

Neural Architecture Search for Efficient Uncalibrated Deep Photometric Stereo

Francesco Sarno¹, Suryansh Kumar¹, Berk Kaya¹, Zhiwu Huang¹, Vittorio Ferrari², Luc Van Gool^{1,3}
Computer Vision Lab, ETH Zürich¹, Google Research², KU Leuven³

Abstract

We present an automated machine learning approach for uncalibrated photometric stereo (PS). Our work aims at discovering lightweight and computationally efficient PS neural networks with excellent surface normal accuracy. Unlike previous uncalibrated deep PS networks, which are handcrafted and carefully tuned, we leverage differentiable neural architecture search (NAS) strategy to find uncalibrated PS architecture automatically. We begin by defining a discrete search space for a light calibration network and a normal estimation network, respectively. We then perform a continuous relaxation of this search space and present a gradient-based optimization strategy to find an efficient light calibration and normal estimation network. Directly applying the NAS methodology to uncalibrated PS is not straightforward as certain task-specific constraints must be satisfied, which we impose explicitly. Moreover, we search for and train the two networks separately to account for the Generalized Bas-Relief (GBR) ambiguity. Extensive experiments on the DiLiGenT dataset show that the automatically searched neural architectures performance compares favorably with the state-of-the-art uncalibrated PS methods while having a lower memory footprint.

1. Introduction

Photometric stereo (PS) aims at recovering an object's surface normals from its light varying images captured from a fixed viewpoint. Although range scanning methods [45, 35, 46, 44], multi-view methods [15, 32, 33, 29, 30, 31] and single image dense depth estimation methods [54, 13, 34] can recover the object's surface normals, photometric stereo is excellent at capturing high-frequency surface details such as scratches, cracks, and dents from images. Therefore, it is a favored choice for fine-detailed surface recovery in many scientific and engineering areas such as forensics [53] and molding [70].

Seminal work on PS assumes a Lambertian object under calibrated setting *i.e.*, the directions of the light sources are known [65]. Firstly, the Lambertian object assumption does not hold for surfaces with general reflectance property.

As a result, several robust methods [67, 47, 23], and realistic Bidirectional Reflectance Distribution Function (BRDF) based methods [16, 11, 17, 21, 58] were proposed. Robust methods treat non-Lambertian effects as outliers, and popular realistic BRDF models confine to isotropic BRDF modeling of non-Lambertian surfaces [22, 58]. Hence, these methods can only model the reflectance property of a restricted class of materials. In general, modeling surfaces with unknown reflectance properties is challenging.

In recent years, deep neural networks have significantly improved the performance of many computer vision tasks, including photometric stereo. Their powerful ability to learn from data has helped in modeling surfaces with unknown reflectance properties, which was a challenge for traditional PS methods. Further, neural networks can implicitly learn the image formation process and global illumination effects from data, which classical algorithms cannot pursue. As a result, several deep learning architectures were proposed for PS [20, 63, 8, 7, 41, 40, 19]. Hence, by leveraging a deep neural network, we can overcome the shortcoming of PS due to the Lambertian object assumption. However, these methods still rely on the other assumption of calibrated setting *i.e.*, the light source directions are given at test time, limiting their practical application. Accordingly, uncalibrated deep PS methods that can provide results comparable to calibrated PS networks are becoming more and more popular [6, 27, 9].

The impressive results demonstrated by deep uncalibrated PS methods have a few critical issues: the network architecture is manually designed, and therefore, such networks are typically not optimally efficient and have a large memory footprint [27, 6, 8, 9]. Moreover, the authors of such networks conduct many experiments to explore the effect of empirically selected operations and tune hyperparameters. But, we know from the popular research in machine learning that not only the type of operation but sometimes their placement (ordering) matters for performance [18, 73]. And therefore, a separate line of research known as Neural Architecture Search (NAS) has gained tremendous interest to tackle such challenges in architecture design. NAS methods automate the design process, greatly reducing human effort in searching for an efficient network de-

sign [1]. NAS algorithms have shown great success in many high-level computer vision tasks such as object detection [61, 72], image classification [66], image super-resolution [68], action recognition [60], and semantic segmentation [37]. Yet, its potential for low-level 3D computer vision problem such as uncalibrated PS remains unexplored.

Among architecture search methods, evolutionary algorithms [51, 52] and reinforcement learning-based methods [74, 75] are computationally expensive and need thousands of GPU hours to find architecture. Hence they are not suitable for our problem. Instead, we adhere to the cell-based differentiable NAS formulation. It has proven itself to be computationally efficient and demonstrated encouraging results for many high-level vision problems [39, 36]. However, in those applications, differentiable NAS is used without any task-specific treatment. Unfortunately, this will not work for the uncalibrated PS problem. There exists GBR ambiguity [4] due to the lack of light source information. Moreover, certain task-specific constraints must be satisfied (*e.g.*, unit normal, unit light source direction), and the method must operate on unordered image sets. Unlike typical NAS-based methods, we incorporate human knowledge in our search strategy to address those challenges. To resolve GBR ambiguity, we first search for an efficient light calibration network, followed by a normal estimation network’s search [6]. To handle PS-related constraints, we fix some network layers and define our discrete search space for both networks accordingly. We model our PS architecture search space via a continuous relaxation of the discrete search space, which can be optimized efficiently using a gradient-based algorithm.

We evaluated our method’s performance on the DiLiGenT benchmark PS dataset [59]. The experiments reveal that our approach discovers lightweight architectures, which provides results comparable to the state-of-the-art manually designed deep uncalibrated networks [8, 6, 27]. This paper makes the following contributions:

- We propose the first differentiable NAS-based framework to solve uncalibrated photometric stereo problem.
- Our architecture search methodology considers the task-specific constraints of photometric stereo during search, train, and test time to discover meaningful architecture.
- We show that automatically designed architecture outperforms the existing traditional uncalibrated PS performance and compares favorably against hand-crafted deep PS network with significantly less parameters.

2. Proposed Method

This section describes our task-specific neural architecture search (NAS) approach. We utilize the seminal classical photometric stereo formulation [65, 4] and previous handcrafted deep neural network design [6] as the basis of

our NAS framework. Utilizing previous methods knowledge in the architecture design process not only helps in reducing the architecture’s search time but also provides an optimal architecture with better performance accuracy [6, 27]. Before we describe the NAS modeling of our problem, we define the classical photometric stereo setup.

