applied sciences

MDPI

Article

HUD-DPCNet: A Joint Learning Framework for Distortion Pre-Correction in AR-HUD Systems

Ying Huang [1], Huaixin Chen [1,*] and Zhixi Wang [2]

[1] School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China; 202322070225@std.uestc.edu.cn
[2] Novel Product R & D Department, Truly OptoElectronics Co., Ltd., Shanwei 516600, China; wangzx.rd@trulyopto.cn
* Correspondence: chenhuaixin@uestc.edu.com

Abstract

As a next-generation automotive display technology, Augmented Reality Head-Up Display (AR-HUD) has demonstrated immense potential in reshaping driving safety and enhancing the human–computer interaction experience. To address the challenges of barrel distortion and perspective distortion inherent in HUD systems, we propose a joint-learning-based dual-path pre-correction method. This approach employs a shared encoder to extract image features, which are then decoupled into two parallel branches: a classification branch and a distortion flow prediction branch. Building upon this architecture, a model-fitting method is introduced to estimate the distortion model parameters in the parameter space using the predicted distortion types and flows, thereby reconstructing a refined distortion flow. Finally, image rectification is achieved through a resampling method. On the ARHDD dataset, the proposed method achieves a PSNR of 24.617 dB (barrel) and 25.062 dB (perspective), an SSIM of 0.845 and 0.873, and an NRMSE of 0.163 and 0.157, respectively. On the Places 365 dataset, it achieves a PSNR of 23.914 dB (barrel) and 21.870 dB (perspective), an SSIM of 0.812 and 0.748, and an NRMSE of 0.174 and 0.211, respectively. Both quantitative and qualitative comparative experiments against other state-of-the-art methods demonstrate that the proposed approach achieves superior correction performance for both types of distortion. Finally, the simulation verification of the HUD system proved that this correction method demonstrated excellent potential, but further verification is still needed in a real or semi-real environment.

Keywords: head-up display; picture generation unit; distortion pre-correction; convolutional neural network

Academic Editor: Douglas O'Shaughnessy

Received: 29 April 2026
Revised: 9 June 2026
Accepted: 18 June 2026
Published: 25 June 2026

1. Introduction

The AR-HUD system projects virtual images onto the driver's eye-box area through a set of precisely designed freeform optical elements. In practical applications, the presented virtual images are prone to geometric distortion—most notably barrel and perspective distortions—due to optical aberrations and the non-ideal curvature of the windshield. Consequently, pre-distortion correction is required to inversely deform the original projected image, effectively compensating for the distortions introduced by the projection equipment and the optical environment.

Extensive research has explored pre-correction techniques based on known parameters and camera models [1–3]. However, acquiring distortion parameters is often challenging, and the correction process is frequently cumbersome, making it difficult to adapt to complex

real-world environments. Literature [4] mentions an automatic calibration template method for parameter acquisition; however, this approach relies heavily on predefined chessboards or other geometric templates, posing significant limitations. Another category of methods extracts explicit texture features from images to estimate the intrinsic and extrinsic camera parameters by establishing pixel-mapping relationships between real and captured textures. For instance, Alvarez et al. [5] proposed an algebraic method to estimate lens distortion parameters. By treating the first four coefficients of a polynomial distortion model as variables, they detected image edges to construct point sets and solved for parameters by minimizing the distance between edge points and ideal lines. Furthermore, based on the theory of concyclic distorted points, Wang et al. [6] achieved image correction by detecting circular arcs formed by distorted lines and calculating the distortion center and coefficients using points on these arcs. While these methods attempt to improve versatility through various algorithms, they remain dependent on specific image textures like straight lines or arcs. When texture features are sparse, the correction performance degrades significantly.

To address the prevalent distortion in HUD projections, we designed a joint-learning dual-path correction network, termed HUD-DPCNet. Unlike the aforementioned methods, the approach proposed in this chapter is based on deep learning technology. Rather than relying solely on feature point detection or known textures, it estimates distortion parameters from global image information and generates the corresponding distortion flow. The contribution of this paper lies in employing a shared encoder to extract features from distorted images, which are then decoupled into two parallel branches: a classification branch and a distortion flow prediction branch. Furthermore, a model fitting method is introduced to reconstruct a more accurate distortion flow, thereby enhancing the precision of pre-correction.

2. Related Work

In the early stages of the development of augmented reality head-up displays (AR-HUDs), distortion correction was primarily achieved by designing two freeform surfaces within the projection system [7–9]. However, the design and manufacturing process of freeform reflective systems is highly complex, involving the precise computation of multiple optical components and surface shapes, which leads to high fabrication costs. There also exist image processing-based methods, which typically map distorted images onto a standard grid and reconstruct image pixels through interpolation. Nevertheless, these methods often struggle to handle complex, nonlinear, or dynamically varying distortions. With breakthroughs in deep learning, particularly convolutional neural networks (CNNs), in the field of image processing, deep learning-based approaches for pre-distortion correction of AR-HUD projection images have gradually become a research hotspot. Deep learning-based distortion correction methods can automatically learn distortion characteristics and their mapping relationships with original images by training on large-scale image datasets. Li et al. [10] proposed an AR-HUD virtual image correction framework based on a multilayer feedforward neural network (MFNN), which predicts the vertex coordinates on the equivalent plane of the AR-HUD virtual image and the pre-distortion map; the pre-distorted image is then projected onto the display to improve the image perceived by the driver. Yu et al. [11] adopted a deep neural network (DNN) to learn distortion features in AR-HUD images, directly training the network model using ideal and distorted point pairs. The network predicts distortion offsets, and based on these offsets, inverse mapping is performed to complete image pre-distortion correction. Although these neural network-based methods are not yet fully mature, they provide new insights and potential solutions for pre-distortion correction of projected images in AR-HUD systems.

3. Method

Inspired by the learning methods in the literature [12,13], predicting the input-to-output mapping is generally simpler than directly predicting the final rectified image. Therefore, the proposed network architecture is designed to learn an intermediate image distortion flow representation. This approach uses the pixel displacement field between the distorted and ground-truth images to represent the distortion relationship, transforming the correction problem into a pixel-level prediction task from the image domain to the flow field domain. Since the distortion flow is a forward mapping from the distorted image to the rectified image, a resampling method is employed to generate the final corrected result.

3.1. Network Architecture

The architecture of HUD-DPCNet is shown in Figure 1, mainly consisting of an encoder, a classification sub-net, a decoder, model fitting, and resampling.

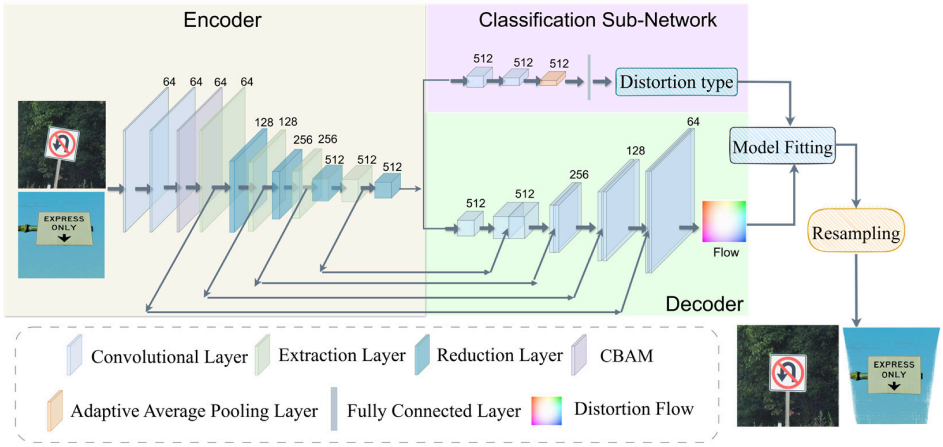


Figure 1. HUD-DPCNet Architecture.

