Distribution Shift Detection in Monocular Depth Estimation for Autonomous Systems

1st Anestis Christidis
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
canestist@auth.gr

2nd Evangelos Charalampakis
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
evangelc@csd.auth.gr

3d Ioannis Mademlis
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
imademlis@csd.auth.gr

4th Ioannis Pitas
School of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
pitas@csd.auth.gr

***Abstract*—Distribution Shift Detection (DSD) is utilized when a pretrained machine learning model deployed in-the-field has to analyze test-phase inputs drawn from a data distribution that may differ from that of the training dataset. DSD algorithms for Deep Neural Networks (DNNs) attempt to identify Out-of-Distribution (OOD) test samples, so as to avoid incorrect decisions at the inference stage. However, like effort has been expended to developing methods for DSD in monocular depth estimation (MDE), which is a safety-critical task in autonomous systems such as self-driving cars. This paper presents a novel DSD method for MDE DNNs. The outputs of a pretrained MDE are used as ground-truth to train an Image-to-Image Translator (I2I) that maps intermediate MDE representations to depth maps. During deployment, both the MDE and the I2I estimate a depth map from each test-stage image. The underlying intuition is that the MDE and the I2I outputs are in little agreement for OOD images, due to their different neural architectures and learning paradigms. Extensive experimental evaluation on autonomous driving-compatible datasets showcases that the proposed method significantly surpasses competing distribution shift detectors for MDE tasks.**

***Index Terms*—Distribution Shift Detection, Monocular Depth Estimation, Autonomous Vehicles, Generative Adversarial Network, Self-Supervised Learning**

I. Introduction

Extensive effort is typically expended on pretraining Deep Neural Networks (DNNs) so that they achieve the best possible accuracy and performance during deployment. However, real-world environments are often unpredictable and ever-changing: there can be factors unconsidered during training, or slight variations in the inputs, that may lead the DNN to false conclusions with high confidence [21]. The penalties can vary greatly, especially in autonomous driving where significant material losses might occur, or even human harm. In fact, the rapid emergence of a variety of autonomous systems operating in the real world [27] [26] [16] [19] [28] [34] [1] [8] has made this a critical, high-priority issue.

The need for a pretrained system that can maneuver through unfamiliar inputs has fueled research on Distribution Shift Detectors (DSD) [14]. Such methods are expected to successfully discern In-Distribution (ID) test-stage data points from Out-of-Distribution (OOD) ones, during model deployment, so as to avoid erroneous inference for OOD inputs. The underlying distribution of such OOD test-stage data points essentially differs from that of the training set. The importance of DSD in autonomous systems is rather clear, since classifying an unknown data point into one of the known classes with high confidence may cause significant negative effects in the real world.

An example would be dense image segmentation tasks, where a different label is assigned per-pixel [3], but also other types of dense image prediction tasks, such as monocular depth estimation (MDE) [11] [10] [12] [24] [25]. Given that Deep Neural Networks (DNNs) are becoming more and more common in depth estimation from visual data, this is a highly safety-critical area for robotics and autonomous vehicles. In such applications, the sense of depth plays a major role in environmental perception and safe navigation. A MDE DNN typically analyzes an input image (e.g., an RGB video frame captured on-the-fly from the camera of an autonomous system) and outputs a depth map with the same spatial resolution in pixels. Each output pixel value is a normalized distance from the camera.

Most relevant work up to now handles uncertainty estimation on depth maps by assessing depth prediction confidence [29] [36] [18]. However, it is not necessary to actually compute a formal prediction uncertainty score, which is a rather difficult task, since the only practically useful information is the final OOD or ID label assigned to the entire image. Thus, this paper presents a simpler, yet effective method for DSD in MDE DNNs.

In the proposed approach, the outputs of a pretrained MDE are used as ground-truth to train an Image-to-Image Translator (I2I) [15], that maps intermediate/internal MDE representations of the input images to depth maps. During deployment, both the MDE and the I2I estimate a depth map from each test-stage image. The aggregated per-pixel

absolute difference between the two outputs can be viewed as an informal uncertainty index that increases for unfamiliar inputs. The underlying intuition is that the MDE and the I2I outputs are less correlated for OOD images, due to their different neural architectures and learning paradigms. A final OOD decision is made by thresholding this index.

