DeFeat-Net: General Monocular Depth via Simultaneous Unsupervised Representation Learning

Jaime Spencer, Richard Bowden, Simon Hadfield
Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey
{jaime.spencer, r.bowden, s.hadfield}@surrey.ac.uk

Abstract

In the current monocular depth research, the dominant approach is to employ unsupervised training on large datasets, driven by warped photometric consistency. Such approaches lack robustness and are unable to generalize to challenging domains such as nighttime scenes or adverse weather conditions where assumptions about photometric consistency break down.

We propose DeFeat-Net (Depth & Feature network), an approach to simultaneously learn a cross-domain dense feature representation, alongside a robust depth-estimation framework based on warped feature consistency. The resulting feature representation is learned in an unsupervised manner with no explicit ground-truth correspondences required.

We show that within a single domain, our technique is comparable to both the current state of the art in monocular depth estimation and supervised feature representation learning. However, by simultaneously learning features, depth and motion, our technique is able to generalize to challenging domains, allowing DeFeat-Net to outperform the current state-of-the-art with around 10% reduction in all error measures on more challenging sequences such as nighttime driving.

1. Introduction

Recently there have been many advances in computer vision tasks related to autonomous vehicles, including monocular depth estimation [22, 83, 73] and feature learning [13, 61, 65]. However, as shown in Figure 1, these approaches tend to fail in the most complex scenarios, namely adverse weather and nighttime conditions.

In the case of depth estimation, this is usually due to the assumption of photometric consistency, which starts to break down in dimly-lit environments. Feature learning can overcome such strong photometric assumptions, but these approaches tend to require ground truth pixel-wise correspondences and obtaining this ground truth in cross-seasonal situations is non-trivial. Inconsistencies between GPS measurements and drift from Visual Odometry (VO) makes automatic pointcloud alignment highly inaccurate and manual annotation is costly and time-consuming.

We make the observation that depth estimation and feature representation are inherently complementary. The process of estimating the depth for a scene also allows for the computation of ground-truth feature matches between any views of the scene. Meanwhile robust feature spaces are necessary in order to create reliable depth-estimation systems with invariance to lighting and appearance change.

Despite this relationship, all existing approaches tackle these challenges independently. Instead, we propose DeFeat-Net, a system that is capable of jointly learning depth from a single image in addition to a dense feature representation of the world and ego-motion between consecutive frames. What’s more, this is achieved in an entirely self-supervised fashion, requiring no ground truth other than a monocular stream of images.

We show how the proposed framework can use the existing relationships between these tasks to complement each other and boost performance in complex environments. As has become commonplace [23], the predicted depth and ego-motion can be used to generate a correspondence map.
between consecutive images, allowing for the use of photometric error based losses. However, these correspondences can also be used as positive examples in relative metric learning losses [65]. In turn, the learnt features can provide a more robust loss in cases where photometric errors fail, i.e. nighttime conditions.

The remainder of the paper provides a more detailed description of the proposed DeFeat-Net framework in the context of previous work. We extensively show the benefits of our joint optimization approach, evaluating on a wide variety of datasets. Finally, we discuss the current state-of-the-art and opportunities for future work. The contributions of this paper can be summarized as:

1. We introduce a framework capable of jointly and simultaneously learning monocular depth, dense feature representations and vehicle ego-motion.

2. This is achieved entirely self-supervised, eliminating the need for costly and unreliable ground truth data collection.

3. We show how the system provides robust depth and invariant features in all weather and lighting conditions, establishing new state-of-the-art performance.

2. Related Work

Here we review some of the most relevant previous work, namely in depth estimation and feature learning.

2.1. Depth Estimation.

Traditionally, depth estimation relied on finding correspondences between every pixel in pairs of images. However, if the images have been stereo rectified, the problem can be reduced to a search for the best match along a single row in the target image, known as disparity estimation. Initial methods for disparity estimation relied on hand-crafted matching techniques based on the Sum of Squared Differences (SSD), smoothness and energy minimization.

