

# Neuronale Netze in der Bildverarbeitung

## Versuch-2

Julian Schmidt MtkNr : 4705623, Vincenz Forkel MtkNr : 4097900

22. Dezember 2021

# 1 Einführung

Ziel des zweiten Versuchs des Praktikums Neuronale Netze in der Bildverarbeitung im Wintersemester 2021/22 ist es Grundlagen der Bildrekonstruktion mittels Neuronaler Netze kennen zu lernen.

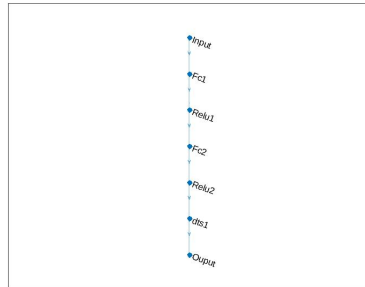
Konkret werden hierbei Ein - und Ausgänge von Multimodefasern betrachtet. Der Praktikumsversuch vergleicht ein Multi-Layer-Perceptron mit einem unet.

Hierfür wird ein Datensatz mit Bildern bestehend aus 16 mal 16 Pixeln zur Verfügung gestellt.

Im Verlauf des Versuches werden Probleme wie Overfitting oder zu kleine Datensätze behandelt.

## 2 Datensatz

Zunächst wird ein Multi-Layer-Perceptron der folgenden Architektur:



erstellt und mit dem gegebenem Datensatz trainiert. Nach dem Training zeigt sich, dass die Genauigkeit des Netzes trotz Hyperparameter tuning nicht sehr hoch ist.

Um eine höhere Genauigkeit zu erzielen wird der Datensatz erweitert.

Hier werden die gegebenen Daten augmentiert.

Es kann Beispielsweise ein Bild leicht rotiert werden und dadurch steht ein neues Bild zur Verfügung. Mit diesem Prinzip kann der Datensatz schnell um ein vielfaches erweitert werden.

Um zu untersuchen ob der erweiterte Datensatz für bessere Ergebnisse sorgt, wirds das Nets erneut trainiert, allerdings dieses mal mit dem erweitertem Datensatz.

Erstellt wird der Augmentierte Datensatz in dem Script `create_augmented_data.m`

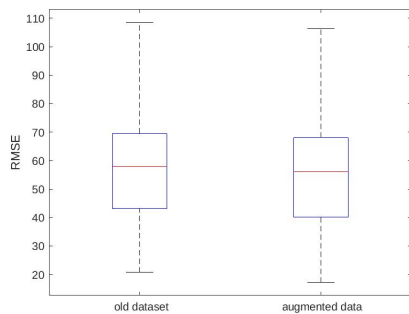
## 2.1 Ergebnisse beider Datensätze

Im folgendem sind verschiedene Gütekriterien in Boxplots dargestellt, um die Genauigkeit beider Netze zu bewerten.

Hierfür wird jeweils der Testdatensatz der ursprünglichen Datensatzes verwendet.

Es wird kein augmentierter Testdatensatz verwendet, damit für den Vergleich die gleichen Daten verwendet werden.

Zunächst wird der RMSE zwischen Testdatensatz und rekonstruierten Bildern beider Netze verglichen.

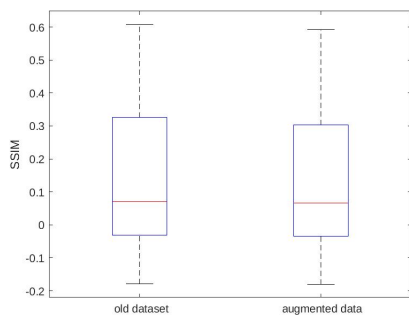


Hier zeigt sich, dass sich der RMSE beider Netze fast gleich verhält.

Der Interquartilsabstand des Netzes, welches mit augmentierten Daten trainiert wurde ist etwas größer.

Der Median ist bei beiden Netzen annähernd gleich.

Da die RMSEs sich sehr ähnlich sehen untersuchen wir ebenfalls die Strukturellen Ähnlichkeiten (SSIMs) der rekonstruierten Bilder



Hier zeigt sich ebenfalls das sich die rekonstruierten Bilder, beider Netze ähneln und so auch die struk. Ähnlichkeit. Es werden ebenfalls boxplots für PSNR und Korrelationskoeffizient erstellt, diese zeigen aber ähnliches Verhalten, wie beim RMSE und SSIM.