In short, this paper contributes the following:

1) It proposes a novel, simple but effective algorithm to compute informal, pixel-level prediction uncertainty estimates for pretrained MDE DNNs.
2) It exploits these per-pixel informal uncertainty estimates to classify the entire input image as OOD or ID, through aggregation and thresholding.

Experimental evaluation on common autonomous driving datasets demonstrates the advantages of the proposed method.

II. Related Work

Typical DSD methods for classification tasks fall under either the *discriminative* or the *generative* family. Both approaches rely on comparing test data points with the training dataset, irrespective of the actual DNN model being employed during inference. A more modern approach is to design model-specific variants of similar discriminative or generative models, where the representations of the input data points constructed internally by the classifier DNN may also be used for distinguishing between ID and OOD data. Additionally, several different methods have been proposed that rely on directly or indirectly measuring *epistemic uncertainty* (i.e., uncertainty due to lack of relevant knowledge encoded in the model parameters) for each test-stage prediction.

DSD [30] [20] [5] [23], which concerns domain/data-space distributions shifts, can be contrasted with Out-of-Distribution Detection (OODD): a similar problem that concerns label-space distribution shifts in classification tasks. In that case, the OOD data points belong to classes unseen during training (e.g., training with images of people and testing with images of animals). In contrast, in domain shifts the OOD data points come from classes encountered at the training stage but altered due to test-stage noise patterns (e.g., training with sunny images and testing with foggy ones).

One notable DSD method for semantic segmentation DNNs is [22], which proposes MetaSeg: a DNN designed to successfully detect OOD samples in semantic segmentation tasks. MetaSeg is tasked to detect segments of high-interest classes but of low Intersection-over-Union (IoU) values. Alternatively, [4] relies on spatial entropy heatmaps and pixel-level dispersion metrics for identifying OOD images. During training, the softmax output of the DNN is considered to be a set of pixel-wise probability distributions that express per-class affiliation for each pixel of an input image. Then, by combining neighboring pixels that their normalized entropy exceeds a theshold, OOD segments/ojects are formed.

Moving on to Monocular Depth Estimation (MDE), only few published methods specifically target DSD/DSD; instead attempt to quantify depth prediction uncertainty. For example [2] exploits visual odometry data to train an autoencoder (AE) alongside a depth estimator. Training consistency is enforced by using two instances of the AE, for both left and right stereo images, to produce denser depth maps than the input ones. The depth estimator, with the appropriate loss functions, produces the final depth maps using the autoencoder's output.

In [29] two instances of the same MDE DNN, i.e., Monodepth2 [11], are optimized in a teacher-student manner: the teacher is trained regularly on its own, while its outputs are employed as ground-truth for optimizing the student with a directly supervised learning approach. Uncertainty estimation can then proceed by decoupling depth estimation from the camera pose.

A different DSD detection method is the Self-Oracle Auto-Encoder, which detects distributional shifts by comparing the input and the output of a Variational Auto-Encoder (VAE), so as to assess the reconstruction error [33]. Then, [35] put forward the Likelihood-Regret (LR) metric, which estimates the log-likelihood model improvement that maximizes the likelihood of one sample over the model configuration that maximizes the likelihood across all samples, deeming anomalous samples that cause large LR fluctuations. Alternatively, the Self-Supervised outlier Detection (SSD) framework [32] uses a self-supervised generative model to extract latent feature representations from depth maps, using Mahalanobis distance to compare test samples against established ID ones. Finally, [13] evaluates the three previous methods with respect to their DSD performance on a MDE DNN. Experiments involve synthetic autonomous vehicle footage with various degrees of fog causing distribution shift. SSD proved to perform best, with its output scores reflecting the actual level of distributional shift.

