Ava
(Dynamic sky replacement and video harmonization)

Arfy Slowy
slowy.arfy@gmail.com

muhammad rafi
onxdl16427@outlook.com

ABSTRACT

This Paper proposes a vision-based method for video sky replacement and harmonization, which can automatically generate realistic and dramatic sky background in video with controllable style.Different from previous sky editing methods that either focus on static photos require inertial measurement units integrated in smartphones on shooting videos. Our method is purely vision-based, without any requirements on the capturing devices, and can be well applied to either online or offline processing scenarios. Our method runs in real-time and is free of user interactions, and we decompose this artistic creation process into a couple of proxy tasks including sky matting, motion estimation, and image blending. Experiments are conducted on videos diversely captured in the wild by handheld smartphone, dash camera and show high fidelity and good generalization of our method in both visual quality lighting and motion dynamics. our code and animation are available on https://github.com/slowy07/ava.

1. **Introduction**

The sky is a key component of outdoor photography. Photographers in the wild sometimes have to face uncontrollable weather and lighting conditions. As a result, the photos and video captured may suffer from an over exposed or plain-looking sky. Let’s imagine if you were at a beautiful beach but the weather is bad. You took several photos but wanted a perfect sunset. Thanks to the rapid development of computer vision and augmented reality, the story above is not far from reality for everyone. In the past few years, we saw automatic sky editing / optimization as an emerging research topic. Some recent photo editors like photoshop have featured sky replacement tool boxes where users can easily swap out the sky in their photos with only a few clicks.

With the explosive popularity of short videos on social platforms, people are now getting used to recording their daily life with videos instead of photos. Despite the huge application needs, automatic sky editing in videos is still a rarely studied problem in computer vision. On one hand, although sky replacement has been widely applied in film production, manually replacing the sky regions in the video is laborious, time-consuming and even requires professional post-film skills. The editing typically involves frame-by-frame blue screen matting and background motion capturing. The users may spend hours on such tasks after considerable practice, even with the help of professional software. On the other hand, sky augmented reality started to be featured in recent mobile apps. For example, the stargazing app StarWalk, which can help users track stars, planets, constellations, and other celestial objects in the night sky with interactive augmented reality. Also, in a recent work Tran *et al*. a method called "Fakeye" is proposed for real-time sky segmentation and blending in mobile devices. However, the above approaches have critical requirements on camera hardware and cannot be directly applied to offline video processing tasks. Typically, to calibrate the virtual cameras to the real-world, these approaches require real-time pose signals of the camera from the inertial measurement units (IMU) integrated with the camera like the gyroscope in smartphones.

In this paper, we investigate an interesting question that whether the sky augmentation in videos can be realized in a purely vision-based manner, and propose a new solution for such a task. As we mentioned above, previous methods of this research topic either focus on static photos or require the gyroscope signals available along with the video frames captured on-the-fly. Our method, different from the above ones, has no specifications on capturing devices and is suitable for both online augmented reality and offline video editing applications. The processing is "one click and go" and no user interactions are needed.

Out method consist of multiple components

- **A sky matting network** for detecting sky regions in video frames. Different from previous methods that frame this process as binary pixel-wise classification (foreground v.s sky) problem, we design deep learning based coarse-to-fine prediction pipeline that produces soft sky matte for a more accurate detection result and more visually pleasing blending effect.

- A **motion estimator** for recovering the motion of the sky video captured by the virtual camera needs to be rendered and synchronized under the motion of the real camera. We suppose the sky and the in-sky object (e.g sun, clouds) are located at infinity and their movement relative to the foreground is affine.
- **A skybox** for sky image warping and blending. Given a foreground frame, a predicated sky matte, and the motion parameters, the skybox aims to warp the sky background based on the motion and blend it with the foreground. The skybox also applies relighthing and recoloring to make the blending result more visually realistic in its color and dynamic range.

We test our method on outdoor videos diversely captured by both dash cameras and handled smartphones. The results show our method can generate high fidelity and visually dramatic sky videos with very good lighting/motion dynamics in outdoor environments. Also, by using the proposed sky augmentation framework we can easily synthesize different weather and lighting conditions.