3.1.1. Encoder

The input image first enters the encoder to encode geometric features and capture structural textures. Subsequently, the network splits into two task branches: the first utilizes a classification sub-network for distortion type identification, while the second employs a decoder to predict the distortion flow. This joint learning framework facilitates both the classification of geometric distortions and the refined recovery of the distortion flow.

As illustrated in Figure 1, the Encoder is composed of convolutional layers, extraction layers, reduction layers, and CBAM (Convolutional Block Attention Module). The convolutional layers primarily include convolution operations, Batch Normalization, and ReLU activation functions. During training, the distribution of feature values in each layer shifts as the network depth increases, often approaching the upper and lower limits of the activation function's output range. This phenomenon leads to vanishing gradients, forcing the network to use smaller learning rates and hindering convergence speed. The Batch Normalization layer realigns the feature distribution to a standard normal distribution, ensuring values fall within the sensitive region of the activation function, thereby effectively mitigating vanishing gradients and accelerating convergence [14,15].

The primary function of the extraction layer is to extract features from the distorted images. As shown in Figure 2, this layer applies different convolutional kernels across multiple branches, finally merging the feature maps from each branch via a Concat operation. This structure moderately expands the network width, thereby enhancing performance [16]. Simultaneously, the use of multi-scale kernels enables the network to focus on image features across different scales, improving versatility. The leftmost branch utilizes only

a 1×1 kernel; while it adds no significant computational cost, it allows shallow features to pass into deeper paths with minimal computation, facilitating feature reuse. This is crucial for enhancing information flow between layers, slowing gradient vanishing, and improving the network's representational capacity [17–19].

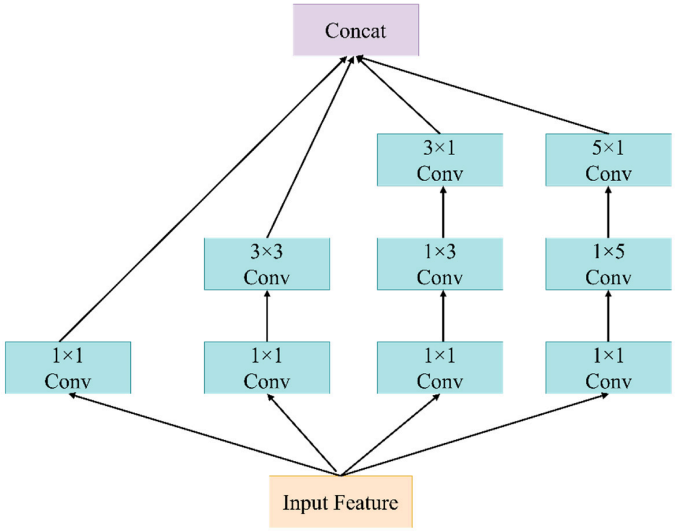


Figure 2. Structure of the Extraction Layer.

Additionally, the use of asymmetric convolution (AC) kernels, such as 1×3, 3×1, 1×5, and 5×1, helps the network better capture locally salient features [20], such as horizontal and vertical edge textures. These kernels strengthen the role of local pixels, improving the network's universality in implicitly learning texture features—an idea analogous to traditional geometric correction methods [21].

The Reduction Layer shares a similar structure with the extraction layer, as shown in Figure 3. Its main function is to gradually reduce the spatial dimensions of the feature maps while increasing the number of channels. While striving to preserve texture information, 2×2 and 4×4 Max Pooling operations are used to lower resolution, reduce redundancy, and decrease computational costs while expanding the receptive field. Acting as a bridge between extraction layers and subsequent components, it also increases the channel count. Since HUD-DPCNet primarily uses the ReLU activation function—which can lead to information loss in low-dimensional data—increasing dimensionality enhances the non-linear effects, allowing subsequent layers to learn richer features in a high-dimensional space [22].

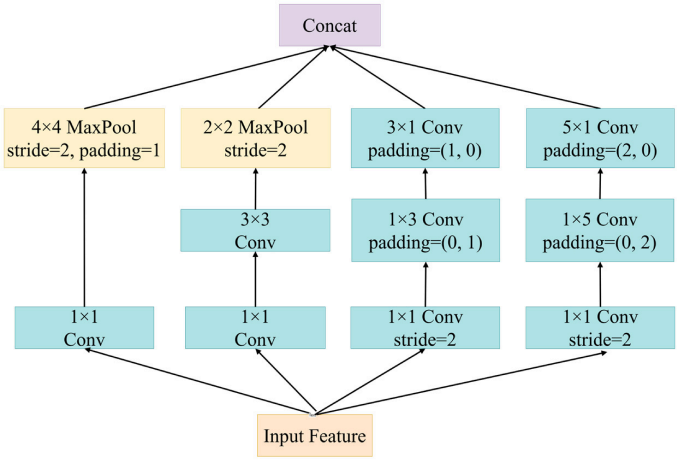


Figure 3. Structure of the Reduction Layer.

Attention mechanisms play a vital role in human perception and have been widely applied in CNNs to enhance representational power [23–28]. The CBAM proposed by Woo et al. [29] learns "what to emphasize" and "where to emphasize," effectively refining intermediate features. Experiments have confirmed that CBAM outperforms baseline models on datasets such as ImageNet-1K, MS COCO, and VOC 2007.

As shown in Figure 4 below, CBAM is an efficient attention mechanism that adaptively adjusts feature map weights across both channel and spatial dimensions, helping the network focus on key regions and significant channel features. In distortion correction, distortions often involve complex spatial and channel variations that a single convolution may struggle to capture. By introducing channel and spatial attention, the network automatically suppresses irrelevant features and strengthens regions related to distortion correction. To further enhance distortion feature extraction, this paper introduces the CBAM module between the first dimensionality reduction layer and the extraction layer.

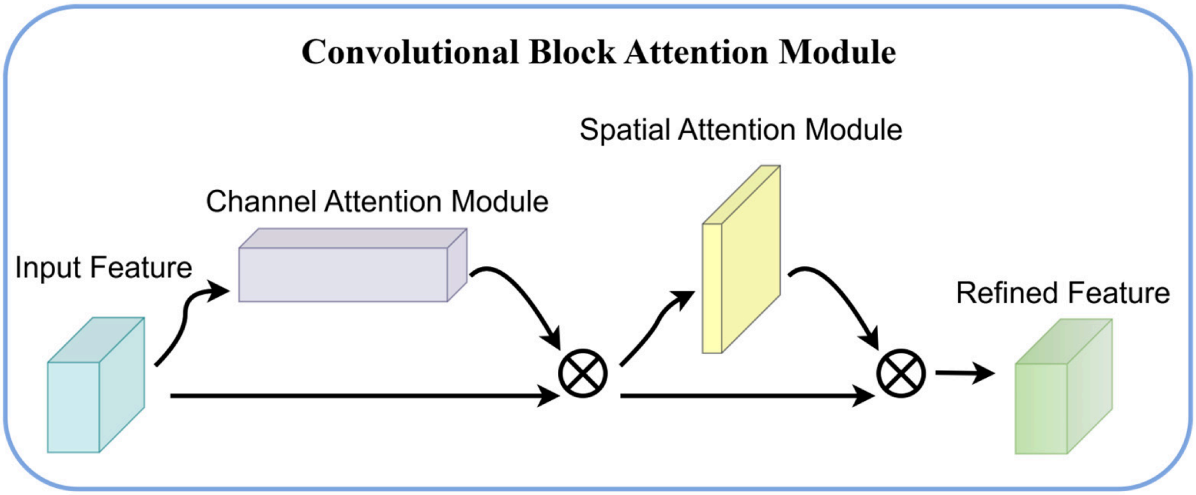


Figure 4. CBAM Module.

