symmetry

MDPI

Article

Facial Feature Model for a Portrait Video Stylization

Dongxue Liang, Kyoungju Park * and Przemyslaw Krompiec

Department of Software, Chung-Ang University, Seoul 06974, Korea; liang.laurel@hotmail.com (D.L.); przemek.krompiec@hotmail.com (P.K.)

* Correspondence: kjpark@cau.ac.kr; Tel.: +82-2-820-5823

Received: 15 August 2018; Accepted: 21 September 2018; Published: 28 September 2018

Abstract: With the advent of the deep learning method, portrait video stylization has become more popular. In this paper, we present a robust method for automatically stylizing portrait videos that contain small human faces. By extending the Mask Regions with Convolutional Neural Network features (R-CNN) with a CNN branch which detects the contour landmarks of the face, we divided the input frame into three regions: the region of facial features, the region of the inner face surrounded by 36 face contour landmarks, and the region of the outer face. Besides keeping the facial features region as it is, we used two different stroke models to render the other two regions. During the non-photorealistic rendering (NPR) of the animation video, we combined the deformable strokes and optical flow estimation between adjacent frames to follow the underlying motion coherently. The experimental results demonstrated that our method could not only effectively reserve the small and distinct facial features, but also follow the underlying motion coherently.

Keywords: facial feature model; portrait video; non-photorealistic rendering; Mask Regions with Convolutional Neural Network features (R-CNN)

1. Introduction

Portrait painting plays an important role in artwork. Since the advent of non-photorealistic rendering (NPR), more and more techniques for image stylization have been proposed [1–3]. Furthermore, many researchers have focused on extending the techniques from generic images to portraits [4–6]. Nevertheless, for portrait video stylization, there still exist some challenges to achieving a consistent and temporally coherent animation from the video. One of the most important things for portrait painting is to keep the rendering of distinct facial features while following the consistent animation, especially for videos that contain small faces such as interview programs, news, and lectures, etc. An important task for this kind of video is to keep the exact emotional expression of the person.

To generate a temporally coherent animation from a video, many kinds of video stylization techniques [1,7–13] have been implemented since the 1990s. In the process of achieving this goal, it is very important to avoid the effect of the source image's illumination and noise. Recently, researchers have paid more and more attention to two main kinds of techniques: the point-based rendering method [14–20], which works on a per-pixel basis to obtain stylized video; the stroke-based rendering method [7–10], which uses stroke as a basic rendering unit to achieve the rendering results. However, the point-based methods have no stroke models and the stroke-based methods lack the deformation of strokes, so their animation results tend to be insensitive to the motion of the object, which is important for following the underlying motion coherently.

As the deep learning method has become more popular, more techniques [6,21] based on deep learning have been proposed for portrait images or video stylization. Gatys, L.A. [6] presented a method that could transfer the painting texture from the source image with the Visual Geometry

Group (VGG) [22] convolution neural network [23]. This method renders the painting with a mixture of two parts: the first maintains the identity of the source image and then transfers the painting texture; the second leads to the poor capture of the painting texture, and such deformations are the problem in portrait rendering for the face. Zili Yi [24] also presented a method that could generate the stylization for face images based on a single exemplar. However, the limitation of deep learning methods is that the results might lose some of the detailed textures and thin edges, which is more serious for small faces in some forms of video such as interview programs.

Therefore, it is necessary to develop a method that can automatically generate a coherent stylized portrait video, keep the distinct facial features, and follow the underlying motion coherently at the same time. To address this, we present a method that combines the NPR with stroke deformation and facial feature model extraction. We first defined and generated a facial feature model, which consisted of the 36 contour landmarks and the facial feature model (i.e., nose, mouth, eye) of the face. By adding a feature point detecting convolutional neural network (CNN) at the end of the Mask R-CNN [25], we could obtain the coordinates of the 36 contour landmarks and a mask image representing the facial feature model. Then, we divided the input frame into three regions based on these outputs: the region of the inner face surrounded by 36 contour landmarks, the region of the facial features; and the region of the outer face. Next, we started to render the image by placing the strokes or using the original pixel color for different regions and modeling the strokes using the mass-spring model. Then, we deformed the strokes on the model dynamics by using the physics-based algorithm which applies forces on the pixels inside the strokes. After the deformations, we deleted and added strokes and pixels for the appearing and disappearing objects, then proceeded to the next frame until the video finished. The experiment results showed that for portrait video stylization, our method could automatically generate a coherent stylized portrait video, keep the distinct facial features, and follow the underlying motion coherently.

The main contributions of our work include the following:

1. The implementation of an extended deep learning method to detect the face contour landmarks and facial feature mask at the same time and treat them as a model instead of single facial landmarks.
2. The proposal of a new stylization method for portrait videos to keep small faces recognizable.
3. A novel approach was presented that combined the facial feature model using the extended Mask R-CNN and deformable strokes for video NPR.

The rest of the paper is organized as follows. We review some related work in Section 2, and present our method in Section 3. Section 4 shows our experiment results, and Section 5 presents our conclusions and scope for future work.

