

# Neural Style Transfer

Agustinus Kristiadi

University of Bonn

24 Jan 2018

# Outline

## Introduction

Motivation

Introduction

## Lesson Learned

## Experiments

Styles used

Results

## Conclusion

# What is style transfer?

- ▶ Given an image, can we extract its style and apply it to other images?



Figure 1: From: Gatys, 2015

# Method of Johnson, 2016

- ▶ Two networks:
  - ▶ Image transform network (Deconv Net)
  - ▶ Loss network (VGG Net)
- ▶ Two losses: content loss + style loss

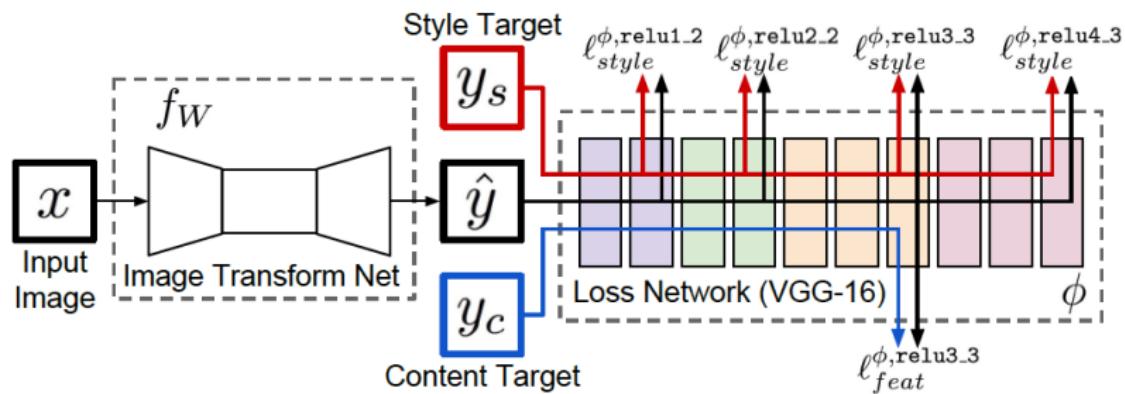


Figure 2: From: Johnson, 2016

## Lesson Learned

- ▶ 20,000 training images is enough. Full epoch ( 160k imgs) for COCO2014 needs 3 hours of training!
- ▶ No free lunch: no hyperparams work for all styles. Also more on this later.
- ▶ Always use InstanceNorm instead of BatchNorm. More on this later.
- ▶ All of these images henceforth will use InstanceNorm unless specified.

