TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK
INSTITUT FÜR SOFTWARE- UND MULTIMEDIATECHNIK
PROFESSUR FÜR COMPUTERGRAPHIK UND VISUALISIERUNG
PROF. DR. STEFAN GUMHOLD

Großer Beleg

Parameterrekonstruktion für Röntgen-Kleinwinkelstreuung mit Deep Learning

Patrick Stiller
(Mat.-Nr.: 3951290)

Betreuer: Dr. Dmitrij Schlesinger(TU Dresden), Dr. Heide Meißner(HZDR), Dr. Michael Bussmann(HZDR)

Dresden, 28. Januar 2019

Aufgabenstellung

aufgabenstellung

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag dem Prüfungsausschuss der Fakultät Informatik eingereichte Arbeit zum Thema:

Parameterrekonstruktion für Röntgen-Kleinwinkelstreuung mit Deep Learning

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 28. Januar 2019

Patrick Stiller

Inhaltsverzeichnis

1 Einleitung

2 Grundlagen

2.1 Physikalischer Hintergrund

2.1.1 Kleinwinkelstreuung

Die Kleinwinkelstreuung in Englisch Small Angle X-Ray Scattering (SAXS) ist eine universelle Technik zur Untersuchung von Feststoffen. Dabei kann SAXS Informationen über die kristalline Struktur, chemische Komposition und die physikalische Eigenschaften eines untersuchten Feststoffes liefern [Cro]. Die Untersuchung von Feststoffen unter Einfluss von Hochintensitätslasern ist ein Schwerpunkt der heutigen Physik. Wissen über das Verhalten von Feststoffen unter Einfluss von Hochintensitätslasern kann offene Fragen in der Krebsforschung und in der Astrophysik beantworten. Dieser große Beleg bezieht sich auf die Publikation von Thomas Kluge, welche 2018 mit dem Titel:“Observation of ultrafast solid-density plasma dynamics using femtosecond X-ray pulses from a free-electron laser“veröffentlicht wurde. In der angesprochenen Publikation wurde SAXS für die Untersuchung von Plasma eingesetzt [KRM$^+$18]. Bisher konnte die Plasmadynamik, welche bei der Interaktion eines Feststoffes und eines Hochintensitätslasern entsteht, nur durch Simulationen erfasst werden mit Hilfe von SAXS gibt es neue Möglichkeiten, die Laser-Plasma-Interaktion auf einem Feststoff zu charakterisieren. SAXS erreicht außerdem eine räumliche Auflösung im Femtosekunden-Bereich und eine zeitliche Auflösung im Nanosekunden-Bereich. Mit SAXS ist es möglich anhand der Einstrahlung auf den Detektor mit Hilfe von numerischen Simulationen Aussagen über die Elektronendynamik im Plasma zu treffen [KRM$^+$18].

2.1.2 Experimentbeschreibung

Das in [KRM$^+$18] beschriebene Experiment besteht aus vier Hauptkomponenten (siehe Abbildung 2.1 links): Dem Target(Mitte), dem Hochintensitätslasern (UHI) (Links), dem Röntgen-Freie-Elektronen-Laser (XFEL) (Links Unten) und dem Detektor (Rechts Oben). Der Hauptbestandteil des Targets ist Kupfer. Um eine analytische Beschreibung des Experimentausgangs zu ermöglichen, wurde eine Gitterstruktur in das Kupfertarget eingraviert. Zusätzlich wurde das Taget mit einer 2 μm breiten Siliziumschicht überzogen. Durch die eingravierte Gitterstruktur besitzt das Target einen eindimensionalen

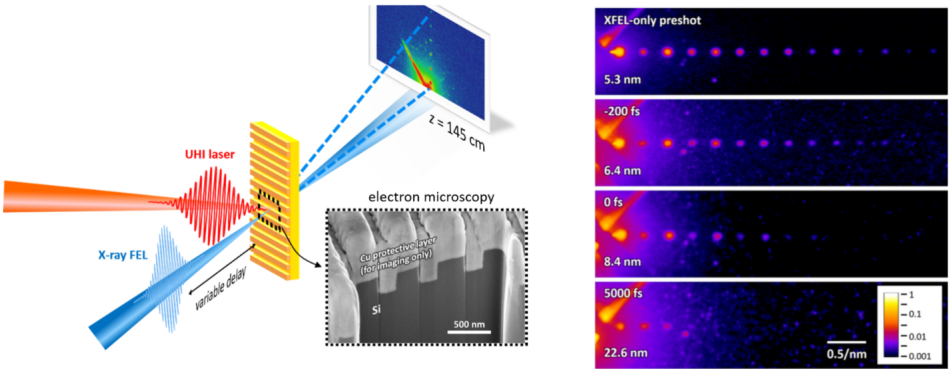

Abbildung 2.1: Links: Schematischer Aufbau des Experimentaufbaus. Rechts: der Abbildung sind Detektorbilder in Abhängigkeit der Bestrahlungsdauer des Hochintensitätslasern dargestellt. Bildquelle: [KRM+18]

Informationsgehalt und kann somit durch drei Parameter vollständig beschrieben werden. Die drei Paramter sind Pitch, Feature-Size und die Aufweichungsbreite Sigma(σ). Dabei beschreiben Pitch und Feature-Size die Struktur des Targets und Sigma(σ) den Aufweichungseffekt des Hochintensitätslasers (siehe Abbildung 2.2). Die Gesamtheit des Targets wird auch als Grating bezeichnet. Erhebungen des Gratings werden als Features bezeichnet. Im weiteren Verlauf dieser Arbeit wird der Begriff Elektronendichteverteilung η verwendet, welcher oft im Zusammenfang des SAXS-Ansatzes fällt. Der Begriff Elektronendichteverteilung η wird in dieser Arbeit als Synonym für das Grating verwendet.

