Towards Domain Independence for Learning-based Monocular Depth Estimation

Michele Mancini[1], Gabriele Costante[1], Paolo Valigi[1] and Thomas A. Ciarfuglia[1]
Jeffrey Delmerico[2] and Davide Scaramuzza[2]

***Abstract*— Modern autonomous mobile robots require a strong understanding of their surroundings in order to safely operate in cluttered and dynamic environments. Monocular depth estimation offers a geometry-independent paradigm to detect free, navigable space with minimum space and power consumption. These represent highly desirable features, especially for micro aerial vehicles. In order to guarantee robust operation in real world scenarios, the estimator is required to generalize well in diverse environments. Most of the existent depth estimators do not consider generalization, and only benchmark their performance on publicly available datasets after specific fine-tuning. Generalization can be achieved by training on several heterogeneous datasets, but their collection and labeling is costly. In this work, we propose a Deep Neural Network for scene depth estimation that is trained on synthetic datasets, which allow inexpensive generation of ground truth data. We show how this approach is able to generalize well across different scenarios. In addition, we show how the addition of Long Short Term Memory (LSTM) layers in the network helps to alleviate, in sequential image streams, some of the intrinsic limitations of monocular vision, such as global scale estimation, with low computational overhead. We demonstrate that the network is able to generalize well with respect to different real world environments without any fine-tuning, achieving comparable performance to state-of-the-art methods on the KITTI dataset.**

Supplementary Material

A video showing the results of our monocular depth estimation approach is available at `https://youtu.be/UfoAkYLb-5I`.

The datasets we collected and the trained models to reproduce our results are available at `http://www.sira.diei.unipg.it/supplementary/ral2016/extra.html`.

I. Introduction

As autonomous vehicles become more common in many applications outside the research laboratory, the requirements for safe and optimal operation of such systems become more challenging. In particular, the ability to detect and avoid still or mobile obstacles is crucial for field operations of the vast majority of ground and low altitude flight vehicles. Depth information can be used to estimate proximity to obstacles and, in general, to obtain an understanding of the surrounding 3D space. This perception of the 3D environment can then be used in reactive [1] or planned [2] control strategies to navigate safely. LIDAR and sonar sensors can provide sparse 3D information, but their installation may be costly, in terms of weight, space and power, all of which are constraints for mobile robots, and especially Micro Aerial Vehicles (MAVs). Vision-based systems, both mono and stereo, can provide dense depth maps and are more suitable for deploying on small vehicles. A primary shortcoming, though, is that the detection range and accuracy of stereo cameras are limited by the camera set-up and baseline [3], [4]. Exploiting geometric constraints on camera motion and planarity, obstacle detection and navigable ground space estimation can be extended far beyond the normal range ([5], [6]). However, these constraints hold mostly for ground, automotive applications, but do not generalize to MAVs.

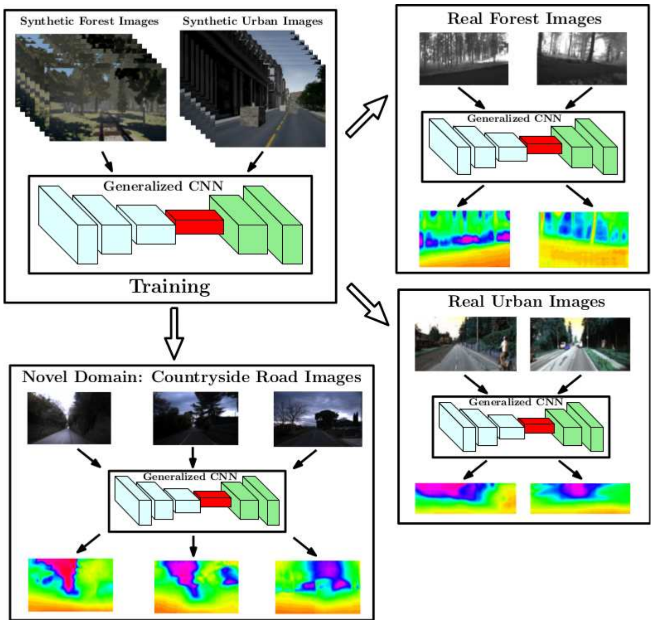

Fig. 1: Overview of the proposed domain independent approach for monocular depth estimation based on CNN. We first train our model on labeled synthetic data. We then deploy it for evaluation on real world scenarios. Our experiments show how the model is able to generalize well across different scenarios without requiring any domain specific fine-tuning procedures.

