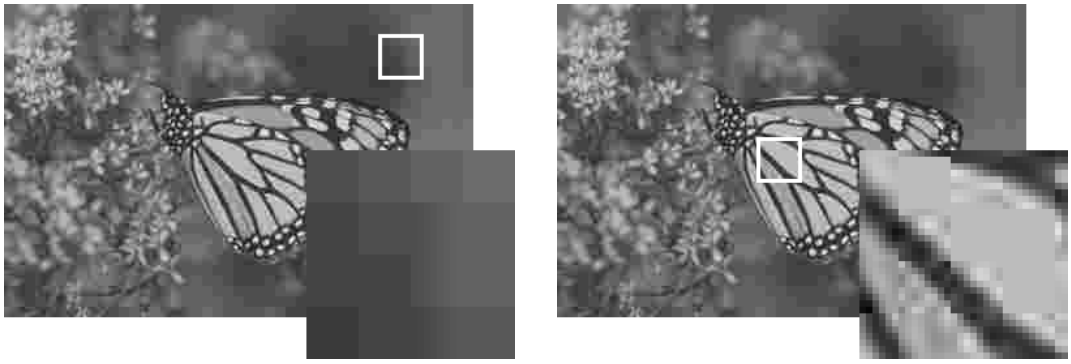
capturing process which is not present in the convolutional model. Cho et al. [17] analyzed some common types of outliers that cause deconvolution to fail, namely the pixel saturation and non-Gaussian noise. A new deconvolution method was proposed which contains an explicit component for outliers modeling. The pixel of the image is divided based on linear model blur fulfillment, inliers that can be well recovered using traditional deconvolution methods, and outliers which are iteratively refined based on expectation-maximization. The latent image is then restored solely based on the inliers.

Levin et al. [18] published their analysis of blind deconvolution algorithms. Authors stated the several published methods and algorithms are based on estimation the not-blurred image $\hat{x}$ and PSF kernel g simultaneously. The estimation of $\hat{x}$ is often build upon the natural image statistics e. g. the histogram of derivatives [14, 16] followed by the PSF estimation based on (2.2). Results of the survey pointed out that the natural image priors do not overcome the limitations of such an approach as the favorable solution under the priors usually yield to a blurry image. The research of more precise priors of natural images was not discouraged but the effort, according to the study, shall be more directed to estimators. In this work cited publications [5, 13, 14, 16] are unfortunately not compared from several reasons Levin et al. [18] but the majority tend to be based on estimating both, latent sharp image $\hat{x}$ and the PSF g. All the described motion deblurring methods have in common the engineered approach including the recent state-of-the-art methods. This is in contrary with the later presented CNN based methods which are built on the idea that the network should learn the deblurring itself.

2.2 JPEG IMAGE COMPRESSION

Citing the ITU [19] *Recommendations*, Joint Photographic Experts Group (JPEG) was formed in 1986 to establish a standard for the sequential progressive gray-scale and color images. The abbreviation JPEG used for the file format itself is an informal name for the JPEG File Interchange Format (JFIF) [20] used mainly for images processed by computer software or Exchangeable Image File Format (EXIF) [21] used by imaging cameras. A typical compression ratio of lossy JPEG is approximately 10:1. In the case of the higher compression ratio, the image degradation becomes much more perceptible indicated by the blocking and ringing artifacts Figure 2.5. This section provides the short description of JPEG compression pipeline focusing on the source of artifacts. Follow an introduction of several methods used for restoration including deblocking or deringing.

JPEG compression artifacts suppression has several considerable applications where data acquisition is expensive, difficult or demanding. For instance, the image or video playing over unreliable or low-bandwidth data connection. Image processing with low compression quality in surveillance systems encompasses application from traffic to production line monitoring. Its massive employment can be in the low-quality images preview in systems where the storage together with bandwidth capacity matters.



(a) (b)

Figure 2.5: The JPEG artifacts in the form of the blocking (a) on the left and ringing (b) on the right which is visible on the edges. The monarch image is here compressed with the quality 10 and it is selected from *LIVE1* image dataset [22].

2.2.1 *JPEG Compression Pipeline*

The compression pipeline as introduced in [19] consists of various steps which differ according the lossy or lossless compression. The first one, lossy, is Discrete Cosine Transform (DCT) based Figure 2.6 and allows depending on the characteristics of the particular image as well as on desired picture quality to set the required amount of compression. Lossy image compression, generally, achieves high compression ratios through an elimination of information that does not contribute to a human perception of images, or contributes as little as possible. The second one, lossless coding, is based on predictor definition and Huffman or arithmetic coding rather than DCT.

Firstly, the image color space is transformed from RGB to $Y'C_BC_R$ representing the luma Y', C_B blue-difference, and C_R red-difference chroma components. Usually, the chroma components are down-sampled due to lower human sensitivity to colors compared to brightness intensities. Secondly, during encoding, the input image is split into 8×8 blocks which are transformed by the forward DCT into a 64 values referred as the DCT coefficients which represent the particular frequencies the DCT block consist of. General DCT transform of 2D image is written

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{mn} \cos \frac{\pi (2m+1) p}{2M} \cos \frac{\pi (2n+1) q}{2N} , \tag{2.14}$$

where B_{pq} is the computed DCT coefficient from all the values of the image block sample X with the size M, N. The coefficient block size is equal to the image block, i. e. $0 \leq p < M$ and $0 \leq q < N$. The values of α_p and α_q are defined as

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}} , & p = 0 \\ \frac{2}{\sqrt{M}} , & 1 \leq p \leq M-1 \end{cases} \qquad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}} , & q = 0 \\ \frac{2}{\sqrt{N}} , & 1 \leq q \leq N-1 , \end{cases} \tag{2.15}$$

where in case of JPEG compression the constants M, N are equal to $8, 8$. The first value represents the DC coefficient, an average intensity for the entire block, while

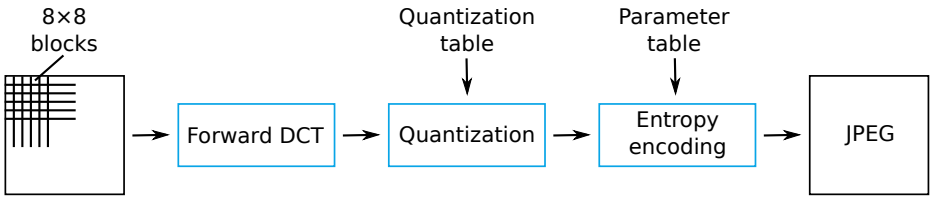


Figure 2.6: The JPEG compression pipeline with the highlighted DCT encoder part.

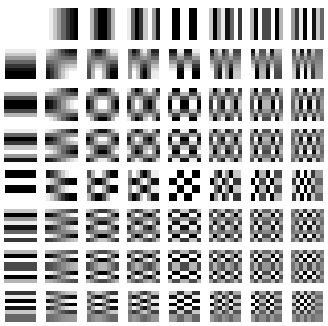
the rest of 63 values are called the AC coefficients. The DCT coefficients are quantized using corresponding values from the quantization table. The uniform quantizer [19] is defined by the equation

$$\mathcal{B}_{pq} = \text{round}\left(\frac{B_{pq}}{Q_{pq}}\right), \tag{2.16}$$

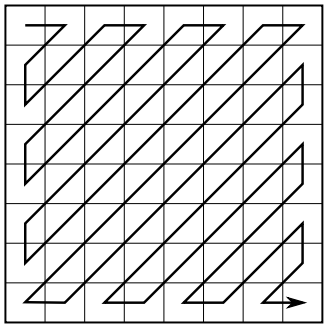
where $\mathcal{B}_{pq}$ is the rounded quantized coefficient, B_{pq} is the DCT coefficient, and Q_{pq} is the corresponding value from the quantization table. The quantization step in JPEG compression pipeline is actually the cause of non-linear degradation based on the compression amount. The quantized DC coefficient is then treated separately from the remaining quantized AC coefficients. Its value is based on the difference of the previous DC value i. e. the very first DC coefficient is the reference one for all the subsequent DCs. Next, the coefficients are passed to an entropy encoding process which, lossless, compress the data. Decoding is proceed in the reverse order, where the dequantized coefficients B are transformed by the Inverse Discrete Cosine Transform (IDCT) defined as

$$X_{mn} = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} \alpha_p \alpha_q \, B_{pq} \cos\frac{\pi\,(2m+1)\,p}{2M} \cos\frac{\pi\,(2n+1)\,q}{2N}, \tag{2.17}$$

where the notation is the same as in the DCT equation (2.14). According to image degradation sources, the main causes are 8×8 block sampling and related quantization step with following rounding operation.



(a)



(b)

40	28	25	40	60	100	128	153
30	30	35	48	65	145	150	138
35	33	40	60	100	143	173	140
35	43	55	73	128	218	200	155
45	55	93	140	170	255	255	193
60	88	138	160	203	255	255	230
123	160	195	218	255	255	255	253
180	230	238	245	255	250	255	248

(c)

Figure 2.7: The cosine basis visualization (a) with the indicated *zig-zag* direction (b) and a particular quantization table used for the quality Q20 (c).

2.2.2 *JPEG Artifacts Restoration*

The common JPEG related artifacts are the blocking and ringing. Both are tightly tied together. Block artifact is based on the 8×8 block splitting and subsequent quantization. The block segmentation employed in the JPEG standard results in discontinuities at the blocks dividing edges. The more it is apparent the more high frequencies are suppressed. The ringing artifact (Gibbs phenomenon) is the induced oscillation resulting from the inverse DCT of quantized block values and proceeds from the same cause as the blocking artifact — loss of the higher frequencies Figure 2.5.

A large number of methods designed to reduce compression artifacts exist ranging from relatively simple and fast hand-designed filters [23] to fully probabilistic image restoration methods with complex priors [24].

One of the base post-processing approach widely used in the FFmpeg[5] framework is the Simple Postprocessing (SPP) filter [23]. The method is based on the idea of re-encoding the decoded JPEG image with the several shifts i. e. encoding various shifted overlapping blocks. The result pixel intensity is equal to the weighted mean of all the contributing re-encoded blocks. Despite relative simplicity this post-processing enhancement method performs well compared to other methods operating on block boundaries in all bitrates, i. e. various quality compression.

In video compression domain, advanced in-loop filters (deblocking and sample adaptive offset filters) known from video compression standards like H.264 or H.265 are obligatorily applied.

A completely different approach is the restoration based directly on the DCT coefficients which were introduced in [25]. Authors applied the DCT-based lapped transform on the signal already in the DCT domain in order to undo the harm done by DCT processing. According to the paper, the odd-symmetric DCT coefficients excessive energy indicate the blocking artifact. The incorporating non-linear weighting of such DCT coefficients provides the selective removal of the blocking artifact without affecting the real structure of the image.

A document image model prior incorporated into the decoding JPEG images mainly based on the text was introduced in [24]. The image is segmented via its luminance component into three disjunct regions based on its content i. e. the picture, background, and text. While picture regions are decoded with a JPEG decoder, background and text regions are decoded with an appropriate algorithm designed for the given block class. Text and background decoding is posed as an inverse problem in Bayesian framework based on the regularized MAP. In a case of document images, the method is able to decode the text blocks such that they are free from ringing artifacts with largely smoothed block artifacts.

Currently considered the state-of-the-art deblocking method is the Shape Adaptive Discrete Cosine Transform (SA-DCT) [6, 7]. The thresholded or attenuated transform coefficients are used to reconstruct a local estimate based on a local polynomial approximation of the signal within the adaptive-shape support. Since

[5] A complete, cross-platform solution to record, convert and stream audio and video.

this shape supports can overlap the possible local overlapping estimates are therefore averaged using adaptive weights that depend on the region statistics. However, similarly to other deblocking methods [23, 25], SA-DCT over smooths images and it is not able to sharpen edges.

2.2.3 *Summary on JPEG Artifacts Removal*

Several of the introduced methods, the video in-loop deblocking methods, SA-DCT deblocking (only to estimate parameters), and methods derived from the lapped DCT transform rely on the cognizance of the DCT grid. The described sample of JPEG restoration methods can be divided into two main categories, pixel based, where the restoration is solely done in the image domain, and the DCT based working primarily with the coefficients. Restoring the image directly from coefficients violates the standard [19] and leads to incompatibility with existing decoders. Though the idea of processing the DCT directly can benefit from the dequantization step or already mentioned lapped transform. A lot of pixel based methods exist including simple averaging the shifted re-compressed image blocks up to a sparse filtering based on machine learning methods. Later in the work introduced CNN based JPEG artifacts reduction contributes to both i. e. DCT coefficients and pixel based categories.

2.3 SUMMARY ON ENGINEERED IMAGE RESTORATION

Two different image degradation types were introduced, motion blur and the JPEG related artifacts. The blur in the image is usually a consequence of a single reason, the long exposure time, which is often caused by several factors. The motion blur is a linear transformation where the image information is not reduced but only transformed. This yields to the straightforward solution, i. e. the deconvolution of the blurred image to restore the latent sharp image. Several related problems can and often do occur like the noise in the image which makes the deconvolution hard and requires specialized approaches. Often the estimation of PSF is performed with the external knowledge represented like the prior as for example the distribution of gradients in the sharp image.

The JPEG artifacts solely caused by the high compression ratio differs from the motion blur primarily in lost image information. The artifact removal is therefore completely different from the methods for deblurring. However, the prior in the form of a regular grid is often used to deal with the blocking artifacts. An important thing to notice is the diversity of approaches the engineered restoration consists of.

3

CONVOLUTIONAL NEURAL NETWORKS

A Convolutional Neural Network is a merit of this thesis. Their theoretical background, together with the description of image degradation, comprises the basis of the CNN image restoration. The purpose of this chapter is to provide the formal definition of NN and based on this to gradually move to a formulation of CNN. Particular drawbacks, or precisely features, the NN have, and which directly arise from the provided definition, directs the attention to CNN. Although, there are other reasons and probably more important, if compared to the NN drawbacks, why the CNN was proposed and recently prioritized. Regardless, these are included in the following CNN related paragraphs.

The description of two important CNN application is given. The majority of the recent approaches, including the one presented in this work, are directly based on the later introduced application, the *Imagenet classification* by Krizhevsky et al. [2]. However, the older predecessor, *handwritten digit recognition* published by LeCun et al. [1], defined the convolution filter as shared weights throughout the layer which on all the recent approaches build on.

Several today training related extension, modification, and approaches are described. The majority of them is accepted as the methods allowing for training deep models. Herein it is the activation function or a batch normalization layer. Furthermore, several approaches are considered providing better network model capabilities as described convolution stacking.

Nevertheless, first, a brief introduction to a history of NN and CNN is given. Interestingly, there are three noticeable periods when the CNN were in the center of attention throughout the last 70 years. The very first wave of interest starts with the general introduction of NN architecture in the form of the simplified biological neuron-inspired unit called Temporal Propositional Expressions (TPE) [26] which was later named as McCulloch Pitts Neuron (MCP). Later on, Perceptron [27, 28], a binary classifier with a linear decision boundary, was presented. The significant architecture and its application were Adaptive Linear Element (ADALINE) subsequently arranged into a network known as Many Adaline (MADALINE) [29]. While the MCP related work did not provide any learning algorithm, Perceptron provided a class error based weight update, and ADALINE introduced the gradient descent of quadratic error-based learning approach. ADALINE was expected to be a predecessor for an adaptive computer, a piece of hardware able to modify its weighted connections. Despite its application in speech recognition, echo suppression, and weather forecasting, the concept of parallel processing was overshadowed by a rapidly developing digital computers based on serial processing Van Neumann architecture. As Widraw noticed[1], MADALINE based computer

[1] Science in action Dr. Widrow's youtube: https://goo.gl/iuJPky

supposed to be used for ten years, starting from the beginning of the sixties, for variety pattern recognition problems, language translation, information retrieval and other data processing tasks. The overall expectation at that time was that this concept would become a major innovative in data processing. Widraw was right. However, he was wrong with the timing.

Von Neumann architecture based computers put the NN on a sidetrack till the eighties. During the time of NN interest decline, in 70s, there had been published a thesis of Werbos [30] which proposed backpropagation learning for NN. Unfortunately, this work passed unnoticed for a decade till the backpropagation was re-invented [31] or more precisely experimentally shown that it can be used for NN training. Besides that, substantially important became the Convolutional Neural Network introduced by LeCun et al. [1] in computer vision. The concept of CNN showed a tremendous success recognizing the handwritten zip codes. Despite this, the NN and CNN also became again forsaken for next decade because of several reasons mainly based on the lack of sufficient amount of appropriate data, and the unavailable computation power [32].

The third "renaissance" of NN begins in 2012 with the significant result the architecture of Krizhevsky et al. [2] provided in ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Plethora of methods and extensions based on the CNN were subsequently published and directly influenced this work. Among the relevant methods, this includes, besides others, the work of Long et al. [33] and Noh et al. [34] focused on a Fully Convolutional Network for image semantic segmentation, Hradis et al. [35] and their CNN application for text image deblurring, and Dong et al. [36, 37] with their direct mapping approach of superresolution network later used for image artifacts reduction as well. Nowadays, the significant impact of the CNN based methods is apparent in various fields where some of them are not directly related to computer vision.

3.1 NEURAL NETWORKS

A simple neuron which is building unit of a fully connected feed-forward NN is defined as a dot product followed by the activation function and is written

$$\begin{aligned} y &= f\left(\boldsymbol{w}, \boldsymbol{x}\right) \\ f\left(\boldsymbol{w}, \boldsymbol{x}\right) &= h\left(\sum_{i=0}^{M} w_i x_i\right), \end{aligned} \tag{3.1}$$

where y is the output, $h(\cdot)$ represents the non-linear activation function, often a sigmoid for its simple differentiation, and $\boldsymbol{w}$ denotes the weight vector of size $M+1$ which is because it consists of a bias w_0 and the weights w_i where $1 < i \leq M$. The vector $\boldsymbol{x}$ includes an additional bias $x_0 = 1$ and the input data x_i where $1 < i \leq M$. The result of a dot product $\sum_{i=0}^{M} w_i x_i$ defines the activation [38]. The NN is therefore a triplet defined as $\langle \boldsymbol{x}, \mathcal{W}, \cdot \rangle$, where $\boldsymbol{x}$ is the input vector, $\mathcal{W}$ is a set of weights vectors and $\cdot$ represents dot product.

Arbitrarily wide (N neurons in layer) and deep (L number of layers) restoration network can be defined

$$\begin{aligned} \boldsymbol{y}_0 &= \boldsymbol{x} \\ f_l &= h_l\left(\mathcal{W}_l \boldsymbol{y}_{i-1}\right) \\ F_L\left(\mathcal{W}, \boldsymbol{y}_0\right) &= \left(f_L \circ f_{L-1} \circ \ldots \circ f_1\right)\left(\boldsymbol{y}_0\right), \end{aligned} \tag{3.2}$$

where $f_i(\cdot)$ is the i-th layer which include several neurons and their activation functions $h_i(\cdot)$, the neurons weights of i-th layer are represented by a subset of vectors $\mathcal{W}_i$, and the layers arrangement is defined via a composition operator $\circ$. All the trainable parameters are represented as a set $\mathcal{W}$ of vectors $\boldsymbol{w}$. There exists a subset of weights vectors $\mathcal{W}_l$ per layer l. The notation follows

$$\begin{aligned} \mathcal{W}_l &= \{\boldsymbol{w}_0, \ldots, \boldsymbol{w}_{N-1}\} \\ \mathcal{W} &= \{\mathcal{W}_1, \ldots, \mathcal{W}_L\}. \end{aligned} \tag{3.3}$$

With the formulation of NN, the drawbacks of its application in computer vision are introduced. NN is not implicitly invariant to several transformations [38]. For example, these transformations include translation, rotation, and scale. Despite this, there are various approaches which support or even provide the invariance to the NN based application.

DATA AUGMENTATION The training data can be augmented by all possible cases of the transformation which the model should be invariant to [39]. The augmentation may be based on the artificial training data transformation before they are presented to the network during training.

BUILD INVARIANCE Another way how to achieve an invariance is to provide an architecture which is naturally invariant to a particular transformation. The CNN is a common example of a model based on the neurons which act as filters. These filters, represented as shared trainable weights, are locally computed from the entire input with. Together with max-pooling layers, such networks can reach the translation invariance.

The NNs without the non-linearity in its activation functions $h(\cdot)$, actually represent the linear model regardless the number of layers and can be directly replaced by a single layer network. The function the NN performs can be defined according to the last layer output. In the case of single neuron output with a thresholded value, the network provides a binary classification. If the activation function of such a neuron is a sigmoid, the network provides a thresholded logistic regression of preprocessed data. The network with the last layer providing the softmax function computes the multi-classification. Finally, the network with the last layer without the activation function performs the regression.

3.2 CONVOLUTIONAL NEURAL NETWORKS

A Convolutional Neural Network represents a feed forward network designed in regards to the observation and several studies[2] concerning on processing the visual stimuli by mammals. A comprehensive theory exists how the visual cortex, the part of a brain responsible for vision, works. The primary visual cortex consists of cells which are sensitive to simple and complex features. In a CNN, these sensitives cells are represented by spatial filters implemented as convolutions. The neurons, i. e. the convolutional filters, exploit the strong local spatial correlation which often appears in the natural images. These filters are implemented as convolution kernels where the kernel values represent the trainable parameters. From the implementation point of view, the image is not convolved with a single kernel but with several kernels sharing the weights. Sometimes, the neurons represented by the convolutional filters are therefore denoted as the *shared weights*. The network usually first learns filters similar to several simple wavelets which can be found in its front layers. The network with added subsequent layers can compose these simple wavelets and therefore builds complex features. The intermediate results of feed-forwarding the input data, i. e. the outcomes of convolutional layers, are denoted as the activation respectively feature maps. These represent the filtered input of the previous layer with accentuated responses on the learned filters. A CNN can be mathematically expressed as well as the NN formulation, i. e. it is a composition of networks layers represented by functions as was defined in (3.2).

The NN can be reformulated into the CNN as a triplet $\langle x, \mathcal{W}, * \rangle$, where x represents the input image, $\mathcal{W}$ is the set of weight tensors, and $*$ is the convolution operator. Usually, the neurons, i. e. filters in case of CNN, comprise a tensor with a spatial dimensions w, h and number of channels ch. The set of weight tensors $\mathcal{W}$, therefore, includes subsets of weight tensors $\mathcal{W}_l$ per layer. A subset l of layer weights is $\mathcal{W}_l = \{\boldsymbol{W}_0, \ldots, \boldsymbol{W}_N\}$, where $\boldsymbol{W}_n$ represents a particular nth filter weights.

The *Neocognitron* networks published by Fukushima and Miyake [40] at the beginning of 80s were considered to be a predecessor of CNNs. The network which was used for handwritten digit recognition by LeCun et al. [1] at the turn of 80s and 90s was the milestone in CNN showing potential power such models can have. Twelve years later, Krizhevsky et al. [2] published a work on a deep CNN for image classification, a breakthrough CNN, which became the fundamental architecture in various other computer vision related tasks.

CNN FOR HAND-WRITTEN DIGIT RECOGNITION LeCun et al. [1] used the network shown in Figure 3.1 with shared weights based on the assumption that features useful in one part of the image is likely to be useful in other parts as well. A neuron, in such a case represented as small convolution filter with the local receptive field, a filter size, is applied to the image with its states stored in the feature map. Besides the reduction of a lot of parameters, weights $\boldsymbol{W}$, the network be-

[2] Starting with work of Hubel and Wiesel who were for their work on the information processing in the visual system awarded in 1981 by Nobel Prize.

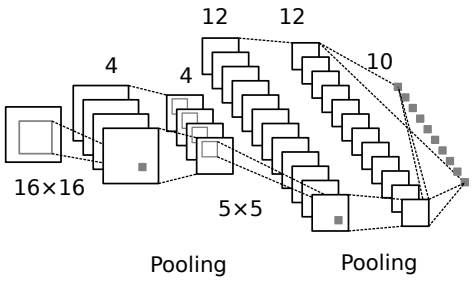


Figure 3.1: The architecture of LeCun et al. [1] for hand written digits recognition with the shared weights in the 1st and 3rd layer, pooling 2nd and 4th layer, and 5th dense fully connected layer.

comes spatial invariant based on the neuron's local receptive field. The mentioned parameter reduction allowed training the model in a reasonable time even in the beginning of 90s. The following layers were designed similarly, ie neurons convolving the feature maps of previous layers. The only different was an outermost fully-connected layer with ten outputs providing the digit probability. The original network consists of 4 layers where first and third layer are considered to be the feature extractors while 2nd and 4th the sub-sampling layers. The concept of shared weights layers followed by the sub-sampling layers is considered as a reminiscent of neocognitron architecture published by Fukushima and Miyake [40].

Neocognitron together with its successor CNN were primarily inspired by the architecture of primary visual cortex. The very first idea was to achieve the ability to recognize stimulus patterns according to the differences in their shapes. Neocognitron provided several ideas to build CNN, however, the time it was proposed, a backpropagation used for learning was not widely known, and the network was unsupervised trained via self-organization [40]. The CNN trained with backpropagation showed significant results at the time the handwritten digits recognition was published. Although, there were still problems to train these networks contributing to the fact that other well-working machine learning approaches assembled from separated feature extraction and classification, features and support vector machines, for instance, overshadowed the whole approach.

CNN FOR LARGE SCALE VISION RECOGNITION Krizhevsky et al. [2] successfully trained a deep CNN later known as *AlexNet* in the ImageNet Large Scale Visual Recognition Challenge and achieved substantial results which overcame at that time the state-of-the-art methods by more than 10 %, i. e. from 26.2 % to 16.4 % in top-5[3] error classification task.

Such improvements are often attributed to several aspects including the available computation power in the form of Graphics Processing Unit (GPU), availability of large datasets as *ImageNet* [41], and the Rectified Linear Unit (ReLU) [32, 42]. While two first reasons are understandable, it the ReLU activation function which

[3] The quality of a labeling was evaluated based on the label that best matches the ground truth label for the image of top 5 classes.