The contribution of our paper is summarized as follows:

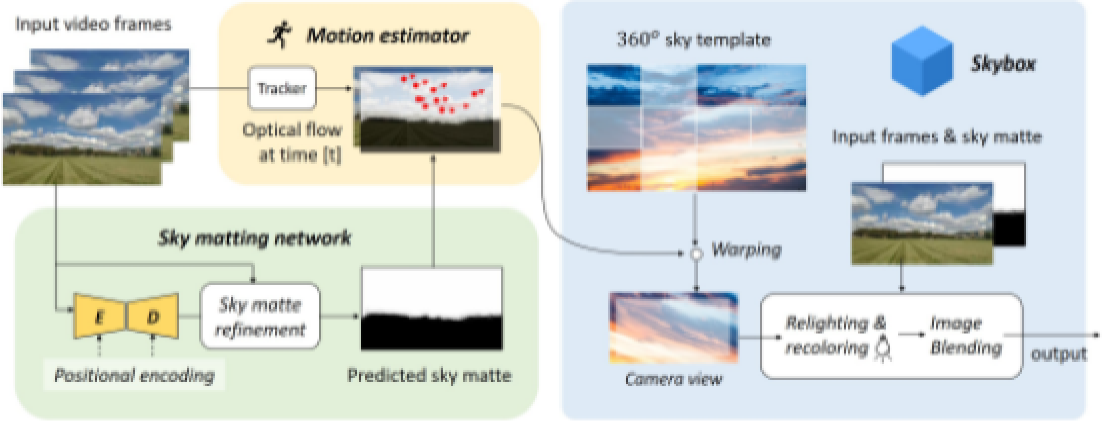
- We propose a new framework for sky augmentation in outdoor videos. Previous methods on this topic simply focus on the sky augmentation/edit on static images and rarely consider dynamic videos.

- Different from previous methods on outdoor augmented reality which require real-time camera pose signals from IMU. we propose a purely vision-based solution on such a task and applies to both online and offline application scenarios.

2. Related work

Sky replacement and editing are common in photo processing and film post-production. In computer vision, some fully automatic methods for sky detection and replacement in static photos have been proposed in recent years.Using these methods, users can easily customize the style of sky with their preference, with no need to have professional editing skills.

SkyFinder proposed by tao *et al* is to our best knowledge one of the first works that invoke sky replacement. This method first trains a random forest based asky pixel detector and then applies graph cut segmentation to produce refined binary sky masks. With the sky mask obtained, the authors further achieve attribute-based sky image retrieval and controllable sky replacement. Different from the SkyFinder in which the sky replacement is considered as a by-product,focus on solving this problem and at the same time producing high-quality sky masks by using deep CNN’s and conditional random fields. Apart from the sky segmentation.



PIC 1 an overview of our method. Our method consists of a skymatting network for sky matte prediction, a motion estimator for background motion estimation, and a skybox for blending the use-specified sky template into the video frames.

3. Methodology

PIC 1 shows an overview of our method consisting of three modules, a sky matting network, a motion estimator, and a skybox. Below give a detailed introduction of each of them.

a. Sky Matting

Image matting is a large class of methods that aims to separate the foreground of interests from animage, which plays an important role in image and video editing. Image matting usually involves prediction of a soft “matte” which is used to extract the foreground finely from the image, which naturally corresponds to our sky region detection process. Early works of image matting can be traced back to 20 years ago. Traditional image matting methods include sampling-based methods, and propagation-based methods. Recently, deep learning techniques have greatly promoted the progress of image matting research.

In our work, we take advantage of the deep convolutional neural network (CNN) and frame the prediction of sky matting networks consisting of a segmentation encoder E, a mask prediction decoder *D*, and a soft refinement module. The encoder aims to learn intermediate feature representations of a down-sampled. They are trained to predict the course of the sky matte. The refinement module takes in both of the coarse sky matte and the high resolution input and produces a refined sky matte. We use ResNet-like convolutional architecture as the backbone of our encoder (other choices like VGG also applicable) and build our decoder as an upsampling network with several convolutional layers. Note that since the sky region usually appears at the upper part of the image, we replace the conventional convolution layers with coordinate convolution layers at the encoder’s input layer and all the decoder layers, where the normalized y-coordinate values are encoded. We show such a simple modification brings noticeable accuracy improvement on the sky matte.

