Computer Modeling in Engineering & Sciences CMES, vol.122, no.2, pp.593-617, 2020

Data Augmentation Technology Driven By Image Style Transfer in Self-Driving Car Based on End-to-End Learning

Dongjie Liu[1] , Jin Zhao[1,*], Axin Xi[2], Chao Wang[1], Xinnian Huang[1], Kuncheng Lai[1] and Chang Liu[1]

Abstract: With the advent of deep learning, self-driving schemes based on deep learning are becoming more and more popular. Robust perception-action models should learn from data with different scenarios and real behaviors, while current end-to-end model learning is generally limited to training of massive data, innovation of deep network architecture, and learning in-situ model in a simulation environment. Therefore, we introduce a new image style transfer method into data augmentation, and improve the diversity of limited data by changing the texture, contrast ratio and color of the image, and then it is extended to the scenarios that the model has been unobserved before. Inspired by rapid style transfer and artistic style neural algorithms, we propose an arbitrary style generation network architecture, including style transfer network, style learning network, style loss network and multivariate Gaussian distribution function. The style embedding vector is randomly sampled from the multivariate Gaussian distribution and linearly interpolated with the embedded vector predicted by the input image on the style learning network, which provides a set of normalization constants for the style transfer network, and finally realizes the diversity of the image style. In order to verify the effectiveness of the method, image classification and simulation experiments were performed separately. Finally, we built a small-sized smart car experiment platform, and apply the data augmentation technology based on image style transfer drive to the experiment of automatic driving for the first time. The experimental results show that: (1) The proposed scheme can improve the prediction accuracy of the end-to-end model and reduce the model's error accumulation; (2) the method based on image style transfer provides a new scheme for data augmentation technology, and also provides a solution for the high cost that many deep models rely heavily on a large number of label data.

Keywords: Deep learning, self-driving, end-to-end learning, style transfer, data augmentation.

[1] School of Mechanical Engineering, Guizhou University, Guiyang, 550025, China.

[2] Faculty of electronic and information engineering, Xi'an Jiaotong University, Xi'an, 710049, China.

[*] Corresponding Author: Jin Zhao. Email: zhaoj@gzu.edu.cn.

Received: 18 September 2019; Accepted: 9 December 2019.

1 Introduction

In recent years, with the rapid development of machine learning algorithms, it has become more and more important to apply artificial intelligence technology to solve problems in autonomous vehicles and unmanned aerial vehicle. However, due to the complicated driving environment and high cost of high-precision laser radar sensors, there are also many challenges in the realistic use of unmanned driving. The application of low-cost vision cameras to solve control problems in self-driving has been one of the important research directions in the field of intelligent vehicles [Sotelo, Rodriguez, Magdalena et al. (2004); Urmson, Anhalt, Bagnell et al. (2008)]. Researchers apply machine learning methods to train agents (such as vehicles, unmanned aerial vehicles, game agents, etc.) to complete navigation tasks in an unknown environment [Bojarski, Del Testa, Dworakowski et al. (2016); Chen, Seff, Kornhauser et al. (2015); Koutník, Cuccu, Schmidhuber et al. (2013)]. ALVINN was the first one to prove that end-to-end learning had the ability to form an autopilot system. In the absence of an CNN (convolution neural network), they used a three-layer fully connected network by inputting monocular cameras and radar data, ultimately enabling vehicle to drive 400 meters along the road [Pomerleau (1989)]. Muller et al. built a smart car consisting of two cameras and other sensors, collected 24,000 frames of labeled data generated by human-controlled vehicles, and fed the data into a six-layer CNN for training [Muller, Ben, Cosatto et al. (2006)]. The bench was tested on an unknown open terrain and successfully avoided obstacles such as trees and rocks. Bojarski et al. collected 72 hours of data through a camera placed in front of the car and trained the agent using Deep Neural Network (DNN), which finally led to autonomous driving in the parking lot [Bojarski, Del Testa, Dworakowski et al. (2016)]. Xu et al. advocated learning a general vehicle motion model from large-scale crowd video data and developing an end-to-end trainable architecture for observation from instantaneous monocular cameras and previous vehicle states [Xu, Gao, Yu et al. (2017)]. Kim et al. constructed a wide data set by collecting data under various road conditions, achieving robust control of the vehicle [Kim and Park (2017)].

All of the above works are based on the collection of a large amount of data to achieve autonomous driving, which requires a lot of labor and time. Coincidentally, in the past research work, self-driving was also achieved by improving the neural network structure. In [Du, Guo and Simpson (2017)], Du et al. proposed a three-dimensional convolution model with residual connections and recursive LSTM layers using different deep learning techniques such as transfer learning, 3D-CNN, LSTM and RESNET. In the work of Viswanath et al., an embedded convolutional neural network (Jacintonet) structure was raised and its reliability was verified in the simulation environment [Viswanath, Nagori, Mody et al. (2018)]. Mahdavian et al. also put forward an end-to-end neural network architecture for training unconstrained autonomous vehicles in a simulation environment [Mahdavian and Martinez (2018)]. However, in our current research, it is clear that the robustness of the model trained with the previously collected data will be significantly reduced when the surrounding environment changes (light or roadside information changes, etc.) or road features are not obvious. Although the traditional data augmentation methods (such as horizontal flip, random clipping, scaling, rotation and elastic deformation, etc. [Krizhevsky, Sutskever and Hinton (2012)]) are used to augment the data, these can only make the model learn rotation and proportional invariance

[Bojarski, Del Testa, Dworakowski et al. (2016); Kim and Park (2017)]. When the texture, color and light in the environment change, the adaptability of the model will be greatly reduced [Jackson, Atapour-Abarghouei, Bonner et al. (2018)]. Our work is a new data augmentation technique based on image style transfer drive to augment limited data sets, and then obtain a robust end-to-end control model.

Style transfer refers to a kind of image processing algorithm that modifies the visual style of an image while preserving the semantic content of the image. It can change the likelihood of distribution of low-level visual features in images, as proposed by Gates et al. [Gatys, Ecker and Bethge (2016)]. Tobin et al. [Tobin, Fong, Ray et al. (2017)] and Atapour & Breckon [Atapour-Abarghouei and Breckon (2018)] successfully promoted the graphics in the virtual environment to the real-world through style transfer. Early research methods modeled the parametric model of visual texture by constructing constraints to match the edge space statistics of visual patterns [Julesz (1962); Portilla and Simoncelli (2000); Freeman and Simoncelli (2011)]. In recent years, spatial image statistics collected from intermediate features of the image classifier have been shown to capture visual textures [Simonyan and Zisserman (2014); Gatys, Ecker and Bethge (2015)]. However, these methods have low flexibility, poor real-time performance, and high requirements for computer operations [Ghiasi, Lee, Kudlur et al. (2017)]. Dumoulin et al. have recently demonstrated that a transfer network of 32 different painting styles can be trained by using conditional normalization parameters [Dumoulin, Shlens and Kudlur (2016)]. Our work is to create an arbitrary style generation network that uses a novel style transfer method to disturb the color, texture and contrast ratio of the content image to achieve the purpose of randomization of the style, and finally augments to the scenarios that was not observed before the vision system.

The rest of this paper is organized as follows: Section 2 describes the research work related to this work. Section 3 describes the proposed arbitrary style generation network architecture in detail. In the fourth section, Section 4.1 determines the appropriate image style intensity through image classification experiments, and verifies the effectiveness of our method by comparing with traditional image data augmentation techniques. The classification results of data augmentation techniques based on image style transfer drive on invisible domains are tested in Section 4.2. As for Section 4.3, simulation experiments are carried out in three different experimental environments. Finally, an experimental platform is set up and a real vehicle experiment is tested under four various sites. What follows in Section 5 is a brief conclusion of the process and outlook toward the full text. In summary, this article has made the following contributions:

1. We propose a novel, fast and arbitrary image style transfer method, which can be used for image style randomization, and explore the possibility of applying it to image data augmentation. Based on the existing image classification network architecture, our method further improves the accuracy of image classification.

2. We introduce a multivariate Gaussian distribution function, which provides random style embedding vectors for image style, and linearly interpolates with the vector predicted on the style learning network to control the style intensity. Finally, a set of random normalization constants is provided for the style transfer network. This breaks through the limitations of limited-style images and can quickly transform any given

content image to produce an infinite number of stylized images.

3. Style-transferred images are used as data augmentation to provide a new approach for traditional image data augmentation techniques. We applied it to self-driving based on end-to-end learning for the first time, reducing the difficulty of collecting enough data and improving the performance of image-based deep learning algorithms.

2 Related work

2.1 Image style transfer

Image style transfer is to transform the style of any other image to the input one by computer, but to keep the semantic content of the original image unchanged, that is, given a content image *c* and a style image *s*, the generated image *g* is similar to image *c* in content and similar to image *s* in style. It is assumed that each layer of the convolutional neural network outputs a number of feature maps in the form [n_H, n_W, n_C], where n_H, n_W, and n_C are the height, width, and number of channels of the feature map, respectively. The output of each layer is a 3-dimensional array. This paper uses a^l to represent the output of a layer of convolutional neural network. According to the study by Gatys et al., the image texture feature, i.e. style, can be extracted using the *Gram* matrix [Gatys, Ecker and Bethge (2016)]. The content loss of the content image *c* and the generated image *g* is:

$$L_{content} = \frac{1}{n_H n_W n_C} \| a^l(c) - a^l(g) \|_2^2 \tag{1}$$

The style loss of the style image *s* and the generated image *g* is:

$$L_{style} = \frac{1}{(2n_H n_W n_C)^2} \| G(a^l(s)) - G(a^l(g)) \|_F^2 \tag{2}$$

The total optimization goal can be indicated as:

$$L_{total} = \alpha L_{content} + \beta L_{style} \tag{3}$$

Among them, α and β are the weighting coefficients of the content loss and style loss in the total loss, and the two values are adjusted to obtain the generated map with different emphasis. $G(a^l(s))$ is the *Gram* matrix associated with the layer *l* activation function.

