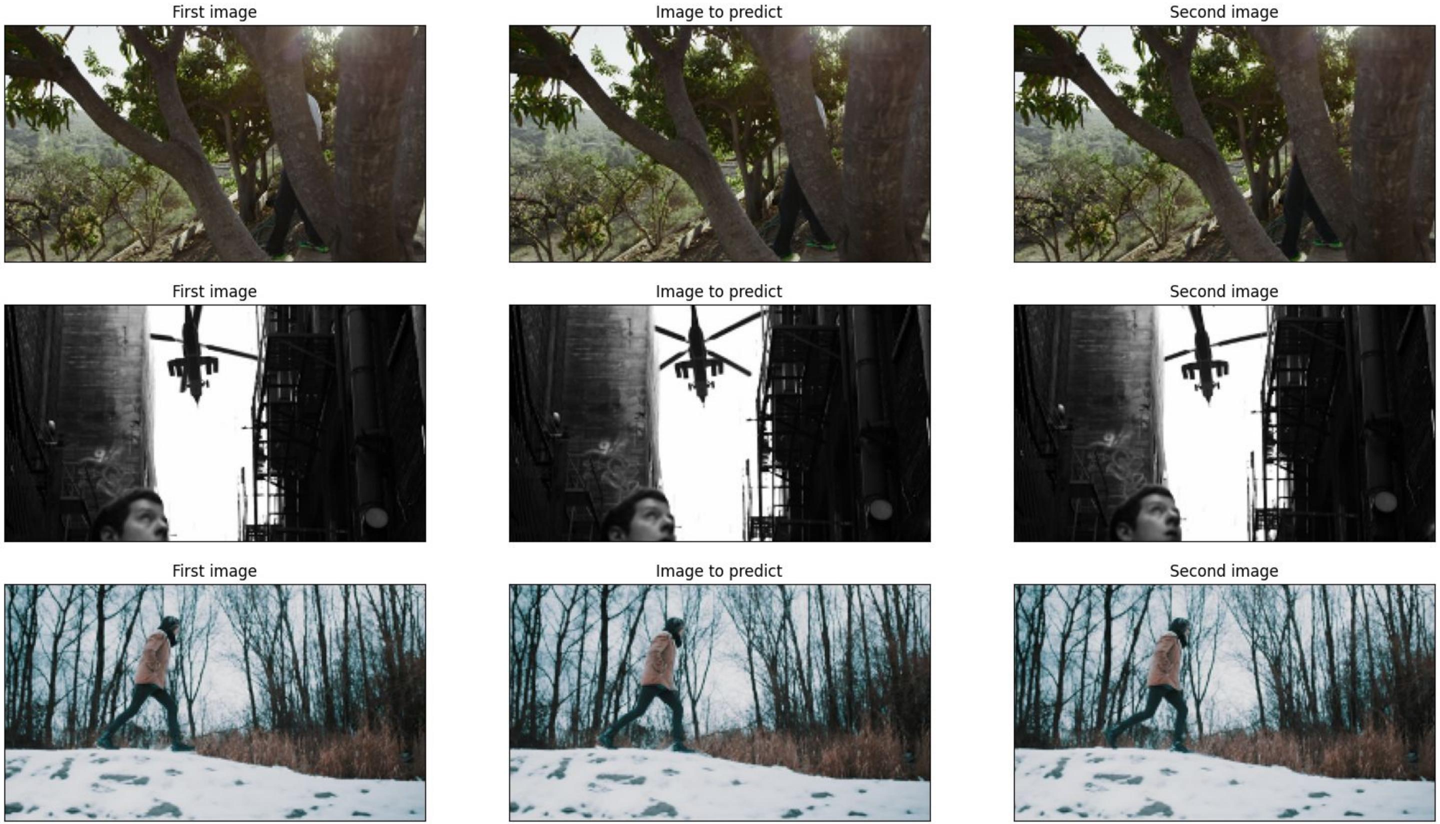
Frame Booster

Kamil Barszczak

Frame interpolation

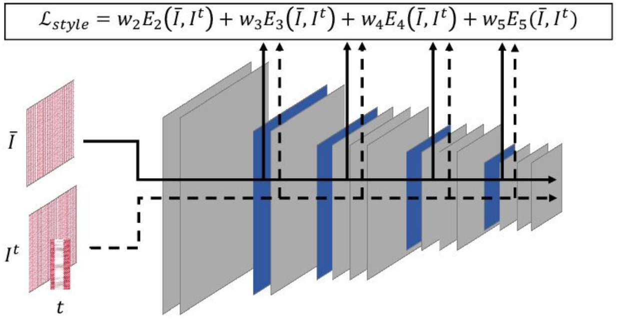


Dataset

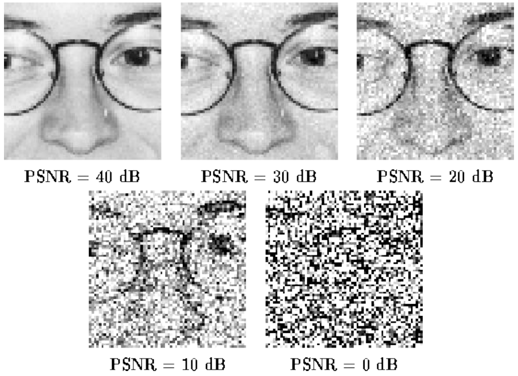
- Vimeo90K - triplet (256x144px RGB)
    - 50000 testing samples
    - 3000 testing samples
    - 1000 validating samples