Supervised. Ladický [33] and Žbontar [79] showed how learning the matching function can drastically improve the performance of these systems. Mayer et al. [46] instead proposed DispNet, a Fully Convolutional Network (FCN) [40] capable of directly predicting the disparity map between two images, which was further extended by [50]. Kendall et al. [30] introduced GC-Net, where the disparity is processed as a matching cost-volume in a 3D convolutional network. PSMNet [9] and GA-Net [81] extended these cost-volume networks by introducing Spatial Pooling Pyramid (SPP) features and Local/Semi-Global aggregation layers, respectively.

Estimating depth from a single image seemed like an impossible task without these disparity and perspective cues. However, Saxena [58] showed how it is possible to approximate the geometry of the world based on superpixel segmentation. Each superpixel’s 3D position and orientation is estimated using a trained linear model and an MRF. Liu et al. [38, 39] improve on this method by instead learning these models using a CNN, while Ladický et al. [34] incorporate semantic information as an alternative cue.

Eigen et al. [14, 15] introduced the first methods for monocular depth regression using end-to-end deep learning by using a scale-invariant loss. Laina [35] and Cao [7] instead treated the task of monocular estimation as a classification problem and introduced a more robust loss function. Meanwhile, Ummenhofer et al. [66] introduced DeMoN, jointly training monocular depth and egomotion in order to perform Structure-from-Motion (SfM). In this paper we go one step further, jointly learning depth, egomotion and the feature space used to support them.

Unsupervised - Stereo Training. In order to circumvent the need for costly ground truth training data, an increasing number of approaches have been proposed using photometric warp errors as a substitute. For instance, DeepStereo [17] synthesizes novel views using raw pixels from arbitrary nearby views. Deep3D [74] also performs novel view synthesis, but restricts this to stereo pairs and introduces a novel image reconstruction loss. Garg [18] and Godard [23] greatly improved the performance of these methods by introducing an additional autoencoder and left-right consistency losses, respectively. UnDeepVO [37] additionally learns monocular VO between consecutive frames by aligning the predicted depth pointclouds and enforcing consistency between both stereo streams. More recently, there have been several approaches making use of GANs [1, 53]. Most notably, [62] uses GANs to perform day-night translation and provide an additional consistency to improve performance in nighttime conditions. However, the lack of any explicit feature learning makes it challenging to generalize across domains.

Unsupervised - Monocular Training. In order to learn unsupervised monocular depth without stereo information, it is necessary to learn a surrogate task that allows for the use of photometric warp losses. Zhou et al. [82, 83] introduced some of the first methods to make use of VO estimation to warp the previous and next frames to reconstruct the target view. Zhan [80] later extended this by additionally incorporating a feature based warp loss. Babu et al. [3, 44] proposed an unsupervised version of DeMoN [66]. Other published methods are based upon video processing with RNNs [69] and LSTMs [51] or additionally predicting scene motion [67] or optical flow [29, 70, 78].

The current state-of-the-art has been pushed by methods that incorporate additional constraints [68] such as temporal [45], semantic [10], edge & normal [75, 76], cross-task [84] and cycle [52, 73] consistencies. Godard et al. [22]
expanded on these methods by incorporating information from the previous frame and using the minimum reprojection error in order to deal with occlusions. They also introduce an automasking process which removes stationary pixels in the target frame. However, they still compute photometric losses in the original RGB colourspace, making it challenging to learn across domains.

2.2. Feature Learning

Hand-Crafted. Initial approaches to feature description typically relied on heuristics based on intensity gradients in the image. Since these were computationally expensive, it became necessary to introduce methods capable of finding interesting points in the image, i.e. keypoints. Some of the most well-known methods include SIFT [41] and its variant RootSIFT [2], based on a Difference of Gaussians and Non-Maxima Suppression (NMS) for keypoint detection and HOG descriptors.

Research then focused on improving the speed of these systems. Such is the case with SURF [5], BRIEF [6] and BRISK [36]. ORB features [56] improved the accuracy, robustness and speed of BRIEF [6] and are still widely used.

Sparse Learning. Initial feature learning methods made use of decision trees [55], convex optimization [63] and evolutionary algorithms [31, 32] in order to improve detection reliability and discriminative power. Intelligent Cost Functions [24] took this a step further, by using Gaussian Processes to learn appropriate cost functions for optical/scene flow.