2. Related Work

A wide variety of methods for the NPR of portraits have been proposed recently, some of which include algorithmic methods [26,27]. This kind of technique uses information such as Difference of Gaussian (DOG), tangent flow, and edge detection extracted from the original image for the stylization. With the development of deep learning, some works have started to use CNNs to achieve this goal, such as in [28] where they used the convolution neural network to extract the exemplar image's feature, then transferred the source image to the example image's style. However, neither the algorithmic methods nor the deep learning method lost the information from the small edges nor failed to keep the feature details when it came to small faces. To keep the features of the small faces, we chose to use the deep learning method for the detection of face features.

The task of locating an accurate set of feature landmarks on the faces in the NPR of portrait videos is very important; most of the previous methods accomplish this by giving a set of feature points [29,30] or a few contours [31,32] from the source image. However, for facial landmarks, it is unclear as to how many feature points are needed when describing, for example, the eye corner, especially for small

facial features in a video, and with the contour-based method, some important parts such as teeth are hard to model with contours.

To generate an animation from a portrait video, many stroke-based methods have been created. The method in [9] was an early development for video painting, where they conducted the video animation by translating and rotating the strokes within its limitations; the method in [15] treated the strokes as points and used the optical flow for animation, and Krompiec, P. [33] presented a good method that could animate the strokes along with the underlying objects; however, for rendering small faces in videos, just keeping different sizes of the strokes to maintain the distinct features of the face is not enough.

Furthermore, our method can automatically generate a coherent stylized portrait video that not only keeps the distinct facial features, but also follows the underlying motion coherently.

3. Methods

Figure 1 shows an overview of the proposed method. The method uses the image sequences as input, and outputs the rendered version with the region-based rendering strategy. Steps of the proposed approach are as follows.

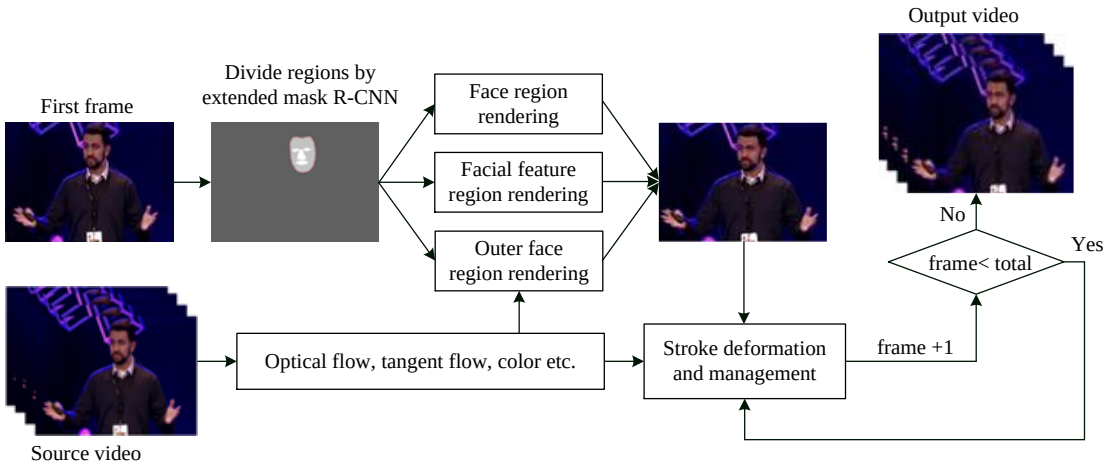


Figure 1. Overview of the proposed method.

First, according to our objective, we generated the facial feature model from the source video sequences using the extended Mask R-CNN, and based on this model, we separated each frame into two categories: the facial feature model area and non-face area. Furthermore, the facial feature model area was also subdivided into two regions: the region surrounded by 36 contour landmarks of the face and the brow-eye-nose-mouth area. The results are shown in Figure 1, where three different regions were obtained: subregion 1 (the face region surrounded by 36 contour landmarks), subregion 2 (the facial feature region of the brow, eye, nose and mouth) and subregion 3 (the outer face region).

Second, based on the extracted saliency map and tangent flows from the sequences, the render started with the first frame of the source video through different rendering methods on the sub-regions. For subregion 3, following the method in [33], strokes were placed and oriented based on the saliency map and tangent flows, before we performed the texture mapping on them. However, for subregion 1, the difference from [33] is that we located the smallest modeled strokes based only on the tangent flow. In subregion 2, we colored the facial feature area in the rendered image with the same pixel colors of the areas from the source image.

Finally, after rendering the first frame, we simulated the deformed strokes due to forces which were calculated by the optical flows between frames, and during the simulation, we compared the results with the reference frame to delete and add strokes until all video sequences were processed.

3.1. Extended Mask R-CNN for the Facial Feature Model

As shown in Figure 2, the facial feature model referred to the combination of subregion 1 and subregion 2. As one of the most effective instance segmentation methods, Mask R-CNN [25] is able to give the pixel-level segmentation for the input. Thus, it can be used to generate the mask representation of subregion 2. For the 36 contour landmarks of subregion 1, we designed a CNN regressor based on the protogenic Mask R-CNN to obtain their coordinates in a more precise way.

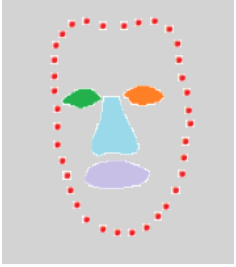


Figure 2. Facial feature model.