Consider an orthographic camera observing a rigid object from a given viewpoint $\mathbf{v} = (0, 0, 1)^T$. For PS setup, the images are captured by firing one unique directional light source per image. Let $I \in \mathbb{R}^{m \times n}$ be the measurement matrix comprising of n images with m pixels stacked as column vectors. Let $L \in \mathbb{R}^{3 \times n}$ and $N \in \mathbb{R}^{3 \times m}$ denote all the light sources and surface normals respectively. Then, the image formation model under Lambertian surface assumption is formulated as follows:

$$I = \rho \cdot N^T L + E. \quad (1)$$

Here, $\rho \in \mathbb{R}$ is the diffuse albedo and E accounts for error due to shadows, specularities, or noise. When all non-Lambertian effects are ignored, solving Eq.(1) can recover the actual surface up to a GBR transformation, such that $I = (G^{-T} \bar{N})^T (GL)$. Here, $\bar{N} \in \mathbb{R}^{3 \times m}$ denotes the albedo scaled normals and $G \in \mathbb{R}^{3 \times 3}$ is the transformation matrix with 3 unknown parameters [4, 5]. It indicates that there are many solutions leading up to the same image. Nevertheless, it is well known that specularities [16, 12], interreflections [5], albedo distributions [3, 56] and BRDF properties [62, 69, 42] provide useful cues for disambiguation. However, such cues are not well exploited in a single-stage network designed for regressing per-pixel normals, and therefore, we adhere to use two different neural networks following Chen *et al.* [6]. We first learn the light sources from images by training a light calibration network in a supervised way. Then, we use its results at inference time for the normal estimation network to predict the surface normals. Unlike other uncalibrated deep PS methods, our approach allows automatic search for the optimal architecture both for the light calibration and normal estimation networks.

2.1. Architecture Search for Uncalibrated PS

Leveraging the recent one-shot cell-based NAS method *i.e.*, DARTS [39], we first define different discrete search spaces for light calibration and normal estimation networks. Next, we perform a continuous relaxation of these search spaces, leading to differentiable bi-level objectives for optimization. We perform an end-to-end architecture search for light calibration and normal estimation networks separately to obtain optimal architectures. Contrary to high-level vision problems such as object detection, image classification, and others [39, 10, 75], directly applying the one-shot NAS to existing uncalibrated PS networks [6, 27] may not necessarily lead to a good solution. Unfortunately, for our task, a single end-to-end NAS seems challenging. It may lead

to unstable behavior due to GBR ambiguity [4]. And therefore, we search for an optimal light calibration first and then search for a normal estimation network by keeping some of the necessary operations or layers fixed —such a strategy is used in other NAS based applications [14]. The searched architectures are then trained independently for inference.

• **Background on Differentiable NAS.** In recent years, Neural Architecture Search (NAS) has attracted a lot of attention from the computer vision research community. The goal of NAS is to automate the process of deep neural network design. Among several promising approaches proposed in the past [51, 75, 51, 39, 38, 26, 50], the DARTS [39] has shown promising outcomes due to its computational efficiency and differentiable optimization formulation. So, in this paper, we use it to design an efficient deep neural network to solve uncalibrated PS.

DARTS searches for a computational cell from a set of defined search spaces, which is a building block of the architecture. Once the optimal cells are obtained, it is stacked to construct the final architecture for training and inference. To find the optimal cell, we define search space \mathcal{O} , that is a set of possible candidate operations. The method first performs continuous relaxation on the search spaces and then searches for an optimal cell. A cell is a directed acyclic graph (DAG) with N nodes and E edges. Each node is a latent feature map representation say $x^{(i)}$ for the i^{th} node and each edge is associated with an operation say $o^{(i,j)}$ between node i and node j (see Fig.1(a)). In a cell, each intermediate node is computed from its preceding nodes as follows:

$$x^{(j)} = \sum_{i < j} o^{(i,j)}(x^{(i)}) \quad (2)$$

Let $o^{(i,j)}$ be some operation among K candidate operations $\mathcal{O} = \{o_1^{(i,j)}, o_2^{(i,j)}, \dots, o_K^{(i,j)}\}$. The categorical choice of a specific operation is replaced by the continuous relaxation of the search space by taking softmax over all the defined candidate operations as follows:

$$\tilde{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x) \quad (3)$$

Here, $\alpha^{(i,j)}$ is a vector of dimension $|\mathcal{O}|$ which denotes the operation mixing weights on edge (i, j) (see Fig.1(b)). As a result, the search task for DARTS reduces to a learning set of continuous variable $\alpha^{(i,j)} \forall (i, j)$. The optimal architecture will be determined replacing each mixed operation $\tilde{o}^{(i,j)}$ on edge (i, j) with: $o^{(i,j)} = \arg \max_{o \in \mathcal{O}} \alpha_o^{(i,j)}$ corresponding to the operation which is the “most probable” among the ones listed in \mathcal{O} (see Fig.1(c)-Fig.1(d)). The introduced relaxation allows joint learning of architecture α and its weight ω within the mixture of operations. So, the goal of architecture search now becomes to search for

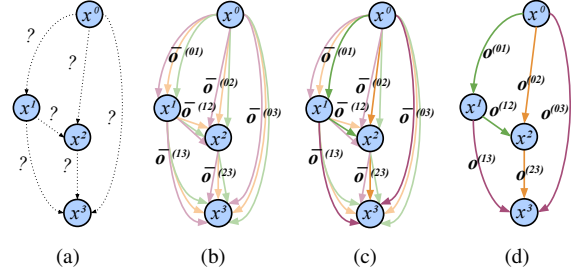


Figure 1: Illustration of a cell. (a) Initially, the optimal operations $\tilde{o}^{(i,j)}$ between nodes $x^{(i)}$ and $x^{(j)}$ are unknown. (b) Each node is computed by a mixture of candidate operations. (c) Architecture encoding is obtained by solving the continuous relaxation of the search space. (d) Optimal cell obtained after selection of most probable candidate operation.

an optimal architecture α using the validation loss with the weights ω that minimizes the training loss for a given α . This leads to following bi-level optimization problem.