Since the widespread use of deep learning, several methods have been proposed to learn feature detection and/or description. Balntas et al. [4] introduced a method for learning feature descriptors using in-triplet hard negative mining. LIFT [77] proposes a sequential pipeline consisting of keypoint detection, orientation estimation and feature description, each performed by a separate network. LF-Net [49] builds on this idea, jointly generating dense score and orientation maps without requiring human supervision.

On the other hand, several approaches make use of networks with shared encoder parameters in order to simultaneously learn feature detection and description. Georgakis et al. [20] learn 3D interest points using a shared Fast R-CNN [21] encoder. Meanwhile, DeTone introduced SuperPoint [12] where neither decoder has trainable parameters, improving the overall speed and computational cost. More recently, D2-Net [13] proposed a describe-then-detect approach where the network produces a dense feature map, from which keypoints are detected using NMS.

Dense Learning. Even though SuperPoint [12] and D2-Net [13] produce dense feature maps, they still focus on the detection of interest points and don’t use their features in a dense manner. Weerasekera et al. [72] learn dense features in the context of SLAM by minimizing multi-view matching cost-volumes, whereas [60] use generative feature learning with scene completion as an auxiliary task to perform visual localisation.

The Universal Correspondence Network [11] uses optical correspondences to create a pixel-wise version of the contrastive loss. Schmidt [59] instead propose semi-supervised training with correspondences obtained from KinectFusion [47] and DynamicFusion [48] models. Fathy [16] and Spencer [65] extended the pixel-wise contrastive loss to multiple scale features through a coarse-to-fine network and spatial negative mining, respectively. On the other hand, SDC-Net [61] focuses on the design of the network architecture, increasing the receptive field through stacked dilated convolution, and apply the learnt features to optical flow estimation.

In this work we attempt to unify state-of-the-art feature learning with monocular depth and odometry estimation. This is done in such a way that the pixel-wise correspondences from monocular depth estimation can support dense feature learning in the absence of ground-truth labels. Meanwhile, computing match-costs in the learned feature space greatly improves the robustness of the depth estimation in challenging cross-domain scenarios.

3. Methodology

The main objective of DeFeat-Net is to jointly learn monocular depth and dense features in order to provide more robust estimates in adverse weather conditions. By leveraging the synergy between both tasks we are able to do this in a fully self-supervised manner, requiring only a monocular stream of images. Furthermore, as a byproduct of the training losses, the system additionally learns to predict VO between consecutive frames.

Figure 2 shows an overview of DeFeat-Net. Each training sample is composed of a target frame \( I_t \) and a set of support frames \( I_{t+k} \), where \( k \in \{-1, 1\} \). Using the predicted depth for \( I_t \) and the predicted transforms to \( I_{t+k} \) we can obtain a series of correspondences between these images, which in turn can be used in the photometric warp and pixel-wise contrastive losses. The code and pre-trained models for this technique will be available at https://github.com/jspenmar/DeFeat-Net.

3.1. Networks

DispNet. Given a single input image, \( I_t \), its corresponding depth map is obtained through

\[
D_t = \frac{1}{a \Phi_D(I_t) + b}.
\]

where \( a \) and \( b \) scale the final depth to the range \([0.1, 100] \), \( \Phi_D \) represents the disparity estimation network, formed by a ResNet [25] encoder and decoder with skip connections.
This decoder also produces intermediate disparity maps at each stage, resulting in four different scales.

**PoseNet.** Similarly, the pose prediction network $\Phi_P$ consists of a multi-image ResNet encoder, followed by a 4-layer convolutional decoder. Formally,

$$
P_{t\rightarrow t+k} = \Phi_P(I_t, I_{t+k}),$$

where $P_{t\rightarrow t+k}$ is the predicted transform between the cameras at times $t$ and $t+k$. As in [22, 68] the predicted pose is composed of a rotation in axis-angle representation and a translation vector, scaled by a factor of 0.001.

**FeatNet.** The final network produces a dense n-dimensional feature map of the given input image, $\Phi_F : \mathbb{N}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times n}$. As such, we define the corresponding L2-normalized feature map as

$$
F = ||\Phi_F(I)||. \tag{3}
$$

In this case, $\Phi_F$ is composed of a residual block encoder decoder with skip connections, where the final encoder stage is made up of an SPP [9] with four scales.