Mask R-CNN was proposed by He, K.-M. [25] for image segmentation and yields good performance. Given an image as input, it can efficiently detect objects in an image while simultaneously generating a high-quality segmentation mask for each instance. Furthermore, the Mask R-CNN itself has a CNN framework to automatically extract feature maps for classification or regression; it is easy to add a CNN branch to regress more precisely the points set on the boundary of the whole face. Consequently, it is of importance to design the repressors with the feature maps extracted by the Mask R-CNN and train the network to obtain a sufficiently good model. With the extended Mask R-CNN, the edge of the facial features (brow, eye, nose, mouth) can become more smoothly fitted.

The architecture of the whole process of generating the facial feature model from the extended Mask R-CNN is shown in Figures 3 and 4. As has been reported, Mask R-CNN uses the Resnet as the backbone to extract the feature map and an FCN (Fully Convolutional Network) for mask prediction. A ROIAlign layer has also been proposed to properly align the extracted features with the input. In our extended Mask R-CNN, we kept the above modules to obtain the mask of the face model and added another CNN branch to regress the 36 facial landmarks. A detailed description of the Mask R-CNN can be seen in [25]. Figure 4 shows the detailed architecture of the extended CNN branch. Specifically, the CNN took the aligned feature map with a size of 48 × 48 from ROIAlign and output the coordinates of a total of 36 points (i.e., the output dimension is 72). Moreover, the CNN was composed of four convolution layers and two fully connected layers. Each convolution layer was followed by Relu activation and max Pooling for sampling. Additionally, the kernel size of the convolution layer was 3 × 3 except for the last one of 2 × 2.

During training, the loss function in the extended Mask R-CNN is as follows:

$$L = L_{cls} + L_{box} + L_{mask} + L_{landmarks} \tag{1}$$

where the L_{cls}, L_{box}, and L_{mask} are the same as the Mask R-CNN, and the $L_{landmark}$ is the ordinary least square loss function,

$$L_{landmark} = \sum_{i=1}^{n} (\hat{y}_i^{landmark} - y_i^{landmark})^2 \tag{2}$$

where n = 36; $\hat{y}_i^{landmark}$ is the value of the i^{th} andmark predicted by this CNN; $y_i^{landmark}$ is the value of the i^{th} true landmark. Therefore, we minimized the $L_{landmark}$ to finally obtain our output which had 36 face contour landmarks, here y belongs to 72 tuples.

The experiments for the training step are shown in Section 4.

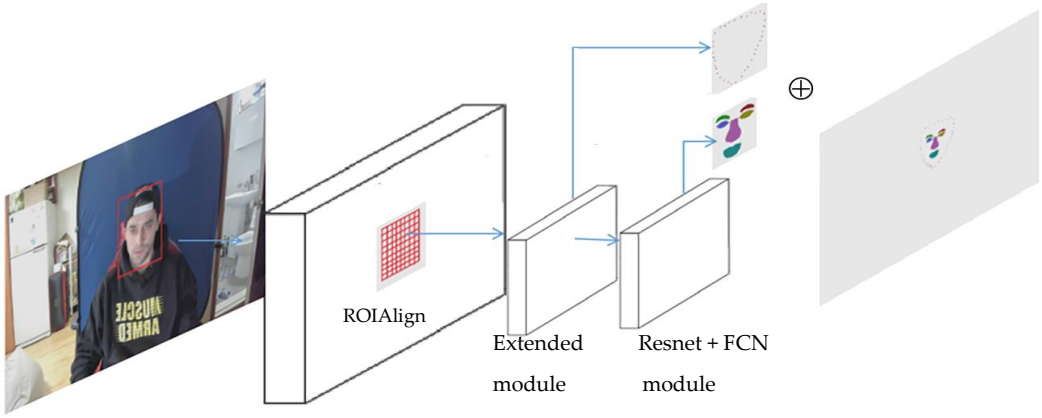


Figure 3. The architecture of the extended Mask R-CNN for facial feature model generation.

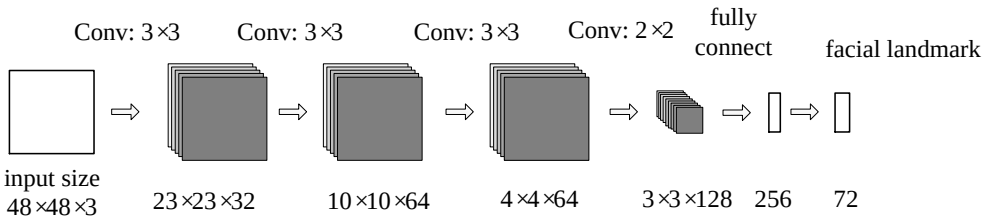


Figure 4. Details of our additional CNN regression branch.

3.2. First Frame NPR

The stroke-based rendering method was applied in our method to generate the non-photorealistic results. The strokes were initialized at the beginning of the video and were then deformed according to the relative motion between the adjacent frames. Thus, it is of much significance to properly design and initialize the strokes in the first frame. Following [33], we used the stroke model shown in Figure 5a for the non-facial area to follow the underlying motion coherently during the video sequences and keep the least number of strokes. This model is composed of six particles and nine springs for the deformation and motion between frames. Furthermore, as our aim was for small faces in our videos, we adopted another more suitable stroke model shown in Figure 5b to keep the distinct features in the inner face area. To keep the details of the facial features in subregion 3, we blended the feature image with the rendered image based on the mask for subregion 3. In general, the region-based rendering procedure can be divided into two steps: the outer face area rendering and inner face area rendering.