3.1.2. Classification Sub-Network

To enhance the discriminative power of distortion features, the deep-level output of the encoder is fed into a classification sub-network. This branch first employs two convolutional layers to further compress the dimensions of the feature maps. Subsequently, an adaptive average pooling layer is utilized to transform the $8 \times 8 \times 512$ three-dimensional feature tensor into a $4 \times 4 \times 512$ representation. Finally, a fully connected layer converts the feature maps into a 1D score vector representing the probability for each category. Since both task branches share a common encoder, this joint learning strategy enables the classification branch to assist the encoder in learning more effective geometric features tailored to different distortion types.

3.1.3. Decoder

The decoder restores high-dimensional abstract features into a 2D prediction map with spatial dimensions identical to the input image. Upsampling of spatial resolution is achieved via transposed convolutions with a stride of 2. To mitigate the loss of spatial details inherent in deep feature maps, skip connections are integrated into the network architecture. These connections facilitate the channel-wise fusion of local features generated at specific encoder levels with the corresponding upsampled features in the decoder. This feature reuse mechanism allows the decoder to leverage shallow-level edge information to refine the boundaries of the distortion flow, thereby maintaining pixel-level correction accuracy.

3.2. Model Fitting

The Hough Transform [30–34] is a technique widely applied for image feature extraction. HUD-DPCNet can predict the distortion type and regress the distortion flow from the input image in parallel; thus, the actual parameters of the distortion model can be estimated via the Hough Transform to regenerate a more precise distortion flow. Each distortion type β is associated with a model M_β, which defines the mapping from the distorted image to the distortion-free image. When generating the distortion flow $F = M_\beta(\rho_\beta)$, it is equivalent to recording how the pixels in the distorted image are moved to their corresponding points in the original distortion-free image, where ρ_β is the distortion parameter that controls the degree of distortion.

For an input image I, given the distortion type β predicted by the network and the corresponding distortion flow field N, the associated distortion model M_β and its distortion parameter ρ_β need to be fitted. In this section, data points N_{ij} at coordinates (i, j) in the flow field N can be mapped into the distortion parameter space. The transformation formula is shown in Equation (1) below:

$$\rho_{ij} = M^{-1}(N_{ij}) \tag{1}$$

The range of the distortion parameter ρ is set as $(\rho_{\min}, \rho_{\max})$, which is uniformly divided into m grid cells. All points ρ_{ij} are assigned to their corresponding cells based on their values. The grid cell that receives the highest number of votes—meaning the highest count—determines the optimal fitting result. The final value of the distortion parameter ρ is taken as the average of all points within that specific cell. The value of m directly determines the quantization resolution of the Hough voting space. If m is too small, the grid division becomes overly coarse, leading to significant errors in the fitted parameters and making it difficult to capture subtle geometric distortions. If m is too large, although the resolution increases, the voting distribution becomes too sparse, making it susceptible to interference from noisy points in the predicted flow field and causing misjudgments. Moreover, a finer grid discretization incurs higher computational and memory overhead. In the experiment, m is set to 100. Once the model fitting is complete, a refined and smooth reconstructed flow field $F = M_\beta(\rho_\beta)$ can be obtained.

3.2.1. Barrel Distortion

The main cause of barrel distortion is the lower magnification at the edges of the field of view compared to the center, resulting in radial displacement of image pixels and thus image distortion. It can be described using a standard radial distortion model.

$$\begin{pmatrix} x_d - x_c \\ y_d - y_c \end{pmatrix} = L(r) \begin{pmatrix} x_o - x_c \\ y_o - y_c \end{pmatrix} \tag{2}$$

where (x_d, y_d) are the pixel coordinates after distortion, (x_o, y_o) are the pixel coordinates before distortion (after correction), and (x_c, y_c) is the center of the camera distortion model, usually the center of the image. r is the Euclidean distance from the pixel to the center of the distortion model (x_c, y_c), as shown in Equation (3), and $L(r)$ is the amplification function as shown in Equation (4):

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{3}$$

$$L(r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots + k_n r^{2n} \tag{4}$$

where k_i is the barrel distortion coefficient ($k_i < 0$, and the larger the absolute value, the more severe the distortion). Taking the first distortion coefficient k_1 for correction is sufficient to meet the basic correction requirements. In order to reduce the amount of computation, our network only performs regression on the first distortion coefficient k_1.

3.2.2. Perspective Distortion

Representing the affine transformation in the form of matrix multiplication yields:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} A_{2\times2} & T_{2\times1} \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{5}$$

where $A_{2\times2}$ can be any 2×2 matrix. It can be observed that, in addition to altering the position of the target, an affine transformation also changes its shape while preserving the straightness of objects—that is, two parallel lines in the figure remain parallel after the transformation.

Similarly, the perspective transformation is expressed as:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} A_{2\times2} & T_{2\times1} \\ V^T & s \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H_{3\times3} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{6}$$

where $A_{2\times2}$ represents the affine transformation parameters in Equation (5); $T_{2\times1}$ represents the translation parameters; and $V^T = [v_1, v_2]$ denotes a relationship concerning the intersection of edges after the transformation, as illustrated in Figure 5.

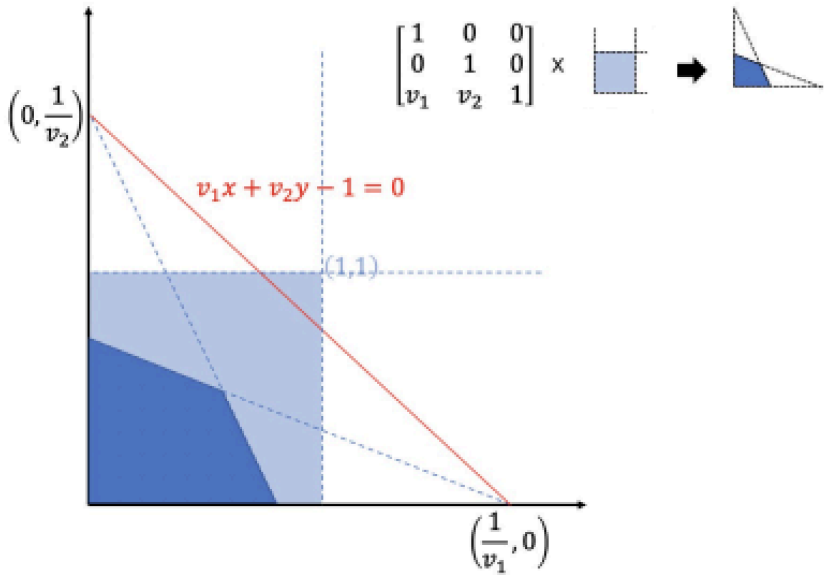


Figure 5. An example of perspective transformation when $A_{2\times2}$ is the identity matrix.

