ORIGINAL PAPER

Representing Camera Response Function by a Single Latent Variable and Fully Connected Neural Network

Yunfeng Zhao[1] · Stuart Ferguson[1] · Huiyu Zhou[2] · Karen Rafferty[1]

Abstract
Modelling the mapping from scene irradiance to image intensity is essential for many computer vision tasks. Such mapping is known as the camera response. Most digital cameras use a nonlinear function to map irradiance, as measured by the sensor to an image intensity used to record the photograph. Modelling of the response is necessary for the nonlinear calibration. In this paper, a new high-performance camera response model that uses a single latent variable and fully connected neural network is proposed. The model is produced using unsupervised learning with an autoencoder on real-world (example) camera responses. Neural architecture searching is then used to find the optimal neural network architecture. A latent distribution learning approach was introduced to constrain the latent distribution. The proposed model achieved state-of-the-art CRF representation accuracy in a number of benchmark tests, but is over twice as fast as the best current models when performing the maximum likelihood estimation during camera response calibration due to the simple yet efficient model representation.

Keywords Relative colour constancy · Colour correction · Colour alignment · Camera colour calibration

1 Introduction

A camera response function (CRF) describes the mapping between the radiant energy received by an image sensor and the intensity output of a camera in the final images [1]. Most cameras are manufactured with nonlinear CRFs [2]. Such nonlinearity is introduced during the stages of image formation in the camera. For instances, analog-digital conversion in the image sensor, white balance adjustment that minimises image colour drift due to differing illumination, gamma correction that expands the luminance range to be interpreted,

and tone mapping for optimising the image visual quality [3]. Popular CRF models include Empirical Model of Response (EMoR), generalised gamma curve model (GGCM), polynomial, and gamma [2,4].

Calibration of a camera response is crucial in many computer vision tasks. Examples of such tasks include image mosaicing where multiple images need to be flawlessly coupled together [5], high dynamic range imaging where images of multiple exposures are used to produce images with greater dynamic ranges [6], and denoising that removes motion blur [7]. CRF calibration also has application in digital forensics [1].

Elaborate CRF representation modelling is the foundation for accurate and rapid CRF calibration. The calibration can be seen as an optimisation process where often the optimal parameters of a selected CRF representation model are calculated to best describe the camera response. The existing CRF models are mostly parametric with multiple parameters. The solution spaces for optimising these parameters are complex with arbitrary distributions. Thus, it takes a long time to calibrate the optimal parameters using existing models.

In this paper, a novel and high-performance non-deterministic CRF representation model, the Single Latent Representation model (SLR), is proposed based on the autoencoder, neural architecture search (NAS), and latent dis-

✉ Karen Rafferty
k.rafferty@qub.ac.uk

Yunfeng Zhao
yzhao25@qub.ac.uk

Stuart Ferguson
r.ferguson@ee.qub.ac.uk

Huiyu Zhou
hz143@leicester.ac.uk

[1] School of Electronics, Electrical Engineering & Computer Science, Queen's University Belfast, 16A Malone Road, Belfast BT9 5BN, United Kingdom

[2] School of Computing and Mathematical Sciences, University of Leicester, University Road, Leicester LE1 7RH, United Kingdom

tribution learning (LDL) techniques. This work presents the following contributions. 1) Pattern of real-world CRFs were extracted by unsupervised learning and represented by a single latent variable using autoencoder. 2) Two approaches (i.e. a LDL and a supervised learning approach using handcrafted feature) are proposed and applied during model representation learning to constrain the latent distribution which further improves the accuracy of camera calibration. 3) A naïve NAS algorithm is used to seek for the optimal autoencoder architecture considering both model accuracy and complexity. 4) The proposed model achieves state-of-the-art performance in terms of accuracy and efficiency of CRF modelling but executes in less than half the time than current best algorithms during CRF calibration.

2 Related work

The latest successful CRF representation model is perhaps the EMoR by Grossberg and Nayar proposed in 2004 [2]. The EMoR describes a CRF by linearly composing a collection of principal components or eigenvectors generated by applying Principal Component Analysis (PCA) on 201 real-world CRFs known as the Database of Response Function (DoRF). Each CRF curve is composed by 1024 uniformly sampled irradiance-intensity converting ratios and is normalised between and passes through $(0, 0)$ and $(1, 1)$. By EMoR, an approximation $\widetilde{f}$ to the CRF f can be constructed from k coefficients and the corresponding eigenvectors:

$$\widetilde{f} = f_0 + \boldsymbol{c}_k^T \boldsymbol{H}_k \tag{1}$$

where f_0 is the base function calculated by averaging all the CRFs in DoRF, $\boldsymbol{c}_k = \boldsymbol{H}_k^T (f - f_0)$ is the model coefficients, and $\boldsymbol{H}_k := [\boldsymbol{h}_1 \cdots \boldsymbol{h}_k]$ is the first k eigenvectors with the largest eigenvalues.