The gradient descent method is used to optimize the image iteratively, which requires at least 1000 iterations [Gatys, Ecker and Bethge (2016)]. Even with GPU acceleration, it takes about 20 minutes, resulting in low computational efficiency [Ghiasi, Lee, Kudlur et al. (2017)]. Some researchers have solved this problem by establishing a secondary network [Johnson, Alahi and Li (2016); Li and Wand (2016); Ulyanov, Lebedev, Vedaldi et al. (2016)]. Although these methods increase the speed of calculation, the flexibility is very low. Since the neural network only learns a single style, if other styles are needed, the network must be retrained, which makes the robustness of the model worse.

Yanai [Yanai (2017)] proposed a feedforward neural style migration network that matched the learned style embedding vector to the convolutional layer in the style transformer network and transferred invisible style. Ghiasi et al. [Ghiasi, Lee, Kudlur et

al. (2017)] designed a style prediction network specifically for predicting affine transformation parameters for each style image. The network requires large-scale style and content images for training, and eventually can be extended to any image style. However, this method has an obvious disadvantage, that is, the data-driven approach will inevitably lead to a stylization result that is very relevant to the type and number of training concentrated styles.

In this paper, we being inspired by the flexibility, rapidity and style diversity of the method in Ghiasi et al. [Ghiasi, Lee, Kudlur et al. (2017); Yanai (2017)], the style prediction network *inception-v3* architecture in literature [Szegedy, Vanhoucke, Ioffe et al. (2016)] is fine-tuned, and the multivariate Gaussian distribution is added to provide the normalization constant of linear interpolation for style transfer networks. This provides a new solution for arbitrary image style transfer technique.

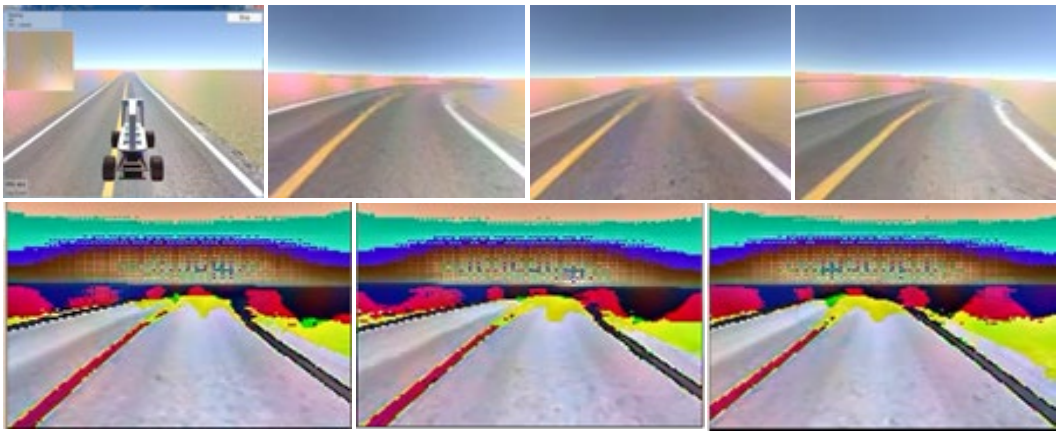
2.2 Image data augmentation

Image data augmentation is a method of deriving new data from raw data. Traditional image augmentation methods include horizontal flipping, random cropping, scaling, rotation, and elastic deformation [Krizhevsky, Sutskever and Hinton (2012)]. One of their functions is to improve the generalization ability of the model by increasing the amount of data, and the other is to increase the robustness of the model by adding noise data. For example, the literature [Claudiu Ciresan, Meier, Gambardella et al. (2010); Simard, Steinkraus and Platt (2003)] simulates the stroke changes caused by hand muscle oscillations by the elastic deformation method, and then augments the handwritten datasets MNIST. In the work of Bojarski et al., they provided data augmentation by adding two additional perspectives for simulating visual offset and correction control [Bojarski, Del Testa, Dworakowski et al. (2016)]. Similarly, Du et al. increased the offset of the steering angle by horizontally moving the camera to simulate the effect of the car on different locations on the road [Du, Guo and Simpson (2017)]. Then arbitrary vertical movement of the image was also to simulate the vehicle's operating conditions on uphill and downhill. At the same time, image color space conversion is used to simulate shadow and brightness changes in the external ambience. Finally, the generalization ability and robustness of the end-to-end control model to the unknown environment are improved.

Self-driving with end-to-end learning methods relies heavily on datasets. The vision sensor is sensitive to the factors of camera distortion, light, background change, and disturbance in the environment, which ultimately lead to model failure. Fig. 1 shows the driving decision heatmap of the benchmark model in the simulator environment and the real-world scenarios. The model focuses on the high-lighting part of the picture, which provides a basis for controlling vehicle steering or acceleration and deceleration. As can be seen from the results of the simulator scenarios, the model makes decisions by identifying lane lines. In Real-world Scenario 1, the model not only identifies lane lines, but also makes decisions by identifying other objects in the environment, such as pillars and bright windows. In Scenario 2 and Scenario 3, due to the changes of light and external environment (pedestrian walking, billboards shift, etc.), the previously trained model recognizes the more prominent texture in the current environment, such as the reflection of smooth floor surface, billboards and flare, etc., and unable to identify clear

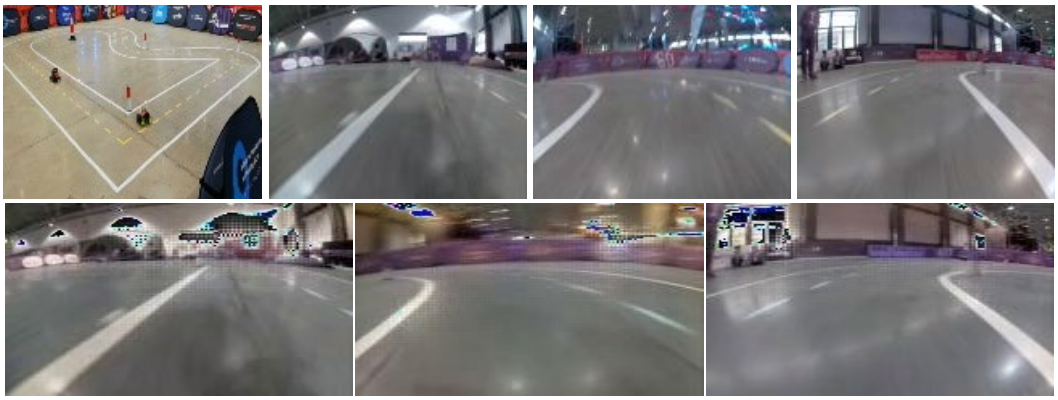
lane lines. Unfortunately, the benchmark model failed to make decisions by identifying lane lines. Even if the external environment changes, the lane lines that does not change is the key to achieving self-driving. Therefore, our main work is how to make the model re-focus the lane lines itself.

From the analysis results of Fig. 1, it can be found that the models trained on specific data in specific areas are difficult to generalize to other scenarios that are not seen during learning. Therefore, our work seems to solve the domain adaptation problem, but data augmentation is not a domain adaptation, usually it is used to reduce overfitting and improve the generalization of unseen scenarios in the same domain [Gretton, Smola, Huang et al. (2009)]. A study by Geirhos et al. showed that models trained on ImageNet datasets were more dependent on image textures, but ResNet-50 trained on ImageNet with random textures could make CNNs depend on shapes rather than textures [Geirhos, Rubisch, Michaelis et al. (2018)]. Therefore, we propose a new image style transfer method that disrupts the texture and color of an image through a style transfer network. Then we explored the application of this image style migration-driven data augmentation technology in self-driving cars based on end-to-end learning.