Loss function

Perceptual Loss:



PSNR:



MAE: $$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MSE: $$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Loss = 0.5*Perceptual + PSNR + 5.0*MAE + 10.0*MSE

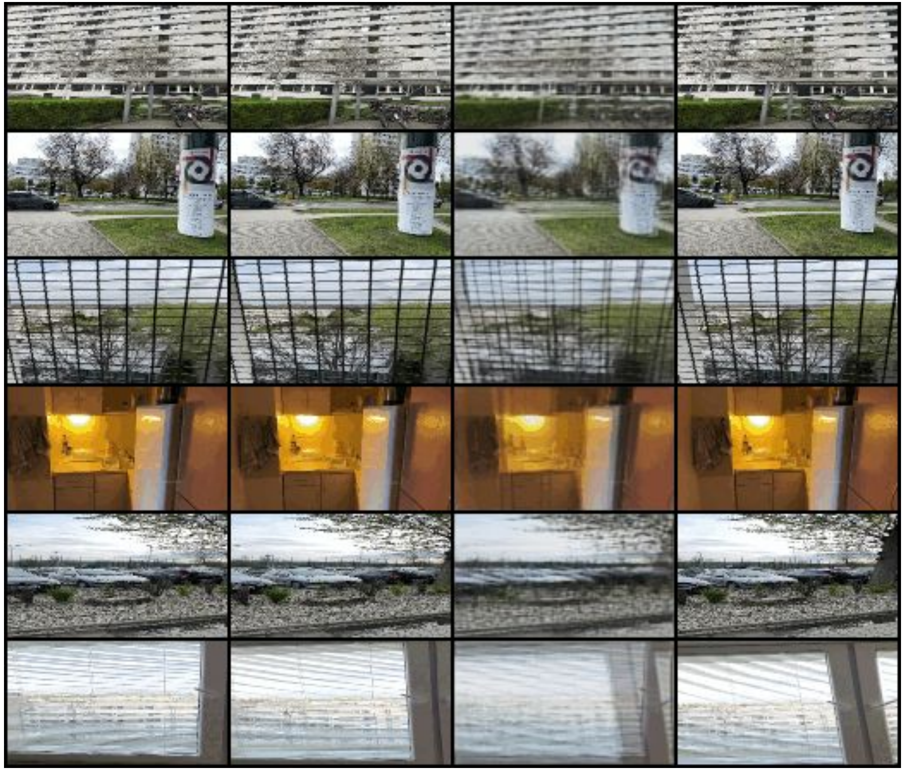
Trening

- Tested optimizers
  - Nadam (lr=0.0001)
  - Adam (lr=0.0001)
- Epochs: ~5
- Batch size: 2
- Metrics: MAE, MSE, PSNR
- Average epoch time: ~2.5h
- GPU used: GTX970 4GB