EMoR is an efficient model to represent CRFs by a very small number of parameters or coefficients. As reported in the paper [2], three eigenvalues encode 99.5 percent of the cumulative energies associated with the eigenvalues in DoRF. So far, it is the most widely adopted model for representing a CRF due to its high representing accuracy and simplicity [3,5,8–12].

Polynomial and gamma curves are the two other popular models used for CRF representation whose performances are slightly worse than the EMoR according to a benchmark [4]. A high-order polynomial has the general form:

$$f_{\boldsymbol{\omega}} (\boldsymbol{x}) = \sum_{i=1}^{M} \boldsymbol{\omega}_i \boldsymbol{x}^i \tag{2}$$

where M and $\boldsymbol{\omega}$ are the order number and model coefficients, respectively, and they are the parameters to be determined

through camera calibration. $\boldsymbol{x} \in [0, 1]$ is the model input and represents image pixel intensity.

In general, gamma curves follow the basic form:

$$f (\boldsymbol{x}) = \boldsymbol{x}^{\gamma} \tag{3}$$

where γ is the gamma value typically determined through calibration. This model has been applied in numeral works [13,14].

An extended version of gamma curve named GGCM has been proposed [4] and applied [15]. It is denoted in (4).

$$f_{\boldsymbol{\omega}} (\boldsymbol{x}) = \boldsymbol{x}^{P_{\boldsymbol{\omega}}(\boldsymbol{x})} \tag{4}$$

where the gamma value in the basic form is replaced by a polynomial term $P_{\boldsymbol{\omega}} (\boldsymbol{x}) = \sum_{i=0}^{N} \boldsymbol{\omega}_i \boldsymbol{x}^i$.

A limitation of the current CRF representation models is certainly the high dimensional and complex solution space for solving the optimal model parameters during calibration. A CRF representation with a minimum number of model parameters, e.g. the single gamma value for gamma curves, is demanded for simplifying the calibration. Autoencoder has the ability of generalising and has been used for representation modelling [16]. It compresses data into a much lower dimensional latent space represented by a few latent variables and provides a potential solution to CRF representation. However, such work has not been reported yet.

In general, autoencoder is a neural network that consists of an encoder and a decoder. The encoder maps the input data $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$. And the decoder reconstructs $\boldsymbol{z}$ back to the input data $\tilde{\boldsymbol{x}}$. The latent representation and the model weights are trained by minimising the difference between the input and reconstructed data in an unsupervised process [17].

In the work by Makhzani et al. [18], Adversarial Autoencoder (AAE) has been introduced combining autoencoder with generative adversarial training to deliver unsupervised learning on multiple objectives. It can impose a constraint on the latent distribution by the adversarial training process. The value function of adversarial training can be represented as:

$$\min_{G} \max_{D} V (D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_d} \left[\log D (\boldsymbol{x}) \right] \\ + \mathbb{E}_{\boldsymbol{z} \sim p(z)} \left[\log (1 - D (G (\boldsymbol{z}))) \right] \tag{5}$$

where the encoder in the autoencoder also acts as the generator G in adversarial network to produce the latent representations $\boldsymbol{z}$ from the data distribution P_d. And at the same time, a discriminator D calculates the probability that a representation is generated from the data or prior distribution. AAE has been successfully applied in applications such as image anomaly detection [19] and classification [18].
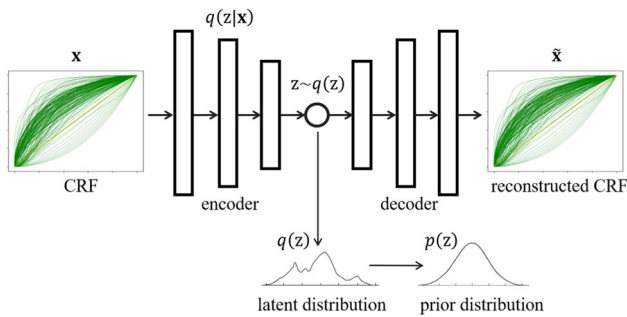
Fig. 1 Architecture of the proposed model for CRF representation. The top row represents a multi-layer fully connected autoencoder with a single latent variable. The bottom row demonstrates the latent distribution and an objective prior distribution

A Variational Autoencoder (VAE) [20] is another popular autoencoder model capable of constraining the distribution of latent representation. Such constraint is achieved by a recognition network that predicts the posterior distribution of the latent space.

Recent advancement of neural networks for end-to-end feature representation and data processing increases the demand for automating the architecture engineering which is time-consuming and usually done manually. NAS automates the neural network engineering process. It can be summarised as three topics: search space, search strategy, and performance evaluation strategy [21]. Search space defines the architecture searching scope and usually involves human prior knowledge. Search strategy determines how the search space is explored. And performance evaluation strategy quantifies candidate model performance.

3 Proposed method

3.1 Autoencoder-based CRF representation model

As shown in Fig. 1, the proposed Single Latent variable camera response Representation (SLR) model inputs a CRF represented by 1,024 uniformly sampled points on the function, reduces the dimensionality to the latent space by the encoder, and outputs the reconstructed CRF by the decoder. As a result, a CRF can be represented by the latent variables in the latent space of the proposed model. A multi-layer fully connected autoencoder with the same number of input and output neurons is selected as the representation model of CRFs.