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} \quad \mathcal{L}_{val}(\omega^*(\alpha), \alpha); \\ & \text{subject to: } \omega^*(\alpha) = \underset{\omega}{\arg \min} \quad \mathcal{L}_{train}(\omega, \alpha) \end{aligned} \quad (4)$$

where, \mathcal{L}_{val} and \mathcal{L}_{train} are the validation and training losses respectively. This optimization problem is solved iteratively until convergence is reached. The architecture α is updated by substituting the lower-level optimization gradient approximation. Concretely, update α by descending $\nabla_{\alpha} \mathcal{L}_{val}(\omega - \xi \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha), \alpha)$. Subsequently update ω by descending $\nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha)$, where:

$$\nabla_{\alpha} \mathcal{L}_{val}(\omega - \xi \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha), \alpha) \approx \nabla_{\alpha} \mathcal{L}_{val}(\omega^*(\alpha), \alpha) \quad (5)$$

$\xi > 0$ is the learning rate of the inner optimization. The idea is that, $\omega^*(\alpha)$ is approximated with a single learning step which allows the searching process to avoid solving the inner optimization in Eq.(4) exactly. We refer this formulation as second-order approximation [39]. To speed up the searching process, common practice is to apply first-order approximation by setting $\xi = 0$. For more details on the bi-level optimization refer Liu *et al.* work [39].

2.1.1 Our Cell Description

For our problem, we search for both light calibration and normal estimation networks. Our cells consist of two input nodes, four intermediate nodes, and one output node for both of the networks. Each cell at layer k uses the output of two preceding cells (\mathcal{C}_{k-1} and \mathcal{C}_{k-2}) at input nodes and outputs \mathcal{C}_k by channel-wise concatenation of the features at the intermediate nodes. To adjust the spatial dimensions, we define two cells *i.e.*, *normal cells* and *reduction cells*. Normal cells preserve the spatial dimensions of the

input feature maps by applying convolution operations with stride 1. The reduction cells use operations with stride 2 adjacent to input nodes, reducing the spatial dimension by half. Although the cell definition for both networks is the same, the network-level search spaces are different due to the problem’s constraints. Next, we describe our procedure to obtain optimal network architecture for uncalibrated PS.

2.2. Light Calibration Network

Light calibration network predicts all the light source’s direction and intensity from a set of PS images. Here, we assume the object mask is known. One obvious way to estimate light is to regress a set of images with the source direction vectors and intensities in a continuous space. However, converting this task into a classification problem is more favorable for our purpose. It stems from the fact that learning to classify light source directions to predefined bins of angles is much easier than regressing the unit vector itself. Further, using discretized light directions makes the network robust to small input variations.

We represent the light source direction in the upper-hemisphere by its azimuth $\phi \in [0, \pi]$ and elevation $\theta \in [-\pi/2, \pi/2]$ angles. We divide the angle spaces into 36 evenly spaced bins ($K_d = 36$). Our network perform classification on azimuth and elevation separately. For the light intensities, we assign the values in the range of $[0.2, 2]$ divided uniformly into 20 bins ($K_e = 20$) [6].

NAS for Light Calibration Network. To perform NAS for light calibration network, we use the backbone shown in Fig.2(a). The backbone consists of three main parts (*i*) local feature extractor (*ii*) aggregation layer and (*iii*) classifier. The feature extraction layers provide image-specific information for each input image. The weights of these feature extraction layers are shared among all input images. The image-specific features are then aggregated to a global feature representation with the max-pooling operation. Later, global feature representation is combined with the image-specific information and fed to the subsequent layers for classification. The fully connected layers provide softmax probabilities for azimuth, elevation, and intensity values.

We use the NAS algorithm to perform search only over the feature extraction layer and classifier layers for architecture search (shown with dashed box Fig.2(a)), while keeping other layers fixed. For NAS to provide optimal architecture over the searchable blocks in the light-calibration network backbone, we define our search space as follows:

1. *Search Space.* Our candidate operations set in search space for light calibration network is composed of $\mathcal{O}^{light} = \{“1 \times 1$ separable conv.”, “ 3×3 separable conv.”, “ 5×5 separable conv.”, “skip connection”, “zero”}. The “zero” operation indicates the lack of connection between two nodes. Each convolutional layer defined in the set first applies ReLU [71] and then convolution with given kernel

size followed by batch-normalization [24]. As before, our cells consist of two input nodes, four intermediate nodes, and one output node §2.1.1. Just for the initial cell, we use stem layers as its input for better search. These layers apply fixed convolutions to enrich the initial cell input features.

2. *Continuous relaxation and Optimization.* We perform the continuous relaxation of our defined search space using Eq.(3) for differentiable optimization. During searching phase, we perform alternating optimization over weights ω and architecture encoding values α as follows:

- Update network weights ω by $\nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha)$.
- Update architecture mixing weights α by $\nabla_{\alpha} \mathcal{L}_{val}(\omega - \xi \nabla_{\omega} \mathcal{L}_{train}(\omega, \alpha), \alpha)$. (see Eq.(5))

\mathcal{L}_{train} and \mathcal{L}_{val} denote the loss computed over training and validation datasets, respectively. We use multi-class cross-entropy loss on azimuth, elevation, and intensity classes to optimize our network [6]. The total light calibration loss is

$$\mathcal{L}_{light} = \mathcal{L}_{\phi} + \mathcal{L}_{\theta} + \mathcal{L}_e \quad (6)$$

where, \mathcal{L}_{ϕ} , \mathcal{L}_{θ} , and \mathcal{L}_e are the losses for azimuth, elevation, and intensity respectively. We utilize the synthetic Blobby and Sculpture datasets [8] for this optimization where ground-truth labels for lighting are provided.

Once the searching phase is complete, we convert the continuous architecture encoding values into a discrete architecture. For that, we select the strongest operation on each edge (i, j) with: $o^{(i,j)} = \arg \max_{o \in \mathcal{O}^{light}} \alpha_o^{(i,j)}$. We preserve only the strongest two operations preceding each intermediate node. We train our designed architecture with optimal operations from scratch on the training dataset again to optimize weights before testing §3.1.

