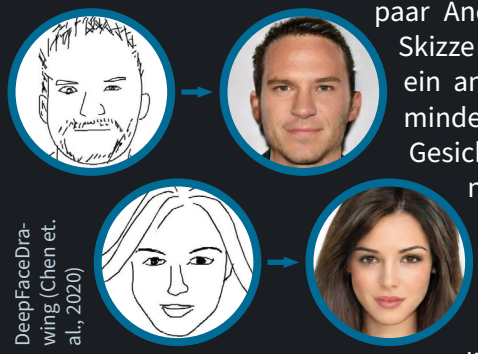
Bilder generieren mit KI

Versteckte Räume

Eine schnelle ungenaue Skizze erstellt und schon existiert ein Bild, welches einem realen Porträt gleicht. Ein paar Änderungen in der Skizze und man erhält ein angepasstes, nicht minder realistisches Gesicht. Die Nutzung neuronaler Netzwerke zum Erschaffen neuer Bilder ermöglicht Ergebnisse, die per Hand nur schwer zu erreichen sind. In diesem Whitepaper werden hierfür grundlegende Techniken und Netzwerke näher betrachtet.



DeepFaceDrawing (Chen et. al., 2020)

Convolutional Neural Networks

Beschäftigt man sich mit der Generierung von Bildern durch neuronale Netzwerke, fällt sehr schnell der Begriff des Convolutional Neural Networks, kurz CNN. Die ersten Arbeiten zu dieser Art von Netzwerk stammen von dem französischen Forscher Yann LeCun. In seiner Veröffentlichung von 1989 wird die Erkennung von handgeschriebenen Zahlen erforscht. Mit einem Datensatz, wie auf der linken Seite in Abbildung 1 zu sehen, wird das Netzwerk trainiert und besitzt danach die Fähigkeit, frei geschriebene Zahlenfolgen, wie rechts daneben, zu erkennen. In der Veröffentlichung (LeCun et. al., 1989) wird das Netzwerk zwar noch nicht so genannt, funktioniert aber ähnlich einem CNN.

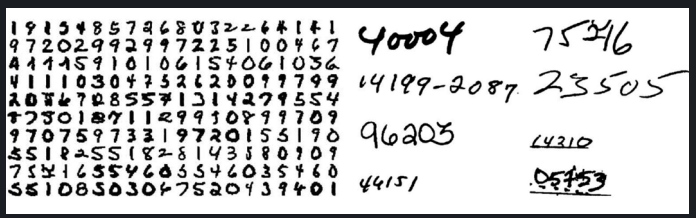


Abbildung 1. Trainingsdatensatz und Nummernfolgen zum Testen für ein CNN (LeCun et al., 1989)

Die dem Netzwerk zugrunde liegende Operation ist die namensgebende Convolution, im Deutschen *Faltung* genannt. Die Berechnung wird im Folgenden in einer einfachen Form, anhand der Matrizen in Abbildung 2, erläutert. Der Input besteht aus einer Matrix M, welche beispielsweise Werte für Pixel in einem Bild repräsentiert. Für die Faltung wird eine Faltungsmatix F, auch *Kernel* oder *Filter* genannt, benötigt. Nun wird F auf einen gleich großen Teil von M angewandt. Hierbei wird jeder Wert aus M mit dem dazugehörigen aus F multipliziert und anschließend alle aufaddiert, was einen Wert für die Ausgangsmatrix E ergibt. Ist die Berechnung für jeden Wert durchgeführt, stellt E die gefaltete Matrix M dar. In welchem Raster F über M bewegt wird, kann je nach Implementierung variiert werden.
Das Ergebnis hängt von dem verwendeten Filter ab. Beispielsweise gibt es Filter für vertikale Linien, nach deren Anwendung nur diese im Bild übrig bleiben. Somit können einzelne Strukturen aus einem Bild extrahiert werden. Ein einzelnes Convolutional Layer in einem CNN entsteht aus der Anwendung vieler verschiedener Filter, die jeweils ein anderes Feature extrahieren. Je tiefer das Layer im Netzwerk liegt, desto komplexer und spezifischer werden die Filter.

