Universal Style Transfer via Feature Transforms

Team Name : Udolhtrap

Content

Problem Statement

Proposed Method

Results Expected

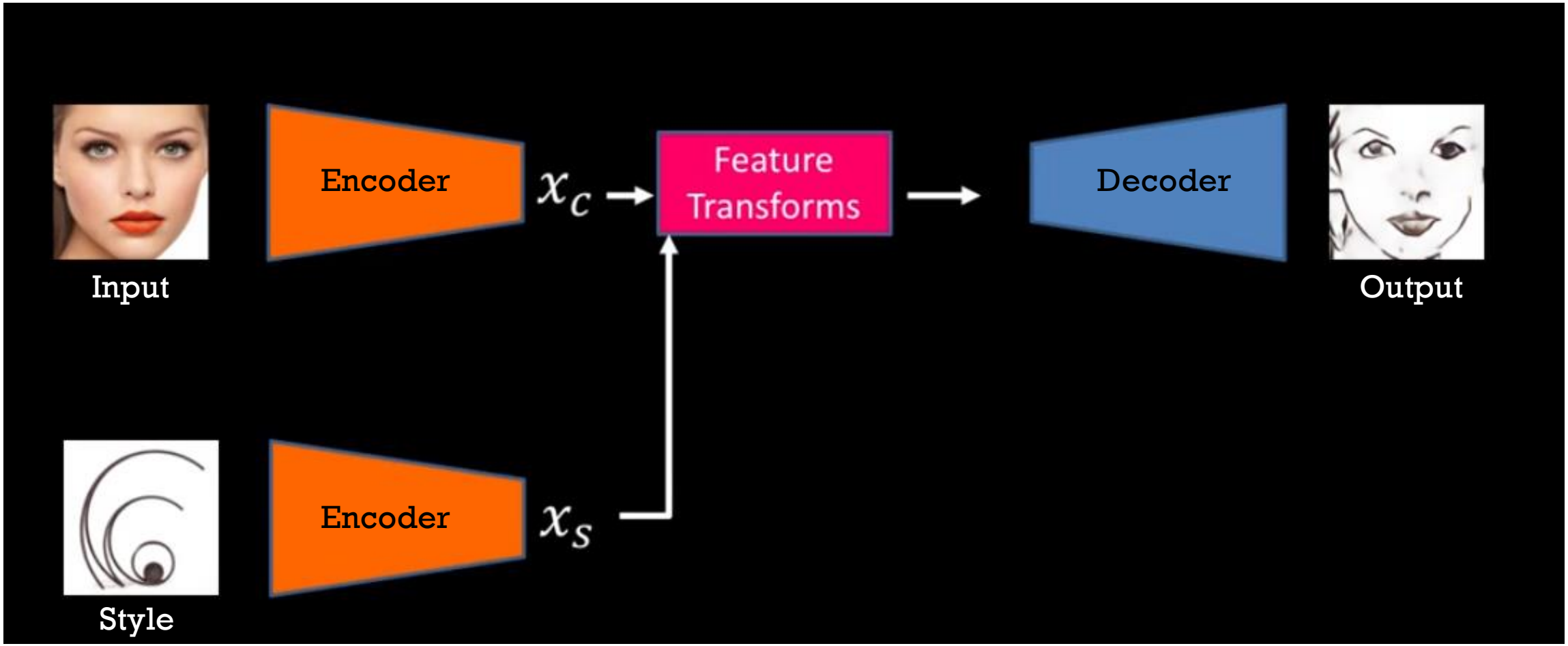
Timeline

Universal Style Transfer via Feature Transforms

- Universal style transfer aims to transfer arbitrary visual styles to content images.
- We propose a feed forward method to realize the fast transferring for arbitrary styles.
- A pair of feature transformations, whitening and coloring is embedded in an image reconstruction network.
- We present a effective method that does not require training on any pre-defined styles.

Method Overview

- We will use a VGG-19 network as encoder to extract features. A decoder is then trained to reconstruct original image.
- Original Image and style are both input to the encoder. The combined output is fed to a Whitening and coloring(WCT) module which transform features to match the style.
- The output from WCT is sent to the trained decoder to get the final styled image.
- For higher visual quality multi-layer pipeline is used.