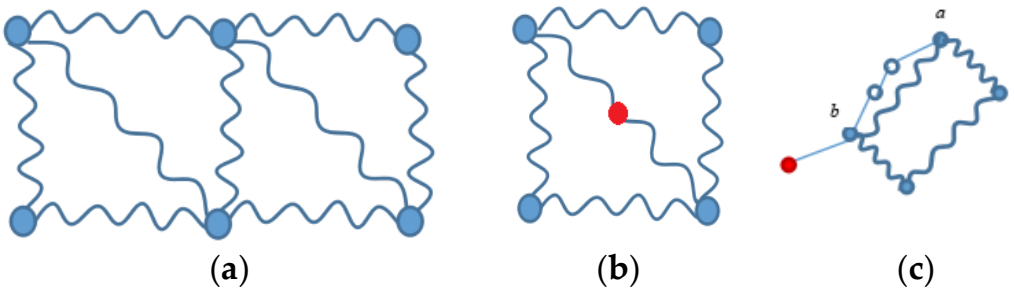


Figure 5. (**a**,**b**) are different stroke models for subregion 1 and subregion 3; (**c**) location and orientation calculation.

3.2.1. Outer Face Area Rendering

For subregion 3, we used the stroke model composed of particles and springs as shown in Figure 5a for the rendering. The render started by randomly selecting an unprocessed pixel in subregion 3, then created a stroke model that covered the pixels found by the saliency map and tangent flow, and this procedure was repeated until all the pixels were processed. Then, the brush texture was mapped to the stroke model with parameters such as position, length, thickness, orientation, and color. Based on the saliency map, and keeping the boundaries along the flow directions, the strokes were adaptively generated for different regions in this subregion.

From the above, we adopted the adaptive strokes with different models for subregion 1 and subregion 3, while at the same time keeping the distinct features of the small faces based on the facial feature mask in subregion 2. The result of the rendered experiment for first frame rendering is as shown in Figure 6. Section 4 presents a detailed comparison of the results.

(**a**) (**b**)

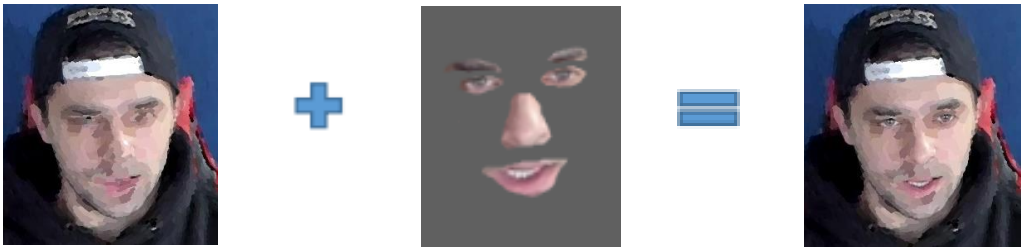
Figure 6. First frame result. (**a**) Source image; (**b**) Result of our NPR method.

3.2.2. Inner Face Area Rendering

The inner face area contained two components: subregion 1 and subregion 2. For subregion 1, we employed a simpler stroke model, shown in Figure 5b, due to two reasons. First, the small faces in the video account for fewer pixels, hence the strokes should be smaller to guarantee that the strokes are not larger than the faces. The second reason is that small enough strokes can also retain more detail in the faces. For instance, different areas, except the facial feature area, should be rendered with a different color due to the light or makeup. Larger strokes would only be mapped with a unique color texture and cannot keep such small differences in the face.

Therefore, we used four particles in the stroke model, and for subregion 1, the calculation method of the particles' location and orientation had some modifications. On one hand, instead of using the saliency map, we selected a random pixel a in subregion 1, and picked b along the tangent flow direction one by one until the flow difference of a and b was more than 30° or the number of the pixels between a and b was more than 5, and then chose b as the second particle of the model. As the face area was small, we set the thickness of the stroke model as a constant value 3, and the direction was the gradient direction. This step is shown in Figure 5c.

To render subregion 2, the facial feature mask image was employed to obtain the original color of the pixels in the reference source frame. As shown in Figure 7, by using the Alpha blending method [34], we blended the pixels based on the facial feature mask image (Figure 7b) with the rendered image (Figure 7a), and obtained the blended result (Figure 7c). The reason for this is that when previously rendering the strokes on small faces, much of the orientation information will be lost if using the tangent flow calculation method, and the strokes with a unique color will affect the emotional expression of the person, which is important for interviews or talk-show programs. Therefore, we extracted the facial feature mask image from the original image with the extended Mask R-CNN to keep the distinct facial features.



(**a**) (**b**) (**c**)

Figure 7. Rendering of subregion 2. (**a**) Rendered image with Figure 5b; (**b**) Mask image; (**c**) Blended image.

3.3. Video Sequence NPR

After rendering the first frame, we started to animate the video sequences. Unlike most video stylization methods that only focus on the translation or orientation of moving strokes, we also performed deformations on strokes based on our previous work [33]. The exact calculation of the position and deformation of each stroke in the next frame was based on the physics-based framework [35], and the motion estimation method [36]. Furthermore, during the animation of the whole video, to avoid excessive memory use when rendering a large amount of strokes where some special phenomena exist, for example, when someone disappeared or appeared in the video, some strokes needed to be deleted and added in the corresponding region. Therefore, as in [33], the video sequence NPR had two steps: (1) Calculating the position of the strokes, and (2) The deletion and addition of the strokes.