*We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

*This work was supported in part by the DARPA FLA Program and by the M.I.U.R. (Minstero dell'Istruzione dell'Università e della Ricerca) under Grant SCN_398/SEAL (Program Smart Cities).

[1] First Author, Second Author, Third Author and Fourth Author are with the Department of Engineering, University of Perugia, Italy {michele.mancini,gabriele.costante, thomas.ciarfuglia, paolo.valigi}@unipg.it

[2] Fifth Author and Sixth Author are with the Robotics and Perception Group, University of Zurich, Switzerland {jeffdelmerico,sdavide}@ifi.uzh.ch

Differently from stereo systems, monocular systems do not make specific assumptions on camera motion or setup. Several monocular depth estimation methods have been proposed in recent years, mostly exploiting machine learning paradigms ([7], [8], [9], [10], [11]). The advantages of such systems are that they are able to learn scale without the use of external metric information, such as Inertial Measurement Unit (IMU) measurements, and are not subject to any geometrical constraint. On the downside, these systems rely on the quality and variety of the training set and ground truth provided, and often are not able to adapt to unseen environments.

The challenge of domain independence is one of the main obstacles to extensive use of learned monocular systems in place of stereo geometrical ones. The question of how does these systems perform in uncontrolled, previously unseen scenarios can be addressed by learning features that are more invariant to environment changes and also by using different network architectures that are able to learn more general models from the training samples they have. Unfortunately, there are only a few labeled depth datasets with the required ground truth density, and the cost and time required to create new ones is high.

In our previous work [12] we showed that training a Convolutional Neural Network (CNN) with a inexpensive generated, densely-labeled, synthetic urban dataset, achieved promising results on the KITTI dataset benchmark using RGB and optical flow inputs.

In this work, by using a deeper architecture and an extended synthetic dataset able to generalize from synthetic data to real unseen sequences, we take an important step towards domain independence for vision-based depth estimation applications. With robotic-based operations in mind, we reduce the computational complexity of the network by removing the network dependence on optical flow, even if it often acts as a environment-invariant feature. To balance this loss of information, we exploit the input stream sequentiality by using Long Short Term Memory (LSTM) layers, a specific form of Recurrent Neural Networks (RNNs).

Our experiments show that this solution significantly outperforms previous results. We validate our model on the KITTI dataset, where we obtain comparable performance to state-of-the-art, specially tuned methods.

We also perform validation on two challenging and different new datasets consisting of sequences captured in a dense forest and in a country road, in order to evaluate possible MAV operation environments. We show how the model is capable of reliable estimation even on video streams with vibration and motion blur, making our model suitable for tasks as obstacle avoidance and motion planning for mobile robots.

II. Related Work

Traditional vision-based depth estimation is based on stereo vision [13]. Its main limitations lie on the lack of robustness on long range measurements and pixel matching errors. This aspect is even more critical in MAV applications where maneuvers are on 6DOF and the lack of platform space makes it difficult to mount a stereo rig with a proper baseline. Finally, weight and power consumption minimization is highly desirable. For these reasons, monocular vision is becoming more and more important when it comes to MAV applications.

Monocular depth estimation based on geometric methods is grounded on the triangulation of consecutive frames. Despite the impressive results achieved by state-of-the-art approaches [14], [15], [16], the performance of their reconstruction routines drops during high-speed motion, as dense alignment becomes extremely challenging. In addition, it is not possible to recover the absolute scale of the object distances. Driven by the previous considerations, in this work, we address both the aforementioned aspects by exploiting the learning paradigm to learn models that compute the scene depth and the associated absolute scale from a single image (*i.e.*, without processing multiple frames).

Learning-based methods for depth estimation have demonstrated good performance in specific scenarios, but these results are limited to these environments, and have not been shown to generalize well. Saxena et al. [17] first proposed a Markov Random Field to predict depth from a monocular, horizontally-aligned image, which then later evolved into the Make3D project [10]. This method tends to suffer in uncontrolled settings, especially when the horizontal alignment condition does not hold. Eigen et al. ([7], [18], exploit for the first time in their work the emergence of Deep Learning solutions for this kind of problems, training a multi scale convolutional neural network (CNN) to estimate depth. Following this, several other CNN-based approaches have been proposed. Liu et al. [8] combine a CNN with a Conditional Random Field to improve smoothness. Roy et al. [9] recently proposed a novel depth estimation method based on Neural Regression Forest. However, the aforementioned methods [17], [7], [18], [8], [9] are specific for the scenario where they have been trained and, thus, they are not domain independent.

