On Extending Image Style Transfer to Video Style Transfer

Di Wang
Carnegie Mellon University
Pittsburgh, PA 15213
diwang2@andrew.cmu.edu

Hiroshi Wu
Carnegie Mellon University
Pittsburgh, PA 15213
bw1@andrew.cmu.edu

Alvin Pan
Carnegie Mellon University
Pittsburgh, PA 15213
qpan@andrew.cmu.edu

1 Introduction

Neural networks are a fundamental building block of many machine learning applications, due to many of their desirable properties. For example, they are shown to be more adept at feature engineering than the best human engineers. Expanding on neural networks' abilities, [1] presents a versatile algorithm to perform image-to-image style transfer, which generates a new image from a content image and a style image, preserving the "content" of the content image and the "style" of the style image.

To date, most neural style transfer literature presents innovations to image-to-image style transfer. For example, to facilitate real-time image transferring, [3] pre-trains the style image to allow one feed-forward pass per image transferred at the expense of realism. To facilitate the quality of results, [6] uses generative-adversarial networks to allow greater realism per transferred image at the expense of computational cost.

However, the topic of video-to-video style transfer from a single style image remains relatively untouched. A naive method of video-to-video style transfer is to take the sequence-of-frames video representation and perform image-to-image style transfer by taking each frame in the representation as a content image and reusing the style image. Since two neighboring frames in the video are processed entirely separately, it is possible for the resulting frames to have differing textures even in locations that are supposed to be relatively continuous from one frame to another. This results in a great amount of flickering, artifacts, and significant variation in lower-level details, which degrades the transferred video's aesthetic and greatly downgrades the quality of the resulting video, as frame-to-frame consistency, which is crucial for video aesthetics, is virtually non-existent.

We consider two facets of this problem. First, we formulate video-to-video style transfer and explore some implications of our formulation in the real-world. Second, we utilize some techniques to improve the transfer quality of videos over the baseline naive model.

2 Data

To examine the compatibility of our models, modern videos and ancient style images with different kinds of features are taken into account.

The main goal with our data is to take into consideration where and how our video-to-video transfer models will break. Hence, we tend to select a wide variety of genres of both style images and content images, with a particular emphasis on art in our style images and realism or anime in our content videos. The reason being, art, whether it's contemporary art with highly regular or well-formed shapes or contours like Cubism or Impressionism, or ink paintings with feature-dense delicate strokes from ancient China, contrasts nicely with the characteristics of features of contemporary art, while maintaining vastly different style characteristics from the content videos we have tried. Note that we shy away from styles that are similar between a content video and style image, as such work has

Preprint. Under review.

been explored thoroughly in [6] to great effect with cycle-consistent generative adversarial networks (GANs). In the end, we believe that a representative sample of style images to analyze our results is Van Gogh's "Starry Night", which represents the case of well-formed lines and an unclear divide between background and foreground without much feature density, Picasso's "Still-life" (dubbed "Leek, Skull, and Peaches" by the author) which offers sharp edges and well-formed shapes, while maintaining adequate feature density especially with the skull, and an ancient Chinese ink painting by anonymous, which contains a high density of features and an extremely sharp divide between the foreground and background.

Our content video dataset consists of modern videos, including animations with blocky frames and flashy scenes like Naruto and Gundam, which are contrasted by more traditional, mostly continuous videos like Star Wars, The Hobbit and realistic recordings from social channels like Youtube and Bilibili. Two extremes are selected in order to test a good variety of scene transition frequency, where the scenes in anime often have dramatic and poor frame-to-frame continuity could stress test any scheme we employ while the realistic videos contain the more common case of an unlikely scene transition amongst a long sequence of good continuity.

In addition, for our videos, we also pay attention to a wide range of features and feature density of our content videos. For example the arrow barrage in a scene of The Hobbit are tiny and dense but in the Star Wars movie the shots of the characters are taken in close distance so that significant details like facial expressions and gestures can be clearly observed. This variety is to further stress our models for robustness and validate our analysis, since bias towards videos with certain characteristics may result in a Clever Hans scenario.

3 Background

In the midway report, we compared the performance of naive model that simply apply Gatys' model to each frame of the videos and the model with adding mean-square-error loss between current frame and previous frame. For the naive model video, low coherency is clearly observable in the common case, but when we apply our MSE model to a video that violates scene alignment, severe ghosting is observed as the penalty for the coherence dominates all other optimization terms in our loss function. The common case for any two frames is to be in a scene, so in general, the MSE method with the coherency constraint performs much better than the model without the coherency constraint.

