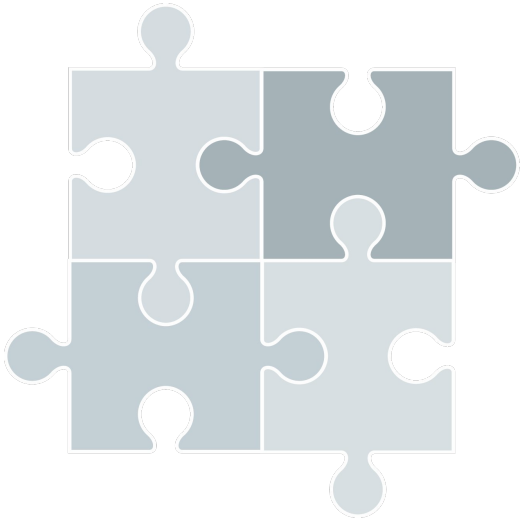
VIDEO PROCESSING USING STYLE TRANSFER

WITH CONVOLUTIONAL NEURAL NETWORK

Burdziński, Bończyk, Sobkowiak, Świda
dr hab. inż Wojciech Kotłowski

Overview

1. Project objective
2. Mobile Application
3. Server
4. Neural network
5. Pruning
6. Features
7. Results
8. Demo
9. Future work



Artistic Style Transfer

Motivation

- **new possibilities** in the domain of video processing
- neural style transfer approach is still **fresh**
- gap in the **real-time** video processing area
- existing algorithms are **not fast** enough
- deficit of availability of free and **easy-to-use** applications

Project objective

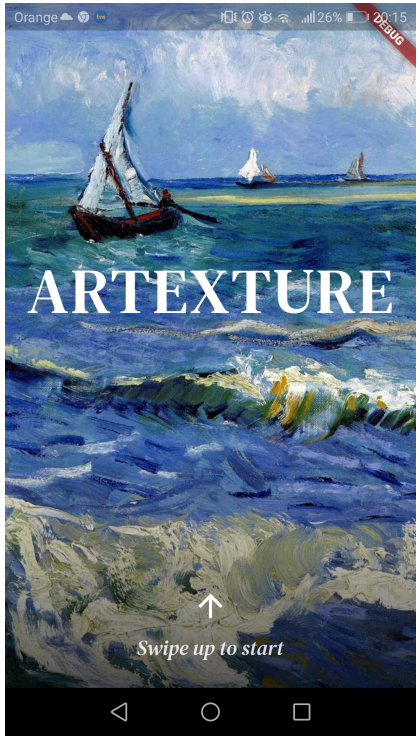
to enable fast real-time video processing using artistic style transfer by speeding up the existing convolutional neural network and ensure easy access by mobile application and lightweight server