Suppose I and I_l represent and input image with full resolution and its down-sampled substitute. Our coarse segmentation network $f = \{E, D\}$takes in the I_l. suppose $Al = f(I_l)$ and $\widehat{A}$ represent the output of f and the groundtruth sky alpha matte. We train the f to minimize the distance between Al and the groundtruth $\widehat{A}$ in their raw pixel space. The objective function is defined as follows:

$$\mathcal{L}_f(I_l) = E_{I_l \in \mathcal{D}_l}\{\frac{1}{N_l}\|f(I_l) - \hat{A}_l\|_2^2\},$$

where $||.||\frac{2}{2}$ represent the pixel-wise distance between to images, $N_l$is the number of pixel in the output image, and D_l is a dataset consisting of down-sampled images on which the model F is trained.

After we obtain the coarse sky matte, we up-sample the sky matte to the original input resolution in the refinement stage. We leverage the guided filtering technique to recover the detailed structures of the sky matte by referring to the original image. The guided filter is a well-known edge/structure preserving filtering method which has better behavior and better computational complexity than other population filters like bilateral filters. Although recent image matting literature shows that using up-sampling convolutional architecture and adversarial training may also produce fine grained matting output, we choose guided filters mainly because of their high efficiency and simplicity. In our method, we use the full resolution image *I* as the guidance uimage and remove its red and green channels for better color contrast. With such a configuration, the filtering transfers the structures of the guidance image to the low resolution sky matte, and produces a more detailed and sharper result than the CNN's output minor computational overhead. We denote *A* as the predicted full resolution sky alpha matte after the refinement and *A* can be represented as follows:

$$A = f_{gf}(h(A_l), I, r, \epsilon),$$

b. Motion Estimation

Instead of estimating the motion of the camera as suggested by the previous method, we directly estimate the motion of the object at infinity and then render the virtual sky background by warping a 360° skybox template image into the perspective window. We built a skybox for image blending. The middle-right part of PIC. 2 briefly illustrates how we blend the frame and sky template based on the estimated motion and the predicted sky matte. In our method, we assume the motion of the sky patterns is modeled by an Affine matrix M ∈ R^{3x3} . Since the objects in the sky (e.g, the clouds, thesun, or the moon) are supposed to be located at infinity, we assume their perspective transform parameters are in fixed values and are already contained in the skybox background image. We compute optical flow using interactive Lucas-Kanade method with pyramids, and thus a set of sparse feature points can be frame-by-frame tracked. For each pair of adjacent frames, given two sets of 2D feature points, we use the RANSAC-based robust Affine estimation to compute the optimal 2D transformation with four degrees of freedom (limited to translation, rotation and uniform scaling). Since we focus on the motion of the sky background, we only use feature points located within the sky area to compute the Affine parameters. When there are not enough feature points detected, we run depth estimation on the current frame, and then use the feature points in the second far regions to compute the Affine parameters.

Since the feature points in the sky area usually have lower quality than those in the close-range area, we find that it is sometimes difficult to obtain stable motion estimation results solely based on setting a low tolerance on the RANSAC reprojection error. We, therefore, assume that the background motion between two frames is dominated by translation and rotation and apply kernel density estimation on the Euclid distance between the paired points in each adjacent frame. the matched points with small probability will be removed.

After obtaining the movement of each adjacent frame, the Affine matrix across between the initial frame and the *t*-th frame in the video can be written as the following matrix multiplication form:

$$\widetilde{\mathbf{M}}^{(t)} = \mathbf{M}^{(c)} \cdot (\mathbf{M}^{(t)} \cdot \mathbf{M}^{(t-1)} \dots \mathbf{M}^{(1)}),$$

where it represents the estimated motion parameters between frame *i* - 1 and *i*. $M^{(c)}$ are the center crop parameters of the skybox template, i.e shift and scale, depending on the field of view set by the virtual camera. The final sky background within camera’s perspective field at the frame *t* can be thus obtained by warping the background template *B* using the affine parameters M_t. In our method, we use the simple way to build the 360° sky background image where we tile the image when the perspective window goes out of the image border during warping. We set the final center cropping region of the warped sky template to the field of view of the virtual camera. WE set the size of the virtual camera’s view as ½ height x ½ width of the template skyimage. Note that other center cropping sizes are also applicable, and using a smaller range of cropping will produce a distant view effect captured by a telephoto camera.

c. Sky Image Blending

Suppose $I^{(t)}$, $A^{(t)}$, and $B^{(t)}$ are the video frame, the predicted sky alpha matte, and the aligned sky template image at time. In $A^{(t)}$A higher output pixel value means a higher probability the pixel belongs to the sky background. Based on the matting equation, we represent the newly composed $Y^{(t)}$ as the linear combination of the $I^{(t)}$ and the background $B^{(t)}$, with