For our intended embedded application, computational efficiency is very important, and, in this respect, most of the existing methods for monocular depth estimation are not appropriate. In [8] and [9], although they reported slightly improved performances on several benchmarks with respect to Eigen et al.'s work, they cannot guarantee real-time performance on embedded hardware. They report a single image inference time of $\sim$ 1s both on a GTX780 and a Tesla k80, far more powerful hardware than the ones generally embedded on MAVs. Conversely, Eigen et al. method is able to estimate a coarser resolution ($1/4$ of the input image) of the scene depth map with a inference time of about 10ms. Our system's inference time is less than 30ms on a comparable hardware (Tesla k40) and less than 0.4s on an embedded hardware (Jetson TK1), making real-time application feasible. Based on these various factors, we chose the Eigen et al. [7] method to serve as a reference to the state of the art during our experiments.

Although we are interested in performing well against the state of the art in accuracy, our primary goal is to develop a robust estimator that is capable of generalizing well to previously unseen environments, in order to be useful in robotic applications. For this reason, we did not perform any finetuning on evaluation benchmarks, focusing on how architectural choices and synthetic datasets generation influence generalization. Our previous work propose a baseline solution to the problem, suggesting a Fully Convolutional Network (FCN) fed with both the current frame and the optical flow between current and previous frame [12]. Despite optical flow acts as a good environment-invariant feature, it is not sufficient to achieve generalization across different scenarios. Furthermore, the computation of the optical flow considerably increase the overall inference time. In this work, only the current frame is fed into the network: by using a deeper architecture and the LSTM paradigm together with a wise mix of different synthetic datasets we report a significant performance gain in a simpler and more efficient fashion.

A relatively unexplored area of research is the training of networks given data scarcity. Recently, Garg et al. [11] proposed an unsupervised approach for monocular depth estimation with CNNs. In their work they propose a data augmentation technique to deal with the cost of acquiring real images with depth ground truth. However, the augmented dataset has to be generated from already acquired images, and thus this technique is unable to generate unseen environments. For this reason the authors train and test only on the KITTI dataset. Our work is similar to theirs in the aspect of finding ways to effectively augment training data, but is aimed to generalize performances across different environments. We achieve this exploiting synthetic data, for which exact labels are easily generated. Synthetic training sets are able to represent any kind of scenario, illumination conditions, motion trajectory and camera optics, without any limitation imposed by real world data collection equipments. This allows us to reach good performance on different domains, using different training and test images, and not requiring fine-tuning. However, at the time of the writing of this work, the authors of [11] did not yet make their trained model publicly available for an effective comparison.

III. Network Overview

A. Fully Convolutional Network

We propose as a baseline method a fully convolutional architecture, structured in a encoder-decoder fashion, as depicted in Figure 2. This a very popular architectural choice for several pixel-wise prediction tasks, as optical flow estimation [19] or semantic segmentation [20]. In our proposed network, the encoder section corresponds to the popular VGG network [21], pruned of its fully connected layers.

We initialize the encoder weights with the VGG pretrained model for image classification. Models trained on huge image classification datasets, as [22], proved to act as a great generic-purpose feature extractor [23]: low-level features are extracted by convolutional layers closer to the input layer of the net, while layers closer to the output of the net extract high-level, more task-dependent descriptors. During training, out of the 16 convolutional layers of the VGG net, the weights of the first 8 layers are kept fixed; remaining layers are fine-tuned. The decoder section of the network is composed by 2 deconvolutional layers and a final convolutional layer which outputs the predicted depth at original input resolution. These layers are trained from scratch, using random weight initialization.

B. Adding LSTM layers into the picture

Any monocular, single image depth estimation method suffers from the infeasibility of correctly estimating the global scale of the scene. Learning-based methods try to infer global scale from the learned proportions between depicted objects in the training dataset. This paradigm inevitably fails when previously unseen environments are evaluated or when the camera focal length is modified.

We can try to correct these failures by exploiting the sequential nature of the image stream captured by a vision module mounted on a deployed robot. Recurrent neural networks (RNN) are typically used in tasks where long-term temporal dependencies between inputs matter when it comes to performing estimation: text/speech analysis, action recognition in a video stream, person re-identification [24], [25], [26]. Their output is a function of both the current input fed into the network and the past output, so that memory is carried forward through time as the sequence progresses:

$$\mathbf{y}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{y}_{t-1}) \tag{1}$$

where W represents the weight matrix (as in common feedforward networks) and U is called *transition matrix*.