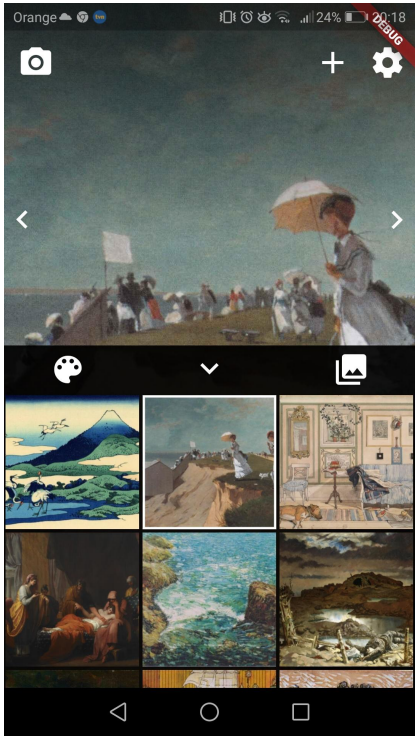
Project objective

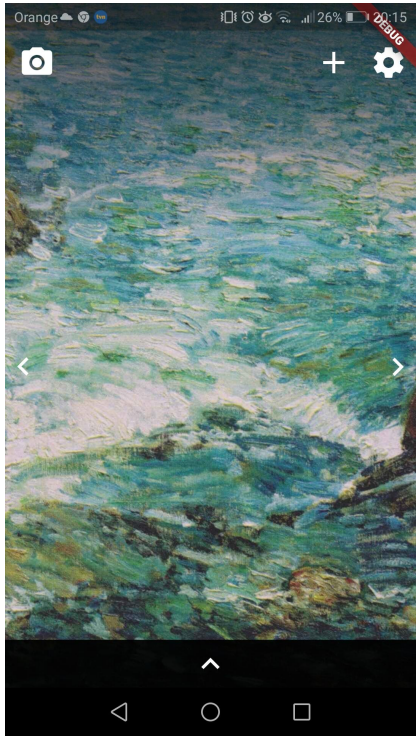
- reduce **computational time** by use of network pruning and TensorRT
- maintain good video **quality** (1024x576)
- achieve a comfortable **frame rate** (~24 FPS)
- create user-friendly and **cross-platform** mobile application
- develop server which connect application and algorithm **without delays**