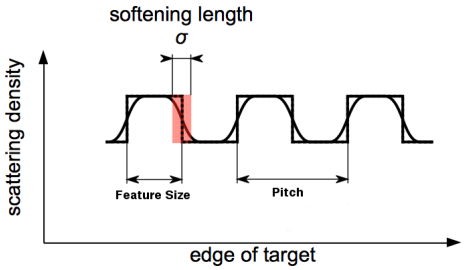

Abbildung 2.2: Analytische Beschreibung des Querschnittes des Targets. Dabei ist das Grating zu erkennen und dass es durch drei Parameter: Pitch, Feature-Size und Sigma beschrieben werden kann. Bildquelle: [Zac17]

Während des Experiments wird das Target durch einen Hochintensitätslasers in einem Winkel von 90° beschossen. Augrund des elektrischen Feldes des Lasers entsteht Plasma. Bei Plasma handelt es sich um ein Gemisch aus freien Elektronen, positiven Ionen und neutralen Teilchen, welche unter ständiger Wechselwirkung miteinander und mit Photonen stehen. Dadurch kann es zu unterschiedlichen Energie- bzw. Anregungszuständen kommen. Der Plasmazustand eines Stoffes wird auch als vierter Aggregatzustand

bezeichnet [Wis]. Während des Beschusses durch den Hochintensitätslasers ist außerdem ein Schmelzprozess und somit eine Aufweichung der Gitterstruktur des Targets wahrzunehmen (siehe Abbildung 2.3). Das durch den Hochintensitätslasers erzeugte Plasma soll auf seine Struktur und Elektrodynamik mit Hilfe von SAXS untersucht werden. Dafür wird leicht zeitversetzt ein zweiter Laser benutzt. Dabei handelt es sich um einen Röntgen-Freie-Elektronen-Laser, welcher in einem Winkel von 45° mit einer Pulsdauer von 40 Femtosekunden auf das Target schießt. Bei diesem Vorgang kommt es zu einer Streuung des Lichts des Röntgenlasers an den Elektronen des Targets. Die Lichtintensitäten des gestreuten Röntgenlasers werden durch einen Detektor, welches hinter dem Target platziert ist, gemessen (Abbildung 2.1 links).

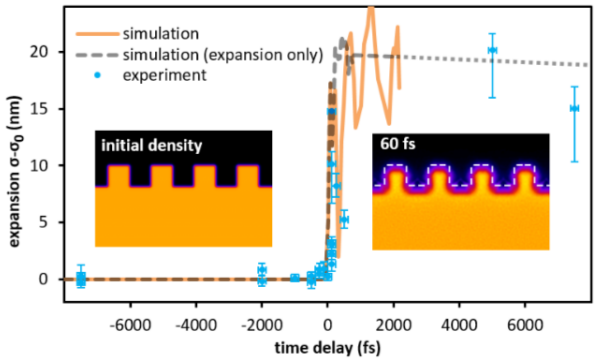

Abbildung 2.3: Veränderung des Gratings nach einer Bestrahlungsdauer von 60 Femtosekunden Bildquelle: [KRM+18]

Beim Detektor handelt es sich um ein Raster von Lichtdetektoren, welche die ankommenden Lichtintensitäten messen. Dabei integriert der Detektor über die Zeit, das heißt, dass ankommende Lichtintensitäten über die Zeit summiert werden. Dieser Effekt ist in der Abbildung 2.4 zu erkennen, welche einen beispielhaften Streuprozess an zwei Elektronen zeigt.

Durch die Zeitintegration des Detektors, gehen die zeitlichen Abstände der Wellen verloren (Phase). Das entstandene Detektorsignal skaliert mit dem Betragsquadrat der Fouriertransformation der Gitterstruktur η (siehe Gleichung (2.1)) [KRM+18].

$$\Phi \propto \left| \int \eta(\vec{r}) \cdot e^{i\vec{q}\vec{r}} d\vec{r} \right|^2 \tag{2.1}$$

2.2 Problemidentifikation

Eine der Herausforderungen des beschriebenen Experimentes ist die Zeitintegration des Detektors. Diese Detektoreigenschaft ruft den Verlust der Phase hervor, sodass eine Rekonstruktion der Elektronendichte-

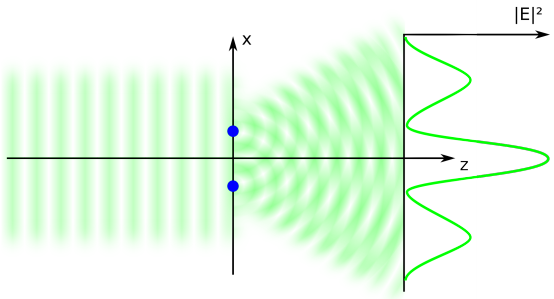

Abbildung 2.4: Beispielhafte Darstellung einer Streuung an zwei Elektronen. Die Röntgenlaserpulse treffen geradlinig auf die Elektronen und werden dann bei den Elektronen gestreut. Der Detektor dahinter misst die ankommenden Intensitäten der kreisförmigen Wellen und summiert diese über die Zeit. Bildquelle: [Zac17]

verteilung η mit Hilfe einer inversen Fouriertransformation (IFT) nicht möglich ist. Um dieses Problem zu lösen, werden iterative Algorithmen, sogenannte Phaseretrieval-Algorithmen verwendet. Beispiele für solche Algorithmen sind [Fie82]: Error-Reduction Algorithm, Gradient Search Methods und der Input-Output Algorithmus. Probleme bei diesen Algorithmen sind, dass erstens bei allen Algorithmen keine Konvergenz garantiert ist und zweitens sehr viele Iterationen notwendig sind um eine Optimierung der errechneten Phase zu erzeugen. Diese Eigenschaften führen dazu, dass eine hochfrequente Verarbeitung von Experimentergebnissen verwehrt bleibt. Deswegen soll mit Hilfe von Deep Learning die Charakterisierung der Elektronendichteverteilung η (Parameter zur Beschreibung der Gitterstruktur) rekonstruiert werden. Neuronale Netze haben den Vorteil, dass sie sich im Gegensatz zu iterativen Verfahren parallelisieren und durch GPUs beschleunigen lassen und die Parameter der Elektronendichtenverteilung nicht-iterativ bestimmt werden (Einschrittverfahren).

2.3 Fouriertransformation

Die Fouriertransformation (benannt nach Jean Baptiste Joseph Fourier) ist eine Transformation, welche zeitbezogene Wellen im Ortsraum in ihre frequenzmäßigen Spektralanteile zerlegt (Abbildung **??**). Die Fouriertransformation wird auch als Transformation vom Orts- in den Frequenzraum bezeichnet. Die Fouriertransformation F(u) einer Welle f(x) ist folgendermaßen definiert:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x u} dx \tag{2.2}$$