2.3. Normal Estimation Network

We independently search for optimal normal estimation network using the backbone shown in Fig.2(b). To use the light source information into the network, we first convert n light direction vectors into a tensor $\mathcal{X} \in \mathbb{R}^{n \times 3 \times h \times w}$, where each 3-vector is repeated over spatial dimensions h and w . This tensor is then concatenated with the input image to form a tensor $\mathcal{I} \in \mathbb{R}^{n \times 6 \times h \times w}$. Similar to the light calibration network, we use a shared-weight feature extraction block to process each input. After image-specific information is extracted, we combine them in a fixed aggregation layer with the max-pooling operation and obtain a global representation. Keeping the aggregation layer fixed allows the network to operate on an arbitrary number of test images and improves robustness. The global information is finally used to regress the normal map, where a fixed normalization layer is used to satisfy the unit-length constraint.

NAS for Normal Estimation Network. Similar to light calibration network, the cells here consist of two input nodes, four intermediate nodes, and one output node. To

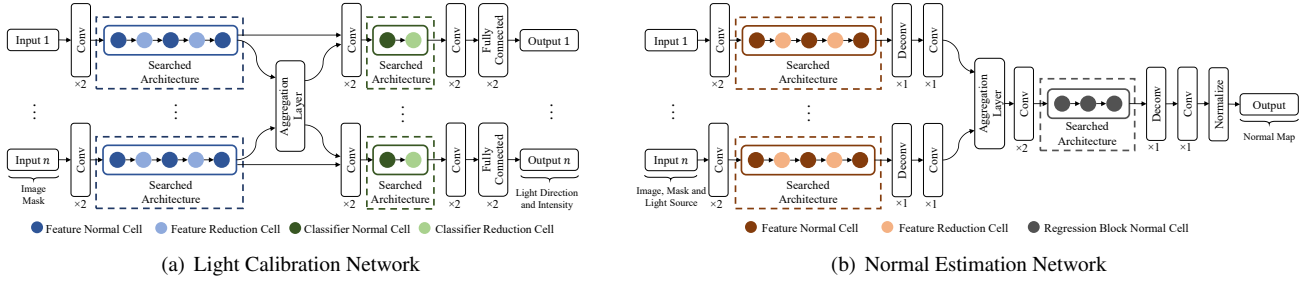


Figure 2: Our pipeline consists of two networks: (a) Light Calibration Network predicts light source directions and intensities from images. Our search is confined to feature extraction module and classification module. (b) Normal Estimation Network outputs the surface normal map from images and estimated light sources. Our search is confined to feature extraction module and regression module.

efficiently search for architectures at initial layers, we make use of stem layers prior to each search space [39]. These layers apply fixed convolutions to enrich the input features.

1. Search Space. It is a well-known fact that the kernel size has great importance in vision problems. Recent work on photometric stereo has verified that using bigger kernel size helps to explore the spatial information, but stacking too many of them leads to over-smoothing and degrades the performance [73]. Therefore, we selectively use different kernel sizes in the candidate operations set $\mathcal{O}^{normal} = \{“1 \times 1 \text{ separable conv.}”, “3 \times 3 \text{ separable conv.}”, “5 \times 5 \text{ separable conv.}”, “\text{skip connection}”, “\text{zero}”\}$. Here also, each convolutional layer defined in the set first applies ReLU [71] and then convolution with given kernel size followed by batch-normalization [24]. The selection of candidate operation sets is further investigated in §3.3 of the supplementary material.

2. Continuous Relaxation and Optimization. Similar to light calibration network, we use Eq.(3) to make the search space continuous. We then jointly search for the architecture encoding values and the weights using the ground-truth surface normals and light source information during optimization. The optimization is performed using the same bi-level optimization approximation strategy (see Eq.(4) and Eq.(5)). We normalize the images before feeding them to the network. The normalization ensures the network is robust to different intensity levels. To search normal estimation network, we use the following cosine similarity loss:

$$\mathcal{L}_{normal} = \frac{1}{m} \sum_i^m (1 - \tilde{\mathbf{n}}_i^T \mathbf{n}_i) \quad (7)$$

where, $\tilde{\mathbf{n}}_i$ is the estimated normal by our network and \mathbf{n}_i is the ground-truth normal at pixel i . Note that $\tilde{\mathbf{n}}_i$ is a unit-vector due to the fixed normalization layer.

After the search optimization for normal estimation network is done, we obtain optimal discrete architecture by keeping the operation $o^{(i,j)} = \arg \max_{o \in \mathcal{O}^{normal}} \alpha_o^{(i,j)}$ on each edge (i, j) . Similar to [39], we only preserve the two preceding operations with highest weight for each node. Fi-

nally, we train our normal estimation network from scratch using the searched architecture. Our normal estimation network uses the light directions and intensities estimated by the light calibration network to predict normals at test time.

3. Experiments and Results

This section first describes our procedure in preparing the dataset for the searching, training, and testing phase. Later, we provide the implementation of our method, followed by statistical evaluations and ablation.

3.1. Dataset Preparation

We used three popular photometric stereo datasets for our experiments, statistical analysis, and comparisons, namely, Blobby [25], Sculpture [64], and DiLiGenT [57].

Search and Train Set Details. For architecture search and optimal architecture training, we used 10 objects from the Blobby dataset [25] and 8 from the Sculpture dataset [64]. We considered the rendered photometric stereo images of these datasets provided by Chen *et al.* [8]. It uses 64 random lights to render the objects. In search and train phase, we randomly choose 32 light source images. Following Chen *et al.* [8], we considered 128×128 sized images for both Blobby and Sculpture dataset.