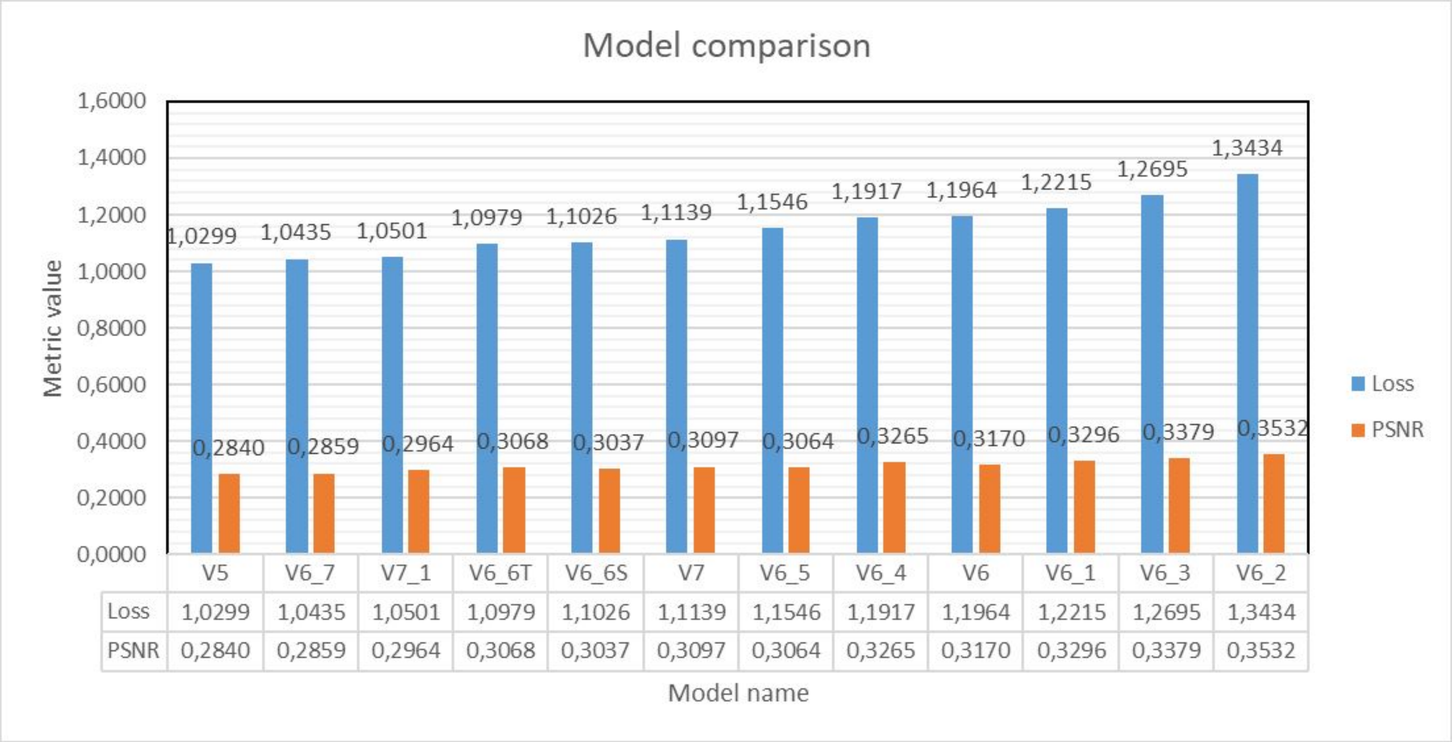
Benchmark

- Automatic tool for testing multiple models sequentially
- Saves training progress, model state, and other features such as activation masks
- Executes training on a smaller training dataset (5000 samples of 128x72px RGB images)
- Saves the learning history (measurements for each step)
- Creates png files with training and validating histories
- Evaluates the training on the testing dataset

Example of training progress captured by the benchmark:



Models results (benchmark)



Best 3 Models

Model V5

- U-Net type architecture (with warping layers between the encoder and the decoder)
- Warping both images
- Feedforward flow prediction CNN
- Encoder with skip connections (same as FILM model)
- Sigmoid activation

Model V6_7

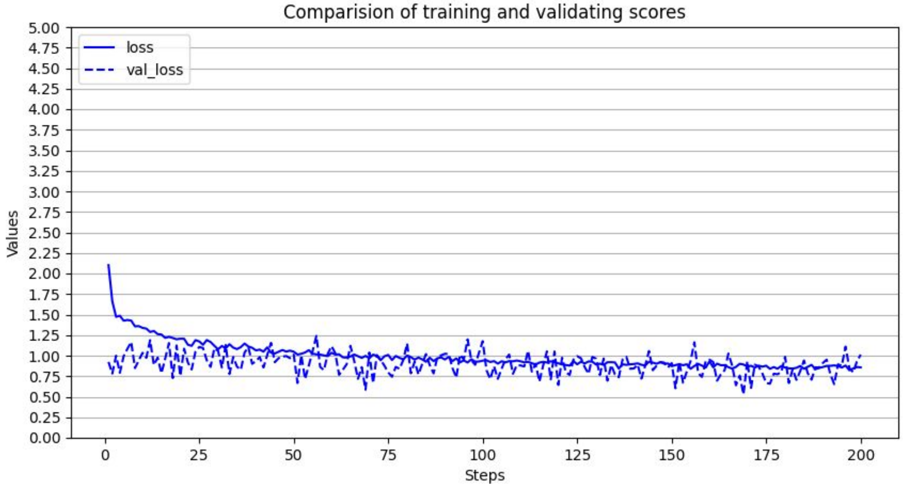
- Attention U-Net type architecture (with warping layers between the encoder and the decoder)
- Warping both images
- DenseNet flow prediction CNN
- Encoder same as model V5
- Clipped linear activation (into the range from 0 to 1)

Model V7_1

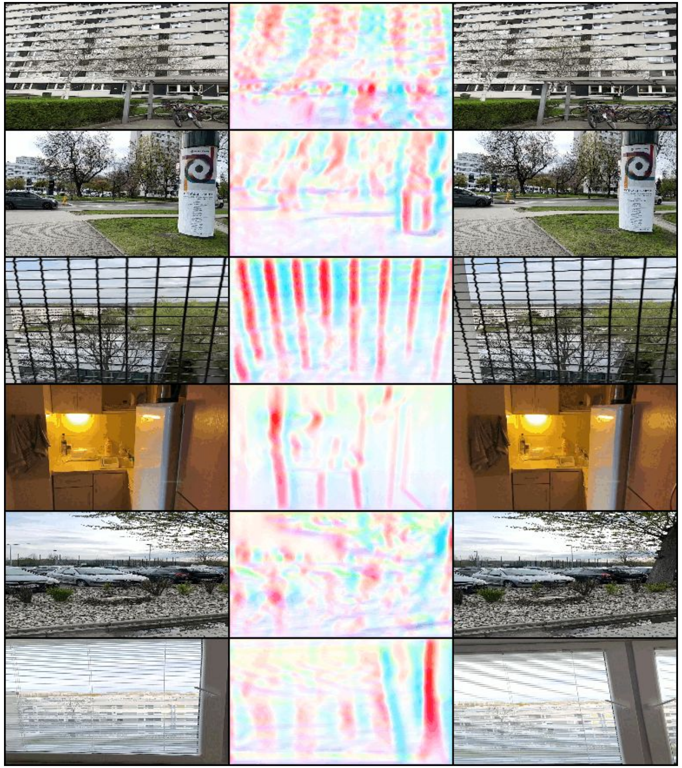
- Encoder-Decoder architecture (with U-Net type flow prediction layer between the encoder and the decoder)
- Warping both images
- ResNet flow prediction CNN
- DenseNet encoder
- Partial ResNet decoder
- Activation same as model V6_7