Die Spektralanteile *F(u)* werden wie bei einem Basiswechsel durch ein Integral bestimmt. Dabei ist das Ergebnis des Integrals die Länge der Projektion in Richtung der durch u kodierten Frequenz. Im endlich-diskreten Fall wird das Integral durch eine Summe bis zur Anzahl der zu verwendeten Wellen ersetzt. Jede Stelle u der Fouriertransformation kodiert eine Welle $e^{-2\pi i x u}$, welche über die Euler-Identität $e^{ikx} = \cos(kx) + i \cdot \sin(kx)$ auch in ihren Kosinus- und Sinusanteilen beschrieben werden kann. Der Funktionswert der Fouriertransformation *F(u)* kodiert mit welchem Anteil die durch *u* kodierte Frequenz verwendet wird. Das Ergebnis der Fouriertransformation kann auch durch das Amplituden- und durch das Phasenspektrum beschrieben werden.. Dabei bestimmt die Amplitude das Maximum und das Minimum der jeweiligen Welle und die Phase die Verschiebung der jeweiligen Welle. Um die Amplituden- $|F(u)|$ und die Phaseninformation $\phi(u)$ einer komplexen Zahl u zu errechnen, werden folgende Gleichungen verwendet [Sch]:

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \tag{2.3}$$

$$\phi(u) = \tan^{-1} \frac{I(u)}{R(u)} \tag{2.4}$$

2.4 Filter

Ein Filter ist eine Operation, welche auf einem Signal verwendet wird, um Signale zu glätten, Signalstörungen zu vermeiden, oder um Rauschen zu verhindern. In den meisten Fällen wird die Filteroperation mit Hilfe von Faltung realisiert. Die Faltung zweier Funktionen $(f * g)$ durch den funktionalen Zusammenhang in Formel (2.5) beschrieben. Dabei ist f die Funktion, welche das Signal beschreibt und g die Funktion, welche den Filter beschreibt. Im diskreten Fall wird das Integral durch eine Summe bis zur entsprechenden Filtergröße ersetzt [Kla13].

$$(f * g)(x) := \int_{\mathbb{R}^n} f(\tau) g(x - \tau) \mathrm{d}\tau \tag{2.5}$$

2.5 Neuronal Networks

Neuronal Networks sind eine mathematische Adaption des realen menschlichen Gehirns. Wie im realen menschlichen Gehirn sind Neuronen miteinander verbunden, um einen Informationsfluss zu gewährleisten. Die ersten Ansätze für neuronale Netze wurden bereits 1943 von Warren McCulloch und Walter Pitts entwickelt [Kri07]. Heute sind Neuronale Netze der Schwerpunkt des Machine Learnings und werden auf großen Datenmengen angewendet, um ein gewünschtes Verhalten zu trainieren .

2.5.1 Aufbau

Ein Neuronal Network besteht aus vielen kleinen Komponenten, Neuronen, welche durch gerichtete und gewichtete Verbindungen verbunden sind. Sämtliche Beschreibungen und Notationen beziehen sich auf [Kri07]. Mathematisch definiert ist ein Neuronal Network als ein Tripel (N, V, w) mit den Beiden Mengen N und V und der Funktion w. N ist die Menge aller Neuronen und V = $\{(i,j)|i,j\epsilon\mathbb{N}\}$ die Menge der Verbindungen zwischen Neuron i und Neuron j. Die Funktion $w : V \longrightarrow \mathbb{R}$ beschreibt die Gewichte des Neuronalen Netzes. Wobei w(i,j) das Gewicht zwischen dem Neuron i und Neuron j beschreibt. Im Allgemeinen wird anstatt der Funktionsnotation die Notation $w_{i,j}$ für die Gewichte zwischen zwei Neuronen verwendet. Die nächste wichtige Komponente ist die Propagierungsfunktion net_j eines Neurons, welche einen wichtigen Teil des Informationsflusses in einem Neuronalen Netz definiert. Dabei wird der Input des Neurons j durch dessen Propagierungsfunktion net_j bestimmt. Die Propagierungsfunktion net_j nimmt den Output aller Neuronen welche eine ausgehende Verbindung zum Neuron j besitzen als Input. So wird die Propagierungsfunktion net_j durch folgenden funktionalen Zusammenhang beschrieben :

$$net_j = \sum_{i \in I_j} (o_i \cdot w_{i,j}) \text{ mit } I_j = \{i\epsilon N|(i,j)\epsilon V\} \tag{2.6}$$

Der Output o_j eines Neurons j wird mit Hilfe der Aktivierungsfunktion a_j und der der Propagierungsfunktion net_j berechnet. Dazu wird noch der Schwellwert θ_j zur Hemmung des Aktivierungszustandes des Neurons zur Hilfe genommen. Somit ergibt sich für den Output des Neurons folgender funktionaler Zusammenhang:

$$o_j = a_j(net_j - \theta_j) = f(x) \tag{2.7}$$

In den meisten Fällen werden differenzierbare Aktivierungsfunktionen benutzt, da sie den Lernprozess des neuronalen Netzes erleichtern. Für diesen Beleg ist die Rectified Linear Units- Aktivierungsfunktion (Formel (2.8)) relevant.

$$f(x) = max(0, x) \tag{2.8}$$

Die ReLu-Aktivierungsfunktion ist im Vergleich anderen Aktivierungsfunktionen schneller zu berechnen und bietet größere Gradienten. Jedoch können Neuronen, welche einen negativen Input bekommen nur noch 0 Ausgeben. Es wird in diesem Zusammenhang von einem gestorbenen Neuron gesprochen. Ein weiterer Nachteil der ReLu-Aktivierungsfunktion ist der Wertebereich, denn dieser ist nach oben nicht eingeschränkt. Somit können in einem Neuronalen Netz sehr große Werte entstehen, welche den Wertebereich von Zahlenstandards wie zum Beispiel IEEE 754 (Single Precision Float 32-Bit) [Kah97] überschreiten, womit kein valider Datenfluss im Neuronalen Netz gegeben ist.

2.5.2 Feed-Forward Neuronal Networks

Für ein Neuronal Network können verschiedene Netzwerktopologien werden. Ein Beispiel dafür sind Feed Forward Neuronal Networks (Abbildung 2.5).Bei einem Feed Forward Neuronal Network werden die Neuronen als Schichten (Layer) angeordnet. Diese Layer sind miteinander verbunden. Die erste Layer, welche die Input Daten bekommt, wird als Input-Layer bezeichnet. Die letzte Schicht, welche die Netzberechnung ausgibt, wird als Output-Layer bezeichnet. Schichten, welche sich zwischen Input- und Output-Layer befinden und somit keinen Kontakt nach Außen haben, werden als Hidden Layer bezeichnet. Ein wichtiges Merkmal von Feed-Forward Neuronal Networks ist, dass der Datenfluss geradlinig vom Input-Layer über die Hidden-Layer zum Output-Layer ohne Rückkopplung verläuft.

