

L<sup>A</sup>T<sub>E</sub>X-Vorlage für diverse Ausarbeitungen  
—  
oder so ähnlich

**STUDIENARBEIT**

für die Prüfung zum  
Bachelor of Science  
des Studienganges Informatik  
an der

Dualen Hochschule Baden-Württemberg Karlsruhe

von

**Lukas Hörnle   Marc Gökce**

Abgabedatum 22.05.2023

Bearbeitungszeitraum

Matrikelnummer

Kurs

Ausbildungsfirma

1

Gutachter der Studienakademie

TODO raussuchen

TODO

TINF20B4

CAS Software AG

1

Karlsruhe

Ralph Lausen

## Erklärung

Ich versichere hiermit, dass ich meine Studienarbeit mit dem Thema: » $\text{\LaTeX}$ -Vorlage für diverse Ausarbeitungen

—

oder so ähnlich« selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt. \_\_\_\_\_

\_\_\_\_\_  
Ort      Datum

\_\_\_\_\_  
Unterschrift

*Sofern vom Dualen Partner ein Sperrvermerk gewünscht wird, ist folgende Formulierung zu verwenden:*

## Sperrvermerk

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anderslautende Genehmigung vom Dualen Partner vorliegt.

## **Zusammenfassung**

Dieses L<sup>A</sup>T<sub>E</sub>X-Dokument kann als Vorlage für einen Praxis- oder Projektbericht, eine Studien- oder Bachelorarbeit dienen.

Zusammengestellt von Prof. Dr. Jürgen Vollmer <juergen.vollmer@dhbw-karlsruhe.de>  
<https://www.karlsruhe.dhbw.de>. Die jeweils aktuellste Version dieses L<sup>A</sup>T<sub>E</sub>X-Paketes ist  
immer auf der *FAQ-Seite* des Studiengangs Informatik zu finden: <https://www.karlsruhe.dhbw.de/inf/studienverlauf-organisatorisches.html> → *Formulare und Vorlagen*.

Stand \$Date: 2020/03/13 15:07:45 \$

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
<b>2</b>	<b>Grundlagen der Bildverarbeitung und der Skalierung von Bildern</b>	<b>8</b>
2.1	Einblick in die Bildverarbeitung . . . . .	8
2.2	Skalierung von Bildern . . . . .	8
2.2.1	Arten der Skalierungen: Interpolation und Skalierung . . . . .	8
2.2.2	Bildformate . . . . .	8
2.2.3	Wichtige Aspekte der Skalierung . . . . .	11
2.3	Echtzeitverarbeitung von Bildern . . . . .	12
2.4	Anwendungen von Skalierungsmethoden . . . . .	12
<b>3</b>	<b>Klassische Skalierungsmethoden</b>	<b>14</b>
3.1	Pixel-Verdopplung . . . . .	14
3.2	Nearest-Neighbor-Interpolation . . . . .	16
3.3	Bilineare Interpolation . . . . .	17
3.4	Bicubische Interpolation . . . . .	18
3.4.1	Mathematische Grundlagen . . . . .	18
3.4.2	Algorithmische Implementierung . . . . .	19
3.4.3	Analyse der Leistung . . . . .	19
3.4.4	Mathematische Grundlagen . . . . .	20
3.4.5	Algorithmische Implementierung . . . . .	20
3.4.6	Implementation . . . . .	20
3.4.7	Analyse der Leistung . . . . .	20
3.4.8	Anwendungen . . . . .	21
3.4.9	Erweiterungen und Variationen . . . . .	21
3.5	Lanczos-Interpolation . . . . .	21
3.5.1	Mathematische Grundlage . . . . .	21
3.5.2	Praktische Anwendung . . . . .	22
3.6	Zusammenfassung zu Vor- und Nachteilen von Techniken in der Bildverarbeitung . . . . .	22
3.6.1	Pixelverdopplung . . . . .	23
3.6.2	Nearest Neighbour Interpolation . . . . .	23
3.6.3	Bilineare Interpolation . . . . .	23
3.6.4	Bicubische Interpolation . . . . .	23

3.6.5	Lanczos Interpolation . . . . .	23
3.6.6	Vergleich und Zusammenfassung . . . . .	23
<b>4</b>	<b>Fortgeschrittene Skalierungsmethoden</b>	<b>24</b>
4.1	Convolutional Neural Networks / Deep learning . . . . .	24
4.1.1	Grundlagen von Convolutional Neural Networks (CNNs) . . . . .	24
4.1.2	Architekturen von CNNs . . . . .	24
4.1.3	Anwendungen von CNNs . . . . .	25
4.1.4	Transfer Learning mit CNNs . . . . .	25
4.1.5	Limitationen von CNNs und aktuelle Forschungsrichtungen . . . . .	25
4.2	Super Resolution . . . . .	25
4.2.1	Grundlagen von Super Resolution (SR) . . . . .	25
4.2.2	Super Resolution-Methoden auf Basis von Deep Learning . . . . .	26
4.3	Generative Adversarial Networks (GANs) . . . . .	27
4.3.1	Grundlagen von Generative Adversarial Networks (GANs) . . . . .	27
4.3.2	Architekturen von GANs . . . . .	27
4.3.3	Anwendungen von GANs . . . . .	28
4.3.4	Training von GANs und Evaluierung von generierten Ergebnissen . . . . .	28
4.3.5	Ethische und soziale Implikationen von GANs . . . . .	28
4.4	Multiscale-Skalierung . . . . .	29
4.4.1	Grundlagen von Multiscale-Skalierung . . . . .	29
4.4.2	Methoden zur Multiskalenanalyse . . . . .	29
4.4.3	Methoden zur Multiskalenanalyse . . . . .	30
4.4.4	Anwendungen von Multiscale-Skalierung . . . . .	30
4.4.5	Limitationen und zukünftige Forschungsziele von Multiscale-Skalierung . . . . .	31
4.5	Vor- und Nachteile der fortgeschrittenen Methoden . . . . .	31
<b>5</b>	<b>Evaluation von Skalierungsmethoden</b>	<b>32</b>
5.1	Qualitätsmetriken zur Bewertung von Skalierungsmethoden . . . . .	32
5.1.1	Peak Signal-to-Noise Ratio (PSNR) . . . . .	32
5.1.2	Structural Similarity Index (SSIM) . . . . .	32
5.1.3	Computational Speed . . . . .	32
5.1.4	Root-Mean-Square Error (RMSE) . . . . .	32
5.2	Kriterien zur Wahl der besten Methode . . . . .	32
5.2.1	Mean Opinion Score (MOS) . . . . .	32
5.2.2	Bildvergleich und visuelle Bewertung . . . . .	33
5.2.3	Effektivität und Effizienz . . . . .	33
5.2.4	Zukunftsansichten und Herausforderungen . . . . .	33
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>34</b>
6.1	Auswertung der Ergebnisse . . . . .	34
6.2	Diskussion offener Fragen und zukünftiger Forschungsbedarf . . . . .	34

<b>7</b>	<b>Einführung in Deep Learning</b>	<b>35</b>
7.1	Motivation hinter der Studie von CNNs und deren Training . . . . .	35
7.2	Struktur des Papiers . . . . .	36
<b>8</b>	<b>Basics of Convolutional Neural Networks</b>	<b>37</b>
8.1	How do Machine learning models learn? . . . . .	38
8.1.1	Definition Machine Learning and Deep Learning . . . . .	38
8.1.2	Definition and applications of CNNs . . . . .	38
8.1.3	Applications of CNNs . . . . .	38
8.1.4	How are CNNs different from other neural network architectures? . . . . .	38
8.1.5	Why are CNNs particularly useful for image and video data? . . . . .	38
8.2	Applications of CNNs . . . . .	38
8.2.1	Image classification . . . . .	38
8.2.2	Object detection . . . . .	38
8.2.3	Face recognition . . . . .	38
8.2.4	Natural Language Processing . . . . .	38
8.2.5	Other applications . . . . .	38
8.3	How do CNNs work? . . . . .	38
8.4	Architecture of CNNs . . . . .	38
8.4.1	Input layer . . . . .	38
8.4.2	Hidden layers . . . . .	38
8.4.3	Output layer . . . . .	38
8.5	Convolutional layers . . . . .	38
8.5.1	Pooling layers . . . . .	38
8.5.2	Fully Connected layers . . . . .	38
8.5.3	Activation functions . . . . .	38
8.5.4	Loss functions . . . . .	38
<b>9</b>	<b>Data Preprocessing</b>	<b>39</b>
9.1	What is data? . . . . .	39
9.2	Importance of the right data . . . . .	39
9.2.1	Data cleaning . . . . .	39
9.2.2	Data normalization . . . . .	39
9.2.3	Data augmentation . . . . .	39
<b>10</b>	<b>Convolutional Neural Network Training Process</b>	<b>40</b>
10.1	What is training? . . . . .	40
10.2	Training process . . . . .	40
10.3	Training process for CNNs . . . . .	40
10.4	Stochastic Gradient Descent (SGD) . . . . .	40
10.5	Backpropagation . . . . .	40
10.6	Hyperparameter tuning . . . . .	40
10.7	Regularization techniques . . . . .	40

<b>11 Transfer Learning</b>	<b>41</b>
11.1 Introduction to Transfer Learning . . . . .	41
11.2 Fine-tuning of pre-trained models . . . . .	41
11.3 Using pre-trained models as feature extractors . . . . .	41
<b>12 Common Challenges and Solutions</b>	<b>42</b>
12.1 Overfitting and underfitting . . . . .	42
12.2 Vanishing and exploding gradients . . . . .	42
12.3 Gradient descent optimization . . . . .	42
12.4 Solutions to common challenges . . . . .	42
<b>13 Tools and Frameworks for CNN Training</b>	<b>43</b>
13.1 PyTorch . . . . .	43
13.2 TensorFlow . . . . .	43
13.3 Keras . . . . .	43
13.4 Caffe . . . . .	43
13.5 Other popular frameworks . . . . .	43
<b>14 Conclusion and Future Work</b>	<b>44</b>
14.1 Summary of the paper . . . . .	44
14.2 Key takeaways . . . . .	44
14.3 Future research directions . . . . .	44
<b>Anhang</b>	<b>49</b>
<b>Index</b>	<b>49</b>
<b>Literaturverzeichnis</b>	<b>49</b>
<b>Liste der ToDo's</b>	<b>53</b>

# Kapitel 1

## Einleitung

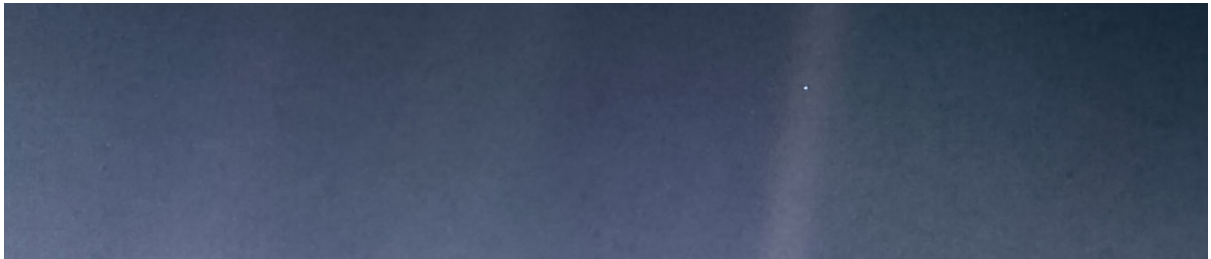


Abbildung 1.1: “The Pale Blue Dot” Feb. 14, 1990, by NASA<sup>1</sup>.

”Ein Bild sagt mehr aus als tausend Worte.“ Dieses bekannte Sprichwort drückt aus, wie mächtig Bilder als Kommunikationsmittel sind. Bilder beinhalten Informationen, vermitteln Emotionen, erzählen Geschichten und sind ein Fenster in die Vergangenheit. Bilder sind in der modernen Gesellschaft omnipräsent und im Alltag digital als auch analog unentbehrlich. Bilder unterscheiden sich je nach Aufnahme in den verschiedenen Eigenschaften ihrer Speicherung und Darstellung. Zwei dieser Eigenschaften sind die Größe und die Auflösung eines Bildes. Die Größe eines digitalen Bildes gibt an, wie viele Pixel es enthält, während die Auflösung eines Bildes angibt, wie viele Pixel pro Flächeneinheit vorhanden sind (Pixel per Inch (PPIs)).