3.3.1. Calculation of the Position of the Strokes

During the animation of the frame sequences, we calculated the position of each particle in the stroke model with two kinds of forces: the external force from the reference frame, and the internal spring forces. The external forces were calculated based on the motion estimation between frames where we used an accurate method [36] to calculate the smooth optical flow, treating this as the external force. Internal spring forces were calculated based on Hooke's law. When obtaining the position of the stroke, the only step different from [33] was in choosing the material point, as we had a different stroke model for the face area model. We registered the material point as simpler with the center point of the particles as in the red point shown in Figure 5b, the reason being that the stroke model in the face area was small, so this would minimize the calculation when undertaking the animation.

Finally, the position of the stroke was calculated by

$$\ddot{x}(p) = f_t(p) + f_t(s) \tag{3}$$

where $\ddot{x}(p)$ is the second-order time derivative of $x_t(p)$, and $x_t(p)$ represents the position of the particle p in frame t, $f_t(p)$ is the external forces from the reference frame, and $f_t(s)$ denotes the spring forces. We solved this equation by using a simple Euler's method, finally obtained the position of all particles, and then used this result in register the point.

Next, the target position of the material points of the stroke was calculated based on the reference material point and the estimated optical flow.

3.3.2. Stroke Deletion and Addition

As mentioned above, we directly rendered subregion 2 by using the original frame's pixels. When undertaking the animation, the render in this region did not need to be modified. However, the render in subregion 1 and subregion 3, through the relative motion between neighbor frames, should be accounted for when the person is moving, disappearing, or appearing in the video, thus some deleting and adding operations for the corresponding strokes should be considered to avoid the memory

problem. Following the rendering steps for the first frame, we deleted the strokes based on the position and the color difference of the strokes, and added new strokes on the deleted stroke area based on the new render parameters such as color, tangent flow, position and the model styles, etc.

4. Results and Discussion

In this section, we evaluate the performance of our proposed method on the tasks of automatic portrait video painting. We first introduce the videos and datasets used in our experiments in Section 4.1. Then, the effectiveness and the performance of the extended Mask R-CNN is described in Section 4.2. Section 4.3 discusses the proposed method's limitations based on the existing facial landmark detection method and the experimental results. Finally, a series of experiments for the NPR of portrait videos and comparisons are presented in Section 4.4.

4.1. Datasets and Source Videos

We chose the open source dataset 300-VW [37–39] to prepare our own annotations and train our extended Mask R-CNN. 300-VW was developed to be a comprehensive benchmark for evaluating facial landmark tracking algorithms in the wild and it contains many long facial videos, especially videos that contain small faces. Hence, we chose 20 videos containing small faces in this dataset, and the folder index list for the videos is shown in Table 1, and five annotators in our laboratory helped to annotate the frames (100 frames per video) with an open annotation tool called LabelMe [40]. The annotations per image included six classes of facial features (brow_l, brow_r, eye_l, eye_r, nose and mouth) for the facial feature mask, and 36 landmarks for the face boundary. Based on [38], we used part of the annotations for the 36 landmarks on face contour annotation, and used the same algorithm in [38] to evaluate the annotation accuracy.

Table 1. Video index list in 300-VW.

Name	Index No.
Video Folder index	001–004, 009–011, 013, 015, 016, 018, 019, 022, 025, 028, 031, 041, 047, 053, 405

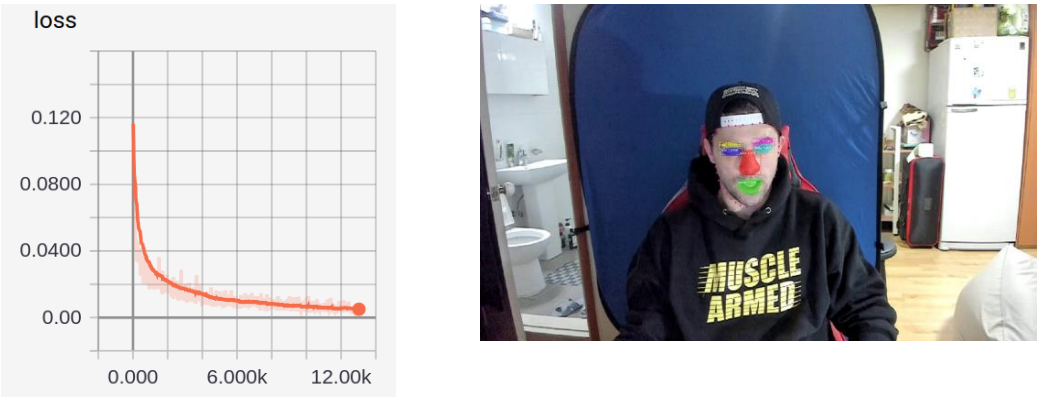
To generate the portrait video stylization result, some videos were picked from 300-VW and others were captured in real life by us to imitate interviews and talk-show programs.