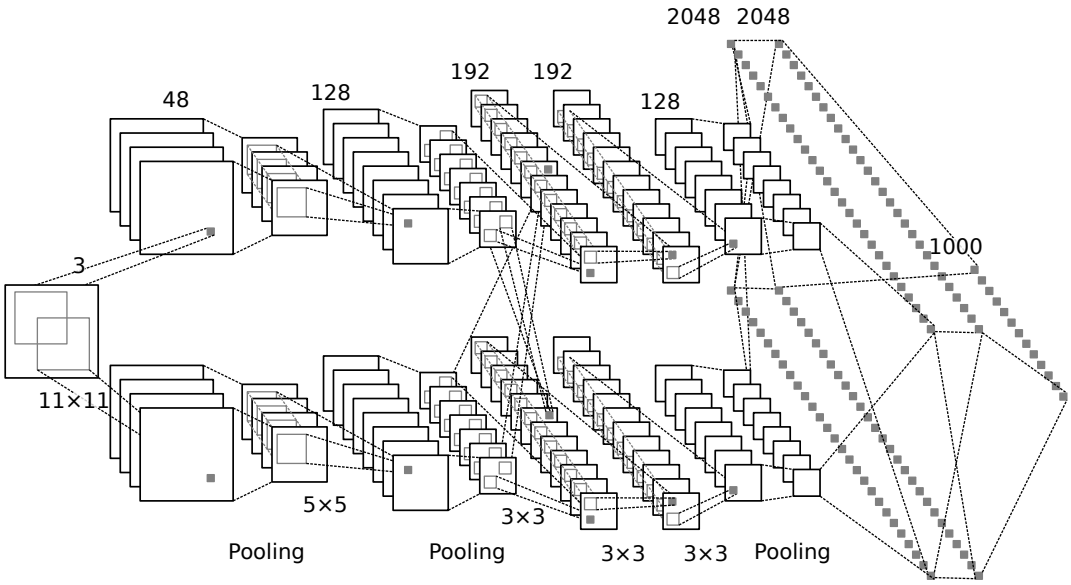


Figure 3.2: The architecture of Krizhevsky et al. [2]. The model was designed to run on two GPUs, i. e. half channels of the particular layer were per GPU. The last three layers are dense fully connected.

contributes a lot to the wide spread of depth CNN. The ReLU primarily helped to suppress the problem of vanishing gradients.

The CNN of Krizhevsky et al. [2] consists of 5 convolutional and 3 fully-connected layers Figure 3.2. The importance of the convolutional layers was analyzed as it, in this particular model, contains less than 5 % of all network parameters. The reduction of the weights related to convolutional layers resulted in an inferior performance which points out the importance of learned features and also supports the assumption of the importance of the network depth.

While training data consisted of quite a large part of ImageNet[4], the network, despite, tended to over-fit due to its enormous number of parameters—60 millions. Besides the data augmentation, a problem of over-fitting was substantially reduced based on a dropout introduced in [43] and later in detail analyzed in [44]. The idea of dropout comes from the assumption that model combination often improves the performance of machine learning methods. There are randomly dropped out several hidden or visible neurons and all its incoming and outgoing connections which yield to a smaller amount of weights, i. e. model, and at the same time efficiently combining many different CNN architectures. The dropout therefore forces the network to learn more salient features [43, 44].

In the following years of ILSVRC competition, CNN based approaches became the primary matter of an interest bringing other new concepts based rather on architecture, i. e. more deep and complex structures. The CNN became the main framework several later introduced approaches are based on. An example is the very deep convolutional network *VGG* of Simonyan and Zisserman [45], or the *Inception* network of Szegedy et al. [46] utilizing the idea of network in network

[4] ImageNet consists of more than 14 M images.

and at the same time reducing the number of weights of 60 millions in *AlexNet* with the top-5 error 16.4 % to 5 millions in *GoogLeNet* with 6.6 % top-5 error, which became the winner of ILSVRC 2014. In 2015, the best model *ResNet* [47] achieved 3.6 % top-5 error with a deep network of 152 layers.

3.2.1 *Convolutions in CNN*

Convolutions in CNNs represent the workhorse of a network computation. The GPU architecture and fast matrix algorithms focused the attention to convolutions represented as matrix multiplication. A discrete 2D convolution with a filter g of size $(w/2+1, h/2+1)$ is written

$$(x * g)_{m,n} = \sum_{j=-h/2}^{h/2} \sum_{i=-w/2}^{w/2} x_{m-i,n-j}\, g_{i,j}\,, \tag{3.4}$$

where x represents the convolved image. Several vectorization approaches based on unrolling the convolutions and utilizing the GPU were published [48, 49, 50]. The tensor $\boldsymbol{x}_T$ lowered to a matrix $\boldsymbol{x}_M$ and a tensor the filters are stored in $\boldsymbol{g}_T$ lowered to a matrix $\boldsymbol{g}_M$ shows the Figure 3.3 is based on [48] latterly abbreviated as *im2col* method. A drawback of this approach is the duplication of elements overlapping in the receptive field i. e. when the stride is smaller than the filter size. The result matrix representations size depends on the input data size $\boldsymbol{x}_T = [m, n, c]$ width, height, channels, filter size $\boldsymbol{g}_T = [h, w, c]$ filter width, filter height, channels, and its number f, and the stride s the step size filter slides over the input data. Matrix $\boldsymbol{x}_M$ is than $(m-h+1) \times (n-w+1)$ and matrix $\boldsymbol{g}_M = hwc \times f$.

Currently, the trend is to use smaller filters, for instance of the size 3×3, stacked in more layers. Such a pattern is distinct from the ILSVRC challenge, specifically, *AlexNet* uses in the first layer the filters size 11×11, *VGG* network is based solely on 3×3 filters, and *GoogLeNet* factorizes 7×7 filters used in the year 2014 into several smaller convolutions [51] a year later.

Another trend is a convolution architecture based on utilization of a convolution separability. This leads to stack of horizontal convolution of $1 \times n$ form and its

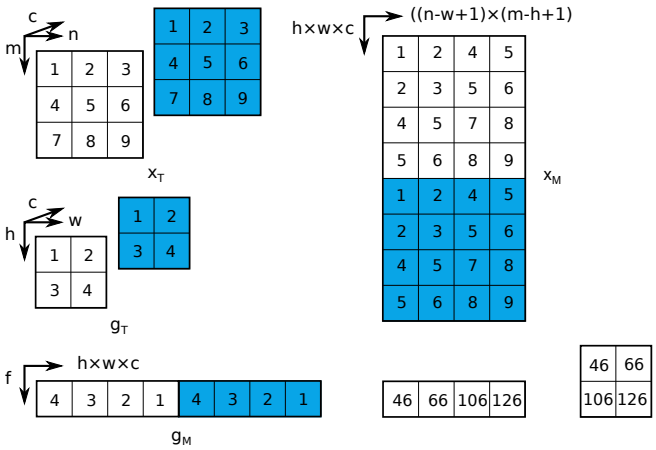
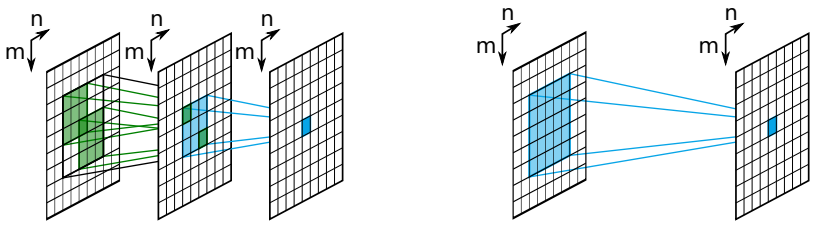


Figure 3.3: Convolution lowered into the matrix multiplication. The tensor of data x_T and filters g_T is lowered into the data x_M and filter g_M matrix.



(a) (b)

Figure 3.4: The illustration of stacked convolutions (a) compared to single large convolution kernel (b).

orthogonal corresponding convolution $n \times 1$ which yields to a spatial convolution with $n \times n$ filter. Such an architecture can be found in the *Inception* of *GoogLeNet* network [46, 51].

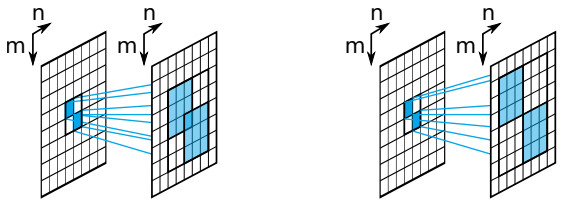
The benefit of stacking the smaller convolutions is merely based on reducing the parameters, increasing the number of nonlinearities, and often decreasing the number of floating point operations compared to convolution with one big filter. The effectiveness related to multiplication and add operations increases with the input size. On the other hand, the filter size affects the number of weights $\boldsymbol{W}$ in the network. Therefore the convolution theorem, i. e. the convolution expressed as a point-wise product in the Fourier domain, is recently in the shadow of fast matrix multiplication implementation. Its utilization would be appropriate in case of bigger filters where such an assumption is in contradiction with actual trend of computing small stacked filters.

3.2.2 *Fully Convolutional Network (FCN)*

In both works of LeCun et al. [1] and Krizhevsky et al. [2] which steered the attention towards CNN, the number of weights in the convolutional layers was in the substantial minority compared to a number of weights in the fully connected layers. The part of the network which consists of convolutional layers represents the feature extraction while the fully connected layers represent the classification part. Nevertheless, the fully connected layers can be replaced by the convolutional layers with filters of size $[1, 1, c]$ which provide the ability to compute an arbitrary input data size. The FCNs are usually referred as end-to-end or pixel-to-pixel. Semantic segmentation followed by several image processing approaches became the important application where FCN were used.

Long et al. [33] introduced the per pixel image segmentation based on networks omitting the fully connected layers trained directly pixel-to-pixel. This work encapsulates several ideas currently dominant in other works. Authors directly transfers the recent classification models *AlexNet*, *VGG*, or *GoogLeNet* to dense prediction based on the fine-tuned FCN architecture.

The original networks based on fully connected layers highly amortize the computation in a case of overlapping input patches. For instance, *AlexNet*'s input of fixed size 227×227 produces a class vector which in a naïve use of the patch-to-



(a) (b)

Figure 3.5: The examples of two transposed convolutions, in CNN known as deconvolution, with a kernel size 3 and stride 2 on the left (a) and the same size kernel but stride 3 on the right (b).

pixel classification would consist of repeated computation of the overlapping data again and again. Such waste calculation is suppressed by using the FCN on bigger inputs, i. e. reducing the number of overlapping patches.

In a feed-forward step, the input image is processed by several *convolutional* layers which yields to cropping the image, if no padding is used, and the *pool* layers which subsample the data. The upsampling was computed by the trainable backward convolution, i. e. a deconvolution layer, which maps convolution output back to its input. Because of relatively coarse results, the FCN was extended by a skip architecture [38] delivering the finer spatial information to more deep layers. Final results, evaluated beside others on Pascal Visual Object Classes (Pascal VOC), outperformed the state-of-the-art methods including R-CNN giving the 20 % relative improvement, i. e. the 62.7 % mean intersection over the union between ground truth and the predicted segmentation.

Noh et al. [34] extended the FCN of pixel-wise semantic segmentation pointing out several limitations of Long et al. [33] work. Namely, the problem of fine details related to a scale of the object in the image was addressed. The approach of deconvolution with the skip architecture was replaced by unpooling, and deconvolution layers included in a deeper model denoted a *DeconvNet* (30 compared to 8 layers [33]) which expects the already proposed object instances in the image as the input. *DeconvNet* consists of two parts, first 15 layers of the *VGG* network are followed by the same but mirrored architecture providing the up-sampling based on the unpooling and deconvolution layers.

The unpooling of max-pooled regions is the non-invertible operation approximated from recorded maxima locations within each pooling region. The deconvolution architecture, in contrary to the convolutional one, firstly creates a hierarchical structure where the first filters capture the overall shape while the later deconvolution filters provide the class-specific fine details. In summary, it behaves the opposite way compared to convolutional architecture. The reported results on Pascal VOC were 72.5 % of mean intersection over union.

This work is primarily based on the concept of FCN which, related to image processing, was introduced in several recent papers and its applications are quoted in Chapter 4.

3.3 NETWORK TRAINING

Given the NN architecture with an arbitrary width and depth, the model is usually trained with a gradient descent based approach [52] utilizing the backpropagation for efficient gradient computation. An objective function represented by the optimization of a loss function, specifically a minimization in a case of gradient descent, is in the case of regression output usually based on the square ℓ^2-norm where ℓ corresponds to a particular ℓ^p space. The minimization of a loss function is written

$$\arg\min_{W} \frac{1}{2} \sum_{i=1}^{N} \left\| F_L\left(\mathcal{W}, \boldsymbol{x}_i\right) - \boldsymbol{t}_i \right\|_2^2 \tag{3.5}$$

where $F_L\left(\cdot\right)$ is the NN output based on the input vector x_i, model weights $\mathcal{W}$, and $\boldsymbol{t}_i$ represents the ith related ground truth vector. The selection of a loss function depends on the task the NN should be trained for. The loss function corresponds to minimizing the cross-entropy between an empirical distribution defined by the training data[5] and a distribution described by the model. For instance, the classification is usually based on the cross-entropy minimization between the Bernoulli distribution used for binary classification or soft-max distribution used for multi-class classification with the distribution defined by the training set. On the other hand, the loss function as defined in (3.5) represents the cross-entropy between the Gaussian and empirical distribution [53].

Usually, the loss function of deeper architecture with non-linearity activation functions is non-convex, i. e. it does not guarantee the optimization will converge. The non-convexity seems like an inconvenient property because the model optimization tends to get stuck in the local minimum. Such an observation emphasizes the importance of the initialization as the achieved minima can differ principally based on the optimization beginning. Empirical results show that the several local minima of non-convex loss function usually provide more or less well working models in case of similar initialization.

3.3.1 *Initialization*

The NN training starts by an initialization of the network, i. e. its weights. The importance of the initialization was quite underestimated till the introduction of unsupervised pretraining based on the deep belief nets [54] and Auto-Encoders (AEs) [55] training. Various recommendations based on empirical observations were summarized and gradually extended in several papers including [56, 57, 58]

The purpose and impact of the initialization strategy are solely related to the gradient based optimization of the loss function (3.5). The typical trait of the high dimensional space which the NN model is part of is a problem of global optimum which is practically never reached. Therefore, the optimization often gets stuck in the local optima which rely heavily on the network initialization. The problem of

[5] The size of training data is an application related value which spans from hundreds up to millions of samples.

various local optima existence can be partial suppressed. The standard approach is, with some constraints, to randomly and several times initialize the network consequently train it and in the end select the best performing model or combine better-working models together. Such an approach is handled by the technique called *dropout* [44] which also provides the regularization of the network.

Generally, the random initialization of the network should fulfill two main assumptions. Firstly, the initialization of layer weights should avoid or break the symmetry [56]. The symmetry is indicated as identical weights values shared between neurons of the same layer. Based on the same weights values of the neurons, i. e. the symmetry, the network produces the same output, hence having the same gradient and therefore performing the same update which yields to same change for all the symmetry neurons.

Secondly, the input values, the training data, are assumed to have variance 1, expected to be normalized, transformed to have the mean around zero and to be uncorrelated if possible. Following the expected input, the weights should be relatively small numbers with a reasonable variance [42, 58]. Various approaches to getting the proper values in the initialization are briefly described in the following text.

The earlier proposed initialization emphasized the weights values with the range over the *tanh* linear regions. These are all the values around zero, where the tanh function behaves almost linearly Figure 3.7b. That should keep enough large gradients and force to train the network the linear part of the mapping before the more complicated non-linear part. The weights were suggested to be initialized by randomly sampled values with a zero mean and a standard deviation based on the number of neuron inputs, i. e. the number of trainable convolutional filter coefficients

$$\mathrm{Var}\left(\mathcal{W}_{ij}\right) = \frac{1}{n_{ij}}\,, \tag{3.6}$$

where $\mathcal{W}_{ij}$ represents weights of a single neuron j in a layer i and n_{ij} number of the inputs to a neuron [58]. In the equation above the variance instead of standard deviation, i. e. the square root of the variance, is used.

Based on the study focused on properties of backpropagated gradients, the *Xavier* initialization [57] was proposed

$$\mathrm{Var}\left(\mathcal{W}_{i}\right) = \frac{2}{n_i + n_{i+1}}\,, \tag{3.7}$$

where the variance is related to the number n_i of neurons of the ith layer and the number n_{i+1} of neurons in the following layer. The Xavier initialization allows avoiding the time consuming per layer pretraining described in Image Denoising Based on Auto-Encoder which was formerly used to initialize deep models. The Xavier initialization reflects the back-propagated gradients related to the networks using the sigmoid activation function which, as was noticed, is almost linear around zero and outputs the gradient for the negative input values as well Figure 3.7a. In contrary, the ReLU Figure 3.7c activation function always returns

zero for non positive input, which yielded He et al. [59] to propose a modified initialization

$$\mathrm{Var}\left(\mathcal{W}_i\right) = \frac{2}{n_i}\,. \tag{3.8}$$

Taking such a property into consideration. Here n_i represents the number of neurons in the ith layer.

The initialization of a deep model is still under research. There exist recently published papers, which evaluate the impact of several initialization types as for instance the work of Dmytro and Matas [60].

3.3.2 *Backpropagation*

The weights $\mathcal{W}$ of particular NN are updated based on a backward propagation of errors, the backpropagation. Backpropagation comprises a chain rule of loss function differentiation w.r.t. network's weights $\boldsymbol{W}$. Backpropagation in the form of a computational graph can be used to compute the required gradients using a reverse automatic differentiation [61]. That allows decomposing the backpropagation computation on layer related differentiations, which, for instance, can be seen in caffe framework [62]. A simple 2 layer network with its partial differentiations used in backpropagation is shown in Figure 3.6 and is written as

$$F_2\left(\mathcal{W}, x\right) = \mathcal{W}_2\left(h_1\left(\mathcal{W}_1 x\right)\right), \tag{3.9}$$

where x represents the input data, $\mathcal{W}_1$ are the weights of first layer as defined in (3.3), $\mathcal{W}_2$ are the weights of second layer, and h is the activation function. For simplicity and clarity, the single NN components are substituted into a functional form

$$F_2\left(\mathcal{W}, x\right) = f_2\Big(h\big(f_1\left(x\right)\big)\Big)\,, \tag{3.10}$$

where x is the input data, $f_1\left(x\right) = \mathcal{W}_1 x$, $h(\cdot)$ is the activation function and $f_2 = W_2 h(\cdot)$. In accordance with this NN, the backpropagation is computed as network graph path factorization which represents the chain rule. Differentiation of the loss function $C\left(\cdot\right)$ defined in (3.5) is thus written

$$\frac{\partial C\big(F_2\left(\mathcal{W}, x\right), t\big)}{\partial \mathcal{W}}\,. \tag{3.11}$$

Factorizing the computational graph, the derivation w.r.t. $\mathcal{W}_1$ and $\mathcal{W}_2$ of the loss function $C(\cdot)$ is equal to

$$\begin{aligned}\frac{\partial C}{\partial \mathcal{W}_2} &= \frac{\partial C}{\partial f_2}\frac{\partial f_2}{\partial \mathcal{W}_2}\\ \frac{\partial C}{\partial \mathcal{W}_1} &= \frac{\partial C}{\partial f_2}\frac{\partial f_2}{\partial h}\frac{\partial h}{\partial f_1}\frac{\partial f_1}{\partial \mathcal{W}_1}\,.\end{aligned} \tag{3.12}$$

The weight updates $\triangle\mathcal{W}$ are computed based on the partial derivatives $\frac{\partial C}{\partial \mathcal{W}_1}$ and $\frac{\partial C}{\partial \mathcal{W}_2}$ based on various gradient descent algorithms. Backpropagation is highly affected by the gradients size. The size of a gradient in the case of a saturated neuron with the sigmoid-based activation function becomes almost zero and yields to a problem of vanishing gradients the backpropagation may suffer with.

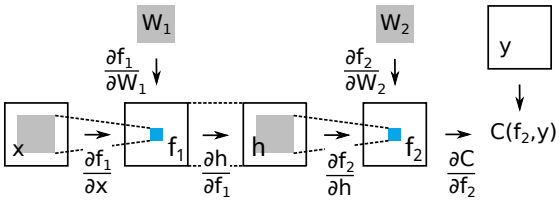


Figure 3.6: A shallow two layer network with the partial differentiations for the backpropagation chainrule.

3.3.3 *Unstable Gradients*

are Problem of unstable gradients made it difficult to train deep architectures based on backpropagation till 2006 [57, 63] when the ReLU was firstly introduced. This is solely related to gradient based methods using backpropagation where the weights are iteratively updated according to the propagated errors. The problems occur when the gradients become too small, and since they are multiplied with gradients of next layer they decrease exponentially. The smaller gradient the smaller update the slower learning. The frontest layers, i. e. layers near to the input, are therefore trained very slowly because of gradually smaller gradients propagated back through the network.

The two main sources of vanishing gradients were described in [57], namely inappropriate activation functions and the improper network initialization. The sigmoid (3.17) was often the most used activation function because of the easy differentiation and the squeeze transformation of input into the $(0, 1)$ interval, which was considered as a suitable property for training the network. Initialization of the network is, therefore, important as the high or low weights multiplied with the input can lead to so-called saturated neuron, i. e. the output is too close to 0 or 1 which causes the almost zero gradients in a case of the sigmoid activation function.

continually in combination with improper network initialization.

The problem of unstable gradients is addressed by several activation functions including ReLU, PReLU, ELU and batch normalization which keeps the forwarded data normalized. The precise impact of both, activation functions and the forwarded data normalization are yet under research.

3.3.4 *Weight Update*

During NN training, the $C(\cdot)$ loss function is differentiated and the error back propagated using the backpropagation. The network weights can be updated based on the gradients reflecting the amount of error they cause. The amount of their change, i. e. the size of a step towards the minima, is controlled by a learning rate γ. The weight updates can be computed with different approaches. The majority of the update strategies aim to speed up the convergence process and reach a better semi-optimal minimum. The following text introduces several widely used weights update approaches.

GRADIENT DESCENT Gradient descent is based on moving a small distance controlled by learning rate γ in the direction of the negative gradient, i. e. towards the minimum of $C(\cdot)$ loss function with respect to weights. The update can be formulated as

$$\begin{aligned} \triangle \mathcal{W}^{\tau+1} &= \mathcal{W}^{\tau} - \gamma \nabla C\left(F_L\left(\mathcal{W}^{\tau}, x\right), t\right) \\ \mathcal{W}^{\tau+1} &= \mathcal{W}^{\tau} + \triangle \mathcal{W}^{\tau+1}, \end{aligned} \tag{3.13}$$

where $\mathcal{W}^{\tau}$ represent the actual weights of the network F_L which consist of L layers. The loss is computed every iteration from all training data x, t to obtain $\mathcal{W}^{\tau+1}$. Replacing the scalar learning rate γ with the inverse of the loss Hessian matrix [52] which is gradient proportional can speed up the optimization from linear to quadratic convergence due to the per weight w_i controlled update. The basic gradient descent is seldom applied as the computation of the related gradients can become intractable on large datasets.

STOCHASTIC GRADIENT DESCENT Replacing the batch gradient descent computation with the mini-batch, i. e. using the Stochastic Gradient Descent (SGD) as in [1] allows to learn on-line based on large data. The $C_n(\cdot)$ loss function is computed from stochastically selected independent train data which comprise the mini-batch of size n. Loss function based on the maximum-likelihood of sampled data reflects the batch loss function $C(\cdot) = \sum_n^N C_n(\cdot)$. The form of stochastic gradient (3.13) with $C(\cdot)$ replaced by $C_n(\cdot)$ defines the SGD. For instance, in Krizhevsky et al. [2], the mini-batch consisted of 128 samples.

SGD WITH MOMENTUM SGD does not behave optimally in case of variously deformed shape representing the loss function C_n [64] which can yield to the refracted trajectory or oscillation during approaching the minimum. Momentum SGD, in contrary, computes the update of $\boldsymbol{W}$ based on the previous momentum μ altered by actual gradients in the form

$$\begin{aligned} \triangle \mathcal{W}^{\tau+1} &= \mu \triangle \mathcal{W}^{\tau} - \gamma \nabla C\left(F_L\left(\mathcal{W}^{\tau}, x\right), t\right) \\ \mathcal{W}^{\tau+1} &= \mathcal{W}^{\tau} + \triangle \mathcal{W}^{\tau+1}. \end{aligned} \tag{3.14}$$

In SGD with momentum update, the μ coefficient damps the kinetic energy to prevent the infinite move. The update step using a momentum provides better convergence in the sense of speed and better minimum [65, 66] compared to an ordinary SGD. The majority of experiments presented in this thesis are utilizing the SGD with momentum.

SGD WITH NESTEROV MOMENTUM A possible enhancement of the momentum SGD was shown in [67, 68] based on the Nesterov adaptive momentum. First, the update of previous step is added to actual weights according which the actual loss is differentiated, and based on such prediction a new update is estimated

$$\begin{aligned} \triangle \mathcal{W}^{\tau+1} &= \mu \triangle \mathcal{W}^{\tau} - \gamma \nabla C\left(F_L\left(\mathcal{W}^{\tau} + \mu \triangle \mathcal{W}^{\tau}, x\right), t\right) \\ \mathcal{W}^{\tau+1} &= \mathcal{W}^{\tau} + \triangle \mathcal{W}^{\tau+1}. \end{aligned} \tag{3.15}$$

ADAM Recently published adaptive moment estimation Adam for stochastic gradient optimization [69] provides an efficient and fast convergence. Adam is referred to be well suited for non-stationary loss and noisy and sparse gradients. According to empirical results based on the NN evaluation on CIFAR dataset and assessment of CNN on MNIST dataset images [69], Adam achieved better model compared to SGD with Nesterov momentum. The learning rate adaptation is controlled by the second order moment of the loss gradient. According to the authors, the improvement was marginal. However, its benefit comes from adapting the learning rate based on particular CNN layer. The update computation is written

$$\begin{aligned} \nabla C &= \nabla C\left(F_L\left(\mathcal{W}^\tau, x\right), t\right) \\ m^{\tau+1} &= \beta_1\, m^\tau + (1-\beta_1)\nabla C \\ v^{\tau+1} &= \beta_2\, v^\tau + (1-\beta_2)\nabla C^2 \\ \triangle \mathcal{W}^{\tau+1} &= -\frac{\gamma}{\sqrt{v^{\tau+1}}+\epsilon} m^{\tau+1} \\ \mathcal{W}^{\tau+1} &= \mathcal{W}^\tau + \triangle \mathcal{W}^{\tau+1}\,, \end{aligned} \tag{3.16}$$

where Adam update is controlled by four hyper-parameters β_1 and β_2 behaving similarly like momentum, the ϵ preventing zero in the denominator, and the learning rate γ.

There are often used other SGD based methods, namely the adaptive subgradient methods for online learning and stochastic optimization AdaGrad [70], an adaptive learning rate method AdaDelta [71], unpublished RMSprop[6], and other more or less rare algorithms. Besides the first order methods, second order, Newton methods like Limited Memory Broyden–Fletcher–Goldfarb–Shannon (LM-BFGS) [72] are rarely used as well.

3.4 ACTIVATION FUNCTIONS

According to a description of network training, the following text reflects the activation functions which besides being the source of nonlinearity in the network affect the gradient and feed forward data scale. Therefore they are closely related to training and in fact recently allowed to train very deep networks. Earlier, squelch activation functions based on sigmoid were usually used including the *sigmoid* itself and a hyperbolic tangent *tanh*. Recently, the *ReLU* based functions are almost solely used in the CNNs models. The importance of these **LU* functions is strongly related to a problem of vanishing gradients and saturated neurons.