$A^{(t)}$ as their pixel-wise combination weights:

$$Y^{(t)} = (1 - A^{(t)})I^{(t)} + A^{(t)}B^{(t)}.$$

Note that since the foreground $I^{(t)}$ and the background $B^{(t)}$ may have different color tone and intensity, directly performing the above combination may result in an unrealistic result. We thus apply the coloring and relighting techniques to transfer the colors and intensity from the background to the foreground. For each video frame $I^{(t)}$, we apply the following steps make corrections before the linear combination (4):

$$\widehat{I}^{(t)} \longleftarrow I^{(t)} + \alpha(\mu^{(t)}_{B(A=1)} - \mu^{(t)}_{I(A=0)}),$$
$$I^{(t)} \longleftarrow \beta(\widehat{I}^{(t)} + \mu^{(t)}_{I} - \widehat{\mu}^{(t)}_{I}),$$

where $\mu(t)_{(I)}$, μb (t) I, are the mean color pixel value of the image image $I^{(t)}$ and $\mu_i^{(t)}.(a = 0)$ and $\mu_i^{(t)}.(a = 1)$ are the mean color pixel value of the image $I^{(t)}$ at the pixel location of sky regions and the mean color pixel value of the image $B^{(t)}$ at the pixel location of non-sky regions. α and β are predefined coloring and relighting factors. In the above correction steps, the transfers the regional colortone from the background to the foreground while correcting the pixel intensity range of the re-colored foreground and make it compatible with the skybox background.

Table1: A detailed configuration of our sky matting network. “CoordConv” denotes a Coordinate convolution followed by a ReLU layer, “BN”, “UP” and “Pool” denote batch normalization, bilinear upsampling, and max pooling, respectively.

Layer	Config	Filter	Reso
E_1	CoorConv-BN-Pool	64/2	1/4
E_2	3 x ResBlock	256/1	1/4
E_3	4 x ResBlock	512/2	1/8
E_4	6 x ResBlock	1024/2	1/16

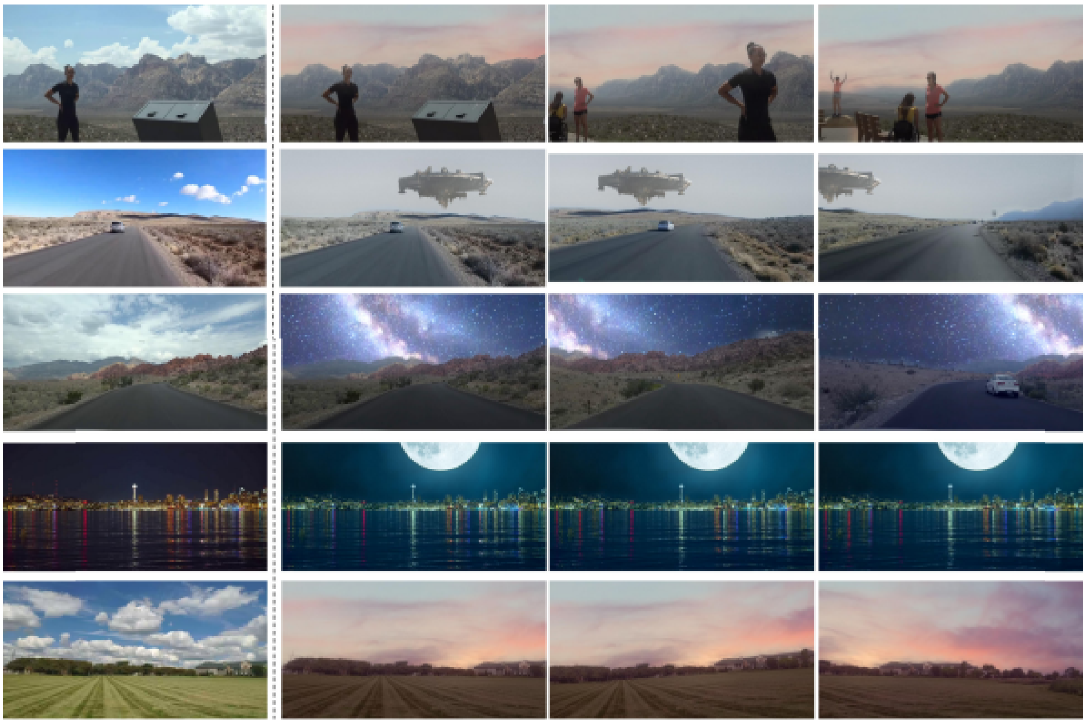
E_5	3 x ResBlock	2048/2	1/32
D_1	CoordConv-UP + E_5	2048/1	1/16
D_2	CoordConv-UP + E_4	1024/1	1/8
D_3	CoordConv-UP + E_3	512/1	1/4
D_4	CoordConv-UP + E_2	256/1	1/2
D_5	CoordConv-UP	64/1	1/1
D_6	CoordConv	64/1	1/1

