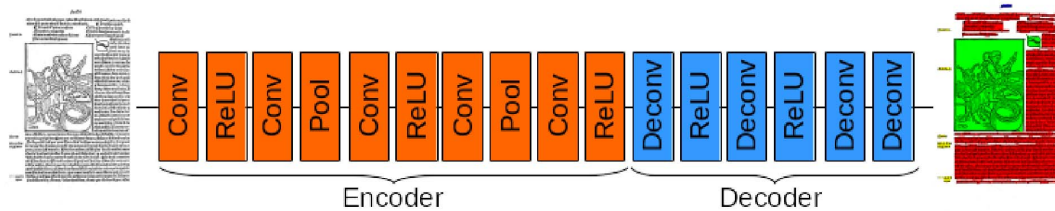


# Übungsblatt: 7

Bearbeitung am 18. Juni

## Aufgabe 1: FCN

In einem Paper ist folgende FCN-Struktur angegeben:

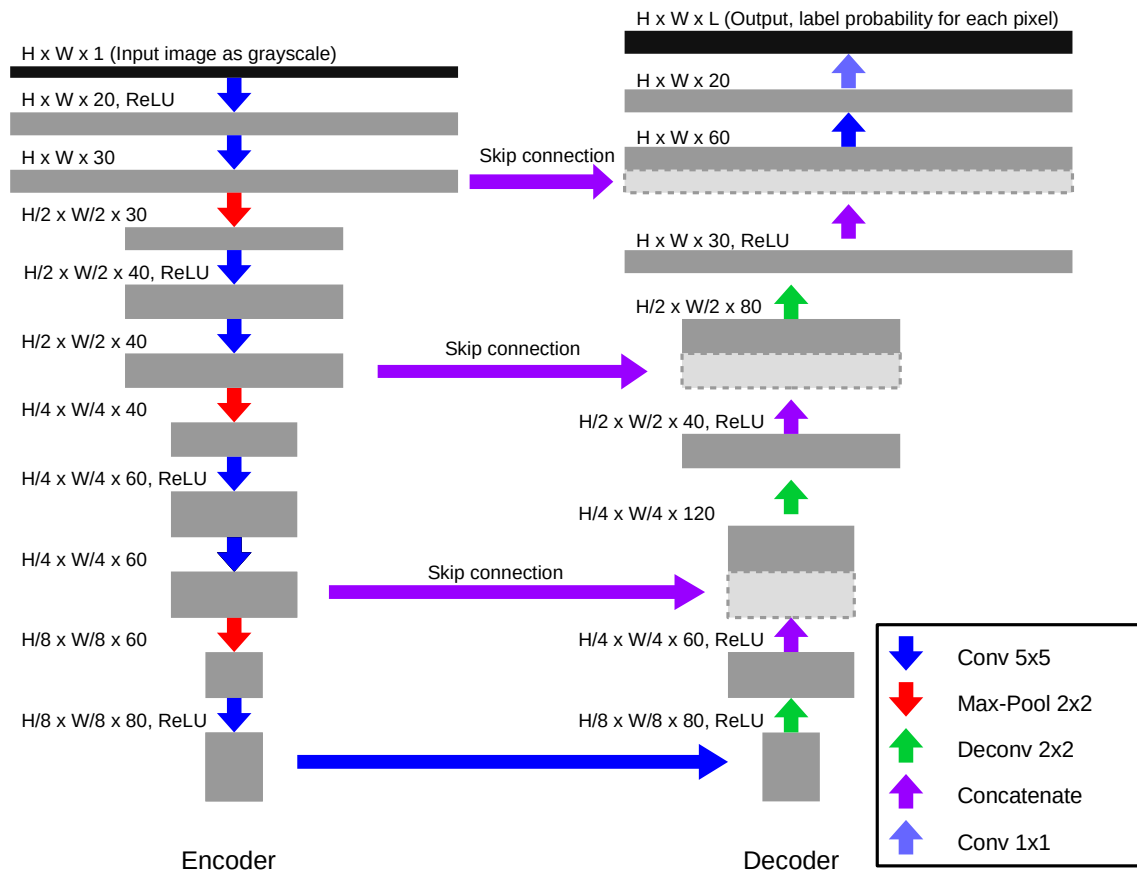


Alle CNN-Layer haben hierbei einen Stride von 1, eine Kernel-Size von 3x3 und nutzen *same* Padding, d. h. ein Padding von 1. Die Filteranzahl im Encoder ist 20, 40, 60, 80, 100, und im Decoder 80, 60, 40, 5. Alle Pooling-Layer sind normale 2x2-Max-Pooling-Layer. Das erste Deconv-Layer ist äquivalent zu einem normalen Conv-Layer, da es einen Stride 1x1 mit „Same“ Padding verwendet. Die nächsten zwei Deconv-Layer werden zum Upsampling verwendet (Stride 2x2), um das Bild wieder auf Originalgröße zu bringen. Die ersten drei Deconv-Layer haben eine Kernel-Size von 3, das letzte Layer nur von 1x1. Die finale Ausgabe wird in ein Softmax geschickt um die Wahrscheinlichkeitsverteilung zu bestimmen. Dieses FCN soll genutzt werden, um eine Seite großflächig in Hintergrund, Text, Bild, Überschriften und Marginalien zu segmentieren.