# Style images



Figure 3: Style images as in Johnson 2016

# Style images



Figure 4: Our own style images

## Effect of noise

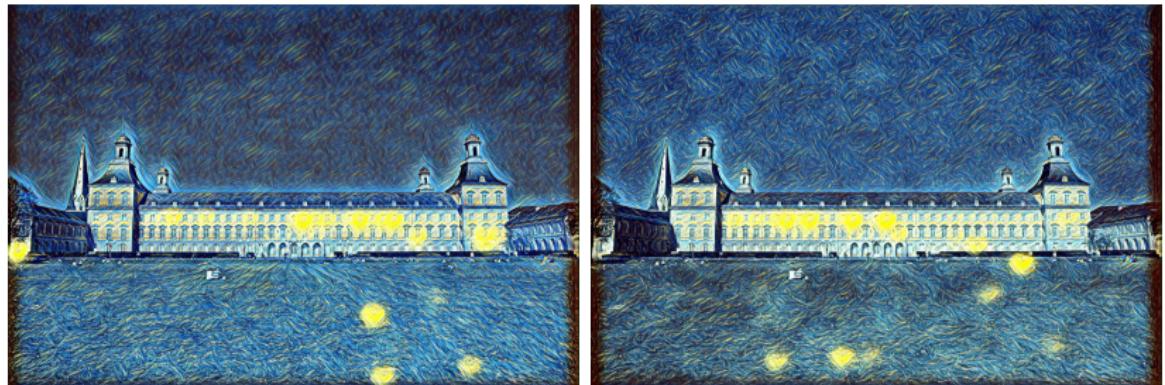


Figure 5: Starry Night: Clean vs Noisy

## Effect of clutter



Figure 6: Starry Night: Clean vs Clutter

## Outdoor vs indoor

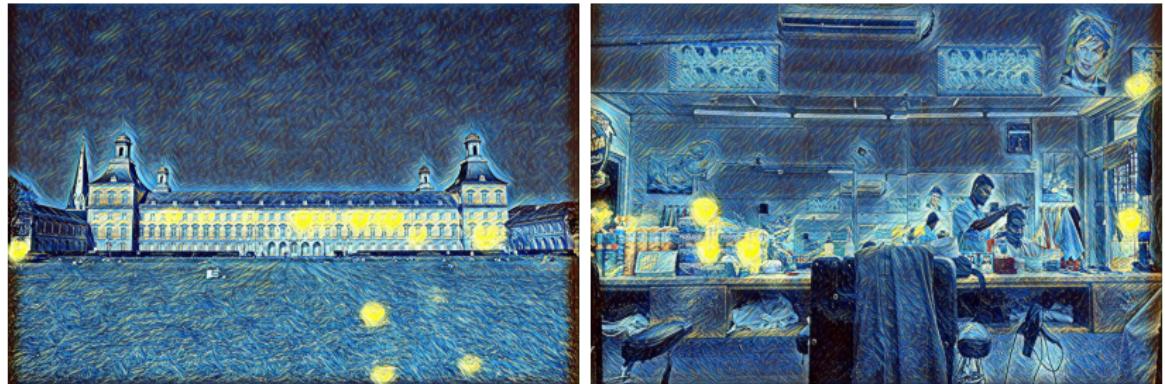


Figure 7: Starry Night: Outdoor vs Indoor

## Effect of image size



Figure 8: Great Wave of Kanagawa: Small vs Large

## New styles

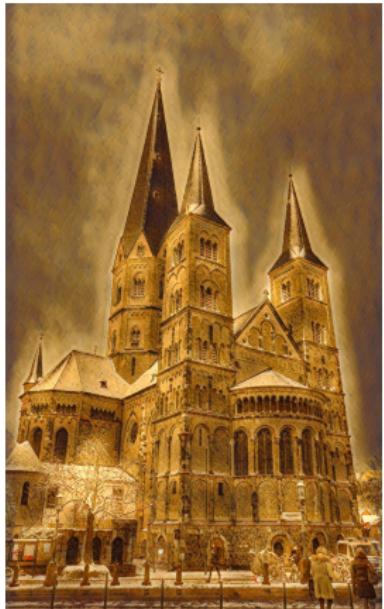
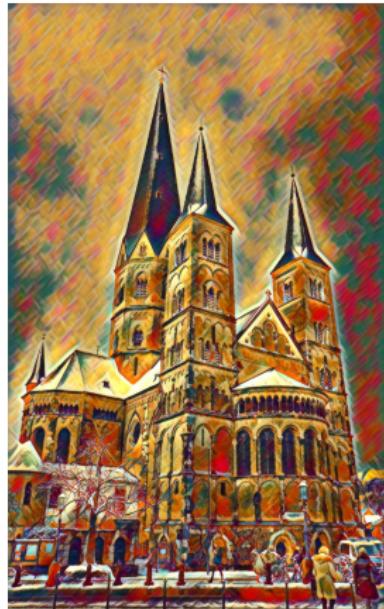