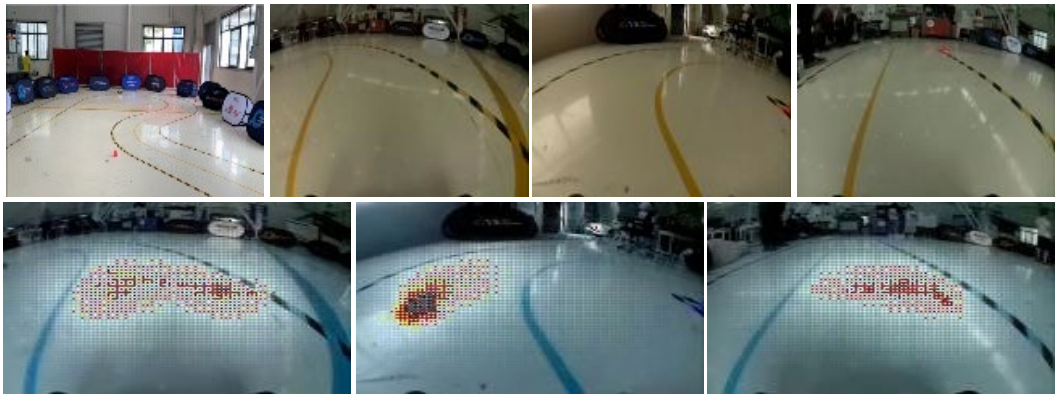


(a): Udacity's Self-Driving Simulator Scenario

(b): Real-World Scenario 1



(c): Real-World Scenario 2



(d): Real-World Scenario 3

Figure 1: Visualized heatmap of the benchmark model

3 Data augmentation method based on image style transfer drive

Early image style transfer methods were either slower in iteration [Portilla and Simoncelli (2000); Gatys, Ecker and Bethge (2015)] or only applicable to limited styles (poor flexibility) [Johnson, Alahi and Li (2016); Li and Wand (2016); Ulyanov, Vedaldi and Lempitsky (2016)]. Our method is inspired by the construction of the style transfer network in Ghiasi et al. [Ghiasi, Lee, Kudlur et al. (2017)] as the codec structure and the literature [Dumoulin, Shlens and Kudlur (2016)] to integrate the style image into normalization parameters, and proposes an image style transfer algorithm for data augmentation. It not only improves the flexibility of style transfer (the image style can be arbitrarily), but also improves the prediction accuracy of the steering angle of the self-driving based on end-to-end control.

3.1 Introduction to arbitrary style generation network architecture

As shown in Fig. 2, our system consists of a style migration network $\boldsymbol{T}$, a style learning network $\boldsymbol{P}$, a proposed multivariate Gaussian distribution function $\boldsymbol{G}$ (A two-dimensional Gaussian distribution represents the G.), and a VGG-16 style loss network $\boldsymbol{\Psi}$. The dashed

lines represent the process training the style transfer network T and the style learning network P on the Painter by Numbers Dataset and Describable Textures Dataset. The solid lines represent the random stylization process of the image. The specific details are as follows:

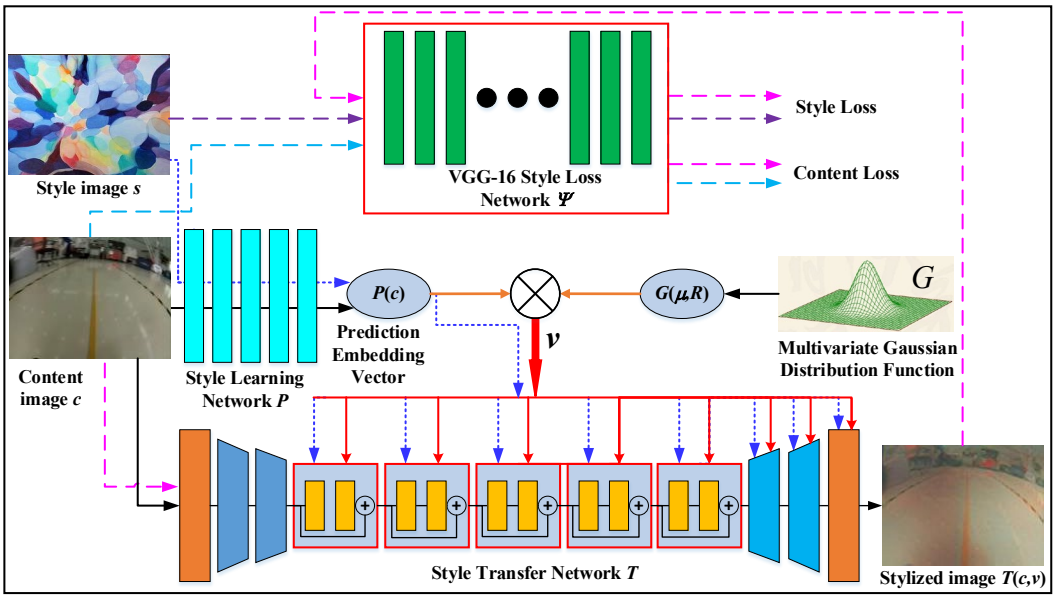


Figure 2: An overview of arbitrary style generation network. It includes a style learning network ***P***, a proposed multivariate Gaussian distribution function ***G***, a style transfer network ***T***, and a *VGG-16* style loss network ***Ψ***. ***v*** is style embedding vector after linear combination of ***P(c)*** and ***G(μ,R)***

Multivariate Gaussian distribution function *G*. The style transfer network proposed by Ghiasi et al. needs to be trained by a large number of images (Painter By Numbers (PBN) dataset), and this data-driven approach inevitably leads to the style result is closely related to the type and quantity of style in the training dataset [Ghiasi, Lee, Kudlur et al. (2017)]. Based on this, we introduce a multivariate Gaussian distribution function to generate arbitrary style embedding vector ***G(μ,R)***, and the mean μ and covariance R of this distribution match the distribution of style embedding vector generated by the PBN datasets. Therefore, the style embedding vector ***G*** obtained by randomly sampling the multivariate Gaussian distribution is equivalent to selecting one image in the PBN dataset as the style image without the entire dataset, thus reducing the computational cost. Due to the randomness of the multivariate Gaussian distribution, it produces a myriad of styles image s that augment the content image c to arbitrary stylized image, which is then extended to previously unobserved styles, ultimately achieving image augmentation.

Style learning network *P*. It mainly follows the *Inception-v3* architecture [Szegedy, Vanhoucke, Ioffe et al. (2016)] and is employed to predict the embedding vector ***P(c)*** of a content image c. As a result, this guarantees that most of the styles on an image c remain the same. On the Mixed-6e layer, the mean value of each activation channel is calculated

and a 768-dimensional feature vector. Then, two fully connected layers are employed to predict final embedding vector. The first fully connected layer is constructed into 100 units for the purpose of compression representation. Finally, the predicted embedding vector ***P(c)*** is linearly interpolated with the random style embedding vector ***G(μ,R)*** from the multivariate Gaussian distribution to provide a set of random normalization constants for the style transfer network ***T***. The interpolated style embedding vector is ***v***. where $\boldsymbol{v}=\alpha G(\mu, R)+(1-\alpha)P(c)$ and $\boldsymbol{v} \in \mathbb{R}^{100}$.

Style transfer network *T*. It is a deep residual network, mainly following [Johnson, Alahi and Li (2016)], which is used to generate stylized images. The network includes 3 downsampling layers, 5 residual blocks of 2 convolutional layers each, and 3 upsampling layers. This is a total of 16 convolutional layers, with the middle 10 layers having 128 channels each and does not contain a pooling layer. In order to avoid the checkerboard phenomenon, the nearest neighbor interpolation is used in the upsampling, and then convolution calculation is performed. In order to avoid black blocks, the activation function of the last layer of convolutional layer is replaced by ReLU with Sigmoid [Dumoulin, Shlens and Kudlur (2016)].

In addition to the content image *c*, the input of the network requires a multi-Gaussian-style embedding vector and a predictive embedding vector ***P(c)*** from the learning network ***P*** on the input image *c* to affect the style transfer network. This process is conditional instance normalization. Since the convolution weights in the style transfer network can be shared by multiple styles, the scaling parameter λ and the shift parameter δ are added to the network. That is, the renormalized feature map of the feature map *x* before normalization can be represented as:

$$x' = \lambda(\frac{x-\mu_c}{\sigma}) + \delta \tag{4}$$

where μ_c and σ are the mean and standard deviation on the spatial axis of the feature map *x*, and λ, δ are scalars obtained by embedding the style through the fully connected layer. Consequently, an arbitrary style transfer network ***T*** can be represented as Eq. (5). That is, the output *x* of the network ***T*** depends both on the content image *c* and on the linear combination ***v*** of ***P(c)*** and ***G(μ,R)***.

$$x = T(c, v) \tag{5}$$

Style loss network *Ψ*. It is a *VGG16* network consisting of content loss and style loss, respectively, for constraining L_{total} during neural network training. The content image *c* does not consider low-level information such as edge texture, and only selects one layer in the high-level information. The style loss calculates each large convolution layer and then calculates the *Gram* matrix. Next, the generated image *x* is compared with the feature map of the content image *c* and the style image *s* input into the network, respectively. Finally, the difference between the stylized image *x* and the content image and the style image is reduced.

3.2 Style randomization procedure

In general, the normalization parameter for linear interpolation comes from the

predictive vector of the style prediction network for the style image ***s*** [Ghiasi, Lee, Kudlur et al. (2017)]. This style prediction network needs to be trained through large-scale style images and content images, but this data-driven approach inevitably leads to a stylization effect that is highly correlated with the type and number of image style of training dataset. Therefore, in order to eliminate this correlation, we introduce a multivariate Gaussian distribution as a random style-embedding vector. The image stylization effect is shown in Fig. 3. The shape of the stylized image is preserved, but the color, texture, and contrast in the image are randomly varying. Qualitatively speaking, our approach can produce arbitrary styles and provide solutions for generalizing to unknown scenarios. The images used here for stylization come from the Flowers-17 dataset and the road images we collected.

