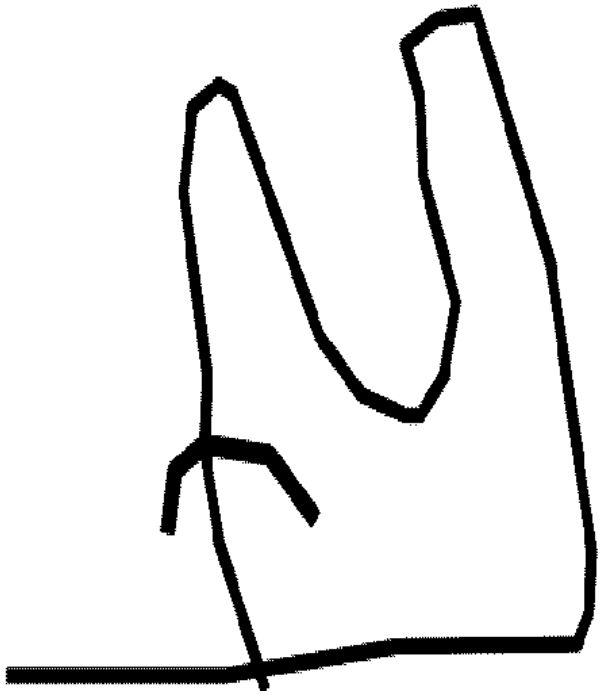
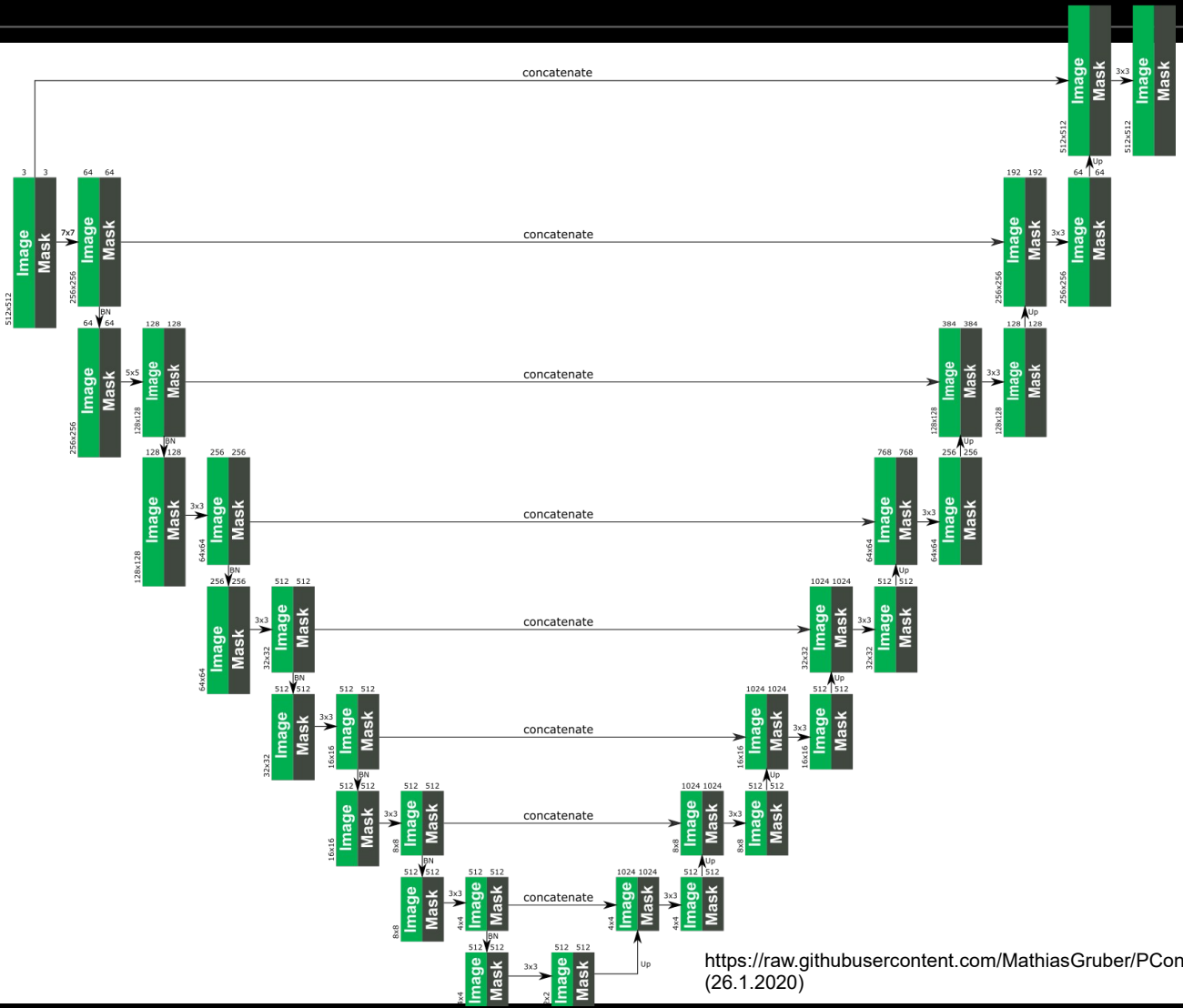
DeFINe – Deep Facial Inpainting Network