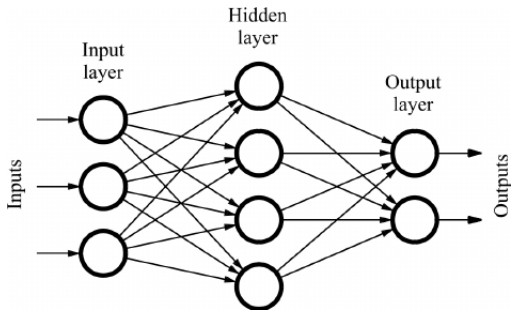

Abbildung 2.5: Beispielhafte Darstellung eines Feed-Forward Neuronal Networks. Bildquelle: [QR11]

2.5.3 Lernprozess

Das Anlernen von Neuronalen Netzen wird in den meisten Fällen durch überwachtes Lernen (supervised Learning) realisiert. Im Speziellen wird der Lernprozess durch den Backpropagation-Algorithmus und Gradientenabstiegsverfahren durchgeführt. Bei einem überwachten Lernansatz besteht der Daten-

satz zum Trainieren des Neuronalen Netzen aus zwei Teilen. Zu jedem Netzinput x_i gibt es ein zugehöriges Label y_i, welches den gewünschten Netzoutput definiert. Der Lernprozess eines neuronalen Netzes kann in drei Phasen eingeteilt werden: Dem Forward-Pass, die Loss-Calculation und dem Backward Pass [Bec]. Zu Beginn des Trainingsprozesses werden die Gewichte des Neuronalen Netzes mit Zufallszahlen initialisiert. In vielen Implementationen werden die Gewichte zufällig und normal verteilt initialisiert. Beim Forward Pass werden die Input-Daten zur Kalkulation des derzeitigen Netz-Outputs zum Input-Layer des Neuronalen Netzes gegeben. Das Neuronale Netz bestimmt dann durch die Rechenvorschriften des Neuronalen Netzes den Netz-Output $\hat{y}_i$. Im zweiten Schritt wird die Qualität des Netz-Ouputs $\hat{y}_i$ bestimmt.. Dazu wird ein Fehlermaß benutzt, das den Grad des Unterschiedes zwischen $\hat{y}_i$ und y_i bestimmt. Eine beispielhafte Errorfunktion ist Mean-Squared Error (2.9).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{2.9}$$

Im dritten Schritt, dem Backward-Pass, kommt es zur Optimierung des Neuronalen Netzes. Mithilfe des Fehlers, welcher bis zur Eingabeschicht zurück propagiert wird, werden die Gewichte entsprechend ihres Einflusses auf den Net-Output (2.11) mittels Gradientenabstiegsverfahrens angepasst. Für die Anwendung des Gradientenabstiegsverfahrens werden die partiellen Ableitungen des Fehlerterms benötigt. Die Gewichtsveränderung Δw_{ij} des Gewichts zwischen Neuron i und Neuron j ergibt sich durch folgenden unktionalen Zusammenhang :

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \delta_j o_i \tag{2.10}$$

Dabei ist *E* die Errorfunktion, δ_j des Gradient Neurons j, o_i die Ausgabe des Neurons i. Der Parameter η , welcher die Learning-Rate des Gradientenabstiegsverfahren bestimmt. Schlussendlich fehlt die Definition des Gradienten. Dieser ist davon abhängig, wie stark ein Neuron den Output des Neuronalen Netzes beeinträchtigt. Deswegen wird eine Unterscheidung getroffen, ob ein Neuron sich im Output-Layer oder dem Hidden- bzw. Input-Layer befindet. In der folgenden Gleichung (2.11) ist die Definition des Gradienten und die zugehörige Propagierung des Fehlers.

$$\delta_j = \begin{cases} a_j\,(\text{ net }_j)\,(o_j - t_j) & \text{falls } j \text{ sich in der Output-Layer befindet} \\ a_j\,(\text{ net }_j)\sum_k \delta_k w_{jk} & \text{falls } j \text{ verdecktes Neuron oder ein Input-Neuron ist} \end{cases} \tag{2.11}$$

Dabei ist die Gleichung (2.11) in Abhängigkeit von den Variablen o_j , dem Output des Neurons j, t_j, die Soll-Ausgabe des Neurons j und der Aktivierungsfunktion a_j des Neurons j angegeben. Somit ergibt sich das neue Gewicht durch eine Addition des alten Gewichts mit der errechneten Gewichtsveränderung.

$$w_{ij}^{\text{neu}} = w_{ij}^{\text{alt}} + \Delta w_{ij} \tag{2.12}$$

Der Backpropagation-Algorithmus wird iterativ solange angewandt, bis eine bestimmte Anzahl von Iterationen erreicht ist oder andere Kriterien erfüllt sind [Wik18].

2.5.4 Convolutional Neuronal Networks

Klassische Neuronal Networks besitzen die Einschränkung, dass deren Inputs als Vektoren geliefert werden müssen. Soll ein klassisches Neuronal Network zum Beispiel ein Bild der Größe $N \times M \times C$ (C steht für Kanäle) verarbeiten, muss das Bild in einen Vektor der Größe $N * M * C$ transformiert werden (Flatten). Convolutional Neuronal Networks (CNNs) besitzen im Gegensatz zu klassischen Neuronal Networks Convolutional Layer, und können somit Inputs höherer Dimension verarbeiten. Dazu sind die Neuronen als Filter k der Größe $H \times W \times D$ (Höhe Breite,Tiefe) angeordnet. In diesem Kapitel wird von einem Filter mit quadratischer Grundfläche ausgegangen ($H = W$). Zur Berechnung des Layer-Outpus führt ein Filter k an einer Position (x, y) des Inputs I eine zweidimensionale diskrete Faltung (2.13) [Tho18] durch und berechnet somit einen Datenpunkt der Feature-Map I^* (Ergebnis eines Filters des Convolutional Layers). Der Parameter a in der Faltungsformel (2.13) steht für die Koordinate des Mittelpunktes des Filters. Ist der Filter die Grundfläche des Filters zum Beispiel 5×5, so ist der Wert von a drei [ON15].

$$I^*(x, y) = \sum_{i=1}^{H} \sum_{j=1}^{W} I(x - i + a, y - j + a)k(i, j) \tag{2.13}$$

Zur Berechnung des nächsten Datenpunktes der Feature Map I^* wird der Filter um eine Schrittweite (*stride*) auf dem Input verschoben. Sollte für die Berechnung von $I^*(x, y)$ Datenpunkte benötigt werden, welche über den Rand des Inputs hinaus liegen, müssen diese Punkte auf eine andere Weise bestimmt werden. Eine mögliche Strategie ist das vernachlässigen dieser Punkte, was je nach Größe des Filters eine Verkleinerung der Feature-Map I^* zur Folge hätte. Die zweite mögliche Strategie, welche die fehlenden Datenpunkte zur Verfügung stellt, ist das Zero-Padding , welches die fehlenden Datenpunkte mit dem Wert Null ersetzt. Außerdem besteht als dritte Möglichkeit, dass die fehlenden Datenpunkte mit dem dazugehörigen Randwert ersetzt wird [Des] .