However, when two consecutive frames are not in the same scene, or they greatly differ from each other in some other manner, the naive method will outperform the one with added coherency. Here, the addition of a coherency loss to the gradient omits certain content features and causes ghosting (due to the inclusion of some features from the previous frame) of the original work in an effort to optimize over coherency.

4 Related work

The natural way to attempt the problem is to use a frame-by-frame approach based on image-to-image neural style transfer. However, from [4] and our experimental results, a naive application of [1] produces flickering and false discontinuities, which [4] attempts to solve by introducing a temporal consistency constraint based on optical flow, which encourages frame-to-frame consistency as well as discourage the formation of dis-occluded regions, or near blank regions. However, the algorithm has only been applied to videos that span only one scene.

[2] introduces a sequence of reference frames, which are a sequence of photos that correspond to frames in the video, but in the desired style. This method produces high-quality results, both in terms of realism image quality, and the features transferred, but it is inherently niche, as the existence of a sequence of a different style does not apply to the general problem of video-to-video neural style transfer from a style image.

5 Methods

5.1 Definitions and Problem Formulation

We begin by introducing some definitions. An image $x \in [0,1]^{3\times h\times w}$ is a 3-D tensor where each $h \times w$ sub-matrix represents the RGB color channels of the image, and let I be the set of all such images ($I = [0,1]^{3\times h\times w}$). h and w are the height and width of the image. Define a video to be a continuous flow of such images, which can be defined as a function $V_{\text{cont}} : [0,T] \to [0,1]^{3\times h\times w}$ where T is the running time of the video, and $V_{\text{cont}}(t)$ is the image being displayed at a real-valued time t. Let $\mathcal{V}$ be the set of all such videos ($\mathcal{V} = [0,T] \to I$).

In the real world, however, videos are only represented by discrete samples from the function V_{cont}, since we cannot record, store, or replay videos down to arbitrary temporal precision. Hence, we usually handle videos at 30 (or 60 in many cases, but we will use 30) frames per second, which means that videos only consist of 30 discrete images $x_1 \dots x_{30}$ for each second $\Delta t = 1$. We therefore redefine a video V as a 4-D tensor $V \in [0,1]^{n\times 3\times h\times w}$ where n is the number of frames and $n = 30t$. In other words, a video is a sequence of images. For the discrete case, we define $S(x)$ to be the style, and $C(x)$ to be the content of an image x in some latent space. Define $S'(V_{\text{cont}})$ to be the style of a video in the same latent space as S, and $C'(V_{\text{cont}})$ to be the content of a video V_{cont} in some latent space.

However, one must keep in mind that V is only a sequence of images from V_{cont} evaluated at discrete time intervals. Hence we assume that values in V must be able to smoothly interpolate into the corresponding function V_{cont}. Visually, this means that consecutive frames of a video must be smooth with respect to motion, brightness, and more. Further, we assume that the function V_{cont} is itself mostly continuous. This assumption is valid, since in a typical scene in a video, we don't have sudden changes in brightness, color, motion, etc, since objects in videos usually move continuously and don't teleport.

The reason we make this distinction between a continuous and discrete representation for videos with the above assumptions is because the subsequent methods to improve the baseline model we selected can be directly derived from these formulations of our video representation.

5.1.1 Video-to-video Style Transfer Formulations

For our formulation, we need to define a few components. Let $\mathcal{V}$ be the set of all videos, I be the set of style images, and $\mathcal{V}'$ be the set of candidate transfer videos. Let $\psi : \mathcal{V} \times \mathcal{V}' \to \mathbb{R}^+$ be the content divergence, and $\phi : I \times \mathcal{V}' \to \mathbb{R}^+$ be the style divergence.

We then want to find an output video that has similar content to the original video and similar style as the style image. Therefore we formulate the video style transfer problem as such: given a style image s, a content video $V_{\text{cont}} \in \mathcal{V}$, find

$$\underset{V'_{\text{cont}}\in\mathcal{V}'}{\operatorname{argmin}}(\psi(V_{cont}, V'_{cont}) + \phi(s, V'_{cont}))$$

Since realistically we are unable to compute everything in the continuous domain, we need to discretize the above process in our real-world implementations. However, the definition on continuous images still hold, and discretizations should approximate the continuous formulation.

5.2 Models and Methods

We now present our baseline model, and some methods to improve our baseline model by making certain assumptions about the properties of video and image functions with respect to the content video and style image.