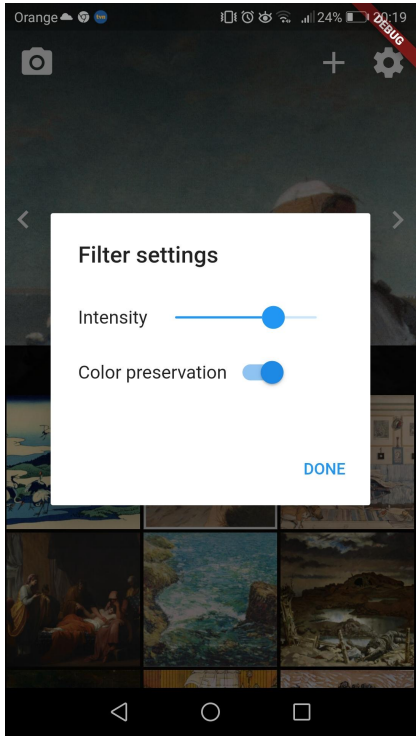
Mobile application

Mobile application



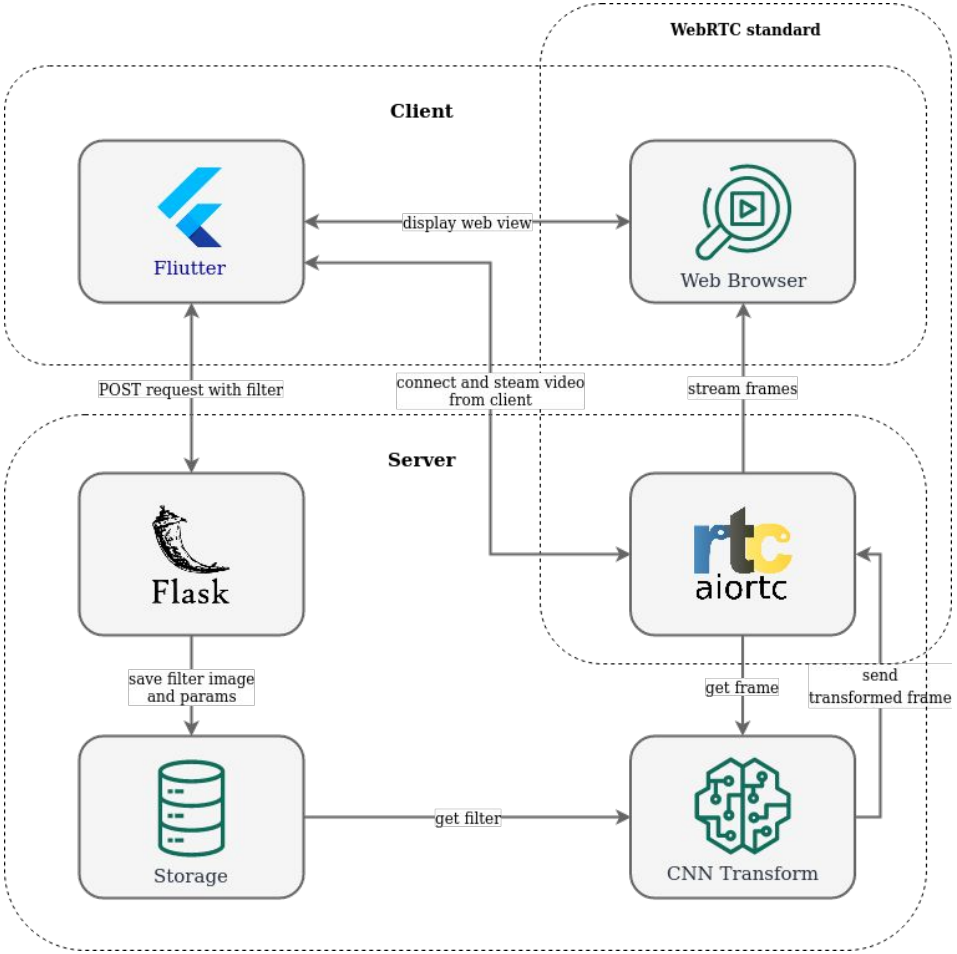




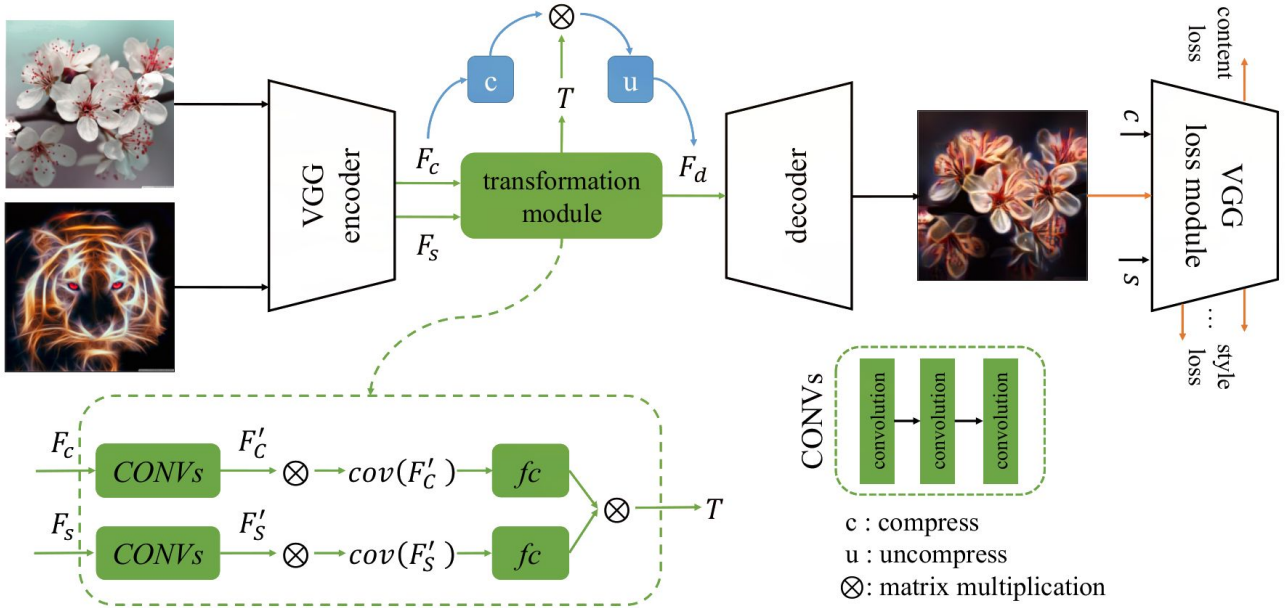


Server

- real-time communication
- direct peer-to-peer
- for web browsers / Android / iOS
- supported by Google