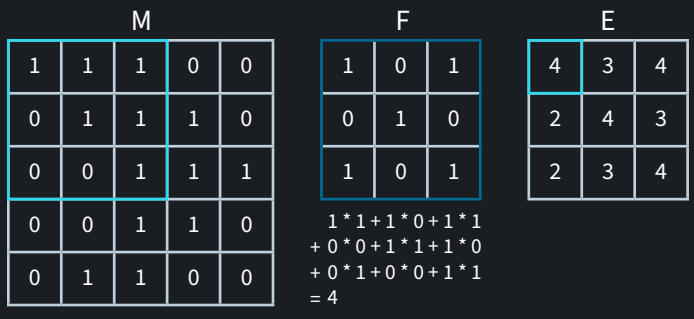


Abbildung 2. Faltung der Matrix M mit dem Filter F

In Abbildung 3 ist die Struktur des von LeCun beschriebenen Convolutional Neural Network zu sehen. Aus einem Bild werden pro Layer, durch verschiedene Filter, mehrere Feature Maps erstellt. Das dazwischen stattfindende *subsampling* wird auch *pooling* genannt. Hier wird das Ergebnis der Faltung weiter reduziert, beispielsweise, indem aus vier Einträgen nur der Maximalwert genommen wird. Dadurch verstärkt sich der Filtereffekt und die zu bearbeitenden Daten werden drastisch reduziert, was

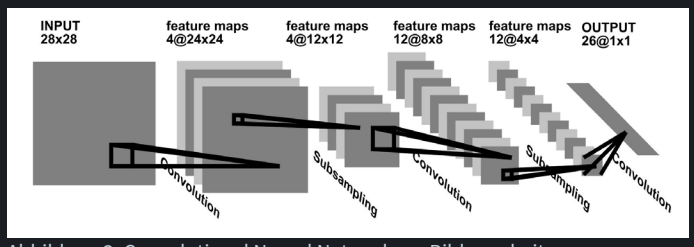


Abbildung 3. Convolutional Neural Network zur Bildverarbeitung (LeCun et al., 1995)

folglich weniger Arbeitsspeicher benötigt und tiefere Netze ermöglicht. Am Ende ist aus dem Bild eine eindimensionale Kategorisierung geworden, die wie in dem vorherigen Beispiel, aussagt, welche Zahl zu erkennen ist.
Die Ergebnisse eines CNN waren 1998 bereits so gut, dass LeCun meinte: “Convolutional NN’s have been shown to eliminate the need for hand-crafted feature extractors.” (LeCun et al., 1998)