In our model, the number of hidden layers in either the encoder or decoder is denoted by L. Both the encoder and decoder contain either one, two, or three hidden layers, i.e. $L \in \{1, 2, 3\}$. Each hidden layer contains varied number of neurons, denoted by C_l where $l \in \{1, \ldots, L\}$ is the layer index. C_z is the number of latent variables in the model.

Dropout operation is added to prevent the model from overfitting [22]. Nonlinearity is introduced by an activation function on each unit. The feed-forward operation of the proposed model has the form:

$$\begin{aligned}
r_j &\sim \text{Bernoulli}\,(p) \\
\tilde{u}^{(l)} &= r^{(l)} * u^{(l)} \\
v_i^{(l+1)} &= w_i^{(l+1)} \tilde{u}^l + b_i^{(l+1)} \\
u_i^{(l+1)} &= g\left(v_i^{(l+1)}\right)
\end{aligned} \tag{6}$$

where $r^{(l)}$ is a vector of independent Bernoulli random variables with each element a probability p of being 1, $*$ denotes an element-wise product, $u^{(l)}$ denotes the output vector calculated from the input vector $v^{(l)}$ into layer l, $w^{(l)}$ and $b^{(l)}$ are the model weights and bias at layer l, g is the activation function.

The output vector from layer l is firstly sampled by the dropout operation and then processed by the weights and bias. The processed outputs are nonlinearly activated and used as inputs to the next layer. This process is repeated layer by layer. At test time, the model weights are scaled by p to infer without the effect of the dropout. For CRF construction, the latent variable is used as the input, and the reconstructed CRF $\tilde{x}$ can be obtained at the final output layer.

The model weights in the autoencoder are learnt by independently back-propagating the gradients calculated from the derivatives of the losses. The reconstruction loss is the mean-square-error (MSE) between the input x and reconstructed CRFs $\tilde{x}$:

$$MSE\,(x, \tilde{x}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \tilde{x}_i)^2 \tag{7}$$

where N is the number of training data. Meanwhile, a smoothness loss is imposed on the reconstructed CRF $\tilde{x}$ as a CRF is usually a smooth and continuous function based on the observation from the CRFs in the DoRF:

$$\mathcal{L}\,(\tilde{x}) = \left\| \tilde{x}' \right\|_2 \tag{8}$$

where $\tilde{x}$ is the first derivative of the reconstructed CRF, $\|\cdot\|_2$ denotes the l2-norm.

The optimal number of hidden layers and number of neurons in each hidden layer are determined by NAS.

3.2 Naïve neural architecture search

The optimal architecture of the proposed SLR model in terms of both model accuracy and complexity is determined by NAS. NAS not only helps find the desired model architecture but also brings flexibility to the model design (e.g. when an

extension of the number of latent variables is needed). Since devices with relatively limited computing resources such as mobile phones are being considered for running the proposed model, the model performance estimation needs to be taken cognisance of both the model complexity and its accuracy.

The search space defined is three hidden layers with optional neuron numbers $h_1 = [10, 20, 50, 100, 200, 500]$, $h_2 = [0, 10, 20, 50, 100, 200]$, and $h_3 = [0, 10, 20, 50, 100]$ for both encoder and decoder. Note that when hidden layer two has no neuron, hidden layer three must also have no neuron.

We aim to minimise the model complexity while maximising its accuracy. However, balancing the trade-offs between model complexity and accuracy is a persistent challenge in NAS.

In this paper, the optimal model architecture is determined by a newly proposed NAS method named naïve NAS. Initially, the naïve NAS searches for possible neural architecture in the search space. It then selects M candidate architectures with the highest accuracies from the search. Eventually, the optimal architecture is chosen from those M candidates with the lowest model complexity. The naïve NAS is illustrated in Algorithm 1.

Algorithm 1 Naïve neural architecture search algorithm

Require: $\mathcal{A}$ is the architecture search space. M is the number of architectures with the highest accuracies to be selected from $\mathcal{A}$ for the complexity estimation.
Ensure: The optimal architecture $A \in \mathcal{A}$
1: Construct a Set vector S with length N which equals to the number of elements in $\mathcal{A}$
2: **for** $k \leftarrow 1$ to N **do**
3:   Obtain the architecture object from the search space $A \leftarrow \mathcal{A}_k$
4:   Estimate the architecture accuracy $a \leftarrow Accuracy(A)$
5:   Estimate the architecture complexity $c \leftarrow Complexity(A)$
6:   Construct and store a Set object $S_k \leftarrow \{a, c, A\}$
7: **end for**
8: Descending order S based on the accuracy metric a
9: Select M candidate architectures with the highest accuracies $S' \leftarrow S[0 : M]$
10: Ascending order S' based on the complexity metric c
11: Obtain the architecture object A from the first element of the ordered S'
12: **return** A

Existing variable search strategies can be coupled with the proposed naïve NAS. The Grid Search [23] is the selected strategy since it is thorough and the proposed model is lightweight (i.e. the performance estimation of each candidate architecture can be completed in less than a minute) and the search space is discrete and small (i.e. with only a total of 156 valid candidate architectures in the search space).