The proposed method is most closely related to [29], which also employs teacher-student training. However, instead of utilizing two architecturally identical instances of a single MDE and exploiting the difference of their test-stage outputs to construct a formal prediction uncertainty estimation model, our method operates in a very different manner. It exploits two entirely different architectures, i.e., a regular MDE and an I2I, in order to directly use the difference of their test-stage outputs as an informal uncertainty index. This index is then thresholded to decide whether the input test image is OOD or ID. The underlying intuition differs significantly from that of [29]: our DSD method relies on the idea that the MDE and the I2I outputs are less correlated for OOD images, due to their different neural architectures and learning paradigms. This is simpler, more intuitive and more effective for DSD in MDE DNNs, as shown by the experimental evaluation.

III. Proposed Method

The proposed method exploits an Image-to-Image Translator (I2I) in order to achieve DSD during the inference stage of a Monocular Depth Estimator (MDE). Thus, MDE and first briefly reviewed, before our novel algorithm itself is presented.

A. Monocular Depth Estimation

MDE is typically achieved by simple supervised image regression, when ground-truth depth maps are available, or

by self-supervised learning that exploits geometric relations inherent in consecutive video frames or in stereoscopic 3D image pairs. Below, the self-supervised setting is briefly reviewed.

A typical loss function [11] for training a self-supervised MDE DNN mainly involves a photometric reprojection error L_p:

$$L_p = \min_{t'} pe(\mathbf{I}_t, \mathbf{I}_{t'\to t}) \tag{1}$$

$$\mathbf{I}_{t'\to t} = \mathbf{I}_{t'} \langle proj(\mathbf{D}_t, \mathbf{T}_{t'\to t}, \mathbf{K}) \rangle \tag{2}$$

where pe is a photometric reconstruction error, and can be calculated as:

$$pe(\mathbf{I}_A, \mathbf{I}_B) = \frac{a}{2}(1-SSIM(\mathbf{I}_A, \mathbf{I}_B)) + (1-a)\|\mathbf{I}_A - \mathbf{I}_B\|_1 \tag{3}$$

where $a \in [0, 1]$ is a weight coefficient. The projection in Eq. 2 refers to the projection of depth map $\mathbf{D}_t$ in $\mathbf{I}_{t'}$, $\langle\cdot\rangle$ represents a bilinear sampling operator, while $\mathbf{K}$ is the camera intrinsic parameters matrix. Additionally, $\mathbf{I}_t$ is the target image pose, $\mathbf{I}_{t'}$ the source image pose, and $\mathbf{T}_{t'\to t}$ the relative pose of $\mathbf{I}_{t'}$ with respect to $\mathbf{I}_t$. This loss metric allows regions that are occluded in one frame but visible in an adjacent one to be brought up, hence the minimum operation. This *per-pixel* photometric reprojection loss is computed over all source images, meaning a frame window.

This pivotal loss term is typically combined with various smoothness terms, in order to train a DNN for self-supervised MDE. The most common architectural choice is a Convolutional Neural Network (CNN) arranged according to an Encoder-Decoder approach [12].

B. Image-to-Image Translation

Generative Adversarial Networks (GANs) are generative models that learn a mapping $G : \mathbf{z} \mapsto \mathbf{Y}$ from a random noise vector $\mathbf{z} \in \mathbb{R}^n$ to output image $\mathbf{Y} \in \mathbb{R}^{k\times l}$ or a tensor $\mathbf{Y} \in \mathbb{R}^{k\times l\times m}$. In contrast, conditional GANs learn a mapping $G : \{\mathbf{X}, \mathbf{z}\} \mapsto \mathbf{Y}$ from observed input image $\mathbf{X} \in \mathbb{R}^{p\times r}$ and random noise vector $\mathbf{z}$, to $\mathbf{Y}$.

The Generator G is trained to produce outputs that cannot be distinguished from "real" images by an adversarially trained Discriminator D, which gradually learns to discern the synthetically generated images from real ones. The objective of a conditional GAN can be expressed as:

$$L_{cGAN}(G, D) = \mathbb{E}_{\mathbf{X},\mathbf{Y}}[logD(X, Y)] + + a\mathbb{E}_{X,z}[log(1 - D(\mathbf{X}, G(\mathbf{X}, \mathbf{z})))],$$

where G tries to minimize this objective against an adversary D that tries to maximize it:

$$G^* = arg\min_G \max_D L_{cGAN}(G, D). \tag{4}$$

In the unconditional variant, where the Discriminator does not observe $\mathbf{X}$, it holds that:

$$L_{cGAN}(G, D) = \mathbb{E}_Y[logD(\mathbf{Y})] + + \mathbb{E}_{X,z}[log(1 - D(G(\mathbf{X}, \mathbf{z})))].$$