4.2. Experiments and Performance for the Extended Mask R-CNN Method

With the 300-VW annotated datasets mentioned in the previous section, we divided the samples into two groups: 90% for training and the remainder for testing. During training, we used the Adam optimizer [41] for optimization with a learning rate beginning at 0.0001 and decayed by half after every 50 epochs. The batch size was set as 16. The training code was implemented in Keras and TensorFlow and the whole procedure took about 14 hours on GTX1080.

Next, we analyzed the network performance. First, we used the RMSE (root mean square error) to evaluate the error of the predicted value and real value. As shown in Figure 8a, the error of our extended Mask R-CNN was less than 0.08. Figure 8b exhibits the testing result. As the testing result showed, our extended Mask R-CNN could effectively generate the facial feature model we needed in our NPR process.

Furthermore, the minimum size of the face that can be detected by the features was about 80×80 in 1080 p images. As we added a simple CNN branch based on the Mask R-CNN, the test time for the 1080 p image could be 4 fps, as the branch took about 20 ms separately.

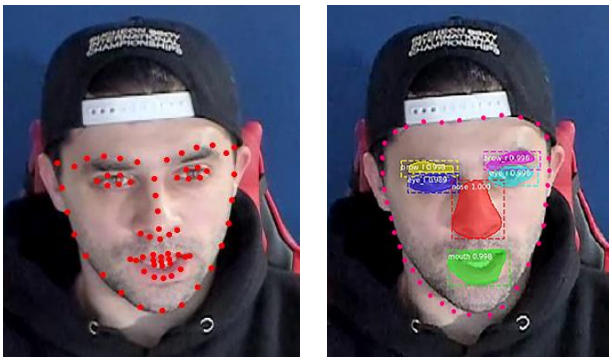


(**a**) (**b**)

Figure 8. Performance analysis. (**a**) Root Mean Square Error (RMSE) curve; (**b**) Facial Feature model result.

4.3. Comparison and Discussion of the Extended Mask R-CNN Method

For further comparison with the recent face landmark detection methods, most of the existing methods only detect a finite number of landmarks [42]. The result of detecting 68 facial landmarks is shown in Figure 9a, where we can see that the location of the landmark was very accurate, but for our goal of keeping the detailed information of small faces, just the exact location is not enough. However, when compared with our mask result (Figure 9b), the nose area could not be well defined in the result in [42], and the eye area, which could be surrounded by six landmarks, was too small to keep the details on the edge of the eye. Therefore, the extended Mask R-CNN was better than the existing face landmark detection methods. However, one disadvantage of our method is that for the side face, the performance was not more than 90% for the video taken by us when the head rotated more than 30°, so it needs to be improved in future work.



(**a**) (**b**)

Figure 9. Comparison. (**a**) Result of 68 facial landmarks detection; (**b**) Result of extended Mask R-CNN.

4.4. Portrait Video NPR Experiments and Comparisons

As mentioned above, we started the first frame rendering by using the facial feature model, tangent flow, and saliency map, and used different rendering strategies for different regions. At the same time, comparisons with the state-of-art were also conducted to demonstrate the performance of the proposed method.

Figure 10 shows a comparison of our results with those of [33]. We used a similar method for the deformation of the strokes during the animation, the difference being the rendering method. We created a facial feature model to separate the frame into three regions instead of just using the saliency map, and used a new stroke model for rendering the face region as this method can keep the small

features of a face. As can be seen in Figure 11a, the small distinct features are not expressed well in [33]. The first row in Figure 11 demonstrates the detailed comparison with [33]. The eyes were rendered by only several strokes, so it is hard to see the expression in detail; however, our results (Figure 11c) using the facial feature model kept the facial features that were clearly seen. Figure 11c,d show that during the frame motion, the method could follow the underlying object motion with stroke deformation, which in this sample, was the small deformation for whiskers.

(**a**)

(**b**)

Figure 10. Comparison of our results (**a**) with Krompiec, P. et al. (**b**).

(**a**) (**b**) (**c**) (**d**) (**e**)

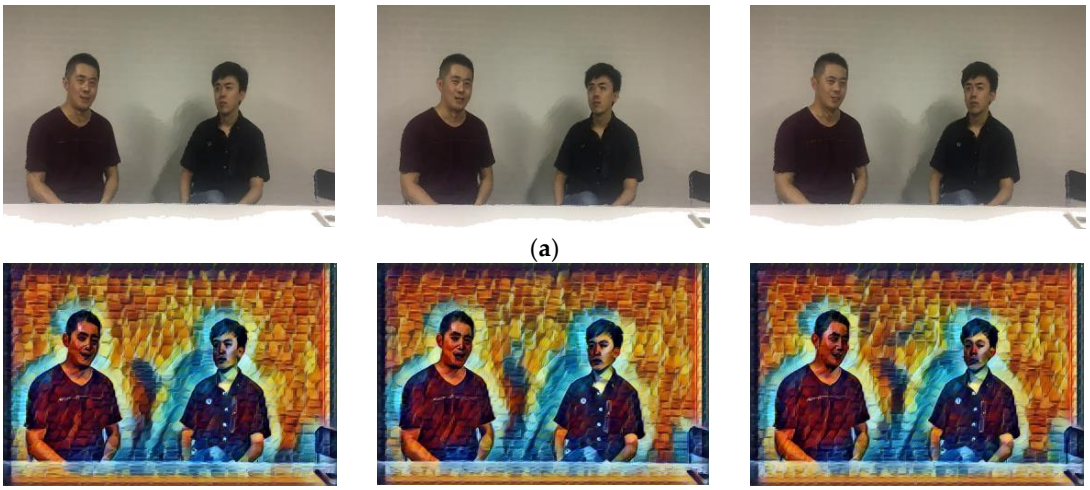
Figure 11. Rendering results. (**a**) Source image; (**b**,**c**) are the comparison for when the distinct facial features are maintained; (**d**,**e**) keep the underline motion part based on the deformation strokes.