(a) Preparation of Search Set. Searching for an optimal architecture using one-shot NAS [39] can be computationally expensive. To address that, we use only 10% of the dataset such that it contains subjects from all the categories present in the Blobby and Sculpture dataset. Next, we resized all those 128×128 resolution images to 64×64 . We refer this dataset as Blobby search set and Sculpture search set. Our search set is further divided into search train set and search validation set. This train set is prepared by taking eight shapes from Blobby search set and six shape from Sculpture search set. The search validation set is composed of two shapes from Blobby and Sculpture search sets, respectively. Hence, approximately 80% of the search set is used as search train set and 20% is used as search validation

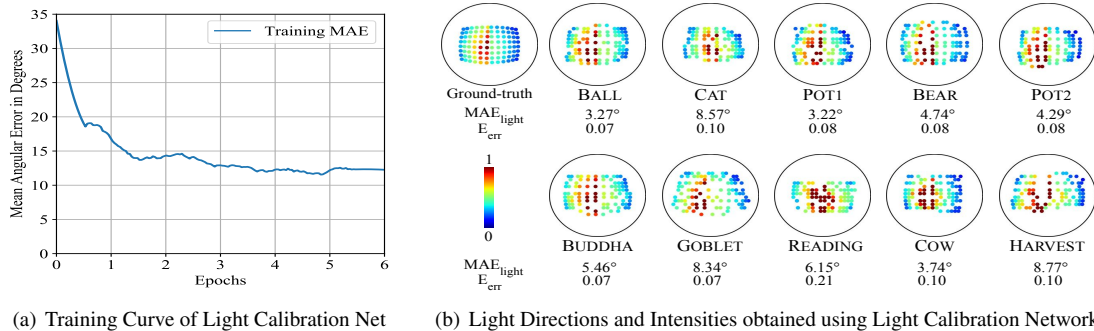


Figure 3: (a) Training curve of the light calibration network. (b) Light calibration network results on DiLiGenT objects. We show the light direction by projecting the vector $[x, y, z]$ to a corresponding point $[x, y]$. The color of the point shows the light intensity value in $[0, 1]$ range. MAE_{light} is the mean angular error in the estimation of light source direction and E_{err} stands for the intensity error.

set. This is done in a way that there is no common subject between train and validation sets. We used a batch size of four at train and validation time during search phase. The search set is same for the light calibration and normal estimation network’s search.

(b) Preparation of Train Set. Once the optimal architectures for light calibration and normal estimation are obtained, we use the train set for training these networks from scratch. Since, we searched architecture using 64×64 size images, we use convolution layer with stride 2 at the train time for the light calibration network’s training. Following Chen *et al.* [8], we use 99% of the Blobby and Sculpture dataset for training and 1% for the validation. For light calibration we used batch size of thirty-two at train time and eight for validation. For normal estimation, instead, we considered batch size of four both at training and validation.

3.1.1 Test Set Details.

We tested our networks on the recently proposed DiLiGenT PS dataset [57]. It consists of 10 real-world objects, with images captured by 96 LED light sources. It provides ground-truth normals and calibrated light directions making it an ideal dataset for evaluation. Following Chen *et al.* [8], we use 96 images per object at 128×128 resolution to test our light calibration and normal estimation network.

3.2. Implementation Details

The proposed method is implemented with Python 3.6, and PyTorch 1.1 [49]. For both networks, we employ the same optimizer, learning rate, and weight decay settings. The architecture parameters α and the network weights ω are optimized using Adam [28]. During the architecture search phase, the optimizer is initialized with the learning rate $\eta_{alpha} = 3 \times 10^{-4}$, momentum $\beta = (0.5, 0, 999)$ and weight decay of 1×10^{-3} . At model train time, the optimizer is initialized with the learning rate $\eta_w = 5 \times 10^{-4}$, momentum $\beta = (0.5, 0, 999)$ and weight decay of 3×10^{-4} .

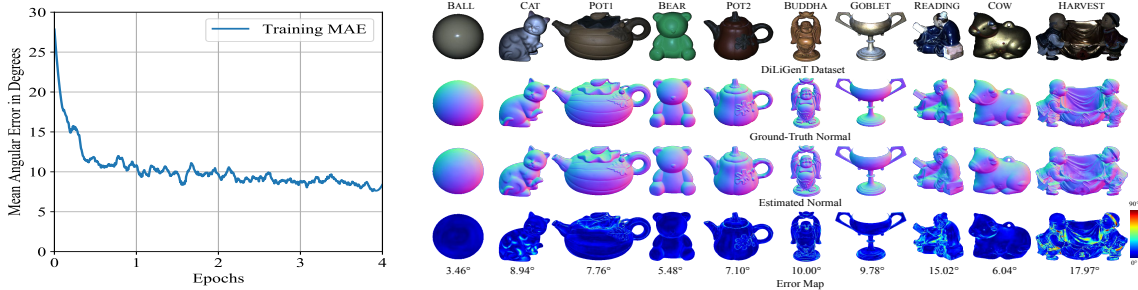
We conducted all the experiments on a computer with a single NVIDIA GPU with 12GB of RAM.

We search for two types of cells, namely normal cell and reduction cell. We use the loss function defined in Eq.(6) and Eq.(7) during search phase to recover optimal cells for each network independently. Fig.2(a) and Fig.2(b) show the light calibration and the normal estimation backbone and its searchable parts, respectively. For light calibration network, we have two searchable blocks (i) Feature block and (ii) Classification block. Here, we design our feature block using three normal cells, two reduction cells, and the classification block using one normal cell and one reduction cell. Similarly, we have two searchable blocks (i) Feature block and (ii) Regressor block for normal estimation network. Here, the feature block comprises three normal cells and two reduction cells, while the regressor block is composed of three normal cells. To construct the network design for searchable blocks, each normal cell is concatenated sequentially to the reduction cell in order. We use 3 epochs to search architecture for each network.

At train time, we regularize the normal estimation network loss function using the concept of auxiliary tower [39] for performance gain. Consequently, we modify its loss function at train time as follows:

$$\mathcal{L}_{normal} = \frac{1}{m} \sum_i^m (1 - \tilde{\mathbf{n}}_i^T \mathbf{n}_i) + \lambda_{aux} \frac{1}{m} \sum_i^m (1 - \hat{\mathbf{n}}_i^T \mathbf{n}_i) \quad (8)$$

where, λ_{aux} is a regularization parameter, and $\hat{\mathbf{n}}_i$ is the output surface normal at pixel i due to auxiliary tower. We set $\lambda_{aux} = 0.4$. We observed that the auxiliary tower improves the performance of the normal estimation network. It can be argued that a similar regularizer could be used for the light calibration network. However, in that case, we have to incorporate that regularizer for each image independently, which can be computationally expensive. Fig.3(a) and Fig.4(a) show the training curve for the light calibration and normal estimation network respectively. We trained the



(a) Training Curve of Normal Estimation Net

(b) Surface Normals obtained using Normal Estimation Network

Figure 4: (a) Training curve of the normal estimation network. (b) Qualitative surface normal results on the DiLiGenT benchmark. The bottom row demonstrates the angular error maps and mean angular errors of our results.