It is best practice to augment the GAN objective with a more traditional loss, such as L_1 or L_2 norm. Although the Discriminator's job remains unchanged, the Generator is additionally constrained to stay near the corresponding ground-truth output as follows:

$$L(G) = E_{X,Y,z}[\|\mathbf{Y} - G(\mathbf{X}, \mathbf{z})\|]. \tag{5}$$

The overall training objective is:

$$G^* = arg\min_G \max_D L_{cGAN}(G, D) + \lambda L(G). \tag{6}$$

C. Proposed DSD Method

Assume a pretrained MDE DNN. Let $\mathbf{X}_0$ be an input image for which the MDE must estimate a final depth map $\mathbf{X}_N$. Since the MDE is a DNN, several intermediate representations will also be constructed while mapping $\mathbf{X}$ to $\mathbf{X}_N$ during inference. Thus, overall, a sequence of tensors $\mathbf{X}_i, 0 \le i \le N$ is defined for each input image $\mathbf{X}_0$, where N is the number of consecutive layer blocks in the MDE neural architecture: $f_{MDE}(\mathbf{X}_0) = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N\}$.

The first step of the proposed method is to train a I2I model, under the conditional Generative Adversarial Network (GAN) paradigm [15], so that the Generator learns to map tensors $\mathbf{X}_i$, for a specific $i, 0 \le i < N$, to corresponding dense targets $\mathbf{X}_N$ (predicted final depth maps). The training dataset is derived by feeding images to the pretrained f_{MDE} model. Thus, the I2I learns to mimic the predicted depth map by producing an estimate $f_{I2I}(\mathbf{X}_i) = \tilde{\mathbf{X}}_N$.

During inference, the pretrained MDE is deployed in-the-field along with the trained I2I. When a test-stage image/data point comes along, the two models output the predicted depth map $\mathbf{X}_N$ and the estimated depth map $\tilde{\mathbf{X}}_N$, respectively. Obviously, $\tilde{\mathbf{X}}_N$ is an estimation of $\mathbf{X}_N$. Moreover, with the sole exception of the scenario where $i = 0$, an intermediate-layer image representation formed during the inference stage of the MDE is exploited as the input to the I2I model.

Given the two depth maps, the next step is to compute the per-pixel absolute difference between them:

$$\mathbf{Y} = |\mathbf{X}_N - \tilde{\mathbf{X}}_N| = |f_{MDE}(\mathbf{X}_0) - f_{I2I}(\mathbf{X}_i)|. \tag{7}$$

Subsequently, the scalar absolute differences contained in matrix $\mathbf{Y}$ are combined via an aggregation strategy (e.g., averaging), in order to derive a single scalar *informal uncertainty estimate* for the entire image. Finally, by simply thresholding this estimate, the image can be categorized as ID or OOD. This pipeline is graphically depicted in Fig. 1.

The MDE has been pretrained (under a supervised or self-supervised setting) on the dataset $\mathcal{D}_{in}$, while the I2I has been optimized to mimic its predictions (under an adversarial setting). Hence, during inference, $\tilde{\mathbf{X}}_N \approx \mathbf{X}_N$ in $\mathcal{D}_{in}$. However,

this approximate equality does not hold on images from a different, OOD dataset $\mathcal{D}_{out}$. This is because it is highly unlikely that the two DNNs, which are architecturally different and have been trained under different learning settings, will respond similarly to test images they are both unfamiliar with. Thus, the informal uncertainty estimate is expected to be significantly larger for OOD images.