5.2.1 Baseline Model

Our baseline model for style-transfer is a naive approach, where we simply perform image-to-image style transfer for each frame in a discrete video representation with no interdependencies between different frames of the video.

Our image-to-image style transfer is directly based on [1], which utilizes a pretrained VGG-19 convolutional neural network as a backbone to parse the various features of an image. The authors note that the activations from the first few layers correspond to low-level features, which is interpreted as style, while the activations from deeper layers correspond to high-level features, which is interpreted as content.

Let x_l denote the activations at layer l in the neural network when we pass in an image x. [1] defines C and S as

$$C(x) = x_{\text{Conv4\_2}},\ S(x) = (G(x_{\text{Conv1\_1}}), G(x_{\text{Conv2\_1}}), \ldots, G(x_{\text{Conv5\_1}}))$$

where G returns the normalized gram matrix of the features.

To perform an image style transfer from style image s to content image $V[i]$, we first calculate a feed-forward step with both s and $V[i]$ through VGG-19 to obtain $S(s)$ and $C(V[i])$. Following [1], we further define the content loss and style loss between two images as

$$L_{\text{content}}(x, y) = MSE(C(x), C(y)) \text{ and } L_{\text{style}} = \frac{1}{5}\sum_{i=1}^{5} MSE(S(x)_i, S(s)_i)$$

and finally

$$L_{\text{total}}(x, y, s) = \alpha L_{\text{content}}(x, y) + \beta L_{\text{style}}(x, s)$$

where $\alpha, \beta \in \mathbb{R}$ are hyperparameters. MSE returns the mean squared error of its two arguments, where the two arguments are n-D tensors of the same shape. To complete the transfer, our objective is

$$\underset{x}{\text{argmin}}\ L_{\text{total}}(x, y, s)$$

This loss is minimized through a gradient descent procedure, specifically the L-BFGS optimization algorithm, as [1] suggests. We then repeat this same procedure for each frame of the input video to obtain each frame of the transferred video. Going forward, this is referred to as the naive model.

5.2.2 Coherence Metrics

The results obtained in the midway report shows that video style transfer using this naive algorithm has clear issues. Although each individual frame looks good, the variance between frames is large, the artifacts in the transferred representation are abundant, and the overall aesthetic is not pleasing. At a fundamental level, this is because this discretization failed to approximate the continuous formulation of video style transfer. This model completely disregards the assumption that the resulting transferred video, V', must smoothly interpolate to our desired output from the formulation, V'_{cont}. As a result, the pixel brightness, location of objects, and other properties of frames in V' do not show continuity across time.

Therefore, our next step is to enforce this in our training process, which we do by imposing a coherence loss in our loss function. We do this by redefining L_{total} to be

$$\begin{aligned} L_{\text{total}}(V'[i], V[i], s) = \alpha L_{\text{content}}(V'[i], V[i]) + \beta L_{\text{style}}(V'[i], s) \\ + \eta L_{\text{coherence}}(V'[i], V'[i-1], V[i], V[i-1]) \end{aligned}$$

Where $L_{\text{coherence}}$ is the coherence loss based on the current frame in question $V'[i]$, previous frame transferred $V'[i-1]$, current frame of content video $V[i]$, and the previous frame of content video $V[i-1]$. η is a hyperparameter representing the weight of the coherence metric.

In the midway report, we have explored one simple option for $L_{\text{coherence}}$, which is to simply take the MSE of two consecutive frames in the transferred video:

$$L_{\text{coherence}}(V'[i], V'[i-1], V[i], V[i-1]) = MSE(V'[i], V'[i-1])$$

This loss increases as the divergence between the two consecutive frame increases, so it is a decent primitive approach to continuity. From the results of the midway report, this eliminated flickering and discontinuities very well. However, as the results showed, this produced significant ghosting effects where contents from the previous few frames can be seen in the new frames. This is due to overimposing coherence, as MSE encourages two frames to be as similar as possible, disregarding motion in the frames.

5.2.3 Enhanced Coherence Model

Here we will describe two methods to approximate frame-to-frame coherence that attempts to fix the aforementioned problem.

We can enforce coherence metrics based on the assumption that the transferred video is directly correlated to the content video, so the continuity of a transferred discrete representation can be well-approximated by the continuity of the input representation. Under such an assumption, it is sensible to enforce coherence between frames by including a term in the coherence loss that will encourage consecutive output frames to mirror the coherence of consecutive input frames. This can be done by including $MSE(V'[i] - V'[i-1], V[i] - V[i-1])$.