Long Short Term Memory networks (LSTM) are a special kind of recurrent neural network introduced by Hochreiter & Schmidhuber in 1997 to overcome some of the RNN main issues, as vanishing gradients during training, which made them very challenging to use in practical applications [27]. Memory in LSTMs is maintained as a gated cell where information can be read, written or deleted. During training, the cell learns autonomously how to treat incoming and stored information. We insert two LSTM layers between the encoder and decoder section of the previously introduced FCN network, in a similar fashion of [24]. Our motivation is to refine features extracted by the encoder according to the information stored in the LSTM cells, so that the decoder section can return a more coherent depth estimation. The proposed LSTM network is depicted in Image 3. Dropout is applied before, after and in the middle of the two LSTM layers to improve regularization during training.

C. Training the networks

We developed two synthetic datasets for learning depth estimation: the *Urban Virtual Dataset (UVD)* and the *Forest Virtual Dataset (FVD)*, producing a total of more than 80k images (Figure 4). We create the scenarios with Unreal Engine, and extract noise-free ground truth depth maps using its tools. To reduce network's output space dimensionality

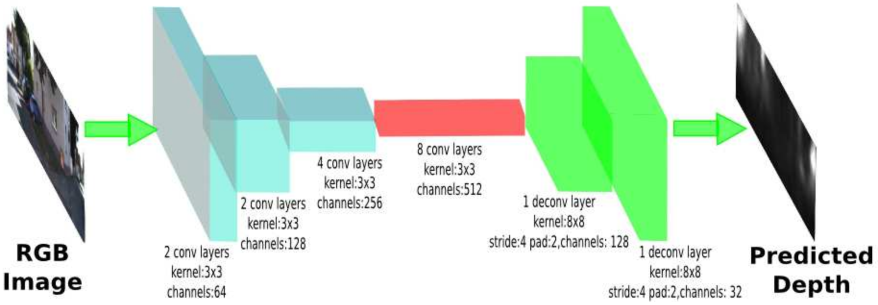

Fig. 2: FCN high-level architecture. Each block represent a set of layers with the depicted specifications. For the encoder section, pooling is applied between each block. Blue boxes: Unchanged VGG encoder layers. Red boxes: Finetuned VGG encoder layers. Green Boxes: Deconvolutional decoder layers.

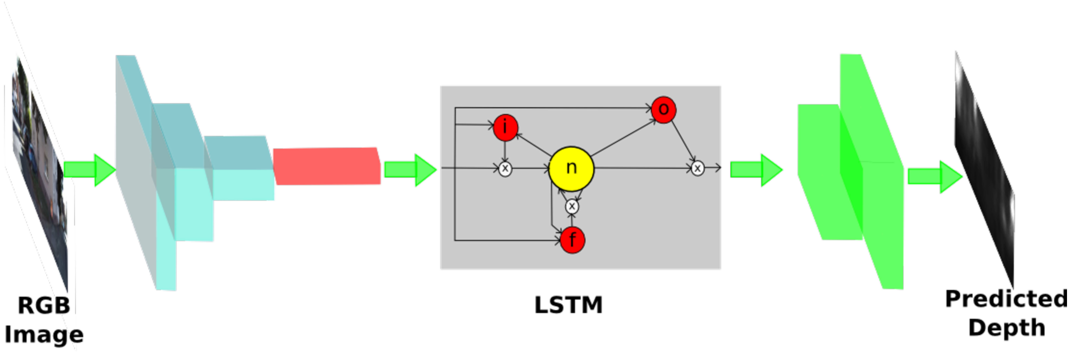

Fig. 3: In our LSTM network, we plug in two LSTM layers with 180 neurons between the encoder and the decoder section of the network.