In addition, in order to control the image style intensity and achieve the best end-to-end model, we linearly interpolate the predictive style embedding vector ***P(c)*** of the input image and the randomly sampled style embedding vector ***G(μ,R)***. Among them, the style learning network keeps most of the content of the input image unchanged, and the multivariate Gaussian distribution is used to extend to previously unobserved scenarios. Therefore, the final style embedding vector is defined as:

$$v = \alpha G(\mu, R) + (1-\alpha)P(c) \tag{6}$$

$$\mu = \mathrm{E}_s[P(c)] \tag{7}$$

$$R_{i,j} = \mathrm{cov}[P(c)_i, P(c)_j] \tag{8}$$

where style intensity α is the style intensity ratio; the content image c is predicted by the style learning network ***P*** to generate the embedded vector ***P***(c); $\boldsymbol{\mu}$ and ***R*** are the mean and covariance of the multivariate Gaussian distribution, which are derived from the mean and covariance of the vector ***P***(c). Fig. 4 shows the stylized results with different α values. As α gradually increases, the shape of flowers, lane lines and surrounding objects in the image is more and more prominent. In addition, when α is relatively smaller, surrounding objects in the image containing the lane lines are blurred. As for how to choose the appropriate α value, we will introduce it in the next chapter.

Figure 3: Image stylized display

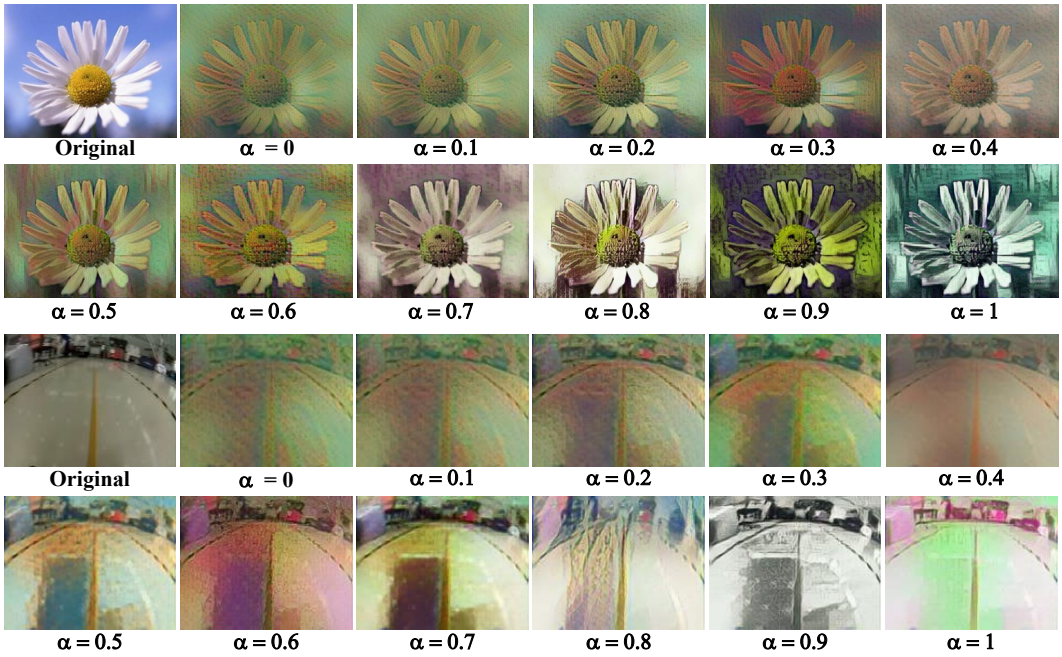


Figure 4: Stylized images display with different α values

4 Experimental results

We use experiments to verify the feasibility of data augmentation technology based on image style transfer drive, that is, to improve the generalization and robustness of the end-to-end control model by the proposed algorithm to disturb the color, texture and contrast ratio of the image. Therefore, our work only compares our method with existing image augmentation techniques.

We validate the rationality of the proposed method and determine the best style strength α by general image classification and cross-domain classification experiments. Among them, cross-domain classification experiments further put forward higher requirements and challenges for our image style transfer algorithm. In the image classification experiment, we used the Flowers-17 benchmark dataset and determined the best α by hyperparameter search. Then, we did a cross-domain classification experiment on the Office-Caltech dataset to test the generalization ability of the proposed style transfer algorithm for invisible domains. To ensure the fairness of the experiment, we set the number of augmented and unaugmented images to 1:2. Finally, simulation experiments were carried out employing image data in the real-world scenarios, and a reduced-size smart car experiment platform was built and verified in four real-world environments.

4.1 Image classification

We use the *Flowers-17* small data set to find the appropriate style intensity α and test the performance of the proposed method. The dataset has 17 categories, each containing 80 images. This is a challenge to avoid overfitting when training deep learning models, so it

puts high demands on data augmentation algorithms. We use a *VGG-16* network with a learning rate of 10^{-3}, a weight attenuation of 10^{-6}, and an iteration of 100 times. In addition, the batch size is 64, and the early stopping is set to 10.

We find the appropriate style intensity α by the Hyperparameter search experiments, where the α value is set from zero to one, with an interval of 0.1. In addition, in order to ensure the fairness of the experiment, we set the ratio of unaugmented data size to augmented data size to 2: 1. When the images of style augmentation are mixed with the traditional data augmentation, we set the ratio of its data size to 1:1, which guarantees that the ratio of unaugmented data to augmented data is still 2:1. We obtained a more suitable value through the mean and standard deviation of the five experiments. In the hyperparameter search experiments shown in Fig. 5, the red line depicts the mean of the accuracy obtained at differentα values, and the error bars (blue line) denote the standard deviation. It can be obtained from Fig. 5 that when α=0.4, the result is optimal.

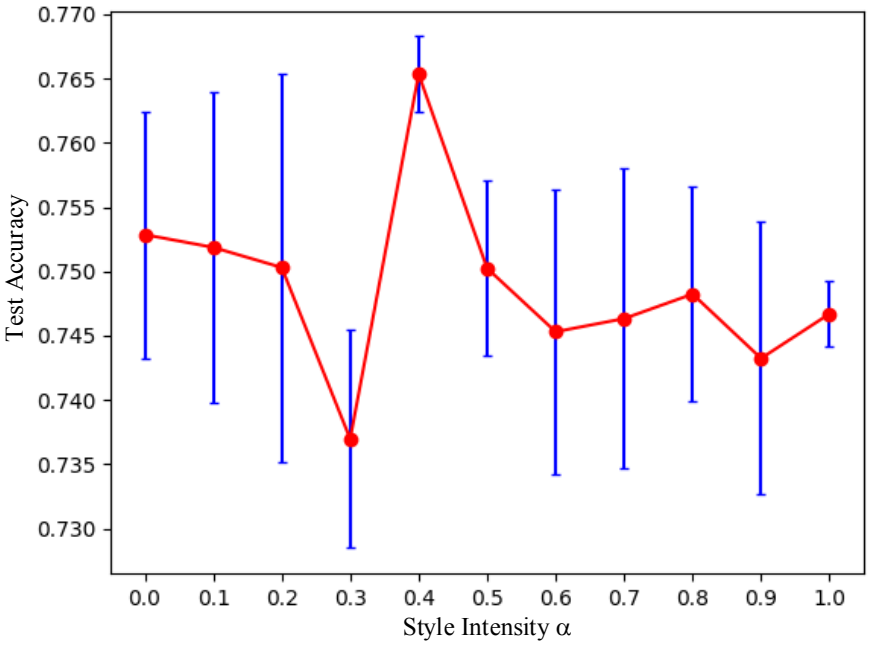


Figure 5: Hyperparameter Search experiments

At α=0.4, we compared the style transfer model with the traditional augmentation techniques (such as flipping, translation, adding random noise, random clipping, and random brightness, etc.). As shown in Fig. 6(b), the unaugmented model obtains the worst classification accuracy, and the model with style transfer is superior to it. By comparing the training accuracy in Fig. 6(a), it can be seen that the unaugmented model has overfitting, and the models with data augmentation have obtained better outcomes. Obviously, our image transfer-driven data enhancement method (Style Transfer) has a faster convergence speed. The mixed augmentation (style transfer and traditional augmentation techniques) achieves the fastest convergence rate and highest accuracy (Mix). In summary, we can apply the style transfer algorithm to data augmentation to

avoid overfitting and improve the convergence rate, but combined with the traditional data augmentation technique, we get unexpected results on the *VGG-16* network that is not currently advanced.