The inference flowchart of the proposed method is depicted in Algorithm 1:

Algorithm 1 Distribution Shift Detection

$\mathbf{X}_0 \leftarrow RGB_{image}$
$[\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_N] \leftarrow f_{MDE}(\mathbf{X}_0)$
$\tilde{\mathbf{X}}_N \leftarrow f_{I2I}(\mathbf{X}_i)$
$\mathbf{U} = abs(\mathbf{X}_N - \tilde{\mathbf{X}}_N)$
for $u \in \mathbf{U}$: **do**
  if $u < t$ **then**
    $u \leftarrow 0$ ▷ Marked as OOD
  else
    $u \leftarrow 1$ ▷ Marked as ID
  end if
end for
return $\mathbf{U}$

IV. Experimental Evaluation

This Section discusses implementation details of the proposed method, as well as the followed evaluation process.

A. Datasets

Three datasets were employed for training and evaluation, with the prospect that one will serve as the ID dataset (D_{in}), and the other two as the OOD datasets (D_{out}). This is an evaluation protocol similar to the ones typically utilized for assessing OODD methods [32] [31] [17]. In the MDE scenario, where there is only domain shift and no label-space shift, such a protocol implies a very large shift in image appearance between the training and the test dataset; this is the case where DSD is most critical (since MDE is most likely to fail).

The datasets employed were KITTI RAW [9], Cityscapes [7] and DDAD [12]; all of them collected by driving a car. The first two have been derived using stereo rigs on the front side of the vehicle, while the latter one has only one on the driver's side (corresponding to the "left" camera of the former two). The DDAD configuration has more cameras for 360° information, but this paper only utilized videos from the front-facing one. All three datasets were collected using RGB cameras, with KITTI RAW and DDAD leveraging LiDAR sensors to generate ground-truth depth maps. In Cityscapes, depth maps were generated by stereoscopic 3D analysis.

B. Training

The selected MDE neural architecture was Monodepth2 [11], also used in [29]. This DNN is comprised of a ResNet-18 encoder analyzing the input RGB image and a decoder that outputs the final depth map prediction. The feature tensors $\mathbf{X}_i, 1 \leq i \leq K$, as defined in Section III, refer to the outputs of the i-th layer of the encoder. $\mathbf{X}_0$ refers to the raw input RGB image.

The popular Pix2Pix C-GAN [15] was selected as the I2I Translator. Training on Monodepth2 is done using the KITTI RAW dataset for 20 epochs using a batch size of 12 and a learning rate of 10^{-4}, in a self-supervised monocular scenario. Since Monodepth2 produces sharp, high quality depth maps in this training setting [11], we chose to use a feature vector derived from its intermediate layers as training input for our novel method.

The resulting depth maps, as well as the original RGB images just before entering Monodepth2 are scaled down to 640×192 spatial resolution.

For each training image, the I2I was trained with the Monodepth2-predicted depth map $\mathbf{X}_N$ as target and a tensor $\mathbf{X}_i$, $0 \leq i \leq K$ as input, for a specific i that was empirically chosen. Since $K = 4$, there are 5 possible types of inputs and 5 corresponding trained I2I instances (for $i \in \mathbb{N}, i \in [0, \ldots, 4]$). In all cases, training proceeded for 200 epochs, with 100 of them having a stable learning rate of $2 \cdot 10^{-4}$ and the rest decaying it towards zero.

C. Metrics

To evaluate the proposed method and compare it with MDE uncertainty estimation methods in DSD, standard performance metrics were used [6]. Having an informal uncertainty index, which can also be viewed as a confidence by taking its complement ($confidence = 1 - uncertainty$), the following common metrics can be utilized:

1) False Positive Rate (**FPR**) at 95% True Positive Rate (lower is better).
2) The Detection Error (**DERR**), which illustrates the minimum probability to misdetect an OOD sample for all possible thresholds (lower is better).
3) The Area Under the Receiver Operating Characteristic (**AUROC**) curve, which represents the chance for a positive sample to be given a higher detection score than a negative one (higher is better).
4) The Area Under the Precision Recall (AUPR) curve, which measures the precision and recall curves against each other. So for TP being True Positive, FP False Positive, and FN False Negative:

$$precision = \frac{TP}{TP + FP} \quad \& \quad recall = \frac{TP}{TP + FN}$$

The AUPR metric is used in two variants, one where the ID samples are considered positive (**AUIN**), and another with the OOD samples considered so (**AUOUT**) (higher is better).