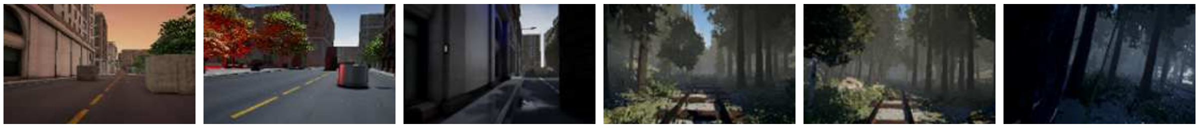

Fig. 4: Some images from UVD and FVD dataset used for training the models.

and ease training, we clip the depth maximum range to 40m, although it is theoretically possible to measure depth up to an unlimited range. Different illumination conditions, motion blur, fog, image noise and camera focal lengths can be easily simulated or modified, offering us a great sandbox to inexpensively generate highly informative datasets and high precision ground truths. The camera moves at speeds up to about 15m/s with six degrees of freedom inside the built scenarios, collecting frames and corresponding depth maps at a resolution of 256x160 pixels and a frame rate of 10Hz. Using these datasets, we trained the following networks:

- **UVD_FCN:** Fully convolutional network trained on the Urban Virtual Dataset.
- **FVD_FCN:** Fully convolutional network trained on the Forest Virtual Dataset.
- **FVD_LSTM:** LSTM network trained on the Forest Virtual Dataset.
- **MIX_FCN:** Fully convolutional network trained on both Urban and Forest Virtual Datasets.
- **MIX_LSTM:** LSTM network trained on both Urban and Forest Virtual Datasets.

Networks have been implemented using the Caffe framework and trained on Log RMSE (Eq. 2) using an Adam solver with a learning rate of $l = 10^{-4}$ until convergence. FCN networks required about 24 hrs for training, while LSTM networks took about 48 hrs on a Tesla K40 GPU.

$$\sqrt{\frac{1}{T}\sum_{Y\in T}||\log y_i - \log y_i^*||^2} \tag{2}$$

IV. Experiments

	UVD_FCN	FVD_FCN	MIX_FCN	MIX_LSTM
thr. $\delta < 1.25$	**0.705**	0.211	0.462	0.599
thr. $\delta < 1.25^2$	**0.899**	0.365	0.778	0.872
thr. $\delta < 1.25^3$	**0.968**	0.493	0.938	0.950
RMSE	**4.527**	15.697	6.581	5.966
Log RMSE	**0.264**	1.076	0.356	0.327
Scale Inv. MSE	**0.065**	0.907	0.072	0.087
Abs.Rel.Diff.	0.211	0.825	0.269	**0.188**

TABLE I: Results on UVD dataset. For threshold errors, higher values are better. For RMSE, Log RMSE, Scale Inv. MSE and Abs.Rel.Diff., lower values are better

We test generalization capability of our proposed networks on the KITTI dataset [28], and on two datasets we gathered in a dense forest in the surroundings of Zurich, Switzerland and in the countryside near Perugia, Italy, respectively.[1]

	UVD_FCN	FVD_FCN	MIX_FCN	MIX_LSTM
thr. $\delta < 1.25$	0.326	**0.574**	0.469	0.511
thr. $\delta < 1.25^2$	0.571	**0.853**	0.777	0.766
thr. $\delta < 1.25^3$	0.733	**0.939**	0.911	0.897
RMSE	8.802	**4.132**	5.134	5.460
Log RMSE	0.656	**0.340**	0.402	0.413
Scale Inv. MSE	0.357	**0.091**	0.106	0.132
Abs.Rel.Diff.	0.564	**0.248**	0.300	0.316

TABLE II: Results on FVD dataset.

We measure our performances with the following metrics:

- Threshold error: % of y_i s.t. $\max(\frac{y_i}{y_i^*} \frac{y_i^*}{y_i}) = \delta < thr$
- Absolute relative difference: $\frac{1}{T}\sum_{Y \in T} \frac{|y - y^*|}{y^*}$
- Log RMSE: $\sqrt{\frac{1}{T}\sum_{Y \in T} || \log y_i - \log y_i^* ||^2}$
- Linear RMSE: $\sqrt{\frac{1}{T}\sum_{Y \in T} ||y_i - y_i^*||^2}$
- Scale-invariant Log MSE (as introduced by [7]): $\frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}(\sum_i d_i)^2$, with $d_i = \log y_i - \log y_i^*$

We test on the same benchmark also our previous method proposed in [12], later referred as OPT_FLOW_FCN.

Furthermore, to properly compare our approach with respect to [7], we also implement their coarse+fine network following the details provided by the authors. We train it on both UVD and FVD datasets (*i.e.*, the same training set we use for our networks) with a Scale Inv. Log MSE loss. We first train their coarse model alone for 50 epochs, with a learning rate of 10^{-4}. Afterwards, we keep the weight of the coarse model fixed and train the fine network for about 40 epochs. Their method returns a 4× downsampled depth image, thus, during the evaluation, we upsample the obtained prediction with a nearest neighbor filter to match the original input resolution. In the following, we refer to this baseline as MIX_EIGEN.