Nach Ausführung der Faltung wird die errechnete Feature-Map durch eine ausgewählte Aktivierungsfunktion verarbeitet. Ein Convolutional-Layer besteht nicht nur aus einem Filter, sondern kann mehrere Filter enthalten. Für jeden Filter wird die vorher beschriebene Faltung separat ausgeführt und die entstandenen Feature-Maps werden konkateniert. Somit ergibt sich für den Output eines Convolutional-Layer

mit einer Stride von 1, Zero-Padding und 32 Filtern eine Dimension von $(N, M, 32)$. In einem Convolutional Neuronal Network können dann beliebige viele Layer hintereinander platziert werden. Zur Verarbeitung des letzten Convolutional Layers wird in den meisten Fällen die letzte Layer in einen Vektor transformiert, um durch ein Fully-connected Layer weiterverarbeitet werden zu können (siehe Abbildung 2.7.

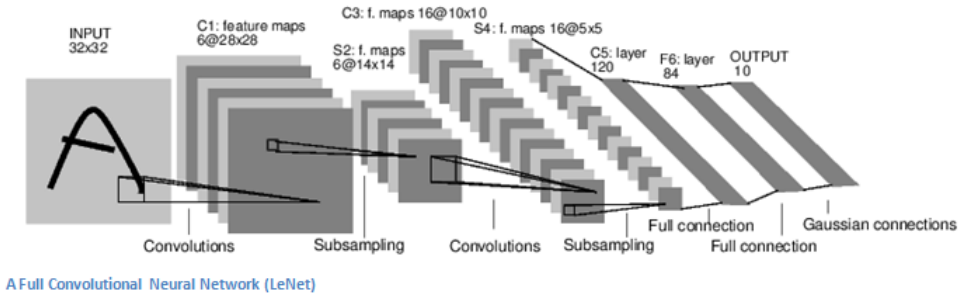

Abbildung 2.6: Beispielhafte Darstellung eines Convolutional Neuronal Networks. Bildquelle: [Des]

2.5.5 Residual Strukturen

Wie in Kapitel 2.5.4 beschrieben, können beliebig viele Convolutional Layers hintereinander platziert werden. Je mehr Layer verwendet werden, desto tiefer wird das Netz. Mit steigender Tiefe des Netzes erhöht sich das Problem kleiner werdender Gradienten (Vanishing-Gradient Problem). Das Vanishing Gradient Problem ist auf die Eigenschaft der Backpropagation zurückzuführen, dass die Gradienten Neuronen-Gewichte in Abhängigkeit zu dessen Einfluss auf den Net-Output stehen. Bei tieferen Netzstrukturen verringert sich dieser Einfluss. Um den Einfluss der Neuronen auf den Net-Output zu erhöhen, werden Skip-Connections, auch Residual Strukturen, genannt. Dazu wird der Output eines Layers auf den Input z.B. der übernächsten Schicht addiert. Es sind auch Skip-Connections mit einer höheren Schrittweite möglich [HZRS15].

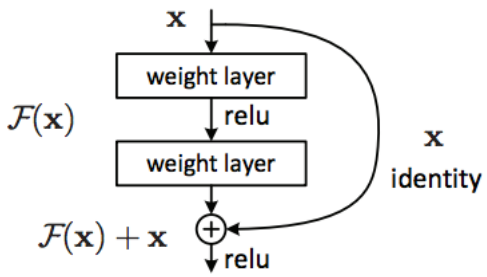

Abbildung 2.7: Realisierung einer Skip-Connection in einem Neuronal Network. Bildquelle: [HZRS15]

3 Simulation

3.1 Motivation

Um das Ergebnis des SAXS-Experiments bestmöglich zu verstehen und aufgrund einer geringen Anzahl von Experimentdaten, wurde im Rahmen der Master-Arbeit von Malte Zacharias mit dem Titel "Model-Driven Parameter Reconstruction from Small Angle X-Ray Scattering Images" eine Modellierung entwickelt. Die Modellierung beinhaltet das Design der Gitterstruktur (Elektronendichteverteilung), die Modellierung des Hochenergielasereinflusses, die Modellierung des Streuvorgangs und die Modellierung des Detektorbildes. Das ganze Kapitel bezieht sich auf die Master-Arbeit von Malte Zacharias [Zac17].

3.2 Simulationsbeschreibung

3.2.1 Design der Gitterstruktur und Simulation des Hochenergielasereinflusses

Wie in Kapitel 2 beschrieben sind die Targets des SAXS-Experiments als Gitter strukturiert, um eine analytische Beschreibung des Targets zu ermöglichen. Der erste Simulationsschritt ist die Bestimmung der Gitterstruktur. Die Breite des Gratings ist auf N Pixel beschränkt. Die Struktur des Gratings ist über die drei Startparameter Pitch, Feature-Size und Sigma definiert. Der Parameter Feature-Size legt die Breite eines Features fest und der Parameter Pitch bestimmt die Periodizität eines Features, womit beide Parameter für die Struktur des Gratings ohne Einfluss des Hochintensitätslasers verantwortlich sind. Der Parameter Sigma σ bestimmt die Aufweichungsbreite des Gratings und modelliert den Einfluss des Hochenenintensitätslasers.

Der erste Schritt der Simulation ist die Modellierung eines Features mit Hochintensitätslasereinfluss. Dazu wird eine Rechteckfunktion ($1_{[0,\text{Feature-Size}]}$), welche die Breite eines Features festlegt mit einer Gaussverteilung ($\exp\left(-x^2/2\sigma^2\right)$) gefaltet [KRM+18]. Das Ergebnis der Faltung wird mit Hilfe der Errorfunktion (Gleichung (3.1)) dargestellt.

$$\text{erf(x)} = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\xi^2} d\xi \tag{3.1}$$

Das Ergebnis der Faltung des Rechteckimpulses und der Gauss-Verteilung ist die Modellierung eines Features:

$$\tilde{\eta} = \frac{\sqrt{\pi}\sigma}{2} (\text{erf}(\frac{x}{\sqrt{2}\sigma}) - \text{erf}(\frac{x - \text{fsize}}{\sqrt{2}\sigma})) \tag{3.2}$$

Das modellierte Feature wird durch eine weitere Faltung mit mehreren um Pitch verschobenen Dirac-Impulsen [Beu11] periodisch fortgesetzt wird, um die Grating-Struktur zu komplettieren.