### 3.2. Correspondence Module

Using the predicted $D_t$ and $P_{t\rightarrow t+k}$ we can obtain a set of pixel-wise correspondences between the target frame and each of the support frames. Given a 2D point in the image $p$ and its homogeneous coordinates $\hat{p}$ we can obtain its corresponding location $q$ in the 3D world through

$$
q = \pi^{-1}(\hat{p}) = K_t^{-1} \hat{p} D_t(p), \tag{4}
$$

where $\pi^{-1}$ is the backprojection function, $K_t$ is the camera’s intrinsics and $D_t(p)$ the depth value at the 2D pixel location estimated using (1).

We can then compute the corresponding point $c_{t\rightarrow t+k}$ by projecting the resulting 3D point onto a new image with

$$
c_{t\rightarrow t+k}(p) = \pi(\hat{q}) = K_t P_{t\rightarrow t+k} q, \tag{5}
$$

where $P_{t\rightarrow t+k}$ is the transform to the new coordinate frame, i.e. the next or previous camera position from (2). Therefore, the final correspondences map is defined as

$$
C_{t\rightarrow t+k} = \{c_{t\rightarrow t+k}(p) : \forall p\}. \tag{6}
$$

These correspondences can now be used in order to determine the sampling locations for the photometric warp loss and the positive matches in a pixel-wise contrastive loss to learn an appropriate feature space.

### 3.3. Losses

Once again, it is worth noting that DeFeat-Net is entirely self-supervised. As such, the only ground truth inputs required are the original images and the camera’s intrinsics.

**Pixel-wise Contrastive.** In order to train $\Phi_F$, we make use of the well established pixel-wise contrastive loss [11, 59, 65]. Given two feature vectors from the dense feature maps, $f_1 = F_1(p_1)$ and $f_2 = F_2(p_2)$, the contrastive loss is defined as

$$
l(y, f_1, f_2) = \begin{cases} 
\frac{1}{2}(d)^2 & \text{if } y = 1 \\
\frac{1}{2}\left(\max(0, m - d)\right)^2 & \text{if } y = 0 \\
0 & \text{otherwise} 
\end{cases} \tag{7}
$$

with $y$ as the label indicating if the pair is a correspondence, $d = ||f_1 - f_2||$ and $m$ the target margin between negative pairs. In this case, the set of positive correspondences is
given by $C_{t\rightarrow t+k}$. Meanwhile, the negative examples are generated using one of the spatial negative mining techniques from [65].

From both sets, a label mask $Y$ is created indicating if each possible pair of pixels is a positive, negative or should be ignored. As such, the final loss is defined as

$$L_C = \sum_{p_1} \sum_{p_2} l(Y(p_1, p_2), F_t(p_1), F_{t+k}(p_2)).$$

This loss serves to drive the learning of a dense feature space which enables matching regardless of weather and seasonal appearance variations.

**Photometric and Feature Warp.** We also use the correspondences in a differentiable bilinear sampler [28] in order to generate the warped support frames and feature maps

$$I_{t+k\rightarrow t} = I_{t+k}(C_{t\rightarrow t+k}),$$

$$F_{t+k\rightarrow t} = F_{t+k}(C_{t\rightarrow t+k}),$$

where $\langle \rangle$ is the sampling operator. The final warp losses are a weighted combination of SSIM [71] and L1, defined by

$$\Psi(I_1, I_2) = \alpha \frac{1 - \text{SSIM}(I_1, I_2)}{2} + (1 - \alpha) ||I_1 - I_2||.$$  

$$L_P = \Psi(I_t, I_{t+k\rightarrow t}),$$  

$$L_F = \Psi(F_t, F_{t+k\rightarrow t}),$$

The photometric loss $L_P$ serves primarily to support the early stages of training when the feature space is still being learned.

**Smoothness.** As an additional regularizing constraint, we incorporate a smoothness loss [27]. This enforces local smoothness in the predicted depths proportional to the strength of the edge in the original image, $\partial I_t$. This is defined as

$$L_S = \frac{\lambda}{N} \sum_p |\partial D_t(p)| e^{-||\partial I_t(p)||}.$$  

where $\lambda$ is a scaling factor typically set to 0.001. This loss is designed to avoid smoothing over edges by reducing the weighting in areas of strong intensity gradients.