d. Implementation Details

Network architecture, We use ResNet-50 as the encoder of our sky matting networks (fully connected layers removed). The decoder part consists of five convolutional upsampling layers (coordinate conv + relu + bilinear upsampling) and pixel wise prediction layer (coordinate + sigmoid). We follow the configuration of the “UNet” and add skip connections between the encoder and decoder layers with the same spatial size. Table 1 shows a detailed configuration of the networks.

Dataset. We train our sky matting network on the dataset introduced by Liba. This dataset is built based on the AED20K dataset and includes several subset where each of them correspond to using different methods for creating their ground truth of sky matte. We use the subset “ADE20K+DE+GF” for the training set and 885 images in the validation set.

Training details. We train our model by using adam optimizer. We set batch size to 8, learning rate to 0.0001, and betas to (0.9, 0.999). We stop training after 200 epochs. We reduce the learning rate to its 1/10 every 50 epochs. The input image size for training is 384x384. Our matting network is implemented based on PyTorch. Image augmentation we used in training include horizontal flip, random-crop with scale=(0.5, 1.0) and ratio=(0.9, 1.1), random-brightness with brightness_factor=(0.5, 1.5), random-gamma with $\gamma = (0.5, 1.5)$, and random-saturation with saturation_factor=(0.5, 1.5)



PIC 2: Video sky augmentation results by using our method. Each row corresponds to a separate video clip, where leftmost is the starting frame of the input video, and the image sequences on the right aare the processed output at different time steps.

Other details, In the sky matte refinement stage, we set the radius and regularization coefficient of the guided filter to r = 20 and ϵ = 0.01. In the motion estimation stage, when estimating the probability of the key points moving distance, we set the kernel type as “Gaussian” and set its bandwidth to 0.5 We set η = 0.1, i.e, remove those points whose probability is smaller than 0.1.

4. Experimental Analysis

We evaluate our methods on video sequences that are diversely captured in the world, including those by handled smartphones.

a. Sky Augmentation and Weather Simulation

PIC 2 shows a group of sky video augmentation results by using our method. Each row shows an input frame from the original video and the processing output at different time steps. The videos in the top-4 rows are downloaded from youtube
.

Testing scenario	PI [4]	NIQE [27]
CycleGAN (cloudy2cloudy)	7.094	7.751

Ours (cloudy2cloudy)	5.926	6.492
CycleGAN (cloudy2sunny)	6.684	7.070
Ourss (cloudy2sunny)	5.702	7.014

Table2: Quantitative evaluation on the image fidelity of our method and CycleGAN under different weather translation scenarios. For both PI and NIQE, lower scores indicate better.

We generate dynamic sky effects of “sunset”, “District-9 ship”, “super moon” and “Galaxy” (image generate from stable diffusion luna, see on : https://github.com/slowy07/luna) for the above video clips. Our method generates visually striking results with a high degree of realism. When we synthesize rainy images, we also add a dynamic rain layer (video source) and a haze layer on top, a dynamic rain layer (video source) and a haze layer on top of the result by using screen bending. We also compare our method with CycleGAN, a well-known unpaired image to image translation method, which is built based on conditional generative adversarial networks. We train CycleGAN on BDD100K dataset which contains outdoor traffic scenes under different weather conditions.

Resolution	speed	phase 1	phase 2	phase 3
640×360 pxl	98.03fps	0.0235	0.0070	0.0070
854×480 pxl	87.92fps	0.0334	0.0150	0.0186
1280×720 px	62.804fps	0.0565	0.0329	0.0386

Table 3: speed performance of our method at different output resolution and the time spent in different processing phases (1. sky matting; 2. motion estimation; 3. blending)

(NIQE). The two metrics were originally introduced as a no-reference image quality assessment method based on the natural image statistic and are recently widely used for evaluating image synthesis results. We see our method outperforms CycleGAN with a large margin in both quantitative and visual quality.