Variational Autoencoders

Die Möglichkeit, den Inhalt eines Bildes zu extrahieren, zu kategorisieren und eine beschreibende Darstellung zu schaffen ist wichtig um neue Bilder generieren zu können. Bei einem Autoencoder wird das Bild nicht nur auseinandergenommen, sondern auch versucht es wieder zu rekonstruieren. In Abbildung 4 ist die grundlegende Struktur des Netzwerkes gegeben. Hierbei wird das Bild durch einen Encoder, beispielsweise unter Einsatz von CNNs, in eine Repräsentation geringerer Dimension umgewandelt. Da das Bild nun durch weniger zur Verfügung stehende Informationen beschrieben ist, wird die Repräsentation auch *Bottleneck* genannt. Mit diesen Daten versucht der Decoder das ursprüngliche Bild zu rekonstruieren. Um das Netzwerk zu trainieren, werden Original und Rekonstruktion verglichen und in Richtung einer möglichst geringen Differenz optimiert. Ein Autoencoder kann beispielsweise als Denoiser eingesetzt werden, indem auf das Original eine Noise gelegt und zusammen encodiert wird. Das daraus rekonstruierte Bild wird gegen das Original ohne Noise abgeglichen, sodass der Decoder darauf trainiert wird, die Noise aus dem Input zu entfernen. Doch eine Rekonstruktion ist noch kein neu generiertes Bild. Hierfür lohnt es sich, die mit dem Encoder erstellte Repräsentation näher zu betrachten, welche die Form eines Vektors haben können. Die Bedeutung der verschiedenen Einträge können ausschließlich von dem trainierten Netzwerk interpretiert werden, weshalb man davon spricht, dass die Repräsentationen einen „verstecken“ Raum – den *Latent Space* bilden. Nun kann ein Autoencoder zum Beispiel mit geometrischen Formen trainiert werden, sodass Vektoren für einen Kreis, ein Quadrat und ein Dreieck existieren. Alle Vektoren unterscheiden sich in den Einträgen und können jeweils durch den Decoder zum Ursprungsbild rekonstruiert werden. Es stellt sich die Frage wie die einzelnen Einträge verändert werden müssen, um eine sinnvolle neue Form daraus zu rekonstruieren. Hierfür muss eine nachvollziehbare Ordnung in den Latent Space gebracht werden, wie es bei einem *Variational Autoencoder (VAE)* geschieht. Das Besondere ist, dass der Encoder nicht einen einzelnen Punkt zurückgibt, sondern eine durch Erwartungswert und Standardabweichung beschriebene Normalverteilung. Im optimalen Fall ist das die Standard-Normalverteilung mit einem Erwartungswert von 0 und einer Standardabweichung von 1. Mittels der sogenannten Kullback-Leibler-Divergenz kann die Abweichung überprüft werden. Es gilt: Je strikter die Einschränkung auf eine bestimmte Verteilung ist, desto schwieriger wird es, das Bild zu rekonstruieren. Das heißt, dass das Netzwerk auf einen möglichst geringen Rekonstruktionsfehler, aber auch auf eine Repräsentation möglichst nahe der Standard-Normalverteilung im Latent-Space trainiert werden muss. Durch die Ordnung, auch *Regularisierung* genannt, ist es dann möglich, aus dem Latent Space einen einzelnen Punkt zu sampeln und daraus eine sinnvolle Rekonstruktion zu gewinnen. In Abbildung 5 ist der Unterschied vor und nach der Regularisierung verdeutlicht. Es ist zu erkennen, dass naheliegende Punkte einen ähnlichen Output und Punkte innerhalb der Verteilung einen sinnvollen Output liefern. Somit kann beispielsweise ein Punkt gesampelt werden, der eine Mischung zwischen Quadrat und Kreis als Output liefert, was einem neu generierten Bild entspricht.

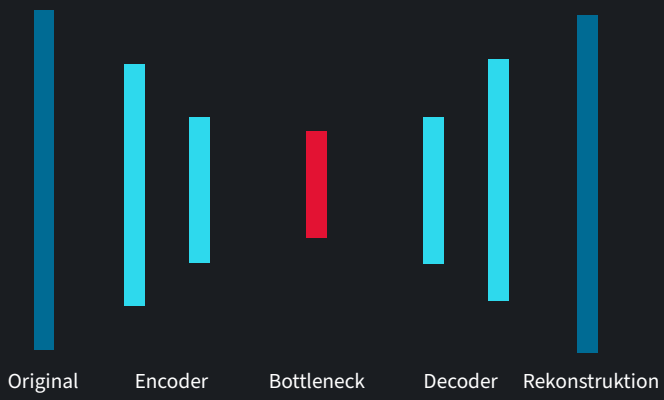


Abbildung 4. Aufbau eines Autoencoders

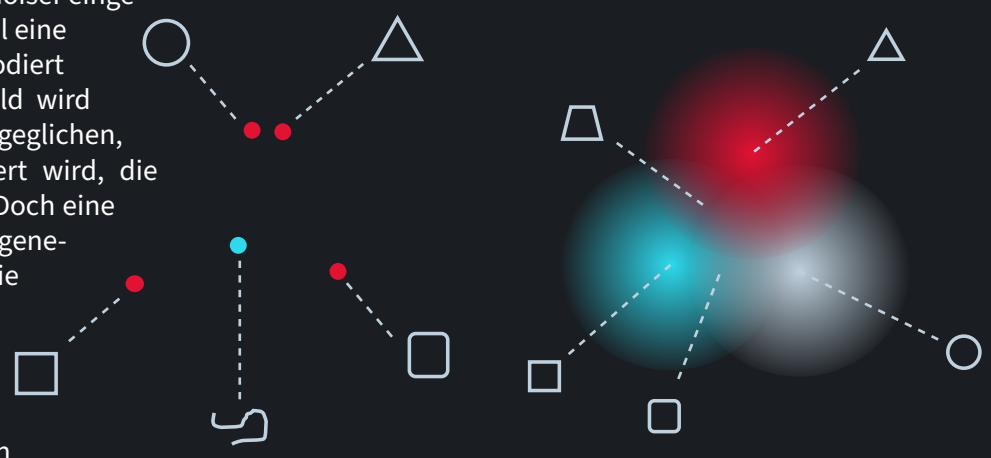


Abbildung 5. Schematische Darstellung des Latent-Space vor und nach der Regularisierung nach (Rocca, 2019)