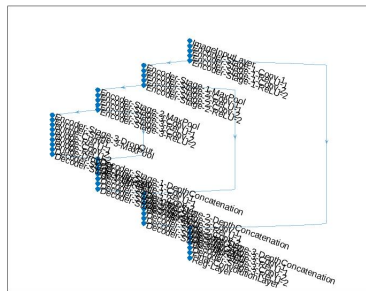
### 3 unet

Da durch den augmentierten Datensatz noch keine deutliche Verbesserung erkennbar war, wird ebenfalls ein convolutional Netz trainiert.

Es wird das UNET verwendet.

Hier wird der finale convolutional Layer angepaßt.

Die Architektur des Netzes sieht wie folgt aus:

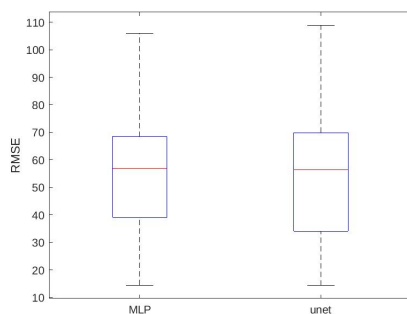


#### 3.1 Auswertung unet

Nachdem das Unet mit dem augmentiertem Datensatz trainiert wurde wird es mit dem MLP, welches ebenfalls mit dem augmentiertem Datensatz trainiert wurde verglichen.

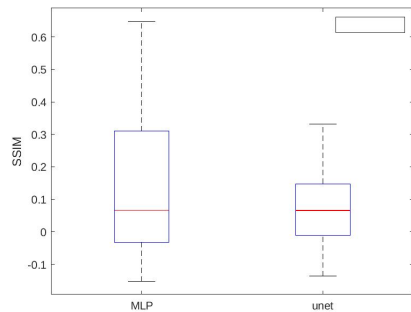
Hierzu wurden erneut die Güte Kriterien RMSE, SSIM, PSNR und Korrelationskoeffizient verwendet.

Es ist zuvor noch zu erwähnen, dass das unet eine deutlich höhere Zeit für das Training beansprucht, als das MLP. Zunächst wird wieder der RMSE inspiziert.



Hier zeigt sich erneut ein annähernd gleiches Verhalten, der einzige Unterschied ist, dass die RMSE Streuung beim unet etwas stärker ist.

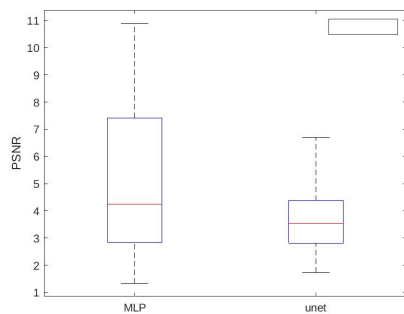
Um einen besseren Vergleich zu bekommen wird das SSIM ebenfalls analysiert.



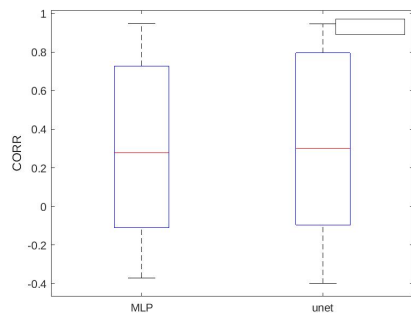
Anders als beim Vergleich beider MLPs ist dieses mal ein deutlicher Unterschied zu erkennen. Ins Auge sticht sofort, dass der SSIM Maximalwert deutlich geringer beim unet, als beim MLP, ist.

Dazu ist der Interquartilsabstand ebenfalls deutlich geringer, der Median ist etwas, aber nicht deutlich, höher als beim MLP.

Ein annähernd gleiches Verhalten ist beim PSNR zu erkennen.



Der Korrelationskoeffizient hingegen verhält sich bei beiden Netzen fast gleich.



## 4 Trainingsverhalten

Beim Hyperparameter-tuning der MLPS, hat sich gezeigt, dass der validation loss nach 10 Epochen leicht zu steigen beginnt, deshalb werden alle Netze mit 10 Epochen trainiert.

Es ist ebenfalls möglich über mehr Epochen zu trainieren und die validation patience entsprechend anzupassen.

Wir konnten hiermit keinen deutlichen Performance Unterschied erkennen hätten uns aber etwas Zeit erspart, wenn wir die validation patience von Anfang an verwendet hätten, anstatt die ideale Epochen zahl von Hand zu bestimmen.

Die validation patience ist ein automatisierter Prozess und somit weniger Fehler anfällig, als manuelles Einstellen.

Abschließend ist zu sagen, dass das unet eine etwas höhere Genauigkeit erzielt, allerdings eine deutlich höhere Trainingszeit beansprucht und auch deutlich größere Rechenleistung benötigt.

Mit einer leistungsfähigen GPU ist der Unterschied vermutlich deutlich geringer.