Another potential application of our method is data augmentation. Domain gap between datasets with limited samples and the complex real-world poses great challenges for data-driven computer vision methods. Even in modern scale datasets like MS-COCO and ImageNet, the sampling bias is still severe, limiting the generalization of the model to the real world. For example, domain sensitive visual perceptron models in self-driving may face problems at night or rainy days due the limited examples in training data. We believe our method has great potential for improving the generalization ability of deep learning models in a variety of computer vision tasks such as detection, segmentation, tracking, etc.

b. **Speed performance**

Table 3 shows the speed performance of our method. The inference speeds are tested on a desktop PC with an NVIDIA Geforce RTX 3090 GPU card and an 16 core of processor AMD Ryzen 9 Model 5900x. The speed at different output resolution and time spent in different processing stages are recorded. We can see our method reaches a real-time processing speed (98 fps) at the output resolution 640 x 320 and a near real-time processing speed (87 fps) at 854x480 but still has large rooms for speed up. As there is a considerable part of the time spent in the sky matting stage, one may easily speed up the processing pipeline by replacing the ResNet-50 with a more efficient CNN backbone, e.g MobileNet or EfficientNet.

c. **Controlled Experiments**

Segmentation vs Soft Matting. To evaluate the effectiveness of our sky matting network. We design the following experiment where we visually compare the blending results generated by 1) hard pixel segmentation, 2) soft matting before refinement, and 3) soft matting after refinement. We also compare the matting accuracy on the validation set of the dataset (ADE20K + DE + GF).

	w/ refinement	w/o refinement
W/positional encoding	27.31 / 0.924	27.17 / 0.929
w/o positional encoding	27.01 / 0.919	26.91 / 0.914

Table 4: Mean pixel accuracy (PSNR / SSIM) of our sky matting model on the dataset w/ and w/o using positional encoding (“CoordConv”s in table 1). The accuracy before and after the refinement is also reported. Higher scores indicate better.

Positional embedding. We also test the matting network without using the coordinate convolution and replace all those “CoordCoonv” layers in Table 1 with conventional convolution layers. We see a noticeable pixel accuracy drop when we remove the coordinate convolution layers (PSNR -0.26 and SSIM -0.015 before refinement; PSNR -0.30 and SSIM -0.015 before refinement; PSNR -0.30 and SSIM -0.015 after refinement), which suggest that the position encoding provides important priors for the sky matting task.

Color transfer and relighting. We compare the blending result of our method w/ or w/o using color transfer and relighting. These results provide a clear demonstration of the significance of our two-step correction - the color transfer can help eliminate the color-tone conflict between foreground and background while the relighting can correct the intensity of ambient light and can further enhance the sense of reality.

d. Controlled Experiments

The limitation of our method is twofold. First, since our sky matting network is only trained on daytime images, our method may fail to detect the sky regions on nighttime videos. Second, when there are no sky pixels during a certain period of time in a video, or there are no textures in the sky, the motion of the sky background cannot be accurately modeled. This is because the feature points we used for motion estimation are assumed to be located at infinity and using the feature points of objects that are second far away to estimate motion will introduce inevitable errors. In our future work, we will focus on three directions - the first one is scene-adaptive sky matting, the second one is robust background motion estimation, and the third one is to explore the effectiveness of sky-rendering based data augmentation for object detection and segmentation.

5. Conclusion

We investigate sky video augmentation, a new problem in computer vision, namely automatic sky replacement and harmonization in video with purely vision-based approaches. We decompose this problem into three proxy tasks: soft sky matting, motion estimation, and sky blending. Our method does not rely on the inertial measurement unit integrated on the camera devices, and also does not require user interaction. Using our method, users can easily generate highly realistic translation, and hopefully, it can be used as a new data augmentation approach to enhance the generalization ability of deep learning models in computer vision tasks.