SIGMOID Formerly often used a non-linear activation function in NN was the sigmoid activation function Figure 3.7a. Its definition is written

$$h(x) = \frac{1}{1+e^{-x}}\,, \tag{3.17}$$

where the output $h(x)$ is always in the interval $(0,1)$. Sigmoid was earlier used quite often because of its suitable differentiation. Based on the work of Glorot

[6] https://www.coursera.org/learn/neural-networks

and Bengio [57], several sigmoid drawbacks were identified. First, the problem with vanishing gradients can occur as the input value saturates the neuron, the outcome is close to 0 or 1. Second, the output is constantly rising propagating through the network as the function is not centered around 0 which slows down the convergence [57].

The not zero centered problem of the sigmoid function addresses the hyperbolic tangent on Figure 3.7b precisely defined by LeCun et al. [58]

$$h(x) = 1.7159 \tanh\left(\frac{2}{3}\, x\right), \tag{3.18}$$

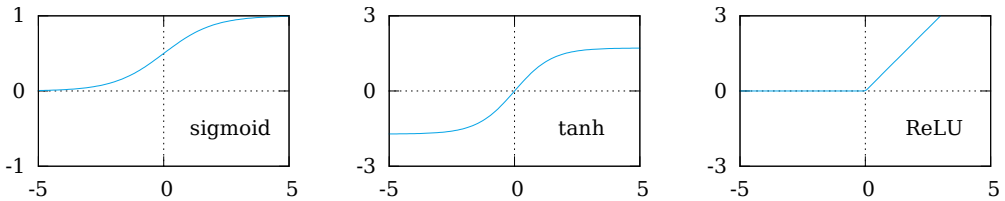
where the output values are in the interval $\approx (-1.7, 1.7)$. However, the problem of vanishing gradients during training may remain for deep architectures due to its tiny gradients in the case of higher input values.

RECTIFIED LINEAR UNIT The sigmoid based activation functions became recently often replaced by the ReLU activation function Figure 3.7c. ReLU were earlier suggested based on the study of cortical neurons and later used in recurrent networks [42]. However, it became prioritized mainly because of significant results provided by *AlexNet* by Krizhevsky et al. [2] where such activation functions were used. The ReLU support the model sparsity which is a suitable property of deep networks where a reasonable amount of sparsity is related to a model generalization capabilities. The ReLU which is a biologically plausible model is defined

$$h(x) = \max(0, x) \ . \tag{3.19}$$

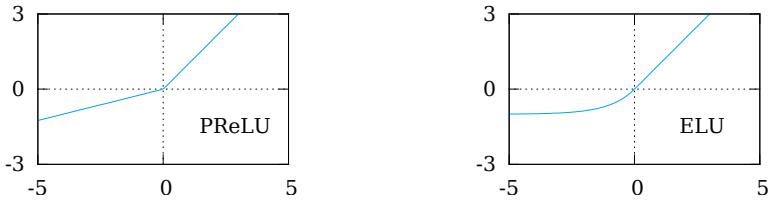
The network with ReLU activation functions can be understood as a sparse representation of an exponential number of linear models that share parameters. The nonlinearity is given by different feed-forwarded data through network paths [73].

A disadvantage of ReLU results from its linear characteristics of the input being above the threshold. Firstly, in a case of neuron saturation using ReLU activation function the non-linear property can be shadowed and therefore the neuron can become overly linear. Secondly, high input value can produce during training a large error which is used to update corresponding weights. Such a large error can lead to a dead neuron in such a way that the neuron will not activate based on other input data. Nevertheless, the dead neuron can be re-activated again in case the sufficiently large value appears which may cause the neuron activation again.



(a) (b) (c)

Figure 3.7: The activation functions including the sigmoid (a), note different vertical scale, tanh (b) suggested by LeCun et al. [58], and recently most often used ReLU (c).



(a) (b)

Figure 3.8: The Parametrized Rectified Linear Unit (a) with the trainable parameter α defining the negative slope. The continuous Exponential Linear Unit (b).

In contrary to the assumption of favorable sparsity supported by true zero activation, He et al. [59] introduced the Parametrized Rectified Linear Unit (PReLU) activation function Figure 3.8a which avoids the true zero. PReLU was introduced in a work presenting a CNN model surpassing the human-level performance in ImageNet classification achieving 4.94 % top-5 error. PReLU is written

$$h(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}, \tag{3.20}$$

where α is a trainable parameter usually initialized to 0.25 [59]. An inconvenient property of ReLU and PReLU is their discontinuity in 0 and that they are not zero centered. The undefined differentiation in this point has to be defined ad-hoc. The zero centered property is desired because usually a faster convergence is achieved in case the values average is close to zero [58]. Recent work of Clevert et al. [74] presents the Exponential Linear Unit (ELU) activation function Figure 3.8b addressing both, the discontinuity and non zero center of ReLU and PReLU. ELU is defined

$$h(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha\left(e^x - 1\right) & \text{if } x < 0 \end{cases}, \tag{3.21}$$

where α is trainable parameter usually initialized by 1 [74]. Results, published in [74], shows beneficial properties of ELU together with faster convergence. On the other hand, ELU compared with the ReLU network is about 5 % slower. Nevertheless, the whole subject of activation functions is still under research.

3.4.1 *Batch Normalization*

An approach of designing the activation functions to produce suitable outputs keeping the distribution centered around zero in some reasonable variance interval can be looked into from the other point of view, the data itself. That is the underlying idea how to deal with an *internal covariate shift* which is understood as the changes in the output data distribution of network layer during training. Such a shift leads to several problems requiring careful initialization settings, per layer

learning rate tuning etc. Ioffe and Szegedy [75] addressed the problem directly by including the normalization into the NN model itself. *Batch normalization* also keeps the feed forwarded data in an almost constant scale which in the case of ReLU prevents the dead neurons caused by an high error.

The learning speed improvement based on the input data normalization, more precisely whitening, i. e. centered around 0 with covariance 1 and decorelated was mentioned already in [58]. Direct whitening of layers activations, activation function inputs, brings complications during training when the gradient descent do not take the normalization into account. That was empirically proofed and theoretically outlined in [75].

The normalization is based on mini-batch statistics defined as

$$\hat{x_i} = \frac{x_i - E\left[x_i\right]}{\sqrt{\text{Var}\left[x_i\right]}}\,, \tag{3.22}$$

where the mean $\mathbb{E}\left[x_i\right]$ and variance are computed from the training data x_i.

Such a transformation do not allow the network shift or scale the data if needed. It also can suppress the non-linearity in the case of sigmoid activation function (see the function shape around 0). That is why the batch normalization is defined

$$\text{BN}_{\gamma,\beta}\left(x_i\right) = \gamma\hat{x_i} + \beta\,, \tag{3.23}$$

where γ and β are trainable parameters allowing the network to shift and scale the normalized activation if needed during training. Using batch normalization allows to rapidly speed up training as the learning rate can be increased. An ensemble of *GoogLeNet* models based on batch normalization reached the 4.9% top-5 error on ILSVRC

3.5 SUMMARY ON CNN

The Artificial Neural Network was predicted to become the important data-driven approach in several tasks. However, the expectation was often too high and together with a too optimistic time prediction of the deployment, the NNs were several times abandoned. At the beginning of the nineties, the important work of LeCun and Hinton on CNN restored the community interest. That is the time the basis of CNN was given and where ends the part of NN history introduction.

The NN formulation was given with an intention to move to the CNN. Given the definition of CNN, two important approaches were described. These include the hand-written digit recognition and the Krizhevsky network used in the ImageNet Large Scale Visual Recognition Challenge. Followed convolution and its implementation in the form of matrix multiplication together with the transposed convolution usually called the deconvolution in the CNN community. The closely CNN related type of network, the FCN was introduced which omits the fixed fully connected layers using the fully convolutional architecture. The last part consisted of training which consists of initialization, weight update, different activation function description and the related problem of unstable gradients.

4

IMAGE PROCESSING BASED ON NEURAL NETWORKS

Image processing includes the wide domain of various low-level image tasks. The recent restoration approaches for several corruption types based on NN follows. Specifically, the restoration methods for natural or text blurred images are presented together with work of Hradis et al. [35] which is directly extended in this thesis. The denoising methods build on the Auto-Encoder (AE), NN, and CNN networks follow with a description of AE and its modification Denoising Auto-Encoder (DAE) used in other low-level image processing tasks. Recently, the denoising is often included directly in the CNN models deployed for several tasks which are often noise affected. A pixel classification for object segmentation, which preceded the FCN and which comprises the basis for following introduced architectures, are described. The super-resolution approaches are presented with an introduction of Super-Resolution CNN (SR-CNN) which is closely related to the JPEG artifacts reduction method described by the end of this chapter and which the proposed models later in this work are compared with.

4.1 DEBLURRING

The overview of deblurring methods have one central common part the NN. First, the specific methods for text deblurring are introduced. These are end-to-end based approaches directly mapping the blurred image to its estimated sharp representation. In contrary, the majority of referred methods for natural image deblurring are used to estimate the inverse Point Spread Function (PSF) with which the image is later on deconvolved.

4.1.1 *Text Image Deblurring*

A SHALLOW NN BASED TEXT DEBLURRING Tansley et al. [76] already in 1996 published a short paper about NN based image deconvolution used for text deblurring. The text images significantly differ from the natural images. That, generally, yields to a task with a strong prior based on the font type, clear distinction between the foreground and background, and others restrictions of to text images compared to natural ones. A utilization of NN proceeded from the idea to train the model to reflect the prior itself i. e. the data-driven model. An optimal linear filter based on the assumption of convolution linearity was modeled by the NN with a single linear neuron (3.1) followed by the sigmoid function. Such a simple and shallow network was used to classify the pixels to be part of the background or foreground based on the thresholded sigmoid function. The NN was trained on patches of size 13×13 which also defined the NN perception field. A single value

output, the foreground or background classification, corresponded to the center of the input patch. The trained network was slid over the whole blurred image to obtain the deblurred binary text image.

That represented the end-to-end mapping, where the network learned the PSFs and noise allowing to deblur the input. Despite the fact, the network consisted of single neuron only, i. e. 170 parameters, the results of this data-driven approach outperformed the optimal linear filter and simultaneously incorporate the noise information.

The introduced end-to-end mapping of the degraded image to its clean and sharp representation is a very simple predecessor of the method presented by Hradis et al. [35] and later reapplied in this thesis which is based on much larger model in regard to width and depth incorporating many various convolutional kernels stacked into several layers and interposed by ReLU based non-linear activation functions.

A DEEP CNN BASED TEXT DEBLURRING An image deblurring based on CNN presented in [35] was motivated by processing the handheld or phone camera taken text images including several notes or public information boards. The best performing model consisted of 15 layers interleaved by non-linear ReLU and provided a blind-deblurring approach for images blurred with single or multiple PSF functions as well. The architecture of the deep model was inspired by the vision-based approaches introduced by Krizhevsky et al. [2].

Compared to the shallow NN based text deblurring model [76] of 170 parameters, the largest 15 layer presented model with a global perception field 50×50 consisted of 2.3×10^6 trainable parameters. The L15 network was a regression model which produced the normalized pixel values. This end-to-end model was directly trained on a pair of noisy blur and clear patches where the minimum size of input was due to the large global perception bigger than 50 pixels to provide the about the same amount smaller output. The approach of training followed the trends of earlier introduced approaches for tasks of computer vision [2, 45] i. e. the backpropagation followed by the SGD weight update.

The experiment results based on various deep models yielded to affirm the assumption that the performance of CNN is beside other aspects strongly related to the model depth The final model, 15-layer end-to-end mapping network, achieved the state-of-the-art results and went beyond compared to the non-blind L0-regularized method [5]. There were two different metrics to evaluate the model based on the Peak Signal to Noise Ratio (PSNR) and the Optical Character Recognition (OCR) accuracy recognition as well. This approach, broadly speaking, became the subject of exploitation for the task of motion deblurring presented in this thesis related to the surveillance images of cars with the motion blurred license plates.

4.1.2 *Natural Image Deblurring*

Natural images, compared to the text images, represent much wider domain and complex problem for the NN based methods. Therefore, the end-to-end mapping approaches occur less often compared to several methods estimating the inverse PSF. The exceptions, i. e. end-to-end mapping approaches, are represented by the work of Schuler et al. [77] and Xu et al. [78]. The almost traditional approach of inverse PSF estimation represented by several different methods utilizing the NN follow.

NN BASED RESTORATION OF DECONVOLVED IMAGE Schuler et al. [77] presented an end-to-end mapping approach of blurred and consequently deconvolved image. The method relied on 2-step procedure which consisted of a direct deconvolution step and mapping of a deconvolved yet corrupted image to its clean representation. The deconvolution was based on the regularized inversion of a blur PSF in Fourier domain obtaining the directly deconvolved image y', i. e. a uniform PSF was expected. The NN model consisted of 4 layers of in total 1.6×10^7 trainable parameters and was trained using the backpropagation and SGD optimizing the ℓ^2-norm loss function. Training data were artificially blurred with fixed amount of noise and consisted of pairs (y', x), where x represents the ground truth clean image and y' the directly deconvolved blurred image y obtained applying the blur PSF g

$$\begin{aligned} y &= x * g \\ y' &= \mathcal{F}^{-1}\left(R\,\mathcal{F}\left(y\right)\right) , \end{aligned} \tag{4.1}$$

where R is the regularized deblurring operator and $\mathcal{F}\left(\cdot\right)$ is the Fourier transform. The final NN model was compared based on deblurring the naturally and the artificially blurred data with the state-of-the-art method represented by BM3D. The NN model delivered slightly better results.

Beside the deblurring, the same model was trained to restore the images corrupted by the Poisson noise with the similarly good results which showed the benefit of this machine learned model as the semi-universal solution. Interestingly, the same model trained for the complete end-to-end mapping, i. e. deblurring returned worse results compared to restoring the already deconvolved images. From the today's point of view, it is interesting that the one model trained for various amount of blur and noise significantly worsen the results which yield to the necessity of having a trained model for a specific amount of blur, i. e. such an approach is close to the non-blind deconvolution.

SEPARABLE DECONVOLUTION BASED ON CNN The end-to-end mapping model for image deblurring was presented by Xu et al. [78]. The proposed approach, in contrary to the NN model of [77], was based on a CNN model encapsulating two main steps of the deblurring pipeline. The deep CNN for non-blind image deblurring was based a on a deconvolution and a denoising part Figure 4.1.

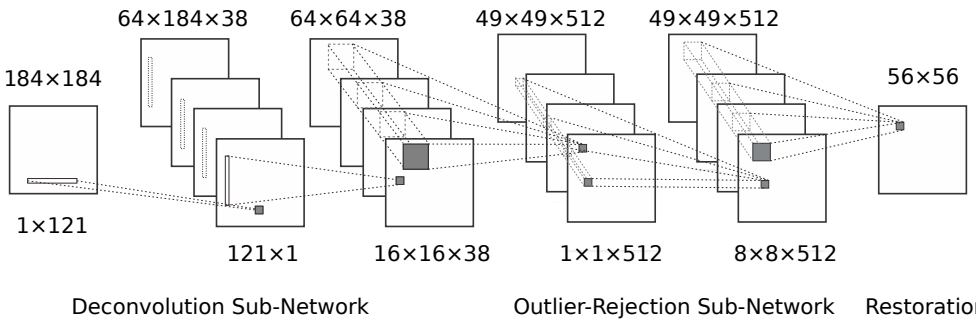


Figure 4.1: The architecture of deep deconvolution method [78].

The part for deblurring was proposed according to conventional engineered deconvolution method. The convolution separability theorem and the analysis of pseudo inverse PSF obtained by the Wiener filter yielded to an architecture where the first layer consisted of 38 1×121 filters and the second layer of 38 121×1 filters. The filter size was empirically chosen based on the plausible deconvolution results achieved using the Wiener filter. The Wiener based PSF estimation filter was also used to initialize the first two layers of the CNN model which, compared to random initialization, yielded to better results. The second part of the final model consisted of a denoising network based on the network presented by Eigen et al. [79] which consisted of 4 layers. The denoising subnetwork was used for several types of noise including the JPEG artifacts, clipped saturated values, etc.

The single sub-networks were trained separately and later on fine-tuned together using the end-to-end learning. Such a model which consisted of two pretrained parts was compared with engineered methods including the outliers handling in non-blind deconvolution [17], and the learning based method NN Based Restoration of Deconvolved Image [77]. The final model outperformed both the engineered state-of-the-art methods and the NN based image restoration method. The model was trained for a specific blur represented by a uniform PSF. The direct end-to-end mapping is recently not so common as the approaches estimating the inverse PSF which are later used for deconvolution. All the remaining approaches are more or less PSF based and provide various levels of performance compared to state-of-the-art considered methods.

NON-UNIFORM PSF ESTIMATION The non-uniform deblurring CNN approach which estimates the probability distribution of fixed PSF per patch over the whole blurred image was presented by Sun et al. [80]. The set of possible PSF lengths consisted of 13 samples in the range from 1 to 25 with the interval of 2 where the length 1 was considered as identity i. e. a zero move. The directions were represented by 6 samples from 0 to 150 with the step of 30. There were 78 combination in total but only 73 preserved because of 6 of length 1, the identity, were squeezed into one. The training set consisted of pairs of blurred image patches and corresponding PSFs. The final 6 layer CNN included approximately 6.5×10^5 of trainable parameters Figure 4.2.

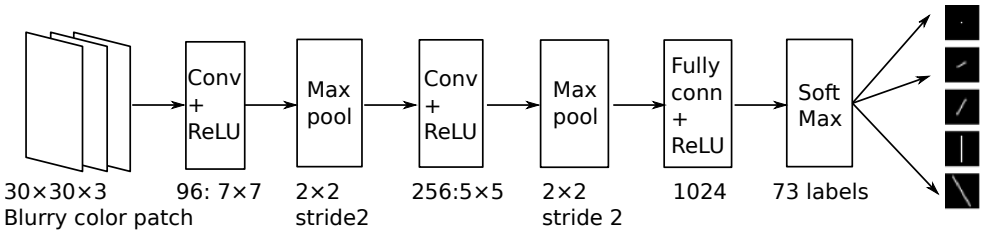


Figure 4.2: The presented architecture for non-uniform PSF estimation [80].

The set of possibly predicted PSFs was extended up to 361 samples. That was achieved by feeding the original patch and its 5 rotated versions with the step 6 in the range 0 – 24 into the CNN to estimate the probability of PSF not included in the training set.

The input image, divided into overlapping patches, was processed by the CNN model to estimate the probabilities of 361 PSFs representing the different motion blurs. A final dense motion blur field of the input image was computed using Markov random field to enforce the motion smoothness. With the dense non-uniform motion PSFs estimated by CNN, the blurry image was deconvolved to estimate the sharp image.

Experiments were evaluated using both, artificial and naturally blurred images. The qualitative comparison was based on computing the MSE of CNN and several state-of-the-art considered methods estimating PSF. Based on the synthetic motion blurred images, the CNN based PSF estimation approach achieved the beyond state-of-the-art results.

ITERATIVE PSF ESTIMATION IN FOURIER DOMAIN Schuler et al. [81] published their end-to-end trained NN natural image deconvolution method based on stacked network architectures providing the non-uniform blind deblurring. The base model consisted of 3 parts, feature extraction, PSF estimation, and latent sharp image estimation. Furthermore, a final deep architecture consisted of these 3 stacked base models.

The first part of a base model, feature extraction, consisted of learned filters producing the gradient like image representations. The learned filters extracted the features found in the blurred or sharp images. These convolutional layers were interlaced with non-linear tanh activation functions. The first trainable part is shown in Figure 4.3 The second part, the PSF estimation was computed in the Fourier domain from sharp and blurry features of the input image. The estimated PSF was later on used in the third part, image estimation. The training took mostly on the feature extraction represented by the convolutional layers. The second and third part depended solely on hyper-parameters for regularization and were rather fixed.

The proposed architecture of base models had a multi-scale support of PSF sizes from 17×17 up to 33×33. The deep architecture was gradually trained. First, one base model including the feature extraction and PSF estimation parts was trained. The training was based on minimizing the ℓ^2-norm between the ground truth blur

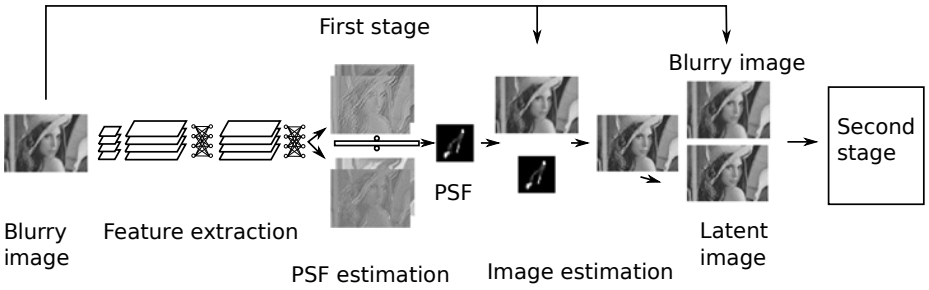


Figure 4.3: Iterative PSF estimation in Fourier domain [81].

PSF and the estimated PSF from an input blur image. Later on, the architecture was trained end-to-end using backpropagation and the AdaDelta weight update based on the ℓ^2-norm of estimated sharp image and its corresponding ground truth.

The experiments were evaluated considering several tasks including noised blur image, a model for specific image content, spatially varying blur in the image, and several scales of the blur, i. e. a non-uniform PSF. The network was able to adapt and learn the filters which were able to counter the added Gaussian noise. The specific model trained on specific image content provided improved performance compared to the general model for the trained type of image content. This method bridges the real end-to-end mapping approaches with the more classical PSF estimation oriented methods. The model structure, which includes the pure machine learning based part with the semi-engineered Fourier domain related part, comprised a quite complicated combination of an engineered with learning methods.

PSF ESTIMATION IN FOURIER DOMAIN An approach focused on Fourier domain for motion deblurring was introduced by Chakrabarti [82]. Their model consisted of two parts, the rough estimation based on NN similar to CNN and the second part utilizing the optimization to obtain the refined final restored image. The method was designed to restore the uniformly motion blurred image blindly.

The particular NN estimated the complex Fourier coefficients of the inverse blur PSF. The input was a 65×65 patch which, based on the NN architecture, was decomposed into 4 different frequency bands to provide a multi-resolution frequency decomposition. The output PSF in Fourier domain was of size 33×33. The NN architecture used the weight-sharing approach related to filters of CNN, with the difference that in spatial domain expected space locality was replaced by in Fourier domain frequency locality Figure 4.4. Training was based on SGD using the objective function ℓ^2-norm between estimated and ground truth sharp image patch. The final model consisted of approximately 4.5×10^7 parameters.

The image reconstruction consisted of two steps. First, the overlapping input image patches were deconvolved with NN estimated PSF and combined using the Hanning window approach of weighted average into the rough deblurred image estimation. Second, the whole roughly restored image was used to estimate the global motion blur PSF and finally deconvolved. The last PSF was a fixed size support of 51×51 allowing to compute blind deconvolution.

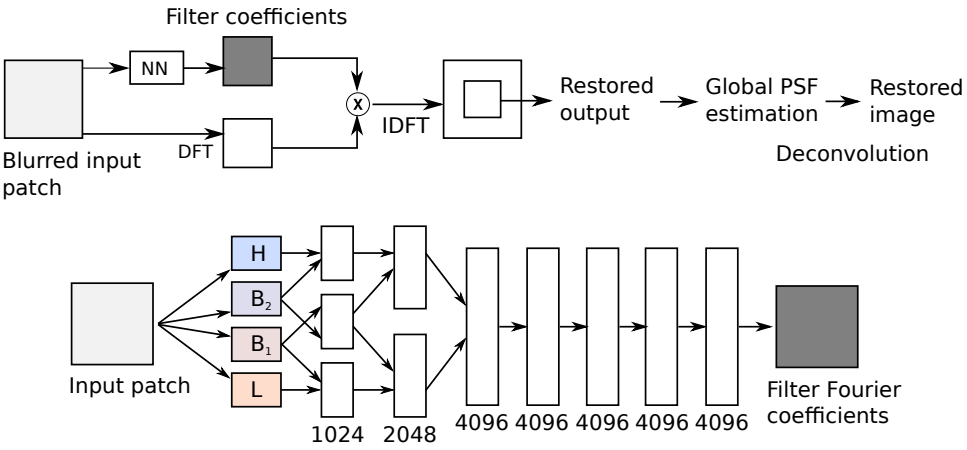


Figure 4.4: The processing pipeline utilizing the NN to estimate the Fourier PSF coefficients [82]. The image is per patch restored. Finally, the whole such a restored image is used to estimate the global PSF. That is used to deconvolve the image to obtain the final restored image. Note the shared filters between low and middle frequencies utilizing the frequency locality in Fourier domain.

The achieved results are comparable with the state-of-the-art methods yet did not significantly outperform them. Nevertheless, the benefit of the NN approach is, compared to engineered methods, in a relatively fast implementation based on parallelization using the GPU.

MULTI-FRAME PSF ESTIMATION IN FOURIER DOMAIN Wieschollek et al. [83] presented the multi-frame based deblurring method tightly related to the Chakrabarti [82] approach of estimation the inverse PSF in Fourier domain. Authors combined the deblurring method based on the inverse PSF deconvolution with the method of Fourier Burst Accumulation (FBA) which combines several images of same content into one sharp image. FBU was trained to get the data-dependent weighted average.

The whole architecture was divided into 3 parts. First, the slightly modified NN of [82] was used to estimate the inverse PSF of every frame from the input group of patches. Second, the deconvolution of the input patches with the estimated PSF led to roughly estimated sharp patches represented in the frequency domain. Finally, third part consisted of the trainable FBA which combined the estimated sharp patches into one sharp image.

This approach, based on the multi-frame images, assumed the static scene only. The results, according to authors, were comparable with the state-of-the-art methods mainly represented by the baseline composed from the PSF estimating NN [82] and the fixed not learned FBA. However, no qualitative comparison was provided.

4.1.3 *Summary on Deblurring*

Interestingly, the majority of he presented general data-driven approaches for deblurring based on NN did not significantly outperform the engineered methods. The exception represents the Separable Deconvolution Based on CNN which,

based on the combining of deblurring and denoising networks achieved the beyond state-of-the-art results on natural images. The benefit of recently published methods often lay in the parallel processing allowing to exploit the GPU and reduce the restoration time. An exceptional performance provides the text image focused end-to-end mapping approach of text deblurring introduced by Hradis et al. [35] which significantly outperformed the state-of-the-art methods.

4.2 DENOISING

The task of denoising based on NN is in the recent literature solely focused on restoring several types of noise including the additive Gaussian noise, salt-and-pepper noise and usually some kind of structural noise like stripe corruption. Two slightly different approaches emerged, generally, the NN based methods which consist of the fully connected or convolutional networks, and the Auto-Encoder (AE) based approaches earlier primarily used for deep architectures initialization. Recently, in contrary to the deblurring approaches, the strategy of training the denoising networks is to learn the end-to-end mapping of noised image to its uncorrupted representation. The model specifications including mostly the model size are emphasized to track the change of model size in recent history.

4.2.1 *Image Denoising Based on NN*

Recently, the models trained for other low-level image processing related tasks including deblurring reckon the already noise-corrupted input data. Slightly older approaches presented by Jain and Seung [84] and Burger et al. [85] are described to emphasize the gradually simplified procedure of training clearly visible in the denoising NN.