3rd best model (V7_1)

Loss	PSNR	MSE	MAE
0.8502	30.11	0.0013	0.0161

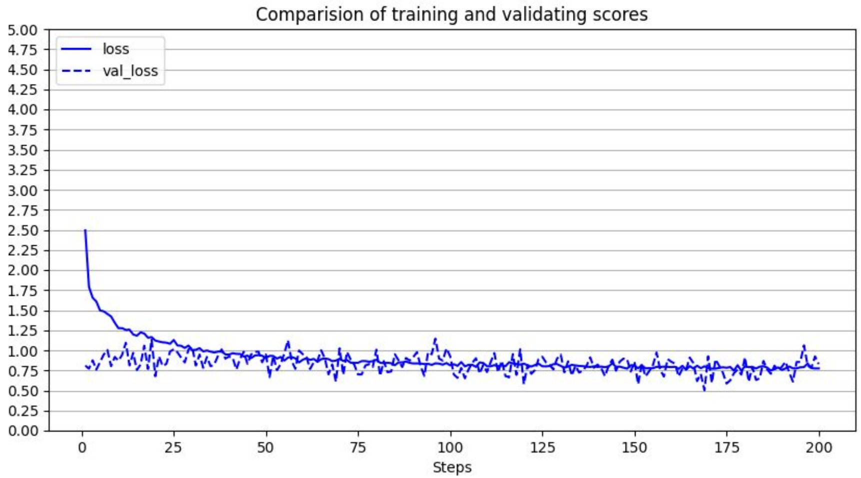


Learning progress of model V7_1

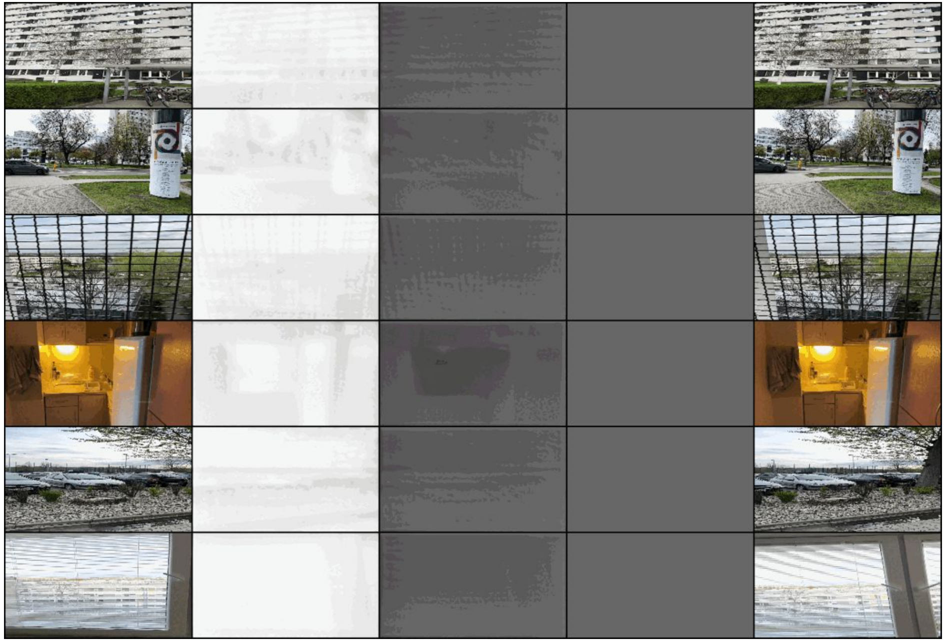


2nd best model (V6_7)