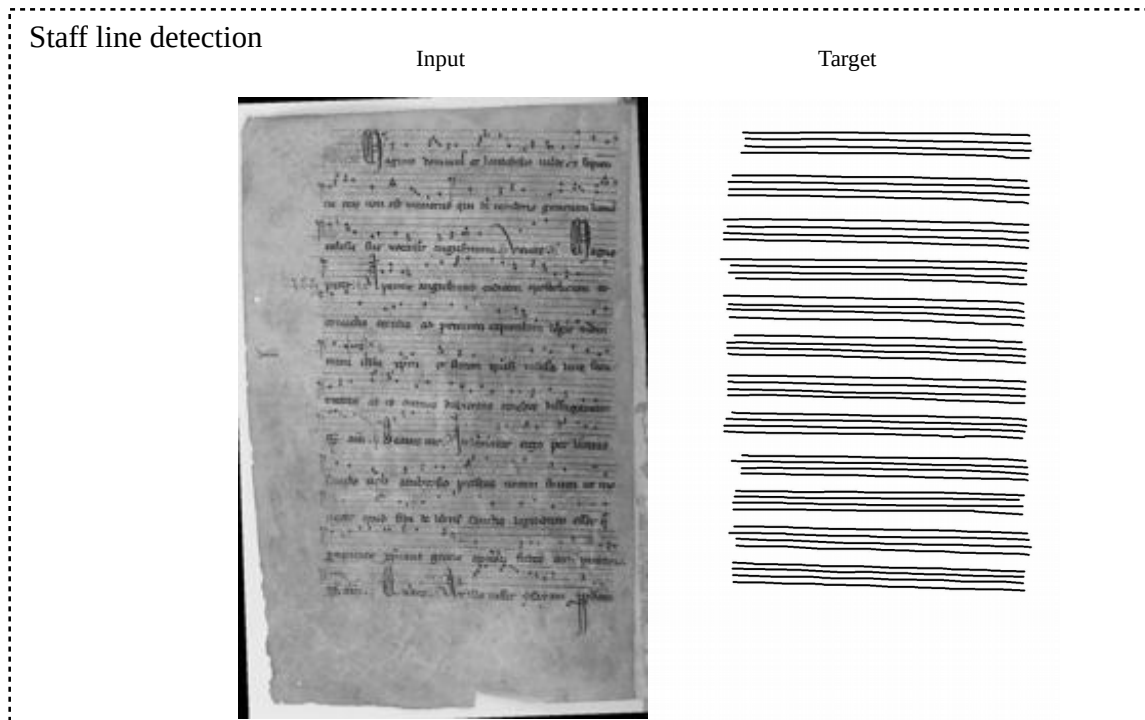
- Warum hat die letzte FCN-Schicht 5 Kanäle?
- Hat das Netzwerk-Skip-Connections? Ist das so sinnvoll?
- Geben Sie den Pseudo-Code für das Netzwerk an.
- Geben Sie für jede Schicht des Netzwerks die Eingabe- und Ausgabedimension sowohl die Anzahl der trainierbaren Gewichte an.
- Implementieren Sie die Netzstruktur und trainieren Sie das Netzwerk auf den vorgegebenen Daten.

## Aufgabe 2: FCN

In einem anderen Paper ist folgende FCN-Struktur angegeben:



Die finale Ausgabe (Logits) wird in ein Softmax geschickt um die Wahrscheinlichkeitsverteilung zu bestimmen. Dieses FCN soll genutzt werden, um auf einer Seite Notenlinien zu erkennen.



- Wie viele Kanäle hat die letzte Schicht?
- Hat das Netzwerk-Skip-Connections? Ist das so sinnvoll?
- Geben Sie den Pseudo-Code für das Netzwerk an.
- Geben Sie für jede Schicht des Netzwerks die Eingabe- und Ausgabedimension sowohl die Anzahl der trainierbaren Gewichte an.
- Implementieren Sie die Netzstruktur und trainieren Sie das Netzwerk auf den vorgegebenen Daten.