The model complexity is calculated by the total number of weights in either the encoder or decoder of the SLR with considering the latent variable:

$$Complexity \sim \left[\left(\sum_{l=1}^{L} C_{l-1}C_l + C_l \right) + C_L C_z + C_z \right] \quad (9)$$

where L is the total number of layers in the encoder or decoder, C is the number of neurons in a specific layer, and C_z is the number of latent variables in the model. The model accuracy is measured by a three-fold cross-validation and (7):

$$Accuracy \sim MSE(x, \tilde{x}) \quad (10)$$

where x and $\tilde{x}$ are the reconstructed and validated CRF curves.

3.3 Constraint on the latent distribution

The latent variable z in the autoencoder follows an arbitrary distribution by default. Two approaches (i.e. a distribution learning and a supervised learning approach using heuristics) have been proposed to constrain the latent variable to follow a prior distribution to help the optimisation process to more accurately find the best z during calibration.

In the first approach, the latent distribution is constrained by "learning" from the objective distribution. It is named *latent distribution learning* (LDL) and achieved by minimising the Kullback–Leibler Divergence (KL-divergence) between the latent and objective distributions.

The latent distribution is approximated by a normal distribution $y \sim \mathcal{N}(\mu, \sigma)$. The normal distribution maximum likelihood of the latent variable is estimated by:

$$\mu = \bar{y},$$
$$\sigma^2 = \sum_{i=1}^{N} (y_i - \mu)^2 \quad (11)$$

where y is the samplings on the latent distribution and M is the number of samplings.

KL-divergence between two normal distributions has the form:

$$KL(\mathcal{N}_1(\mu_1, \sigma_1), \mathcal{N}_2(\mu_2, \sigma_2))$$
$$= -\int \mathcal{N}_1 \log(\mathcal{N}_2)dy + \int \mathcal{N}_1 \log(\mathcal{N}_1)dy$$
$$= \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \quad (12)$$

The KL-divergence between the estimated latent distribution $\mathcal{N}(\mu_1, \sigma_1)$ and the objective standard normal distribution $\mathcal{N}(0, 1)$ can be simplified to:

$$KL\left(\mathcal{N}\left(\mu_1, \sigma_1\right), \mathcal{N}\left(0, 1\right)\right)$$
$$= \frac{1}{2}\left(\mu_1^2 + \sigma_1^2 - 2\log\sigma_1 - 1\right) \quad (13)$$

This is used as the cost for the latent distribution learning in the proposed SLR model. The second approach, named AUC, generates a label for each CRF as the true latent value for distribution constraining. The label is generated by a so-called area under curve approach which calculates the area between the CRF and diagonal curve:

$$\iota = \sum_{i=1}^{N}\left(x_i - \frac{i}{N}\right) \quad (14)$$

where N is the number of samplings in each CRF curve, which is 1,024 for those in the DoRF. The latent distribution is trained using supervised learning by minimising the MSE between the latent and true values.

4 Experiments and results

This section details the experimental setup used to examine and test the proposed model. All the processing and evaluations were performed on a laptop computer with a 2.6-GHz Intel Core i7 processor and a 16-GB memory. To accelerate the optimisation process, a NVIDIA GeForce RTX 2060 GPU was employed.
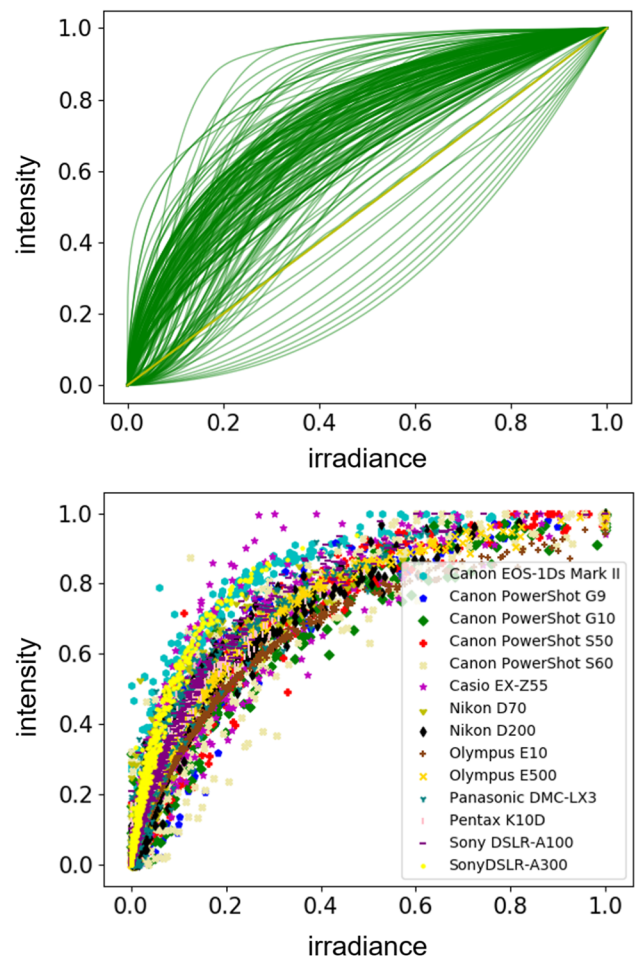
4.1 Datasets

Two datasets, i.e. the DoRF and a modified Middlebury [24], were prepared for performing the validations and benchmarks. Data distribution of these two datasets is demonstrated in Fig. 2.