Loss	PSNR	MSE	MAE
0.7803	31.00	0.0010	0.0143

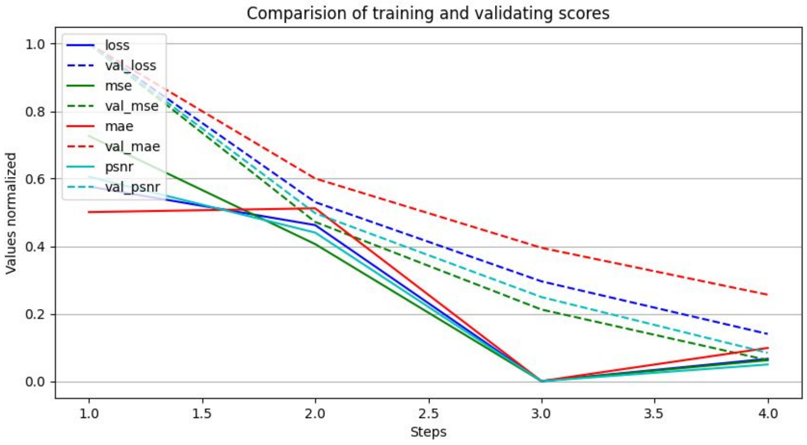


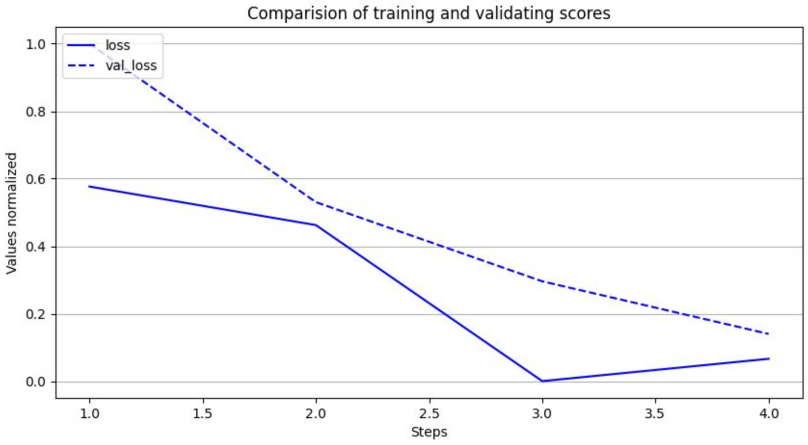
Learning progress of model V6_7



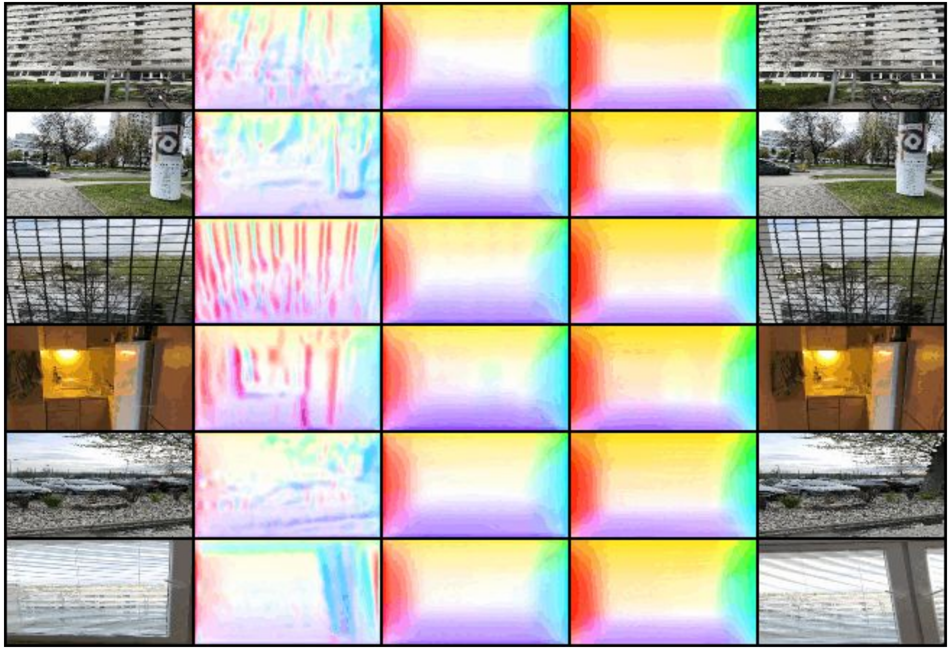
Best model (V5)