For each test image, the uncertainty map $\mathbf{Y}$ was aggregated into a single scalar informal uncertainty index. Empirical investigation revealed simple averaging to be the best aggregation strategy amongst the ones that were tried.

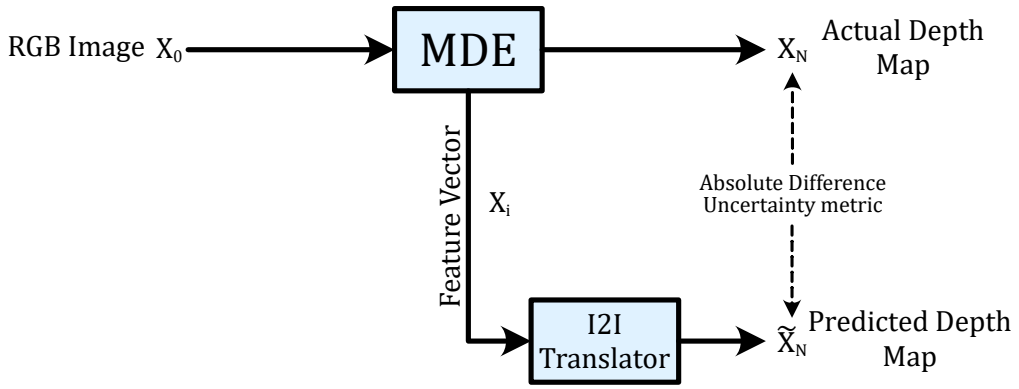

Fig. 1. The proposed DSD pipeline when analyzing a test-stage image. The aggregated absolute difference between the MDE-predicted depth map and the I2I-derived one can be used as an informal uncertainty index. Its thresholding provides an effective OOD indicator.

TABLE S1
DSD PERFORMANCE RESULTS FOR DIFFERENT i VALUES. D_{in}: KITTI RAW, D_{out}: DDAD, CITYSCAPES.

D_{in}: KITTI RAW - D_{out}: DDAD					
	FPR↓	DTERR↓	AUROC↑	AUIN↑	AUOUT↑
$\mathbf{X}_0$	84,07	28,34	78,03	80,48	71,77
$\mathbf{X}_1$	95,27	27,76	75,73	80,19	65,93
$\mathbf{X}_2$	<u>52,8</u>	<u>17,93</u>	<u>88,63</u>	<u>90,19</u>	<u>83,72</u>
$\mathbf{X}_3$	**36,01**	**13,34**	**93,90**	**94,65**	**92,13**
$\mathbf{X}_4$	74,32	19,94	85,8	88,27	79,86

D_{in}: KITTI RAW - D_{out}: CITYSCAPES					
	FPR↓	DTERR↓	AUROC↑	AUIN↑	AUOUT↑
$\mathbf{X}_0$	57,82	23,46	84,94	85,82	84,24
$\mathbf{X}_1$	63,56	22,09	85,57	87,66	83,38
$\mathbf{X}_2$	**46,48**	**16,21**	**90,56**	**91,5**	**88,55**
$\mathbf{X}_3$	<u>46,34</u>	<u>18,29</u>	<u>89,84</u>	<u>90,21</u>	<u>88,77</u>
$\mathbf{X}_4$	75,75	27,19	78,94	79,29	76,58

TABLE S2
DSD PERFORMANCE COMPARISONS.

D_{in}: KITTI RAW - D_{out}: DDAD					
	FPR↓	DTERR↓	AUROC↑	AUIN↑	AUOUT↑
[29]	82,93	34,86	69,62	68,43	68,24
[35]	63,6	21,9	84,37	84,95	81.24
[32]	77,63	29,41	76,87	75,47	73,12
[33]	61,49	19,8	87,76	88,90	84,42
Ours	**36,01**	**13,34**	**93,90**	**94,65**	**92,13**

D_{in}: KITTI RAW - D_{out}: CITYSCAPES					
	FPR↓	DTERR↓	AUROC↑	AUIN↑	AUOUT↑
[29]	83,64	34,65	69,75	69,57	67,42
[35]	98,00	43,00	51,40	48,90	51,16
[32]	97,41	45,62	53,13	52,29	51,30
[33]	73,88	26,97	78,61	75,00	76,77
Ours	**46,34**	**18,29**	**89,84**	**90,21**	**88,77**

D. Results

DSD performance was independently measured for different values of i and for both possible dataset combinations. The results are shown in Table S1. Best results in each metric are in **bold**; second best are <u>underlined</u>.