Methods↓	Dataset →	Ball	Cat	Pot1	Bear	Pot2	Buddha	Goblet	Reading	Cow	Harvest	Average
Alldrin et al. (2007)[2]		7.27	31.45	18.37	16.81	49.16	32.81	46.54	53.65	54.72	61.70	37.25
Shi et al. (2010)[55]		8.90	19.84	16.68	11.98	50.68	15.54	48.79	26.93	22.73	73.86	29.59
Wu & Tan (2013)[69]		4.39	36.55	9.39	6.42	14.52	13.19	20.57	58.96	19.75	55.51	23.93
Lu et al. (2013)[43]		22.43	25.01	32.82	15.44	20.57	25.76	29.16	48.16	22.53	34.45	27.63
Papadh. et al. (2014)[48]		4.77	9.54	9.51	9.07	15.90	14.92	29.93	24.18	19.53	29.21	16.66
Lu et al. (2017)[42]		9.30	12.60	12.40	10.90	15.70	19.00	18.30	22.30	15.00	28.00	16.30
Ours		3.46	8.94	7.76	5.48	7.10	10.00	9.78	15.02	6.04	17.97	9.15

Table 1: Quantitative comparison with the traditional uncalibrated photometric stereo methods on DiLiGenT benchmark. Our searched architecture estimates accurate surface normals of the object with general reflectance property.

light calibration and normal estimation networks for six and three epochs, respectively for inference.

3.3. Qualitative and Quantitative Evaluation

Evaluation Metric. To measure the accuracy of the estimated light directions and surface normals, we adopt the standard mean angular error (MAE) metric as follows:

$$\text{MAE}_{light} = \frac{180}{\pi} \frac{1}{n} \sum_i \arccos(\tilde{\ell}_i^T \ell_i) \quad (9)$$

$$\text{MAE}_{normal} = \frac{180}{\pi} \frac{1}{m} \sum_i \arccos(\tilde{\mathbf{n}}_i^T \mathbf{n}_i) \quad (10)$$

where, n is the number of images, and m is the number of object pixels. $\tilde{\ell}_i$ and ℓ_i denote the estimated and ground-truth light directions. Similarly, $\tilde{\mathbf{n}}_i$ and \mathbf{n}_i denote the estimated and ground-truth surface normals. As the auxiliary tower is not used at test time, we define metrics using $\tilde{\mathbf{n}}_i$. Following previous works [6, 8], we report MAE in degrees.

Unlike light directions and surface normals, light intensity can only be estimated up to a scale factor. For this reason, instead of using the exact intensity values for evaluation, we use a scale-invariant relative error metric [6]:

$$E_{err} = \frac{1}{n} \sum_i \left(\frac{|s\tilde{e}_i - e_i|}{e_i} \right) \quad (11)$$

Here, \tilde{e}_i and e_i are the estimated and ground-truth light intensities, respectively with s as the scale factor. Following

Chen *et al.* [7], we solve $\text{argmin}_s \sum_i^n (s\tilde{e}_i - e_i)^2$ using the least squares to compute s for intensity evaluation.

3.3.1 Inference

Once optimal architectures are obtained, we train these networks for inference. We test their performance using the defined metric on the Test set. For each test object, we first feed the object images at 128×128 resolution to the light calibration network to predict the light directions and intensities. Then, we use the images and estimated light sources as input to the normal estimation network to predict the surface normals. Visual diagram of the optimal cell architectures is provided in the supplementary material.

(a) Performance of Light Calibration Network. To show the validity of our searched light calibration network, we compared its performance on DiLiGenT ground-truth light direction and intensity. Fig.3(b) shows the quantitative and qualitative results obtained using our network. Concretely, it provides light directions MAE_{light} and intensity error (E_{err}) for all object categories. The results indicate that the searched light calibration network can reliably predict light source direction and intensity from images of object with complex surface profile and different material properties.

(b) Comparison of Surface Normal Accuracy. We documented the performance comparison of our approach against the traditional uncalibrated photometric stereo methods in Table 1. The statistics show that our method performs significantly better than such uncalibrated approaches

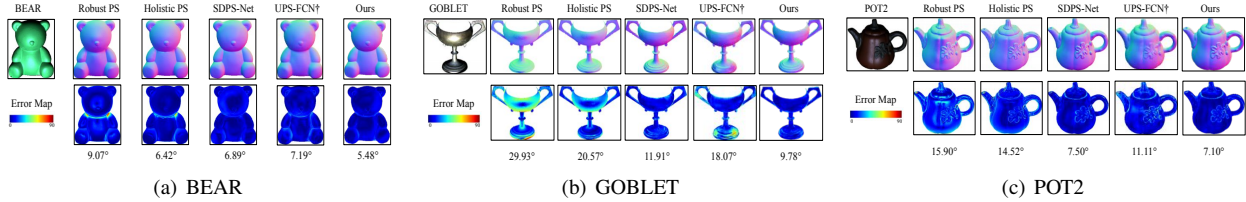


Figure 5: Visual comparison against Robust PS [48], Holistic PS[69], SDPS-Net [6] and UPS-FCN [8] on (a) BEAR (b) GOBLET and (c) POT2 object from DiLiGenT dataset. The statistics show the superiority of our searched architecture.

Methods	Params (M)	Ball	Cat	Pot1	Bear	Pot2	Buddha	Goblet	Reading	Cow	Harvest	Average
UPS-FCN [†] (2018)[8]	6.1	3.96	12.16	11.13	7.19	11.11	13.06	18.07	20.46	11.84	27.22	13.62
SDPS-Net (2019) [6]	6.6	2.77	8.06	8.14	6.89	7.50	8.97	11.91	14.90	8.48	17.43	9.51
GCNet (2020) [9] + PS-FCN [8]	6.8	2.50	7.90	7.20	5.60	7.10	8.60	9.60	14.90	7.80	16.20	8.70
Kaya et al. (2021) [27]	8.1	3.78	7.91	8.75	5.96	10.17	13.14	11.94	18.22	10.85	25.49	11.62
Ours (w/o auxiliary)	4.4	4.86	9.79	9.98	4.97	8.95	10.29	9.46	15.59	8.06	18.20	9.98
Ours	4.4	3.46	8.94	7.76	5.48	7.10	10.00	9.78	15.02	6.04	17.97	9.15

Table 2: Quantitative comparison of deep uncalibrated photometric stereo methods on DiLiGenT benchmark [59]. Our searched architecture on average provides results that are better compared to other deep networks not only in surface orientation accuracy (MAE) but also in model size. The blue show the statistics where our method has the second best performance. We used deeper version of UPS-FCN [8].

for all the object categories. That is because we don't explicitly rely on BRDF model assumptions and the well-known matrix factorization approach. Instead, our work exploits the benefit of the deep neural network to handle complicated BRDF problems by learning from data. Rather than using matrix factorization, our work independently learns to estimate light from data and use it to solve surface normals.