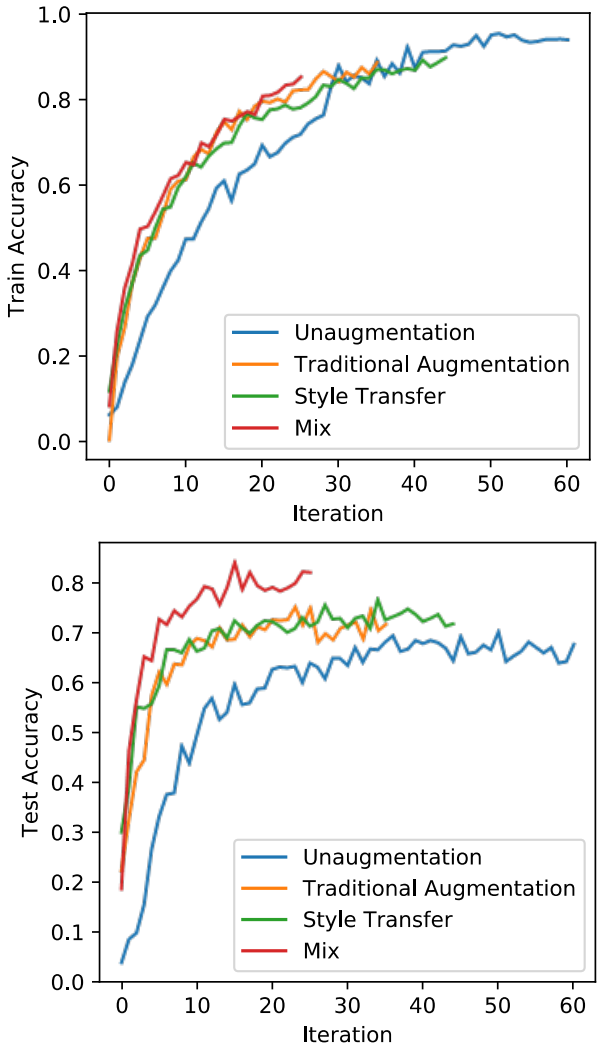


Figure 6: With the *Flowers-17 dataset* as the training set, we recorded the accuracy of four different data augmentation techniques in the training process. Among them, (*a*) is the training accuracy curve of different models; (*b*) is the test accuracy curve of different models

4.2 Cross-Domain classification experiment

To challenge and test the generalization of the proposed style transfer method for invisible domains, we tested it on the *Office-Caltech dataset*. The Office dataset consists of Webcam, DSLR, and Amazon domains. Each domain consists of 10 classes (backpack, bike, calculator, headphones, keyboard, laptop, monitor, mouse, mug and projector). The background of the image in Amazon domain is simple, and we select 958 pictures. The

Webcam domain is taken from the camera and we have 295 pictures. The Caltech dataset also consists of 10 categories, and its image is a mix of camera shots and Amazon product images, with a total of 1123 pictures.

Table 1: Test accuracy on the Office-Caltech dataset. *A*, *C*, and *W* represent *Amazon, Caltech, and Webcam*

Task	Network	Accuracy (%)			
		None	Trad	Style	Mix
A→C	VGG-16	74.6	75.1	76.3	**79.9**
	GoogleNet	84.5	87.3	89.6	**91.6**
A→W	VGG-16	55.9	56.1	59.3	**66.5**
	GoogleNet	67.8	68.4	71.8	**78.9**
C→A	VGG-16	83.1	83.6	84.1	**86.4**
	GoogleNet	91.3	92.2	92.8	**93.6**
C→W	VGG-16	61.4	62.1	64.5	**72.7**
	GoogleNet	73.1	74.6	76.6	**83.5**
W→A	VGG-16	53.1	55.9	57.4	**60.2**
	GoogleNet	66.3	66.9	68.8	**77.8**
W→C	VGG-16	56.8	57.0	58.6	**60.1**
	GoogleNet	67.4	67.5	69.2	**72.9**

We evaluated on an advanced network architecture (GoogleNet [Szegedy, Liu, Jia et al. (2015)]) and on the less advanced VGG-16 network. We experimented with each domain as a source and target domain, and compared the test accuracy of No Data Augmentation (None), Traditional Data Augmentation (Trad), Style Augmentation (Style), and Mixed Augmentation (Mix). Tab. 1 shows the average test accuracy for three experiments on VGG-16 and GoogleNet. Obviously, the mixed augmentation method achieved the highest accuracy in each experiment. The accuracy of the unaugmenteded model is lower than the other three methods, and the style augmentation method is slightly better than the traditional method. When the source domain and the target domain differ greatly (such as *A→W, C→W, W→A, and W→C*), the traditional method does not improve the accuracy of the model obviously, but the style augmentation method has been greatly improved. Even though GoogleNet has achieved good classification results in previous image classifications, it still achieves higher accuracy than the unaugmented and traditional augmentation methods when using style augmentation or mixed augmentation. Therefore, the proposed style transfer method has better generalization ability for invisible domains, and can provide a new scheme for improving classification accuracy on the datasets of the small batch data.

4.3 End-to-End vehicle control experiment based on monocular vision

In image classification and cross-domain classification experiments, we determined both the best style intensity α and the feasibility of the proposed method in the invisible

domain. Next, we designed an end-to-end vehicle control simulation experiment and a real road test experiment based on monocular vision.

4.3.1 Simulation experiment

We tested the performance of the proposed method in end-to-end vehicle control based on monocular vision in three different environments. The experimental environment is the self-driving simulator provided by Udacity and the tracks of two real-world scenarios (Figs. 1(c) and 1(d)). The real-world scenarios shown in Figs. 1(c) and 1(d) (there are significant reflections and light spots on the smooth floor, etc.) poses a major challenge to our approach. Then, two types of data are collected in each experimental environment, one for training the end-to-end model and the other for testing the performance of the model. The convolutional neural networks we use here for training are all based on the architecture of reference [Bojarski, Del Testa, Dworakowski et al. (2016)].

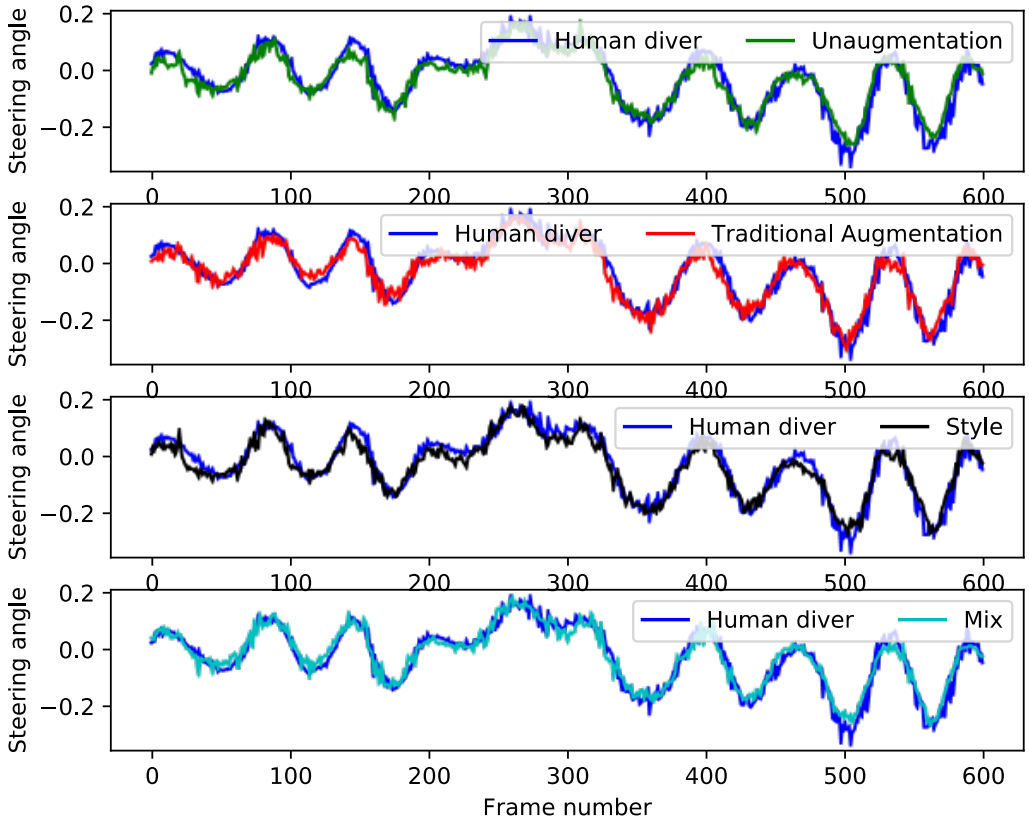


Figure 7: Comparison between steering angle values of human driver and predicted values of different data augmentation models

Fig. 7 shows a comparison between the steering angle values of the human driver and the predicted values of the different data augmentation models. Tab. 2 shows the Mean and MSE between the predicted values of the different data augmentation models and the steering angle values of the human driver. It is worth noting that Mean represents

the average error between the predicted steering angle values and the true values. It can be seen from Fig. 7 and Tab. 2 that the difference in the results of the different tasks is very small. This is because the environment inside the simulator is single and stable, as shown in Fig. 1(a), regardless of which data augmentation methods are applied, the depth model all makes steering decisions by identifying lane lines. The work of Yang et al. [Yang, Wang, Liu et al. (2017)] supports our research. They point out that road-related features are the most important for training controllers. The roadside-related features are helpful to improve the generalization of the controller to the complex roadside information scenarios.