We also continue the use of a basic MSE between consecutive output frames, but only enforce it when the divergence between the corresponding frames in the content video is low. Let $f(x)$ be any monotone, continuous function such that $\forall x \in [0,1]$, $f(x) \in [0,1]$, $f(0)$ is close to 1, and $f(1)$ is close to 0. Then we simply scale the original MSE by $f(MSE(V[i], V[i-1]))$. Combining the two, our new coherence loss is

$$\begin{aligned} L_{\text{coherence}} = f(MSE(V[i], V[i-1]))MSE(V'[i], V'[i-1]) \\ + MSE(V'[i] - V'[i-1], V[i] - V[i-1]) \end{aligned}$$

In practice we used $f(x) = \tanh(e^{-x})$ as our coherence penalty function. e^{-x} provides heavy penalization when $|V[i] - V[i-1]|$ is large, and acts like a discriminator that distinguishes between incoherent and coherent input frames. We then add an additional control of $\tanh$, which prevents the divergence of the coherence loss, while maintaining the penalty. We refer to this model as the enhanced coherence model.

5.2.4 Optical Flow

Another way to correlate the content video's coherence with the output video is by detecting motion in the content video, and attempting to replicate it in the output video. We assume that each pixel in a frame of the content image represents part of some object, and that the pixel moves and corresponds to another nearby pixel in the next frame, dictated by the motion of that object, independent of input video continuity. This motion of pixels can be calculated using a technique called optical flow, and then imposed on the output frames. This technique is used by [4], and their results show good coherence on their dataset.

Let $V[i]_{jk}$ denote the pixel at jth row, kth column, of the ith frame of original content video. Assuming that this pixel is present in the next frame at a nearby location, optical flow calculates a first-order approximation of the motion Δx_{jk} and Δy_{jk} for each pixel in frame $V[i-1]$. By our assumption about the correlation of coherence in the content video vs the output video, this quantity should be the same for frame $V'[i-1]$. Using this, we construct a prediction P of the new frame $V'[i]$ as follows: for each pixel $V'[i-1]_{jk}$, $P_{(j+\Delta x_{jk})(k+\Delta y_{jk})} = V'[i-1]_{jk}$. It is most likely the case that some of these values are fractional, and some of the pixels in P is unfilled. In both of these cases, we fill the appropriate integer-indexed pixels using linear interpolation of the surroundings. Then $V'[i]$ should be similar to P, so we define the coherence loss

$$L_{\text{coherence}} = MSE(V'[i], P)$$

We refer to this model as the optical flow model.

6 Results

We display some representative results for the naive, enhanced coherence, and optical flow models. We display some key frames from our output videos, as well as the residual image between certain frames to analyze the coherence. We examine each model under different circumstances, where the assumptions are satisfied and dissatisfied for each method.

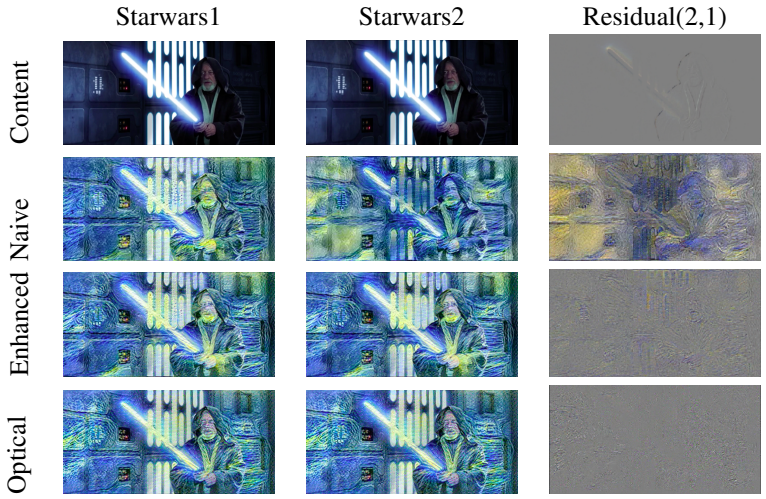

Fig 1. Consecutive frames of style transfer on a Star Wars fight scene to the style of Van Gogh's Starry Night

As we see from the residual of the content frames, the original video hardly moves. However, the naive model's output contain severe flickering and discontinuities across a two frame transition. The face of Obi-Wan turns from green to blue, the background to the left turns from blue to yellow, and the light saber appears broken. From the residual image we can also see that there is in general a great difference between the two frames. In contrast, our enhanced coherence model and optical flow model both demonstrate smooth coherence as expected.