References

1. Blend models, accessed Sep., 2022 . https://en.wikipedia.org/wiki/Blend modes
2. Yen Le Anh-Thu Tran. Fakeye: Sky augmentation with realtime sky segmentation and texture blending. In Fourth Workshop on Computer Vision for AR/VR, 2020
3. Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. International journal of computer vision, 56(3):221–255, 2004.
4. Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In Proceedings of the European Conference on Computer Vision (ECCV), pages 0–0, 2018.
5. Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In Proceedings eighth IEEE international conference on computer vision. ICCV 2001, volume 1, pages 105–112. IEEE, 2001.
6. Guanying Chen, Kai Han, and Kwan-Yee K Wong. Tomnet: Learning transparent object matting from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9233–9241, 2018.
7. Qifeng Chen, Dingzeyu Li, and Chi-Keung Tang. Knn matting. IEEE transactions on pattern analysis and machine intelligence, 35(9):2175–2188, 2013.
8. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009
9. Eduardo SL Gastal and Manuel M Oliveira. Shared sampling for real-time alpha matting. In Computer Graphics Forum, volume 29, pages 575–584. Wiley Online Library, 2010
10. Clement Godard, Oisin Mac Aodha, Michael Firman, and ´ Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In Proceedings of the IEEE international conference on computer vision, pages 3828–3838, 2019
11. Ariel Gordon, Elad Eban, Ofir Nachum, Bo Chen, Hao Wu, Tien-Ju Yang, and Edward Choi. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1586–1595, 2018.
12. Tavi Halperin, Harel Cain, Ofir Bibi, and Michael Werman. Clear skies ahead: Towards real-time automatic sky replacement in video. In Computer Graphics Forum, volume 38, pages 207–218. Wiley Online Library, 2019.

13. Kaiming He, Christoph Rhemann, Carsten Rother, Xiaoou Tang, and Jian Sun. A global sampling method for alpha matting. 2011.
14. Kaiming He, Jian Sun, and Xiaoou Tang. Guided image fil- tering. IEEE transactions on pattern analysis and machine intelligence, 35(6):1397-1409, 2012.
15. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770-778, 2016.
16. Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1, volume 1, pages 654-661. IEEE, 2005.
17. Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.
18. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
19. Cecilia La Place, Aisha Urooj, and Ali Borji. Segmenting sky pixels in images: Analysis and comparison. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1734–1742. IEEE, 2019.
20. Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. IEEE transactions on pattern analysis and machine intelligence, 30(2):228–242, 2008.
21. Orly Liba, Longqi Cai, Yun-Ta Tsai, Elad Eban, Yair Movshovitz-Attias, Yael Pritch, Huizhong Chen, and Jonathan T Barron. Sky optimization: Semantically aware image processing of skies in low-light photography. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 526–527, 2020.
22. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence ´ Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.
23. Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In Advances in Neural Information Processing Systems, pages 9605–9616, 2018.
24. Cewu Lu, Di Lin, Jiaya Jia, and Chi-Keung Tang. Two-class weather classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3718–3725, 2014.
25. Sebastian Lutz, Konstantinos Amplianitis, and Aljosa Smolic. Alphagan: Generative adversarial networks for natural image matting. arXiv preprint arXiv:1807.10088, 2018.
26. Radu P Mihail, Scott Workman, Zach Bessinger, and Nathan Jacobs. Sky segmentation in the wild: An empirical study. In 2016 IEEE Winter

Conference on Applications of Computer Vision (WACV), pages 1-6. IEEE, 2016.

27. Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer. IEEE Signal processing letters, 20(3):209-212, 2012.
28. Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. IEEE signal processing magazine, 32(3):53-69, 2015
29. Saumya Rawat, Siddhartha Gairola, Rajvi Shah, and PJ Narayanan. Find me a sky: A data-driven method for color consistent sky search and replacement. In the International Conference on Multimedia Modeling, pages 216-228. Springer, 2018.
30. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. Unet: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234-241. Springer, 2015.
31. Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510-4520, 2018.
32. Ehsan Shahrian, Deepu Rajan, Brian Price, and Scott Cohen. Improving image matting using comprehensive sampling sets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 636-643, 2013.
33. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
34. Alvy Ray Smith and James F Blinn. Blue screen matting. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 259-268, 1996
35. Litian Tao, Lu Yuan, and Jian Sun. Skyfinder: attribute based sky image search. ACM transactions on graphics (TOG), 28(3):1-5, 2009.
36. Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In European conference on computer vision, pages 352-365. Springer, 2010.
37. Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In Computer Vision and Pattern Recognition (CVPR), 2017.
38. Yuanjie Zheng and Chandra Kambhamettu. Learning based digital matting. In Computer Vision, 2009 IEEE 12th International Conference on, pages 889-896. IEEE, 2009.
39. Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle consistent adversarial networks. In Computer Vision (ICCV), 2017 IEEE International Conference on, 2017.