

Colorful Image Colorization with Tensorflow

Sophia Schulze-Weddige

Malin Spaniol

Maren Born

Integrating Artificial Neural Networks with Tensorflow

Universität Osnabrück

February 15, 2020

1 Introduction/Motivation

- we want to build a network that can colorize images
- this project aims to produce colorful images, given a greyscale picture.
- transforming greyscale into plausible colors is an easy task for humans
- We see a greyscale picture showing a woman playing volleyball at the beach. As we can recognize the scene and the form and relate to it. The sand is yellow, the sea is blue and the ball is white.
- But coloring it in life would be a much more difficult task. As we also need to consider different textures, shades and so on. Seeing and imagining things does not make people a proper painter.
- Surface structure and the semantics of the scene are necessary to validly color images.
- this project aims does not aim to generate the true color for pictures but at least a good and prediction.
- aus dem paper: model enough of the statistical dependencies between semantics and the textures of greyscale images and their color versions in order to produce visually compelling results.

2 Important background knowledge (including reference to most relevant publications)

```
#packages needed  
import numpy as np
```

```

import tensorflow as tf
import cv2
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Activation,
    BatchNormalization
    for loop
'ich bin ein string'
if
return
MyClass, __init__

```

3 “In-text” listing highlighting

- We started looking at this paper:

<https://arxiv.org/abs/1603.08511> ,

- at their best solution

4 The model and the experiment (MAIN PART). This part should feature code.

4.1 Dataset

- loading large amount of data

<https://machinelearningmastery.com/how-to-load-large-datasets-from-directories-for-deep>

-

4.2 Preprocessing

- image preprocessing documentation

<https://keras.io/preprocessing/image/#imagedatagenerator-class>

- preprocessing via ImageDataGenerator() from keras.preprocessing.image

- takes in traindata, validation data and test data

- featurewise-center and featurewise std

- classmode: none -> is for predictions

4.3 Layer

4.4 Loss-Function

5 Visualization and discussion of your results.

5.1 Training

5.2 Testing