Table 2: Mean and MSE between the results predicted by different data augmentation models and the steering angle values of human drivers

(*a*): Udacity's Self-Driving Simulator Scenario

Task	None	Trad	Style	Mix
Mean	2.555×10^{-2}	2.381×10^{-2}	2.322×10^{-2}	$\mathbf{2.154\times10^{-2}}$
MSE	1.051×10^{-3}	8.671×10^{-4}	8.583×10^{-4}	$\mathbf{7.450\times10^{-4}}$

(*b*): Real-world Scenario 2

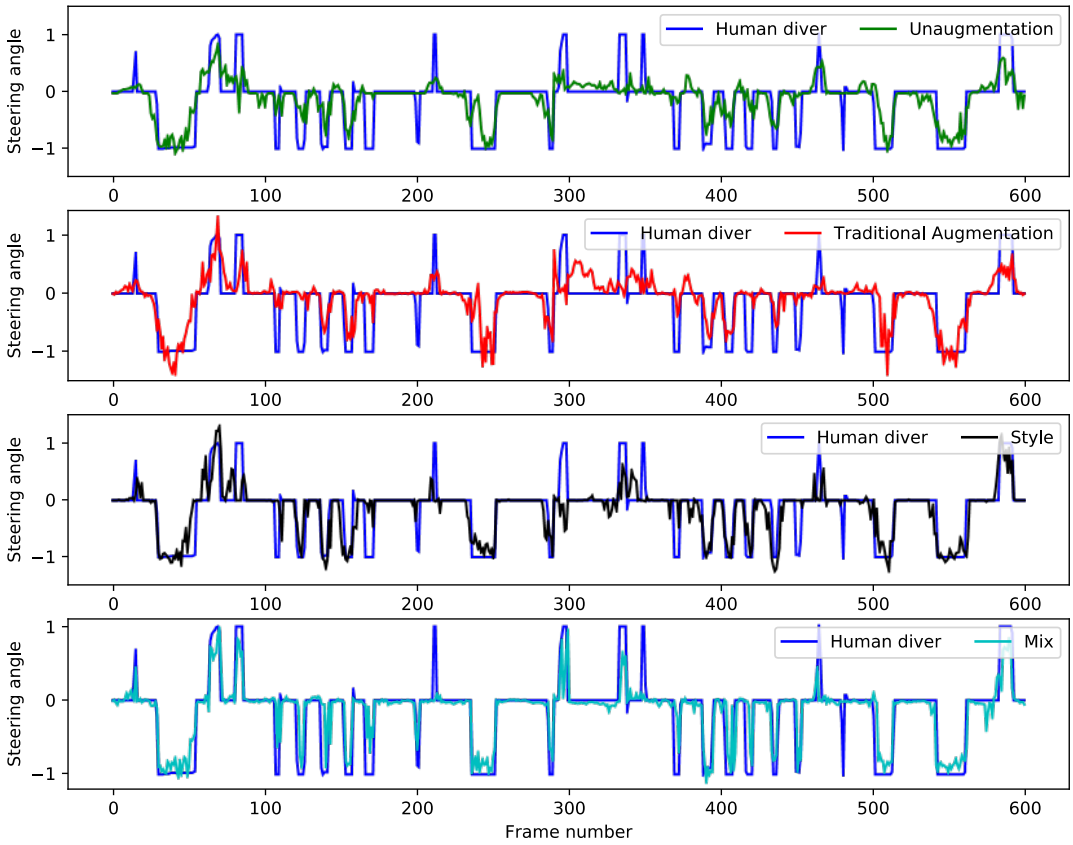
Task	None	Trad	Style	Mix
Mean	2.268×10^{-1}	2.200×10^{-1}	1.735×10^{-1}	$\mathbf{1.387\times10^{-1}}$
MSE	1.327×10^{-1}	1.170×10^{-1}	1.078×10^{-1}	$\mathbf{7.255\times10^{-2}}$

(*c*): Real-world Scenario 3

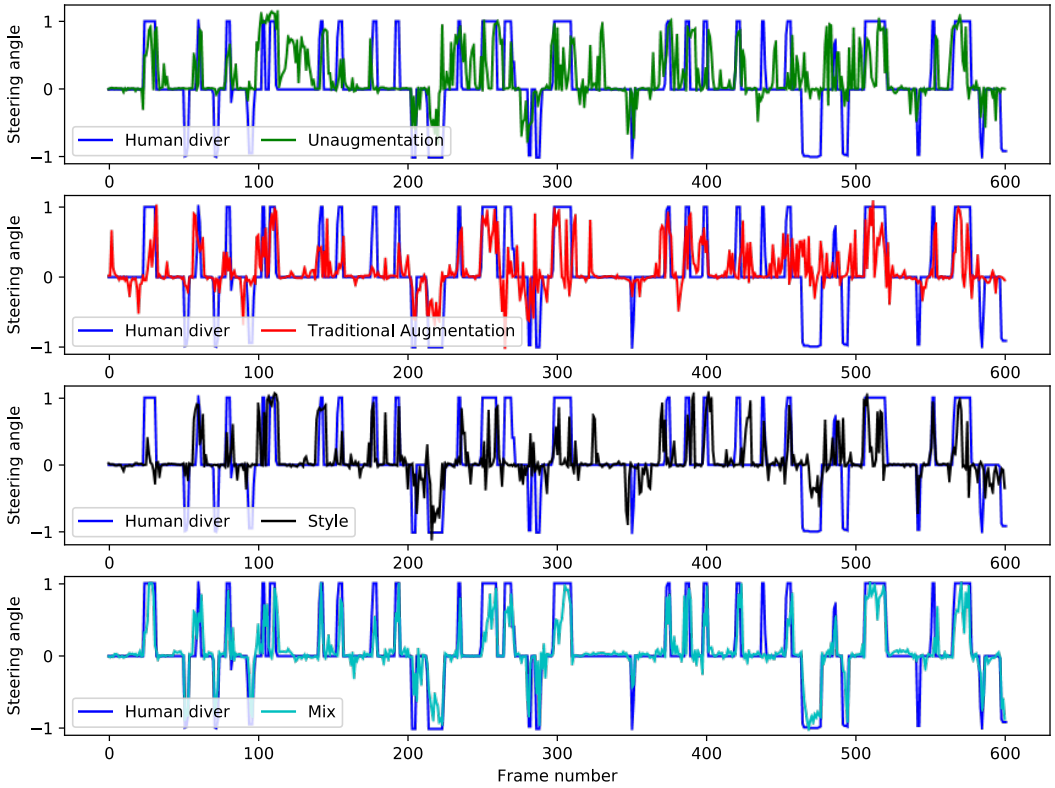
Task	None	Trad	Style	Mix
Mean	3.221×10^{-1}	2.910×10^{-1}	2.527×10^{-1}	$\mathbf{1.542\times10^{-1}}$
MSE	2.633×10^{-1}	2.264×10^{-1}	1.975×10^{-1}	$\mathbf{1.349\times10^{-1}}$

However, in the real-world environment (as shown in Figs. 8(a), 8(b)), the model without data augmentation has a large degree of deviation when going straight, especially in the environment 3 with large interference (Fig. 8(b) shown). The angle values predicted by the model with style augmentation and mixed augmentation are less deviated from the true steering angle. When the vehicle turns, because the left steering angle data in environment 2 is less (the data between [0, 1]), both the unaugmented and traditional augmentation models may exhibit significant understeering or non-steering. When in Scenario 3 (less right turn angle data), this phenomenon is more obvious. When the data transfer method is adopted, only the style-augmented model has a better prediction result, as shown in Fig. 8(a). Although in the Scenario 3, the style augmentation technique has better results than the unaugmented and traditional augmentation techniques, because of its more interference, it eventually leads to the prediction outcome that is lower than the result of the Scenario 2. However, the model with mixed augmentation gets good prediction results. As shown in the figure, the prediction outcome of the vehicle going straight is less fluctuating, and the phenomenon of understeering has also been well solved. Obviously, even if the steering angle data in a certain direction is lacking, the mixed augmentation yields good prediction results. In addition, from the results shown in

Tabs. 2(b) and 2(c), it can be intuitively seen that the "Mix" method is optimal, and the steering angle error and the MSE value are the smallest, then the Mean and MSE of "Style" are better than those of "Trad" and "None".

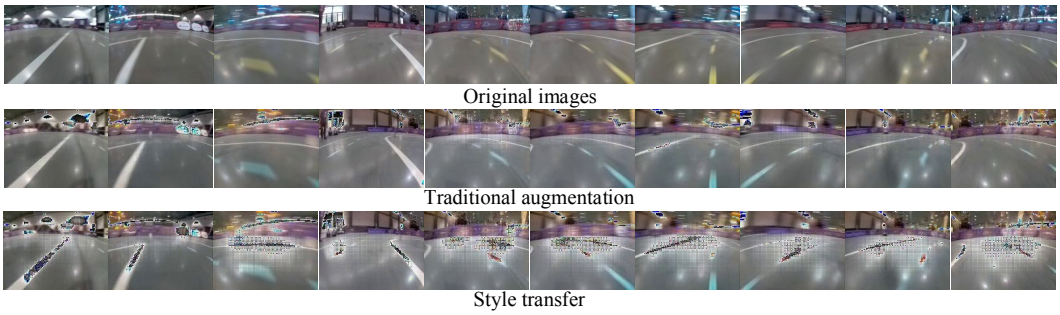


(a): Real-world Scenario 2

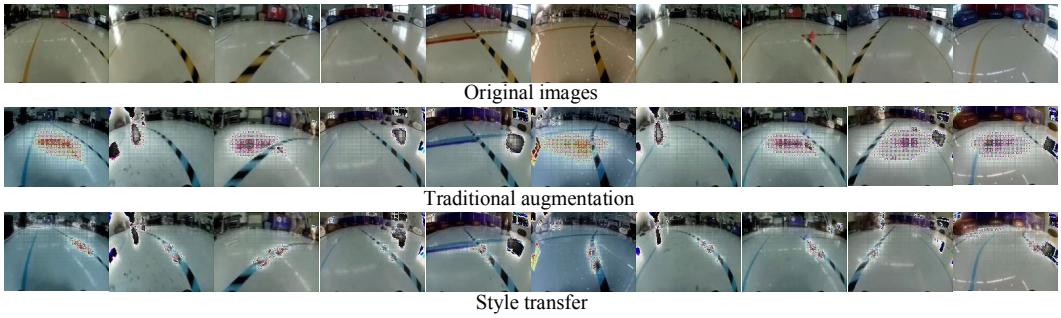


(b): Real-world Scenario 3

Figure 8: Comparison between steering angle values of human driver and predicted values of different data augmentation models