Here, s is a scaling factor associated with $V^T = [v_1, v_2]$, and in general, normalization is applied to set s = 1.

From the above analysis, it follows that a point on Figure 6a can be mapped via the perspective transformation to the corresponding position in Figure 6b, i.e., it satisfies:

$$\begin{pmatrix} x_l \\ y_l \\ 1 \end{pmatrix} = \begin{pmatrix} A_{2\times2} & T_{2\times1} \\ V^T & s \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = H_{3\times3} \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} \tag{7}$$

where (x_l, y_l) is a point in Figure 6a and (x_r, y_r) is its corresponding point in Figure 6b. Therefore, the key to solving the distortion flow corresponding to perspective distortion lies in determining this homography matrix $H_{3\times3}$.

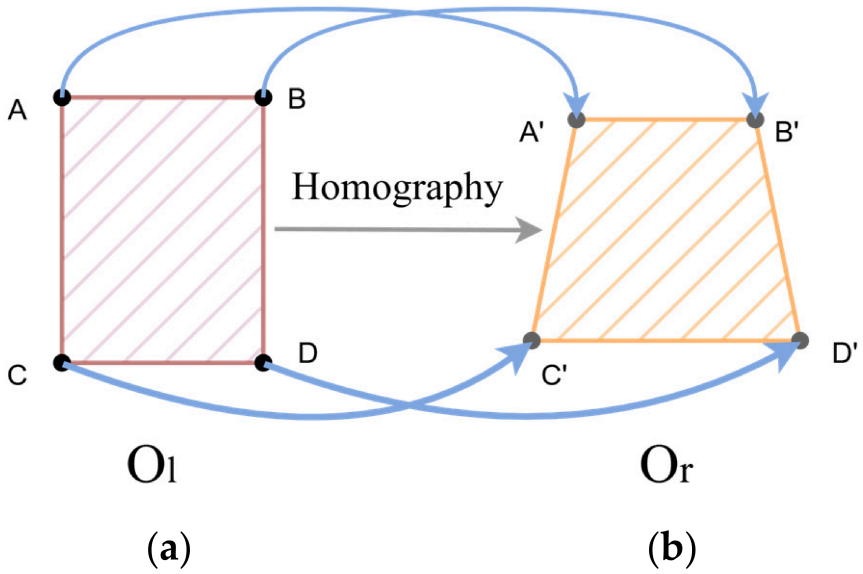


Figure 6. Perspective transformation. (**a**) Before transformation; (**b**) after transformation.

3.3. Loss Function

In the distortion flow regression branch, the Endpoint Error (EPE) loss L_{Flow} is adopted. EPE is commonly used to measure the accuracy of optical flow estimation in research areas such as autonomous driving, and video editing. It is defined as the average sum of the Euclidean distances between the predicted optical flow vectors and the ground-truth optical flow vectors, where a smaller EPE indicates a more accurate prediction, as shown in Equation (8).

$$EPE = \left((u_p - u_r)^2 + (v_p - v_r)^2\right)^{1/2} \tag{8}$$

where u_p and v_p denote the predicted pixel-flow components in the x- and y-directions, respectively, and u_r and v_r denote the ground-truth pixel-flow components in the x- and y-directions.

In the classification branch, the Cross-Entropy Loss, which is most commonly used in classification tasks [35,36], is employed as shown in Equation (9):

$$L_{CE} = -\sum_i l_i \log p_i, \tag{9}$$

where l_i is the one-hot encoding of the i-th label, and p_i is the probability predicted by the network that the sample belongs to the i-th class. By minimizing the total loss function, joint training and optimization are performed for these two branches:

$$\theta = \arg\min_{\theta}(L_{Flow} + \lambda L_{class}), \tag{10}$$

where the weight coefficient λ is used to balance the flow field prediction task and the distortion type classification task. On the validation set, a grid search was performed over $\lambda \in \{0.01, 0.05, 0.1, 0.5, 1.0\}$, using EPE and classification accuracy as evaluation metrics. The results show that when $\lambda = 0.5$, the performance of the two branches achieves a good balance, with flow prediction accuracy slightly better than other values and classification accuracy maintained above 95%. Joint learning helps reduce the distortion flow prediction error. Since the two branches share the same encoder, the classification branch assists the encoder in better learning the geometric features specific to different distortion types.

4. Experiments and Analysis

4.1. Dataset

Since datasets specifically for barrel and perspective distortions are extremely scarce, we adopted a self-synthetic dataset approach. InteriorNet [37] is a high-quality dataset containing diverse interior scenes; from it, we randomly selected 10,000 images. After preprocessing, these images were warped using random parameters via specified distortion mappings to construct a distorted image dataset and its corresponding distortion flow dataset for pre-training.

Places 365 [38], a large-scale multi-scene dataset released by MIT, contains a vast number of architectural and outdoor scenes. We selected 2200 images from this collection, with 2000 used for training and 200 for testing. Following preprocessing, the images were warped using barrel and perspective distortion mappings with random parameters to generate distorted images and their corresponding .flow files.

Furthermore, we collected images of traffic signs—such as speed limits and road signs—within typical driving scenarios, as well as HUD projection elements including driving status, vehicle speed prompts, and navigation turn indicators. All images were uniformly resized. A portion of these images was warped using random parameters via given distortion mappings to generate images with varying degrees of distortion. The remaining images were imported into an HUD system model constructed in Zemax OpticStudio 2022 R2.02 software, where ray-tracing algorithms were employed to simulate imaging distortion effects at a projection distance of 7.5 m. Consequently, we constructed the AR-HUD Distortion Dataset (ARHDD), which includes distorted images and distortion flows. The training set comprises 3000 image-flow pairs, while the test set contains 300 pairs. This dataset covers a rich array of elements such as traffic signs and navigation information, providing reliable data support for subsequent model training and validation.

4.2. Experimental Settings

The proposed HUD-DPCNet was implemented using the PyTorch 1.9.1 framework on a Linux operating system. The hardware configuration consisted of an Intel (R) Xeon (R) CPU E5-2620 v4 @ 2.10 GHz processor, 32 GB of RAM, and an NVIDIA TITAN RTX GPU with 24 GB of VRAM. For model optimization, the Adam optimizer [39] was employed, incorporating an L2 regularization penalty to prevent overfitting. The initial learning rate was set to 10^{-3}, with a decay period of 8 epochs and a decay coefficient of 0.5. The batch size was configured to 8, and the model was trained for 100 epochs.

4.3. Model Performance Evaluation

4.3.1. Experiments on the ARHDD Dataset

(1) Quantitative Comparison

To evaluate the distortion correction performance of the proposed method in automotive application scenarios, this section compares HUD-DPCNet with several state-of-the-art methods designed for specific types of distortion. Among them, DR-GAN [40], PCN [41], DeepCalib [42], and GeoNetS [43] are capable of rectifying barrel distortion, while the method proposed by Krishnendu [44] and HomographyNet [45] are designed for perspective distortion correction. Table 1 presents the quantitative comparison results of the aforementioned methods on the ARHDD dataset.

As shown in Table 1, HUD-DPCNet outperforms all comparative methods across all metrics on the ARHDD dataset, clearly demonstrating the effectiveness of the proposed approach. In barrel distortion correction the three metrics for HUD-DPCNet—PSNR, SSIM, and NRMSE—reach 24.617 dB, 0.845, and 0.163, respectively. In perspective distortion correction, the corresponding metrics are 25.062 dB, 0.873, and 0.157.