Input
Encoder
x_c
Feature Transforms
Decoder
Output
Style
Encoder
x_s

WCT

- A whitening transformation or sphering transformation is a linear transformation that transforms a vector of random variables with a known covariance matrix into a set of new variables whose covariance is the identity matrix, meaning that they are uncorrelated and each have variance 1.
- A coloring transformation transforms a vector of white random variables into a random vector with a specified covariance matrix

Whitening and Coloring Transforms (WCT)

- Whitening

$$\widetilde{x_c} = E_c \, D_c^{-1/2} E_c^T x_c$$

Eliminate the correlations between content features

$$\mathrm{Cov}(\widetilde{x_c}) = I$$

- Coloring

$$\widehat{x_c} = E_s \, D_s^{1/2} E_s^T \widetilde{x_c}$$

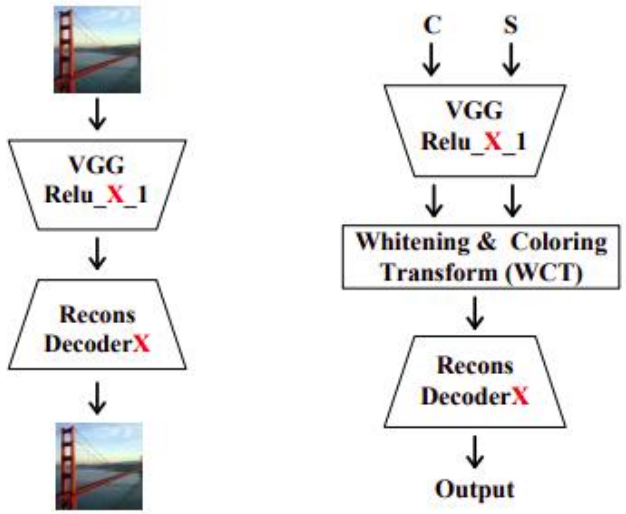
Let the features have the same correlations with those of the style

$$\mathrm{Cov}(\widehat{x_c}) = Cov(x_s)$$

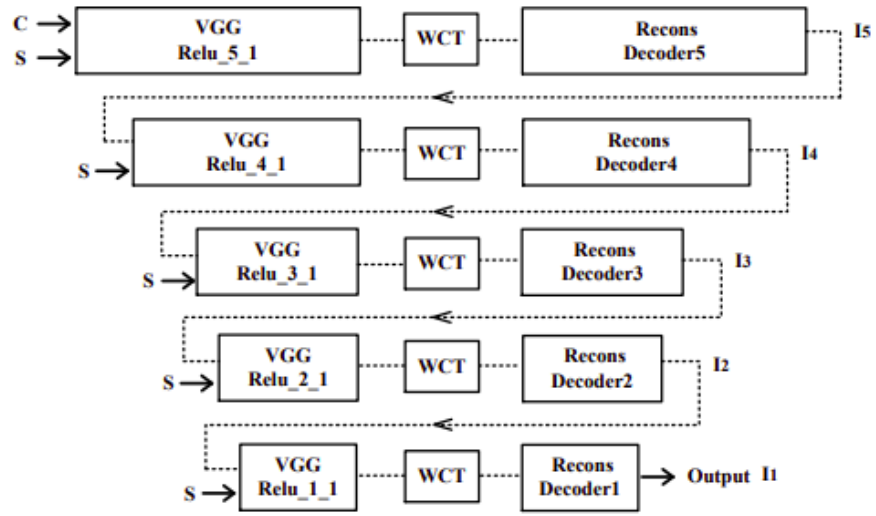
Content (or style) feature $x^{C \times HW}$

$x^{C \times HW} \rightarrow Cov_x$

- D the diagonal matrix of its eigenvalues
- E the orthogonal matrix of its eigenvectors