Die Größe und die Auflösung eines Bildes haben Einfluss auf seine Qualität und seinen Speicherplatzbedarf. Um ein Bild für einen bestimmten Zweck zu nutzen, muss es häufig in seiner Größe und oder Auflösung verändert werden. Der Vorgang zur Veränderung

---

<sup>1</sup>NASA/JPL-CALTECH [Feb. 1990]. *The Pale Blue Dot is a photograph of Earth taken Feb. 14, 1990, by NASA’s Voyager 1 at a distance of 6 billion kilometers from the Sun. The image inspired the title of scientist Carl Sagan’s book, “Pale Blue Dot: A Vision of the Human Future in Space,” in which he wrote: “Look again at that dot. That’s here. That’s home. That’s us.”* <https://solarsystem.nasa.gov/resources/536/voyager-1s-pale-blue-dot/>



der Größe und Auflösung wird als Bildskalierung bezeichnet<sup>23</sup> und ist eine grundlegende Operation in der digitalen Bildverarbeitung. Bildskalierung ist eine grundlegende Methode der Bildverarbeitung. Bildskalierung erlaubt die Änderung der Größe eines digitalen Bildes. Eine Gute Bildskalierung misst sich an ihren Eigenschaften in den Bereichen Rechenaufwand und Qualitätsverlust. Besonders wichtig ist der Qualitätsverlust, wenn man Bilder größer skaliert. Ein geeignetes Modell um diesen Prozess zu erklären ist das Übertragen einer Zeichnung von einem kleinen Papier auf eine große Leinwand. Wird die Zeichnung lediglich unbedacht vergrößert, wird diese unscharf und verliert an Details. Das Ziel einer guten Bildskalierung ist es, diesen Effekt zu verhindern und die Zeichnung größenunabhängig scharf und detailreich darzustellen.

Es gibt viele verschiedene Methoden, um die Größe eines Bildes zu ändern. Klassische Methoden verwenden Interpolationstechniken, die neue Pixel aus den vorhandenen Pixeln berechnen. Diese Methoden sind schnell und stellen einen geringen Rechenaufwand in Kombination mit geringer Komplexität dar. Jedoch kommt es mit diesen Algorithmen oft zu Qualitätsverlusten oder der Erzeugung von Artefakten. Moderne Anwendungen zur Skalierung von Bildern verwenden Deep-Learning-Techniken wie Convolutional Neural Networks (CNNs) oder Generative Adversarial Networks (GANs), die neue Pixel aus einem trainierten Modell erzeugen. Diese neuen Methoden sind komplex und benötigen mehr Rechenaufwand, können allerdings die Qualität des Bildes verbessern oder kreative Effekte erstellen.

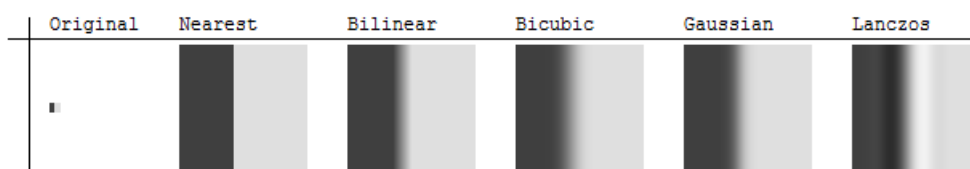


Abbildung 1.2: Verschiedene Beispiele von upscaling Algorithmen[WHUBER 2011].

Die Bildskalierung hat heute viele Anwendungen in verschiedenen Bereichen wie beispielsweise Webdesign, Fotografie, Druck oder Videotechnik. Es gibt auch in modernen Anwendungen verschiedene Arten von Skalierungsverfahren, die sich in ihrer Funktionsweise und ihrem Ergebnis unterscheiden. Diese Arbeit schafft einen Überblick über klassische und moderne Skalierungsverfahren sowie deren Vor- und Nachteile. Diese werden anhand von Beispielen inszeniert. Zuletzt wird basierend auf der Evaluierung der verschiedenen Verfahren eine Empfehlung für die beste Skalierungsmethode für verschiedene Bildtypen geben.

<sup>2</sup>TECH-LIB [2020]. *Image Scaling Definition*. <http://www.dante.de>.

<sup>3</sup>Wikipedia CONTRIBUTORS [2023]. *Bildskalierung Definition*. [https://en.wikipedia.org/wiki/Image\\_scaling](https://en.wikipedia.org/wiki/Image_scaling).

In dieser Arbeit geht es darum herauszufinden, welche Methode zum Vergrößern oder Verkleinern von Bildern das beste Gleichgewicht aus Komplexität, Rechenaufwand und Ergebnissen liefert. Des weiteren werden die Kriterien zur Bewertung von solchen Methoden umschrieben. Dazu erklären wir zuerst die wichtigsten Konzepte der digitalen Bildverarbeitung und der Skalierung von Bildern und zeigen einige Beispiele für ihre Anwendung. Danach stellen wir die traditionellen Skalierungsmethoden vor und vergleichen ihre Stärken und Schwächen. Dann zeigen wir die neueren Skalierungsmethoden und vergleichen ihre Stärken und Schwächen. Zum Schluss bewerten wir die verschiedenen Methoden mit verschiedenen Maßstäben für die Bildqualität und geben eine Empfehlung für die beste Methode. Wir fassen unsere Ergebnisse zusammen und besprechen ihre Bedeutung und Einschränkungen.

# Kapitel 2

## Grundlagen der Bildverarbeitung und der Skalierung von Bildern

### 2.1 Einblick in die Bildverarbeitung

Historie, Entwicklung, aktueller Stand und mögliche Entwicklungen.

### 2.2 Skalierung von Bildern

#### 2.2.1 Arten der Skalierungen: Interpolation und Skalierung

Die Interpolation und die Skalierung von Bildern oder Bildbereichen sind wichtige Konzepte der Bildverarbeitung. Das Verfahren der Interpolation ermöglicht es neue Pixelwerte auf Basis vorgegebener Werte zu berechnen. Die Skalierung ist eine Anpassung der Bildgröße durch das Ändern der Anzahl von Pixeln oder der Auflösung.

Im Kontext der Bildverarbeitung wird Interpolation häufig verwendet, um die Größe von Bildern zu ändern, ohne dass dabei die Anzahl der Pixel verändert wird. Dazu werden neue Pixelwerte berechnet, indem vorhandene Pixelwerte interpoliert werden. Die Wahl der Interpolationsmethode hat einen großen Einfluss auf die Qualität des interpolierten Bildes. In der Bildverarbeitung gibt es verschiedene Interpolationsmethoden, wie z.B. Nearest-Neighbor-Interpolation, Bilineare Interpolation oder Bicubische Interpolation.

Skalierung hingegen verändert die Größe eines Bildes, indem die Anzahl der Pixel oder die Auflösung verändert wird. Im Gegensatz zur Interpolation wird die Anzahl der Pixel bei der Skalierung verändert, um das Bild kleiner oder größer zu machen. Auch hier hat die Wahl der Skalierungsmethode einen großen Einfluss auf die Qualität des resultierenden Bildes.

#### 2.2.2 Bildformate

Bilder können allgemein als zweidimensionaler Array dargestellt werden. Historisch gesehen gab es jedoch viele unterschiedliche Formate für Bilder. Zunächst erschufen unterschied-

liche Softwareentwickler im Bereich der Bildverarbeitung häufig ihre eigenen Formate.<sup>1</sup> Einheitliche Standards, wie sie heute im Einsatz sind, etablierten sich erst später. Ein Vorreiter der modernen Bildformate ist das "Portable Network Graphics Format"<sup>2</sup>, das 1985 in den USA vorgestellt wurde. Moderne Dateiformate zur Speicherung von Bildern werden anhand der Art des Bildes sowie der Kriterien Speicherbedarf und Kompression, Kompatibilität und ihrem Anwendungsbereich bewertet.<sup>3</sup>

### Portable Network Graphics Format

Portable Network Graphics Format (PNG) setzt einen besonderen Fokus auf eine geringe Komplexität und eine einfache Implementierung des Standards. Der Standard kann frei von jedem genutzt werden. Des weiteren profitiert das Format von verlustfreier Kompression.<sup>4</sup> PNG unterstützt Vollfarbbilder, Grauwertbilder sowie Indexbilder.<sup>5</sup> Der PNG-Algorithmus komprimiert Bilder, indem er mehrere Techniken, einschließlich Filterung und Huffman-Codierung anwendet. Zunächst wird das Bild in Blöcke von 16 x 16 Pixeln aufgeteilt und dann wird auf jedem Block ein Filter angewendet, um Redundanzen zu entfernen. Anschließend wird das Ergebnis der Filterung Huffman-codiert, um eine effiziente Darstellung der Daten zu erreichen. Charakteristisch für PNG-Dateien ist auch die Möglichkeit, transparente Flächen einzubauen. Der Standard verwendet eine spezielle Methode, um Transparenz darzustellen. Diese wird als Alpha-Kanal bezeichnet und ermöglicht es, transparente sowie halbtransparente Bilder zu erstellen. Die komplette Komprimierung des PNG-Formats hat dafür gesorgt, dass der Standard im Internet eine hohe Beliebtheit genießt.<sup>6</sup>

### JPG-Format

Hier noch text bittö.

### Scalable Vector Graphics

The main idea motivating SVG was simple: to create a generic document-oriented solution for graphics that can be adapted to modern media “<sup>7</sup>

---

<sup>1</sup>Wilhelm BURGER und Mark James BURGE [2009]. *Digitale Bildverarbeitung: Eine Algorithmische Einführung Mit Java*. Springer-Verlag.

<sup>2</sup>Thomas BOUTELL u. a. [o. D.] *PNG (Portable Network Graphics Format) Version 1.0*. Techn. Ber. RFC 2083.

<sup>3</sup>Wilhelm BURGER und Mark James BURGE [2009]. *Digitale Bildverarbeitung: Eine Algorithmische Einführung Mit Java*. Springer-Verlag.

<sup>4</sup>Thomas BOUTELL [1997]. *Png (portable network graphics) specification version 1.0*. Techn. Ber.

<sup>5</sup>Wilhelm BURGER u. a. [2015]. »Digitale Bilder«. In: *Digitale Bildverarbeitung: Eine algorithmische Einführung mit Java*, S. 1–24.

<sup>6</sup>`^w3c_png`.

<sup>7</sup>Iris FIBINGER [2002]. *SVG. Scalable Vector Graphics.: Praxiswegweiser und Referenz für den neuen Vektorgrafikstandard. Für Fortgeschrittene*. Markt+Technik-Verl. ISBN: 3827262399,9783827262394. URL: <http://gen.lib.rus.ec/book/index.php?md5=1a655693c0779b14c519078151dfdf31>.

Scalable Vector Graphics (SVG) steht für ein Format, das Vektorgrafiken basierend auf XML darstellt. Im Gegensatz zu Rastergrafiken, wie z.B. PNG, die aus Pixeln bestehen und bei Vergrößerung an Schärfe verlieren, sind Vektorgrafiken vektorbasiert und behalten ihre Qualität bei beliebiger Skalierung. Der Standard ermöglicht eine besonders effiziente Speicherung von Bildern. SVG ist außerdem ein offenes Format und unterstützt Interaktivität, Animation und Skripting.<sup>8</sup> Da SVG auf XML basiert, kann es auch mit anderen Webtechnologien wie HTML, CSS und JavaScript integriert werden.<sup>9</sup>

## Weitere Standards

### GIF

Ein Format für animierte Rastergrafiken mit einer begrenzten Farbpalette von 256 Farben. Es verwendet eine verlustfreie Kompression, die aber nicht sehr effizient ist. Es ist geeignet für einfache Animationen und Grafiken mit wenigen Farben.