Loss	PSNR	MSE	MAE
0.7342	32.33	0.0007	0.0123

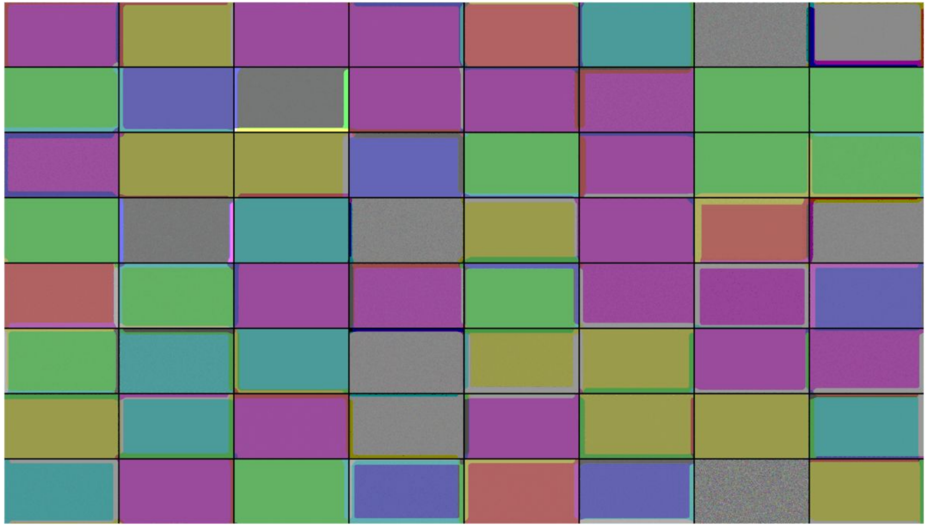


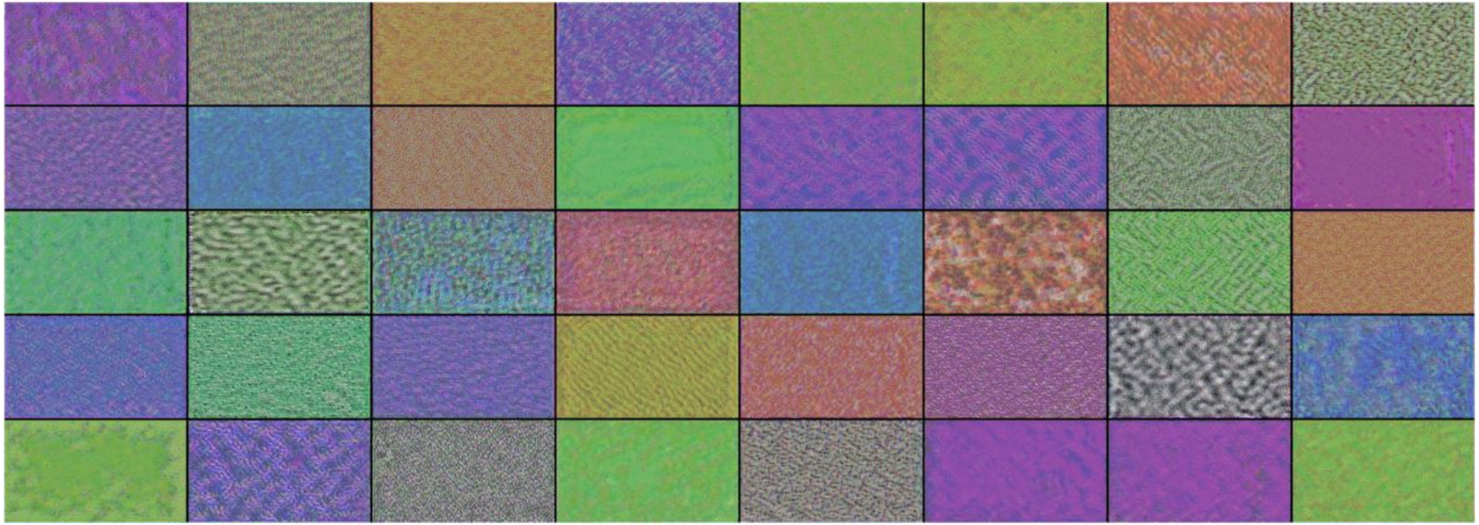


Learning progress of model V5



CNN filters of model V5





Results

GT

V5

V6_7

V7_1

Results

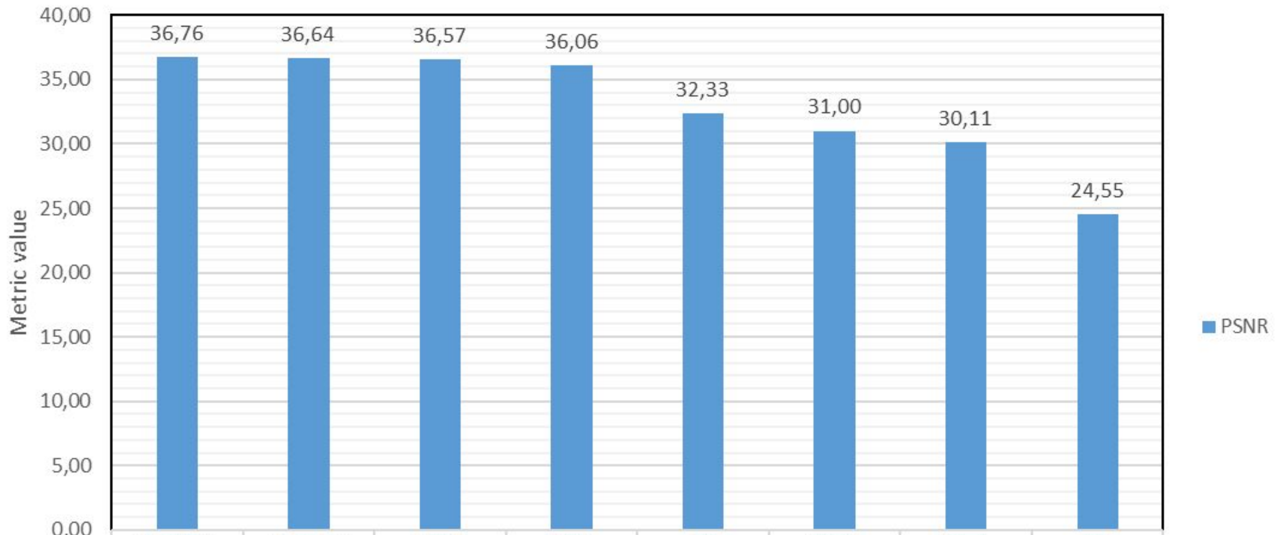
GT

V5

V6_7

V7_1

Comparison to SOTA models

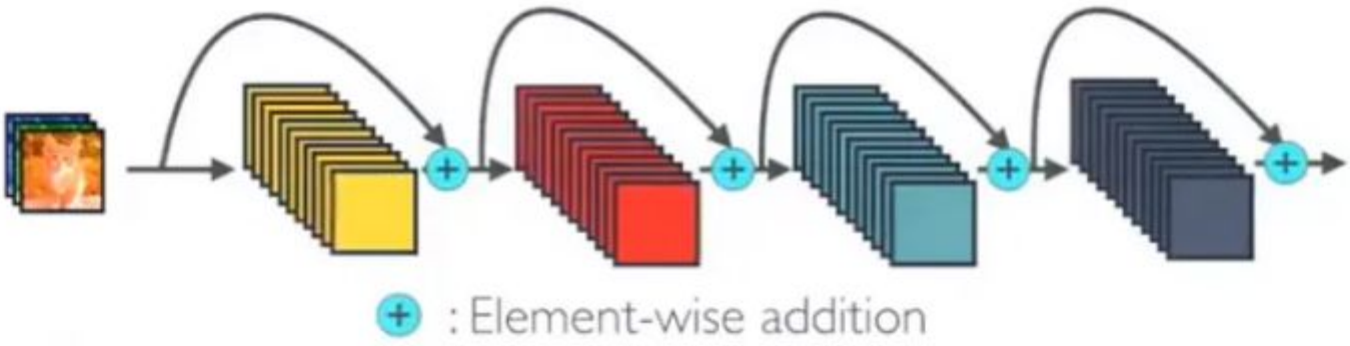


	MA-CSPA	EMA-VFI	DQBC	FILM	V5	V6_7	V7_1	Mix
PSNR	36,76	36,64	36,57	36,06	32,33	31,00	30,11	24,55

Tests

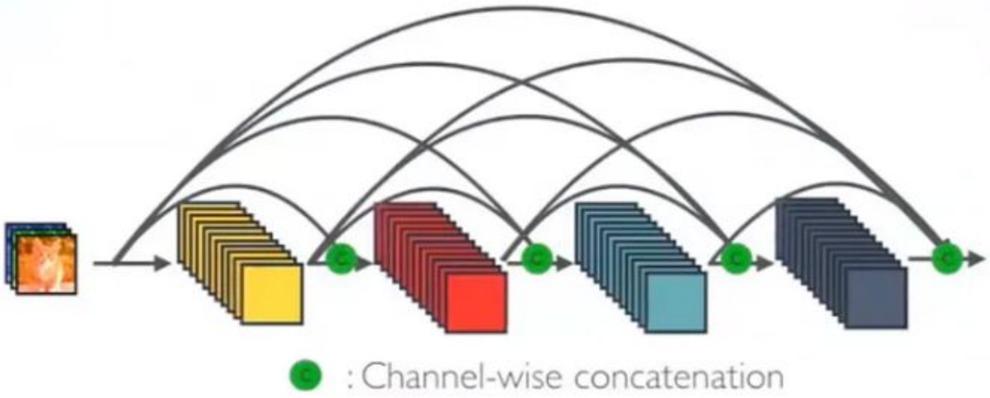
- Batch Normalization - dramatically slowed down the training process and decreased the model's performance
- AvgPool2d vs Conv2d (with stride=2) - no significant difference was noticed
- Upsample vs ConvTranspose2d (with stride=2) - the upsampling layer performed better resulting in a higher PSNR score
- Different net architectures (Feedforward, ResNet, DenseNet, U-Net, Attention U-Net) - results presented in the previous slides
- Nadam vs Adam optimizer - the Nadam optimizer reached the goal faster and lead to the better result after the same training steps
- Dropout - did not make any difference
- LeakyReLU (with a={0.01, 0.1, 0.2, 0.3}) vs PReLU - use of the PReLU activation resulted in a lower loss value
- Sigmoid vs Clipped Linear output activation - Clipped linear activation lowered the loss value by around 0.5%