### 3.4. Masking & Filtering

Some of the more recent improvements in monocular depth estimation have arisen from explicit edge-case handling [22]. This includes occlusion filtering and the masking of stationary pixels. We apply these automatic procedures to the correspondences used to train both the depth and dense features.

**Minimum Reprojection.** As the camera capturing the monocular stream moves throughout the scene, various elements will become occluded and disoccluded. In terms of a photometric error based loss, this means that some of the correspondences generated by the system will be invalid. However, when multiple consecutive frames are being used, i.e. $k \in \{-1, 1\}$, different occlusions occur in each image.

By making the assumption that the photometric error will be greater in the case where an occlusion is present, we can filter these out by simply propagating the correspondence with the minimum error. This is defined as

$$C_{t\rightarrow t+k} = \begin{cases} C_{t\rightarrow t-1} & \text{where } \Psi(I_t, I_{t\rightarrow t-1}) < \Psi(I_t, I_{t\rightarrow t+1}) \\ C_{t\rightarrow t+1} & \text{otherwise} \end{cases}$$

**Automasking.** Due to the nature of the training method and implicit depth priors (i.e. regions further away change less) stationary frames or moving objects can cause holes of infinite depth in the predicted depth maps. An automasking procedure is used to remove these stationary pixels from contributing to the loss,

$$\mu = \left[ \min_k \Psi(I_t, I_{t+k}) < \min_k \Psi(I_t, I_{t+k\rightarrow t}) \right],$$

where $\mu$ is the resulting mask indicating if a correspondence is valid or not and $\llbracket \rrbracket$ is the Iverson bracket. In other words, pixels that exhibit lower photometric error to the unwarped frame than to the warped frame are masked from the cost function.

### 4. Results

Each subsystem in DeFeat-Net follows a U-Net structure with a ResNet18 encoder pretrained on ImageNet, followed by a 7 layer convolutional decoder similar to [23]. The code and pre-trained models will be available at https://github.com/jspenmar/DeFeat-Net. In all our experiments, the warp loss parameter is set to $\alpha = 0.85$ as per [28].

On the KITTI dataset [19] we follow the Eigen-Zhou evaluation protocol of [23, 83]. This dataset split provides 39,810 training images and 4,424 validation images. These images are all from a single domain (sunny daytime driving).

We also make use of the RobotCar Seasons dataset [57]. This is a curated subset of the larger RobotCar dataset [43], containing 49 sequences. The dataset was specifically chosen to cover a wide variety of seasons and weather conditions, leading to greater diversity in appearance than KITTI.

Unlike the KITTI dataset, which provides sparse ground-truth depth from LiDAR, RobotCar Seasons does not include any depth ground truth. Our proposed technique is unsupervised, and can still be trained on this varied dataset, but the lack of ground truth makes quantitative evaluation on RobotCar Seasons impossible. To resolve this, we returned to the original RobotCar dataset and manually created a validation dataset comprising of 12,000 images with
Table 1. Monocular depth evaluation on the KITTI dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Abs-Rel</th>
<th>Sq-Rel</th>
<th>RMSE</th>
<th>RMSE-log</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEGO [75]</td>
<td>0.162</td>
<td>1.352</td>
<td>6.276</td>
<td>0.252</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ranjan [54]</td>
<td>0.148</td>
<td>1.149</td>
<td>5.464</td>
<td>0.226</td>
<td>0.815</td>
<td>0.935</td>
<td>0.973</td>
</tr>
<tr>
<td>EPC++ [42]</td>
<td>0.141</td>
<td>1.029</td>
<td>5.350</td>
<td>0.216</td>
<td>0.816</td>
<td>0.941</td>
<td>0.976</td>
</tr>
<tr>
<td>Struct2depth (M) [8]</td>
<td>0.141</td>
<td>1.026</td>
<td>5.291</td>
<td>0.215</td>
<td>0.816</td>
<td>0.945</td>
<td>0.979</td>
</tr>
<tr>
<td>Monodepth V2 [22]</td>
<td>0.123</td>
<td>0.944</td>
<td>5.061</td>
<td>0.197</td>
<td>0.866</td>
<td>0.957</td>
<td>0.980</td>
</tr>
<tr>
<td>DeFeat</td>
<td>0.126</td>
<td>0.925</td>
<td>5.035</td>
<td>0.200</td>
<td>0.862</td>
<td>0.954</td>
<td>0.980</td>
</tr>
</tbody>
</table>