**TIFF** Ein Format für hochauflösende Rastergrafiken ohne Kompression oder mit verlustfreier Kompression. Es wird oft im Druckbereich verwendet, da es viele Optionen für Farbmanagement und Metadaten bietet. Es ist aber nicht sehr kompatibel mit Webbrowsern

**PSD** Ein Format für Photoshop-Dokumente, das alle Ebenen, Masken, Effekte und andere Informationen speichert. Es ermöglicht eine umfangreiche Bearbeitung von Rastergrafiken, ist aber nur mit Photoshop kompatibel.

**BMP** Ein Format für unkomprimierte Rastergrafiken mit hoher Qualität. Es wird selten verwendet, da es sehr große Dateien erzeugt und keine Transparenz oder andere Funktionen unterstützt.

**EPS** Ein Format für vektorbasierte Grafiken, das Kurven, Texte und andere Elemente speichert. Es kann skaliert werden ohne Qualitätsverlust und wird oft im Druckbereich verwendet. Es ist aber nicht sehr kompatibel mit Webbrowsern oder anderen Programmen.

Weiterhin gibt es unzählige Standards um Grafiken darzustellen. Diese übersteigen jedoch

---

<sup>8</sup>Antoine QUINT [2003]. »Scalable vector graphics«. In: *IEEE MultiMedia* 10.3, S. 99–102.

<sup>9</sup>MOZILLA CONTRIBUTORS [n.d.] *SVG (Scalable Vector Graphics)*. <https://developer.mozilla.org/en-US/docs/Web/SVG>. Accessed: March 9, 2023.

den Umfang dieser Arbeit.<sup>101112131415</sup>

### 2.2.3 Wichtige Aspekte der Skalierung

#### Segmentierung

Die Segmentierung von Bildern ist ein wichtiges Verfahren der Bildverarbeitung, da es ermöglicht, ein Bild in sinnvolle Regionen zu unterteilen. Hierbei können verschiedene Verfahren, wie Schwellenwert- oder Clustering-Methoden eingesetzt werden. Die Genauigkeit der Segmentierung hängt dabei maßgeblich von der Komplexität des Bildes und der gewählten Methode ab. Eine erfolgreiche Segmentierung kann für viele Anwendungen von Nutzen sein. Die automatische Erkennung von Gesichtern oder die Identifizierung von Verkehrszeichen auf Straßenbildern sind die populärsten Beispiele. Jedoch ist es oft schwer genaue Grenzen zwischen Objekten in Bildern mit komplexen Strukturen zu erkennen.

#### Klassifizierung

Die Klassifizierung von Bildinhalten ist ein weiterer wichtiger Aspekt der Bildverarbeitung. Hierbei werden Bilder in automatisch bestimmte Kategorien eingeteilt. Beispielanwendungen inkludieren das Erkennen von Tierarten auf Naturfotos oder das Identifizieren von Gesichtern auf Fotos. Hierfür können verschiedene Techniken verwendet werden. Beispiele sind Deep Learning oder Entscheidungsbaum-Algorithmen. Eine erfolgreiche Klassifizierung kann für viele Anwendungen genutzt werden, wie zum Beispiel zur Automatisierung von Aufgaben oder im maschinellen Lernen. Die Genauigkeit einer Klassifizierung wird von Faktoren, wie zum Beispiel von der Qualität der Trainingsdaten oder der Komplexität der verwendeten Klassifikationsmethode beeinflusst.

#### Objekterkennung und -verfolgung

Die Objekterkennung und -verfolgung ist ein wichtiger Aspekt der Bildverarbeitung, der oft in Anwendungen wie der Überwachung und Robotik genutzt wird. Hier geht es darum, Objekte in Bildern oder Videos zu erkennen und ihre Bewegungen zu verfolgen. Es können verschiedene Techniken eingesetzt werden, wie zum Beispiel Hintergrundsubtraktion oder

<sup>10</sup>PREPRESSURE [n.d.] *Prepressure Library: File Formats*. <https://www.prepressure.com/library/file-formats/>. Accessed: March 9, 2023.

<sup>11</sup>IONOS [n.d.] *Graphic File Formats: Which Formats Are Important?* <https://www.ionos.com/digitalguide/websites/web-design/graphic-file-formats-which-formats-are-important/>. Accessed: March 9, 2023.

<sup>12</sup>Majid RABBANI und Rajan JOSHI [2002]. »An overview of the JPEG 2000 still image compression standard«. In: *Signal processing: Image communication* 17.1, S. 3–48.

<sup>13</sup>Michael W MARCELLIN u. a. [2000]. »An overview of JPEG-2000«. In: *Proceedings DCC 2000. Data Compression Conference*. IEEE, S. 523–541.

<sup>14</sup>ENCYCLOPEDIA BRITANNICA [n.d.] *JPEG*. <https://www.britannica.com/technology/JPEG>. Accessed: March 9, 2023.

<sup>15</sup>ELSEVIER [n.d.] *Artwork and media instructions*. <https://www.elsevier.com/authors/policies-and-guidelines/artwork-and-media-instructions/artwork-overview>. Accessed: March 9, 2023.

optische Flussberechnung. Eine erfolgreiche Objekterkennung und -verfolgung kann für viele Anwendungen genutzt werden, wie zum Beispiel in der Videoüberwachung oder bei der Steuerung von autonomen Fahrzeugen. Allerdings gibt es auch Herausforderungen bei der Objekterkennung und -verfolgung, wie zum Beispiel die Bewältigung von Hintergrundrauschen oder die Verfolgung von Objekten bei hoher Geschwindigkeit.

### 3D-Bildverarbeitung

In der 3D-Bildverarbeitung werden dreidimensionale Bilder und Modelle analysiert und verarbeitet. Die Anwendungsfelder dieser Technologien reichen von medizinischen Umgebungen bis hin zur industriellen Fertigung. In bildgebenden medizinischen Verfahren werden 3D-Bilder für die Diagnose von Krankheiten und Verletzungen verwendet. In der industriellen Fertigung werden 3D-Modelle für die Qualitätssicherung und die Fehlererkennung verwendet.

### Bildkompression

Da Bilder in der Regel große Datenmengen erzeugen, die für die Übertragung und Speicherung unpraktisch sind, benötigt es oft eine Bildkompression. Durch Kompressionstechniken wie z.B. die JPEG-Kompression kann die Größe von Bildern erheblich reduziert werden ohne dabei große Qualitätsverluste zu erleiden.

## 2.3 Echtzeitverarbeitung von Bildern

Die Echtzeitverarbeitung von Bildern stellt eine Herausforderung in der Bildverarbeitung dar, da eine sehr schnelle Verarbeitung notwendig ist, um zeitkritische Anwendungen wie z.B. autonomes Fahren oder Augmented Reality zu realisieren. Bei diesen Aufgaben müssen Bilder in Echtzeit erfasst, verarbeitet und angezeigt werden, um eine reibungslose Funktionalität zu gewährleisten. Die Echtzeitverarbeitung von Bildern erfordert in der Regel eine hohe Rechenleistung, um die Daten schnell genug zu verarbeiten. Hierbei kommen spezielle Algorithmen zum Einsatz, die für eine effiziente Verarbeitung sorgen. Eine weitere Herausforderung bei der Echtzeitverarbeitung von Bildern ist die Echtzeit-Kommunikation zwischen den verschiedenen Komponenten des Systems. Hierbei müssen Daten schnell und zuverlässig ausgetauscht werden, um Verzögerungen zu vermeiden. Hierfür kommen oft spezielle Systeme zum Einsatz, die für eine schnelle Übertragung und parallele Verarbeitung von Daten optimiert sind.<sup>16</sup>

## 2.4 Anwendungen von Skalierungsmethoden

---

<sup>16</sup>Carl-Werner OEHLICH u. a. [1992]. »Ein Transputersystem mit verteiltem Bildspeicher für Echtzeit-Computergrafik und Bildverarbeitung«. In: *Parallele Datenverarbeitung mit dem Transputer: 3. Transputer-Anwender-Treffen TAT'91, Aachen, 17.-18. September 1991*. Springer, S. 170–177.

amsmath amssymb



# Kapitel 3

## Klassische Skalierungsmethoden

### 3.1 Pixel-Verdopplung

Die Pixel-Verdopplung vergrößert das Bild indem jeder Pixel dupliziert wird. Diese Methode kann schnell und einfach umgesetzt werden, indem jeder Pixelwert einfach auf den Nachbapixel übertragen wird. Wenn Bilder mit dieser Methode stark vergrößert werden, ergeben sich oft pixelige und unscharfe Ausgaben, da die Details nicht wirklich vorhanden sind, sondern nur durch die Duplizierung von Pixeln aufgefüllt werden. Aus diesem Grund wird Pixel-Verdopplung oft als eine minderwertige Skalierungsmethode betrachtet und findet in professionellen Anwendungen selten Gebrauch.<sup>1</sup>

---

<sup>1</sup>David C.C WANG, Anthony H VAGNUCCI und C.C LI [1983]. »Digital image enhancement: A survey«. In: *Computer Vision, Graphics, and Image Processing* 24.3, S. 363–381. ISSN: 0734-189X. DOI: [https://doi.org/10.1016/0734-189X\(83\)90061-0](https://doi.org/10.1016/0734-189X(83)90061-0). URL: <https://www.sciencedirect.com/science/article/pii/0734189X83900610>.

Eine Beispielhafte Implementierung in Python sieht folgendermaßen aus:

```
class PixelVerdopplung(Image):
    def __init__(self, path):
        extend = "p2"
        super().__init__(path, extend)

    def manipulate(self, new_size):
        super().manipulate(new_size)

        for y in range(self.new_height):
            for x in range(self.new_width):
                x_old = int(x / (self.new_width / self.width))
                y_old = int(y / (self.new_height / self.height))

                # Check that x_old and y_old are within bounds of original
                if x_old >= self.width:
                    x_old = self.width - 1
                if y_old >= self.height:
                    y_old = self.height - 1

                old_pixel = self.img.getpixel((x_old, y_old))
                self.newImg.putpixel((x, y), old_pixel)

        return self.save()
```

Die vorliegende Implementierung in Python beschreibt die Realisierung der Pixelverdopplungsklasse, welche in der Lage ist, ein größeres Bild zu erzeugen, indem die leeren Pixel mit demselben Pixelwert wie der nächste Nachbarpixel befüllt werden.

Der Code ist darauf ausgelegt, eine einfache Möglichkeit bereitzustellen, um ein Bild auf eine höhere Auflösung zu skalieren, wodurch fehlende Details ausgeglichen werden können. Dabei wird eine lineare Interpolation auf der Basis der Nachbarpixel durchgeführt, um das neue Bild zu generieren.

Die Klasse "PixelVerdopplung" erbt von der Klasse "Image" und besitzt einen Konstruktor, der den Pfad zum Bild und die Erweiterung "p2" als Argumente erhält. Die Methode "manipulate" ist dafür zuständig, das Bild auf die gewünschte Größe zu skalieren und zu manipulieren.

Innerhalb der Methode werden Schleifen durchlaufen, um jeden neuen Pixel im manipulierten Bild zu generieren. Die Koordinaten des jeweiligen alten Pixels werden durch Division der neuen Koordinaten durch die Skalierungsfaktoren berechnet und auf den nächstgelegenen Integer gerundet.

Um sicherzustellen, dass die berechneten Koordinaten innerhalb der Grenzen des ursprünglichen Bildes liegen, werden sie in einem nächsten Schritt auf den maximalen Index des Bildes zurückgesetzt, falls sie außerhalb liegen sollten. Anschließend wird der Pixelwert des entsprechenden alten Pixels abgerufen und als neuer Pixel an der berechneten

Stelle im neuen Bild platziert.

Die Implementierung dieses Algorithmus stellt eine einfache Möglichkeit dar, um ein Bild auf eine höhere Auflösung zu skalieren, wodurch ein besseres visuelles Ergebnis erzielt werden kann. Dabei ist darauf zu achten, dass die lineare Interpolation eine höhere Laufzeit und Speichieranforderungen aufweist als andere Interpolationsmethoden.