(a) Real-world Scenario 2



(b) Real-world Scenario 3

Figure 9: Visualized heatmap of traditional augmentation and style augmentation techniques

In order to check whether the model trained by the four data augmentation techniques really focuses on the lane lines, we generate a saliency heatmap on the test dataset. Fig. 9 shows the heatmap in two real-world environments. It can be seen from the visualization results that neither the unaugmented model nor the traditional augmentation model makes driving decisions by identifying lane lines. Among them, the model trained in Scenario 2 identifies surrounding objects as well as bright windows. In Scenario 3, the model makes driving decisions by identifying light spots on the smooth floor and bright windows. It is conceivable that when the position of the object and light position or light intensity in the environment change, the model trained by the previous data will lose the identification basis and the robustness of the model will be greatly reduced. However, when the method based on image style transfer is applied to augment the data, the model finally re-identifies the lane lines. In Scenario 3, the style augmentation and mixed augmentation models filter out the obvious light spots on the floor. Although there are particularly bright reflections in the picture and cannot be completely filtered out, Yang et al. [Yang, Wang, Liu et al. (2017)] pointed out in their work that roadside-related features help to improve the generalization of end-to-end models for complex scenarios, and lane lines features are critical to the robustness of the model. In addition, their work also verified that it is feasible to identify only the lane lines model. In addition, their work also verified that the model trained only with lane line data could also achieve great steering angle prediction results. Therefore, the data augmentation technique based on the style transfer drive can refocus the model to the lane lines, which greatly improves the robustness of vehicle control. As the comparison between black and cyan curves and blue curves in Fig. 7 shows, their prediction results are better than those of unaugment and traditional augmentation models.

4.3.2 Road test experiment

To further test the performance of data augmentation technique based on style transfer drive in vehicle control, we built a reduced-size smart car (proportion 1:16) experiment platform. The experiment hardware platform is shown in Fig. 10, including WiFi communication module, workstation, smartphone, game controller X-box and smart car donkeycar based on Raspberry Pi. The WiFi module is used for communication between

smart car, smartphone and workstation. Smartphone are used to control the start, stop, and speed of the vehicle. The X-Box is used to control the driving of the vehicle for data collection. Finally, the collected data is imported from the Raspberry Pi into the workstation, and the neural network training is completed in the workstation.

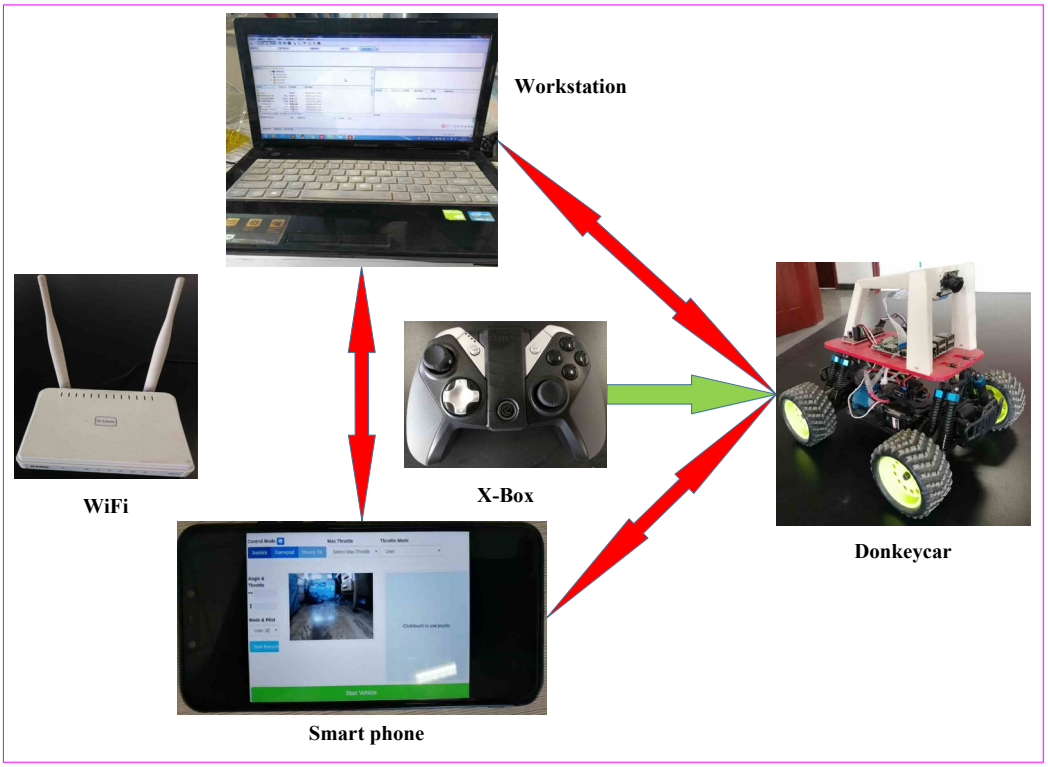


Figure 10: Hardware experiment platform. Its include WiFi communication module, workstation, smartphone, game controller X-box and smart car donkeycar based on *Raspberry Pi*

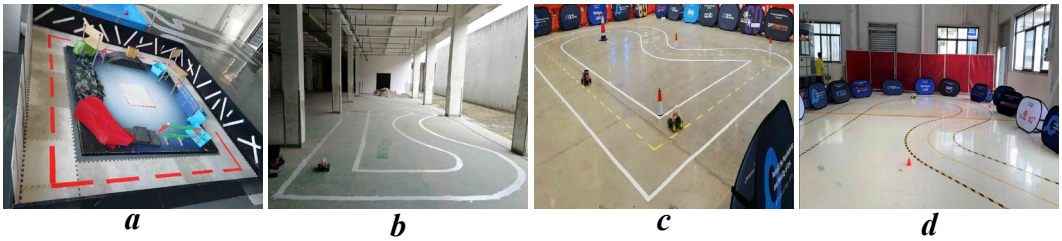


Figure 11: Four different experimental sites

We conducted road test experiments through four different test sites, as shown in Fig. 11. From Site *a* to Site *d*, there are more and more interference factors in the environment. The light and the surrounding objects of Site *a* is relatively fixed. Site *b* has light interference, but the surrounding objects are fixed. Site *c* and Site *d* have more

disturbances, light spots on the floor and changes in the surrounding environment (billboard shifts, walking pedestrians, etc.). Among them, the light interference of the Site *d* is the most serious, which also poses a biggest challenge for our algorithm. In order to visually evaluate the advantages and disadvantages of our proposed method and traditional data augmentation methods, we define two evaluation criteria ourselves:

(1) The smart car drives autonomously 50 times on the track, and runs 2 laps at a time as an experiment. The sign of self-driving success is that the four wheels of the smart car do not completely rush out of the track and do not hit the fence. Then the success rate is defined as $\zeta=n/50$, where n is the number of successes.

(2) After the vehicle has successfully driven autonomously for two laps, let it continue to run and record the total time (in seconds) of the lane keeping. When the car drives autonomously for more than 10 laps, it is recorded as "Finished", which means that the model is very robust (we define it ourselves). Finally, the average lane keeping time is calculated as $\xi=m/n$, where m is the total running time.

Since the vehicle has accumulated error during operation [Chen and Huang (2017)], the purpose of the standard (2) is to observe the error accumulation of the four augmentation techniques. The vehicle speed is the same during the experiment, and the running time of each site is about 7 s, 14.5 s, 17 s and 17 s, respectively. In addition, in order to fully test the robustness of the model, the time of the training model and the experimental time interval are more than 6 hours (for example, data is collected in the morning and the neural network is trained, and the experiment is performed in the afternoon).