Table 1. Quantitative Comparison Results of Different Methods on the ARHDD Dataset. ↑ indicates that a higher metric value signifies better performance; ↓ indicates that a lower metric value signifies better performance. The best results are underlined.

Methods	Distortion Type	PSNR (dB) ↑	SSIM ↑	NRMSE ↓
DR-GAN	Barrel	15.231	0.452	0.396
PCN	Barrel	23.360	0.819	0.167
DeepCalib	Barrel	13.835	0.411	0.583
GeoNetS	Barrel	20.107	0.719	0.213
HUD-DPCNet	Barrel	<u>24.617</u>	<u>0.845</u>	<u>0.163</u>
Krishnendu	Perspective	21.184	0.705	0.216
HomographyNet	Perspective	19.251	0.658	0.281
HUD-DPCNet	Perspective	<u>25.062</u>	<u>0.873</u>	<u>0.157</u>

(2) Qualitative Comparison

In this section, a visual comparative experiment of the aforementioned methods is conducted on the ARHDD dataset. Figure 7 illustrates the correction performance of DR-GAN [40], PCN [41], DeepCalib [42], GeoNetS [43] and HUD-DPCNet on barrel distortion using the ARHDD dataset.

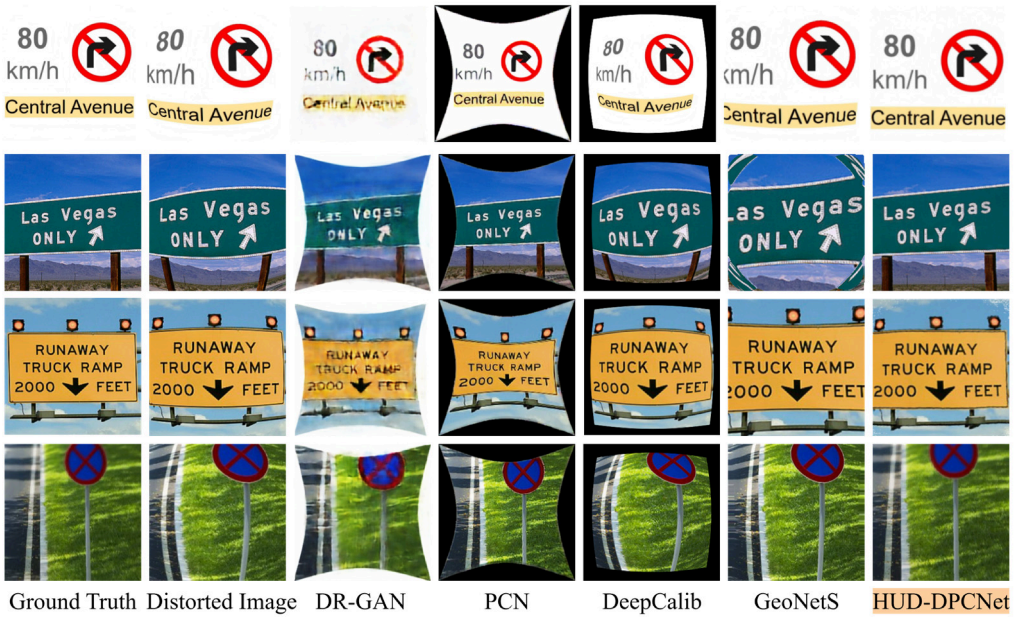


Figure 7. Correction performance of different methods on barrel distortion images in the ARHDD dataset.

As illustrated in Figure 7, DeepCalib [42] exhibits poor correction performance on the distorted images. GeoNetS [43] demonstrates a certain degree of rectification capability, with improvements in both the expansion effect and line curvature; however, the distortions are not fully resolved. PCN [41] generally achieves satisfactory barrel distortion correction, but it suffers from over-correction in the fifth row of images, causing the road signs to shrink inward. The primary issue with DR-GAN [40] lies in the color distortion and blurring effects of the reconstructed images, which make critical information difficult to distinguish. Compared with other methods, our proposed HUD-DPCNet achieves the best performance on the ARHDD dataset, with the rectified images being most consistent with the ground truth and exhibiting more reasonable detail preservation.

Figure 8 shows the correction performance of the method proposed by Krishnendu [44], HomographyNet [45] and HUD-DPCNet on perspective distortion using the ARHDD dataset.

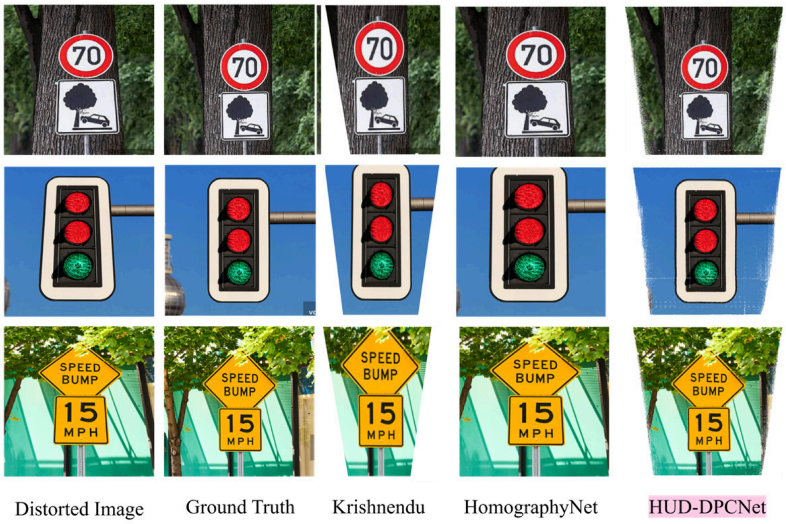


Figure 8. Correction performance of different methods on perspective distortion images in the ARHDD dataset.

It can be observed that the images rectified by HUD-DPCNet are more consistent with the ground truth, whereas the correction performance of Krishnendu [44] and HomographyNet [45] on perspective distortion is limited. While Krishnendu [44] can restore the distorted trapezoidal road signs and traffic light frames into rectangular shapes, it inevitably introduces stretching artifacts. Meanwhile, HomographyNet [45] shows no noticeable correction effect on the navigation information in the first row and fails to fully rectify the trapezoidal signs into rectangles in the second, third, and fourth rows.

4.3.2. Experiments on the Places 365 Dataset

(1) Quantitative Comparison

To further evaluate the performance of the proposed method, this section conducts comparative experiments between HUD-DPCNet and the aforementioned distortion-specific correction methods using the Places 365 dataset. Table 2 presents the quantitative comparison results of these methods on the Places 365 dataset.

Table 2. Quantitative comparison results of different methods on the Places 365 dataset. The best results are underlined. ↑ indicates that a higher metric value signifies better performance; ↓ indicates that a lower metric value signifies better performance. The best results are underlined.