## 3.2 Nearest-Neighbor-Interpolation

Die Nearest-Neighbor-Interpolation ist eine weitere Methode zur Skalierung von Bildern. Es wird für jeden Pixel im Ausgabebild der am nächsten liegende Pixel im Eingabebild ausgewählt und der Farbwert des ausgewählten Pixels wird als Farbwert des entsprechenden Pixels im Ausgabebild verwendet. Die Verwendung von Nearest-Neighbor-Interpolation ist einfach und schnell zu implementieren. Aufgrund ihrer geringen Komplexität ist sie daher sehr beliebt. Die Methode eignet sich besonders gut für die Vergrößerung von Bildern mit großen, einheitlichen Bereichen oder harten Kanten.

Bei der Verkleinerung von Bildern erleiden diese jedoch oft einen Qualitätsverlust. Hier kommt es zu Unschärfe und Blockbildung. Dieser Effekt verstärkt sich, wenn das Verhältniss zwischen Quellbild und Ausgabebild kein Vielfaches ist.

```
import numpy as np
import cv2

def nearest_neighbor_interpolation(image, scale_factor):
    new_size = (int(image.shape[1] * scale_factor), int(image.shape[0] * scale_factor))

    scaled_image = np.zeros(new_size + (image.shape[2],), dtype=np.uint8)
    for i in range(new_size[0]):
        for j in range(new_size[1]):
            x = int(i / scale_factor)
            y = int(j / scale_factor)
            scaled_image[j, i] = image[y, x]

    return scaled_image

image = cv2.imread('example_image.jpg')
scaled_image = nearest_neighbor_interpolation(image, 2)
cv2.imshow('image', image)
cv2.imshow('scaled_image', scaled_image)

2
```

---

<sup>2</sup>Nan JIANG und Luo WANG [2015]. »Quantum image scaling using nearest neighbor interpolation«. In: *Quantum Information Processing* 14, S. 1559–1571.

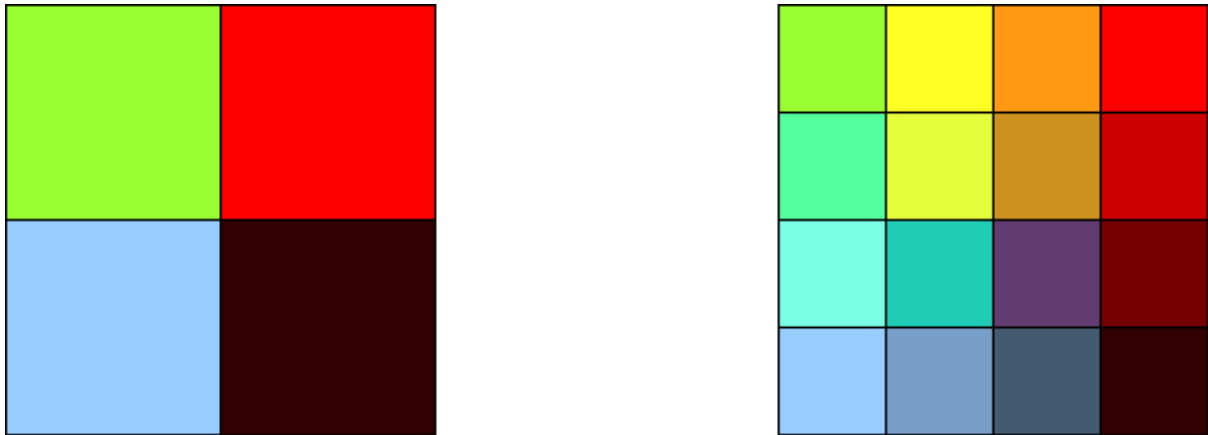


Abbildung 3.1: Beispielgrafik zur Pixelverdopplung.

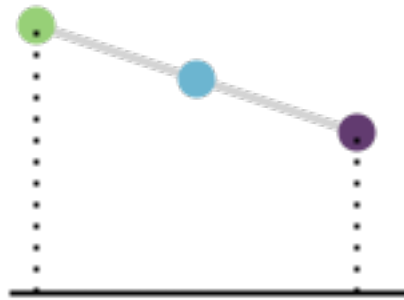


Abbildung 3.2: Graph über bilineare Skalierung.

### 3.3 Bilineare Interpolation

Interpolation cool weil, schnell und ergebniss recht schön.

```
for y in range(self.new_height):
    for x in range(self.new_width):
        x_old = x / (self.new_width / self.width)
        y_old = y / (self.new_height / self.height)

        # Find the surrounding pixels
        x1 = int(x_old)
        x2 = min(x1 + 1, self.width - 1)
        y1 = int(y_old)
        y2 = min(y1 + 1, self.height - 1)

        # Check if x2 and y2 are out of bounds, if they are subtract one
        if x2 == self.width - 1:
            x2 = x1
```

```

        x1 -= 1
    if y2 == self.height - 1:
        y2 = y1
        y1 -= 1

    # Find the weights
    w1 = (x2 - x_old) * (y2 - y_old)
    w2 = (x_old - x1) * (y2 - y_old)
    w3 = (x2 - x_old) * (y_old - y1)
    w4 = (x_old - x1) * (y_old - y1)

    # Get the pixel values of the surrounding pixels
    p1 = self.img.getpixel((x1, y1))
    p2 = self.img.getpixel((x2, y1))
    p3 = self.img.getpixel((x1, y2))
    p4 = self.img.getpixel((x2, y2))

    # Interpolate the pixel value
    new_pixel = (
        int(w1 * p1[0] + w2 * p2[0] + w3 * p3[0] + w4 * p4[0]),
        int(w1 * p1[1] + w2 * p2[1] + w3 * p3[1] + w4 * p4[1]),
        int(w1 * p1[2] + w2 * p2[2] + w3 * p3[2] + w4 * p4[2])
    )
    self.newImg.putpixel((x, y), new_pixel)

```

3

## 3.4 Bicubische Interpolation

### 3.4.1 Mathematische Grundlagen

Die bikubische Interpolation ist ein mathematisches Verfahren zur Schätzung der Werte einer kontinuierlichen Funktion an einer gegebenen Stelle, indem eine Funktion mit kubischen Polynomen verwendet wird, die durch benachbarte Funktionswerte verläuft. Dabei wird das umliegende Gebiet untersucht und die Werte werden basierend auf der Stichprobentheorie geschätzt.

Die bikubische Interpolationsformel ist eine Erweiterung der Bilinearinterpolation auf vier umliegende Pixel und verwendet eine Funktion, die durch benachbarte Funktionswerte verläuft. Die resultierende Funktion ist stetig differenzierbar und besitzt glatte partielle Ableitungen.

---

<sup>3</sup>K.T. GRIBBON und D.G. BAILEY [2004]. »A novel approach to real-time bilinear interpolation«. In: *Proceedings. DELTA 2004. Second IEEE International Workshop on Electronic Design, Test and Applications*, S. 126–131. DOI: 10.1109/DELTA.2004.10055.

Im Vergleich zu anderen Interpolationsverfahren wie der bilinearen Interpolation und der Lanczos-Interpolation hat die bikubische Interpolation den Vorteil, dass sie eine höhere Genauigkeit bei der Schätzung von Pixelwerten bietet. Ein Nachteil ist jedoch, dass sie im Allgemeinen höhere Rechenaufwendungen erfordert.

Die Fourier-Analyse und die Fourier-Transformationen spielen eine wichtige Rolle bei der Bildinterpolation, da sie es ermöglichen, die Funktion in den Frequenzraum zu transformieren und damit eine effektivere Interpolation zu erreichen.

### 3.4.2 Algorithmische Implementierung

Die bikubische Interpolation erfolgt durch die Auswahl von umgebenden Pixeln und deren Gewichte sowie der Berechnung der neuen Pixelwerte. Eine Möglichkeit zur Optimierung der Leistung besteht darin, Lookup-Tabellen vorzubereiten, Parallelisierungstechniken zu verwenden und die Speicherverwaltung zu optimieren.

Grenzfälle und Randbedingungen können bei der bikubischen Interpolation auftreten und müssen effektiv behandelt werden. Die algorithmische Implementierung der bikubischen Interpolation kann mit anderen Interpolationsverfahren wie der bilinearen Interpolation und der Lanczos-Interpolation verglichen werden.

Ein Beispiel für die Implementierung der bikubischen Interpolation in Python ist der folgende Code:

Die Methode ‘manipulate’ der Klasse ‘BicubicInterpolation’ implementiert die bikubische Interpolation für das Vergrößern von Bildern. Der wichtigste Teil des Codes ist die Schleife, die über jedes Pixel einen Kernel berechnet, der die jeweiligen Gewichte der benachbarten Pixel errechnet. In den genannten Zeilen wird ein 4x4-Kern um das aktuelle Pixel herum gebildet und für jeden Pixel im Kern werden die Gewichte berechnet. Dabei wird die ‘*get\_weight*’-Funktion aufgerufen, welche auf der Basis einer kubischen Kurve das Gewicht für einen bestimmten

```
weights = []
for j in range(-1, 3):
    for i in range(-1, 3):
        weight = __get_weight(i - dx) * __get_weight(dy - j)
        weights.append(weight)
```

### 3.4.3 Analyse der Leistung

Zur Bewertung der Leistung von Bildinterpolationsverfahren werden verschiedene Kriterien verwendet, einschließlich der visuellen Qualität, der Genauigkeit und der Berechnungseffizienz. Die Leistung der bikubischen Interpolation kann mit anderen Interpolationstechniken anhand von Testbildern und Datensätzen verglichen werden. Dabei werden verschiedene Parameter wie Bildgröße, Auflösung und Inhalt untersucht, um ihre Auswirkungen auf die Leistung der bikubischen Interpolation zu analysieren.

### 3.4.4 Mathematische Grundlagen

Erläuterung der mathematischen Grundlagen der bikubischen Interpolation, einschließlich der Verwendung von kubischen Polynomen und der Stichprobentheorie. Herleitung der bikubischen Interpolationsformel und der Eigenschaften der resultierenden Funktion. Diskussion der Vor- und Nachteile der Verwendung kubischer Funktionen für die Interpolation im Vergleich zu anderen Funktionstypen. Überblick über die Rolle der Fourier-Analyse und der Fourier-Transformationen bei der Bildinterpolation und wie sich dies auf die bikubische Interpolation bezieht.

### 3.4.5 Algorithmische Implementierung

Überblick über die algorithmischen Schritte bei der bikubischen Interpolation, einschließlich der Auswahl der umgebenden Pixel und der Gewichte sowie der Berechnung der neuen Pixelwerte. Diskussion von Techniken zur Optimierung der Leistung der bikubischen Interpolation, wie z. B. die Vorberechnung von Lookup-Tabellen, Parallelisierung und Speicherverwaltung. Untersuchung von Grenzfällen und Randbedingungen, die bei der bikubischen Interpolation auftreten können, und wie diese effektiv behandelt werden können. Vergleich der algorithmischen Implementierung der bikubischen Interpolation mit anderen Interpolationsverfahren, wie z.B. der bilinearen Interpolation und der Lanczos-Interpolation.

### 3.4.6 Implementation

```
from scipy import interpolate
import numpy as np

def bicubic_interpolation(image, new_shape):
    height, width = image.shape[:2]
    x = np.arange(0, width)
    y = np.arange(0, height)
    f = interpolate.interp2d(x, y, image, kind='cubic')
    new_width, new_height = new_shape
    new_x = np.linspace(0, width-1, new_width)
    new_y = np.linspace(0, height-1, new_height)
    return f(new_x, new_y)
```

### 3.4.7 Analyse der Leistung

Erläuterung der Kriterien, die zur Bewertung der Leistung von Bildinterpolationsverfahren verwendet werden, einschließlich der visuellen Qualität, der Genauigkeit und der Berechnungseffizienz. Vergleich der Leistung der bikubischen Interpolation mit anderen Interpolationstechniken unter Verwendung einer Reihe von Testbildern und Datensätzen. Untersuchung der Auswirkungen verschiedener Parameter wie Bildgröße, Auflösung

und Inhalt auf die Leistung der bikubischen Interpolation. Diskussion der potenziellen Einschränkungen und Kompromisse, die mit der Verwendung der bikubischen Interpolation verbunden sind, sowie der Faktoren, die ihre Leistung in verschiedenen Kontexten beeinflussen können.