Figure 12 shows a detailed comparison of the small stroke model used in subregion 1, where we can see that the edges of the face and the glasses were well kept in our results by using the small stroke model. However, in the previous method [33], the big stroke model did not work well.

(**a**) (**b**)

Figure 12. Small stroke details in subregion 1, the inner face region. (**a**) Result of [32] using the big stroke model; (**b**) Our results using the small stroke model.

Figure 13 shows a comparison of the video sequence results with another deep learning method [6], where Gatys, L.A. [6] transferred the source image to the example image's style. Therefore, the results differed greatly with ours, but our areas of concern were the facial feature details and the temporal rendering. As shown in Figure 14, their results did not clearly distinguish the details of the eyes and teeth and the background strokes were very different for each frame's style transfer; there were many flickers in the whole video; moreover, around the body's contour, a halo looked as if it had been added, which was very unnatural. In our method, the results not only effectively kept the facial features during the animation, but also produced fewer temporal artifacts without any unnatural strokes rendered.



(**a**)

(**b**)

Figure 13. Comparison of our result (**a**) with that of Gatys, L.A. (**b**).

(**a**) (**b**)

Figure 14. Facial feature details. A comparison of our result (**a**) with that of Gatys, L.A. (**b**).

From the above, our method was able to keep the distinct facial features and could also follow the motion of underlying objects. These good performances can be attributed to the introduction of the extended Mask R-CNN to generate a facial feature model and the deformation strokes. In addition, the strokes in our method were also adaptively deleted and added during the frames, which helped to generate a temporally coherent animation.

5. Conclusions

A robust method for stylizing portrait videos containing small faces was proposed in this paper. Compared with previous methods, what was improved in this method was that our rendering method first used the extended Mask R-CNN method to detect the face contour landmarks and facial feature areas at the same time and treat them as a model instead of single facial features. In addition, we combined this facial feature model with deformable strokes for video NPR. The experimental results demonstrated that the proposed method could keep the distinct characteristics of a small face. Future work will focus on the following aspects: First, to improve the accuracy of the side face detection by increasing the diversity of the datasets. Second, to achieve natural rendering results for the boundaries of the face and features by better separating out the different layers of the image and using more different stroke models. Another aspect, i.e., age, sex, and race recognition, is becoming more popular. The combination of age, sex and race recognition information with different styles of NPR methods represents challenging and interesting future work.

Author Contributions: D.L. and K.P. designed this algorithm. D.L. performed the experiments, analyzed the data, and wrote this paper. K.P. provided important comments and suggestions, and revised the paper. P.K. provided some suggestions and recorded the video as a sample for the experiment results.

Funding: This research was supported by the Chung-Ang University research grant in 2017 and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2016R1A2B4016239).

Acknowledgments: We would like to acknowledge the following individuals for their Creative Common license video sources: P.K. one of our authors, and Kang Li, XinTong Hao and Jinseo Jeong for the recorded video.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hertzmann, A.; Perlin, K. Painterly rendering for video and interaction. In Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering (NPAR 2000), Annecy, France, 5–7 June 2000; pp. 7–12.
2. Hertzmann, A.; Jacobs, C.E.; Oliver, N.; Curless, B.; Salesin, D.H. Image Analogies. In Proceedings of the 28th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001), Los Angeles, CA, USA, 12–17 August 2001; pp. 327–340.
3. Kyprianidis, J.E.; Collomosse, J.; Wang, T.-H.; Isenberg, T. State of the "Art": A taxonomy of artistic stylization techniques for images and video. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 866–885. [CrossRef] [PubMed]
4. Wang, X.-G.; Tang, X.-O. Face photo-sketch synthesis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 1955–1967. [CrossRef] [PubMed]
5. Zeng, K.; Zhao, M.; Xiong, C.; Zhu, S.-C. From image parsing to painterly rendering. *ACM Trans. Graph.* **2009**, *29*. [CrossRef]
6. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**; arXiv:1508.06576.
7. Meier, B.J. Painterly rendering for animation. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1996), New Orleans, LA, USA, 4–9 August 1996; pp. 477–484.
8. Litwinowicz, P. Processing images and video for an impressionist effect. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1997), Los Angeles, CA, USA, 3–8 August 1997; pp. 407–414.