- Kevin Gellhaus
- Marcel Früh
- Micha Schilling



Model

- U-Net Architektur (Convolutional Networks for Biomedical Image Segmentation (Ronnenberger et al.)



https://raw.githubusercontent.com/MathiasGruber/PConv-Keras/master/data/images/architecture.png (26.1.2020)

Zielstellung:
Einfluss von verschiedenen Lossfunktionen

Lossfunktionen - V0

- Einfacher L1 Loss
- $L = \frac{1}{N}||O - GT||_1$

Lossfunktionen - V1

- $L_{hole} = \frac{1}{N}||(1 - M) \odot (O - GT)||_1$
- $L_{valid} = \frac{1}{N}||(M) \odot (O - GT)||_1$

Lossfunktionen - V2

- $L_{hole} = \frac{1}{N}||(1 - M) \odot (O - GT)||_1$
- $L_{valid} = \frac{1}{N}||(M) \odot (O - GT)||_1$
- $L_{perceptual} = \sum_{p=0}^{P-1} \frac{||\Psi_p^O - \Psi_p^{GT}||1}{N} + \sum_{p=0}^{P-1} \frac{||\Psi_p^C - \Psi_p^{GT}||1}{N}$

Lossfunktionen - V3

- V2 + Style Loss L_{style}
  - Gram Matrix auf Perceptual Output X
    - $Gram = XX^T$
    - $Gram = X^{B\times\#C\times w\cdot h} * X^{B\times w\cdot h\times\#C}$
    - Batch Size B, number of channels C and width x heigh