Table 3: Smart car self-driving success rate. *N, T, S*, and *M* represent no augmentation, traditional data augmentation, style augmentation, and mixed augmentation, respectively

Site	*a*				*b*				*c*				*d*			
Task	*N*	*T*	*S*	*M*	*N*	*T*	*S*	*M*	*N*	*T*	*S*	*M*	*N*	*T*	*S*	*M*
Unfinished	11	9	9	4	28	19	11	5	32	24	16	9	42	39	18	14
Finished	39	41	41	46	22	31	39	45	18	26	34	41	8	11	32	36
ζ(%)	78	82	82	**92**	55	62	**78**	**90**	36	52	**68**	**82**	16	22	**64**	**72**

Table 4: Average running time for self-driving. It intuitively shows the error accumulation of the model trained by different augmentation techniques. The longer the running time is, the smaller the model error accumulates. "Finished" means that the vehicle can automatically drive 10 laps or more

Site	*a*				*b*			
Task	*N*	*T*	*S*	*M*	*N*	*T*	*S*	*M*
ξ(*s*)	26.6	Finished	Finished	Finished	72.9	99.7	143.3	Finished
Site	*c*				*d*			
Task	*N*	*T*	*S*	*M*	*N*	*T*	*S*	*M*
ξ(*s*)	42.2	59.0	102.0	140.6	35.1	36.7	79.1	111.9

Tab. 3 shows the success rate of self-driving. Although the success rate from the Site *a* to Site *d* is gradually reduced as the interfereence increases, both style augmentation and

mixed techniques yield better results than the other two augmentation techniques. It can be seen from the whole that the success rate is higher in the Site *a* and Site *b* where the environment is relatively stable, and the style augmentation models get better results than the unaugmented and traditional augmentation techniques. Especially when combining traditional methods with style augmentation, the success rate is higher. In the Site *a* where the light and the surrounding objects are relatively fixed, the success rate of the four techniques is high, and the success rate is 92% when the mixed augmentation technique is used. Since the Site *b* and Site *c* have less interference with the Site *d*, we get better results than the Site *d*. Among the sites *d* with the greatest external environment change, the success rate of unaugmented and traditional augmentation techniques is the lowest, with only 12% and 16% accuracy, respectively. And the performance of the traditional data augmentation does not seem to improve much. However, after employing style augmentation technique, the success rate increased to 64%, and the mixed method further made the success rate reach 72%.

From the average time of self-driving maintenance in Tab. 4, it can be seen that the error accumulation of mixed augmentation and style augmentation techniques is lower than that of traditional method. As can be seen in Fig. 3 and Fig. 4, the style transfer technique produces arbitrary stylized images by disturbing the texture, color, and contrast of the image. This technique therefore augments the original image to previously unobserved scenarios. Although this advantage is not obvious in site *a*, only the unaugmented model does not meet the custom criteria 2. However, in the site c and d where there is more interference, the style augmentation and mixed augmentation techniques greatly improve the self-driving time. Although the four augmentation techniques failed to meet the criteria 2, the average lane-keeping time was almost twice that of the other two technologies. This shows that style augmentation technology can significantly reduce the error accumulation phenomenon of the model. In summary, the data augmentation technique based on image style transfer drive is feasible, which can improve the success rate and time of lane keeping, and this advantage is more obvious in the environment with more interference. The mixed augmentation technique is better for reducing the error accumulation of the end-to-end control model.

5 Conclusions

For the first time, we applied image style transfer-driven data technology to self-driving vehicles based on deep learning. In this paper, we propose a novel arbitrary image style transfer algorithm. The style embedding vector is sampled from a multivariate Gaussian distribution and linearly interpolated with the embedded vector predicted on the style learning network, which provides a set of normalization constants for the style transfer network and finally generates arbitrary stylized image. It effectively avoids the high correlation between the traditional image style transfer method and the type, quantity of training data. Next, we also determine the best style intensity α through hyperparameter searches experiments and perform cross-domain experiment on the dataset of small batch data. In the end-to-end vehicle control experiment based on monocular vision, we compare the steering angle from a human driver with the predicted steering angle of four end-to-end models. In the heat map comparison, only our method can identify the lane lines in the environment with more interference. Finally, we have road test experiments

in four different real-world scenarios. The experimental results show that the combination of style augmentation and traditional data augmentation technique can significantly improve the self-driving success rate and reduce the error accumulation of the end-to-end model, which greatly improve the accuracy of steering angle prediction. Therefore, our proposed image style transfer-driven data augmentation technique can be applied to the steering angle prediction of self-driving based on deep learning, which can significantly improve the robustness of the end-to-end model.

Acknowledgement: The authors would like to express gratitude for supporting funding from the National Natural Science Foundation of China (51965008), Science and Technology projects of Guizhou [2018]2168, and Excellent Young Researcher Project of Guizhou [2017]5630.

Conflicts of Interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

Atapour-Abarghouei, A.; Breckon, T. P. (2018): Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2800-2810.

Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B. et al. (2016): End to end learning for self-driving cars. arXiv:1604.07316.

Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. (2015): Deepdriving: learning affordance for direct perception in autonomous driving. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2722-2730.

Claudiu Ciresan, D.; Meier, U.; Gambardella, L. M.; Schmidhuber, J. (2010): Deep big simple neural nets excel on handwritten digit recognition. arXiv:1003.0358.

Chen, Z.; Huang, X. (2017): End-to-end learning for lane keeping of self-driving cars. *IEEE Intelligent Vehicles Symposium*, pp. 1856-1860.

Du, S.; Guo, H.; Simpson, A. (2017): *Self-Driving Car Steering Angle Prediction Based on Image Recognition*. Department of Computer Science, Stanford University, Technical Report. CS231-626.

Dumoulin, V.; Shlens, J.; Kudlur, M. (2016): A learned representation for artistic style. arXiv:1610.07629.

Freeman, J.; Simoncelli, E. P. (2011): Metamers of the ventral stream. *Nature Neuroscience*, vol. 14, no. 9, pp. 1195.

Gatys, L. A.; Ecker, A. S.; Bethge, M. (2016): Image style transfer using convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414-2423.

Gatys, L.; Ecker, A. S.; Bethge, M. (2015): Texture synthesis using convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 262-270.

Ghiasi, G.; Lee, H.; Kudlur, M.; Dumoulin, V.; Shlens, J. (2017): Exploring the

structure of a real-time, arbitrary neural artistic stylization network. arXiv:1705.06830.

Gretton, A.; Smola, A.; Huang, J.; Schmittfull, M.; Borgwardt, K. et al. (2009): Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning*, vol. 3, no. 4, pp. 5.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A. et al. (2018): ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv:1811.12231.

Jackson, P. T.; Atapour-Abarghouei, A.; Bonner, S.; Breckon, T.; Obara, B. (2018): Style augmentation: data augmentation via style randomization. arXiv:1809.05375.

Julesz, B. (1962): Visual pattern discrimination. *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 84-92.

Johnson, J.; Alahi, A.; Li, F. F. (2016): Perceptual losses for real-time style transfer and super-resolution. *European Conference on Computer Vision*, pp. 694-711.

Koutník, J.; Cuccu, G.; Schmidhuber, J.; Gomez, F. (2013): Evolving large-scale neural networks for vision-based reinforcement learning. *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, pp. 1061-1068.

Kim, J.; Park, C. (2017): End-to-end ego lane estimation based on sequential transfer learning for self-driving cars. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 30-38.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 1097-1105.

Li, C.; Wand, M. (2016): Precomputed real-time texture synthesis with markovian generative adversarial networks. *European Conference on Computer Vision*, pp. 702-716.

Muller, U.; Ben, J.; Cosatto, E.; Flepp, B.; Cun, Y. L. (2006): Off-road obstacle avoidance through end-to-end learning. *Advances in Neural Information Processing Systems*, pp. 739-746.

Mahdavian, R.; Martinez, R. D. (2018): Ignition: an end-to-end supervised model for training simulated self-driving vehicles. arXiv:1806.11349.

Pomerleau, D. A. (1989): Alvinn: an autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, pp. 305-313.

Portilla, J.; Simoncelli, E. P. (2000): A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49-70.

Sotelo, M. A.; Rodriguez, F. J.; Magdalena, L.; Bergasa, L. M.; Boquete, L. (2004): A color vision-based lane tracking system for autonomous driving on unmarked roads. *Autonomous Robots*, vol. 16, no. 1, pp. 95-116.

Simonyan, K.; Zisserman, A. (2014): Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. et al. (2015): Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern*

Recognition, pp. 1-9.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. (2016): Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826.

Simard, P. Y.; Steinkraus, D.; Platt, J. C. (2003): Best practices for convolutional neural networks applied to visual document analysis. *Icdar*, vol. 3, no. 2003.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W. et al. (2017): Domain randomization for transferring deep neural networks from simulation to the real world. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23-30.

Urmson, C.; Anhalt, J.; Bagnell, D.; Baker, C.; Bittner, R. et al. (2008): Autonomous driving in urban environments: boss and the urban challenge. *Journal of Field Robotics*, vol. 25, no. 8, pp. 425-466.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; Lempitsky, V. S. (2016): Texture networks: feed-forward synthesis of textures and stylized images. *ICML*, vol. 1, no. 2, pp. 4.

Ulyanov, D.; Vedaldi, A.; Lempitsky, V. (2016): Instance normalization: the missing ingredient for fast stylization. arXiv:1607.08022.

Viswanath, P.; Nagori, S.; Mody, M.; Mathew, M.; Swami, P. (2018): End to end learning based self-driving using JacintoNet. *IEEE 8th International Conference on Consumer Electronics-Berlin*, pp. 1-4.

Xu, H.; Gao, Y.; Yu, F.; Darrell, T. (2017): End-to-end learning of driving models from large-scale video datasets. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2174-2182.

Yanai, K. (2017): Unseen style transfer based on a conditional fast style transfer network. *5th International Conference on Learning Representations.*

Yang, S.; Wang, W.; Liu, C.; Deng, W.; Hedrick, J. K. (2017): Feature analysis and selection for training an end-to-end autonomous vehicle controller using deep learning approach. *IEEE Intelligent Vehicles Symposium*, pp. 1033-1038.