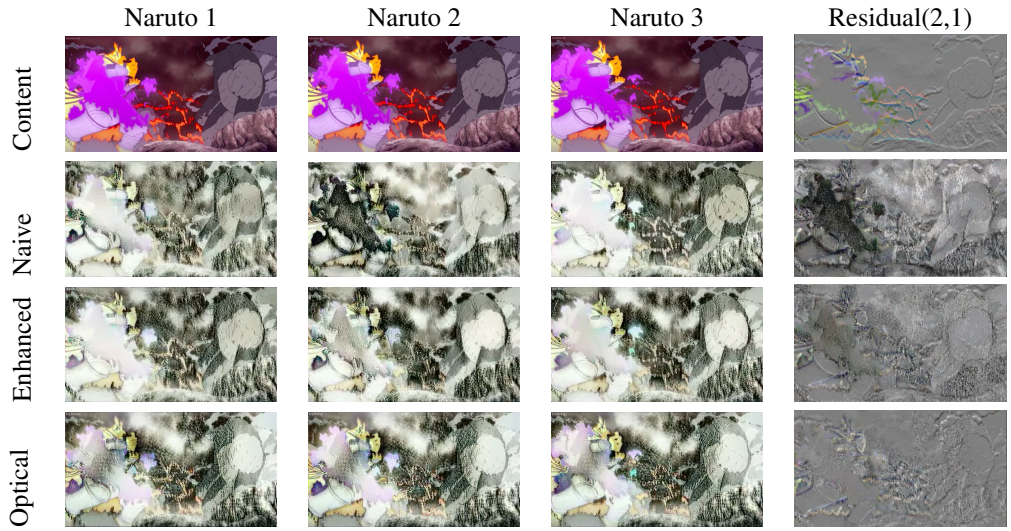

Fig 2. Nearby frames of style transfer on a Naruto fight scene to the style of Chinese traditional painting

For the Naruto scene, we notice that the naive case contains severe color discontinuities. Focusing on the originally purple flames to the left side to the scene, The naive model has a sudden transition to black and then back to white. The enhanced model has a similar problem, but not as much. The best model in this case seems to be the optical flow model. Not only are the colors consistent, but if we look at the residual image, it is rather similar to the original content residual. This indicates that the motion in the original content was successfully captured by the optical flow model to reproduce it with adequate coherence.

In contrast, the optical flow method, independent of the transferred residual, contains less discontinuities across frames that make viewing the coherent case and naive case so unpleasant.

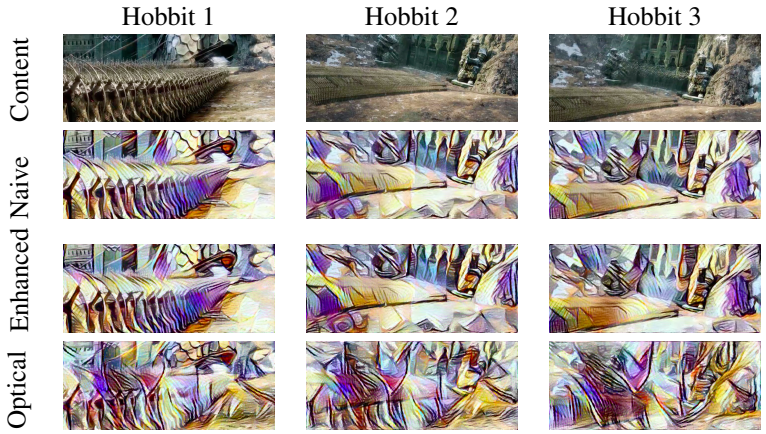

Fig 3. Frames 1 second apart of style transfer on The Hobbit Five-Army Battle scene to the style of Picasso's Still Life *series*

Finally, we examine these scenes in the grand battle scene of The Hobbit. Between frames 1 and 2, there is a scene transition. From frame 2 and on, the scene includes a very rapid background motion. This was much higher than what optical flow could analyze, which led to severe ghosting. However, we note that the ghosting did not appear in the naive and enhanced coherence models, and the content is well-reflected in the output frames. Reasons for such results will be analyzed in the next section.