The DoRF contains 201 CRFs and is currently the most comprehensive dataset of CRFs produced from real-world camera models. This dataset was used in our experiments without modification.

The modified Middlebury dataset contains a total of 112 images. Images of 14 cameras were selected from the original dataset. These cameras were chosen because of their higher cross-channel response uniformity. Each of these cameras took eight images of a Macbeth colour chart under two uniform illuminations and four fixed exposures. This dataset provides an abundance of variation for evaluating CRF calibration accuracy.

The colour patch (CP) locations in the images in the second dataset (24 CP for each image) were carefully labelled by utilising a custom-developed Python script so that the CPs can be extracted and aligned with each other across different images. The true colour values of the CPs are extracted from the RAW images.

Fig. 2 Distribution of the two datasets used for model validation. (a) The CRFs of 201 real-world cameras in the DoRF. Each green curve represents a CRF in the database. (b) Irradiance-intensity scatter plot of the CCPs extracted from the 14 cameras selected from the Middlebury dataset. CCPs of different cameras were rendered in varied colours
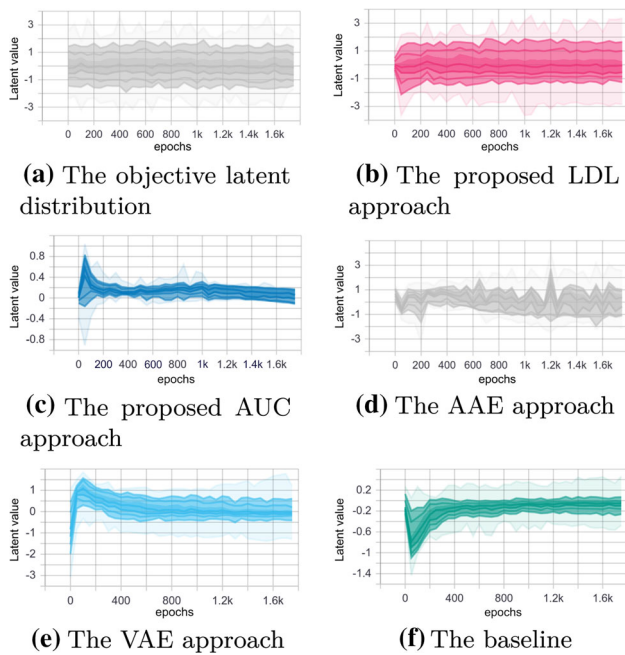
4.2 Evaluation metrics

The root-mean-square error (RMSE) [2,8,25–27] has been widely used to quantify colour difference. It measures the Euclidean distance between two compared vectors:

$$d\left(u_i, v_i\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(u_i - v_i\right)^2} \quad (15)$$

where u and v are the compared vectors and N is the number of items in each of the vectors. A smaller RMSE indicates a better result. A 0 RMSE illustrates identical results.

In the experiments, the RMSEs calculated from comparing the reconstructed CRF with CRFs in the DoRF in the first experiment or those from comparing colour values of JPG and corresponding RAW images in the second experiment

(a) The objective latent distribution

(b) The proposed LDL approach

(c) The proposed AUC approach

(d) The AAE approach

(e) The VAE approach

(f) The baseline

Fig. 3 Visualisation of the (a) objective distribution, compared to the latent distribution developed by the SLR that (b-e) applies four different constraining approaches and (f) imposes no constraint as the training epoch grows

were collected into a result vector h for statistical analysis:

$$h = \begin{bmatrix} e_0 \\ \vdots \\ e_{-1} \end{bmatrix} \Big\} C \qquad (16)$$

where C is the number of camera models to be compared.

The Mean of the result vector h was used as the overall performance evaluation indicator in the first experiment. In the second experiment, five metrics are used to evaluate the result vector produced by each method. The first four are statistical metrics (i.e. Mean, Standard Derivation, Maximum, and 95 Percentile) that reflect model accuracy. Among these four metrics, the Mean of h can be seen as the overall performance metric for accuracy. The time metric was evaluated as the total time needed in seconds (s) for calibrating all the 14 camera models in the second dataset. We considered $\Delta RMSE > 0.005$ to be the thresholds for significant performance difference in the second experiment.

4.3 Latent distribution constraint benchmarks

Constraining the latent distribution using four different approaches (i.e. the two proposed approaches, AAE and VAE approaches), the baseline (i.e. without imposing any constraint method), and the objective distribution were compared in this benchmark as shown in Fig. 3.