GRADUALLY TRAINED CNN FOR NATURAL IMAGE DENOISING A CNN used for low-level image processing was evaluated by Jain and Seung [84] in the task of recovery the underlying image from an observation that has been subject to Gaussian noise, i. e. directly written, the noised image restoration. A CNN model was evaluated for both blind and non-blind denoising.

The incremental per layer training was based on the SGD using the backpropagation optimizing the ℓ^2-norm i. e. the loss function defined between the clean and noised image. First, one hidden layer was trained. After 30 epochs, where an epoch is a unit of a measure which indicates that all the training data were provided to actually trained model, weights from the first hidden layer were copied into the second layer. Such a step was repeated till the fifth CNN layer. The network was solely trained on artificially noised natural images.

The CNN model was compared with the state-of-the-art discriminative Markov random field based method, Bayes least squares-Gaussian scale mixture (BLS-GSM). The CNN model of 5 layers in total gave slightly better results compared to BLS-GSM. A single CNN model was able to cover various Gaussian noise pa-

rameters providing almost identical results compared to CNN trained for a single particular level of Gaussian noise. Such an approach led to a model which was able to provide a blind restoration of an image with unknown noise parameters.

The CNN model for blind deconvolution consisted of 5 layers where 4 hidden layers had 24 filters each of size 5×5. Interestingly, the filters in hidden layers convolved over 8 randomly chosen activation maps of the previous layer which from today's point of view reminds the channel grouping. The whole model consisted of almost 1.6×10^4 parameters.

GAUSSIAN, JPEG, AND STRUCTURED NOISE RESTORATION BASED ON NN An image denoising approach based on NN was proposed by Burger et al. [85]. The goal was to train the NN to directly map the additive Gaussian noised input to its clean output representation. Several other types of noise were studied including the JPEG compression artifacts, salt-and-pepper noise, and structured stripe noise. The NN model which achieved the best results was based on a fully connected network which consisted of 5 layers where the input had size 17×17 and the 4 hidden layers were connected through 2 047 weights each i. e. the model consisted of approximately 9×10^3 trainable parameters using the tanh activation function. The evaluation consisted of denoising only two images which were artificially corrupted by Gaussian noise with a fixed parameter. The model achieved state-of-the-art results but did not significantly outperform them.

Experiments with several Gaussian noise levels showed, similarly to [84], that the NN model had a sufficient capacity to provide next to the non-blind also the blind restoration. Also, in compliance of big data assumption, the increase of training dataset led to better results. The evaluation of other noise types proved the model capabilities for the universal denoising. However, the results did not outperform the state-of-the-art method represented by sparse 3D transform-domain collaborative filtering (BM3D). In the case of JPEG compression artifacts, salt-and-pepper, and stripe based noise, the model outperformed the compared methods including the BM3D and SPP filters often used in FFmpeg [23].

4.2.2 *Image Denoising Based on Auto-Encoder*

The Auto-Encoder (AE) based methods for denoising yet in the recent past comprised the major type of the machine learned approaches. A significant contribution of AE first took a form of the initialization step in deep architectures where the gradually pretrained AE provided a way to train or fine-tune several deep architectures [32]. The denoising itself based on the AE was something like a side product of learning the discriminative features used in computer vision. The concept of AE and description of several methods follows.

DENOISING AUTO-ENCODER FOR FEATURE LEARNING An Auto-Encoder (AE) is a type on NN which was primarily used to train the network providing efficient task related coding. An AE denoising model was published by Vincent

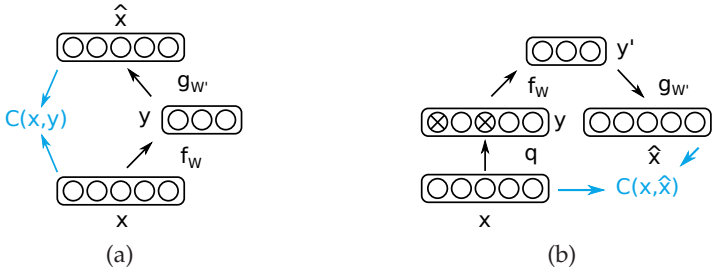


Figure 4.5: An Auto-Encoder (a) and Denoising Auto-Encoder (b) .

et al. [55]. The task of denoising was used as a criterion for an unsupervised AE training with the goal to obtain the useful higher level representation of an input data i. e. the discriminative features.

An auto-encoder consists of encoder $f_W(\cdot)$ mapping data from the input space $\mathcal{P}$ to the encoder space $\mathcal{Q}$ and decoder $g_{W'}(\cdot)$ decoding back from the encoder space to the input data space $\mathcal{P}$ as shown in Figure 4.5a. Both, encoder and decoder are often defined in the form of an ordinary neuron (3.1). Such an auto-encoder is written

$$\mathrm{AE}(x) = g_{W'}(f_W(x)) \ , \tag{4.2}$$

where $\mathrm{AE}(\cdot)$ represents the auto-encoder and W' and W are the trainable parameters of decoder and encoder, and x is the input image. In case of stacked AE the encoder and decoder are defined

$$\begin{aligned} f_W(x) &= \left(f_{W_{L-1}} \circ \ldots \circ f_{W_1} \circ f_{W_0}\right)(x) \\ g_{W'}(y) &= \left(g_{W'_0} \circ g_{W'_1} \circ \ldots \circ g_{W'_{L-1}}\right)(y) \ , \end{aligned} \tag{4.3}$$

where $\circ$ is the composition operator, x is the input image, y is the encoded representation, and L defines number of encoding respectively decoding layers.

Training of AE is based on unsupervised approach, where the model is learning the representation of the input data, usually sparse or in contrary dense, which can be transformed back, decoded to the original input. An AE, based on the linearity constraints, can perform the principal component analysis (PCA) decomposition, in the encoder step, the input image is decomposed based on the eigenvectors and related eigenvalues while the former image representation is composed in the decoder step. AE is therefore trained on input images x based on the criterion of reconstruction the input image x after the feed-forward through the AE.

The modification of the training approach introduced the Denoising Auto-Encoder (DAE) which primarily differs from AE in reconstructing the noised input y. This is the noise $q(\cdot)$ (2.1) corrupted representation of a clean image x which the DAE should decode as shown in Figure 4.5b. Therefore,

$$\begin{aligned} y &= q(x) \\ \mathrm{DAE}(y) &= g_{W'}(f_W(y)) \\ \hat{x} &= \mathrm{DAE}(y) \ , \end{aligned} \tag{4.4}$$

where $\hat{x}$ represents the estimated clean image x from the noise corrupted input image y. Usually, the training is performed based on SGD with a squared ℓ^2-norm loss function (3.5).

Stacked Denoising Auto-Encoder (SDAE) is composed of gradually trained DAEs which, later combined, defines a deep architecture Figure 4.6. The subsequent DAE is trained based on the clean image x processed by the previous DAE encoder $f_{W_{i-1}}$. The gradual training, having the first DAE_0 trained based on the (4.4), is written

$$\begin{aligned} y'_0 &= f_{W_0}(x) \\ y'_1 &= q(y'_0) \\ \text{DAE}_1(y'_1) &= g_{W'_1}(f_{W_1}(y'_1)) \ , \end{aligned} \tag{4.5}$$

where the DAE_1 is trained by optimizing the ℓ^2-norm (3.5) of pair $(y_1, \text{DAE}_1(y_1))$. The clean image x is processed by the DAE_0's encoder part f_{W_0} to get the y'_0 representation which is noised by the $q(\cdot)$ function. The y'_0 is then used as the input into next DAE_1 which is trained the same way the DAE_0 was. From there, the procedure can be repeated according to the required number of layers [55].

Such an approach should force the model to learn more complex mapping than the identity, i. e. one that extracts useful features not only for denoising but also for a classification later used as input of arbitrary classifier. The evaluation of how useful these features were was provided by experiments based on providing learned features to several classifier tasks, MNIST[1] digit classification together with audio genre identification. Three types of noise $q(\cdot)$ were considered, Gaussian noise, salt-and-pepper noise, and so-called masking noise where some fractions of an image were missing.

The results showed that high performance i. e. well discriminative features can be achieved using simple and generic types of noise while the difference between noise used for training was negligible. It was also shown that a deep network pretraining strategy, stacking of DAE, brought in most cases an improvement, based on the features used for classification, compared to ordinary auto-encoders stacking which often leads to just copying the input or similarly uninteresting transformations. Unfortunately, no evaluation of SDAE for image restoration was provided. It was solely used for a deep architecture initialization which consisted of encoder functions f_W trained as SDAE. The biggest model consisted of 4 layers which resulted in approximately 1.2×10^4 trainable parameters.

DENOISING AND IN-PAINTING BASED ON STACKED AUTO-ENCODER An approach of denoising inspired by SDAE introduced in [55] was presented by Xie et al. [86]. Compared to the scheme in (4.5) [55], the SDAE training was modified

[1] Mixed National Institute of Standards and Technology database.

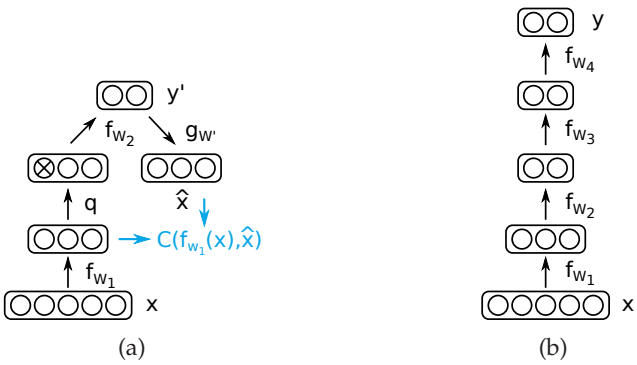


Figure 4.6: A stacked denoising auto-encoder (a) and a deep architecture initialized by a stacked DAE (b).

to reflect the essential denoising task instead of learning the discriminative features for classification. A piece of the model structure for training is written

$$\begin{aligned} y_0 &= q\left(x_0\right) \\ y_1' &= f_{\mathcal{W}_1}\left(y_0\right) \\ \mathrm{DAE}_2\left(y_1'\right) &= g_{\mathcal{W}_2'}\left(f_{\mathcal{W}_2}\left(y_1'\right)\right) \\ \hat{x} &= \mathrm{DAE}_2\left(y_1'\right) \ , \end{aligned} \tag{4.6}$$

where the input data x_0 is corrupted with the noise function $q\left(\cdot\right)$ represented as the noised image y_0. The training of the first SDAE differs from the approach of [55] in gradually using the noise corrupted y_0 transformations instead of the clean image x based on the encoders f_{W_i}. Both approaches are illustrated in FIGURE. The loss function was based on regularized square ℓ^2-norm, and the model was optimized using the LM-BFGS [72] approach, where the regularization induces the sparsity in the hidden layer of actually trained DAE [86].

The architecture based on the SDAE consisted of 2 stacked DAE and is written

$$\hat{x} = \left(g_{\mathcal{W}_0'} \circ g_{\mathcal{W}_1'} \circ f_{\mathcal{W}_1} \circ f_{\mathcal{W}_0}\right)\left(q\left(x\right)\right) \ , \tag{4.7}$$

where $\circ$ is a composition operator of f_{W_i} the encoding functions and $g_{W_i'}$ corresponding decoding functions, and where $q\left(x\right)$ is the noised input image. The SDAE model based on NN consisted of 2 DAE, 4 layers initialized with 2 encoders weights W_i and 2 decoder weights W_i'.

This architecture, called sparse SDAE, was trained, fine-tuned, with standard backpropagation to denoise the Gaussian additive noise and salt-and-pepper noise achieving comparable results with state-of-the-art methods represented by BLS-GSM. With an alternative training scheme proposed the results of stacked denoising autoencoders reached the performance comparable to traditional linear sparse coding algorithm on a simple task of denoising the Gaussian noise. Authors also focused on the blind and non-blind in-painting which differs in the availability of a user region selection of in-painted area, where in both cases model based on sparse SDAE performed well. These results, unfortunately, were not compared with any relevant approach.

SPARSE TRANSFORMATION LEARNED BY AUTO-ENCODER Cho [87] added the sparse transformation $r(\cdot)$ of encoded noisy image y' (4.6), which showed a positive contribution of sparse representation in denoising task and subsequent feature learning. The idea of sparsification is based on the assumption that strongly noised image data $q(x)$ does not belong to the input data space $\mathcal{P}$ and therefore is further expected that the encoded image $y' = f_W(q(x))$ will be outside the encoder space $\mathcal{Q}$ i. e. such a representation need to be projected to the $\mathcal{Q}$ space to be correctly decoded by $g_{W'}(\cdot)$. The projection of y' onto space $\mathcal{Q}$ is proposed as

$$r(y') = \arg\min_{q \in \mathcal{Q}} d(y', q) \ , \tag{4.8}$$

where $d(\cdot, \cdot)$ is a suitable distance metric. The *simple sparsification* [87] $r(\cdot)$ was introduced as a function to decrease each component of y' i. e. to be sparse.

The evaluated architectures included the *simple sparsification* function $r(\cdot)$ in their bottleneck

$$\hat{x} = (g_{W'} \circ r \circ f_W)(q(x)) \ , \tag{4.9}$$

where encoder and decoders consisted of 1, 2, or 4 layers. The $r(\cdot)$ placement influence was not further studied. As in the previous approaches, the final architecture was fine-tuned using the backpropagation. The 4 layer final model consisted of approximately 10^3 trainable parameters where the input patch had size 8×8.

The denoising performance was measured based on the ℓ^2-norm. The comparison of models with and without the *simple sparsification* showed the benefit of sparse representation which was the higher the higher was the amount of noise, specifically Gaussian or salt-and-pepper noise. In contrary, in the case of small amount of noise the results based on the sparse representation were worse. With more hidden layers the benefit of sparse representation was in case [87] smaller.

4.2.3 *Summary on Denoising*

The brief overview of denoising methods based on several types of neural networks showed a good performance yet not significantly outperforming the engineered methods. The presented models were relatively shallow yet in the past perceived as quite deep architectures. Unfortunately, despite the fact that the referred papers are somehow demarcating themselves from the previous works, authors never compared among themselves. From the actual point of view, the approaches based on auto-encoders and stacked denoising autoencoders were solely used for initialization of the architectures which were subsequently fine-tuned using the backpropagation. The question is if such an approach is valuable in recent days of networks with more than 100 layers [47]. The presented models had at most 1.6×10^4 trainable parameters.

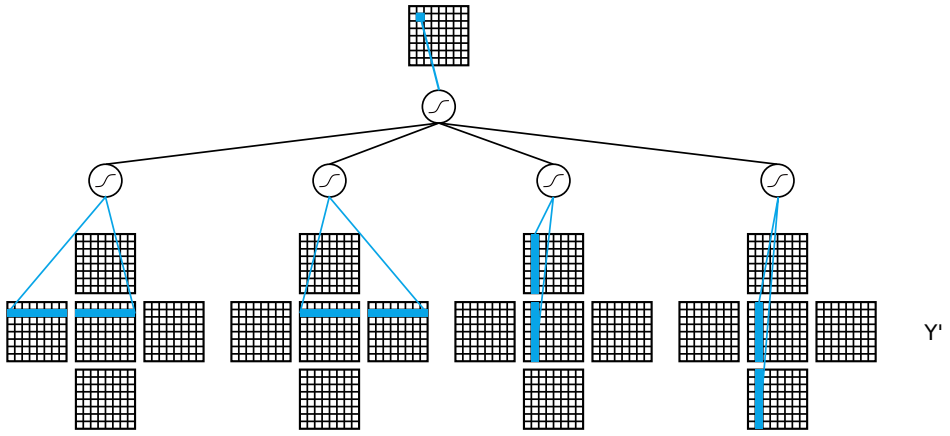


Figure 4.7: JPEG quality transcoder neural network from [88]. There were 64 networks for each JPEG coefficient per channel ($Y'C_BC_R$) with specialized reception field considering the coefficients along the horizontal and vertical directions only. This is the network for a fifth coefficient of luma Y' channel.

4.3 JPEG ARTIFACTS REMOVAL

The degradation based on JPEG compression, namely the blocking and ringing artifacts, are often restored by the engineered methods like SPP [23] or SA-DCT [7]. In contrary, several types of the machine learning based approaches utilizing the NN were introduced including the JPEG quality transcoding and post-processing methods. The methods making the use of information the corruption is partly structurally based yield to flexible NN approaches which provide state-of-the-art or beyond results.

JPEG QUALITY TRANSCODING USING NN A JPEG Quality Transcoder (JQT) based on NN was introduced by Lazzaro and Wawrzynek [88]. A highly compressed JPEG image is transcoded to a larger JPEG image with reduced compression artifacts. That is proceeded without the already non-available source uncompressed image. Such an approach benefits from no intervention into decoding standard which allows utilizing all the existing decoders.

This method transformed the quantized DCT coefficients to provide visually but also quantitatively better-decoded image compared to the originally compressed one based on the estimating the quantization error (2.16). Considering the 64 coefficients in a JPEG block, the 64 different NN architectures Figure 4.7 were proposed to estimate the difference

$$\triangle B_{pq} = \mathcal{B}_{pq}\, Q_{pq} - B_{pq}\,, \tag{4.10}$$

between the dequantized $\mathcal{B}_{pq}\, Q_{pq}$ input coefficient and the original B_{pq} coefficient. The method worked with all three $Y'C_BC_R$ color planes. All 64 different networks which vary in a number of hidden neurons and the structure of perception field were trained base on optimizing the specific objective function related to perceptual quality metric defined by authors using the backpropagation.

All the 64 networks were trained independently. Network input consisted of the horizontal and vertical neighbor coefficients of a particular $\mathcal{B}_{pw}$ coefficient in all 3 color planes $Y'C_BC_R$. Network output comprised 3 coefficients, one of each color plane. A reconstructed pixel was computed in the block the coefficients were transformed from under the assumption that only coefficient $\hat{B}_{pq}$ has been quantized. An error based on the perceptual quality metric was measured and based on it the network weights $\mathcal{W}$ were updated. This procedure, reconstruct, measure, and the update, was repeated for all the 64 pixels in the block. The final model consisted of approximately $64 \times 2 \times 10^2$ trainable parameters, where 64 represents the number of networks.

REDUCING BLOCKING ARTIFACTS BY AN ADAPTIVE NN A NN based algorithm for compression artifacts, particularly block, and ringing, was proposed by Zhang et al. [89]. Compared to the previous approach [88], the method is primarily related to block type instead of pixel information. The blocks the JPEG image consist of were divided into 3 classes, namely the plain, edge, and texture class, where the plain class characterized the smooth block, edge block included high variance, and the texture class was parametrized as something in-between. The classification was proposed based on the assumption that the blocks of the same class have common features compared to other classes. A particular multilayer network was trained for each block type.

The proposed architecture consisted of approximately 1.3×10^4 trainable parameters having 2 layers with the sigmoid-based activation functions. The input was a 5×5 patch, where all values were transformed by subtraction from the central pixel except the central pixel itself. An output value was the transformed pixel. The model was trained using the standard backpropagation and SGD.

For evaluation, two different metrics were used. The traditional PSNR and a subjective error metric which should more accurately reflect the human perception of the restored image. The trained models, per each block class, achieved the comparable state-of-the-art results which were measured only on a few images were the compressed image had approximately 27.8 dB and were restored to achieve 29 dB.

ARTIFACTS REDUCTION BY CNN Recently, a CNN based approach introduced by Dong et al. [37] utilize the end-to-end mapping to suppress the block and ringing artifacts related to JPEG compression. The proposed artifact reduction CNN model, AR-CNN, is based on the model used for super-resolution task SR-CNN. The final AR-CNN consisted of 4 layers interpreted as 4 network parts Figure 4.8. First, the extraction part consisted of layers introduced as feature extractor followed by the second part presented as a feature enhancement. Third part provided the mapping of extracted and enhanced features and the last part, the reconstruction, provided the final restored image.

The suboptimal network initialization forced to train the network gradually, where first the shallow base network was trained and later on the learned weights were transferred to more deep model. The model was initialized based on training

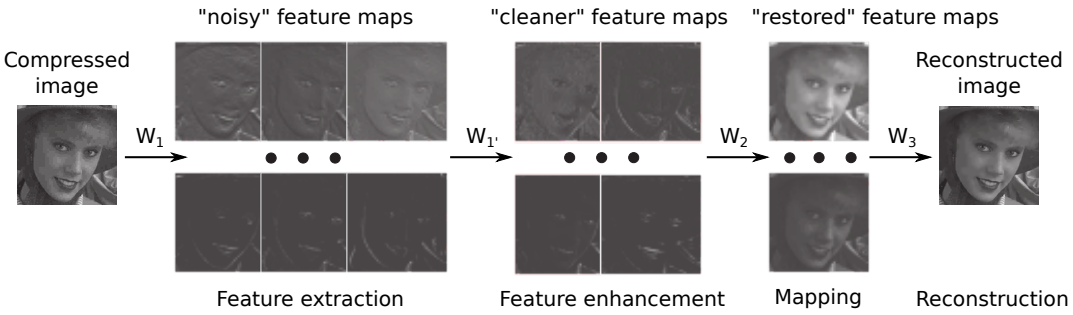


Figure 4.8: ARCNN introduced by [37] which consists of 4 layers end-to-end mapping the JPEG restored image to its restored representation. The intermediate activation maps are shown.

the parameters using the high-quality JPEG compression followed by fine-tuning on low quality. AR-CNN consisted of approximately 10^5 trainable parameters.

The final model was evaluated and compared with baseline SR-CNN model and the state-of-the-art engineered SA-DCT [7] method. The achieved results outperformed the state-of-the-art which was evaluated on several datasets including the Berkeley segmentation dataset [90] with the LIVE1 dataset [22]. The average PSNR of the LIVE1 evaluated dataset was 30 dB improved to 31.29 dB.

4.3.1 *Summary on JPEG Artifacts Removal*

All the presented methods provide a significant performance of JPEG artifacts image restoration. Unfortunately, the metric like PSNR on which these methods could be compared are not always appropriate because it does not have to correlate with the perceived image aesthetic. Therefore, an interesting result can occur like worse PSNR of one restoration method compared to another restoration method while the perceived quality can be counter the PSNR based one. The presented methods are of two classes, the coefficient based method proceed the artifact reduction in a way that no more engagement is needed, while the other strictly post-processing based methods require being performed after the decoding step. From the model size point of view, the approaches employ the bigger models, the more recent they are.

4.4 SEGMENTATION

CNNs, in the field of image processing, were formerly used for image segmentation based on per pixel processing. Such an approach was highlighted in the recent time by Long et al. [33] described in Fully Convolutional Network (FCN) which became substantial for the FCN networks utilized for deblurring and JPEG artifacts restoration presented in this thesis. Several previous approaches of CNN based segmentation follow.

CRACK DETECTION BASED ON CNN CNN based sewer crack detection was published by Browne and Ghidary [91] where the CNN was introduced as a method of adaptive image processing which formed a link between adaptive filters and networks. Application of CNN to image processing on a mobile robot was described. The network consisted of 5 layers where the input and output had single filter with *Sigmoid* based activation while the 3 hidden from input to output 4, 3, and 2 each of size 4×4 with tanh activation function. Thus, the model consisted of 624 trainable parameters. The task was defined as a raw image per pixel filtering and consequently identifying the crack location which allowed detailed analysis including the width and length observation. Taking into account the variability in light conditions, orientation, scale, and other crack-like structures this task was considered complex and challenging. The work, unfortunately, was not compared with other man designed methods, although, according to the authors, 95% of crack pixels were recognized correctly where the training and evaluation were both performed on 15 images only. Authors emphasized the benefit of translation invariant weight sharing, learned filters together with other promising possibilities of CNN application.

3D VOLUMES SEGMENTATION BASED ON CNN Jain et al. [92] compared the CNN model and the Markov random field based model on the electron microscopic images in the task of voxel data restoration. The property of shared weight filters defining the CNN was utilized to emphasize the local image filtering. The CNN provided superior results because, according to authors, it can be trained to represent a highly sophisticated model, where the same is much more difficult with Markov random fields. Interestingly, when the CNN architecture was modified to be similar to Markov random field model the obtained results became similar.

The CNN model consisted roughly of 3.4×10^4 trainable parameters structured in 6 convolutional layers where each of 5 hidden layer included of 8 filters of size $5 \times 5 \times 5$ and using the sigmoid activation function. The model was trained with SGD based on the cross-entropy cost function optimization, usually used for binary classification. To obtain a robust segmentation, the super-sampled restoration based on $2\times$ up-sampled input by having 8 filters each looking at the same location and producing $2 \times 2 \times 2$ interleaved result provided more robust restoration i. e. not merging the objects with very thin boundary into one.

4.4.1 *Summary on Segmentation*

The application of CNN for image segmentation based on identifying the sewer cracks was proposed in 2003 together with various similar applications. The achieved results were often on the state-of-the-art level without a distinct improvement over other methods. This changed with the introduction of Fully Convolutional Network (FCN) for semantic image segmentation by Long et al. [33] which compared to 6×10^2 [91] had up to 1.3×10^8 trainable parameters. Based on the

deep CNN, the per-pixel segmentation substantially outperformed the engineered methods.

4.5 SUPER-RESOLUTION

The task of a super-resolution aims to provide a higher resolution of an input image, thus deliver more details related to higher pixel density. The scope of this thesis is not directly related to the super-resolution itself. Nevertheless, the earlier work of JPEG artifact removal presented by [37], and which was also influential in the work presented in this thesis, was based on the SR-CNN model.

A brief overview of the super-resolution approaches is given. That includes a formerly presented Auto-Encoder (AE), precisely the Denoising Auto-Encoder (DAE) based approach, the highly influential SR-CNN, and the improvements simultaneously introduced by this work and the super-resolution community, the residual based learning.

THE SUPER-RESOLUTION PIPELINE WITH AN AUTO-ENCODER The concept of a DAE, next to the other low-level image processing tasks, was employed in a super-resolution pipeline by Cui et al. [93] who presented a model called deep network cascade which gradually upscaled an input image. The cascade consisted of blocks where each included the nonlocal self-similarity search method and the stacked DAE.

The input image was interpolated to obtain the rough estimate of the upscaled high-resolution image. The base assumption was that more high-frequency details could be obtained from several overlapping image patches. That is, actually, a common assumption of several super-resolution methods. Every patch was taken from the same image yet always differently transformed i. e. scaled, blurred, etc. The input patch was then searched for the similar nonlocal patches. The roughly estimated patch was combined from sampled similar patches via a weighted average. Such an estimated patch contained an abundant, yet distorted, textural information. A collaborative DAE was used to denoise and refine the upscaled and roughly estimated image patch. The block of similar patch search method and the DAE was repeatedly run till the satisfying solution was achieved.

The cascade consisted of number, related to required scale, blocks which gradually produced a fixed scale up-sampled and clean image. The advantage compared to engineered methods is the possibility to stack any number of the upscaling blocks to achieve an arbitrary scale factor yet having the state-of-the-art results.