Figure 9: New styles

# InstanceNorm vs BatchNorm



Figure 10: IN vs BN

# Only one loss

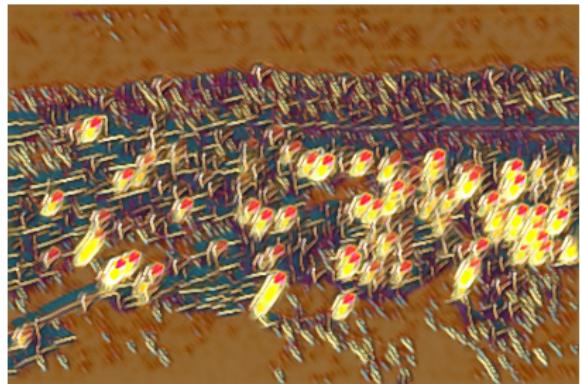


Figure 11: Only content vs only style

## More style vs more content

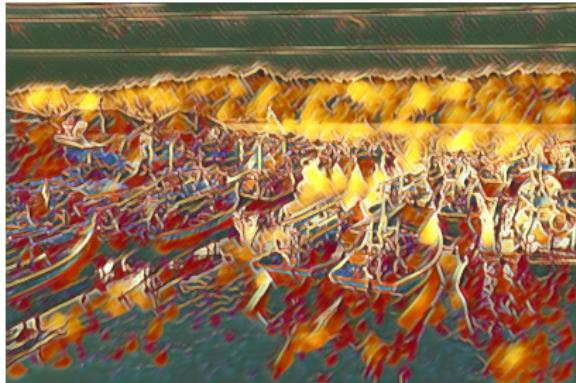


Figure 12: More content vs more style