### 3.4.8 Anwendungen

Überblick über die verschiedenen Anwendungen der bikubischen Interpolation in der Bildverarbeitung, einschließlich Bildskalierung und -größenänderung, Bildrotation und -transformation sowie Bildentrauschung und -wiederherstellung. Diskussion der spezifischen Herausforderungen und Möglichkeiten, die sich in verschiedenen Anwendungsbereichen ergeben, z. B. in der medizinischen Bildgebung, der Satellitenbildgebung und der Videoverarbeitung. Erforschung der potenziellen Vor- und Nachteile der bikubischen Interpolation in verschiedenen Anwendungen und der Faktoren, die ihre Eignung für bestimmte Aufgaben beeinflussen können.

### 3.4.9 Erweiterungen und Variationen

Erläuterung der verschiedenen Erweiterungen und Variationen der bikubischen Interpolation, wie z. B. Super-Resolution-Techniken, Multiskalen- und pyramidenbasierte Interpolation und adaptive Interpolationstechniken. Diskussion der Vor- und Nachteile dieser Varianten und ihrer Eignung für verschiedene Bildtypen und Anwendungen. Erkundung potenzieller künftiger Forschungsrichtungen in diesem Bereich, z. B. auf Deep Learning basierende Ansätze, ungleichmäßige und unregelmäßige Abtastverfahren sowie mehrdimensionale und mehrkanalige Interpolation.

## 3.5 Lanczos-Interpolation

Die Lanczos-Interpolation ist eine Methode zur Rekonstruktion von Werten im Bild aus diskreten Abtastungen. In diesem Abschnitt wird die mathematische Grundlage und die praktische Anwendung der Lanczos-Interpolation kurz erläutert.

### 3.5.1 Mathematische Grundlage

Die Lanczos-Interpolation basiert auf der Idee, ein kontinuierliches Signal  $f(x)$  durch eine Summe von gewichteten Basisfunktionen zu approximieren. Dieses Signal können zum Beispiel Farbwerte in einem Bild sein. Die Basisfunktionen werden durch das sogenannte Lanczos-Kernel definiert:

$$\text{MARCHIERFormel aus BILDEINFÜGEN} \quad (3.1)$$

---

<sup>4</sup>Claude E DUCHON [1979]. »Lanczos filtering in one and two dimensions«. In: *Journal of Applied Meteorology and Climatology* 18.8, S. 1016–1022.



Das Lanczos-Kernel hat eine kompakte Trägerfunktion. Eine kompakte Trägerformel bedeutet, sie ist nur in einem begrenzten Bereich von Null verschieden. In der Praxis hat dies den Vorteil, dass das Signalrauschen in Bereichen außerhalb des Bereichs im Signalraum reduziert wird und somit eine bessere Interpolation des Signals erreicht werden kann. Die Gewichtungen der Basisfunktionen werden durch die Interpolationskoeffizienten bestimmt, die durch die diskreten Abtastungen des Signals berechnet werden.

Die Lanczos-Interpolation wird in der Regel auf gleichmäßig verteilten Stützstellen angewendet. Die Interpolationsmethode verwendet diese Stützstellen als Ausgangspunkt, um eine Schätzung des Signals an anderen Orten zu berechnen. Seien  $x_1, x_2, \dots, x_n$  die Stützstellen des Signals und  $y_1, y_2, \dots, y_n$  die zugehörigen Abtastungen. Die Interpolationsfunktion  $s(x)$  kann dann wie folgt berechnet werden:

$$\text{Interpolationsfunktionbrohilfe@marc} \quad (3.2)$$

wobei  $h$  der Abstand zwischen den Stützstellen ist.<sup>5</sup>

### 3.5.2 Praktische Anwendung

Die Lanczos-Interpolation findet Anwendung in vielen Bereichen der Bildverarbeitung. Ein Anwendungsbeispiel ist die Upsampling von digitalen Bildern, um eine höhere Auflösung zu erreichen.

Die praktische Umsetzung der Lanczos-Interpolation erfordert die Berechnung der Interpolationskoeffizienten und die Bestimmung der Stützstellen des Signals. In der Regel werden hierfür spezielle Algorithmen eingesetzt, die auf der effizienten Berechnung der Basisfunktionen basieren.

Hier Implementierung einfügen (TODO)

## 3.6 Zusammenfassung zu Vor- und Nachteilen von Techniken in der Bildverarbeitung

In der Bildverarbeitung gibt es verschiedene Techniken, die verwendet werden können, um ein Bild zu bearbeiten oder zu manipulieren. In diesem Abschnitt werden wir uns mit einigen gängigen Techniken zur Interpolation von Bildern beschäftigen, nämlich Pixelverdopplung, Nearest Neighbour Interpolation, Bilineare Interpolation, Bicubische Interpolation und Lanczos Interpolation. Wir werden jeweils auf die Vor- und Nachteile dieser Techniken eingehen.

---

<sup>5</sup>A.H. BENTBIB u. a. [2016]. »A global Lanczos method for image restoration«. In: *Journal of Computational and Applied Mathematics* 300, S. 233–244. ISSN: 0377-0427. DOI: <https://doi.org/10.1016/j.cam.2015.12.034>. URL: <https://www.sciencedirect.com/science/article/pii/S0377042715006469>.

#### 3.6.1 Pixelverdopplung

Bei der Pixelverdopplung wird jedes Pixel im Bild einfach dupliziert, um ein größeres Bild zu erzeugen. Diese Methode ist einfach und schnell, aber sie führt oft zu einer Verzerrung des Bildes und kann zu einer verschlechterten Bildqualität führen.

#### 3.6.2 Nearest Neighbour Interpolation

Die Nearest Neighbour Interpolation ist eine einfache Interpolationsmethode, bei der jeder neue Pixelwert durch den nächstgelegenen Pixelwert im Originalbild bestimmt wird. Diese Methode ist einfach und schnell, aber sie führt oft zu einem "Treppeneffekt" im Bild, da die Pixelwerte nicht kontinuierlich interpoliert werden.

#### 3.6.3 Bilineare Interpolation

Die bilineare Interpolation ist eine Methode, bei der die neuen Pixelwerte aus einer bilinearen Funktion berechnet werden, die aus den vier nächstgelegenen Pixeln im Originalbild abgeleitet wird. Diese Methode führt oft zu einer glatteren Interpolation als die Nearest Neighbour Interpolation, aber es kann immer noch zu einem "Verwischen" des Bildes kommen.

#### 3.6.4 Bicubische Interpolation

Die bicubische Interpolation ist eine Methode, bei der die neuen Pixelwerte aus einer bicubischen Funktion berechnet werden, die aus den sechzehn nächstgelegenen Pixeln im Originalbild abgeleitet wird. Diese Methode führt oft zu einer noch glatteren Interpolation als die bilineare Interpolation, aber sie kann auch zu einer Überbetonung von Bildstrukturen führen.

#### 3.6.5 Lanczos Interpolation

y Die Lanczos Interpolation ist eine Methode, bei der die neuen Pixelwerte aus einer Lanczos-Funktion berechnet werden, die aus einer begrenzten Anzahl von nächstgelegenen Pixeln im Originalbild abgeleitet wird. Diese Methode führt oft zu einer sehr glatten Interpolation und reduziert das Rauschen im Bild, aber sie kann auch zu einer gewissen Unschärfe im Bild führen.

#### 3.6.6 Vergleich und Zusammenfassung

Insgesamt gibt es keine "beste" Methode zur Interpolation von Bildern, da jede Methode ihre eigenen Vor- und Nachteile hat. Die Wahl der Methode hängt von den spezifischen Anforderungen und Einschränkungen ab, die für die jeweilige Anwendung gelten. Die Wahl einer geeigneten Interpolationsmethode kann jedoch die Bildqualität erheblich verbessern und zu einer effektiveren Bildverarbeitung führen.

# Kapitel 4

## Fortgeschrittene Skalierungsmethoden

### 4.1 Convolutional Neural Networks / Deep learning

#### 4.1.1 Grundlagen von Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) sind eine Art von Deep-Learning-Modell, welches besonders im Hinblick auf die Verarbeitung von Daten mit räumlicher Struktur den aktuellen Stand der Technik darstellt. Räumliche Daten, wie z.B. Bilder können bearbeitet, verarbeitet, erstellt und analysiert werden. CNNs bestehen aus mehreren Schichten von Neuronen, die so angeordnet sind, dass sie räumliche Beziehungen in den Daten erfassen können. Die grundlegende Idee hinter CNNs ist die Verwendung von Faltung (engl. convolution) anstelle der vollständigen Verbindung (engl. fully connected) zwischen den Schichten. Dies bedeutet, dass jedes Neuron in einer Schicht nur mit einem Teil des Eingangs verbunden ist, anstatt mit jedem Eingangsneuron. Diese Art der Verbindung spart Rechenleistung und ermöglicht eine effektivere sowie schnellere Verarbeitung von großen Datensätzen.<sup>1</sup>(kolbentwicklung,oshea2015introduction)

#### 4.1.2 Architekturen von CNNs

Es gibt mehrere bekannte Architekturen von CNNs, darunter AlexNet, ResNet und Inception. AlexNet war das erste CNN, das auf einem großen Datensatz erfolgreich angewendet wurde. ResNet zeichnet sich durch seine Fähigkeit aus, sehr tiefe Netzwerke zu trainieren, ohne dass das Problem des Verschwindens des Gradienten auftritt. Inception wiederum ist für seine Fähigkeit bekannt, die Effizienz von CNNs durch die Verwendung von sogenannten Inception-Modulen zu erhöhen.

---

<sup>1</sup>(.

### 4.1.3 Anwendungen von CNNs

CNNs haben zahlreiche Anwendungen, darunter Bildklassifizierung, Objekterkennung und Gesichtserkennung. Bei der Bildklassifizierung werden Bilder automatisch in verschiedene Kategorien eingeteilt. Beispielsweise können Bilder in Klassen wie Hunde, Katzen oder Autos eingeteilt werden. Bei der Objekterkennung wird das Modell darauf trainiert, bestimmte Objekte in einem Bild zu erkennen, wie z.B. Personen oder Straßenschilder. Die Gesichtserkennung wird oft zur Identifikation von Personen in Sicherheitsanwendungen eingesetzt.<sup>23</sup>oshea2015introduction)

### 4.1.4 Transfer Learning mit CNNs

Transfer Learning ist eine Technik, bei der ein bereits trainiertes CNN auf eine neue Aufgabe angewendet wird, ohne es von Grund auf neu zu trainieren. Dies ist nützlich, wenn man nur über begrenzte Trainingsdaten verfügt oder wenn das Trainieren eines neuen Modells zu viel Zeit oder Ressourcen in Anspruch nimmt. Ein Beispiel hierfür ist ????????, bei denen das Modell auf einem bereits trainierten CNN basieren kann, das auf ?????? trainiert wurde.

### 4.1.5 Limitationen von CNNs und aktuelle Forschungsrichtungen

Obwohl CNNs sehr erfolgreich bei der Verarbeitung von Bildern sind, haben sie auch einige Limitationen. Zum Beispiel sind sie nicht gut geeignet, um komplexe Abhängigkeiten zwischen verschiedenen Eingabemerkmalen zu erfassen, wie z.B. das Verhalten von Objekten in einem Video. Zudem benötigen CNNs weiterhin viel Rechenleistung und Ressourcen<sup>4</sup>

## 4.2 Super Resolution

### 4.2.1 Grundlagen von Super Resolution (SR)

Super Resolution (SR) ist eine Technik, um aus einer niedrig aufgelösten Eingabe ein hochauflösendes Bild zu generieren. Dies wird oft als Upscaling bezeichnet und findet in vielen Anwendungen wie der Bildrekonstruktion und Videoanalyse Anwendung. Die

---

<sup>2</sup>Max JADERBERG, Karen SIMONYAN, Andrew ZISSERMAN u. a. [2015]. »Spatial transformer networks«. In: *Advances in neural information processing systems* 28; Ziwei LIU u. a. [2015]. »Deep learning face attributes in the wild«. In: *Proceedings of the IEEE international conference on computer vision*, S. 3730–3738; Bichen WU u. a. [2017]. »Squeezednet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, S. 129–137.