A new approach, which puts all these steps into a single end-to-end based model, was introduced by Dong et al. [36, 94]. The CNN was used to cover the whole pipeline of combining the patches of a low-resolution image into one and subsequently refining it to obtain a more dense pixel representation, the high-resolution image.

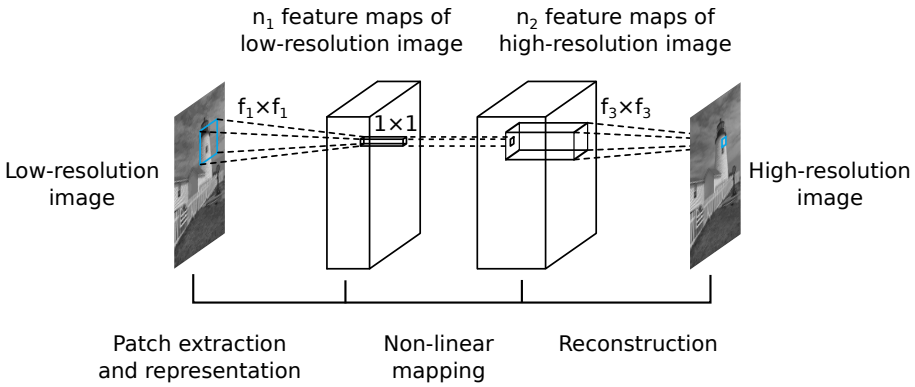


Figure 4.9: Super-resolution end-to-end model presented by Dong et al. [36] is the predecessor of AR-CANA model used for JPEG artifacts restoration.

SUPER-RESOLUTION END-TO-END CNN MODEL Interestingly, the JPEG artifacts removal approaches based on the CNN can be tracked down to the super-resolution task, where the influential end-to-end mapping feed forward model the Super-Resolution CNN (SR-CNN) was introduced by Dong et al. [36, 94]. It could be understood as a significant step which yielded from an engineered pipeline to a single model adapted by pure data-driven approach for the super-resolution task.

The only preprocessing step used fro employing the SR-CNN is the initial image upscaling to the desired size. The network consisted of 3 layers only. The first convolutional layer was identified as the patch extraction layer where the preprocessed upscaled input image is processed with 64 trained filters. The two subsequent layers, which consisted of 1×1 spatial filters, combined the previous activation maps and interleaved by the ReLU units provided a non-linear mapping. The last layer predicted the final high-resolution reconstruction Figure 4.9.

The proposed SR-CNN achieved the significant performance considering the model simplicity and application simplicity. On the other hand, the particular scale required the specifically trained model which was impractical considering arbitrary scale. The interesting and yet negative factor of SR-CNN was the huge number of training iterations to achieve comparable or beyond results compared with state of the art. Both drawbacks of the SR-CNN method were effectively addressed by Kim et al. [95] who presented the modified objective target and CNN model able to cover arbitrary scale.

RESIDUAL BASED LEARNING OF SUPER-RESOLUTION CNN The SR-CNN method immediately grabbed the attention which led to the highly improved much deeper model introduced by [95] based on the *VGG* [45] very deep convolutional network. The single model for multiple scales consisted of 20 layers which compared to 3 layers of SR-CNN brought a significant increase in the depth. Also, the overall receptive field increased from 13×13 to 41×41. The major difference lied in the objective the very deep network was trained for. The *residual* training was engaged which allowed to reduce the number of training iteration by 10^4 from 10^8 to 10^4 and yet achieve considerably better results compared to SR-CNN.

The residual learning forces the network to map the input image to the difference between the high-resolution and low-resolution image. The interpretation of such an objective function is that the network models much less complex problem compared to the mapping from low to the high-resolution image. The final step, after the utilization of the CNN model, is simply the addition of the mapped residual image to the input low-resolution image.

The very deep model with the *residual* learning achieved superior results compared to the SR-CNN and other engineered methods considered as state of the art. Besides, the one model is sufficient to be trained fro arbitrary number scales. Meanwhile, the very similar approach of residual learning yet extended by the skip architecture was simultaneously presented for JPEG artifacts removal.

4.5.1 *Summary on Super-Resolution*

The utilization of NN based methods for the task of super-resolution rapidly changed and evolved from introducing the CNN by Krizhevsky et al. [2]. Compared to the traditional approaches, where the NN based methods were a part of often complex pipeline consisting of several methods carefully combined [93], the nowadays solutions are solely based on the single CNN models. The progress of development and achieved results in recent time are significantly beyond the compared engineered methods. A convenient example is a qualitative shift seen on the relatively shallow SR-CNN to the very deep model of Kim et al. [95] which was based on the *VGG* network introduced by the computer vision community [45]. The trend of successful utilization of deep CNN in computer vision is thus directly projected to the image processing specifically the super-resolution task.

4.6 SUMMARY ON IMAGE PROCESSING BASED ON NN

A short overview of approaches for several types of image processing was presented. The tasks included deblurring, denoising, JPEG artifact removal, segmentation, and super-resolution. All these methods differ in many aspects including the architecture, a number of parameters, the way they are trained, etc. However, they have also something in common, specifically, the gradual simplification from the complex expert designed models to the single end-to-end mapping approaches. Such a trend is distinct from the chronological point of view.

An example is the relatively complicated AE model, primarily used for pretraining the deep computer vision models and later just “incidentally” employed for denoising, which is recently replaced by CNN based models which can restore more types of corruption yet with the higher quality at once. Nevertheless, this should not be mistaken with the model size, which is, in contrary, getting bigger.

A lot of methods exist which are adapted according to expert knowledge, e. g. the models estimating the inverse PSF later one used for deconvolution. Other NN methods are solely focused on processing the data in the Fourier domain etc.

Nevertheless, there are state-of-the-art methods based solely on the end-to-end mapping from corrupted to restored image which is trained without any task-specific model design. These include the text deblurring method with highly constrained textual image data, the super-resolution model based on the general *VGG* network used for image classification, and the JPEG artifact reduction model being one of the actual state-of-the-art considered methods.

The utilization of NN in image processing tasks has been highly influenced by the development of computer vision. That is clearly visible anytime the computer vision models significantly extended state of the art. The recent highly successive development in computer vision attracts, not for a first time, the image processing community to adopt these approaches for low-level image processing tasks.

5

CNN IMAGE RESTORATION

Image restoration based on CNN representing an unified approach is the core idea of this thesis. The unified method is based on an assumption of a single end-to-end model which directly maps the degraded image on the restored image. This model is purely data driven and in fact, may differ in its architecture which comprises the number of neurons, depth of a model and layers arrangement together with their type. That allows the end-to-end models shift the effort from designing the specific methods towards training objective definitions. The end-to-end approach allows simplifying extend or adapt the model on certain degradation level which involves just to train a model on new data.

The field of image restoration includes various types of degradations. Within the scope of this thesis, two different tasks of restoration were selected. It is the motion deblurring which together with the additive noise represents a linear transformation. The other is artifacts removal approach which deals with a non-linear transformation caused by the quantization step in the JPEG compression pipeline. These two types of degradations are a subject of restoration method based on the data-driven CNN models.

The majority of engineered restoration approaches comprise a particular processing pipeline. The deployment of NN in image restoration is usually associated with a certain step in the pipeline. These are, in fact, the vast majority of image processing NN approaches described in the previous chapter. Namely, the L0 regularized method [5] represents the most recent approach for blurred text image restoration. The Shape Adaptive Discrete Cosine Transform (SA-DCT) [6] is considered to be an up-to-date advanced method for JPEG artifact removal. Both represent the engineered approaches with the first-class results.

However, the recently introduced CNN based end-to-end methods provide significant outcomes often beyond what the widely used engineered approaches can achieve. The recent data-driven methods are represented by the text image denoising CNN [35] or JPEG artifact removal CNN model [37] which is an extension of the super-resolution model [36].

This chapter formulates the CNN based methods for license plate deblurring and JPEG artifacts removal. The presented models are based on almost only on the existing approaches often used in the field of computer vision. Both introduced approaches, compared to the vision related CNN methods, are extended and adapted for the image restoration requirements, which yields to regression instead of classification models. The main concepts are introduced which were used to train and deploy the network in both image restoration tasks. Follows the description of direct mapping approach. The improvements based on the skip architecture are introduced with the relation to gradient vanishing and neuron exploding problems.

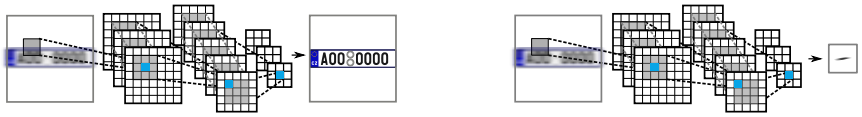
Several different objectives of the direct mapping and an initialization proposals are given. The data resampling is proposed to make the objective easier to learn. The chapter is closed by an introduction to the end-to-end approach for the non-image data restoration focused directly on the JPEG coefficients.

HYPOTHESIS *Most of the different image restoration methods is replaceable by a unified approach represented by CNN models which are end-to-end trained and often achieves state-of-the-art or even beyond results.* These models may differ in particular architecture or in the objectives they are trained for. The term unified covers the data-driven approach which adapts to a particular type of degradation, it does not inherently mean a single model. Different training objectives provide various speeds of convergence and rarely better models as well. The end-to-end mapping considers the direct transformation from a corrupted representation of a restored image. On the other hand, this approach would allow just to obtain a model for a particular type of degradation which needs to be restored. The following text comprises several ideas, assumptions, and considerations framed by the unified CNN based approach for image restoration. Based on the provided experiments, it often does not finally depend on the extensions primarily in the sense of performance, but in particular cases, different train objectives speed up the training in the sense of convergence time.

5.1 END TO END MAPPING

To introduce the end-to-end mapping based on the data-driven learned CNN model, the usual restoration pipelines of deblurring and image artifacts removal are quickly summarized. The common approach of deblurring is to estimate the PSF the image was corrupted with and to use it to restore its sharp representation. The restoration can be computed locally using the deconvolution with the inverse PSF or globally based on some specific global operator. However, estimating the PSF in a case of the blind scenario is an ill-posed problem. Several approaches were presented using the natural image priors, the histogram of gradients in the sharp image distribution, the specific spectrum properties in the frequency domain and other priors. Both steps, the PSF estimation, and the consequent deconvolution are prone to fail due to the noise, significant outliers, and other related causes. Thus, image deblurring is a specialized processing pipeline. The methods for JPEG artifacts removal, from the simple yet widely used SPP included in FFmpeg up to the SA-DCT utilizing the adaptive shape support to estimate the restoration, present the engineered post-processing approaches. These are entirely different from the methods for deblurring. Being highly specialized is the only common thing they share.

That is not the case of the end-to-end mapping approach considered in this thesis. This data-driven approach is based on the CNN, specifically the Fully Convolutional Network. The general FCN model is trained to process the input image directly. The image is transformed, scattered through the network layers in the



(a) (b)

Figure 5.1: The end-to-end mapping (a) is a direct transformation from the degraded image to its restored representation. In contrary, the recent engineered and learned methods (b) usually estimate the PSF to deconvolve the image.

feedforward transformation. That consists of gradually applied nonlinearity operators and convolutions. The whole network is trained to estimate the restored image or the error being the difference between the degraded and restored image. The last layer finally outputs the data straight in the pixel format with an arbitrary number of channels. Compared to the majority of previously learned methods, which comprise several steps including PSF estimation and consequent restoration, this approach provides quantitative simplification and simultaneously the qualitative improvement. The direct end-to-end mapping compared to the different approach based on a PSF estimation is shown in Figure 5.1. The definition is written

$$\hat{x} = F_L\left(\mathcal{W}, y\right), \tag{5.1}$$

where L defines the number of layers, $\hat{x}$ is the estimated non-degraded latent image x, y represents the input image corrupted by an arbitrary distortion, and $\mathcal{W}$ are the network weights and biases.

The common assumption related to the CNN depth, i. e. number of layers, is that the deeper models provide better results [32, 96]. That is in regards to reviewing the network as a complex data transformation where the layers compose a feature hierarchy representation. This thesis put the emphasize on the end-to-end models considering the ability to generalize over various parameter ranges in a restoration task to provide a single and unified model. The regression model is proposed, which in contrary to the classification, is generally harder to train[1] together with higher acquirements on the numerical precision. Finally, the end-to-end mapping architecture allows to be quite easily trained for specific parameters in case if needed, i. e. refine the model in case the parameters are roughly known. This approach was initially applied in the text denoising model presented in [35], for superresolution tasks [36, 94], and also for artifacts reduction [37]. Within this thesis, the end-to-end model is studied for two specific yet different image tasks, the motion deblurring, and artifacts removal.

5.2 ARCHITECTURE EXTENSION

Deeper networks may have problems with exploding and vanishing gradients and they may take a long time to learn to propagate information through a large num-

[1] Classification outcomes are much more limited compared to regression results, namely, compare classifying into two classes and the real number prediction.

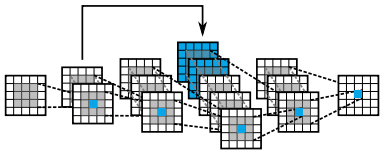


Figure 5.2: The skip architecture allows propagating the low-level image features from the front layer in the network to deeper layers.

ber of layers efficiently. The problems with the gradients can be eliminated by proper initialization [56, 57, 58] which takes effect in the beginning and predicts the overall training speed. The skip architecture influences the network weights during the whole training. This behavior is significant in a case the whole natural image propagates through a deep network in the end-to-end mapping approach.

Training deep models in case of image restoration are still quite a challenge. The problems with propagating information through many layers can be alleviated by bypassing some more deep layers [33]. Such an approach, the skip architecture, can beneficially improve the novel end-to-end methods as it contributes to building a deeper model. The goal of the skip architecture in the image restoration is to allow the network to pass geometric information easily from the input to the output, and to allow for more complex reasoning about the image content in the middle layers, e. g. in case of artifacts removal, what is an artifact and what local context information should be used to restore the image.

An arbitrary CNN model F_L of depth L which utilizes the skip architecture is shown on Figure 5.2 and could be written as

$$\begin{aligned} f_{l\|s}(x) &= h_{l\|s}\Big(\mathcal{W}_{l\|s}\big(f_{l-1}\,(x)\,\|\,f_s\,(x)\,\big)\Big) \\ \mathrm{F}_L\,(\mathcal{W}, x) &= \left(f_L \circ \ldots \circ f_{l\|s} \circ \ldots \circ f_s \circ \ldots \circ f_1\right)(x) \\ y &= \mathrm{F}_L(\mathcal{W}, x)\,, \end{aligned} \tag{5.2}$$

where the operator $\|$ denotes the concatenation and f_s is the skip layer, i. e. the one to be transfered, and $f_{l\|s}$ is the layer to which the skip one is concatenated to. The $f_{l\|s}$ layer is defined as a function which is computed on the concatenated activation maps obtained from f_{l-1} and previous layer f_s. The W denotes the CNN weights, trainable parameters including the biases and h_l is an arbitrary activation function. The skip architecture does not have to utilize the concatenation only but can be based on addition as can be found in Long et al. [33] who adds the activations together. The skip architecture utilizing the concatenation of activations from the arbitrary previous layer is proposed to a more challenging task of JPEG artifacts removing.

5.3 SPECIALIZED OBJECTIVES

The end-to-end mapping forces the network to transfer the whole general image through all the convolutional layers interleaved by non-linearities and to restore the degraded image parts while not touching the uncorrupted patches. It shows that such a straight approach requires more training, measured by a number of

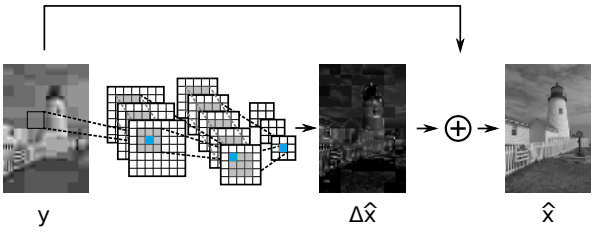


Figure 5.3: Pixel-to-Residual mapping network scheme.

iteration, however, it does not have to reach the best optimum in a case of restoring complex natural images. Moreover, the learning of such autoencoder-like mapping in situations where the input images are highly correlated with the desired outputs may be wasteful especially for broad and deep networks. It may be one of the main reasons why Dong et al. [37] were not able to scale up their networks and why they required approximately 10^7 iterations to train their AR-CNN. Similar problems were reported by Kim et al. [95].

RESIDUAL In specific tasks, the residual image can be learned instead of a highly variable natural image. Such an idea was first introduced by He et al. [47, 59] for a super-resolution based on the CNN, where the input and output images are highly correlated. The same approach for JPEG artifact removal is almost simultaneously introduced in this thesis which supports layers to learn a residual of their inputs. Instead of training the network to restore the whole image, the task could be defined to only complete the degraded image, i. e. to restore the residual $\triangle x$ between the input corrupted image y and the original latent uncorrupted image x. The residual objective is suitable for the task like JPEG compression artifacts removal, where the repeated blocking artifact occurs. The residual objective is written

$$\arg\min_{\mathcal{W}} \frac{1}{2} \sum_{i=0}^{N-1} \left\| F_L\left(\mathcal{W}, y_i\right) - \triangle x_i \right\|_2^2 , \tag{5.3}$$

where the latent residual is defined as $\triangle x = x - y$. The x corresponds to the ground truth image while the $\hat{x}$ is the result obtained by the CNN processing. The residual learning scheme is shown in Figure 5.3. Kim et al. [95] were able to speed up the training by the factor of up to $10^4\times$ with the residual learning and it allowed them to learn much deeper networks, 20 layers compared to three in [36] and four in [37].

EDGE ENHANCEMENT Mean square error used in many image restoration methods does not necessarily well correlate with the image quality perceived by humans. With convolutional networks, it is relatively easy to use more perceptually valid error measures as long as they can be efficiently differentiated. Therefore, next to the residual objective, the edge enhancement learning is proposed to support the human edge sensitivity perception. The partial first derivatives of the

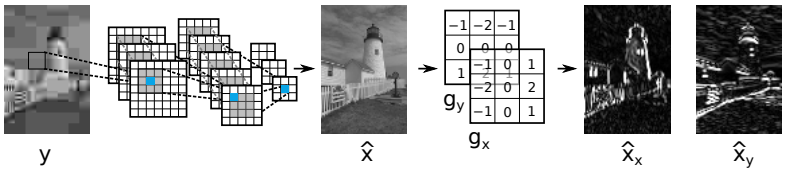


Figure 5.4: Scheme of a restoration network trained with the emphasize on edges.

image with the image itself are the inputs into the loss function. The input is in the form of the transformed image x_e defined as

$$x_e = \left[x,\ x * g_x,\ x * g_y\right] , \tag{5.4}$$

where g_x and g_y represent the *Sobel* [97] horizontal respectively vertical operators. The x_e is thus the concatenation of the original image and its horizontal and vertical edge enhanced representation. The objective utilizing the edge priors in y_e and x_e is defined

$$\arg\min_{\mathcal{W}} \frac{1}{2} \sum_{i=0}^{N-1} \left\| F_L\left(\mathcal{W}, y_{e\,i}\right) - x_{e\,i} \right\|_2^2 . \tag{5.5}$$

The scheme of edge enhancement deployed in the network architecture shows Figure 5.4. The assumption is that the addition of the first derivatives should force the network to focus specifically on high-frequency structures such as edges, ringing artifacts, and blocking artifacts and it could lead to perceptually better restorations. The combined edge emphasized loss can be easily implemented in all existing convolutional network frameworks by defining the Sobel derivative kernels as a convolutional layer with predefined fixed filters.

PSNR The quality of the restored images is measured is measured in several metrics, e. g. the signal focused PSNR and more human perception adapted Structural Similarity (SSIM) index [98, 99]. The loss function usually used based on the squared ℓ^2-norm can be with several assumptions swapped to the loss emphasizing function. The network, therefore, can focus on restoring the image to be more visually plausible or to provide better values measured by particularly metric. The loss function based on PSNR is introduced together with its differentiation needed for the backpropagation, i. e. the chain rule. PSNR based on the MSE is defined

$$\mathrm{MSE}\left(\hat{x}, x\right) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left(\hat{x}_{mn} - x_{mn}\right)^2 \tag{5.6}$$

$$\mathrm{PSNR}\left(\hat{x}, x\right) = 10 \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}\left(\hat{x}, x\right)} \right) , \tag{5.7}$$

where $\hat{x} = F_L\left(\boldsymbol{W}, y\right)$ is the network restored image and x is the latent uncorrupted image, and MAX represents the maximum intensity value the image can be of, i. e. 1 in the case oh having the image values in the range $[0, 1]$. The loss function based on the PSNR is then defined

$$\arg\min_{W} \left(-10 \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}\left(\hat{x}, x\right)} \right) \right) , \tag{5.8}$$

where the minus sign is present to keep the minimization, i. e. the gradient descent approach. Within the CNN based image restoration, the PSNR objective is proposed. Its differentiation w.r.t. to the input, i. e. the restored image is written

$$\frac{\partial \,\mathrm{PSNR}(\hat{x}, x)}{\partial \hat{x}} = \frac{\partial 10 \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}(\hat{x},x)}\right)}{\partial \hat{x}}\,, \tag{5.9}$$

which equals to the partial differentiation written in the *Jacobian* matrix yielding to just rescaled error

$$\begin{aligned} k &= 20\left(\log(10)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(\hat{x}_{mn}-x_{mn})^2\right)^{-1} \\ \frac{\partial PSNR(\hat{x},x)}{\partial \hat{x}} &= \begin{bmatrix} (x_{00}-t_{00})\,k & \dots & (x_{0N}-t_{0N})\,k \\ \vdots & \ddots & \vdots \\ (x_{N0}-t_{M0})\,k & \dots & (x_{MN}-t_{MN})\,k \end{bmatrix}, \end{aligned} \tag{5.10}$$

where the small errors has higher cost compared to the large ones. The interpretation of PSNR loss function in the task of JPEG compression artifacts removal is based on the sensitivity to distortions in the stationary regions of the image like the sky and the clearly visible blocking artifacts in such a region. Finetuning the model could utilize these properties to focus on the ostensibly small errors yet more noticeable compared to high errors in the image areas with heterogeneous structure.

5.4 TASK SPECIFIC MODIFICATIONS

All the mentioned methods operate directly with the image pixels. In a case of a JPEG file, this leads to an additional postprocessing, which is computed after decoding the image. On the other hand, utilizing the technique of JQT [88] allows to process the DCT coefficients directly. In this thesis, the new approach of CNN based JPEG file coefficients processing to suppress or remove the high compression related artifacts is proposed. A scheme of such a network which transforms the JPEG coefficients to coefficients representing the restored image is shown in Figure 5.5 where, nevertheless, the loss is computed through the pixels.

The coding and decoding pipeline described in Section 2.2.1 transforms the 8×8 image patches into the 8×8 of DCT coefficients which correspond to specific frequencies in that patch. These coefficients noted as B are sorted based on their frequencies in the *zig-zag* manner. Based on the user specified compression quality the predefined quantization table Q is selected and the DCT coefficients are quantized and rounded as defined in (2.16). The quantization affects the amount of blocking and ringing artifacts and implicates two potential types of CNN input, the quantized DCT coefficients $\mathcal{B}$, where the network is forced to learn the quantization table Q as well, and the DCT coefficients $\mathcal{B}$ already per element multiplied by the quantization table, the $Q\mathcal{B}$.

The non-linearity caused by the quantization of otherwise linear DCT transform (2.14) affects the network loss function. The properties of the loss computed

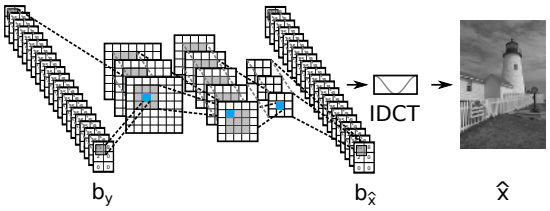


Figure 5.5: DCT-to-Pixel mapping network with a predefined IDCT layer.

on quantized $\mathcal{B}$ or quantization table multiplied coefficients $Q\mathcal{B}$ differ from the loss calculated on the decoded values – the pixels. That means that the network trained on minimizing the loss of coefficients is actually producing different restoration compared to training the network based on the pixel loss. That is given by the different gradients of the loss computed on pixels versus the coefficients $Q\mathcal{B}$. Next, the T.81 recommendation [19] states the IDCT transformed values have to be clipped to fit into the range of the image domain which also influences the loss.

The IDCT layer is defined to being able to compute the loss function directly on the pixels and further backpropagate the loss computed gradients. To follow the chain rule of differentiation described in (3.12) the differentiation of IDCT (2.17) w.r.t. the input dequantized coefficients QB is defined in (5.15). Therefore, the backpropagation through the IDCT layer equals to

$$\frac{\partial \mathcal{F}_c^{-1}\left(Q\mathcal{B}\right)}{\partial Q\mathcal{B}} \triangle d = \mathcal{F}_c\left(g\right) \ , \tag{5.11}$$

where the partial differentiation of the IDCT $\mathcal{F}_c^{-1}$ multiplied by the gradients $\triangle d$ from the layer above is equal to the discrete cosine transform $\mathcal{F}_c$.