Generative Adversarial Networks

Ein weiteres Netzwerk mit der Möglichkeit neue Bilder zu generieren, ist das von Ian Goodfellow 2014 entwickelte *Generative Adversarial Network (GAN)* (Goodfellow et. al., 2014). Der Grundgedanke ist, dass zwei Netzwerke sich als Gegner gegenseitig optimieren.

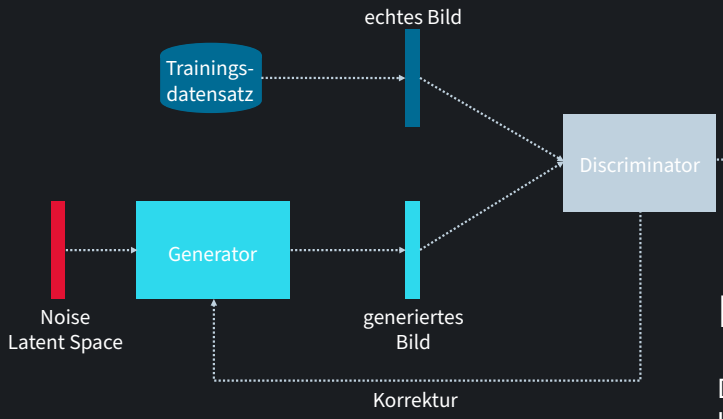
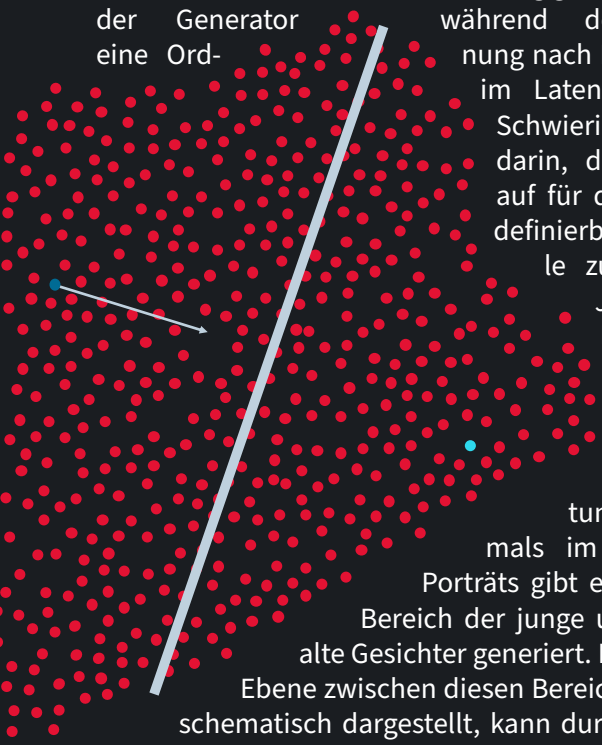


Abbildung 6. Aufbau eines Generative Adversarial Network nach (Goodfellow et.al., 2014)