<sup>3</sup>(.

<sup>4</sup>Hao LI u. a. [2018]. »Visualizing the loss landscape of neural nets«. In: *Advances in neural information processing systems* 31.

Grundidee hinter SR ist, dass hochauflösende Informationen in einem niedrig aufgelösten Bild versteckt sein können. Die Herausforderung besteht darin, diese Informationen zu extrahieren und in ein hochauflösendes Bild zu integrieren. SR ist somit ein Problem der inversen Bildgebung, bei dem eine hohe Auflösung aus einer niedrigen Auflösung abgeleitet werden muss.<sup>5</sup>(7115171)

### 4.2.2 Super Resolution-Methoden auf Basis von Deep Learning

Super Resolution-Methoden auf Basis von Deep Learning haben in den letzten Jahren viel Aufmerksamkeit erhalten und sind derzeit der Stand der Technik für SR. Diese Methoden verwenden Convolutional Neural Networks (CNNs) zur Verarbeitung von Bildern und zur Generierung von hochauflösenden Bildern. Es gibt verschiedene Arten von SR-Methoden auf Basis von Deep Learning, darunter Single-Image Super Resolution (SISR) und Multi-Image Super Resolution (MISR). SISR-Methoden verwenden nur ein niedrig aufgelöstes Bild als Eingabe, während MISR-Methoden mehrere Bilder verwenden, um ein hochauflösendes Bild zu generieren.

#### Anwendungen von SR

SR hat viele Anwendungen in der Bild- und Videoanalyse, einschließlich der Rekonstruktion von Bildern aus medizinischen Scans, der Verbesserung von Bildern für die forensische Analyse und der Verbesserung von Bildern für die Erkennung von Gesichtern und Objekten. In der Videoanalyse kann SR verwendet werden, um Videos zu stabilisieren, indem Bewegungsunschärfe reduziert und die Schärfe der Bilder verbessert wird. SR kann auch bei der Entschlüsselung von unscharfen und verschwommenen Bildern in Überwachungsaufnahmen helfen.

#### Evaluierung von SR-Methoden

Die Evaluierung von SR-Methoden ist eine wichtige Aufgabe, um die Qualität und Effektivität der generierten Bilder zu bestimmen. Die gängigen Evaluierungsmethoden umfassen die Verwendung von visuellen Qualitätsmetriken wie Peak Signal-to-Noise Ratio (PSNR) und Structural Similarity Index Measure (SSIM). Es gibt auch speziellere Evaluierungsmethoden wie die Verwendung von Perceptual Quality Assessment (PQA)-Maßnahmen, die menschliche Wahrnehmungseigenschaften berücksichtigen, um die Qualität der generierten Bilder zu bestimmen.

#### Herausforderungen und zukünftige Forschungsziele von Super Resolution

Obwohl SR-Methoden auf Basis von Deep Learning vielversprechende Ergebnisse erzielt haben, gibt es immer noch Herausforderungen und zukünftige Forschungsziele, die erforscht werden müssen. Eine der Herausforderungen besteht darin, dass SR-Methoden häufig dazu neigen, Artefakte in den generierten Bildern zu erzeugen, insbesondere bei der

---

<sup>5</sup>(.

Verwendung von sehr hohen Upscaling-Faktoren. Dies kann die visuelle Qualität der generierten Bilder beeinträchtigen und die Anwendbarkeit von SR-Methoden in bestimmten Szenarien einschränken. Eine weitere Herausforderung besteht darin, dass SR-Methoden häufig sehr rechenaufwändig sind, insbesondere wenn sie auf großen Datensätzen oder in Echtzeit angewendet werden müssen. Die benötigten Ressourcen sind teuer. Dies kann die praktische Anwendbarkeit von SR-Methoden in einigen Anwendungen einschränken. Zukünftige Forschungsziele könnten sich darauf konzentrieren, diese Herausforderungen zu überwinden, indem sie neue SR-Methoden entwickeln, die sowohl effektiv als auch effizient sind. Eine mögliche Lösung wäre die Verwendung von Generative Adversarial Networks (GANs) zur Verbesserung der visuellen Qualität der generierten Bilder und zur Reduzierung von Artefakten. Eine weitere mögliche Lösung wäre die Entwicklung von neuartigen Architekturen von Deep Learning-Netzwerken, die weniger rechenaufwändig sind und schneller ausgeführt werden können. Insgesamt bleibt SR ein aktives Forschungsfeld mit großem Potenzial für Anwendungen in der Bild- und Videoanalyse. Mit weiteren Fortschritten in der Forschung können SR-Methoden immer leistungsfähiger und praktischer werden, um die Bedürfnisse der Industrie und der Gesellschaft zu erfüllen.

## 4.3 Generative Adversarial Networks (GANs)

### 4.3.1 Grundlagen von Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) sind ein leistungsstarkes Framework für das Training von Deep Learning-Modellen zur Generierung von Daten. GANs bestehen aus zwei miteinander konkurrierenden neuronalen Netzwerken, einem Generator und einem Diskriminator. Der Generator erzeugt neue Daten, während der Diskriminator versucht, zwischen den vom Generator erzeugten Daten und den echten Daten zu unterscheiden. Im Laufe des Trainings passt sich der Generator kontinuierlich an und verbessert seine Fähigkeit, realistische Daten zu generieren, während der Diskriminator gleichzeitig verbessert wird, um zwischen den generierten und echten Daten zu unterscheiden.

### 4.3.2 Architekturen von GANs

Es gibt verschiedene Architekturen von GANs, die für verschiedene Arten von Anwendungen geeignet sind. Ein Beispiel ist das Deep Convolutional GAN (DCGAN), das speziell für die Generierung von Bildern entwickelt wurde. DCGAN nutzt Convolutional Neural Networks (CNNs) und Transposed Convolutional Neural Networks, um Bilder zu generieren, die visuell realistisch aussehen und strukturell konsistent sind. Ein weiteres Beispiel ist das CycleGAN, das für die Bildübersetzung zwischen verschiedenen Domänen verwendet werden kann. CycleGAN nutzt einen Generator und einen Diskriminator sowie zusätzliche Cycle-Verlustfunktionen, um die Transformationen zwischen den Bildern in verschiedenen Domänen zu erlernen.

### 4.3.3 Anwendungen von GANs

GANs finden Anwendungen in verschiedenen Bereichen wie der Bildgenerierung, Style Transfer, der Verbesserung von Bildern und der Videoanalyse. Zum Beispiel können GANs verwendet werden, um realistisch aussehende Bilder von Gesichtern, Landschaften oder anderen Objekten zu generieren. Style Transfer ermöglicht es, das visuelle Erscheinungsbild von Bildern zu verändern, indem der Stil von einem Bild auf ein anderes übertragen wird. GANs können auch verwendet werden, um Bilder mit höherer Auflösung oder besserer Qualität zu generieren, indem sie niedrig aufgelöste Bilder als Eingabe verwenden. In der Videoanalyse können GANs verwendet werden, um Videosequenzen zu generieren oder zu verbessern.

### 4.3.4 Training von GANs und Evaluierung von generierten Ergebnissen

Das Training von GANs ist eine Herausforderung, da es sich um ein adversariales Lernverfahren handelt. Das bedeutet, dass es zwei Netze gibt, die sich gegenseitig trainieren und verbessern. Das generative Netzwerk versucht, Bilder zu erzeugen, die von einem diskriminierenden Netzwerk nicht von echten Bildern unterschieden werden können. Das diskriminierende Netzwerk wird trainiert, um echte Bilder von den vom generativen Netzwerk generierten Bildern zu unterscheiden. Das Training von GANs erfolgt durch die Minimierung einer Verlustfunktion, die als GAN-Verlust bezeichnet wird. Der GAN-Verlust besteht aus zwei Komponenten: dem Verlust des generativen Netzes und dem Verlust des diskriminierenden Netzes. Der Verlust des generativen Netzes wird minimiert, wenn das Netzwerk Bilder erzeugt, die vom diskriminierenden Netzwerk nicht als gefälscht erkannt werden. Der Verlust des diskriminierenden Netzes wird minimiert, wenn das Netzwerk in der Lage ist, besonders zuverlässig und schnell zwischen echten und generierten Bildern zu unterscheiden. Die Evaluierung von generierten Ergebnissen ist eine wichtige Aufgabe bei der Arbeit mit GANs. Es gibt verschiedene Methoden zur Bewertung von GANs, wie beispielsweise die visuelle Bewertung, die qualitative Bewertung und die quantitative Bewertung. Die visuelle Bewertung beinhaltet das Betrachten der generierten Bilder, um zu beurteilen, ob sie realistisch aussehen oder nicht. Die qualitative Bewertung beinhaltet die Verwendung von Bewertungsskalen, um die Qualität der generierten Bilder zu bewerten. Die quantitative Bewertung beinhaltet die Verwendung von Metriken wie der Inception Score oder der Frechet Inception Distance, um die Qualität der generierten Bilder zu bewerten.

### 4.3.5 Ethische und soziale Implikationen von GANs

Obwohl GANs eine vielversprechende Technologie sind, gibt es auch ethische und soziale Implikationen, die berücksichtigt werden müssen. Ein Problem bei der Verwendung von GANs ist, dass sie zur Erzeugung gefälschter Bilder oder Videos verwendet werden können. Dies kann zu Fälschungen und Manipulationen führen, die negative Auswirkungen auf die Gesellschaft haben können. Auch Rufschädigung kann durch GANs erleichtert werden. Ein

weiteres Problem bei der Verwendung von GANs ist, dass sie möglicherweise nicht fair sind. GANs können aufgrund ihrer Lernmethode unbewusste Vorurteile aufnehmen und in ihren generierten Ergebnissen widerspiegeln. Dies kann zu diskriminierenden Ergebnissen führen, die unfaire Entscheidungen unterstützen. Es ist wichtig, dass bei der Verwendung von GANs Ethik und soziale Verantwortung berücksichtigt werden. Es sollten Maßnahmen ergriffen werden, um sicherzustellen, dass GANs fair und ethisch korrekt arbeiten. Zum Beispiel können spezielle Algorithmen entwickelt werden, um unbewusste Vorurteile zu minimieren. Weiterhin können Regierungsbehörden und andere Organisationen Maßnahmen ergreifen, um den Missbrauch von GANs zu verhindern. Das Finden eines Kompromisses aus Forschung und politischer Einschränkung übersteigt jedoch den Rahmen dieser Arbeit.

## 4.4 Multiscale-Skalierung

In vielen Anwendungen der Bildverarbeitung und Computergrafik ist es notwendig, Objekte oder Strukturen in verschiedenen Größenordnungen zu analysieren oder darzustellen. Multiscale-Skalierung bezeichnet Techniken und Vorgehen, die es ermöglichen, Objekte oder Signale auf verschiedenen Skalen zu untersuchen oder zu manipulieren. Dabei können sowohl lokale als auch globale Eigenschaften eines Objekts berücksichtigt werden. Diese Möglichkeiten komplementieren häufig Implementierungen von künstlichen Intelligenzen "Machine learning and multiscale modeling mutually complement one another"<sup>6</sup>

### 4.4.1 Grundlagen von Multiscale-Skalierung

Multiscale-Skalierung ist ein Konzept aus der Signal- und Bildverarbeitung, das auf der Idee basiert, dass Signale und Bilder aus verschiedenen Skalen von Strukturen aufgebaut sind. In der Praxis bedeutet dies, dass ein Signal oder Bild auf verschiedene Skalen abgetastet oder transformiert wird, um Informationen auf verschiedenen Größenskalen zu erhalten. Ein grundlegendes Konzept der Multiscale-Skalierung ist die Skaleninvarianz. Das bedeutet, dass die Informationen in einem Signal oder Bild unabhängig von der Skala erhalten bleiben sollten. Das heißt, dass dieselben Merkmale oder Strukturen auf verschiedenen Skalen erkennbar sein sollten.<sup>7</sup>

### 4.4.2 Methoden zur Multiskalenanalyse

Es gibt verschiedene Techniken zur Multiskalenanalyse, darunter Wavelet-Transformation und pyramidenartige Strukturen. Wavelet-Transformation ist eine Methode zur Analyse

<sup>6</sup>Mark ALBER u. a. [2019]. »Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences«. In: *NPJ digital medicine* 2.1, S. 115.