The inference of the IDCT differentiation consists of several steps. First, consider to dequantized coefficients $Q\mathcal{B}$ to be denoted as c which is defined as $c = Q\mathcal{B}$. First, the partial differentiation of the $\mathcal{F}_c^{-1}$ w.r.t. c is written

$$\frac{\partial \mathcal{F}_c^{-1}\left(c\right)_{mn}}{\partial c} = \begin{bmatrix} \frac{\partial \mathcal{F}_c^{-1}(c)_{mn}}{\partial c_{00}} & \cdots & \frac{\partial \mathcal{F}_c^{-1}(c)_{mn}}{\partial c_{0,N-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{F}_c^{-1}(c)_{mn}}{\partial c_{M-1,0}} & \cdots & \frac{\partial \mathcal{F}_c^{-1}(c)_{mn}}{\partial c_{M-1,N-1}} \end{bmatrix} , \tag{5.12}$$

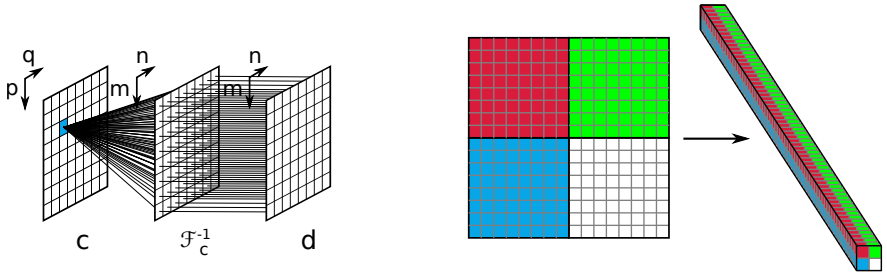
where the differentiated element of the Jacobian matrix reduces from the summation to a single expression

$$\frac{\partial \mathcal{F}_c^{-1}\left(c\right)_{mn}}{\partial c_{pq}} = \alpha_p \alpha_q \ \cos\left(\frac{\pi\left(2m+1\right)p}{2M}\right) \cos\left(\frac{\pi\left(2n+1\right)q}{2N}\right) . \tag{5.13}$$

Second, all the Jacobian matrices (5.12) written in the general expression define the whole 8×8 differentiated patch w.r.t. c in the form of

$$\frac{\partial \mathcal{F}_c^{-1}\left(c\right)}{\partial c} = \begin{bmatrix} \frac{\partial \mathcal{F}_c^{-1}(c)_{00}}{\partial c} & \cdots & \frac{\partial \mathcal{F}_c^{-1}(c)_{0,N-1}}{\partial c} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{F}_c^{-1}(c)_{M-1,0}}{\partial c} & \cdots & \frac{\partial \mathcal{F}_c^{-1}(c)_{M-1,N-1}}{\partial c} \end{bmatrix} . \tag{5.14}$$

Based on this expression, the backpropagation of gradient $\triangle d$ is equal to the summation of per element multiplication of the top layer gradients $\triangle d$ and the corresponding partial differentiations $\partial \mathcal{F}_c^{-1}\left(c\right)_{mn} / \partial c_{pq}$. That is written as the equation



(a) (b)

Figure 5.6: The illustration of the backward gradient propagation (a) through the IDCT layer from right to left. D are the gradients computed in the loss function. The contributions of all the gradients to every coefficient $\mathcal{QB}_{pq}$ shows the left part of the figure (a) and is equal to the discrete cosine transform $\mathcal{F}_c(D)$. The 4 pixel blocks of size 8×8 (b) resampled to the 4 64 channel vectors.

$$\mathcal{F}_c(\cdot)_{pq} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \triangle d_{mn} \frac{\partial \mathcal{F}_c^{-1}(c)_{mn}}{\partial c_{pq}} , \tag{5.15}$$

which, if expanded, directly equals to the discrete cosine transform $\mathcal{F}_c()$

$$\mathcal{F}_c(\cdot)_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \triangle d_{mn} \cos\left(\frac{\pi(2m+1)p}{2M}\right) \cos\left(\frac{\pi(2n+1)q}{2N}\right) . \tag{5.16}$$

The illustration of the gradients backpropagation through the IDCT layer is shown in Figure 5.6a Based on the defined inverse discrete cosine transform layer, the network the decoding layer is deployed in is defined

$$\begin{aligned} \mathrm{F}_L(\mathcal{W}, x) &= \left(\mathcal{F}_c^{-1} \circ f_{L-1} \circ \ldots \circ f_1\right)(x) \\ y &= \mathrm{F}_L(\mathcal{W}, x) , \end{aligned} \tag{5.17}$$

where the loss function is computed directly on the $\mathcal{F}_c^{-1}$ output of a layer, i. e. pixels.

In a case of JPEG artifacts, it is simple to define the prior, e. g. the blocking artifacts occur every 8th pixel. That can be utilized in the form of resampled input which is illustrated in Figure 5.6b. The input patches are resampled from 8×8 into 64D vectors. Meanwhile, the resampled input is proposed to be used with the DCT coefficients. The same technique is introduced for the pixel input data. However, the motivation to resample the data differs in both cases, coefficients and pixels. The resampled input data in the cases of the quantized or dequantized coefficients provides the network the possibility to learn the spatial filters which can utilize the continuity of the related frequencies represented by the coefficients. The resampling, within the pixels based method, is suitable due to the blocking artifact properties, namely its fixed position and repeating structure. Resampling these 8×8 blocks into the 64D channel vectors can directly support the network to utilize the blocking prior.

5.5 SUMMARY

The general end-to-end mapping Convolutional Neural Network approach has been introduced with several adjustments for the text based image motion deblurring and JPEG artifacts removal. The concept of the end-to-end mapping has been clarified. Nevertheless, it is not an entirely new technique, this thesis puts the emphasize on such an approach because it has an impressive potential to be successfully deployed in a variety of different tasks. The end-to-end data-driven direct mapping has been slightly improved using the skip architecture which concatenates the previous activations to the activations deeper in the network. This skip allows to transfer the features of input data deeper into the network and provide a more complex basis for further reasoning.

A set of specialized objectives has been introduced. These allow the network to focus on a specific subject to learn like the residual learning which is much less model capacity demanding compared to the full image end-to-end approach. An edge enhancement technique based on the Sobel operators has been proposed to support the heterogeneous structures in the images. The loss function based on the PSNR has been introduced to allow the narrowly focused optimization which compared the usually used MSE based loss function forces the network to rate the errors differently.

Finally, in a case of JPEG artifacts removal, the possibility to suppress the artifacts directly in the DCT domain is described. The specialized IDCT layer is proposed to allow the direct end-to-end mapping yet training on the pixel loss function instead of coefficient loss function which has different properties. The coefficients arrangement allows utilizing the samples-coefficients between connectivity and directly learn the adapted spatial filters. The similar prior and the same approach has been introduced for pixels, where the resampled data organization from 8×8 block to the 64D channel vector allows the network to adapt directly on the fixed blocking artifacts. The extensions and techniques of CNN based model show the applicability and deployment in the tasks of image restoration which is empirically proved later in this work.

6

EXPERIMENTS

The CNN models based on the proposals given in the previous chapter are deployed and studied in the field of image restoration. Namely, it is the motion deblurring of images captured by the surveillance system and the high compression JPEG artifacts removing. Various experiments show the strengths yet also some weaknesses the CNN models have. The presented approach is viewed from two different perspectives. Firstly, the contribution which the proposed methods deliver in comparison with the other widely used approaches is shown. Secondly, the description of how the models behave, which includes the model generalization possibilities, several model extension impact, and other more or less task-specific traits, is presented. Almost all the experiments have very similar structure. This consist of the way data are retrieved, a model specification, a description of the training procedure, and finally the achieved results with their interpretation.

The vast majority of data is artificially produced from the latent undistorted, i. e. ground truth, images. Interestingly, model based on artificial data works very well as it is shown later in this chapter on the image deblurring task. However, it is not so much surprising in the case of artifacts removal, where this is the only way to acquire the training data. It is important to mention that all the experiments were performed using caffe [62], the fast open framework for deep learning which allowed to concentrate on the model itself instead of the network implementation.

In the beginning, the attention is directed to the deblurring of license plate images [SHMZ16]. That presents the end-to-end mapping model of 15 layer network. Besides the reported results beyond state of the art, an interesting generalization ability these models have is revealed. Various models are trained for an identical degradation of different levels. That shall provide a perspective how well the CNN approach restores the images of different degradation level compared to blind and non-blind approaches. The part describing the JPEG artifacts removal [SHBZ16] addresses the majority of the proposed network enhancements including the different objectives, extended architecture, and processing of DCT coefficients instead of pixel.

The last part of this chapter names the possible CNN exploitations in various fields including the surveillance systems, data storage, transfer based services, and user photo-based applications. The future work and possible research directions based on the results of this thesis are outlined. Finally, the very last brief summary closes this chapter.

LIST OF EXPERIMENTS

License plate motion deblurring

- Length range
- Direction range
- Real data deblurring with model trained on artificially blurred data
- OCR accuracy comparison with state-of-the-art method

JPEG compression artifacts removal

- Three different architectures, L4, L8, and L5
- Direct, residual, edge enhancement and PSNR objectives
- Comparison with state-of-the art methods
- Generalization over various compression qualities restoration
- Training dataset size impact
- Resampled input
- Coefficient based restoration
- An impact of quantized vs dequantized coeficients

6.1 CNN FOR MOTION DEBLURRING

The majority of methods mentioned in the IMAGE PROCESSING BASED ON NEURAL NETWORKS used for deblurring do not utilize the direct end-to-end mapping. The only exception is the work of Hradis et al. [35] who focused on noise corruption and out of focus blurred text restoration. The other methods deploy the end-to-end mapping but not as an integral solution but more as a subtask [77, 78] which estimates the PSF to be later used in the deconvolution itself. An experiment with the non-blind and blind approach as well is performed on the task of license plate motion deblurring utilizing the 15 layer architecture introduced by [35]. This experiment addresses the model generalization properties and the comparison with blind and non-blind deconvolution approaches .

6.1.1 *Architecture*

This 15 layer fully convolutional network architecture, *L*15 *CNN*, is selected to train the motion deblurring end-to-end mapping model. The reason this model has been selected is the success of this model based on the out-of-focus text deblurring results published in [35]. The motion deblurring L15 network definition, based on the notation in (3.2), is written

$$\begin{aligned} \mathrm{L}_{15}\left(\mathcal{W}, y\right) &= \left(f_{15} \circ f_{14} \circ \ldots \circ f_1\right)\left(y\right) \\ \hat{x} &= \mathrm{L}_{15}(\mathcal{W}, y)\,, \end{aligned} \tag{6.1}$$

where y is the degraded input image, $\boldsymbol{W}$ are the network weights including biases, f_i represents the ith convolutional layer with the consequent activation ReLU function, and $\hat{x}$ is the restored estimation of the latent sharp image x. Besides the

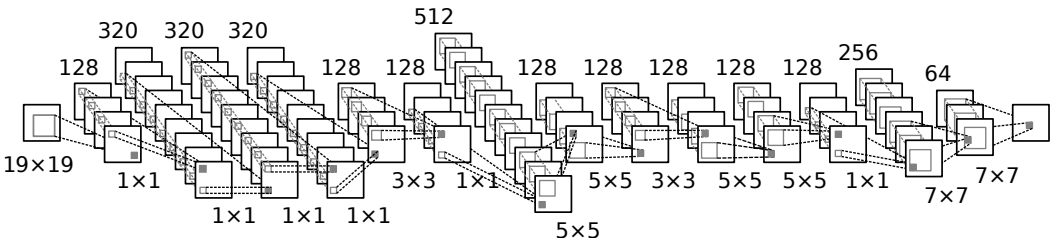


Figure 6.1: L15 architecture with a number of filters per layer, their spatial size, and the preview of grouped channels in the second half of the network. L15 consists solely of convolutional layers followed by the ReLU activation layers (not shown) providing the FCN model.

formal notation, the Figure 6.1 and Table 6.1 show and describe the exact network architecture with several channels grouped together.

The spatial sizes of the network filters and the composition of the layers provides the network with the receptive field of 50 px. The implementation of convolutions yields to 25 px crop of the input image. That is caused by computing the convolutions without any padding. The L15 network architecture consists of grouped data and related filters in its second half. That helps to reduce the total number of parameters. The grouping is shown in Figure 6.2b. Such an architecture is trained on data generated according to various motion blur parameters, namely the length and direction.

6.1.2 *Data*

All the data the presented network is trained on, are artificially generated. A random blur kernel is computed representing simple linear motion blur PSF. The kernels are generated with the sub-pixel accuracy to cover the generally nondiscrete space, That is achieved by drawing a line representing the motion blur PSF with the 100× scale and finally resampled into the required length using pixel area relation method which gives moiré-free results in image decimation. The final motion blur kernel has odd dimensions. The same technique is used to sample various directions. The drawn line, representing the motion blur, is rotated based on the sampled direction. This kernel is subsampled into right sized PSF. The motion blurred image is further corrupted by an additive white noise sampled from the user defined parameters. That helps to generate artificially blurred images

Table 6.1: L15 model architecture. The group parameter represents splitting the input into several groups connected solely with a related group of filters and providing the corresponding group of outputs. The architecture is further defined by the filter spatial size and number of channels i. e. filters of each layer.

Layer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Groups	1	1	1	1	1	1	1	4	2	1	2	2	1	2	1
Filter size	19	1	1	1	1	3	1	5	5	3	5	5	1	7	7
Channel count	128	320	320	320	128	128	512	128	128	128	128	128	256	64	3

reflecting the natural images captured in the real-world conditions. Such a data processing allows generating arbitrary linear motion blur PSF used to produce the final blurred image. With the sub-pixel accuracy, the data augmentation allows generating random sized training dataset.

In a case of the end-to-end mapping approach, it is crucial that the ground truth x images are not corrupted. The data used for generating the artificially blurred images are images captured with various imperfections. These are mostly based on the conditions what the real surveillance system operates in. A small fraction of all images was therefore mostly blur distorted or captured in a poor light, i. e. contained high levels of noise. For this reason, the ground truth dataset was processed to filter out the highly corrupted images. The detection of such images was based on an approach based on the high and low-frequency ratio. An ad-hoc threshold was chosen based on the observation to filter out the degraded images. The final dataset consists of 140 k clean and sharp license plate images.

Nevertheless, the disjunct set of naturally blurred data was collected including 721 images of various motion blurred license plates. These were used for verification the model works well on naturally blurred images as well, where the blur PSF usually is not a straight line but reflects some curved trajectory. These images were taken by two static surveillance cameras controlling the road under different angles with the restricted range of directions the vehicles could approach. The cameras were set to capture the images with near uniformly sampled exposition times from 6 ms to 12 ms with the step of 2 ms on the road where the official speed limit is up to 90 km h^{-1}. These images were cropped around the license plate and normalized to the size of 264×128 px. They were carefully manually annotated with license plate characters such that OCR accuracy could be evaluated. The approximate direction range the captured cars did approach were 37° to 57° and 59° to 79°, see the Figure 6.3.

Figure 6.2: On the left figure (a) is a part of a network where the second layer includes 2 filters with spatial size 3×3 and 4 channels i. e. fan-in$_2 = 3 \times 3 \times 4$, fan-out$_2 = 3 \times 3 \times 2$. On the right figure (b) the example of grouping is shown, where the activation map is split into two groups which yield to a filter applied only inside this group, the blue color.

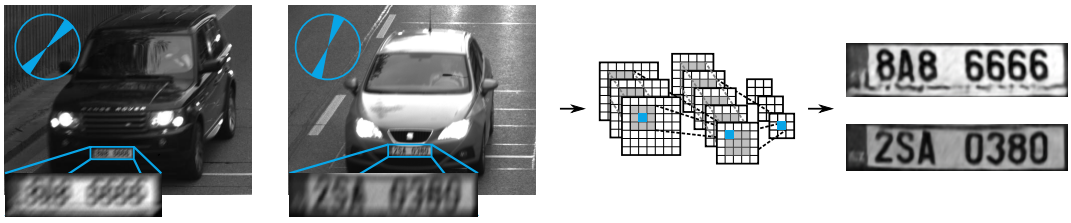


Figure 6.3: The illustration of the surveillance system images and the correspondent results of L15. The blue circle shows the approximate direction range the cars usually approach, where the left range equals to 37° to 57° and the right range to 59° to 79°. The blurred input and restored license plate images are shown.

6.1.3 *Training*

The pairs of artificial blurred image and its sharp undistorted representation (y_i, x_i) were divided into two disjoint parts. The training set which consisted of 126 k pairs and the testing set which had 14 k pairs of images. All the images were of the same size 264 × 128 px. The model was trained on fixed size crops with the dimension of 66 × 66 px, where 5 randomly sampled crops per training image created the set of 630 k input crops. Because the receptive field of the model is 50 px, the output images, the model produce, are only 16 × 16 px central patches of the input cropped images. The pair of training data is shown in Figure 6.4.

The whole network was initialized using the modified[1] *Xavier* initialization [57]

$$\mathrm{Var}\left(\mathcal{W}_i\right) = \sqrt{\frac{3}{\text{fan-out}_i}}\,, \tag{6.2}$$

where the variance distribution of the initialized convolutional layer weights W_i is related to the number of filters, precisely on the fan-out$_i$ parameter which is defined as spatial filter size × number of filters in the layer f_i (Figure 6.2a). The network was trained for 400 k iterations with a mini-batch of 54 samples. The objective was based on minimization the loss function defined as

$$\frac{1}{2N}\sum_{i=1}^{N}\left\|L_{15}\left(\mathcal{W}, y_i\right) - \hat{x}_i\right\|_2^2\,, \tag{6.3}$$

where N is the number of training pairs in the mini-batch of degraded image y_i and its ground truth sharp central patch representation x_i. The network took on average 3 days to train on a single Nvidia GeForce 980 GPU. Initial learning rate was set to 4×10^{-5} and it was reduced five times by a factor of 2. The weight update was performed based on the SGD with the momentum equal to 0.9 and the weight decay 5×10^{-4}. All the input data were normalized and centered around zero.

6.1.4 *Semi Non-Blind Restoration*

Two experiments were performed to assess the behavior of deblurring CNN on motion blur length and a range of blur directions. These experiments were performed

[1] Based on the implementation in Caffe [62].

Figure 6.4: The training image pair with the illustrated blurred crop on the left and the equivalent sharp center patch on the right. In the middle is the magnified motion blur PSF.

on the artificially blurred images. The restored image quality was measured based on PSNR (5.7). The deblurring model is first trained on specific motion blur parameters defined as the range of the motion blur length and the range of direction.

There were 4 models trained with the fixed direction range to 20° and gradually increasing the motion blur lengths including 0–5 px, 0–9 px, 0–13 px and 0–17 px. The length was always uniformly sampled from the corresponding range. Figure 6.5a shows the results of these networks for different blur lengths. These results indicate that networks trained for shorter blur length range perform better inside these ranges. However, their results degrade rapidly outside the trained range. The restoration quality starts to degrade already at the border of the respective ranges. That is probably because no larger blurs are represented in the respective training sets. The reconstruction quality decreases linearly for longer blur kernels.

The second experiment is shown in Figure 6.5b assess the performance of the networks for different blur direction ranges. Seven models were trained, one model per different direction range, including the uniformly sampled, 10°, 20°, 40°. 60°, 90°, 130°, and 180° wide ranges of possible directions. Note that the blur kernels are symmetric and consequently the largest range of 180° covers all the possible directions. All the directions were blurred with a length uniformly sampled from 0–13 px. The observed results show similar trends as in the experiment with different blur lengths, the networks trained for tighter direction ranges perform better inside these ranges, but their performance degrades rapidly outside the respective direction ranges.

6.1.5 *Blind Restoration of Naturally Blurred Data*

Six models were trained to provide the evaluation on the naturally blurred test images captured by two surveillance cameras. These networks were all trained on blur kernels covering both cameras, i. e. the range of the blur directions was 50° wide, which shall be sufficient according to the possible directions of approaching vehicles. The networks were trained for blur lengths 0–9 px, 0–11 px, 0–15 px, 0–19 px, 0–21 px, and 0–23 px. The L0-regularized blind deconvolution method by Pan et al. [5] was selected as a representative of the traditional blind deblurring methods to serve as the baseline for a model comparison. This method is specifically optimized for images containing text and it should be suitable for the license plate images as well. Optimal parameters of L0-regularized were selected using the grid search directly on the test images.

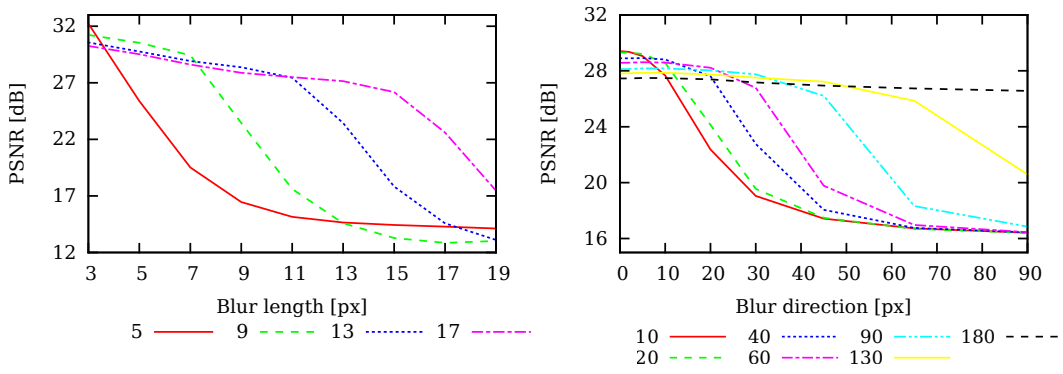


Figure 6.5: The graph on the left shows the result of specific length range trained model on several blur lengths. On the right, the presents results of models trained on specific ranges evaluated on several direction ranges.

Figure 6.6 shows results on the naturally blurred images as an accuracy of an Optical Character Recognition system. The deployed OCR system[2] is optimized for license plates and is used in commercial traffic surveillance systems. The networks trained for shorter blur perform poorly as the set contains blurs up to 19 px long. The networks trained for sufficiently long blurs significantly outperform the baseline blind deconvolution method of Pan et al. [5]. The improvement is from the character error of 23% down to 9% compared to the L0-regularized which corresponds to relative improvement by a factor more than 2. It is worth to emphasize that the OCR accuracy keeps approximately the same for the models trained for long blurs. In a case of nonblind restoration, the L0-regularized method tuned per license plate to performs similarly to the blind approach based on CNN. However, this requires the known motion blur parameters for each license plate. Figure 6.7 presents the original blurred images, reconstructed license plates by L0-regularized blind deconvolution and the L15 restorations.

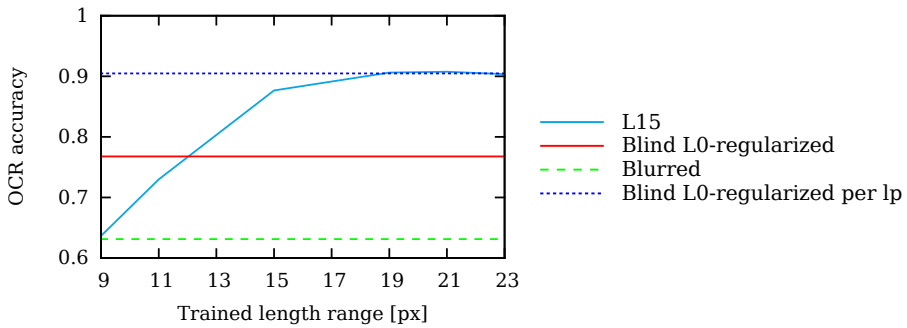


Figure 6.6: The OCR accuracy results of the originally blurred images, L0-regularized blind and non-blind deconvolved images and the L15 restorations.

[2] UnicamLPR, http://www.camea.cz/

Figure 6.7: The naturally blurred license plates sorted from left to right based on the blur amount with the corresponding deconvolved results of L0-regularized method and the restored L15 images.

6.1.6 *Summary on motion deblurring*

The evaluated L15 architecture contains of 2.3 M unique weight parameters, i. e. the model occupies approximately 9 MiB in memory. Compared to convolutional networks used in computer vision tasks, this network is still small and computationally relatively efficient. It requires 2.3 M multiply-accumulate operations per pixel. The CNN proved to be effective for the naturally blurred images even though they were trained only on images which were blurred artificially with a simple line kernel. The deblurring CNN provided superior accuracy of a consequent OCR compared to the state-of-the-art L0-regularized blind deconvolution tuned for text images [5]. These results show for the first time that CNNs provide quantitatively better deblurring quality compared to engineered state-of-the-art blind methods in a practical application.

The experiments showed that the quality of reconstructed images could be improved by customizing the CNNs for the specific range of blurs. However, the improvement is only modest in the target application, and general networks trained for a wide range of blurs still provide the high-quality results. The reconstruction quality declines linearly, in PSNR, with the increasing length of the blur kernels which makes it easy to predict possible reconstruction quality for larger blurs. Although the networks can reconstruct real images which suggest that the kernels used for training do not have to match the shape of kernels in a real application too closely, the reconstruction quality degrades quite sharply for blurs which parameters like direction and length range are outside the trained values. The deblurring CNN are well suited for embedded applications due to their flexibility, relatively low computational power requirements, robustness, and the absence of any tunable parameters. The deblurring CNNs can be considered mature and ready to be deployed in the traffic surveillance systems.

6.2 CNN FOR JPEG ARTIFACTS REMOVAL

The end-to-end mapping network architecture is deployed for JPEG compression artifacts removal. Its utilization is principally based on the achieved results of the CNN model in motion deblurring. The artifacts are caused and clearly visible by

a low compression quality. That is caused by setting the higher frequency related coefficients during the quantization step to zero. On the other hand, this loss is redeemed by achieving the high compression ratio. The way the coefficients are omitted is related to the human perception where the less sensitivity correlates with the high frequencies and vice versa.

Several metrics exist to assess the perceptual quality of images objectively. In this work, the restoration is measured based on PSNR, PSNR-B, and SSIM metrics. Generally, the most commonly used quality metric is the MSE [99] (5.6). This quantity is computed by averaging squared intensity differences of the distorted image and the reference image. That is often expressed in a logarithmic scale as the Peak Signal to Noise Ratio (PSNR) (5.7). Unfortunately, PSNR and MSE are not necessarily well correlated with the perceptual quality.

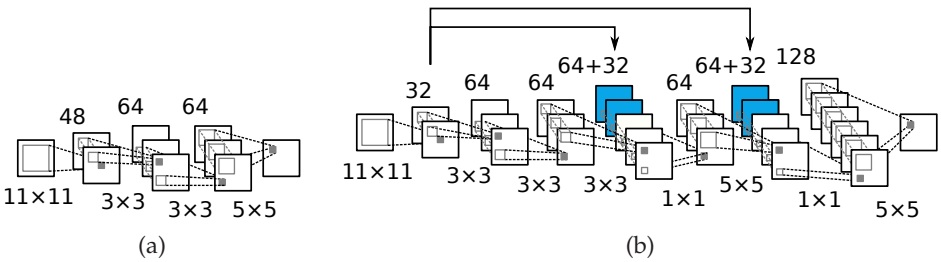
The SSIM [98] that compares local patterns of pixel intensities should better correlate with human perceptual quality. Since the attention is focused on the JPEG artifacts, the blocking artifacts, a block-sensitive metric referred to as the PSNR-B [100] is used to provide additional insights. PSNR-B modifies the original PSNR by including an additional blocking effect factor (BEF). Some experiments report IPSNR which is a PSNR increase compared to PSNR of the degraded image. IPSNR is more stable across different dataset and it directly reflects the quality improvement.

In regard to the color space $Y'C_BC_R$ which represents the luma Y', C_B blue-difference, and C_R red-difference chroma components, the most details are covered in the Y' luma channel. That is the primary reason why the main attention in this work is focused almost on the Y' luma channel only. Note, that the JPEG compression is by definition a nonlinear degradation compared to the almost only linear based motion blur.

In contrary to the deep L15 network, several small architectures are introduced including the 4, 5 and 8 layer networks L4, L5, and L8 respectively. The L4 network is a simple model similar to the AR-CNN [37] with the main distinctions in the training and related objective function. The results are compared to AR-CNN, to the widely regarded deblocking oriented SA-DCT [6, 7], and to a simple postprocessing filter SPP included in the FFmpeg framework [23]. The deepest L8 network introduces an extended skip architecture described in Section 5.2. The L5 network is used to compute, besides the pixels, directly mapping on the DCT coefficients as well. Regarding the specific architecture and different training dataset, L5 is not directly comparable with the other architectures.