## Content loss: effect of layer choice



Figure 13: relu12 vs relu22 vs relu33 vs relu43

## Style loss: effect of layers choice

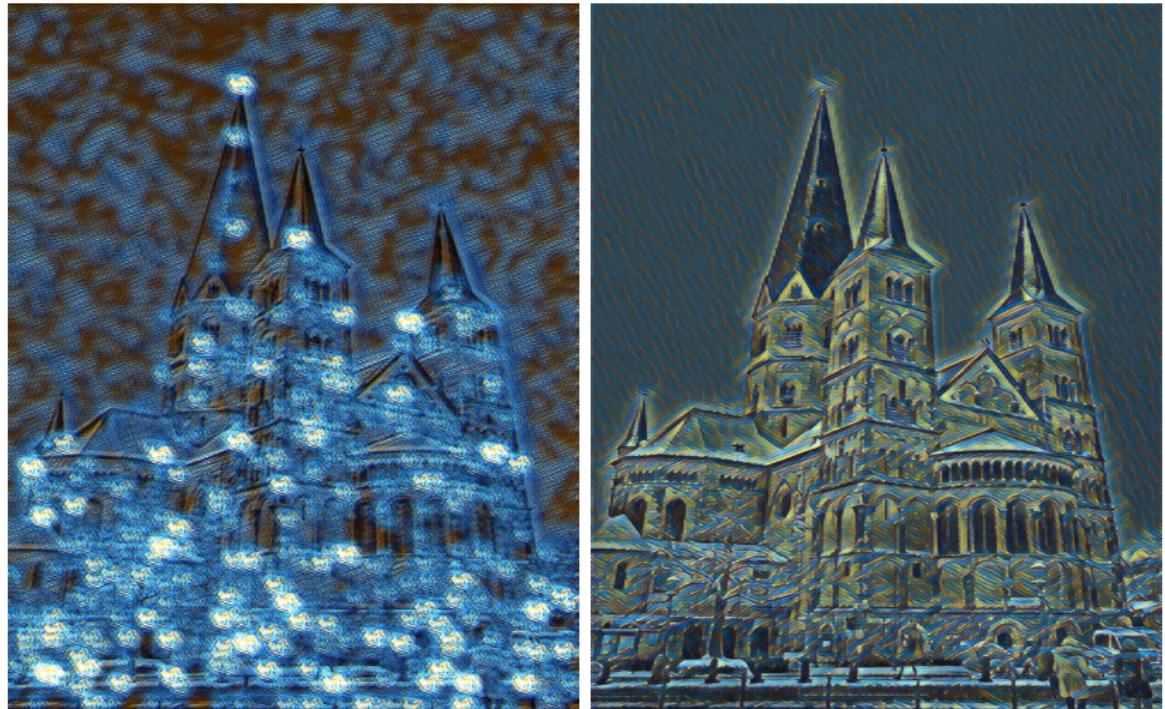
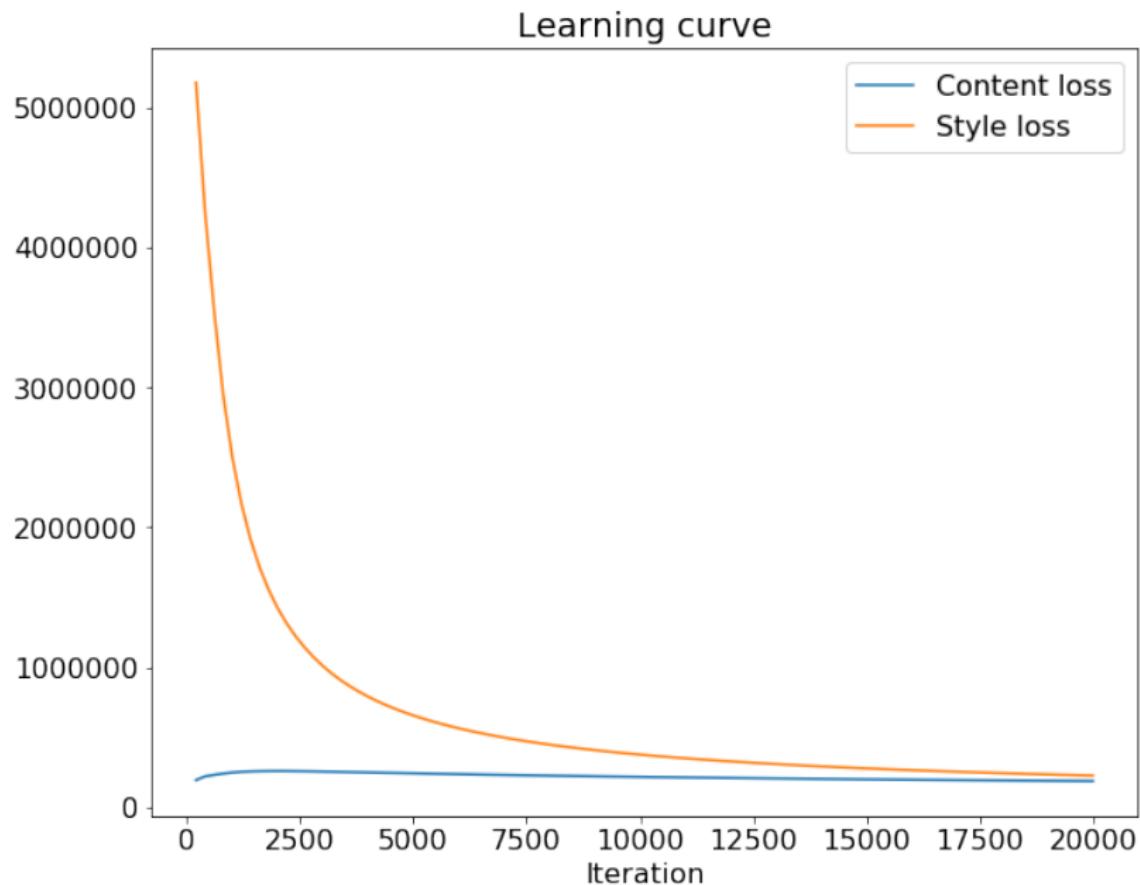
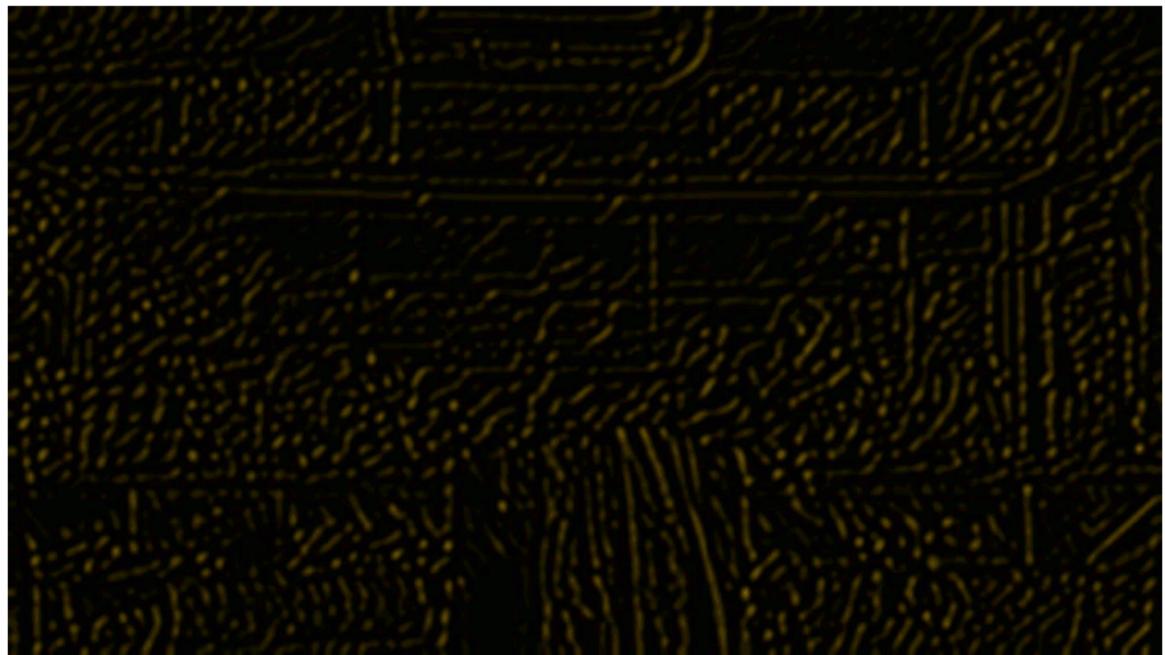