By examining the evaluation results, it seems that the empirically optimal value of i across dataset combinations is $i = 3$. This implies using the intermediate representation produced by the third layer of the MDE's encoder, i.e., $\mathbf{X}_3$, as I2I input. A different choice of $i, 0 \leq i \leq K$ may sporadically lead to slightly higher performance in some metrics, but $i = 3$ gives consistently good results.

Table S2 compares our method for $i = 3$ against competing methods, properly adapted to our evaluation setup. The proposed algorithm surpasses the competition on both datasets, by a substantial margin across all metrics, with the KITTI dataset as D_{in}. The FPR at 95% TPR shows the biggest advantage margin: a considerable and consistent lead of approximately 25% to 26% from the best competing method when we use either DDAD or Cityscapes as the D_{out}.

V. CONCLUSIONS

This paper proposed a novel method for Distribution Shift Detection (DSD) in Deep Neural Networks (DNNs) for Monocular Depth Estimation (MDE). This is an important task in safety-critical applications such as autonomous driving. The method operates by pretraining an Image-to-Image Translator (I2I) to mimic the depth maps generated by the MDE model and eventually, discriminate between OOD and ID inputs during deployment, by exploiting the difference between the two depth map predictions for a test-stage input image, as an informal scalar uncertainty index. Such an approach relies on the insight that the outputs of the MDE and the I2I are less correlated for OOD images, due to their different neural architectures and learning paradigms. Experimental evaluation on autonomous driving-compatible datasets showcased that, by exploiting the feature extraction capabilities of a high-accuracy MDE model and using an intermediate representation produced by its encoder as input, considerably higher DSD performance is obtained, compared to training an independent uncertainty estimator. Thus, our method is shown to be simultaneously simpler and more effective than state-of-the-art

competing algorithms.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951911 (AI4Media). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

REFERENCES

[1] A. Alcántara, J. Capitán, A. Torres-González, R. Cunha, and A. Ollero, *Autonomous execution of cinematographic shots with multiple drones*, IEEE Access **8** (2020), 201300–201316.

[2] L. Andraghetti, P. Myriokefalitakis, P. L. Dovesi, B. Luque, M. Poggi, A. Pieropan, and S. Mattoccia, *Enhancing self-supervised monocular depth estimation with traditional visual odometry*, Proceedings of the IEEE International Conference on 3D Vision (3DV), 2019.

[3] P. Bevandić, I. Krešo, M. Oršić, and S. Šegvić, *Discriminative Out-Of-Distribution Detection for semantic segmentation*, arXiv preprint arXiv:1808.07703 (2018).

[4] R. Chan, M. Rottmann, and H. Gottschalk, *Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[5] V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection: A survey*, ACM Computing Surveys **41** (2009), no. 3.

[6] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, *Robust out-of-distribution detection for neural networks*, arXiv preprint arXiv:2003.09711 (2021).

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, *The Cityscapes dataset for semantic urban scene understanding*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[8] L. Cultrera, L. Seidenari, F. Becattini, P. Pala, and A. Del Bimbo, *Explaining autonomous driving by learning end-to-end visual attention*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, *Vision meets robotics: The KITTI dataset*, International Journal of Robotics Research (IJRR) (2013).

[10] C. Godard, O. Mac Aodha, and G. J. Brostow, *Unsupervised monocular depth estimation with left-right consistency*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[11] C. Godard, O. Mac Aodha, M. Firman, and G. J Brostow, *Digging into self-supervised monocular depth estimation*, Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR), 2019.

[12] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, *3D packing for self-supervised monocular depth estimation*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[13] F. Hell, G. Hinz, F. Liu, S. Goyal, K Pei, T. Lytvynenko, A. Knoll, and C. Yiqiang, *Monitoring perception reliability in autonomous driving: Distributional shift detection for estimating the impact of input data on prediction accuracy*, Proceedings of the Computer Science in Cars Symposium, 2021.