<sup>7</sup>Gao HUANG u. a. [2017]. »Densely connected convolutional networks«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, S. 4700–4708; Stephane G MALLAT [1989]. »A theory for multiresolution signal decomposition: the wavelet representation«. In: *IEEE transactions on pattern analysis and machine intelligence* 11.7, S. 674–693.



und Synthese von Signalen oder Bildern auf verschiedenen Skalen. Dabei wird das Signal oder Bild auf verschiedene Skalen und Frequenzen zerlegt, um Informationen auf verschiedenen Skalen zu erhalten. Pyramidenartige Strukturen sind eine weitere Methode zur Multiskalenanalyse, die in der Bildverarbeitung häufig verwendet wird. Dabei wird das Signal oder Bild auf verschiedenen Skalen durch wiederholte Subsampling- und Filterungsoperationen reduziert. Auf jeder Ebene der Pyramide wird das Signal oder Bild auf eine kleinere Größe reduziert, um Informationen auf verschiedenen Skalen zu erhalten.<sup>8</sup>

### 4.4.3 Methoden zur Multiskalenanalyse

Es gibt verschiedene Methoden zur Multiskalenanalyse, darunter Wavelets und Pyramiden. Wavelets basieren auf der Zerlegung von Signalen in unterschiedliche Frequenzbänder, wodurch eine Multiskalenanalyse ermöglicht wird. Eine Wavelet-Transformation kann auf ein Signal angewendet werden, um es in seine Hoch- und Niederfrequenzkomponenten zu zerlegen. Diese Komponenten können dann unabhängig voneinander verarbeitet werden, um eine Analyse auf verschiedenen Skalen durchzuführen. Pyramiden sind eine weitere Methode zur Multiskalenanalyse, die auf der Idee der rekursiven Unterteilung von Signalen in immer feinere Skalen basiert. Dabei wird ein Signal in eine Pyramide aus verschiedenen Ebenen unterteilt, wobei jede Ebene eine andere Skala darstellt. Die unterste Ebene enthält das Originalsignal, während die oberen Ebenen eine immer gröbere Approximation des Signals enthalten.<sup>9</sup>

### 4.4.4 Anwendungen von Multiscale-Skalierung

Multiscale-Skalierung hat viele Anwendungen in der Bildverarbeitung und Computer Vision. Ein Beispiel ist die Texturanalyse, bei der Texturen auf verschiedenen Skalen analysiert werden, um Muster und Strukturen zu identifizieren. Multiscale-Skalierung wird auch in der Bildkompression verwendet, um Bilder auf verschiedene Auflösungen zu skalieren und so Speicherplatz zu sparen. In jüngerer Zeit wurde die Multiskalenanalyse auch in Verbindung mit Deep Learning eingesetzt, um Modelle zu entwickeln, die auf verschiedenen Skalen arbeiten können. Ein Beispiel ist das Multiscale Dense Network (MSDN), das eine skalierbare Architektur für die Bilderkennung bietet, die auf mehreren Skalen arbeiten kann.

---

<sup>8</sup>Peter J BURT und Edward H ADELSON [1987]. »The Laplacian pyramid as a compact image code«. In: *Readings in computer vision*. Elsevier, S. 671–679; Seungjun NAH, Tae HYUN KIM und Kyoung MU LEE [2017]. »Deep multi-scale convolutional neural network for dynamic scene deblurring«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, S. 3883–3891.

<sup>9</sup>Eero P SIMONCELLI und Edward H ADELSON [1996]. »Noise removal via Bayesian wavelet coring«. In: *Proceedings of 3rd IEEE International Conference on Image Processing*. Bd. 1. IEEE, S. 379–382.

#### 4.4.5 Limitationen und zukünftige Forschungsziele von Multiscale-Skalierung

Obwohl die Multiskalenanalyse in vielen Bereichen der Bildverarbeitung und Computer Vision erfolgreich eingesetzt wurde, gibt es auch einige Limitationen. Eines der Hauptprobleme ist die Herausforderung, die richtige Skala für eine bestimmte Aufgabe zu wählen. In einigen Fällen kann dies schwierig sein, da verschiedene Skalen unterschiedliche Informationen enthalten. Eine weitere Herausforderung besteht darin, die Multiskalenanalyse mit Deep Learning-Methoden zu integrieren. Während einige Fortschritte in diesem Bereich gemacht wurden, gibt es immer noch Raum für Verbesserungen, um die Multiskalenanalyse nahtlos in Deep Learning-Architekturen zu integrieren.<sup>10</sup> Zukünftige Forschungsrichtungen könnten sich auf die Entwicklung von verbesserten Methoden zur Skalierung und Multiskalenanalyse konzentrieren, die eine höhere Genauigkeit und Effizienz ermöglichen. Darüber hinaus könnten Forscher daran arbeiten, die Integration der Multiskalenanalyse in Deep Learning-Architekturen weiter zu verbessern, um noch bessere Ergebnisse zu erzielen.

### 4.5 Vor- und Nachteile der fortgeschrittenen Methoden

- Vergleich der verschiedenen fortgeschrittenen Skalierungsmethoden - Vorteile von fortgeschrittenen Methoden im Vergleich zu traditionellen Methoden - Herausforderungen bei der Anwendung fortgeschrittener Methoden - Auswirkungen von fortgeschrittenen Methoden auf die Leistung und Effizienz von Systemen - Zukunftsaussichten für fortgeschrittene Skalierungsmethoden.

---

<sup>10</sup>Kaiming HE u. a. [2016]. »Deep residual learning for image recognition«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, S. 770–778.

# Kapitel 5

## Evaluation von Skalierungsmethoden

### 5.1 Qualitätsmetriken zur Bewertung von Skalierungsmethoden

#### 5.1.1 Peak Signal-to-Noise Ratio (PSNR)

- Definition und Berechnung von PSNR - Anwendung von PSNR bei der Bewertung von Bildqualität - Limitationen von PSNR

#### 5.1.2 Structural Similarity Index (SSIM)

- Definition und Berechnung von SSIM - Anwendung von SSIM bei der Bewertung von Bildqualität - Vorteile von SSIM im Vergleich zu PSNR - Limitationen von SSIM

#### 5.1.3 Computational Speed

- Bedeutung von Computational Speed bei der Wahl einer Skalierungsmethode - Methoden zur Messung von Computational Speed - Trade-offs zwischen Bildqualität und Computational Speed

#### 5.1.4 Root-Mean-Square Error (RMSE)

- Definition und Berechnung von RMSE - Anwendung von RMSE bei der Bewertung von Bildqualität - Limitationen von RMSE

### 5.2 Kriterien zur Wahl der besten Methode

#### 5.2.1 Mean Opinion Score (MOS)

- Definition von MOS und dessen Bedeutung bei der Bewertung von Bildqualität - Anwendung von MOS bei der Evaluierung von Skalierungsmethoden - Limitationen von MOS -

Vergleich von MOS mit anderen Qualitätsmetriken

### **5.2.2 Bildvergleich und visuelle Bewertung**

- Visuelle Bewertung von Bildern durch menschliche Beobachter - Methoden zur Bildvergleich (z.B. Side-by-Side-Vergleich, Triple-Stimulus-Test) - Anwendung von visueller Bewertung bei der Evaluierung von Skalierungsmethoden - Limitationen und Vorbehalte bei der visuellen Bewertung von Bildern

### **5.2.3 Effektivität und Effizienz**

- Effektivität (Bildqualität) vs. Effizienz (Computational Speed) - Evaluierung der Trade-offs zwischen Effektivität und Effizienz - Anwendung von Kosten-Nutzen-Analysen bei der Wahl der besten Skalierungsmethode - Relevanz von Effektivität und Effizienz in verschiedenen Anwendungsbereichen

### **5.2.4 Zukunftsaussichten und Herausforderungen**

- Technologische Entwicklungen und deren Auswirkungen auf die Evaluierung von Skalierungsmethoden - Herausforderungen bei der Evaluierung von Skalierungsmethoden in spezifischen Anwendungsbereichen (z.B. medizinische Bildgebung, Videokompression) - Potenziale von KI-basierten Evaluierungsmethoden - Ethische und soziale Implikationen der Bewertung von Skalierungsmethoden

# Kapitel 6

## Zusammenfassung und Ausblick

6.1 Auswertung der Ergebnisse

6.2 Diskussion offener Fragen und zukünftiger Forschungsbedarf

# Kapitel 7

## Einführung in Deep Learning

Deep Learning ist eine Unterkategorie des maschinellen Lernens und beschäftigt sich mit der Entwicklung und Anwendung von neuronalen Netzwerken mit mehreren Schichten, um komplexe Probleme zu lösen. Es verwendet künstliche Intelligenz, um aus großen Datenmengen Muster zu erkennen und Entscheidungen zu treffen. In der heutigen Welt hat Deep Learning Anwendungen in Bereichen wie Spracherkennung, Bilderkennung, Robotik, Medizin und Autonomen Fahrzeugen gefunden.

Die Geschichte des Deep Learning reicht zurück bis in die 1940er Jahre, aber erst in den 2000er Jahren wurden die Algorithmen und Computerleistung ausreichend entwickelt, um Deep Learning erfolgreich anzuwenden. Im Jahr 2012 gewann ein Deep Learning-Modell namens AlexNet den ImageNet-Wettbewerb, was als Durchbruch für die Anwendung von Deep Learning in Computer Vision gilt.

Traditionelles maschinelles Lernen verwendet in der Regel flache neuronale Netzwerke, die nur eine Schicht haben, um Daten zu analysieren. Im Gegensatz dazu verwenden Deep-Learning-Modelle mehrere Schichten, um komplexere Merkmale der Daten zu identifizieren. Dies ermöglicht eine höhere Genauigkeit und Effektivität bei der Lösung komplexer Probleme.

Beispiele für reale Anwendungen von Deep Learning sind unter anderem die Bilderkennung in sozialen Medien, die Spracherkennung in Smart Speakern wie Amazon Echo und Google Home, sowie die automatisierte Diagnose von medizinischen Bildern wie CT-Scans und MRT-Bildern.

### 7.1 Motivation hinter der Studie von CNNs und deren Training

Traditionelle maschinelle Lernalgorithmen sind nicht effektiv bei der Verarbeitung komplexer Datentypen wie Bildern. Hier kommen Convolutional Neural Networks (CNNs) ins Spiel. CNNs sind eine spezielle Art von neuronalem Netzwerk, die speziell für die Bilderkennung und Computer Vision entwickelt wurden.

CNNs arbeiten durch die Verwendung von Filtern, die über das Eingabebild geschoben werden, um Merkmale wie Kanten, Farben und Texturen zu extrahieren. Diese Merkmale

werden dann von Schichten von Neuronen verwendet, um die Merkmale zu kombinieren und eine Vorhersage zu treffen.

Eine der größten Herausforderungen beim Training von CNNs ist das Overfitting, bei dem das Modell zu stark auf die Trainingsdaten optimiert wird und dadurch bei neuen Daten schlecht abschneidet. Eine weitere Herausforderung ist das Problem der verschwindenden Gradienten, bei dem die Gradienten, die zur Anpassung der Gewichte verwendet werden, während des Trainings immer kleiner werden und das Modell nicht mehr lernen kann.