Methods	Distortion Type	PSNR (dB) ↑	SSIM ↑	NRMSE ↓
DR-GAN	Barrel	16.416	0.547	0.379
PCN	Barrel	23.617	<u>0.826</u>	0.182
DeepCalib	Barrel	13.429	0.370	0.605
GeoNetS	Barrel	19.685	0.647	0.221
HUD-DPCNet	Barrel	<u>23.914</u>	0.812	<u>0.174</u>
Krishnendu	Perspective	19.013	0.632	0.282
HomographyNet	Perspective	18.692	0.658	0.274
HUD-DPCNet	Perspective	<u>21.870</u>	<u>0.748</u>	<u>0.211</u>

As indicated in Table 2, HUD-DPCNet demonstrates superior performance on the Places 365 dataset. In barrel distortion correction, HUD-DPCNet achieves the best results in terms of PSNR and NRMSE, reaching 23.914 dB and 0.174, respectively; its SSIM reaches 0.812, which is second only to PCN's 0.826. In perspective distortion correction, HUD-DPCNet outperforms the comparative methods across all three metrics, with PSNR, SSIM, and NRMSE values of 21.870 dB, 0.748, and 0.211, respectively.

(2) Qualitative Comparison

To evaluate the performance of HUD-DPCNet more intuitively, this section conducts a visual comparative analysis of the various methods mentioned above using the Places 365 dataset. Figures 9 and 10 illustrate the correction performance of the various methods mentioned above on barrel distortion and perspective distortion, respectively, using the Places 365 dataset.

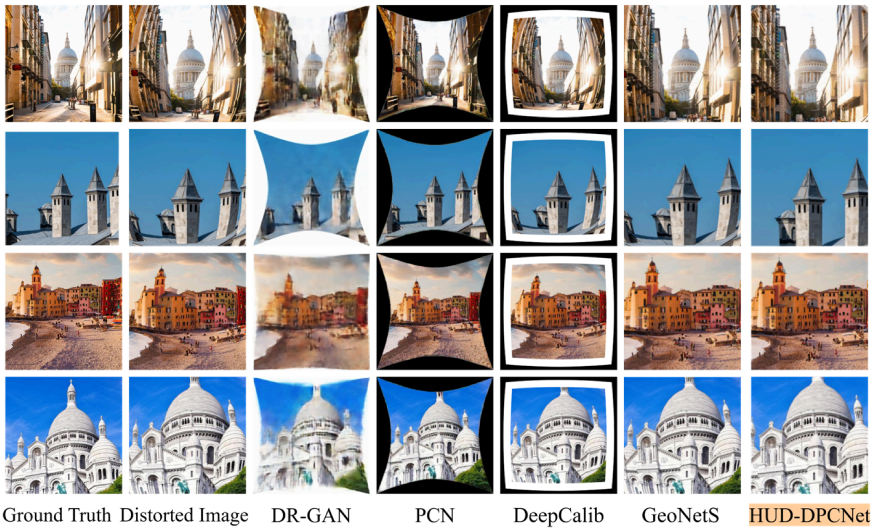


Figure 9. Correction performance of different methods on barrel distortion images in the Places 365 dataset.

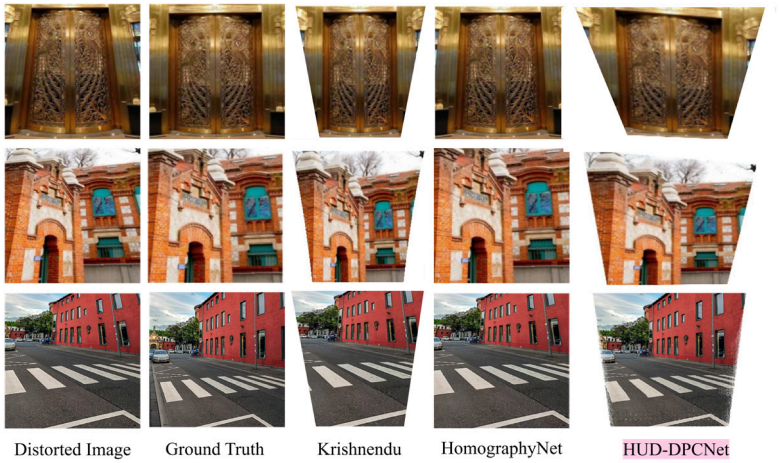


Figure 10. Correction performance of different methods on perspective distortion images in the Places 365 dataset.

As can be observed from Figure 9, DeepCalib [42] does not exhibit any noticeable correction effect. GeoNetS [43] fails to fully rectify the barrel distortion in the first row of images and shows negligible improvement in the second and fourth rows. Although DR-GAN [40] corrects the barrel distortion across all four images, its reconstructed results

suffer from color distortion and degraded clarity. Overall, PCN [41] and HUD-DPCNet demonstrate the most significant performance in handling the distortions of the four images, appearing most consistent with the ground truth as a whole.

Figure 10 illustrates the correction performance of HUD-DPCNet, the method proposed by Krishnendu [44], and HomographyNet [45] on perspective distortion using the Places 365 dataset. By observing the three rows of images in Figure 8, it is evident that the method by Krishnendu [44] exhibits a significant rectification effect on perspective distortion, such as restoring the trapezoidal distorted buildings in the first and second rows to rectangular structures. However, this method causes a reduction in the overall dimensions of the image during the correction process. HomographyNet [45] shows limited correction on the first and third rows of images, where the buildings still retain a certain degree of perspective distortion. In contrast, the images rectified by HUD-DPCNet are most consistent with the ground truth, demonstrating its superiority in perspective distortion correction.

4.3.3. Component Effectiveness Analysis

In this section, a comparative analysis of different component configurations is conducted to verify the effectiveness of individual modules through ablation studies on the ARHDD dataset. Table 3 presents the quantitative comparison results of several HUD-DPCNet variants on the ARHDD dataset, including three configurations: w/o Classification (without the classification branch), w/o Model Fitting (denoted as w/o Hough), and the final HUD-DPCNet model. The experimental results demonstrate that HUD-DPCNet achieves superior performance across all metrics, including EPE, PSNR, SSIM, and NRMSE, which clearly validates the effectiveness of the proposed components.

Table 3. Quantitative comparison results of different component configurations. "w/o" indicates that the corresponding component is not incorporated; ↑ indicates that a higher metric value signifies better performance; ↓ indicates that a lower metric value signifies better performance. The best results are underlined.

Model	Barrel				Perspective			
	EPE ↓	PSNR ↑	SSIM ↑	NRMSE ↓	EPE ↓	PSNR ↑	SSIM ↑	NRMSE ↓
w/o Classification	2.72	20.291	0.789	0.175	3.15	21.550	0.764	0.223
w/o Hough	1.87	23.819	0.816	0.171	2.97	23.026	0.814	0.163
HUD-DPCNet (Ours)	<u>1.57</u>	<u>24.617</u>	<u>0.845</u>	<u>0.163</u>	<u>2.25</u>	<u>25.062</u>	<u>0.873</u>	<u>0.157</u>

Specifically, EPE represents the average Euclidean distance between the predicted flow vectors and the ground-truth flow vectors; thus, a lower EPE indicates higher accuracy in flow prediction. A quantitative comparison between the w/o Classification variant and the final model reveals that on this multi-type distortion dataset, the joint learning framework incorporating a classification branch yields more accurate distortion flow predictions than the model without classification training. Detailed analysis suggests that since the classification and distortion flow prediction branches share a common encoder, the classification branch assists the encoder in learning more robust geometric features specific to different distortion types, thereby enhancing prediction precision and correction performance.

Furthermore, this section examines whether the model-fitting method improves the flow prediction accuracy of HUD-DPCNet. The data in the table show that the model-fitting method based on the Hough transform provides more accurate predicted flows and results in better image quality evaluation metrics for the rectified images. The model-fitting

approach facilitates the estimation of actual distortion parameters and regenerates a more precise flow field, ultimately leading to superior results.