[14] D. Hendrycks and K. Gimpel, *A baseline for detecting misclassified and out-of-distribution examples in neural networks*, Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[16] E. Kakaletsis, C. Symeonidis, M. Tzelepi, I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, *Computer vision for autonomous UAV flight safety: An overview and a vision-based safe landing pipeline example*, ACM Computing Surveys (CSUR) **54** (2021), no. 9, 1–37.

[17] Shiyu Liang, Yixuan Li, and R. Srikant, *Enhancing the reliability of out-of-distribution image detection in neural networks*, International conference on learning representations, 2018.

[18] C. Liu, J. Gu, K. Kim, S. G. Narasimhan, and J. Kautz, *Neural RGB-D sensing: Depth and uncertainty from a video camera*, Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr), 2019.

[19] I. Mademlis, A. Torres-González, J. Capitán, M. Montagnuolo, A. Messina, F. Negro, C. Le Barz, T. Gonçalves, R. Cunha, B. Guerreiro, et al., *A multiple-UAV architecture for autonomous media production*, Multimedia Tools and Applications (2022), 1–30.

[20] M. Markou and S. Singh, *Novelty detection: a review*, Signal Processing **83** (2003), no. 12, 2481–2497.

[21] A. Nguyen, J. Yosinski, and J. Clune, *Deep neural networks are easily fooled: High confidence predictions for unrecognizable images*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[22] P. Oberdiek, M. Rottmann, and G. A. Fink, *Detection and retrieval of out-of-distribution objects in semantic segmentation*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020.

[23] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, *Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift*, Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2019.

[24] S. Papadopoulos, I. Mademlis, and I. Pitas, *Neural vision-based semantic 3D world modeling*, Proceedings of the ieee/cvf winter conference on applications of computer vision (wacvw), 2021.

[25] ———, *Semantic image segmentation guided by scene geometry*, Proceedings of the ieee international conference on autonomous systems (icas), 2021.

[26] C. Papaioannidis, I. Mademlis, and I. Pitas, *Autonomous UAV safety by visual human crowd detection using multi-task deep neural networks*, Proceedings of the ieee international conference on robotics and automation (icra), 2021.

[27] ———, *Fast CNN-based single-person 2D human pose estimation for autonomous systems*, IEEE Transactions on Circuits and Systems for Video Technology (2022).

[28] N. Passalis and A. Tefas, *Continuous drone control using deep reinforcement learning for frontal view person shooting*, Neural Computing and Applications **32** (2020), no. 9, 4227–4238.

[29] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, *On the uncertainty of self-supervised monocular depth estimation*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[30] S. Rabanser, S. Günnemann, and Z. C. Lipton, *Failing loudly: An empirical study of methods for detecting dataset shift* (2018).

[31] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan, *Likelihood ratios for out-of-distribution detection*, Advances in neural information processing systems, 2019.

[32] V. Sehwag, M. Chiang, and P. Mittal, *SSD: A unified framework for self-supervised outlier detection*, arXiv preprint arXiv:2103.12051 (2021).

[33] A. Stocco, M. Weiss, M. Calzana, and P. Tonella, *Misbehaviour prediction for autonomous driving systems*, Proceedings of the ACM/IEEE International Conference on Software Engineering, 2020.

[34] Y. Weng, J. Pajarinen, R. Akrour, T. Matsuda, J. Peters, and T. Maki, *Reinforcement learning based underwater wireless optical communication alignment for autonomous underwater vehicles*, IEEE Journal of Oceanic Engineering **47** (2022), no. 4, 1231–1245.

[35] Z. Xiao, Q. Yan, and Y. Amit, *Likelihood regret: An Out-Of-Distribution Detection score for variational auto-encoder*, Proceedings of Advances in Neural Information Processing Systems (NIPS) (2020).

[36] G. Yang, P. Hu, and D. Ramanan, *Inferring distributions over depth from a single image*, Proceedings of the ieee/rsj international conference on intelligent robots and systems (iros), 2019.