Lossfunktionen - V4

- V3 + Total Variation Loss
  - $L_{tv} = \sum_{(i,j)\in R,(i,j+1)\in R} \frac{||I^{i,j+1}_{comp} - I^{i,j}_{comp}||_1}{N} + \sum_{(i,j)\in R,(i+1,j)\in R} \frac{||I^{i+1,j}_{comp} - I^{i,j}_{comp}||_1}{N}$
  - ```
    loss = torch.abs(self.comp[:, :, :, :-1] - self.comp[:, :, :, 1:]).mean() + \
        torch.abs(self.comp[:, :, :-1, :] - self.comp[:, :, 1:, :]).mean()
    return self.regularize_weight * loss
    ```
  - *comp: Vorhersage, bei welcher die nicht-maskierten Pixel auf den Ground Truth gesetzt werden*
- Glättung von evtl. Checkerboard Pattern
    ```

Final Loss

$$L_{total} = L_{valid} + 6L_{hole} + 0.05L_{perceptual} + 120(L_{style_{out}} + L_{style_{comp}}) + 2L_{tv}$$

Finales Training

- Erste Iteration
- Batch Size: 6
- BatchNormalization
- Learning Rate: 2e-4
- 7 Tage

Finales Training

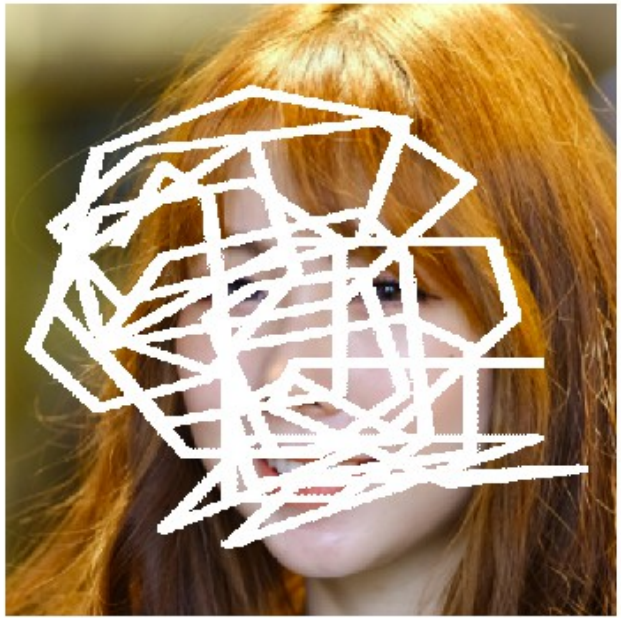
- Erste Iteration
- Batch Size: 6
- BatchNorm
- Learning Rate: 2e-4
- 7 Tage

- Finetuning
- Batch Size: 6
- BatchNorm nur in Decoder
- Learning Rate: 5e-5
- 7 Tage

➡ Final Loss

Vergleich: V0

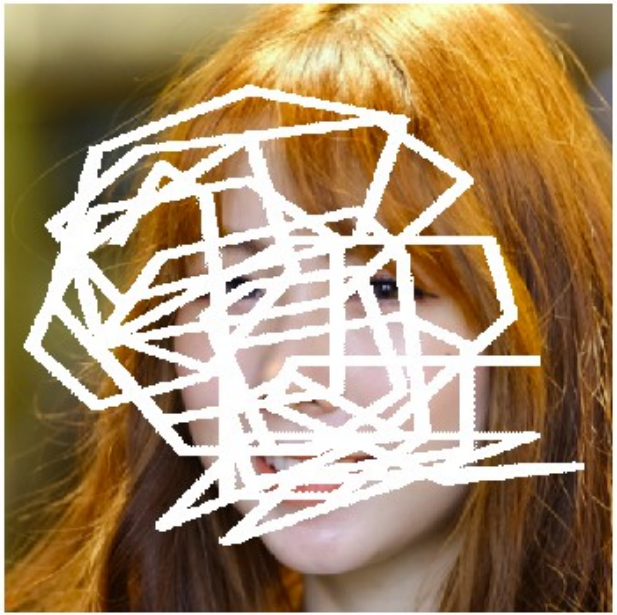
Image with Mask



Prediction

Vergleich: V1

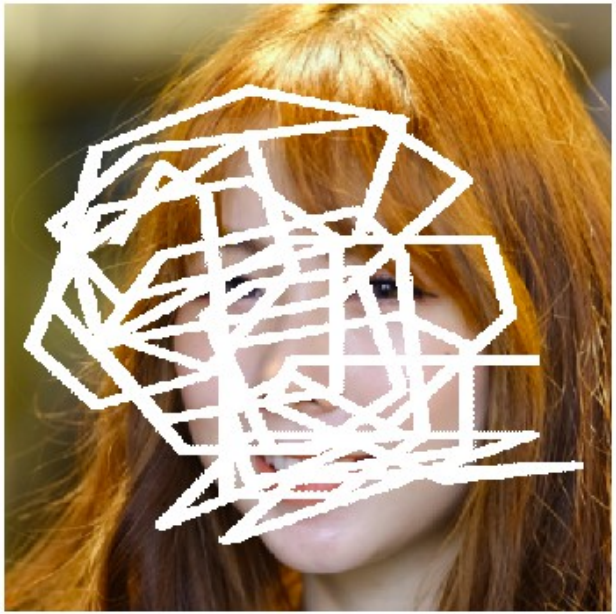
Image with Mask



Prediction

Vergleich: V2

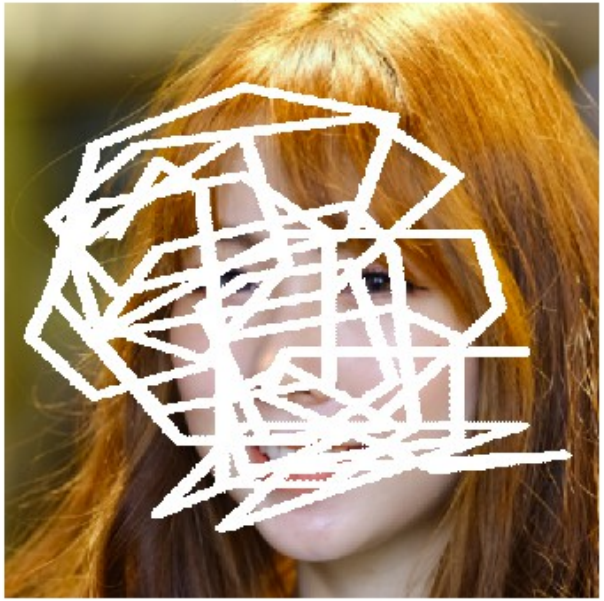