Table 1 DoRF curves fitting performance evaluation of various approximation models with different number of parameters in terms of averaged RMSE

Method	Dim One	Two	Three	Four
Our SLR	**5.57E-4**	N. A.	N. A.	N. A.
Gamma	7.34E-3	N. A.	N. A.	N. A.
Polynomial	8.79E-3	2.36E-2	2.50E-2	2.90E-2
GGCM	6.65E-3	7.55E-3	7.06E-3	6.76E-3
EMoR	3.60E-2	1.24E-2	5.71E-3	3.21E-3

Other than the two proposed approaches, AAE constrains the latent distribution by utilising an adversarial training network. The network employs the encoder in the SLR model as the generator. The discriminator consists of two hidden layers with 100 neurons for each layer and a single neuron for both the input and output layers. The adversarial training process is represented by (5).

Instead of imposing additional constraints on the latent distribution as used by the previous three approaches, VAE incorporates the posterior distribution of the latent space into the autoencoder network architecture. Since a normal latent distribution is demanded, the encoder outputs two neurons (i.e. a Mean and a Standard Derivation) representing a normal distribution and then generates the single latent variable by sampling the posterior normal distribution.

The last approach imposes no constraint on the latent distribution and is seen as the baseline for the comparisons.

The objective latent distribution is the standard normal distribution $\mathcal{N}(0, 1)$ as visualised in Fig. 3(a), except for the AUC approach.

The results demonstrated that applying the proposed latent learning approach converged rapidly and led to an accurate latent distribution compared to the objective distribution. The proposed supervised learning approach produced a constrictive yet sharp latent distribution. The latent distribution developed by AAE was unstable compared to the rests. Both the distributions developed by VAE and baseline converged slowly during model training with the baseline distribution also being constrictive.

Overall, the proposed latent distribution learning (LDL) approach performed the best. Thus, this approach was selected to constrain the latent distribution in the rest experiments.

4.4 DoRF curve-fitting benchmark

We firstly compared the performance of the proposed SLR with four other popular models, i.e. gamma, polynomial, GGCM, and EMoR in a DoRF curve-fitting benchmark. In this experiment, every CRF curve in the DoRF was repre-
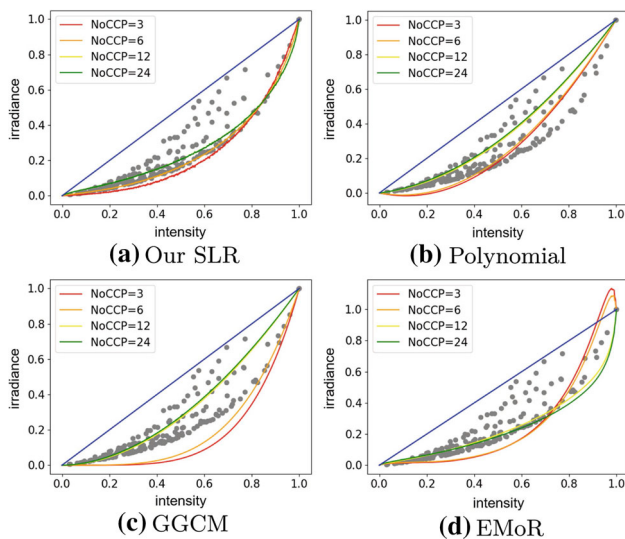
Fig. 4 Radiometric calibration results of a specific camera model (CanonPowerShotG9). The grey dots are the ground truth values. The red, orange, yellow, and red curves are the inverse CRFs calibrated using 3, 6, 12, and 24 NoCCPs on a Macbeth colour chart, respectively. And the blue diagonal line is for reference to. Our model (a) produced more accurate and stable inverse CRFs than the other three tested methods (b, c, d)

sented by each model with a specific number of parameters using the optimal parameters calculated. Four number of parameters (i.e. 1, 2, 3, and 4) were tested for each of the polynomial, GGCM, and EMoR. While gamma and our method were tested with only one parameter since our method works with a single latent variable. The benchmark results are demonstrated in Table 1.

The results indicate that our model with only a single parameter achieved greater than ten-folds better performance (Mean RMSE: 5.61E-4) over most of the other tested methods in the DoRF curve-fitting benchmark. This is not surprising as our model learned the nonlinear CRF features from the real-world CRFs.

4.5 Camera radiometric calibration

The performance and applicability of the proposed SLR model is further validated by a camera radiometric calibration application [28]. This computer vision task estimates inverse-CRF from real camera images.