Further, we compared our method with the state-of-the-art deep uncalibrated PS methods. Table 2 shows that our method achieves competitive results with an average MAE_{normal} of 9.15° , having the second best performance overall. The best performing method [9] uses a four-stage cascade structure, making it complex and deep. On the contrary, our searched architecture is light and it can achieve such accuracy with 2.4M fewer parameters. Fig.5 provides additional visual comparison of our results with several other approaches from the literature [48, 69, 6, 8]. Table 2 also shows the benefit of using an auxiliary tower at train time (see supplementary for more details and results).

(c) Ablation Study. (i) *Analysing the performance with the change in number of input images at test time.* Our light calibration and normal estimation network can work with an arbitrary number of input images at test time. In this experiment, we analyse how the number of images affects the accuracy of the estimated lighting and surface normals. Fig. 6(a) and 6(b) show the variation in the mean angular error with different number of images. As expected, the error decreases as we increase the number of images. Of course, feeding more images allows the networks to extract more information, and therefore, the best results are obtained by using all 96 images provided by the DiLiGenT dataset [59]. For more experimental results, ablations and visualizations, refer to the supplementary material.

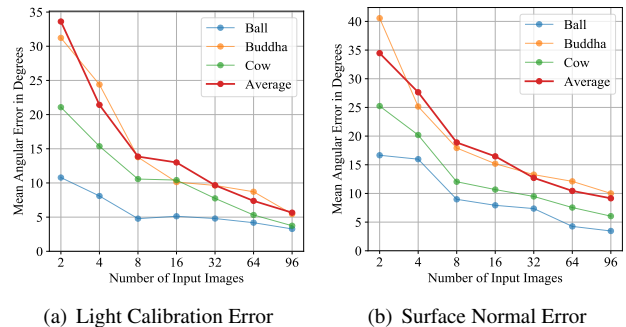


Figure 6: Variation in MAE w.r.t the change in the number of input images at test time. Observation with (a) light calibration and (b) normal estimation network, respectively.

4. Conclusion

In this paper, we demonstrated the effectiveness of applying differentiable NAS to deep uncalibrated PS. Though using the existing differentiable NAS framework directly to our problem is not straightforward, we showed that we could successfully utilize NAS provided PS-specific constraints are well satisfied during the search, train, and test time. We search for an optimal light calibration network and normal estimation network using the one-shot NAS method by leveraging hand-crafted deep neural network design knowledge and fixing some of the layers or operations to account for the PS-specific constraints. The architecture we discover is lightweight, and it provides comparable or better accuracy than the existing deep uncalibrated PS methods.

Acknowledgement. This work was funded by Focused Research Award from Google (CVL, ETH 2019-HE-318, 2019-HE-323).

References

- [1] Auttml. <https://www.auttml.org/auttml>. Accessed: 02-06-2021.
- [2] Neil Alldrin, Todd Zickler, and David Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [3] Neil G Alldrin, Satya P Mallick, and David J Kriegman. Resolving the generalized bas-relief ambiguity by entropy minimization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–7. IEEE, 2007.
- [4] Peter N Belhumeur, David J Kriegman, and Alan L Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999.
- [5] Manmohan Krishna Chandraker, Fredrik Kahl, and David J Kriegman. Reflections on the generalized bas-relief ambiguity. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 788–795. IEEE, 2005.
- [6] Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee K Wong. Self-calibrating deep photometric stereo networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8747, 2019.
- [7] Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee Kenneth Wong. Deep photometric stereo for non-lambertian surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [8] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Ps-fcn: A flexible learning framework for photometric stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–18, 2018.
- [9] Guanying Chen, Michael Waechter, Boxin Shi, Kwan-Yee K Wong, and Yasuyuki Matsushita. What is learned in deep uncalibrated photometric stereo? In *European Conference on Computer Vision*, 2020.
- [10] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair darts: Eliminating unfair advantages in differentiable architecture search, 2020.
- [11] Hin-Shun Chung and Jiaya Jia. Efficient photometric stereo on glossy surfaces with wide specular lobes. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [12] Ondrej Drbohlav and M Chaniler. Can two specular pixels calibrate photometric stereo? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1850–1857. IEEE, 2005.
- [13] Kui Fu, Jiansheng Peng, Qiwen He, and Hanxiao Zhang. Single image 3d object reconstruction based on deep learning: A review. *Multimedia Tools and Applications*, 80(1):463–498, 2021.
- [14] Y Fu, W Chen, H Wang, H Li, Y Lin, and Z Wang. Autogandistiller: Searching to compress generative adversarial networks. *ICML*, 2020.
- [15] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.
- [16] Athinodoros S Georghiades. Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In *ICCV*, pages 816–823. IEEE, 2003.
- [17] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2009.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016.
- [19] Santo Hiroaki, Michael Waechter, and Yasuyuki Matsushita. Deep near-light photometric stereo for spatially varying reflectances. In *European Conference on Computer Vision*, 2020.
- [20] Satoshi Ikehata. Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018.
- [21] Satoshi Ikehata and Kiyoharu Aizawa. Photometric stereo using constrained bivariate regression for general isotropic surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2179–2186, 2014.
- [22] Satoshi Ikehata and Kiyoharu Aizawa. Photometric stereo using constrained bivariate regression for general isotropic surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [23] Satoshi Ikehata, David Wipf, Yasuyuki Matsushita, and Kiyoharu Aizawa. Photometric stereo using sparse bayesian regression for general diffuse surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1816–1831, 2014.
- [24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [25] M. K. Johnson and E. H. Adelson. Shape estimation in natural illumination. *CVPR '11*, page 2553–2560, USA, 2011. IEEE Computer Society.
- [26] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric Xing. Neural architecture search with bayesian optimisation and optimal transport, 2019.
- [27] Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncalibrated neural inverse rendering for photometric stereo of general surfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021.
- [28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [29] Suryansh Kumar. Jumping manifolds: Geometry aware dense non-rigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5346–5355, 2019.
- [30] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 51–60, 2020.
- [31] Suryansh Kumar, Anoop Cherian, Yuchao Dai, and Hongdong Li. Scalable dense non-rigid structure-from-motion: A