Table 2. Learned feature evaluation on the KITTI dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>µ+</th>
<th>Global µ−</th>
<th>Global AUC</th>
<th>Local µ−</th>
<th>Local AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORB [56]</td>
<td>N/A</td>
<td>N/A</td>
<td>85.83</td>
<td>N/A</td>
<td>84.06</td>
</tr>
<tr>
<td>ResNet [26]</td>
<td>8.5117</td>
<td>25.9872</td>
<td>94.77</td>
<td>11.1335</td>
<td>68.26</td>
</tr>
<tr>
<td>ResNet-L2</td>
<td>0.341</td>
<td>1.0391</td>
<td>99.25</td>
<td>0.4371</td>
<td>71.80</td>
</tr>
<tr>
<td>VGG [64]</td>
<td>4.0077</td>
<td>12.6543</td>
<td>92.94</td>
<td>5.9088</td>
<td>70.03</td>
</tr>
<tr>
<td>VGG-L2</td>
<td>0.3905</td>
<td>1.2235</td>
<td>99.57</td>
<td>0.565</td>
<td>77.06</td>
</tr>
<tr>
<td>SAND-G [65]</td>
<td>0.093</td>
<td>0.746</td>
<td>99.73</td>
<td>0.266</td>
<td>87.06</td>
</tr>
<tr>
<td>SAND-L</td>
<td>0.156</td>
<td>0.592</td>
<td>98.88</td>
<td>0.505</td>
<td>94.34</td>
</tr>
<tr>
<td>SAND-GL</td>
<td>0.183</td>
<td>0.996</td>
<td>99.28</td>
<td>0.642</td>
<td>93.34</td>
</tr>
<tr>
<td>DeFeat</td>
<td>0.105</td>
<td>1.113</td>
<td>99.10</td>
<td>0.294</td>
<td>83.64</td>
</tr>
</tbody>
</table>

4.1. Single Domain Evaluation

We first evaluate our approach on the KITTI dataset, which covers only a single domain. For evaluation of depth accuracy, we use the standard KITTI evaluation metrics, namely the absolute relative depth error (ABSOREL), the relative square error (SQREL) and the root mean square error (RMSE). For these measures, a lower number is better. We also include the inlier ratio measures (A1, A2 and A3) of [23] which measure the fraction of relative depth errors within 25%, 25% and 25% of the ground truth. For these measures, a larger fraction is better.

To evaluate the quality of the learned feature representations, we follow the protocol of [65]. We compute the average distance in the feature space for the positive pairs from the ground-truth (µ+), and the negative pairs (µ−). Naturally a smaller distance between positive pairs, and a larger distance between negative pairs, is best. We also compute the Area Under the Curve (AUC) which can be interpreted as the probability that a randomly chosen negative sample will have a larger distance than the corresponding positive ground truth match. Therefore, higher numbers are better. Following [65] all three errors are split into both local (within 25 pixels) and global measurements.

The results of the depth evaluation are shown in Table 1 and the feature evaluation is shown in Table 2. We can see that in this single-domain scenario, the performance of our technique is competitive with MonodepthV2 and clearly outperforms most other state-of-the-art techniques for monocular depth estimation. The results for [22] were obtained by training a network using the code provided by the authors.

Regarding the features, L2 denotes the L2-normalized versions, whereas G, L & GL represent the different negative mining variants from [65]. We can also see that despite being unsupervised, our learned feature space is competitive with contemporary supervised feature learning techniques and greatly outperforms pretrained features when evaluating locally. It is interesting, however, to note that the simple act of L2-normalizing can improve the global performance of the pretrained features.

Our feature space tends to perform better in the Global evaluation metrics than the local ones. This is unsurprising as the negative samples for the contrastive loss in (7) are obtained globally across the entire image.