3.2.2 Effekt der Startparameter auf die Gitterstruktur

Die drei Startparameter Sigma (σ), Pitch und Feature-Size, welche die Struktur des Gratings beschreiben, haben unterschiedliche Effekte auf die Gratingstruktur. Sigma ist der Parameter, welcher, wie bereits beschrieben, den Einfluss des Hochenergielasers auf ein Feature modelliert. Je höher Sigma gewählt wird, desto mehr wird die Kantenstruktur des Targets aufgeweicht. In Abbildung 3.1 ist dieser Effekt dargestellt.

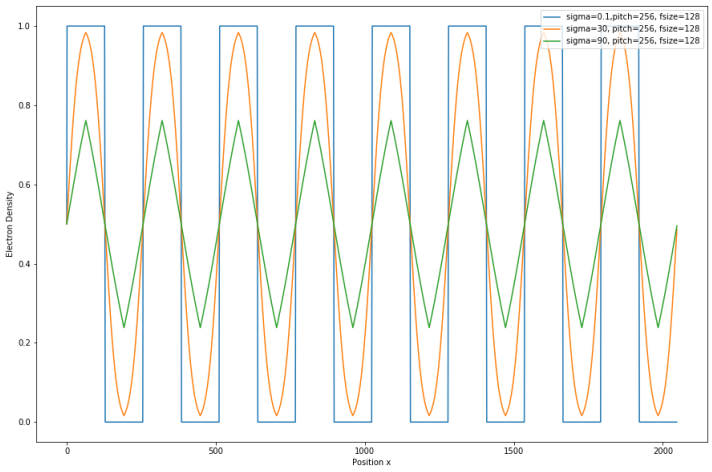

Abbildung 3.1: Vergleich der Kantenstruktur unter variierenden Sigma-Werten bei gleichbleibenden Pitch- und Feature-Size- Werten

Der Parameter Pitch bestimmt die Periodizität der Kantenstruktur. Daraus folgt, dass mit steigendem Pitch-Wert die Anzahl der Features sinkt. Dieser Effekt ist in der Abbildung 3.2 in der rechten Spalte dargestellt. Der dritte Startparamter Feature-Size bestimmt die Größe der Features. Dabei kann die

Feature-Size nicht größer als Pitch gewählt werden, da sonst keine Kantenstruktur mehr erkennbar wäre. Der Effekt der Feature-Size wird in Abbildung 3.2 in der linken Spalte dargestellt.

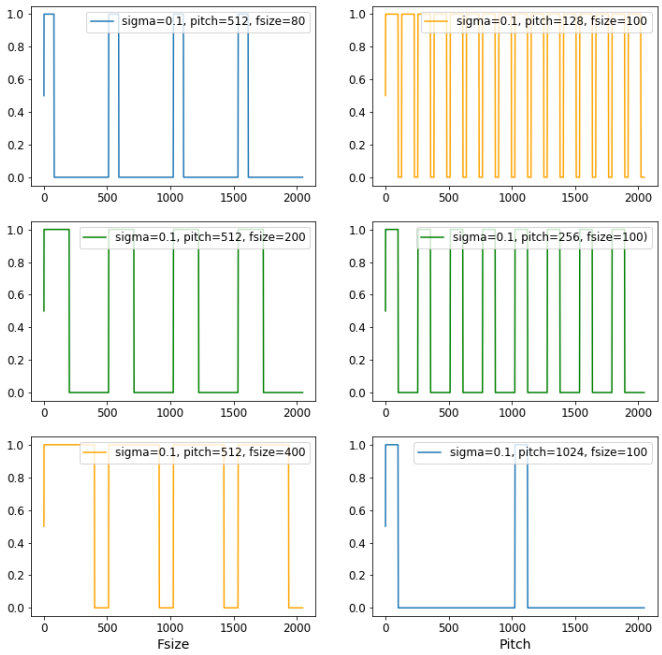

Abbildung 3.2: Vergleich der Kantenstruktur unter variierenden Pitch-Werten (rechte Spalte) und variierenden Feature-Size-Werten (linke Spalte)

3.2.3 Simulation der Streuung

Wie im Kapitel zur Experimentbeschreibung beschrieben, ist das aufgenommene Detektorbild äquivalent zum Betragsquadrat der Fouriertransformation der Elektronendichte η. Im Fall der Simulation ist die Elektronendichte eine diskrete eindimensionale Elektronendichteverteilung. Deswegen wird zur Bildung des Detektorbildes eine eindimensionale diskrekte Fouriertransformation (3.3) verwendet.

$$\hat{a}_k = \sum_{j=0}^{N-1} e^{-2\pi \mathrm{i} \cdot \frac{jk}{N}} \eta_k \text{ für } k = 0, \ldots, N-1 \tag{3.3}$$

Das entstandene Produkt der Fouriertransformation ist im Raum der Komplexen Zahlen und weist den gleichen Informationsgehalt wie die Elektronendichteverteilung auf. Das heißt die Amplituden- und Phaseninformationen sind nach wie vor vorhanden. Wie in Kapitel 1 beschrieben, ist der Detektor Zeit integrierend und die Phase geht verloren, weswegen wird die Phaseninformation im weiteren Simulations-

verlauf nicht weiter verwendet. Um schlussendlich die aufgenommenen Intensitäten des Detektors (siehe Abbildung 3.3) zu erhalten, wird das Betragsquadrat des Amplitudenspektrums gebildet ($\Phi = |\hat{a}_k|^2$)

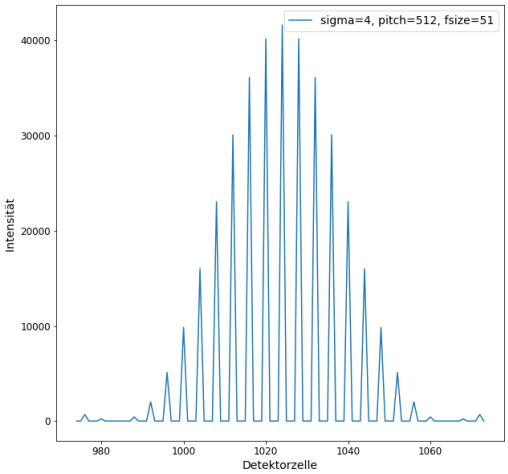

Abbildung 3.3: Resultierendes Detektorbild als Endprodukt der Simulation

3.2.4 Auswirkung der Startparameter auf das Simulationsergebnis

Die Startparameter des Simulation haben ebenfalls Auswirkung auf das Endprodukt der Simulation. Betrachtet man zunächst die Auswirkungen des Startparameters Pitch (Abbildung 3.4), kann man erkennen, dass mit steigendem Pitch sich das Intensitätsmaximum verringert wird. Zusätzlich kommen noch mehr Peaks hinzu. Das hängt damit zusammen, dass sich mit steigenden Pitch die Anzahl der Features verringert und somit mehr Frequenzen benötigt werden, um die Gitterstruktur zu modellieren.