Single-level stylization pipeline

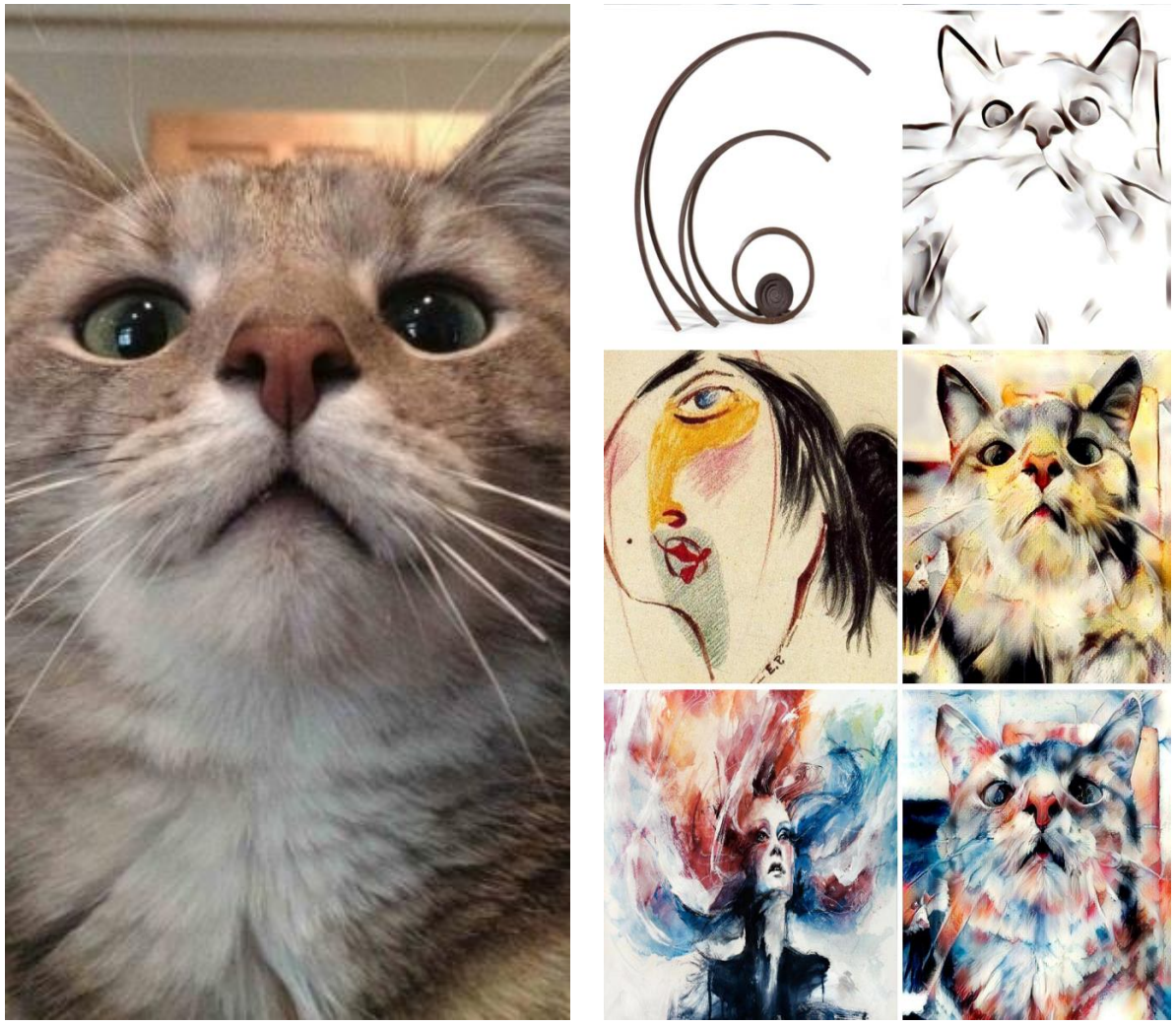


Multi-level stylization pipeline

Frameworks

We'll be using the Pytorch framework to implement the encoder and decoder models.

Input Style Output



Results Expected

Project Timeline

Mid-March – Encoder/Decoder + Whitening module

April – Coloring module and multi-level stylization

Final Deliverable – Complete working pipeline for single-level and multi-level stylization

Thank You

Questions ?