4.2. Multi-Domain Evaluation

However, performance in the more challenging RobotCar Seasons dataset demonstrates the real strength of jointly learning both depth and feature representations. RobotCar Seasons covers multiple domains, where traditional photometric based monocular depth algorithms struggle and where a lack of cross-domain ground-truth has historically made feature learning a challenge. For this evaluation, we select the best competing approach from Table 1 (MonodepthV2) and retrain both it and DeFeat-Net on the RobotCar Seasons dataset. All techniques are trained from scratch.

The results are shown in Table 3 and example depth map comparisons are shown in Figure 3. We can see that in this more challenging task, the proposed approach outperforms the previous state of the art technique across all error measures. While for the daytime scenario, the improvements are modest, on the nighttime data there is a significant improvement with around 10% reduction in all error measures.

We believe that the main reason behind this difference is...
that in well-lit conditions, the photometric loss is already a good supervision signal. In this case, incorporating the feature learning adds to the complexity of the task. However, nighttime scenarios make photometric matching less discriminative, leading to weaker supervision. Feature learning provides the much needed invariance and robustness to the loss, leading to the significant increase in performance.

It is interesting to note that the proposed approach is especially robust with regards to the number of estimated outliers. The A1, A2 and A3 error measures are fairly consistent between the day and night scenarios for the proposed technique. This indicates that even in areas of uncertain depth (due to under-exposure and over-saturation), the proposed technique fails gracefully rather than producing catastrophically incorrect estimates.

Since previous state-of-the-art representations cannot be trained unsupervised, and RobotCar Seasons does not provide any ground-truth depth, it is not possible to repeat the feature comparison from Table 2 in the multi-domain scenario. Instead Figure 4 compares qualitative examples of the learned feature spaces. For these visualizations, we find the linear projection that best shows the correlation between the feature map and the images and map it to the RGB color cube. This dimensionality reduction removes a significant amount of discriminative power from the descriptors, but allows for some form of visualization.

In all cases, the feature descriptors can clearly distinguish scene structures such as the road. It is interesting to note that a significant degree of context has been encoded in the features, and they are capable of easily distinguishing a patch in the middle of the road, from one on the left or right, and from a patch of similarly colored pavement. The feature maps trained on the single domain KITTI dataset can sometimes display more contrast than those trained on RobotCar Seasons. Although this implies a greater degree of discrimination between different image regions, this is likely because the latter representation can cover a much broader range of appearances from other domains. Regarding, the nighttime features, it is interesting that those trained on a single domain seem to exhibit strange behaviour around external light sources such as the lampposts, traffic lights and headlights. This is likely due to the bias in the training data, with overall brighter image content.

### 4.3. Ablation

Finally, for each dataset we explore the benefits of concurrent feature learning, by re-training with the FeatNet subsystem disabled. As shown in Table 4, the removal of
the concurrent feature learning from our technique causes a small and inconsistent change on the KITTI and RobotCar Day data. However, on the RobotCar Night data, our full approach drastically outperforms the version which does not learn a specialist matching representation. For many error measures, the performance doubles in these challenging scenarios, and the reduction in outliers causes a three-fold reduction in the Sq-Rel error.

These findings reinforce the observation that the frequently used photometric warping loss is insufficient for estimating depth in challenging real-world domains.

5. Conclusions & Future Work

This paper proposed DeFeat-Net, a unified framework for learning robust monocular depth estimation and dense feature representations. Unlike previous techniques, the system is able to function over a wide range of appearance domains, and can perform feature representation learning with no explicit ground truth. This idea of co-training an unsupervised feature representations has potential applications in many areas of computer vision beyond monocular depth estimation.

The main limitation of the current approach is that there is no way to enforce feature consistency across seasons. Although depth estimation and feature matching work robustly within any given season, it is currently unclear whether feature matching between different seasons is possible. It would be interesting in the future to explore cross-domain consistency as an additional training constraint. However, this will necessitate the collection of new datasets with cross seasonal alignments.

Acknowledgements

This work was partially funded by the EPSRC under grant agreements (EP/R512217/1, EP/S016317/1 and EP/S035761/1). We would also like to thank NVIDIA Corporation for their Titan Xp GPU grant.
References


[52] Andrea Pilzer, Stéphane Lathuilière, Nicu Sebe, and Elisa Ricci. Refine and Distill: Exploiting Cycle-Inconsistency


[56] Ethan Rublee and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. ICCV, 2011.