WebRTC standard
Client
Fliutter
display web view
Web Browser
POST request with filter
connect and steam video
from client
stream frames
Server
Flask
aiortc
save filter image
and params
get frame
send
transformed frame
Storage
get filter
CNN Transform

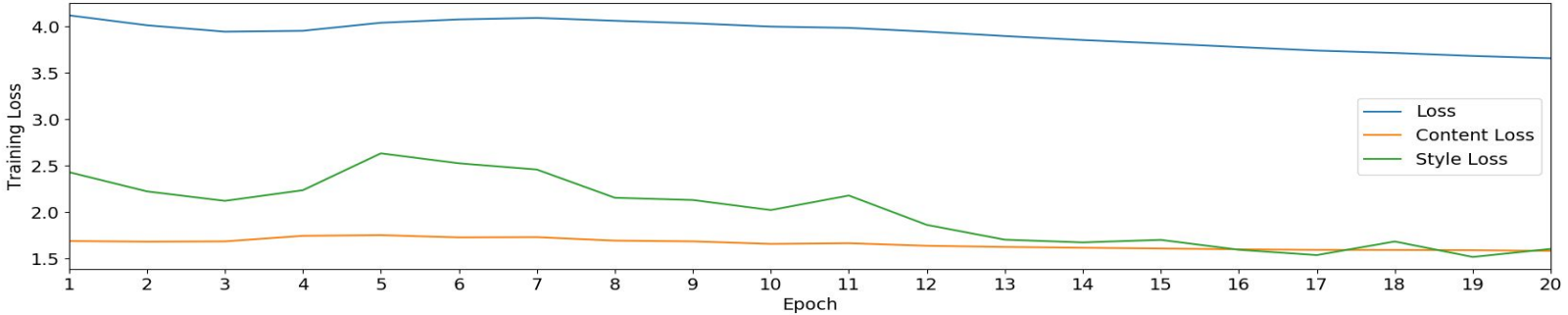
Neural network

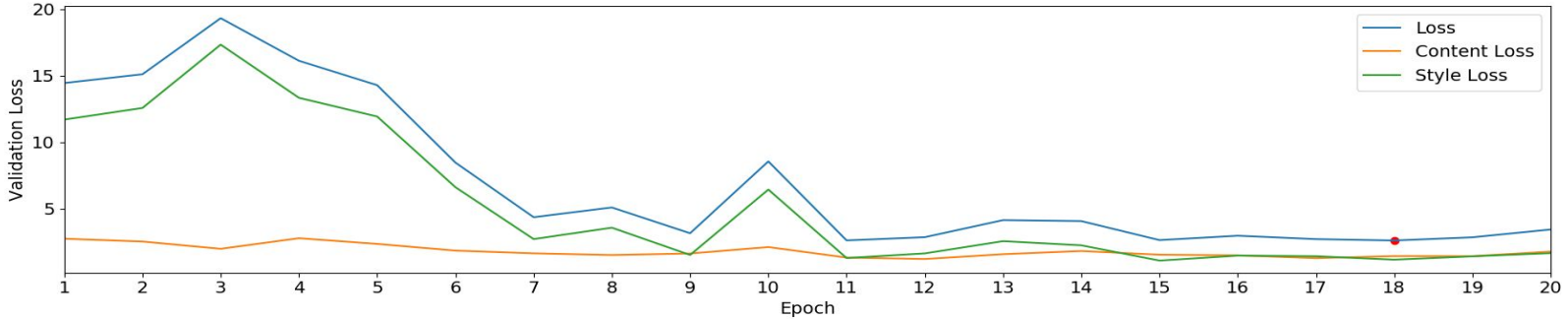


Dataset

- Pretrained model available at https://github.com/sunshineatnoon/LinearStyleTransfer
- Network pruning with distiller library https://github.com/NervanaSystems/distiller
- Inference speedup with NVIDIA TensorRT framework
- Network pruning requires retraining
- content - MSCOCO, style - Wikiart
- Training with 72000 sample pairs; validation with 8000 pairs
- Unsupervised learning