9. Hays, J.; Essa, I. Images and video based painterly animation. In Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering (NPAR 2004), Annecy, France, 7–9 June 2004; pp. 113–120.
10. Huang, H.; Zhang, L.; Fu, T. Video painting via motion layer manipulation. *Comput. Graph. Forum* **2010**, *28*, 2055–2064. [CrossRef]
11. Collomosse, J.P.; Rowntree, D.; Hall, P.M. Stroke surfaces: Temporally coherent artistic animation from video. *IEEE Trans. Vis. Comput. Graph.* **2005**, *11*, 540–549. [CrossRef] [PubMed]
12. O'Donovan, P.; Hertzmann, A. AniPaint: Interactive painterly animation from video. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 475–487. [CrossRef] [PubMed]
13. Lin, L.; Zeng, K.; Wang, Y.; Xu, Y.; Zhu, S. Video stylization: Painterly rendering and optimization with content extraction. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 577–590. [CrossRef]
14. Bousseau, A.; Neyret, F.; Thollot, J.; Salesin, D. Video watercolorization using bidirectional texture advection. *ACM Trans. Graph.* **2007**, *26*. [CrossRef]
15. Hedge, S.; Gatzidis, C.; Tian, F. Painterly rendering techniques: A state-of-the-art review of current approaches. *Comput. Anim. Virtual Worlds* **2013**, *24*, 43–64.
16. Yoon, J.; Lee, I.; Kang, H. Video painting based on a stabilized time-varying flow field. *IEEE Trans. Vis. Comput. Graph.* **2011**, *18*, 58–67. [CrossRef] [PubMed]
17. Seo, S.; Ostromoukhov, V. Pointillist video stylization based on particle tracing. *Multimed. Tools Appl.* **2014**, *71*, 279–292. [CrossRef]
18. Liang, D.-X.; Park, K. Pencil drawing animation from a video. *Comput. Anim. Virtual Worlds* **2013**, *24*, 307–316. [CrossRef]
19. Kang, H.; Lee, S.; Chui, C.K. Coherent line drawing. In Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering (NPAR 2007), San Diego, CA, USA, 4–5 August 2007; pp. 43–50.
20. Achanta, R.; Hemami, S.; Estrada, F.; Susstrunk, S. Frequency-tuned salient region detection. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 1597–1604.
21. Duan, C.; Gao, Y.; Zhang, J. Artistic video stylization for face. In Proceedings of the 3rd International Conference on Systems and Informatics (ICSAI 2016), Shanghai, China, 19–21 November 2016; pp. 949–953.
22. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**; arXiv:1409.1556.
23. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
24. Yi, Z.-L.; Li, Y.; Ji, S.-Y.; Gong, M.-L. Artistic stylization of face photos based on a single exemplar. *Vis. Comput.* **2017**, *33*, 1443–1452. [CrossRef]
25. He, K.-M.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2007 IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
26. Wang, Q.-Y.; Chen, D.-S.; Li, S.-R.; Wu, Q.; Zhang, Q. An adaptive cartoon-like stylization for color video in real time. *Multimed. Tools Appl.* **2017**, *76*, 16767–16782. [CrossRef]
27. Rosin, P.L.; Lai, Y.-K. Non-photorealistic rendering of portraits. In Proceedings of the Workshop on Computational Aesthetics (CAE 2015), Istanbul, Turkey, 20–22 June 2015; pp. 159–170.
28. Selim, A.; Elgharib, M.; Doyle, L. Painting style transfer for head portraits using convolutional neural networks. *ACM Trans. Graph.* **2016**, *35*, 129. [CrossRef]
29. Zhu, X.; Ramanan, D. Face detection, pose estimation and landmark localization in the wild. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012), Providence, RI, USA, 16–21 June 2012; pp. 2879–2886.
30. Belhumeur, P.N.; Jacobs, D.W.; Kriegman, D.J.; Kumar, N. Localizing parts of faces using a consensus of exemplars. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2930–2940. [CrossRef] [PubMed]
31. Gu, L.; Kanade, T. A generative shape regularization model for robust face alignment. In *European Conference on Computer Vision*; Springer: Berlin, Germany, 2008; pp. 413–426.

32. Saragih, J.M.; Lucey, S.; Cohn, J.F. Face alignment through subspace constrained mean-shifts. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (CVPR 2009), Kyoto, Japan, 29 September–2 October 2009; pp. 1034–1041.
33. Krompiec, P.; Park, K.; Liang, D.-X.; Lee, C. Deformable strokes towards temporally coherent video painting. *Vis. Comput. Int. J. Comput. Graph.* **2016**, *32*, 813–823. [CrossRef]
34. Tomas, P.; Duff, T. Compositing digital images. *Comput. Graph.* **1984**, *18*, 253–259.
35. Baraff, D.; Witkin, A. Physically based modeling: Principles and practice. In Proceedings of the SIGGRAPH'97 Course Notes, Los Angeles, CA, USA, 3–8 August 1997; Volume 23, pp. 4142–4150.
36. Lang, M.; Wang, O.; Aydin, T.; Smolic, A.; Gross, M. Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph.* **2012**, *31*, 34. [CrossRef]
37. Chrysos, G.S.; Antonakos, E.; Zafeiriou, S.; Snape, P. Offline deformable face tracking in arbitrary videos. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW 2015), Santiago, Chile, 7–13 December 2015; pp. 1–9.
38. Shen, J.; Zafeiriou, S.; Chrysos, G.S.; Kossaifi, J.; Tzimiropoulos, G.; Pantic, M. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW 2015), Santiago, Chile, 7–13 December 2015; pp. 50–58.
39. Tzimiropoulos, G. Project-out cascaded regression with an application to face alignment. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; pp. 3659–3667.
40. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173. [CrossRef]
41. Kingma, D.P.; Ba, L.J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
42. He, Z.; Zhang, J.; Kan, M.; Shan, S.; Chen, X. Robust FEC-CNN: A high accuracy facial landmark detection system. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 2044–2050.