6.2.1 *Architectures*

The L4 is a shallow network trained regarding direct, edge enhancement, and residual objective. The network size is comparable with the AR-CNN which is actually recognized as the state-of-the-art CNN based method. The entire L4 network receptive field is 19 px where, considering the block size of 8×8, the whole JPEG block and half is covered on each side, which provides the network with possibly



(a) (b)

Figure 6.8: The L4 (a) is a simple and shallow FCN network. The L8 (b) deploys the skip architecture which allows transferring the layer representation deeper into the network. The activation maps of the first layer are transfered into 4th and 6th layer.

sufficient spatial information. The L8, except to be a deeper model, differs mainly in the skip architecture defined as

$$\begin{aligned} f_4(x) &= h_4\Big(W_4\big(f_3(x) \,\|\, f_1(x)\big)\Big) \\ f_6(x) &= h_6\Big(W_6\big(f_5(x) \,\|\, f_1(x)\big)\Big) \\ \mathrm{L}_8(\mathcal{W}, y) &= (f_8 \circ f_7 \circ \ldots \circ f_1)(y) \\ x &= \mathrm{L}_8(\mathcal{W}, y)\,, \end{aligned} \tag{6.4}$$

where the operator $\|$ denotes the concatenation. The layers represented by f_4 and f_6 are defined as functions which are computed on the concatenated activation maps obtained from f_1 and previous f_3 and f_5 layers. The receptive field of whole L8 network is 25 px. L4 and L8 include solely the convolutional layers followed by the nonlinear ReLU units. Both architectures are shown in Figure 6.8.

The last architecture, L5, illustrated in Figure 6.9, is slightly deeper compared to the most shallow L4 network, but in the same time much wider than any here presented network. Such a width is closely related to the data the network is fed with as it mainly is the DCT coefficients resampled from the 2D 8×8 blocks into the 1D 64 channels vectors as illustrated in Figure 5.6b. The same L5 architecture is trained for identically resampled pixels with an assumption that the block structure, which is coded directly into the input data arrangement, provides an additional information the CNN can utilize. The L5 model has several modifications related to the type of input data. In the case of pixel input data, L5 is a straight end-to-end mapping architecture, while in the case of the coefficients input data, the network is extended by a fixed IDCT layer similarly as in Figure 5.5 which allows computing

Table 6.2: L4 and L8 architectures. The architecture is defined by the filter spatial size and number of channels, i. e. filters of each layer.

Layer	1	2	3	4	1	2	3	4	5	6	7	8
Filter size	11	3	3	3	11	3	3	3	1	5	1	5
Channels	32	64	64	64	32	64	64	64+32	64	64+32	128	1
	L4				L8							

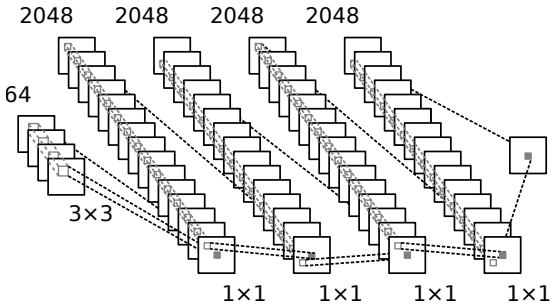


Figure 6.9: L5 architecture compared to L4 or L8 has much wider layers, number of filters to process the input of 64 channels.

the loss of pixels instead of coefficients. All the L5 architectures are trained using the residual objective. The L5 model is based on the convolutional layers followed by the trainable PReLU [59].

6.2.2 *Data*

The majority of the experiments were computed on images from *BSDS500* [90] and *LIVE1* [22] datasets. The L4 and L8 networks were trained solely on the merged train and validation part of BSDS500 which contains 400 images. The L5 training was based on the *INRIA* holidays dataset [101] where the included images were downsampled to correspond the size of images from the other datasets and to suppress the already occurring JPEG artifacts in the original ground truth data.

The images were transformed, as was stated earlier, to the grayscale representation using the $Y'C_BC_R$ color model keeping the luma Y' component only. Only the grayscale images were considered because the attention was solely focused on the ringing and blocking artifacts while the chromatic distortions were left out. Although, the networks can process color images in a case they are trained for them as well. The grayscale images were compressed with the MATLAB JPEG encoder into five disjoint sets based on the JPEG quality. Specifically, the images were compressed with the quality 10, 20, 40, 50, and 60. The DCT coefficients were extracted and stored together with the related quantization tables.

The networks were evaluated on the test set from BSDS500 which includes 100 high-quality compressed images and on the LIVE1 dataset containing 29 color images of uncompressed BMP format. All the evaluation images were transformed to grayscale the same way as the training images and also compressed using the same encoder. It is important to use the same encoder because the quantization tables may differ between different encoder implementations.

6.2.3 *Training*

The training of presented models differs according to the objective, architecture, and data. The formerly presented L4 and L8 networks were trained the same way except for the several objective experiments which were evaluated with L4 archi-

tecture only. The L5 based models differ already in the solver itself. Namely L5 used the Adam solver instead of SGD with momentum.

The importance of the network initialization has been formerly emphasized in several publications [56, 57, 58]. In this work, the assumption of zero mean of the network initialization is recognized as helpful as it prevents mean offsets of activations to propagate through the layers. In case the mean was not zero, any mean offset in input values would result in the non-zero mean of output activations which could force the ReLU non-linearities to get fully stuck either in the positive linear interval or, even worse, in the negative interval where gradients are not propagated rendering the unit useless.

This problem is eliminated by explicitly forcing individual filters to have zero mean during initialization. Such initialization allows to use significantly higher initial learning rates, especially together with residual learning, and it results in trained networks with significantly fewer saturated neurons. The L4 and L8 based models were initialized using the Xavier approach (6.2) and shift to have the zero mean per filter.

All the filters can be forced to have zero mean during the whole training. Such constraint almost entirely eliminates any potential for unit saturation, but it prevents networks to utilize the DC component of input signals. Although reasonably good results were achieved with this constraint in the preliminary experiments, it was not decided to use the offset suppression in the presented experiments. The L4 and L8 based models were trained using the SGD with the momentum with the minibatch of 64 64×64 px patches and 4 128×128 px patches respectively. Solver related parameters are collected in Table 6.3. The patches were randomly sampled from the training images.

In all the experiments, the loss was normalized by the number of output pixels

$$\frac{1}{N \times x_w x_h x_{ch}} \sum_{i=1}^{N} \left\| F\left(\mathcal{W}, y_i\right) - x_i \right\|_2^2 , \tag{6.5}$$

where y_w is the output patch width, y_h the height and y_{ch} number of channels. Such scaling influences the scale of gradients and results in some cases in relatively high learning rates and low weight decay parameters. The number of L4, L8, and

Table 6.3: L4, L8 and L5 training parameters including solver type, learning rate (lr), momentum (m), and weight decay (wd).

Network	solver	lr	m	wd
L4 Direct	SGD	0.4	0.97	5×10^{-7}
L4 Residual	SGD	8	0.97	5×10^{-7}
L4 Edge enh.	SGD	0.05	0.97	5×10^{-4}
L8 Skip arch.	SGD	4	0.95	5×10^{-7}

Network	solver	lr	β_1	β_2	ϵ	wd
L5	ADAM	5×10^{-4}	0.9	0.999	10^{-8}	0

L5 training iterations was fixed to 250 k which is significantly less compared to AR-CNN's 10^7 iterations.

The L5 based models were trained based on the residual objective and using the *ADAM* solver [69]. The specific solver parameters are given in Table 6.3. The learning rate was five times decreased by the factor of 3. The L5 models were all equally trained using 250 k iterations, where the minibatch per iteration consisted of 24 samples. The models were initialized per layer with the Gaussian distribution with zero mean and the standard deviation equal to 10^{-1} for the first layer, 10^{-2} for all the middle layers, and 0.5×10^{-2} for the last 5th layer.

6.2.4 *Artifacts Removal Quality*

The results are compared to AR-CNN [37], to the widely regarded deblocking oriented SA-DCT [6, 7], and to a simple postprocessing filter SPP included in the FFmpeg framework [23]. While L4 architecture was used in most experiments and it was trained for various compression quality levels, L8 was trained only for JPEG quality 20. If not stated otherwise, the residual version of networks was used. The results of L5 are included with the note that it was trained on the INRIA Holiday dataset instead of BSDS500 used for L4 and L8.

The evaluation of removing the artifacts on LIVE1 dataset with JPEG quality 10 and 20 is presented in Table 6.4. The results achieved on BSDS500 test dataset are written in Table 6.5. L8 model outperforms all the other methods with significantly higher scores in all three quality metrics with the exception on BSDS500 test dataset, where the L5, trained completely on different data, achieved a higher SSIM result. Although L4 model performs worse compared to L8, it still surpasses the other methods in most cases even though it is much smaller and computationally efficient compared to both L8 and L5. Interestingly, the L5 performance is between the L4 and L8 having good results based on the SSIM meanwhile surprisingly worse on the B-PSNR. Examples of resulting images are presented in Figure 6.14. There are still visible blocking artifacts of L4 and L8 models trained with residual objective while the L5 model with the worse results based on the PSNR metric seems to restore such a type of artifact very well. That is seen on the monotonic parts of the image like for example the sky.

Table 6.4: Image restoration quality on LIVE1 test dataset for JPEG quality 10 and 20.

	Q10			Q20		
method	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
distorted	27.77	25.33	0.791	30.07	27.57	0.868
spp	28.37	27.77	0.806	30.49	29.22	0.877
SA-DCT	28.65	28.01	0.809	30.81	29.82	0.878
AR-CNN	28.98	28.70	0.822	31.29	30.76	0.887
L4 Residual	**29.08**	**28.71**	**0.824**	31.42	30.83	0.890
L5 Pixel	–	–	–	31.42	30.63	0.890
L8 Residual	–	–	–	**31.51**	**30.92**	**0.891**

Table 6.5: Image restoration quality on BSDS500 test dataset for JPEG quality 10 and 20.

	Q10			Q20		
method	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
distorted	27.58	24.97	0.769	29.72	26.97	0.852
spp	28.13	27.49	0.782	30.11	28.68	0.859
AR-CNN	28.74	**28.38**	0.796	30.80	30.08	0.868
L4 Residual	**28.75**	28.29	**0.800**	30.90	30.13	0.871
L5 Pixel	–	–	–	30.94	29.91	**0.873**
L8 Residual	–	–	–	**30.99**	**30.19**	0.872

JPEG QUALITY GENERALIZATION The attention was focused on the generalization ability of the trained networks regard to a different compression quality. The ability of CNNs to handle various compression qualities is assessed by the experiment which consisted of training the single L4 model for one particular quality and consequently evaluating such a model on all the other qualities. The results in Figure 6.10 show that L4 trained on a range of qualities, from Q10 up to Q60, provides stable results across the equal quality range. However, the quality-specific networks perform better for their respective qualities which yield to a possibility to train the high specialized models in case of the quality of degraded images is known. On the other hand, the quality-specific networks generalize only to similar qualities. In practice, a single network should easily be able to handle smaller quality ranges, e. g. from 10 up to 20 quality points wide, when trained on data from such a range.

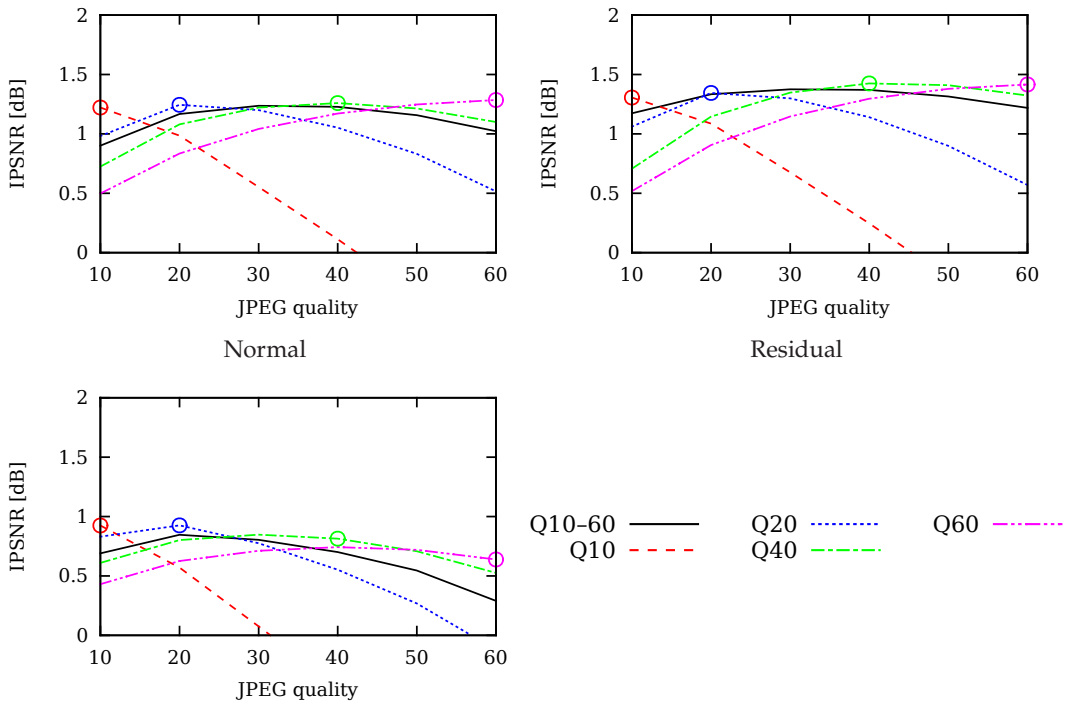


Figure 6.10: Generalization ability of L4 networks trained with normal, residual, and edge preserving objectives for different JPEG quality levels.

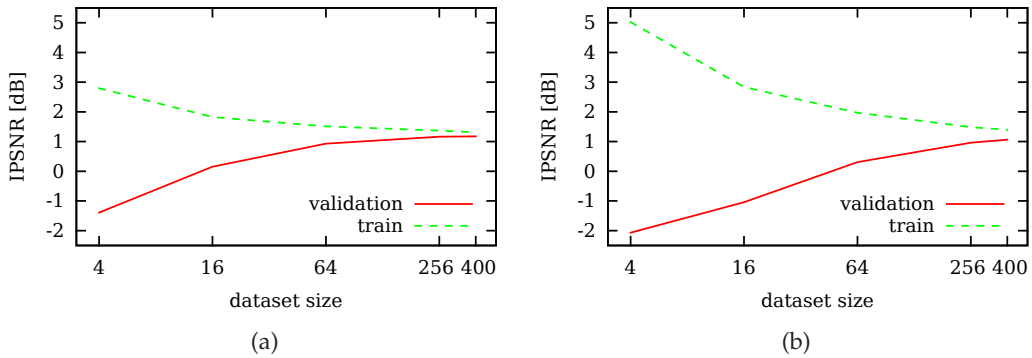


Figure 6.11: Generalization for different sized train set.

DATASET SIZE The quality of restoration achieved by larger networks may suffer due to inadequate size of a training set. In order to assess how the L4 and L8 models behave with respect to training dataset size, the residual versions of the networks were trained on 4, 16, 64, 256, and 400 images from BSDS500 training set. The L4 and L8 models contain approx 70 k and 220 k learnable parameters respectively which suggest that L8 model should require a larger training set for the same generalization. Figure 6.11 shows results of models trained and evaluated on differently sized training datasets together with the evaluation on the corresponding independent test dataset. Both networks clearly overfit on the smaller datasets. L8 model overfits significantly more, and it would require more images to reach proper generalization, while L4 seems to reach its maximal generalization already on the relatively small dataset of 400 images.

6.2.5 *Impact of The Objective*

All the L4 models were trained for direct mapping, residual, and edge enhancement objectives to evaluate the contribution of each. Although the architecture and initialization of all the L4 networks were the same, the suitable learning rates (lr) and weight decay coefficients (wd) had to be selected based on the parameters grid search for each learning objective separately. The solver parameters are noticed in Table 6.3. All the parameters were selected regarding JPEG quality 10 and they were used for all the other qualities as well.

The learning progress is shown in Figure 6.12. The residual network converges much faster compared to the both direct and edge enhancement objectives. The results on LIVE1 based on PSNR, PSNR-B and SSIM metrics are presented in Table 6.6. The results show that the residual based model converges faster and achieves the best restoration quality compared to other objectives. The edge enhancement objective converges a slightly faster in the beginning but stops to develop quite soon letting the direct mapping to overcome its results. It could be expected that the direct objective-based training may achieve a similar restoration quality compared with the residual objective-based training with a clear disadvantage in the form of time needed to converge.

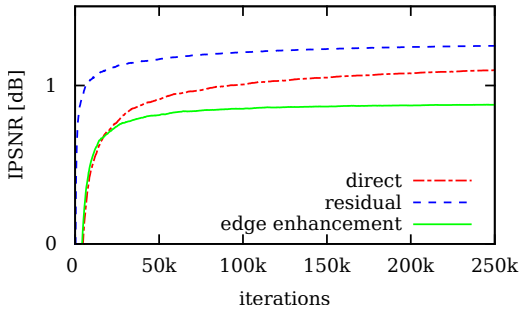


Figure 6.12: Development of L4 with different training objectives.

The progress of training the filters of the first layer during training in different objective based models is shown in Figure 6.13. All the networks formed reasonable-looking filters. The residual objective trained model formed more complex higher frequency filters compared to the other networks. The edge preserving network learned some low-pass filters which are probably needed to transfer the general image appearance through the network. These filters are missing in the residual objective trained model. The filters of the direct objective trained model remain noisy, which could be due to different weight decay coefficient the low learning rate, or their combination. It also implies that the direct mapping would get slightly better results if trained for more iterations which are indicated regarding the IPSNR shown in Figure 6.12.

The results indicate that the residual learning is beneficial for JPEG artifact removal regarding restoration quality and training speed. On the other hand, the edge preserving objective does not improve the quality as is shown in the case of L4.

DCT COEFFICIENTS DCT coefficient based restoration was computed using the L5 architecture with an atypical layers width providing much more filters per layer compared to the L4 or L8 models. The L5 can not be compared directly to both L4 and L8 pixel based models because L5 models were trained on the different training set, the INRIA Holiday [101]. The input data were normalized by a single fixed value to be approximately in the interval from -1 up to 1. The L5 models operating with quantized DCT coefficients – $\mathcal{B}$, JPEG dequantized coefficients – $Q\mathcal{B}$, and directly with pixels were evaluated with the results presented in Table 6.7.

The training dataset was later on augmented by shifting the images by a uniformly sampled shift size in the range from 0 up to 7 pixels in both directions.

Table 6.6: Results of L4 networks with different objectives on LIVE1 dataset with quality 10.

Objective	PSNR	PSNR-B	SSIM
Distorted	27.58	24.97	0.769
Direct mapping	28.99	28.66	0.820
Edge preserving	28.69	28.40	0.813
Residual learn.	**29.08**	**28.71**	**0.824**

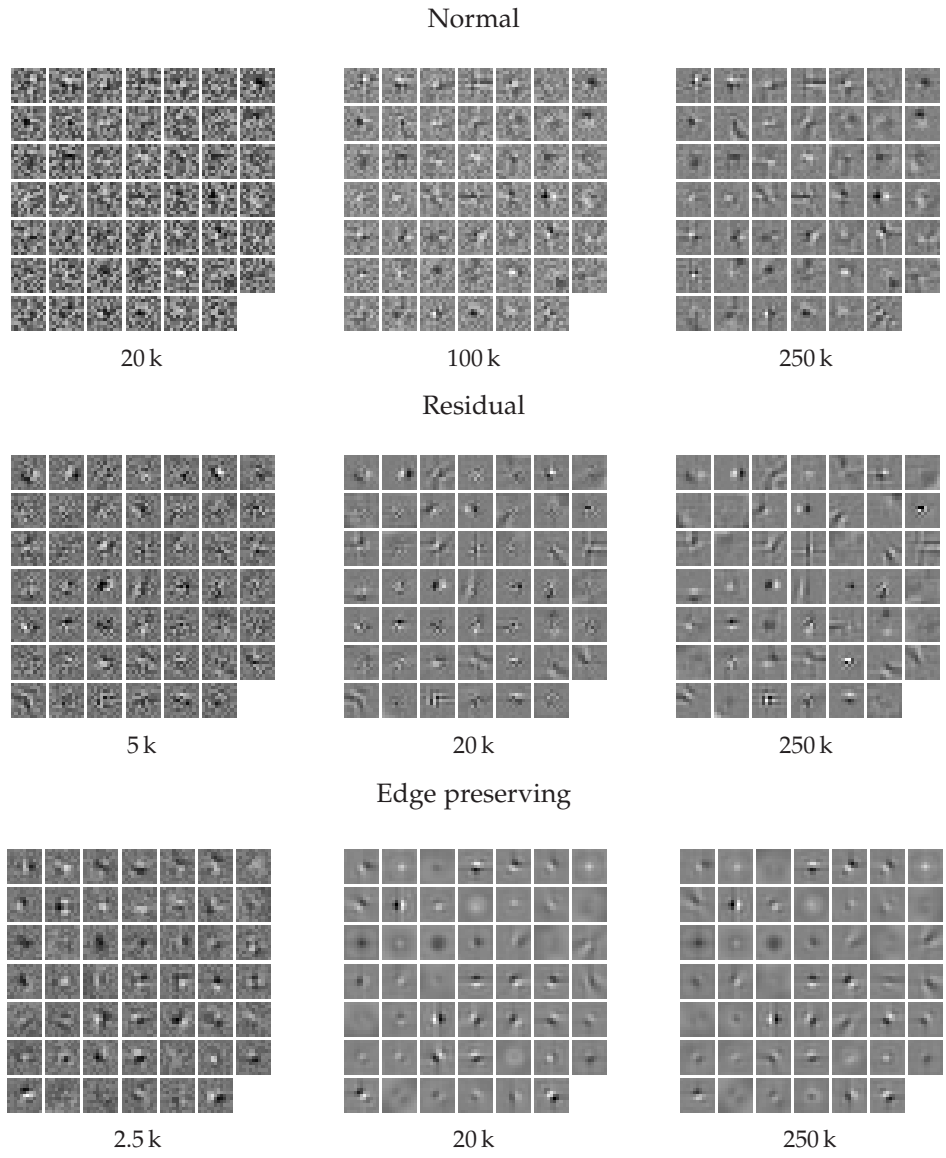


Figure 6.13: Filters from the first layer of L4 networks with normal/residual/edge preserving objective at different stages of training. Iterations are showed below the images.

These shifted original images were then encoded into the JPEG. 64× more data became available leaving the blocking artifact in the same position within the image. The L5 pixel–pixel mapping model trained on the augmented dataset achieved 16% higher IPSNR compared with the same model trained on the original smaller amount if training data but with the same amount of training iterations. Despite the lower achieved PSNR compared to the L8 network, the result images are blocking free while both L4 and L8 models, unfortunately, preserve surpassed still visible blocking artifacts.

The different type data based L5 models, coefficients, dequantized coefficients and pixels, show very similar results, where the differences are apparently related to the model initialization. The exception is the case in which the JPEG DCT coefficients $\mathcal{B}$ are multiplied by the quantization table Q. The results of L5 model operating with such data are slightly better compared to other L5 models. It is appar-

Table 6.7: The different input data and loss function based L5 architecture results. The structure of the model name describes the settings, i. e. the input data and the loss-computed-data. The $\mathcal{B}$ is the JPEG quantized DCT coefficient, $Q\mathcal{B}$ is the $\mathcal{B}$ multiplied by the quantization table, pix stands for pixel data. L5 $\mathcal{B}$–pix represents the L5 model with the JPEG DCT quantized coefficient input data and the loss computed on pixels.

	LIVE1			BSDS500		
method	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
distorted	30.07	27.57	0.868	29.72	26.97	0.852
L5 $\mathcal{B}$–$\mathcal{B}$	31.25	30.51	**0.890**	30.81	29.82	0.871
L5 $Q\mathcal{B}$–$Q\mathcal{B}$	**31.31**	**30.52**	**0.890**	**30.85**	**29.84**	**0.872**
L5 $\mathcal{B}$–pix	31.25	30.51	**0.890**	30.81	29.82	0.871
L5 pix–pix	31.23	30.49	0.889	30.78	29.81	0.871
L5 pix–pix C-PSNR	31.44	30.63	0.892	30.94	29.92	0.873
L5 pix–pix aug	31.42	30.63	0.892	30.94	29.90	0.873

ent that the DCT based restoration models can be successfully deployed without the requirement of any post-processing of the decoded JPEG image. That allows keeping the existing decoders and just use the networks in a preprocessing step being similar to JPEG Quality Transcoder (JQT) approach. The blocking artifacts are well removed by models operating with the resampled input pixels from the 8×8 blocks into the 64 channel vectors. Regarding the results, it is highly probable that such resampled input data explicitly helps the model to train focus on the blocking fixed size and periodicity.