Image with Mask



Prediction

Vergleich: V3

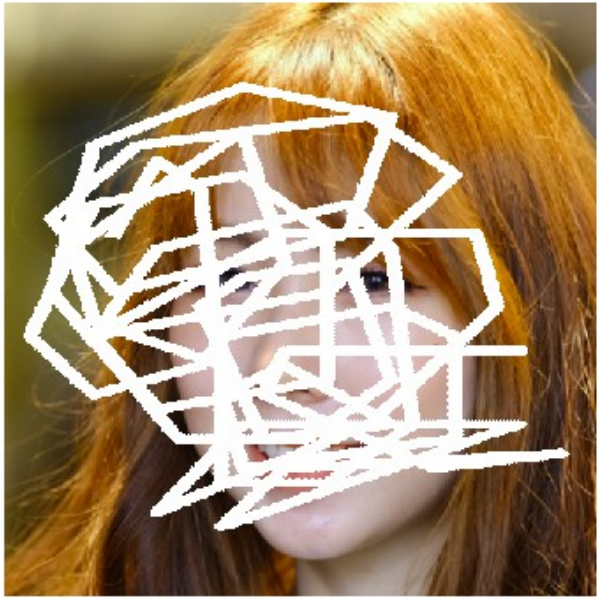
Image with Mask



Prediction

Vergleich: V4

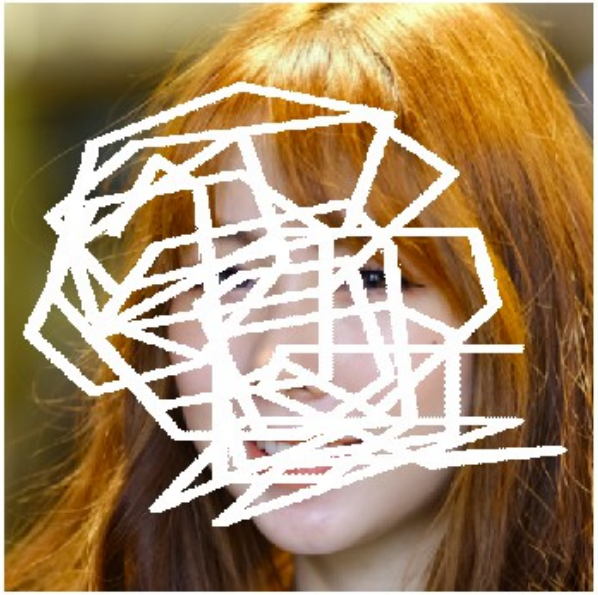
Image with Mask



Prediction

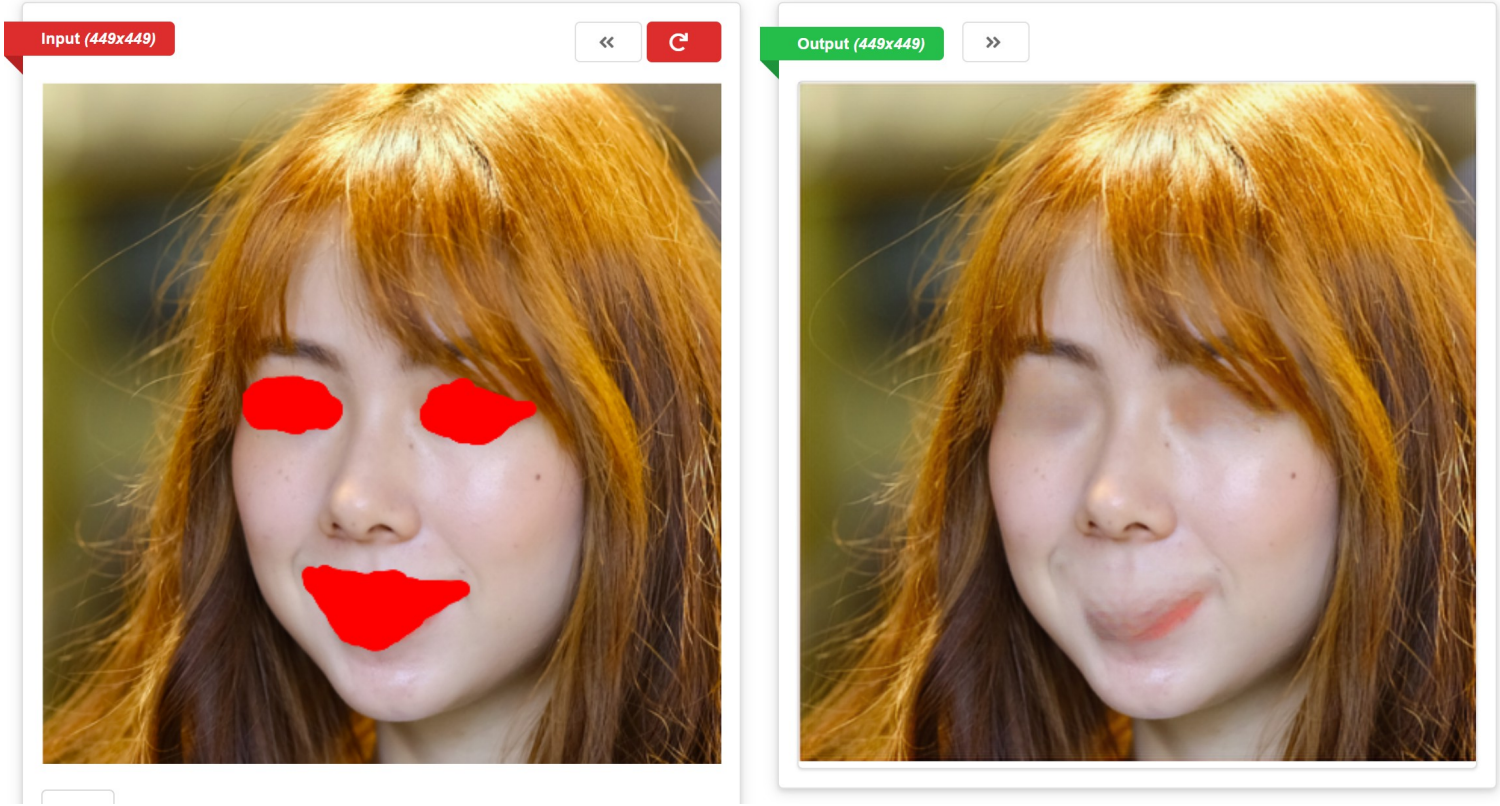
Vergleich: Final

Image with Mask



Prediction

Problem

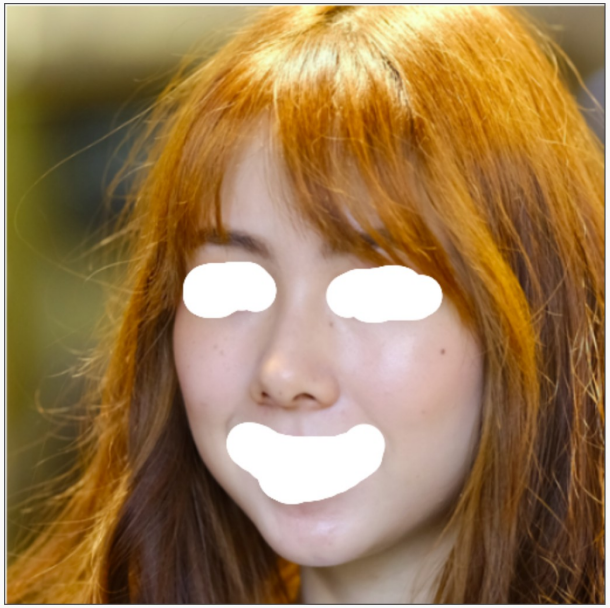


Problem

NVIDIA. RESEARCH AI PLAYGROUND

VOILA!

Inpainting complete.

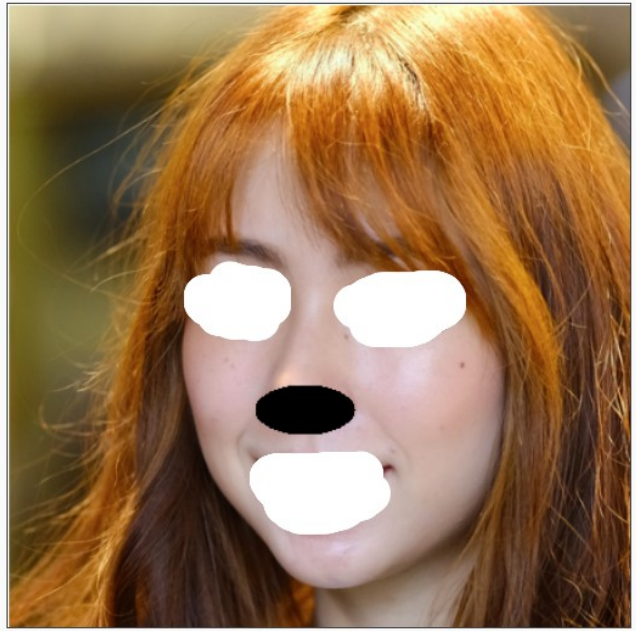


ORIGINAL IMAGE

INPAINTED RESULT

Problem

ORIGINAL IMAGE



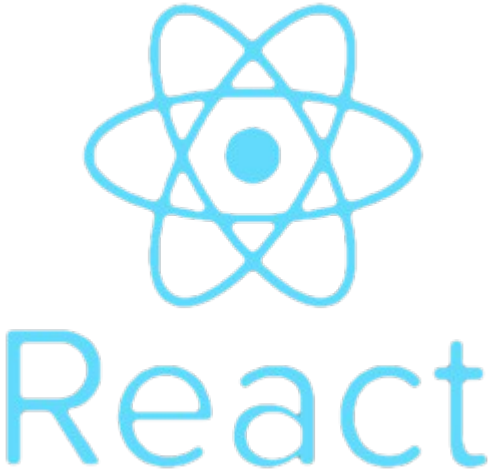
ORIGINAL IMAGE

INPAINTED RESULT

Problem

Sep. 2018: Our inpainting online demo is now available at https://www.nvidia.com/research/inpainting/ (Note: the natural image model is the consistent with model described in ECCV paper; the face image model has been further improved by using GAN loss to train the same network after ECCV. We also suggest to do continuous inpainting, uploading the inpainting results to do second-time inpainting, to get better results.)

Benutzeroberfläche



Benutzeroberfläche

Demo Time

Literaturverzeichnis

- https://github.com/NVlabs/ffhq-dataset
- https://github.com/karfly/qd-imd
- https://arxiv.org/abs/1505.04597 U-Net
- https://arxiv.org/pdf/1804.07723.pdf Partial Convolution

Vielen Dank