Figure 14: Lower layers vs upper layers

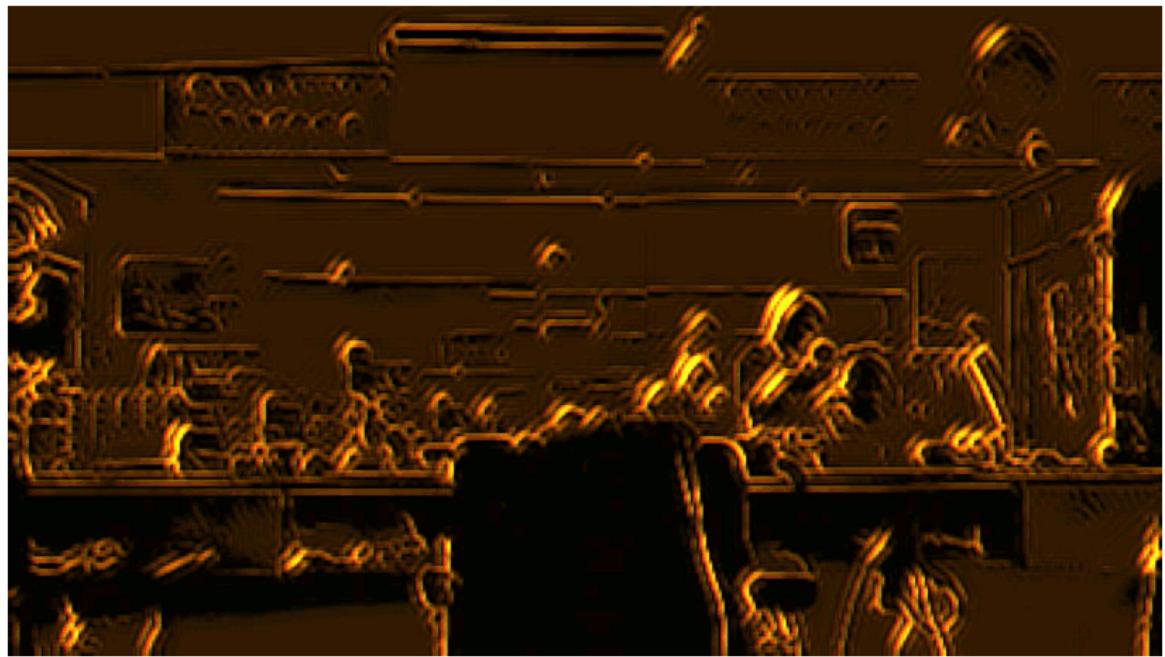
# Learning curve



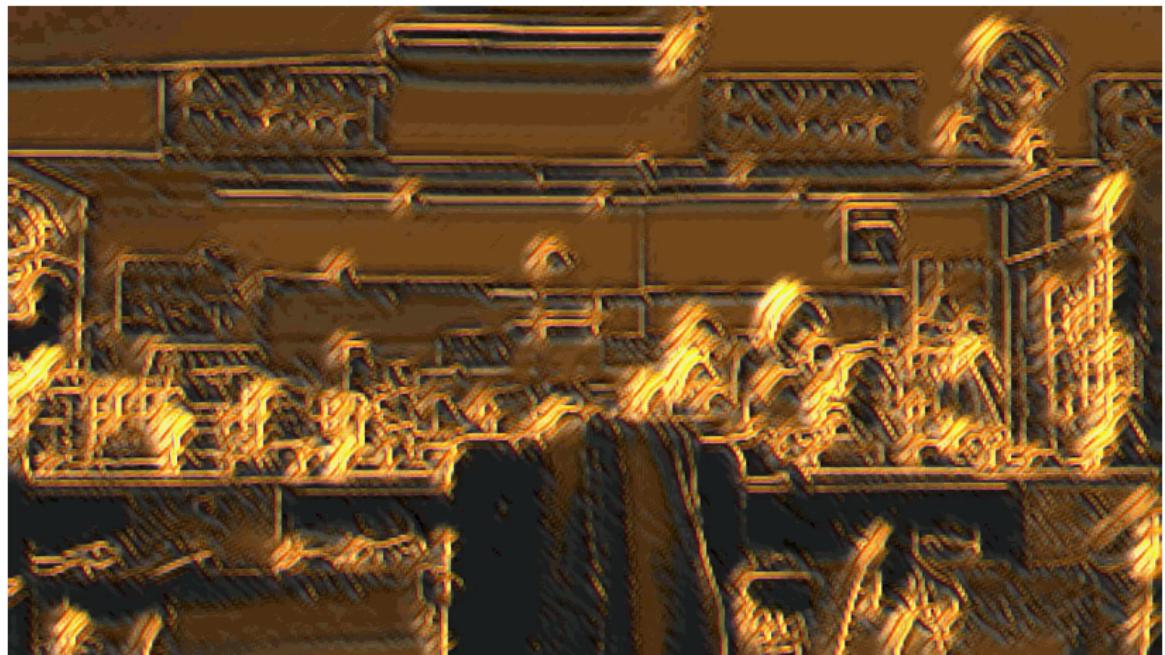
## Stylization progression: iter 100



## Stylization progression: iter 500



## Stylization progression: iter 1000



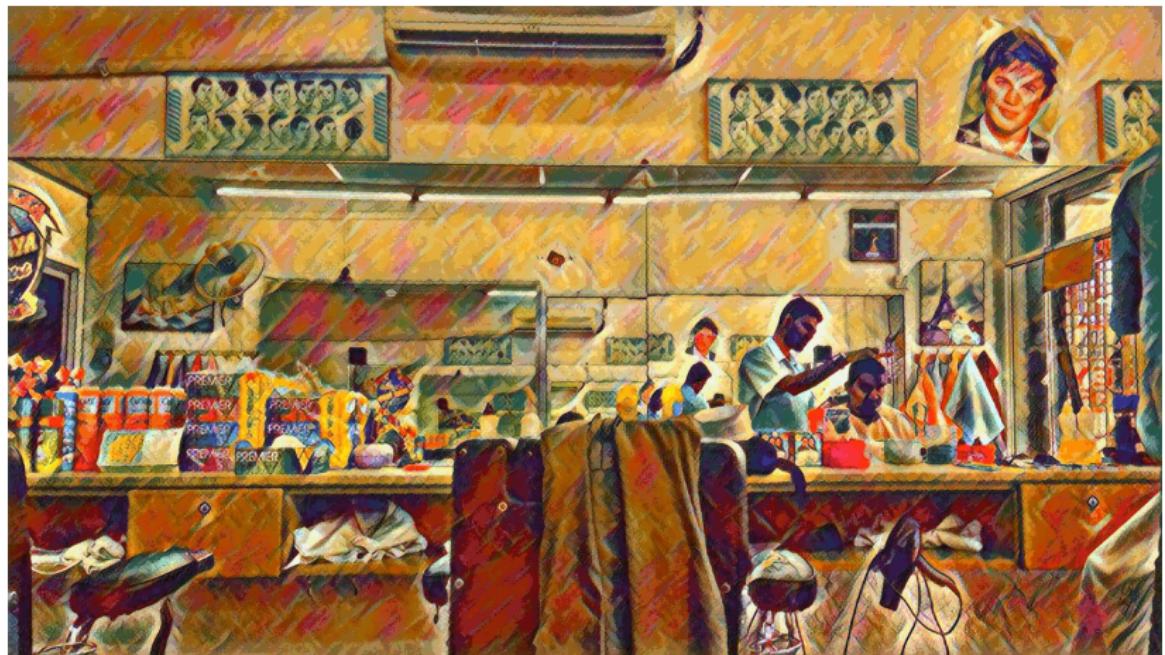
## Stylization progression: iter 5000



# Stylization progression: iter 10000



# Stylization progression: iter 20000



## Conclusion

- ▶ Johnson 2016 architecture is fast.
- ▶ InstanceNorm is a must.
- ▶ Careful tuning needed for different styles.

## References

- ▶ Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).
- ▶ Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European Conference on Computer Vision. Springer International Publishing, 2016.
- ▶ Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Instance normalization: The missing ingredient for fast stylization." arXiv preprint arXiv:1607.08022 (2016).