- grassmannian perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2018.
- [32] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4649–4657, 2017.
- [33] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [34] Suryansh Kumar, Ram Srivatsav Ghorakavi, Yuchao Dai, and Hongdong Li. Dense depth estimation of a complex dynamic scene without explicit 3d motion estimation. *arXiv preprint arXiv:1902.03791*, 2019.
- [35] Kiriakos N Kutulakos and Eron Steger. A theory of refractive and specular 3d shape by light-path triangulation. *International Journal of Computer Vision*, 76(1):13–29, 2008.
- [36] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation, 2019.
- [37] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 82–92, 2019.
- [38] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search, 2017.
- [39] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.
- [40] Fotios Logothetis, Ignas Budvytis, Roberto Mecca, and Roberto Cipolla. A cnn based approach for the near-field photometric stereo problem. *arXiv preprint arXiv:2009.05792*, 2020.
- [41] Fotios Logothetis, Ignas Budvytis, Roberto Mecca, and Roberto Cipolla. Px-net: Simple, efficient pixel-wise training of photometric stereo networks. *arXiv preprint arXiv:2008.04933*, 2020.
- [42] Feng Lu, Xiaowu Chen, Imari Sato, and Yoichi Sato. Symps: Brdf symmetry guided photometric stereo for shape and light source estimation. *IEEE transactions on pattern analysis and machine intelligence*, 40(1):221–234, 2017.
- [43] Feng Lu, Imari Sato, and Yoichi Sato. Uncalibrated photometric stereo based on elevation angle recovery from brdf symmetry of isotropic materials. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 168–176, 2015.
- [44] Davide Menini, Suryansh Kumar, Martin R Oswald, Erik Sandstrom, Cristian Sminchisescu, and Luc Van Gool. A real-time online learning framework for joint 3d reconstruction and semantic segmentation of indoor scenes. *arXiv preprint arXiv:2108.05246*, 2021.
- [45] Shree K Nayar and Mohit Gupta. Diffuse structured light. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2012.
- [46] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. IEEE, 2011.
- [47] Tae-Hyun Oh, Hyeonwoo Kim, Yu-Wing Tai, Jean-Charles Bazin, and In So Kweon. Partial sum minimization of singular values in rpca for low-level vision. In *Proceedings of the IEEE international conference on computer vision*, pages 145–152, 2013.
- [48] Thoma Papadhimetri and Paolo Favaro. A closed-form, consistent and robust solution to uncalibrated photometric stereo via local diffuse reflectance maxima. *International journal of computer vision*, 107(2):139–154, 2014.
- [49] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [50] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing, 2018.
- [51] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search, 2019.
- [52] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041*, 2017.
- [53] Ufuk Sakarya, Uğur Murat Leloğlu, and Erol Tunalı. Three-dimensional surface reconstruction for cartridge cases using photometric stereo. *Forensic science international*, 175(2-3):209–217, 2008.
- [54] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.
- [55] Boxin Shi, Yasuyuki Matsushita, Yichen Wei, and Chao Xu. Self-calibrating photometric stereo. pages 1118–1125, 06 2010.
- [56] Boxin Shi, Yasuyuki Matsushita, Yichen Wei, Chao Xu, and Ping Tan. Self-calibrating photometric stereo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1118–1125. IEEE, 2010.
- [57] B. Shi, Z. Mo, Z. Wu, D. Duan, S. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):271–284, 2019.
- [58] Boxin Shi, Ping Tan, Yasuyuki Matsushita, and Katsushi Ikeuchi. Bi-polynomial modeling of low-frequency reflectances. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1078–1091, 2013.
- [59] Boxin Shi, Zhe Wu, Zhipeng Mo, Dinglong Duan, Sai-Kit Yeung, and Ping Tan. A benchmark dataset and evaluation

- for non-lambertian and uncalibrated photometric stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3707–3716, 2016.
- [60] Rhea Sanjay Sukthanker, Zhiwu Huang, Suryansh Kumar, Erik Goron Endsjo, Yan Wu, and Luc Van Gool. Neural architecture search of spd manifold networks, 2020.
- [61] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.
- [62] Ping Tan, Satya P Mallick, Long Quan, David J Kriegman, and Todd Zickler. Isotropy, reciprocity and the generalized bas-relief ambiguity. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [63] Tatsunori Taniai and Takanori Maehara. Neural inverse rendering for general reflectance photometric stereo. In *International Conference on Machine Learning (ICML)*, pages 4857–4866, 2018.
- [64] Olivia Wiles and Andrew Zisserman. Silnet: Single- and multi-view reconstruction by learning from silhouettes. *arXiv preprint arXiv:1711.07888*, 2017.
- [65] Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139, 1980.
- [66] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019.
- [67] Lun Wu, Arvind Ganesh, Boxin Shi, Yasuyuki Matsushita, Yongtian Wang, and Yi Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *Asian Conference on Computer Vision*, pages 703–717. Springer, 2010.
- [68] Yan Wu, Zhiwu Huang, Suryansh Kumar, Rhea Sanjay Sukthanker, Radu Timofte, and Luc Van Gool. Trilevel neural architecture search for efficient single image super-resolution. *arXiv preprint arXiv:2101.06658*, 2021.
- [69] Zhe Wu and Ping Tan. Calibrating photometric stereo by holistic reflectance symmetry analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1505, 2013.
- [70] Wuyuan Xie, Chengkai Dai, and Charlie CL Wang. Photometric stereo with near point lighting: A solution by mesh deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4585–4593, 2015.
- [71] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- [72] Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhen-guo Li. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6649–6658, 2019.
- [73] Zhuokun Yao, Kun Li, Ying Fu, Haofeng Hu, and Boxin Shi. Gps-net: Graph-based photometric stereo network. *Advances in Neural Information Processing Systems*, 33, 2020.
- [74] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.
- [75] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 8697–8710, 2018.