Loss function





Style scaling

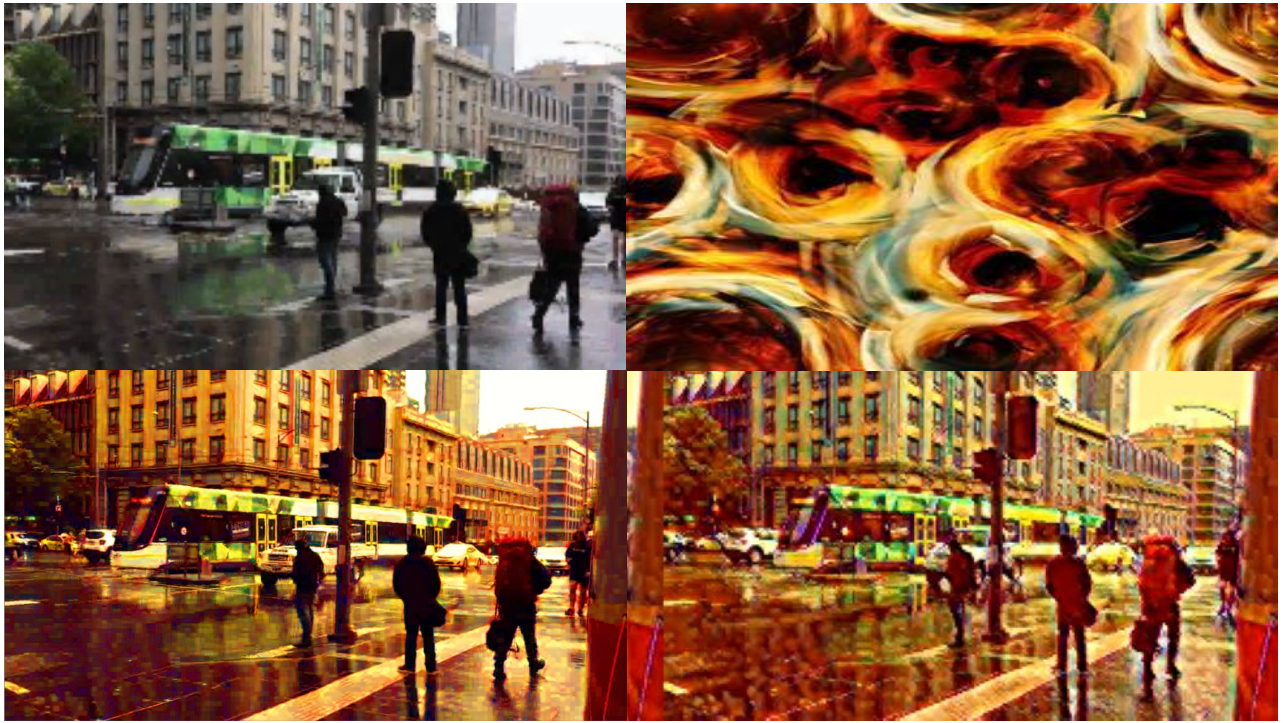
α = 0 α = 0.1 α = 0.2 α = 0.3

α = 0.4 α = 0.5 α = 0.6 α = 0.7

α = 0.8 α = 0.9 α = 1 Style

Color preservation

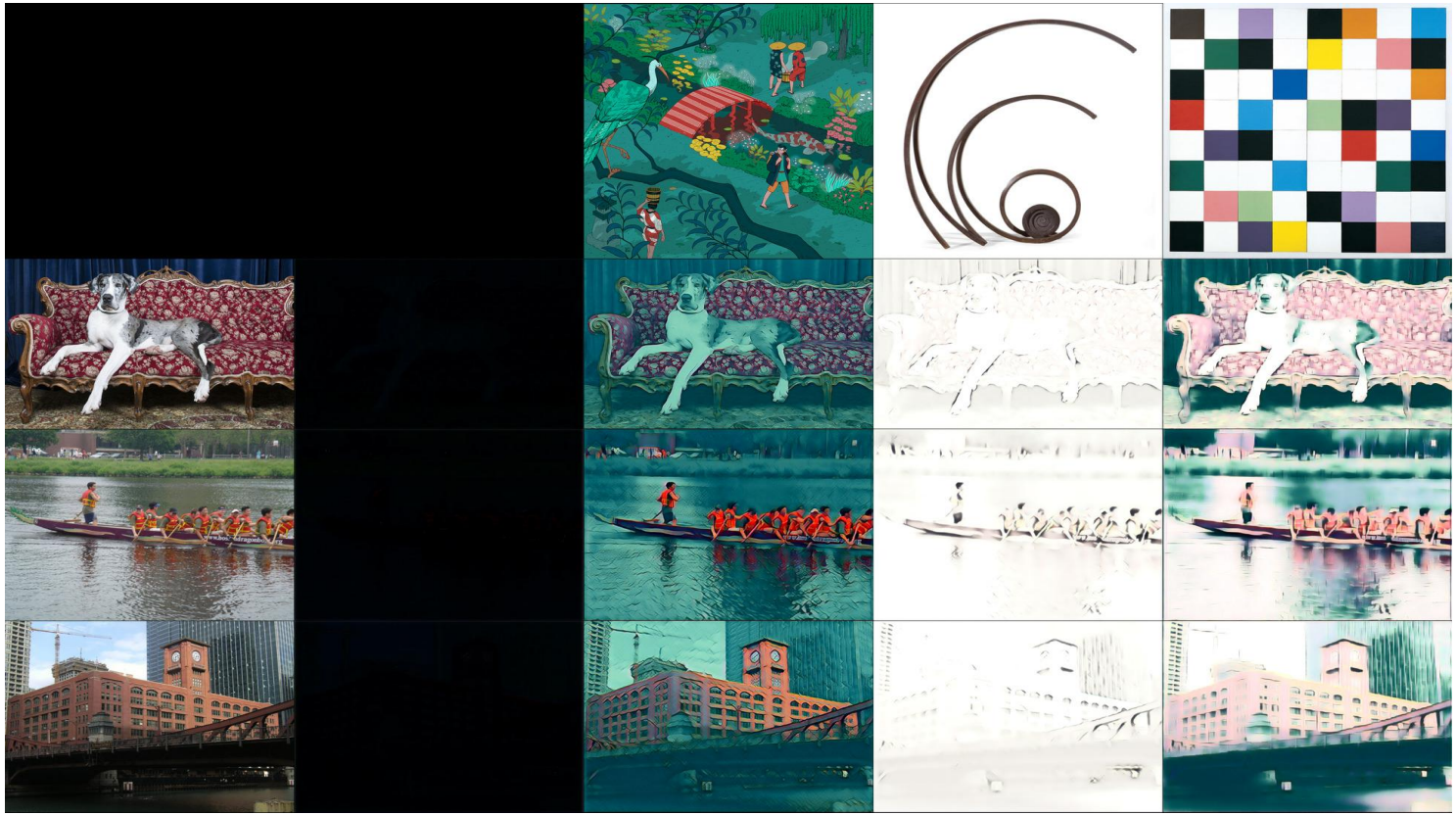
Color preservation



Results

	940M	960	1080 Ti
Original model	0.75	5	23
Pruned model	4.2	22	79
Pruned + TensorRT	7.1	26	70

Mosaic



Demo


Orange
51%
15:42

Future Works

- Feature encoding
- More sophisticated training
- Style interpolation
- Mixed styles using spatial control