Abbildung 6 zeigt den Aufbau des GAN. Im Trainingsprozess generiert der Generator, bestehend aus einem neuronalen Netzwerk, mit einer zufälligen Noise Bilder. Ein zweites Netzwerk, der Discriminator versucht zu bestimmen, ob das Bild ein generiertes oder ein echtes Bild ist. Der Input besteht alternierend aus einem Bild der Trainingsdatenbank, welche beispielsweise Portraits enthält, und dem generierten Bild des Generators. Die Korrektur besteht in der Optimierung des Generators einen Output zu erzeugen, der dazu führt, dass der Fehler des Discriminators größer wird. Das optimale Ziel ist es, das Netzwerk so zu trainieren, dass der Discriminator nicht mehr bestimmen kann, ob das Bild echt oder generiert ist. Gleichzeitig kann der Generator dafür eingesetzt werden, aus einer Noise ein neues Bild zu generieren. Ist die Noise aus einer Normalverteilung gesampelt, schafft der Generator während des Trainings eine Ordnung nach Bildstrukturen im Latent Space. Die Schwierigkeit besteht darin, diese Ordnung auf für den Menschen definierbare Merkmale zu übertragen. Jede Translation in eine Richtung im Latent Space, verändert die Gewichtung eines Merkmals im Output. Für Porträts gibt es somit einen Bereich der junge und einen der alte Gesichter generiert. Kennt man die Ebene zwischen diesen Bereichen, wie links schematisch dargestellt, kann durch Anpassung der gesampelten Werte in Richtung der Normalen das sonst gleichbleibende Gesicht im Alter verändert werden.



Ein GAN erreicht eine hohe Qualität an generierten Bildern. Im Vergleich zu einem VAE lässt sich der Output aber schwieriger hinsichtlich konkreter Merkmale ändern.

Lust auf mehr

Die kurzen Intervalle und die Aktualität der Veröffentlichungen zum Thema Bildgenerierung mit neuronalen Netzwerken, zeigen eine rasante Entwicklung und Verbesserung der Techniken und Ergebnisse. Die heutigen Möglichkeiten sind bereits verblüffend und es bleibt spannend abzuwarten, was in den nächsten Jahren zur Anwendung kommen wird.

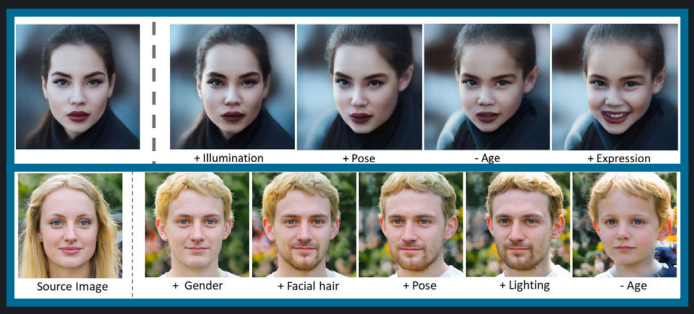


Abbildung 7. StyleFlow (Abdal, Zhu et. al., 2020)

Hinsichtlich der zuvor beschriebenen Veränderung von ausschließlich einzelnen Merkmalen zeigen Abdal et. al. 2020 mit StyleFlow hervorragende Ergebnisse. Die Arbeit baut auf dem von Nvidia entwickelten StyleGAN (Karras et. al., 2018) auf, welches sehr realitätsnahe täuschend echte Bilder generieren kann.
Das von OpenAI entwickelte Netzwerk DALL•E setzt neue Maßstäbe in der Generierung von Bildern aus Textbeschreibungen (Ramesh et. al., 2021). Es basiert auf dem vielseitigen Textverarbeitungsmodell GPT-3. Die generativen Möglichkeiten beinhalten auch die wildesten Kombinationen und zahlreiche Darstellungsstile, wie in Abbildung 8 zu sehen.

Abbildung 8. DALL-E (https://openai.com/blog/dall-e/)

Eine Plattform zur kreativen Nutzung von KI bietet Artbreeder. Basierend auf GAN, auch StyleGAN, können vor allem im Stil von Konzept-Art Porträts, Charaktere, sowie Landschaften generiert werden. Die Anpassung isolierter Merkmale funktioniert nicht immer, aber die Ergebnisse können sich sehen lassen. Die in Abbildung 9 zu sehende Iteration von dem linken generierten Bild zum rechten funktioniert in Sekunden. Hier wurden Veränderungen in den Merkmalen „Sunlight“, „Snow“ sowie „Vegetation“ vorgenommen.

Das Potenzial der Generierung von Bildern mit KI ist enorm und es lässt sich wohl die Prognose wagen, dass in der bildschaffenden Medienbranche dadurch viele Prozesse vereinfacht oder auch übernommen werden.