The true irradiance-intensity mapping values of a specific camera model, i.e. Canon PowerShot G9, and inverse-CRFs produced by four different methods with applying 3, 6, 12, and 24 number of corresponding colour patches (NoCCPs) during calibration are visualised in Fig. 4. Our model fitted more accurately to the true values (see Table 2 for detail). Ours also performed more stable when using varied NoCCPs for the calibration (the Total Variance of the four curves

Table 2 Stability evaluation and comparison of the four commonly used CRF models, i.e. our SLR, polynomial, GGCM, and EMoR, in terms of the total variance between CRFs estimated using 3, 6, 12, and 24 NoCCPs on a Macbeth colour chart

Method	Stability (Total Variance) ↓
Our SLR	**0.66**
Polynomial	1.63
GGCM	8.11
EMoR	3.87

in each plot of Fig. 4 are: our SLR 0.66; polynomial 1.63; GGCM 8.11; EMoR 3.87; the smaller the better).

The radiometric calibration performances of the five methods (i.e. our SLR, gamma, third degree polynomial, third degree GGCM, and EMoR with three parameters) were further evaluated by 14 camera models. Their performances in terms of six metrics are demonstrated in Table 3. The first five metrics are statistical metrics of the RMSEs calculated from the inverse-CRFs of the 14 camera models. The RMSE of each camera model was calculated by comparing the true values and the calibrated inverse-CRF. These five metrics evaluate accuracy of the inverse-CRFs calibrated by each method. Our SLR with a single latent variable and applying the LDL (Mean RMSE 0.062) clearly outperformed the others compared to some of the other methods with using even three parameters. The CRF calibration accuracy improvement contributed by the LDL on our SLR can be quantified by comparing with the baseline. The sixth metric evaluates the total time needed for calibrating all the 14 camera models (i.e. finding the optimal model parameters). It is a metric that reflects model efficiency and is important to be considered for deploying on mobile platforms. Our SLR with LDL (57.4s) completed all the calibrations over twice faster than the gamma (112.6s) that also works with a single parameter and the others that work with more parameters. This is partially contributed by the simple yet efficient autoencoder architecture found by NAS. Our SLR with AUC (43.1s) achieved faster calibration yet sacrificed the calibration accuracy (Mean RMSE 0.105).

5 Conclusion

In this paper, a CRF model that represents camera responses with only a single latent variable has been described. The model used unsupervised learning on real-world CRFs by autoencoder. A simple yet efficient autoencoder architecture was found by applying a naïve NAS algorithm. A latent distribution learning approach was introduced to effectively constrain the latent variable to a normal distribution for improving the accuracy of the CRF calibration process. We

Table 3 Camera radiometric calibration results produced by five different methods (our SLR, gamma, polynomial, GGCM, and EMoR) using eight calibration images and three colour patches in each image are listed. Our SLR was evaluated with four latent distribution constraining approaches and the baseline. Six metrics are used to evaluate the performance of each method. The first five are statistical metrics (mean, median, standard derivation, maximum, and 95 percentile) of RMSE that reflect model accuracy. Among these five metrics, the mean can be seen as the overall performance metric for accuracy. The time metric was evaluated as the total time needed in seconds for calibrating all the 14 camera models

		Mean	Median	S.D.	Max.	95 Pct.	Time
Our SLR (d=1)	Our LDL	**0.062**	**0.056**	**0.056**	**0.106**	**0.101**	57.4
	Our AUC	0.105	0.092	0.064	0.252	0.213	**43.1**
	AAE	**0.064**	**0.057**	**0.027**	0.131	**0.107**	56.4
	VAE	**0.063**	**0.054**	**0.025**	0.108	**0.107**	57.0
	baseline	0.075	0.067	0.039	0.193	0.138	58.5
Our SLR (d=3)	Our LDL	0.097	0.097	0.050	0.184	0.181	61.1
	AAE	**0.063**	**0.051**	0.038	0.178	0.122	60.6
	VAE	0.091	0.091	0.038	0.145	0.139	62.1
	baseline	0.120	0.120	0.051	0.251	0.195	59.72
Gamma (d=1)		0.092	0.080	0.040	0.164	0.162	112.6
Polynomial (d=3)		0.074	0.070	**0.027**	0.156	0.118	129.6
GGCM (d=3)		0.124	0.125	0.055	0.222	0.210	143.3
EMoR (d=3)		0.154	0.105	0.121	0.474	0.408	138.1

demonstrated a superior performance of the proposed model in terms of both the CRF modelling accuracy (i.e. ten-folds better CRF modelling accuracy in the curve-fitting cross-validation benchmark) and calibration efficiency (i.e. around twice as fast as the best current models for CRF calibration in a double-cross-validation benchmark).

Declarations

Competing interests The authors declare no conflicts of interest.

Consent for publication Consent has been obtained from the authors for the publication.

Ethics approval and consent to participate Not applicable.