6.2.6 *Summary on Artifacts Restoration*

The CNN based models, namely L4, L5, and L8 were presented. All three outperformed state of the art with most significant results achieved by the L8 model based on the residual training and skip architecture. The residual objective proved to be appropriate for JPEG artifacts restoration and allowed to train the model faster regard to the number of iterations and achieved results. However, the edge enhancement objective did not show any benefits compared to the direct mapping which would provide any reason to prioritize such a learning objective. The importance of the dataset size in regard to the model capacity showed both experiments, the observed L4 and L8 models trained on several dataset sizes and the L5 model trained on the 64× augmented training dataset which provides more than 16% of IPSNR increase compared to the same model on the original training dataset size. The CNN based models ability to generalize was investigated with the results showing the single model covering a wide range of compression qualities with restoration level. However, the specialized model for specific quality can deliver slightly better results measured by the PSNR metric. The experiments provided support for the JQT which transforms the low-quality JPEG coefficients to the coefficients representing higher quality restored image. The input image blocks

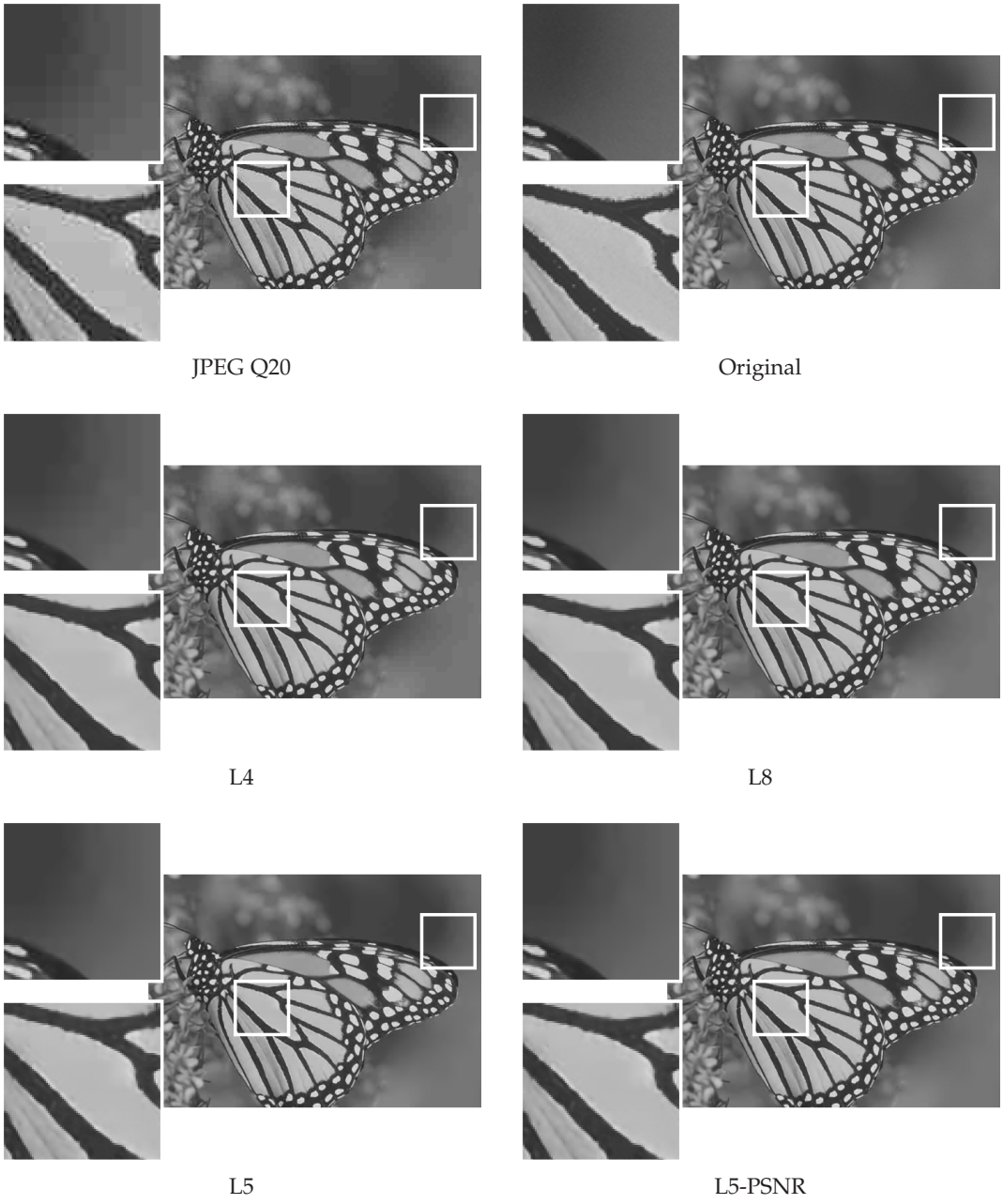


Figure 6.14: Visual comparison of restored monarch image from LIVE1 [22] dataset originally compressed with JPEG quality 20. L4 restores the ringing artifacts but the blocking is to a certain extent preserved. L8-skip compared to L4 provides obviously better results restoring the blocking artifacts. Note, that both L5 smooth the blocking artifacts but performs slightly worse on the ringing artifacts compared to L8 and L4 as well.

resampled from 8×8 spatial size into the 64 channel vectors provided the L5 architecture with the subsidiary information feasible to compute high quality blocking artifacts restoration.

The pixel based architectures, L4 and L8, are with their 70 k and 220 k weight parameters significantly smaller compared to the motion deblurring L15 model with 2.3 M weights. Using cuDNN[3] v3 implementation of convolutions on GeForce GTX

[3] Nvidia GPU-accelerated library of primitives for deep neural networks.

780, the 1 Mpx image takes approximately 220 ms with network L4 and roughly 1052 ms with L8 to be restored. The L4 and L8 networks require approximately 140 k and 440 k floating point operations per pixel.

6.3 SUMMARY OF CONTRIBUTIONS

The core of this thesis is framed by a unified method of image restoration based on Convolutional Neural Network. The data-driven approach has been deployed for particular tasks of image restoration. It is the deconvolution, namely the motion deblurring, which is well described and where the hard part is to estimate the unknown blur parameters such as the length and direction. Further, it is the compression artifacts removal task, which instead of deconvolution restores an image by suppressing the artificial boxing and ghost edges of ringing artifacts. In both cases, i. e. license plate motion deblurring and JPEG artifacts removal, the presented CNN provides beyond state-of-the-art results. Compared to the engineered methods, which significantly differ from each other according to the task they focus on, the CNN approach allows to quickly train a specialized or a universal model solely dependent on the training data. The case of a specific model is related to a limited range of parameters the degradation can be modeled with, e. g. the limited range of lengths the motion blur can consist of. In contrary, the universal model can be used for various levels of particular degradation. That is the case of the single model used for an arbitrary range of motion blur lengths and directions. *Considering the results presented in this thesis, the hypothesis is fulfilled.*

This work extends the approach of text deblurring based on CNN introduced by Hradis et al. [35], which is based on the 15 layer CNN model trained purely on artificially blurred data. This model performs well for various ranges of motions blur lengths and directions. The results on artificially blurred data show model ability to recover an arbitrary range of blur parameters. Simultaneously, the end-to-end model easily outperforms the blind deconvolution L0-regularized method and competes very well compared to the non-blind variation of the same text image specialized L0-regularized method. Further, the L15 model can restore the naturally blurred images as well. Based on the OCR accuracy, L15 CNN model delivers significantly better results compared to L0-regularized method which is considered to be state of the art. The motion deblurring based on CNN reveals how simple it is to obtain a model with the beyond state-of-the-arts outcomes. Model, which generalizes very well and which can handle a wide range of possible blur parameters.

The CNN approach for image compression artifacts restoration presented in this work significantly improves the-state-of-the-art results. Similarly to the L15 model for license plate motion deblurring, the introduced models besides the beyond state-of-the-art results provide a significant generalization ability over various JPEG compression qualities. The analysis of the architectures and the objectives the networks are trained for is given. The residual objective used for artifacts restoration is presented allowing to speed up the training process together with

better outcomes compared to the direct objective. The experiments pointed out the contribution of input data reorganization referring to the deblocking. The work shows that the CNN model used for image restoration in the pixel domain is suitable for transforming the highly compressed JPEG coefficients to the coefficients representing the image, which decoded, becomes artifacts free. The JPEG compression artifacts removal supports the idea of a unified approach to image restoration. There are many others tasks of image restorations. Nevertheless, even these are not reviewed in this thesis, here presented results indicates a possible performance increase in the sense of accuracy and quality based on the data-driven CNN models.

6.4 FUTURE WORK

The combination of all described approaches including the skip architecture, residual objective with further relatively smaller kernel stacking, e. g. like the inception network [46, 51], may provide the results yet far beyond state of the art. Unfortunately, the amount of computational time is directly proportional to the model complexity. Therefore, the recently used architectures take several days to train which makes the exhaustive architecture state space search quite difficult.

The image restoration CNN based models were and yet significantly are influenced by the computer vision research. Based on results in computer vision, the next steps shall lead to architectures of stacked filters comprising the model build from relatively small kernels interleaved with a higher amount of non-linearities, like ReLU, PReLU as used in the L5 models, or recently introduced ELU. Further, classification instead of regression may provide the network with a much easier problem to learn, i. e. the output would be one of 256 possible values representing the image intensity. In such a case an ensemble of models in a form as presented in [HKSS14] or just utilizing the dropout in a network can be simply used to achieve better results.

In the case of JPEG artifacts restoration, the transposed convolution can offer interesting outcomes. It is worth considering networks utilizing the transposed convolution – sometimes noted as deconvolution which spatially scatters the data. That includes various scenarios like deconvolution, in the beginning, gradually stacked deconvolution, and deconvolution at the end of the network. The deconvolution, precisely transposed convolution, is understood as the reverse convolution where the single input value, the result of a convolution, is partially distributed to its source values Figure 3.5. Here, the possible future research regarding JPEG DCT coefficients is likely to provide interesting results.

Although the PSNR based objective did not directly show any significant benefit over the simple MSE loss function, the SSIM loss function is worth a try. The related idea of inpainting the corrupted image to obtain the perceptually plausible image could be used in situations where the scene fidelity is not necessary. Apart from the restoration tasks, the CNN can be deployed in other image processing challenges including in robotics often used visual based parameters estimation. These may include the image matching for the loop closer detection extending the

work [IPSS16] in the mapping and environment reconstruction applications, the rotation-translate estimation between several consequent images, the scene segmentation, the depth from an image estimation, several sensors fusion [SZ10], or descriptors learning [SS11].

Plethora of degradations and corruptions types exist, where the CNN utilization may improve the restoration results compared to the engineered methods, e. g. the whole family of deconvolution methods. In this thesis, the deconvolution CNN is utilized for license plate images deblurring, which is a very narrow image domain compared to the natural images. The end-to-end mapping for such tasks may be much too hard for recent network models. Nevertheless, no such known research has been yet done in this field. A regression CNN models introduced to compute the image restoration are likely to be suitable for similar tasks related to inpainting. An inpainting model can be used to estimate the shape and a texture of partially occluded objects in an image or generate details which may provide better perceptual image quality. An interesting approach to image generation is based on adversarial networks, where the generator network tries to fool the discriminator network with generated images instead of real images. Last but not least, the inpainting may by used for several objects anonymization including the human faces, license plates, advertisements and generally anything in the image.

7

CONCLUSION

This thesis focuses on an image restoration based on models of convolutional neural networks. Particularly, two different tasks were chosen, motion deblurring of license plate images taken by a surveillance system and artifacts removal caused by low quality of JPEG compression. Usually, the methods of image restoration are hand-engineered. That yields to a variety of approaches which are comprised of certain processing pipelines related to a type of degradation. Specifically, in motion deblurring, the pipeline consists of PSF estimation and a subsequent deconvolution to restore the latent sharp image. Compression artifacts restoration methods try to smooth the discontinuities made by blocking or suppress ringing on edges.

In this work, in contrary, a direct end-to-end mapping based on convolutional neural networks is presented. Restoration relies on a data-driven trained model which directly transforms a degraded image to an undistorted image. Recently introduced convolutional neural network architecture, i. e. AlexNet, inspired a model deployed for license plate motion deblurring. Several experiments show that a single model is sufficient for various motion blurs differing in lengths and directions which allow the comparison with blind deconvolution methods. The model trained solely on artificially blurred data outperforms the considered state-of-the-art method deployed on naturally blurred images where the achieved OCR based error accuracy is 9 % compared to 23 % error accuracy of L0-regularized method.

Further, a nonlinear degradation based on the JPEG compression is restored exploiting the same end-to-end approach of data-driven trained models. Compared to motion deblurring, restoration of compression artifacts is a harder problem due to the missing image information. While the approach is the same, various training and architecture related extensions are introduced including the residual objective, skip architecture, and loss computed on an image data in a case of JPEG coefficient transformation. These extensions contribute to train model which achieved in artifacts suppression state-of-the-art results. Particularly, L8 model achieved 31.51 PSNR compared to 31.29 PSNR of recently introduced AR-CNN and 30.81 PSNR of hand-engineered SA-DCT. In the case of JPEG artifacts restoration, direct transformation of JPEG coefficients based on the convolutional network is proposed. Such transformed coefficients allow restoring the artifacts degraded image before decoding itself.

The results achieved in both tasks contribute to the idea of utilizing CNNs as a unified approach to image restoration. It is worth to try to follow the ongoing research in computer vision, where the majority of CNN related trends come from. That includes stacking the spatially small filters interleaved by more nonlinearities providing even better models. An interesting yet challenging deblurring of natural

images should be investigated further. In the case of JPEG coefficients transformation, the fixed IDCT layer can be substituted by the trained transposed convolution allowing the network to adapt the decoding step. Considering the fact that all the presented models are regression based CNN, they can be therefore deployed for a task of inpainting as well. That would allow restoring incomplete data in an image, i. e. occluded objects or simply too much-degraded image regions. The impact of CNN models in various research domains is high. There is a lot of other applications the deployment of data-driven models is worth to try.

This thesis begins with an introduction reminding a year the research on NN is considered to begin. After more than 70 years later the actual state of the art of CNN dynamically evolves providing a significant impact in various domains including the image restoration as well.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

[HKSS14] M. Hradis, M. Kolar, P. Svoboda, and P. Smrz. Large scale image classification by Brno University of Technology. Poster at ILSVRC 2014 in conjunction with ECCV 2014, September 2014.

[IPSS16] V. Ila, L. Polok, M. Solony, and P. Svoboda. Slam++. A highly efficient and temporally scalable incremental SLAM framework. *Intl. J. of Robotics Research*, 2016(123):1–22, 2016.

[SHBZ16] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik. Compression artifacts removal using convolutional neural networks. *J. of WSCG*, 24(2):63–72, 2016.

[SHMZ16] P. Svoboda, M. Hradis, L. Marsik, and P. Zemcik. CNN for license plate motion deblurring. In *Intl. Conf. on Image Processing (ICIP)*, pages 1–4. IEEE Signal Processing Society, 2016.

[SS11] P. Schaffroth and P. Svoboda. Fast corner point detection through machine learning. In *Proc. of the 17th Conf. STUDENT EEICT 2011*, Volume 3, pages 537–541, 2011.

[SZ10] P. Svoboda and P. Zemcik. Applications of LIDAR and camera fusion. In *Proc. of the DT workshop*, pages 105–106, 2010.

BIBLIOGRAPHY

[1] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems (NIPS)*, volume 2, pages 396–404. Morgan-Kaufmann, 1990.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.

[3] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thompson Learning, 3 edition, 2008. ISBN 9780495082521.

[4] S. G. Mallat. *A wavelet tour of signal processing: the sparse way*. Elsevier Inc., 3 edition, 2008. ISBN 9780080922027.

[5] J. Pan, Z. Hu, Z. Su, and M. H. Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2908, June 2014. doi: 10.1109/CVPR.2014.371.

[6] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise Shape-Adaptive DCT for high-quality deblocking of compressed color images. In *Proc. of the European Signal Processing Conf.*, September 2006.

[7] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise Shape-Adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Trans. Image Processing*, 16(5):1395–1411, May 2007.

[8] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Process. Mag.*, 14(2):24–41, March 1997. ISSN 1053-5888. doi: 10.1109/79.581363.

[9] J. Jan. *Medical Image Processing, Reconstruction and Restoration: Concepts and Methods*. CRC Press, 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742, 2006. ISBN 9780824758493.

[10] F. Krahmer, Y. Lin, B. Mcadoo, K. Ott, J. Wang, and D. Widemann. Blind image deconvolution: Motion blur estimation. Technical report, Institute for Mathematics and its Applications, 2006.

[11] J. P. Oliveira, M. A. T. Figueiredo, and J. M. Bioucas-Dias. Blind estimation of motion blur parameters for image deconvolution. *Pattern Recognition and Image Analysis, Lecture Notes in Computer Sci.*, 4478:604–611, 2007. ISSN 1556-4967. doi: 10.1007/978-3-540-72849-8_76.

[12] M. Ebrahimi-Moghaddam and M. Jamzad. Motion blur identification in noisy images using mathematical models and statistical measures. *Pattern Recognition*, 40(7): 1946–1957, 2007. ISSN 00313203. doi: 10.1016/j.patcog.2006.11.022.

[13] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008. ISSN 1063-6919. doi: 10.1109/CVPR.2008.4587834.

[14] A. Levin. Blind motion deblurring using image statistics. *Advances in Neural Information Processing Systems (NIPS)*, 19(3):841–848, 2007. ISSN 1049-5258. doi: 10.1145/1239451.1239521.

[15] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007. ISBN 978-1-4244-1630-1. doi: 10.1109/ICCV.2007.4408904.

[16] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *Intl. J. of Computer Vision*, 98(2):168–186, 2012. ISSN 09205691. doi: 10.1007/s11263-011-0502-7.

[17] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–8, 2011.

[18] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding blind deconvolution algorithms. *IEEE Trans. Pattern Anal. Machine Intell.*, 33(12):2354–2367, 2011. ISSN 01628828. doi: 10.1109/TPAMI.2011.148.

[19] ITU. Recommendation T.81, 1993.

[20] E. Hamilton. JPEG File Interchange Format, September 1992.

[21] Technical Standardization Committee on AV & IT Storage Systems and Equipment. Exchangeable image file format for digital still cameras: Exif version 2.2, April 2002.

[22] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. LIVE image quality assessment database release 2, 2015.

[23] A. Nosratinia. Embedded post-processing for enhancement of compressed images. In *Proc. Data Compression Conference (DCC)*, pages 62–71, March 1999. doi: 10.1109/DCC.1999.755655.

[24] T. S. Wong, C. A. Bouman, I. Pollak, and Z. Fan. A document image model and estimation algorithm for optimized jpeg decompression. *IEEE Trans. Image Processing*, 18(11):2518–2535, November 2009. ISSN 1057-7149. doi: 10.1109/TIP.2009.2028252.

[25] S. Yang, S. Kittitornkun, Y. Hu, T. Q. Nguyen, and D. L. Tull. Blocking artifact free inverse discrete cosine transform. In *Intl. Conf. on Image Processing (ICIP)*, volume 3, pages 869–872, September 2000. doi: 10.1109/ICIP.2000.899594.

[26] W. S. McCulloch and W. Pitts. A logical calculus of the idea immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[27] F. Rosenblat. The perceptron – a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical laboratory, Inc., January 1957.

[28] F. Rosenblatt. *Principles of neurodynamics; perceptrons and the theory of brain mechanisms.* Spartan Books, Woshington. D.C., USA, 1962.

[29] B. Widrow. An adaptive 'Adaline' neuron using chemical 'memistors'. Technical report, Stanford Electronics Laboratories, October 1960.

[30] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard Univ., 1974.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[32] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 0028-0836.

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[34] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 1520–1528, 2015. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.178.

[35] M. Hradis, J. Kotera, P. Zemcik, and F. Sroubek. Convolutional neural networks for direct text deblurring. In *British Machine Vision Conf. (BMVC)*. The British Machine Vision Association and Society for Pattern Recognition, 2015. ISBN 1-901725-53-7.

[36] Ch. Dong, Ch. Ch. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Eur. Conf. on Computer Vision (ECCV)*, Lecture Notes in Computer Sci., pages 184–199. Springer International Publishing, September 2014. ISBN 978-3-319-10593-2. doi: 10.1007/978-3-319-10593-2_13.

[37] Ch. Dong, Y. Deng, Ch. Loy Change, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *Intl. Conf. on Computer Vision (ICCV)*, pages 576–584, December 2015.

[38] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2016. ISBN 978-0-387-31073-2.

[39] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition — tangent distance and tangent propagation. In O. B. Genevieve and K. R. Muller, editors, *Neural Networks : Tricks of the Trade, Lecture Notes in Computer Sci.*, pages 239–274, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-49430-0. doi: 10.1007/3-540-49430-8_13.

[40] K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6): 455–469, 1982. ISSN 00313203. doi: 10.1016/0031-3203(82)90024-3.

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet Large Scale Visual Recognition Challenge. *Intl. J. of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[42] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 15:315–323, 2011. ISSN 15324435. doi: 10.1.1.208.6449.

[43] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, pages 1–18, 2012. ISSN 9781467394673. doi: arXiv:1207.0580.

[44] N. Srivastava, G. E. Hinton, A. Krizhevsky, Y. Sutskever, and R. Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *J. of Machine Learning Research*, 15:1929–1958, 2014. ISSN 15337928. doi: 10.1214/12-AOS1000.

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[46] Ch. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[47] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *XoRR*, 7(3):171–180, December 2015. ISSN 1664-1078. doi: 10.3389/fpsyg.2013.00124.

[48] K. Chellapilla, S. Puri, and P. Simard. High performance convolutional neural networks for document processing. *Intl. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2006.

[49] S. Chetlur and C. Woolley. cuDNN: Efficient Primitives for Deep Learning. *XoRR*, pages 1–9, 2014.

[50] J. S. J. Ren and L. Xu. On vectorization of deep convolutional neural networks for vision tasks. *Nat. Conf. on Artificial Intelligence (AAAI)*, pages 1840–1846, 2015.

[51] Ch. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *XoRR*, 2015. ISSN 08866236. doi: 10.1002/2014GB005021.

[52] L. Bottou. Stochastic gradient descent tricks. In *Neural Networks : Tricks of the Trade, Lecture Notes in Computer Sci.*, volume 1, pages 421–436, 2012. ISBN 978-3-642-35288-1. doi: 10.1007/978-3-642-35289-8_25.

[53] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Book in preparation for MIT Press, 2016.

[54] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–54, 2006. ISSN 0899-7667. doi: 10.1162/neco.2006.18.7.1527.

[55] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. of Machine Learning Research*, 11(3):3371–3408, 2010. ISSN 15324435. doi: 10.1111/1467-8535.00290.

[56] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks : Tricks of the Trade, Lecture Notes in Computer Sci.*, pages 437–478, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_26.

[57] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 9:249–256, 2010. ISSN 15324435. doi: 10.1.1.207.2059.

[58] Y. LeCun, L. Bottou, G. B. Orr, and K. R. Muller. Efficient backprop. In *Neural Networks : Tricks of the Trade, Lecture Notes in Computer Sci.*, volume 1524, pages 9–50, London, UK, UK, 1998. Springer-Verlag. ISBN 9783540494300. doi: 10.1007/3-540-49430-8.

[59] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–11, 2015. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.123.

[60] M. Dmytro and J. Matas. All you need is a good init. *CoRR*, abs/1511.06422, 2015.

[61] A. G. Baydin and B. A. Pearlmutter. Automatic differentiation of algorithms for machine learning. *Intl. Conf. on Machine Learning (ICML)*, pages 1–7, 2014.

[62] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *XoRR*, 2014.

[63] G. E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Sci.*, 11 (10):428–434, 2007. ISSN 13646613. doi: 10.1016/j.tics.2007.09.004.

[64] R. S. Sutton. Two problems with backpropagation and other steepest-descent learning procedures for networks. In *Proc. 8th annual conf. cognitive science society*, pages 823–831, 1986.

[65] G. B. Orr and T. K. Leen. Momentum and optimal stochastic search. *Proceedings of the 1993 Connectionist Models*, pages 477–484, 1994.

[66] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999. ISSN 08936080. doi: 10.1016/S0893-6080(98)00116-6.

[67] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton. On the importance of initialization and momentum in deep learning. *Intl. Conf. on Machine Learning (ICML)*, 28 (2010):1139–1147, 2013. ISSN 15206149. doi: 10.1109/ICASSP.2013.6639346.

[68] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. Advances in optimizing recurrent networks. *Intl. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, pages 8624–8628, December 2012.

[69] D. Kingma and J. B. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980:1–15, 2014.

[70] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *J. of Machine Learning Research*, volume 12, pages 2121–2159, 2011. ISBN 9780982252925. doi: 10.1109/CDC.2012.6426698.

[71] M. D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, December 2012.

[72] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng. Large scale distributed deep networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1223–1231. Curran Associates, Inc., 2012.

[73] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Intl. Conf. on Machine Learning (ICML)*, pages 807–814. Omnipress, 2010.

[74] D. A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by Exponential Linear Units (ELUs). In *International Conference on Learning Representations (ICLR)*, pages 1–13, 2016.

[75] S. Ioffe and Ch. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Intl. Conf. on Machine Learning (ICML)*, pages 448–456, May 2015.

[76] J. E. Tansley, M. J. Oldfield, and D. J. C. MacKay. Neural network image deconvolution. In Glenn R. Heidbreder, editor, *Fundamental Theories of Physics: Maximum Entropy and Bayesian Methods*, pages 319–325, Dordrecht, 1996. Springer Netherlands. ISBN 978-94-015-8729-7. doi: 10.1007/978-94-015-8729-7_25.

[77] Ch. J. Schuler, Ch. H. Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[78] L. Xu, J. SJ. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 27, pages 1790–1798. Curran Associates, Inc., 2014.

[79] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Intl. Conf. on Computer Vision (ICCV)*, December 2013.

[80] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[81] Ch. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *IEEE Trans. Pattern Anal. Machine Intell.*, 38(7):1–28, July 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2481418.

[82] A. Chakrabarti. A neural approach to blind motion deblurring. *CoRR*, abs/1603.04771, 2016.

[83] P. Wieschollek, M. Hirsch, H. P. A. Lensch, and B. Schölkopf. End-to-End Learning for Image Burst Deblurring. *CoRR*, abs/1607.04433, 2016.

[84] V. Jain and S. Seung. Natural image denoising with convolutional networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 769–776. Curran Associates, Inc., 2009.

[85] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2392–2399, June 2012. doi: 10.1109/CVPR.2012.6247952.

[86] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 341–349. Curran Associates, Inc., 2012.

[87] K. Cho. Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images. *J. of Machine Learning Research*, 28:432–440, 2013.

[88] J. Lazzaro and J. Wawrzynek. Jpeg quality transcoding using neural networks trained with a perceptual error measure. *Neural Computation*, 11(1):267–296, January 1999. ISSN 0899-7667. doi: 10.1162/089976699300016917.

[89] Y. Zhang, E. Salari, and S. Zhang. Reducing blocking artifacts in JPEG-compressed images using an adaptive neural network-based algorithm. *Neural Computing and Applications*, 22(1):3–10, January 2013. ISSN 0941-0643. doi: 10.1007/s00521-011-0740-1.

[90] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Intl. Conf. on Computer Vision (ICCV)*, volume 2, pages 416–423, July 2001.

[91] M. Browne and S. S. Ghidary. *Convolutional Neural Networks for Image Processing: An Application in Robot Vision*, pages 641–652. Springer Berlin Heidelberg, Berlin, Heidelberg, December 2003. ISBN 978-3-540-24581-0. doi: 10.1007/978-3-540-24581-0_55.

[92] V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung. Supervised learning of image restoration with convolutional networks. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–8, October 2007. doi: 10.1109/ICCV.2007.4408909.

[93] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. *Lecture Notes in Computer Sci.*, 8693:49–64, 2014. ISSN 16113349. doi: 10.1007/978-3-319-10602-1_4.

[94] Ch. Dong, Ch. Ch. Loy, and K. He. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Machine Intell.*, 38(2):1–14, 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2439281.

[95] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. *CoRR*, abs/1511.04587, 2015.

[96] S. Mallat. Understanding deep convolutional networks. *Philosoph. Trans. of the Royal Soc. of London A: Math., Phys. and Eng. Sci.*, 374(2065), 2016. ISSN 1364-503X. doi: 10.1098/rsta.2015.0203.

[97] I. Sobel. History and definition of the so-called Sobel operator, 2014.

[98] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, 2004. ISSN 1057–7149. doi: 10.1109/TIP.2003.819861.

[99] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Process. Mag.*, 26:98–117, January 2009. ISSN 1053-5888. doi: 10.1109/MSP.2008.930649.

[100] Ch. Yim and A. C. Bovik. Quality assessment of deblocked images. *IEEE Trans. Image Processing*, 20(1):88–98, January 2011. ISSN 1057-7149. doi: 10.1109/TIP.2010.2061859.

[101] H. Jegou, M. Douze, and C. Schmid. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *Eur. Conf. on Computer Vision (ECCV)*, pages 304–317. Springer Berlin Heidelberg, Berlin, Heidelberg. doi: 10.1007/978-3-540-88682-2_24.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". It is available for LaTeX via CTAN as `classicthesis`.