4.3.4. HUD Simulation and Verification

An AR-HUD projection system is constructed using Zemax OpticStudio 2022 R2.02 software. The specific design parameters and specifications for the HUD system are presented in Table 4 below:

Table 4. Key Parameters and Specifications of the HUD System.

HUD System Parameters	Value
Operating Wavelength	486–656 nm
Virtual Image Distance	7.5 m
Field of View	10° × 5°
Eyebox	130 mm × 5 mm

The AR-HUD optical projection system investigated in this paper is designed based on an off-axis triple-reflective system [46,47]. It primarily consists of a picture generation unit (PGU), a folding mirror (M1), a magnifying mirror (M2), and the windshield. Together, these components constitute the display optical path from the image source to the eyebox. The 3D entity model in Zemax is illustrated in Figure 11 below.

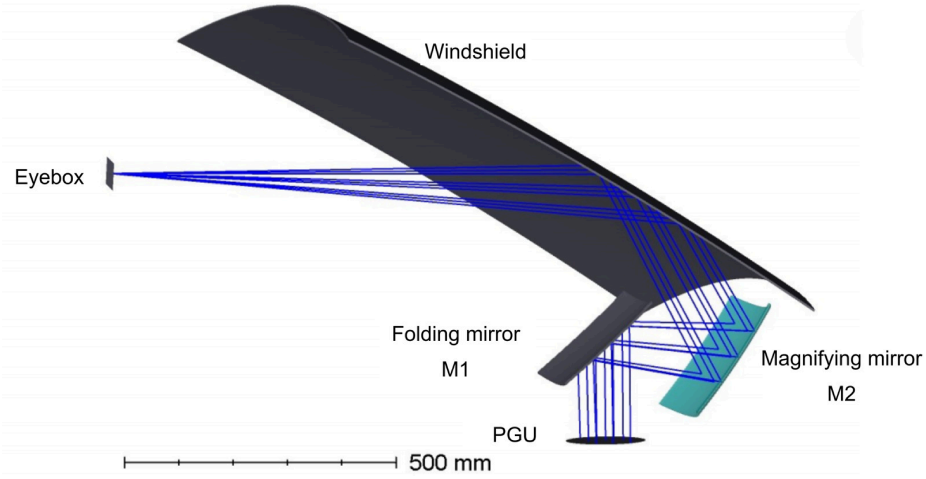
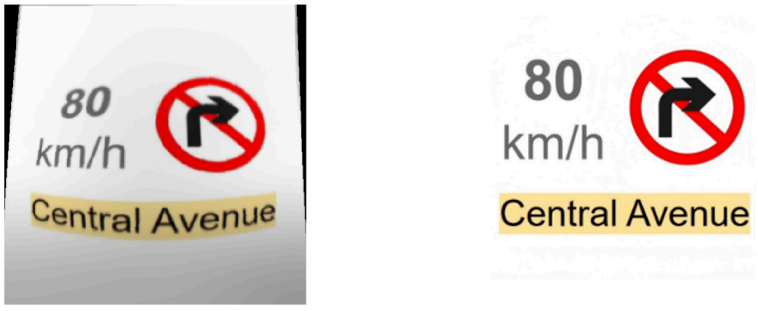


Figure 11. HUD 3D Model.

Utilizing the Extended Image Analysis feature within the image quality analysis module of Zemax, Figure 12 displays the simulated projection result of the navigation information through this AR-HUD system, alongside the original image.



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

Figure 12. AR-HUD Navigation Information Graphics. (**a**) Projection rendering; (**b**) original image.

As can be observed from the simulation results in Figure 12, the HUD system introduces composite optical distortion characteristics, specifically manifested as the superposition of barrel distortion and perspective distortion. For instance, the white rectangular area is warped into the trapezoid shown in Figure 12a upon projection, which reflects the inherent properties of perspective distortion; meanwhile, the elements in the image also exhibit an outward expansion phenomenon, conforming to the typical characteristics of barrel distortion. At this stage, by taking the distorted image generated from the Zemax AR-HUD simulation as input for HUD-DPCNet, the network can regress the distortion coefficients and generate the distortion flow. To obtain the pre-corrected image for this AR-HUD system, the original navigation image is resampled based on the regressed distortion flow. Figure 13 illustrates the results of the navigation information image after this pre-correction processing.

Figure 13. Pre-corrected image.

After obtaining the aforementioned HUD pre-corrected image, the image analysis function in Zemax was restarted to verify the projection results. The simulated effect after system projection is shown in Figure 14.

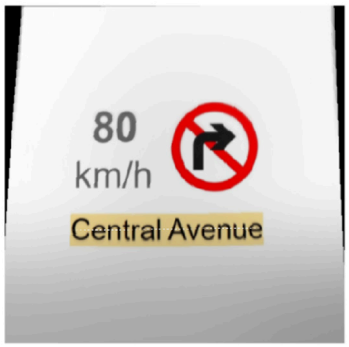


Figure 14. Projection Effect After Pre-calibration.

Compared with Figure 12a, the projection effect shown in Figure 14 is significantly improved, with core information such as numbers, text, and arrow prompts displayed in a more structured and regular manner. The HUD system simulation results verify the effectiveness of the proposed pre-correction method, demonstrating its ability to effectively mitigate both barrel and perspective distortions introduced by the HUD system.

5. Conclusions

To address the barrel and perspective distortions in in-vehicle HUD projection, this paper proposes HUD-DPCNet, a dual-path correction network based on joint learning. The network extracts image features via a shared encoder and splits into two parallel branches for distortion classification and distortion flow prediction. A model-fitting module based on the Hough transform is introduced to reconstruct a refined distortion flow field, enabling image correction through resampling.

On the ARHDD dataset, HUD-DPCNet achieves a PSNR of 24.617 dB (barrel) and 25.062 dB (perspective), outperforming the best baseline (PCN for barrel, Krishnendu for

perspective) by 1.26 dB and 3.88 dB, respectively. The SSIM reaches 0.845 and 0.873, and the NRMSE reaches 0.163 and 0.157. On the Places 365 dataset, the proposed method achieves a PSNR of 23.914 dB (barrel) and 21.870 dB (perspective), an SSIM of 0.812 and 0.748, and an NRMSE of 0.174 and 0.211. Compared to PCN (barrel) and Krishnendu (perspective), HUD-DPCNet improves PSNR by 0.30 dB and 2.86 dB, respectively. Ablation studies confirm that both the classification branch and the model-fitting module contribute positively to the final correction accuracy.

The current validation is primarily conducted on synthetic datasets and Zemax optical simulations. A physical HUD prototype projecting onto a real windshield, captured by a camera, has not yet been tested. Therefore, the practical feasibility claims have been appropriately limited. Additionally, the model assumes distortions follow parametric barrel or perspective models; non-parametric, localized windshield manufacturing defects that do not conform to these standard models remain a challenge.

We plan to deploy HUD-DPCNet on a real or semi-real optical testbed with a physical HUD prototype and a curved windshield, evaluating its performance under varying eyebox positions, windshield curvatures, and real-time constraints. Further extensions include handling non-parametric distortions via more flexible flow representations.

Author Contributions: Conceptualization, Y.H. and H.C.; methodology, Y.H.; writing—original draft preparation, Y.H.; writing—review and editing, H.C.; supervision, H.C.; project administration, Z.W.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The source code is part of an ongoing project and is not publicly accessible. It can be requested from the corresponding author.