References

1. Ng, T.T., Tsui, M.P.: Camera response function signature for digital forensics - Part I: Theory and data selection. In: Proceedings of the 2009 1st IEEE International Workshop on Information Forensics and Security, WIFS 2009, pp. 156–160 (2009). https://doi.org/10.1109/WIFS.2009.5386464

2. Grossberg, M.D., Nayar, S.K.: Modeling the space of camera response functions. IEEE Trans. Pattern Anal. Machine Intell. **26**(10), 1272–1282 (2004). https://doi.org/10.1109/TPAMI.2004.88

3. Fu, L., Qi, Y.: Camera response function estimation and application with a single image. In: Lecture Notes in Electrical Engineering, vol. 133 LNEE, pp. 149–156. Springer (2011). https://doi.org/10.1007/978-3-642-25992-0_21

4. Ng, T.-t., Chang, S.-f., Tsui, M.-P.: Using geometry invariants for camera response function estimation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007). https://doi.org/10.1109/CVPR.2007.383000

5. Kim, S.J., Pollefeys, M.: Robust radiometric calibration and vignetting correction. IEEE Trans. Pattern Anal. Machine Intell. **30**(4), 562–576 (2008). https://doi.org/10.1109/TPAMI.2007.70732

6. Mann, S.: Comparametric equations with practical applications in quantigraphic image processing. IEEE Trans. Image Process. **9**(8), 1389–1406 (2000). https://doi.org/10.1109/83.855434

7. Tai, Y.W., Chen, X., Kim, S., Kim, S.J., Li, F., Yang, J., Yu, J., Matsushita, Y., Brown, M.S.: Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. IEEE Trans. Pattern Anal. Machine Intell. **35**(10), 2498–2512 (2013). https://doi.org/10.1109/TPAMI.2013.40

8. Li, H., Peers, P.: CRF-net: Single image radiometric calibration using CNNs. In: ACM International Conference Proceeding Series. Association for Computing Machinery (2017). https://doi.org/10.1145/3150165.3150170

9. Lin, S., Zhang, L.: Determining the radiometric response function from a single grayscale image **II**, 66–73 (2005). https://doi.org/10.1109/CVPR.2005.128

10. Bergmann, P., Wang, R., Cremers, D.: Online Photometric Calibration of Auto Exposure Video for Realtime Visual Odometry and SLAM. IEEE Robotics and Automation Letters 3(2), 627–634 (2018) arXiv:1710.02081. https://doi.org/10.1109/LRA.2017.2777002

11. Matsushita, Y., Lin, S.: Radiometric calibration from noise distributions. (2007). https://doi.org/10.1109/CVPR.2007.383213

12. Seon, J.K., Frahm, J.M., Pollefeys, M.: Radiometric calibration with illumination change for outdoor scene analysis. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008). https://doi.org/10.1109/CVPR.2008.4587648

13. Zhao, Y., Elliott, C., Zhou, H., Rafferty, K.: Spectral illumination correction: Achieving relative color constancy under the spectral domain. In: IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 690–695. IEEE (2018). https://doi.org/10.1109/ISSPIT.2018.8642637

14. Farid, H.: Blind inverse gamma correction. IEEE Trans. Image Process. **10**(10), 1428–1433 (2001). https://doi.org/10.1109/83.951529

15. Rodrigues, P., Barreto, J.P.: Single-image estimation of the camera response function in near-lighting. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1028–1036. IEEE Computer Society (2015). https://doi.org/10.1109/CVPR.2015.7298705

16. Charte, D., Charte, F., del Jesus, M.J., Herrera, F.: An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges. Neurocomput. **404**, 93–107 (2020)

17. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length, and Helmholtz free energy. Adv. Neural Inform. Process. Syst. **6**, 3–10 (1994)

18. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial Autoencoders (2015) arXiv:1511.05644

19. Beggel, L., Pfeiffer, M., Bischl, B.: Robust Anomaly Detection in Images Using Adversarial Autoencoders. In: Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11906 LNAI, pp. 206–222. Springer (2020). https://doi.org/10.1007/978-3-030-46150-8_13

20. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes (2014)

21. Elsken, T., Metzen, J.H., Hutter, F.: Neural Architecture Search: A Survey. arXiv 20, 1–21 (2018) arXiv:1808.05377

22. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

23. Liashchynskyi, P., Liashchynskyi, P.: Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS (2019) arXiv:1912.06059

24. Chakrabarti, A., Scharstein, D., Zickler, T.E.: An Empirical Camera Model for Internet Color Vision. In: Proceedings of the British Machine Vision Conference, vol. 1, p. 4 (2009). Citeseer

25. Lin, S., Jinwei Gu, Yamazaki, S., Heung-Yeung Shum: Radiometric calibration from a single image. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 938–945. IEEE (2004). https://doi.org/10.1109/CVPR.2004.1315266

26. Sharma, A., Tan, R.T., Cheong, L.-F.: Single-Image Camera Response Function Using Prediction Consistency and Gradual Refinement. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12627 LNCS, 19–35 (2020). https://doi.org/10.1007/978-3-030-69544-6_2

27. Gehler, P.V., Rother, C., Blake, A., Minka, T., Sharp, T.: Bayesian color constancy revisited. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008). https://doi.org/10.1109/CVPR.2008.4587765

28. Grana, C., Pellacani, G., Seidenari, S., Cucchiara, R.: Color calibration for a dermatological video camera system. In: Proceedings - International Conference on Pattern Recognition, vol. 3, pp. 798–801 (2004). https://doi.org/10.1109/ICPR.2004.1334649