Um diese Probleme zu lösen, werden Optimierungsalgorithmen wie der stochastische Gradientenabstieg verwendet, um die Gewichte des Modells anzupassen und das Modell an die Daten anzupassen. Es gibt auch Techniken wie Dropout und Data Augmentation, die dazu beitragen können, Overfitting zu reduzieren.

Es ist wichtig, CNNs zu verstehen und effektiv zu trainieren, um die volle Leistungsfähigkeit von Deep Learning in der Bilderkennung und Computer Vision zu nutzen.

## 7.2 Struktur des Papiers

Dieses Papier ist in drei Hauptabschnitte unterteilt. Im ersten Abschnitt wird eine Einführung in Deep Learning gegeben, einschließlich der Bedeutung von Deep Learning in der heutigen Welt, der Geschichte und Entwicklung des Deep Learning, der Unterschiede zwischen Deep Learning und traditionellem maschinellem Lernen und Beispielen für reale Anwendungen von Deep Learning.

Im zweiten Abschnitt wird die Motivation hinter der Studie von CNNs und deren Training erläutert. Dabei werden die Einschränkungen von traditionellen maschinellen Lernalgorithmen bei der Verarbeitung komplexer Datentypen wie Bildern diskutiert und die Bedeutung von CNNs bei der Bilderkennung und Computer Vision erklärt. Weiterhin werden die Herausforderungen beim Training von CNNs wie Overfitting und verschwindende Gradienten sowie die Rolle von Optimierungsalgorithmen wie dem stochastischen Gradientenabstieg beim Training von CNNs beschrieben.

Im dritten Abschnitt werden die Ergebnisse und Beiträge dieses Papiers skizziert. Dies beinhaltet eine Zusammenfassung der wichtigsten Erkenntnisse und Empfehlungen für das effektive Training von CNNs.





## Kapitel 8

# Basics of Convolutional Neural Networks

### 8.1 How do Machine learning models learn?

#### 8.1.1 Definition Machine Learning and Deep Learning

#### 8.1.2 Definition and applications of CNNs

#### 8.1.3 Applications of CNNs

#### 8.1.4 How are CNNs different from other neural network architectures?

#### 8.1.5 Why are CNNs particularly useful for image and video data?

### 8.2 Applications of CNNs

#### 8.2.1 Image classification

#### 8.2.2 Object detection

#### 8.2.3 Face recognition

#### 8.2.4 Natural Language Processing

#### 8.2.5 Other applications

### 8.3 How do CNNs work?

### 8.4 Architecture of CNNs

#### 8.4.1 Input layer

#### 8.4.2 Hidden layers

#### 8.4.3 Output layer

### 8.5 Convolutional layers

# Kapitel 9

## Data Preprocessing

### 9.1 What is data?

### 9.2 Importance of the right data

#### 9.2.1 Data cleaning

#### 9.2.2 Data normalization

#### 9.2.3 Data augmentation

# Kapitel 10

## Convolutional Neural Network Training Process

10.1 What is training?

10.2 Training process

10.3 Training process for CNNs

10.4 Stochastic Gradient Descent (SGD)

10.5 Backpropagation

10.6 Hyperparameter tuning

10.7 Regularization techniques

# Kapitel 11

## Transfer Learning

11.1 Introduction to Transfer Learning

11.2 Fine-tuning of pre-trained models

11.3 Using pre-trained models as feature extractors

# Kapitel 12

## Common Challenges and Solutions

12.1 Overfitting and underfitting

12.2 Vanishing and exploding gradients

12.3 Gradient descent optimization

12.4 Solutions to common challenges

# Kapitel 13

## Tools and Frameworks for CNN Training

13.1 PyTorch

13.2 TensorFlow

13.3 Keras

13.4 Caffe

13.5 Other popular frameworks

# Kapitel 14

## Conclusion and Future Work

14.1 Summary of the paper

14.2 Key takeaways

14.3 Future research directions

# Abkürzungsverzeichnis

<b>PPIs</b>	Pixel per Inch . . . . .	5
-------------	--------------------------	---



# Abbildungsverzeichnis

5figure.1.1	
1.2	Verschiedene Beispiele von upscaling Algorithmen[WHUBER 2011]. . . . . 6
3.1	Beispielgrafik zur Pixelverdopplung. . . . . 17
3.2	Graph über bilineare Skalierung. . . . . 17

# Tabellenverzeichnis

# Liste der Algorithmen

# Formelverzeichnis

# Literatur

- ALBER, Mark u. a. [2019]. »Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences«. In: *NPJ digital medicine* 2.1, S. 115 [siehe S. 29].
- BENTBIB, A.H. u. a. [2016]. »A global Lanczos method for image restoration«. In: *Journal of Computational and Applied Mathematics* 300, S. 233–244. ISSN: 0377-0427. DOI: <https://doi.org/10.1016/j.cam.2015.12.034>. URL: <https://www.sciencedirect.com/science/article/pii/S0377042715006469> [siehe S. 22].
- BOUTELL, Thomas [1997]. *Png (portable network graphics) specification version 1.0*. Techn. Ber. [siehe S. 9].
- BOUTELL, Thomas u. a. [o. D.] *PNG (Portable Network Graphics Format) Version 1.0*. Techn. Ber. RFC 2083 [siehe S. 9].
- BURGER, Wilhelm und Mark James BURGE [2009]. *Digitale Bildverarbeitung: Eine Algorithmische Einführung Mit Java*. Springer-Verlag [siehe S. 9].
- BURGER, Wilhelm u. a. [2015]. »Digitale Bilder«. In: *Digitale Bildverarbeitung: Eine algorithmische Einführung mit Java*, S. 1–24 [siehe S. 9].
- BURT, Peter J und Edward H ADELSON [1987]. »The Laplacian pyramid as a compact image code«. In: *Readings in computer vision*. Elsevier, S. 671–679 [siehe S. 30].
- CONTRIBUTORS, Wikipedia [2023]. *Bildskalierung Definition*. [https://en.wikipedia.org/wiki/Image\\_scaling](https://en.wikipedia.org/wiki/Image_scaling) [siehe S. 6].
- DUCHON, Claude E [1979]. »Lanczos filtering in one and two dimensions«. In: *Journal of Applied Meteorology and Climatology* 18.8, S. 1016–1022 [siehe S. 21].
- ELSEVIER [n.d.] *Artwork and media instructions*. <https://www.elsevier.com/authors/policies-and-guidelines/artwork-and-media-instructions/artwork-overview>. Accessed: March 9, 2023 [siehe S. 11].
- ENCYCLOPEDIA BRITANNICA [n.d.] *JPEG*. <https://www.britannica.com/technology/JPEG>. Accessed: March 9, 2023 [siehe S. 11].
- FIBINGER, Iris [2002]. *SVG. Scalable Vector Graphics.: Praxiswegweiser und Referenz für den neuen Vektorgrafikstandard. Für Fortgeschrittene*. Markt+Technik-Verl. ISBN: 3827262399,9783827262394. URL: <http://gen.lib.rus.ec/book/index.php?md5=1a655693c0779b14c519078151dfdf31> [siehe S. 9].

- GRIBBON, K.T. und D.G. BAILEY [2004]. »A novel approach to real-time bilinear interpolation«. In: *Proceedings. DELTA 2004. Second IEEE International Workshop on Electronic Design, Test and Applications*, S. 126–131. DOI: 10.1109/DELTA.2004.10055 [siehe S. 18].
- HE, Kaiming u. a. [2016]. »Deep residual learning for image recognition«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, S. 770–778 [siehe S. 31].
- HUANG, Gao u. a. [2017]. »Densely connected convolutional networks«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, S. 4700–4708 [siehe S. 29].
- IONOS [n.d.] *Graphic File Formats: Which Formats Are Important?* <https://www.ionos.com/digitalguide/websites/web-design/graphic-file-formats-which-formats-are-important/>. Accessed: March 9, 2023 [siehe S. 11].
- JADERBERG, Max, Karen SIMONYAN, Andrew ZISSERMAN u. a. [2015]. »Spatial transformer networks«. In: *Advances in neural information processing systems* 28 [siehe S. 25].
- JIANG, Nan und Luo WANG [2015]. »Quantum image scaling using nearest neighbor interpolation«. In: *Quantum Information Processing* 14, S. 1559–1571 [siehe S. 16].
- LI, Hao u. a. [2018]. »Visualizing the loss landscape of neural nets«. In: *Advances in neural information processing systems* 31 [siehe S. 25].
- LIU, Ziwei u. a. [2015]. »Deep learning face attributes in the wild«. In: *Proceedings of the IEEE international conference on computer vision*, S. 3730–3738 [siehe S. 25].
- MALLAT, Stephane G [1989]. »A theory for multiresolution signal decomposition: the wavelet representation«. In: *IEEE transactions on pattern analysis and machine intelligence* 11.7, S. 674–693 [siehe S. 29].
- MARCELLIN, Michael W u. a. [2000]. »An overview of JPEG-2000«. In: *Proceedings DCC 2000. Data Compression Conference*. IEEE, S. 523–541 [siehe S. 11].
- MOZILLA CONTRIBUTORS [n.d.] *SVG (Scalable Vector Graphics)*. <https://developer.mozilla.org/en-US/docs/Web/SVG>. Accessed: March 9, 2023 [siehe S. 10].
- NAH, Seungjun, Tae HYUN KIM und Kyoung MU LEE [2017]. »Deep multi-scale convolutional neural network for dynamic scene deblurring«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, S. 3883–3891 [siehe S. 30].
- NASA/JPL-CALTECH [Feb. 1990]. *The Pale Blue Dot is a photograph of Earth taken Feb. 14, 1990, by NASA's Voyager 1 at a distance of 6 billion kilometers from the Sun. The image inspired the title of scientist Carl Sagan's book, "Pale Blue Dot: A Vision of the Human Future in Space," in which he wrote: "Look again at that dot. That's here. That's home. That's us."* <https://solarsystem.nasa.gov/resources/536/voyager-1-pale-blue-dot/> [siehe S. 5].

- OEHLRICH, Carl-Werner u. a. [1992]. »Ein Transputersystem mit verteiltem Bildspeicher für Echtzeit-Computergrafik und Bildverarbeitung«. In: *Parallele Datenverarbeitung mit dem Transputer: 3. Transputer-Anwender-Treffen TAT'91, Aachen, 17.–18. September 1991*. Springer, S. 170–177 [siehe S. 12].
- PREPRESSURE [n.d.] *Prepressure Library: File Formats*. <https://www.prepressure.com/library/file-formats/>. Accessed: March 9, 2023 [siehe S. 11].
- QUINT, Antoine [2003]. »Scalable vector graphics«. In: *IEEE MultiMedia* 10.3, S. 99–102 [siehe S. 10].
- RABBANI, Majid und Rajan JOSHI [2002]. »An overview of the JPEG 2000 still image compression standard«. In: *Signal processing: Image communication* 17.1, S. 3–48 [siehe S. 11].
- SIMONCELLI, Eero P und Edward H ADELSON [1996]. »Noise removal via Bayesian wavelet coring«. In: *Proceedings of 3rd IEEE International Conference on Image Processing*. Bd. 1. IEEE, S. 379–382 [siehe S. 30].
- TECH-LIB [2020]. *Image Scaling Definition*. <http://www.dante.de> [siehe S. 6].
- WANG, David C.C, Anthony H VAGNUCCI und C.C LI [1983]. »Digital image enhancement: A survey«. In: *Computer Vision, Graphics, and Image Processing* 24.3, S. 363–381. ISSN: 0734-189X. DOI: [https://doi.org/10.1016/0734-189X\(83\)90061-0](https://doi.org/10.1016/0734-189X(83)90061-0). URL: <https://www.sciencedirect.com/science/article/pii/0734189X83900610> [siehe S. 14].
- WHUBER [Sep. 2011]. *What is Lanczos resampling?* <https://gis.stackexchange.com/a/14361> [siehe S. 6].
- WU, Bichen u. a. [2017]. »Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving«. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, S. 129–137 [siehe S. 25].

## Liste der ToDo's