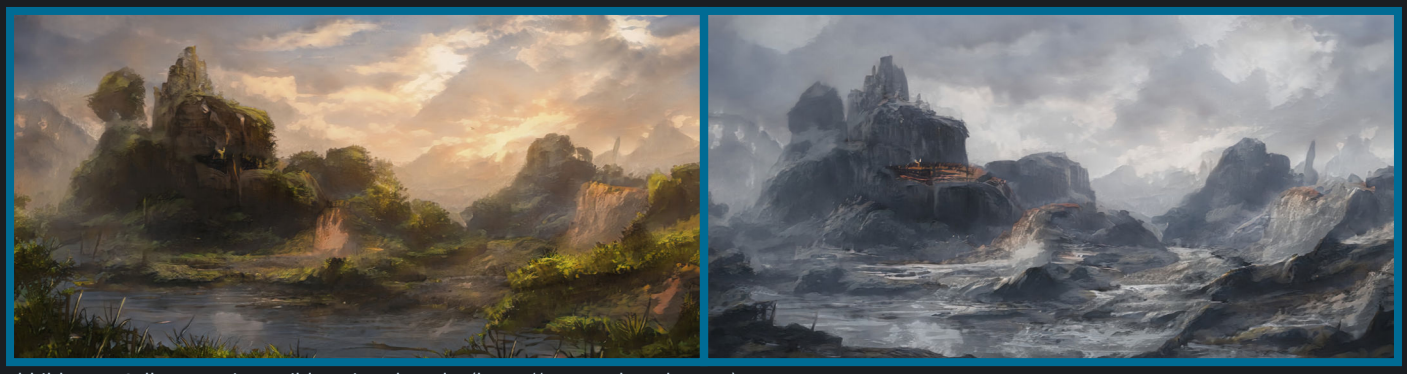


Abbildung 9. Selbst generierte Bilder mit Artbreeder (https://www.artbreeder.com)

Weiterführende Links zu Erklärungen und Videos

CNN
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2
https://www.youtube.com/watch?v=py5byOOHZM8

VAE
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
http://kvfrans.com/variational-autoencoders-explained/
https://www.youtube.com/watch?v=9zKuYvjFFS8&t=634s

GAN
https://www.youtube.com/watch?v=Sw9r8CL98N0
https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29
https://www.youtube.com/watch?v=dCKbRCUyop8&t=2s

Johannes Nitschke, HdM,
Aktuelle Themen KI (WS 2020/21)

Quellen

Abdal, R., Zhu, P., Mitra, N. & Wonka, P. (2020). *StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows.* http://arxiv.org/pdf/2008.02401v2

Chen, S.-Y., Su, W., Gao, L., Xia, S. & Fu, H. (2020). *Deep Generation of Face Images from Sketches.* http://arxiv.org/pdf/2006.01047v2

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y [Yoshua]. (2014). *Generative Adversarial Networks.* http://arxiv.org/pdf/1406.2661v1

Karras, T., Laine, S. & Aila, T. (2018). *A Style-Based Generator Architecture for Generative Adversarial Networks.* http://arxiv.org/pdf/1812.04948v3

Lecun, Y. & Bengio, Y [Y.]. (1995). *Convolutional networks for images, speech, and time-series.* In M. A. Arbib (Hg.), The handbook of brain theory and neural networks. MIT Press.

Lecun, Y., Bottou, L., Bengio, Y [Y.] & Haffner, P. (1998). *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

Lecun, Y., Jackel, L. D., Boser, B., Denker, J. S., Henderson, D., Howard, R. E. & Hubbard, W. (1989). *Handwritten Digit Recognition: Applications of Neural Net Chips and Automatic Learning.* In NATO Neurocomputing.

O‘Shea, K. & Nash, R. (2015). *An Introduction to Convolutional Neural Networks.* http://arxiv.org/pdf/1511.08458v2

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M. & Sutskever, I. (2021). *Zero-Shot Text-to-Image Generation.* http://arxiv.org/pdf/2102.12092v2

Rocca, J. (2019). *Understanding Variational Autoencoders (VAEs).* https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Zeiler, M. D. & Fergus, R. (2014). *Visualizing and Understanding Convolutional Networks.* In D. Fleet, T. Pajdla, B. Schiele & T. Tuytelaars (Hg.), Computer Vision - ECCV 2014 (S. 818–833). Springer International Publishing.