ResNet



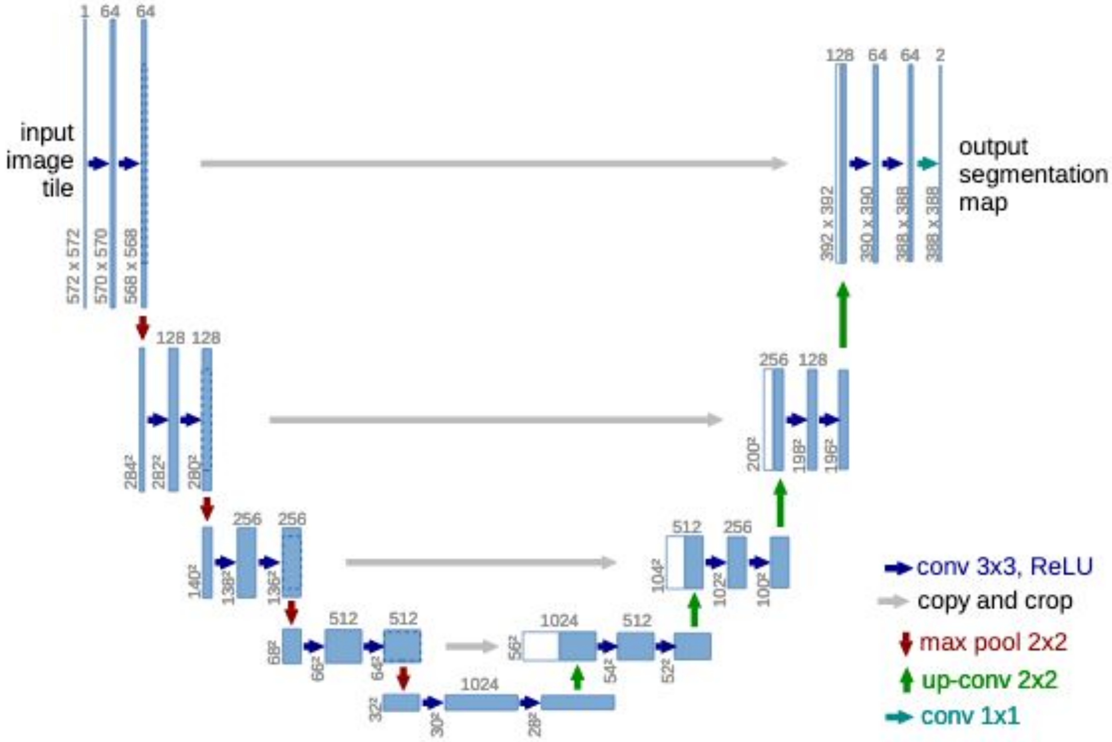
ResNet Concept

DenseNet



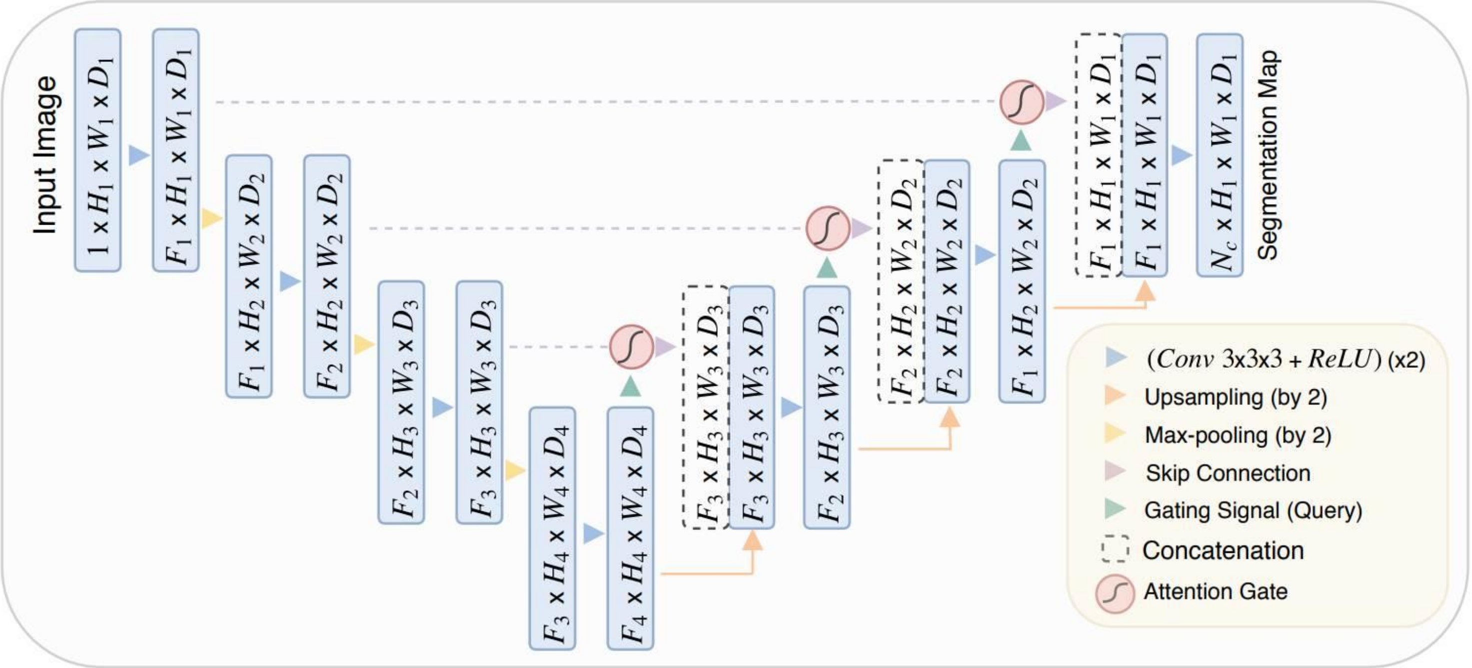
One Dense Block in DenseNet

U-Net



1
64
64
input
image
tile
572 x 572
570 x 570
568 x 568
128
128
284²
282²
280²
256
256
140²
138²
136²
512
512
68²
66²
64²
1024
32²
30²
28²
1024
512
56²
54²
52²
512
256
104²
102²
100²
256
128
200²
198²
196²
128
64
64
2
output
segmentation
map
392 x 392
390 x 390
388 x 388
388 x 388
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Attention U-Net



Resources

1. Frame Booster GitHub repository
2. Single Image Super Resolution with deep convolutional neural networks
3. Real-Time Intermediate Flow Estimation for Video Frame Interpolation
4. Depth-Aware Video Frame Interpolation
5. BiFormer: Learning Bilateral Motion Estimation via Bilateral Transformer for 4K Video Frame Interpolation
6. Attention is all you need
7. Video Frame Interpolation via Adaptive Convolution
8. Large Motion Frame Interpolation
9. FILM: Frame Interpolation for Large Motion
10. Multi-view Image Fusion
11. Perceptual Losses for Real-Time Style Transfer and Super-Resolution
12. Image Style Transfer Using Convolutional Neural Networks
13. Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation
14. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume
15. Review: DenseNet — Dense Convolutional Network (Image Classification)