7 Discussion and Analysis

In the baseline case, we make the assumption that the approximation of the original video stream can be acceptably recovered from naive image-to-image transfer. This method works just as well as any other method if the optimal output video discourages coherence, perhaps through drastic scene changes. This is the case with The Hobbit between frames 1 and 2. For such a scenario, the optimizations performed by subsequent methods will perform no better than the baseline, since adding coherence metrics goes against the original video. Since we did not implement optimizations for drastic changes, our models will in many cases will do worse than the baseline, incoherent model. In general cases where the content video is sufficiently coherent, this baseline does not work nearly as well as other coherence-based models. This can be seen easily from the Star Wars case.

In the enhanced coherence metric case, we make the assumption that the the optimal transferred video stream can be well-approximated by the input video stream in terms of frame-to-frame continuity. Such an assumption is based on the observation that the optimal transferred video often maintains strong semblance to the original content video, and so enforcing frame-to-frame continuity with respect to the input content video often produces acceptable results. Returning to our results, this corresponds to the fact that a matching residual image between the content frames and the output frames is often a good thing, as we see with the Naruto case. In addition, this allows the model to work well on the scene transition between Hobbit 1 and Hobbit 2, since even though there is a frame transition that completely threw off other MSE-based models, our scale factor recognized that there was a drastic difference from the content video, and therefore scaled down the MSE to not produce ghosting. In practice, such an assumption has the potential flaw if the optimal transferred video does not bear strong semblance to the content video, perhaps due to drastic style differences.

In the optical flow case, we make the assumption that the continuity of the optimal transferred video can be approximated by a first-order gradient of the motion in the content video. This allows the model to work well at handling the coherence when there is some motion, enforcing MSE continuity without producing ghosting. This can be seen from the Naruto example. However, optical flow is a fragile process, and whenever the motion estimate is inaccurate, it will produce significant artifacts in the transferred video. In The Hobbit's scenes, optical flow was unable to detect the scene change and unable to track the motion of a fast-moving background. In these cases optical flow produced inaccurate predictions of the next frame, leading to severe problems when we took MSE. [4]'s dataset also mainly consisted of videos where motion was very smooth, with no scene transitions, hence producing good results. One improvement we can make is to use DeepFlow by [5], which Ruder et.

al. uses in their optical flow model. This can enhance the robustness of motion estimation, leading to better results than what we currently have.

8 Conclusion

All in all, there are strengths and weaknesses to each model that we have analyzed. The fundamental source of the coherence problem comes from the fact that V is constrained to be only a discrete sample from V_{cont}. In addition, the huge variance in the image-to-image style transfer process, an inherent flaw that can be traced back to the fragility of neural networks, also contributes to the problem. If similar, related images could reliably transfer to similar images, then the process will handle the coherence by itself. Hence, a more robust image-to-image style transfer model could improve the artistic value of style transferred videos. Within our models, the most robust method should be the enhanced coherence model. This is due to the robustness of the assumptions it makes. The simple MSE model from the midway report makes an assumption that output frames must be very similar, which is not usually the case. The optical flow model assumes that motion of objects are contained within a region. This assumption breaks when there are scene transitions or just in general big movements in the content video, which is not uncommon. In contrast, the only assumption the enhanced coherence model makes is that the coherence of the output video is correlated strongly to the coherence of the input video, which is a fair assumption because the input video typically contains a reasonable continuity reference. One can therefore conclude that in the general case, the enhanced coherence model will work better due to the robustness of its assumption with respect to the average use case, while under specific constraints, like the absence of reasonable continuity in the content video, other models such as the optical flow model may work better.

9 Teammates and work division

Di proposed all the definitions, formulations, abstractions, assumptions, theoretical basis for the coherence models, an analysis framework for experimentation, along with most of the analysis for the project results. In addition, Di wrote the vast majority of the report.

Hiroshi wrote nearly all of the code, proposed the MSE and optical flow models, wrote some and heavily edited most of the methods section to align our models and theory with our code. Hiroshi also presented a significant portion of the results, wrote the analysis on optical flow and edited a significant portion of the analysis.

Alvin prepared all the datasets, modified the appropriate algorithms (came up with the enhanced coherence model [Theoretical basis by Di]), did small portion of the report, ran nearly all of the experiments, presented most of the results, selected all representative scenarios and examples and fine-tuned the model hyperparameters.

References

[1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[2] Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Stylizing video by example. *ACM Transactions on Graphics*, 38(4), 2019.

[3] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[4] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. *CoRR*, abs/1604.08610, 2016.

[5] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE international conference on computer vision*, pages 1385–1392, 2013.

[6] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.