Acknowledgments: This paper is supported by the "Yangfan" Key Program of Guangdong Province of China (Grant No. [2020]5-1).

Conflicts of Interest: Author Zhixi Wang was employed by the company Truly OptoElectronics Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Tardif, J.P.; Sturm, P.; Trudeau, M. Calibration of Cameras with Radially Symmetric Distortion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 1552–1566. [PubMed]
2. Puig, L.; Bermudez, J.; Sturm, P. Calibration of Omnidirectional Cameras in Practice: A Comparison of Methods. *Comput. Vis. Image Underst.* **2012**, *116*, 120–137. [CrossRef]
3. Zhang, Y.; Zhao, L.; Hu, W. A Survey of Catadioptric Omnidirectional Camera Calibration. *Int. J. Inf. Technol. Comput. Sci.* **2013**, *5*, 13–20. [CrossRef]
4. Guo, Y.G.; Ge, Q.P.; Feng, P. Image Geometry Calibration Method Based on Automated Acquirement of Control Points. *Comput. Eng. Appl.* **2007**, *43*, 214–216.
5. Alvarez, L.; Gomez, L.; Sendra, J.R. An Algebraic Approach to Lens Distortion by Line Rectification. *J. Math. Imaging Vis.* **2009**, *35*, 36–50. [CrossRef]
6. Wang, A.Q.; Qiu, T.S.; Shao, L.T. A Simple Method of Radial Distortion Correction with Centre of Distortion Estimation. *J. Math. Imaging Vis.* **2009**, *35*, 165–172. [CrossRef]
7. An, Z.; Meng, X.; Ji, X. Design and performance of an off-axis free-form mirror for a rear mounted augmented-reality head-up display system. *IEEE Photonics J.* **2021**, *13*, 1–15.
8. Park, Y.D.; Jang, T. A study on the ultra precision machining of free-form molds for advanced head-up display device. *J. Korea Acad.-Ind. Coop. Soc.* **2019**, *20*, 290–296.
9. Fujita, K.; Ito, K.; Yoneyama, K. Development of an automotive head-up display using a free-form mirror based optical system. In Proceedings of the International Optical Design Conference 2017, Denver, CO, USA, 9–13 July 2017; p. 54.

10. Ke, L.; Ling, B.; Yinguo, L. Distortion correction algorithm of AR-HUD virtual image based on neural network model of spatial continuous mapping. In Proceedings of the 2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (IS-MAR-Adjunct), Recife, Brazil, 9–13 November 2020; pp. 178–183.
11. Yu, F.; Xu, N.; Chen, S. Direct distortion prediction method for AR-HUD dynamic distortion correction. *Appl. Opt.* **2023**, *62*, 5720–5726. [CrossRef] [PubMed]
12. Gharbi, M.; Shih, Y.; Chaurasia, G.; Ragan-Kelley, J.; Paris, S.; Durand, F. Transform Recipes for Efficient Cloud Photo Enhancement. *ACM Trans. Graph.* **2015**, *34*, 228. [CrossRef]
13. Brito, J.H.; Angst, R.; Koser, K. Radial Distortion Self-Calibration. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 1368–1375.
14. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.
15. Zhai, X.; Qiao, F.; Ma, Y. A Novel Fault Diagnosis Method under Dynamic Working Conditions Based on a CNN with an Adaptive Learning Rate. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]
16. Nguyen, T.N.; Roy, S.; Meunier, J. SmithNet: Strictness on Motion-Texture Coherence for Anomaly Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 2287–2300. [PubMed]
17. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
20. Tian, C.; Xu, Y.; Zuo, W. Asymmetric CNN for Image Superresolution. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 3718–3730.
21. Benligiray, B.; Topal, C. Blind Rectification of Radial Distortion by Line Straightness. In Proceedings of the 24th European Signal Processing Conference (EUSIPCO), Budapest, Hungary, 29 August–2 September 2016; pp. 938–942.
22. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
23. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent Models of Visual Attention. In Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 2204–2212.
24. Ba, J.; Mnih, V.; Kavukcuoglu, K. Multiple Object Recognition with Visual Attention. *arXiv* **2014**, arXiv:1412.7755. [CrossRef]
25. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 2048–2057.
26. Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D.J.; Wierstra, D. DRAW: A Recurrent Neural Network for Image Generation. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 1462–1471.
27. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 2017–2025.
28. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
29. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
30. Illingworth, J.; Kittler, J. A Survey of the Hough Transform. *Comput. Vis. Graph. Image Process.* **1988**, *44*, 87–116. [CrossRef]
31. Huber, S.; Huszár, K.; Kerber, M. Topologically Stable Hough Transform. *arXiv* **2026**, arXiv:2603.08245. [CrossRef]
32. Peng, L.; Zhang, B.; Guo, Z. A Study of Real-Time Track Identification and Data Compression Algorithms for the STCF Main Drift Chamber Based on the Hough Transform. *Acta Phys. Sin.* **2026**, *75*, 120202.
33. Mukhopadhyay, P.; Chaudhuri, B.B. A Survey of Hough Transform. *Pattern Recognit.* **2015**, *48*, 993–1010. [CrossRef]
34. Kazimirov, D.D.; Nikolaev, D.P. Fast Hough Transform with Linear-Log-Cubed Computational Complexity for High-Accuracy Processing of Arbitrary-Shaped Images. *Probl. Inf. Transm.* **2025**, *61*, 289–311. [CrossRef]
35. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
36. Elharrouss, O.; Mahmood, Y.; Bechqito, Y. Task-Based Loss Functions in Computer Vision: A Comprehensive Review. *arXiv* **2025**, arXiv:2504.04242. [CrossRef]
37. Li, W.; Saeedi, S.; McCormac, J. InteriorNet: Mega-Scale Multi-Sensor Photo-Realistic Indoor Scenes Dataset. In Proceedings of the British Machine Vision Conference (BMVC), Newcastle, UK, 3–6 September 2018.

38. Zhou, B.; Lapedriza, A.; Khosla, A. Places: A 10 Million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1452–1464. [CrossRef] [PubMed]
39. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
40. Liao, K.; Lin, C.Y.; Zhao, Y. DR-GAN: Automatic Radial Distortion Rectification Using Conditional GAN in Real-Time. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 725–733. [CrossRef]
41. Yang, S.; Lin, C.; Liao, K. Progressively Complementary Network for Fisheye Image Rectification Using Appearance Flow. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 6348–6357.
42. Bogdan, O.; Eckstein, V.; Rameau, F. DeepCalib: A Deep Learning Approach for Automatic Intrinsic Calibration of Wide Field-of-View Cameras. In Proceedings of the 15th ACM International Conference on Visual Media Production (CVMP), London, UK, 13–14 December 2018; pp. 1–10.
43. Li, X.; Zhang, B.; Sander, P.V. Blind Geometric Distortion Correction on Images through Deep Learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4850–4859.
44. Chaudhury, K.; DiVerdi, S.; Ioffe, S. Auto-Rectification of User Photos. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 3479–3483.
45. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Deep Image Homography Estimation. *arXiv* **2016**, arXiv:1606.03798. [CrossRef]
46. Zhang, X.; Tang, Q. *Zemax Optical Design from Basics to Practice*; Chemical Industry Press: Beijing, China, 2024; p. 394.
47. Ye, Y. Design and Optimization of Augmented Reality Head-Up Display System. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2024.