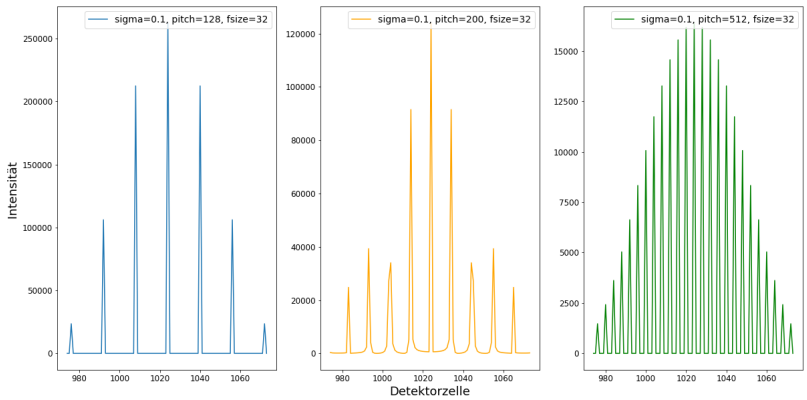

Abbildung 3.4: Verhalten der Detektoraufnahme in Abhängigkeit des Parameters Pitch

Der Startparamter Feature-Size verhält sich invers zum Parameter Pitch. Mit steigendem Feature-Size Wert kommt es zu einer Erhöhung des Intensitätsmaximums und es werden weniger Frequenzen benötigt, um die Gitterstruktur zu modellieren, somit sind auch weniger Peaks im Detektorsignal vorhanden(Abbildung 3.5).

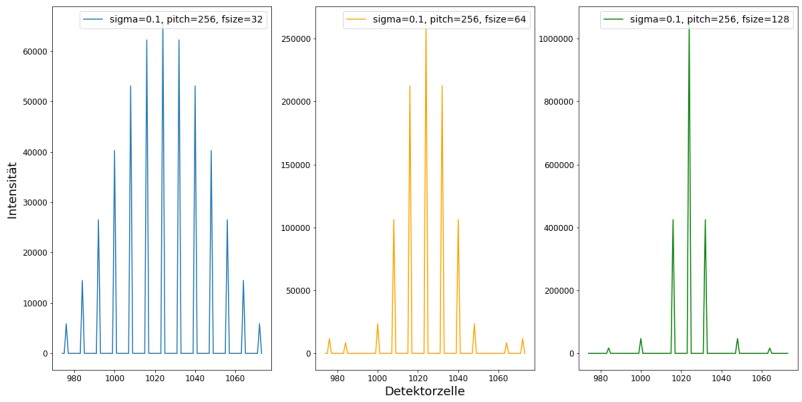

Abbildung 3.5: Verhalten der Detektoraufnahme in Abhängigkeit des Parameters Fsize

Der letzte Startparameter Sigma hat mit steigenden Wert ähnliche Auswirkungen auf die Detektoraufnahmen wie Feature-Size. Bei steigendem Sigma-Wert steigt das Intensitätsmaximum und die Anzahl der Peaks verringert sich (Abbildung Jedoch skaliert der Parameter Sigma anders als Feature-Size. Kleinere Unterschiede im Sigma-Wert können größere Auswirkungen auf die Anzahl der Peaks haben. Auch der Anstieg des Intensitätsmaximums in Abhängigkeit zu Sigma ist geringer im Vergleich zur Abhängigkeit zu Fsize.

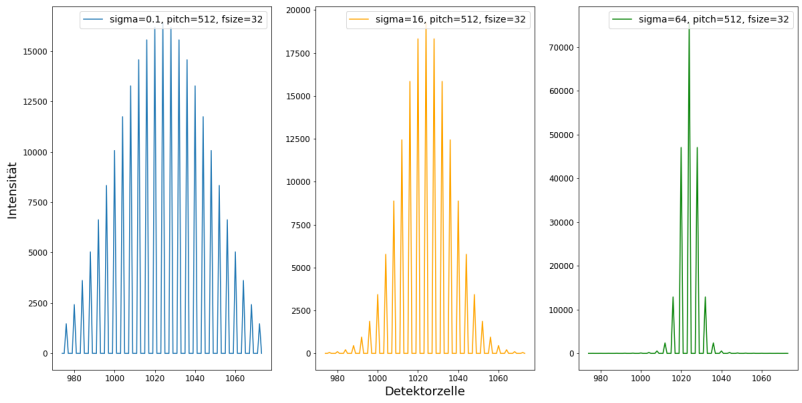

Abbildung 3.6: Verhalten der Detektoraufnahme in Abhängigkeit des Parameters Sigma

4 Datenbasis

4.1 Generator

Um einen Machine-Learning-Ansatz auszuführen, muss eine ausreichend große Datenbasis existieren. Dazu wurde im Rahmen dieser Arbeit ein Generator entwickelt, welcher mit Hilfe der vorher beschriebenen Simulation ausreichend viele Trainings-, Validierungs- und Testsdaten produzieren kann. Der erste Schritt der Genierierung der Datenbasis ist die Festlegung der Menge P von Startparametern, wobei ein Startparamter ein Tripel (Feature-Size, Pitch, Sigma) ist. Um die Berechung der Simulationsdaten, welche aus P resultieren, zu beschleunigen wurde die Simulation aus Kapitel 3 mit Hilfe des Message Parsing Interfaces (MPI) parallisiert. MPI dupliziert auszuführenden Code auf N beliebige Prozesse, welche mit Hilfe von vordefinierten Routinen miteinander kommunizieren können [GL94]. Wurde die Menge von Startparametern festgelegt, wird diese Menge über MPI-Routinen auf N Prozesse aufgeteilt. Jeder Prozess berechnet für seine zugewiesene Menge von Startparametern die Simulationsergebnisse, welche anschließend als Dateien in den Speicher des Rechenclusters gelegt werden. Zusätzlich zum Simulationsergebnis wird die Menge der Startparameter, welche als Labels für das Training des Neuronal Networks vorgesehen sind, gespeichert.

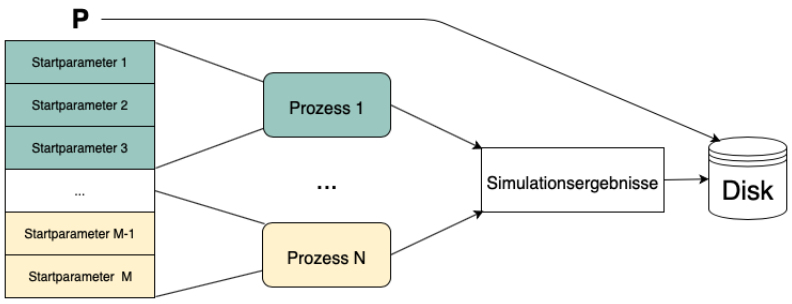

Abbildung 4.1: Schematischer Aufbau des Generators

Die Erstellung der Datenbasis wurde auf dem Rechencluster Hypnos des Helmholtz Zentrums Dresden Rossendorf auf 64 AMD 16-Kern Opteron Prozessoren ausgeführt. Die Parallelisierung der Simulation hat einen großen Speedup zur Folge, so dass 100.000 Bilder in knapp zwei Stunden anstatt in ca. 50 h Stunden(* lineare Regression aus Rechenzeiten kleinerer Datensätze) generiert werden können. Diese

kurze Rechendauer ermöglicht ein schnelleres Validieren und Korrigieren des erstellten Datensatzes. Die Rechenzeiten für kleinere Datensätze sind der Tabelle 4.1 zu sehen. Bei kleineren Datensätzen ist der Speedup nicht signifikant, da durch die Parallelisierung ein Kommunikationsoverhead entsteht.

	1 Bild	100 Bilder	1000 Bilder	100.000 Bilder
sequentiell	1,889 sek	187,613 sek	1870,095 sek	186940,286 sek *
Generator	1,917 sek	4,531 sek	42,831 sek	6900,243 sek

Tabelle 4.1: Vergleich der Berechnungsdauer für verschieden große Datensätze

4.2 Wahl der Eingangsparameter

4.3 Training-, Validation- und Testdatensatz

5 Neuronal Network

5.1 Architektur

5.2 Lossfunction

5.3 Optimizer

6 Ergebnisse und Diskussion

6.1 Lernprozess

6.2 Ergebnisse und statistische Auswertung

6.3 Fazit

Literaturverzeichnis

[Bec] BECK, Fabian: *Backpropagation*. http://www.neuronalesnetz.de/backpropagation1.html

[Beu11] BEUCHER, Ottmar: *Übungsbuch Signale und Systeme Lösungsband zum Lehrbuch Signale und Systeme - Theorie, Simulation, Anwendung*. 2011

[Cro] CROCE, Gianluca: *The Small Angle X-ray Scattering Technique: An Overview*. – http://people.unipmn.it/gcroce/download/theory.pdf

[Des] DESHPANDE, Adit: *A Beginner's Guide To Understanding Convolutional Neural Networks*. http://https://adeshpande3.github.io/A-Beginners-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2

[Fie82] FIENUP, J. R.: Phase retrieval algorithms: a comparison. In: *Appl. Opt.* 21 (1982), Aug, Nr. 15, 2758–2769. http://ao.osa.org/abstract.cfm?URI=ao-21-15-2758

[GL94] GROPP, William ; LUSK, Ewing: An Introduction to MPI - Parallel Programming with the Message Passing Interface. (1994)

[HZRS15] HE, Kaiming ; ZHANG, Xiangyu ; REN, Shaoqing ; SUN, Jian: Deep Residual Learning for Image Recognition. (2015)

[Kah97] KAHAN, William: IEEE Standard 754 for Binary Floating-Point Arithmetic. (1997)

[Kla13] KLAKOW, Dietrich: Grundagen der Signalverarbeitung. (2013)

[Kri07] KRIESEL, David: *Ein kleiner Überblick über Neuronale Netze*. 2007 http://www.dkriesel.com

[KRM+18] KLUGE, Thomas ; RÖDEL, Melanie ; METZKES, Josefine ; PELKA, Alexander ; GARCIA, Alejandro L. ; PRENCIPE, Irene ; REHWALD, Martin ; NAKATSUTSUMI, Motoaki ; MCBRIDE, Emma E. ; SCHÖNHERR, Tommy ; GARTEN, Marco ; HARTLEY, Nicholas J. ; ZACHARIAS, Malte ; ERBE, Arthur ; GEORGIEV, Yordan M. ; GALTIER, Eric ; NAM, Inhyuk ; LEE, Hae J. ; GLENZER, Siegfried ; BUSSMANN, Michael ; GUTT, Christian ; ZEIL, Karl ; RÖDEL, Christian ; HÜBNER, Uwe ; SCHRAMM, Ulrich ; COWAN, Thomas E.: *Ob-*

servation of ultrafast solid-density plasma dynamics using femtosecond X-ray pulses from a free-electron laser. 2018

[ON15] O'SHEA, Keiron ; NASH, Ryan: An Introduction to Convolutional Neural Networks. (2015)

[QR11] QUIZA R., Davim J.: Computational Methods and Optimization. Machining of Hard Materials. 190 5 (2011), S. 1396–405

[Sch] SCHLESINGER, Dmitrij: *Bildverarbeitung: Fourier-Transformation*

[Tho18] THORMÄHLEN, Thorsten: Multimediale Signalverarbeitung, Bildverarbeitung: Filter. (2018)

[Wik18] WIKIPEDIA: *Backpropagation — Wikipedia, The Free Encyclopedia*. `http://de.wikipedia.org/w/index.php?title=Backpropagation&oldid=181456626`, 2018. – [Online; accessed 04-October-2018]

[Wis] WISSENSCHAFT, Spektrum der: *Plasma*

[Zac17] ZACHARIAS, Malte: *MODEL-DRIVEN PARAMETER RECONSTRUCTIONS FROM SMALL ANGLE X-RAY SCATTERING IMAGES*. `http://dx.doi.org/10.5281/zenodo.1208410`. Version: November 2017

Danksagung

Die Danksagung...

Erklärungen zum Urheberrecht

Hier soll jeder Autor die von ihm eingeholten Zustimmungen der Copyright-Besitzer angeben bzw. die in Web Press Rooms angegebenen generellen Konditionen seiner Text- und Bildübernahmen zitieren.