Before discussing the results on the real datasets, we run a set of experiments to measure the performance loss when the test domain differs from the training one. In particular, in Table I, we compare the performance of the UVD models evaluated with respect to the urban domain (the same used for training) and the forest one. Similarly, in Table II, we show the results of the FVD networks. Clearly, performance drop when the network is tested on a domain different from the training one (see column 2 of Table I and column 1 of Table II). However, we can observe that extending the training set with images from multiple domains and with the LSTM structure helps the network to considerably increase the generalization capabilities of the CNNs and, as a consequence, the performance.

A. KITTI dataset

We evaluate our networks on a test set of 697 images used for evaluation in existing depth estimation methods [7] [8]. We do not perform any kind of fine-tuning or retraining on the target dataset. As reference, we compare with the method proposed by Eigen et. al [7]. The publicly available depth predictions they provide were specifically trained on the KITTI dataset, so comparison is not fully fair; our objective is to evaluate how close our performance can get relying solely on synthetic data.

We resize the input images from their original resolution of 1224x386 pixel to a resolution of 256x78 pixels for computational efficiency and feed them into our networks. From the provided sparse ground truth, captured by Velodyne lidar with a maximum range of about 80 meters, we generate a dense depth map utilizing the colorization routine proposed in [30]. As the lidar cannot provide depth information for the upper section of the image space, we perform evaluation only on the bottom section of the image space . We finally compute the performance metrics with respect of the windowed dense ground truth. We discard all the predictions whose corresponding ground truth measurement is beyond 40 meters, to be compliant with our network's maximum detection range.

As for Eigen's method, we compare both their publicly available depth predictions from their coarse+fine model trained on the KITTI dataset (referred as KITTI_EIGEN) and the MIX_EIGEN model we trained with respect to the synthetic images on the KITTI test set with the same dense ground truth we generated, employing the same benchmark used for our networks, to ensure evaluation fairness.

On Table III we report results for our FCN and LSTM networks, the baseline method [12] and Eigen et al.'s work.

The KITTI benchmark naturally favors networks trained on urban scenario datasets, as UVD_FCN. On the other hand, a forest scenario dataset as FVD does not suit well for this benchmark, as FVD_FCN performance clearly depicts. Anyway, mixing together FVD and UVD to form a heterogeneous training set allows MIX_FCN to improve significantly its prediction quality over UVD_FCN. With respect to KITTI_EIGEN, our best network obtains quite comparable performance on all metrics, recording slightly worse performance on threshold errors, Log RMSE and Scale Inv. MSE metrics but even some improvement on Linear RMSE and Absolute Relative Difference metrics. This is a very important result, especially considering how Eigen's work has been specifically trained on the target dataset. Heterogeneous synthetic training sets help the networks to learn a nicely generalizable model, without needing to resort on fine-tuning or collection of costly labeled real world datasets. Furthermore, our MIX_FCN network achieves better performance with respect to all the metrics than the MIX_EIGEN one. This suggests that our model has better generalization capabilities than the one presented in [7].

[1] Link to code, datasets and models: [29]

	OPTFLOW_FCN	UVD_FCN	FVD_FCN	MIX_FCN	MIX_LSTM	MIX_EIGEN	KITTI_EIGEN [7]	
thr. $\delta < 1.25$	0.421	0.414	0.160	**0.512**	0.338	0.183	0.498	Higher is better
thr. $\delta < 1.25^2$	0.679	0.695	0.351	0.786	0.644	0.456	**0.850**	
thr. $\delta < 1.25^3$	0.813	0.849	0.531	0.911	0.848	0.665	**0.957**	
RMSE	6.863	8.108	9.519	**5.654**	6.662	7.929	5.699	Lower is better
Log RMSE	0.504	0.470	0.877	0.366	0.472	0.589	**0.316**	
Scale Inv. MSE	0.205	0.181	0.315	0.107	0.185	0.131	**0.051**	
Abs.Rel.Diff.	–	0.393	0.494	**0.312**	0.430	0.390	0.322	

TABLE III: Results on KITTI dataset. In this benchmark, our best model (MIX_FCN) outperforms the Eigen's one when the latter is trained on our same synthetic dataset (MIX_EIGEN). Furthermore, it gets results close to the ones achieved with the model specifically trained on the KITTI dataset (KITTI_EIGEN)

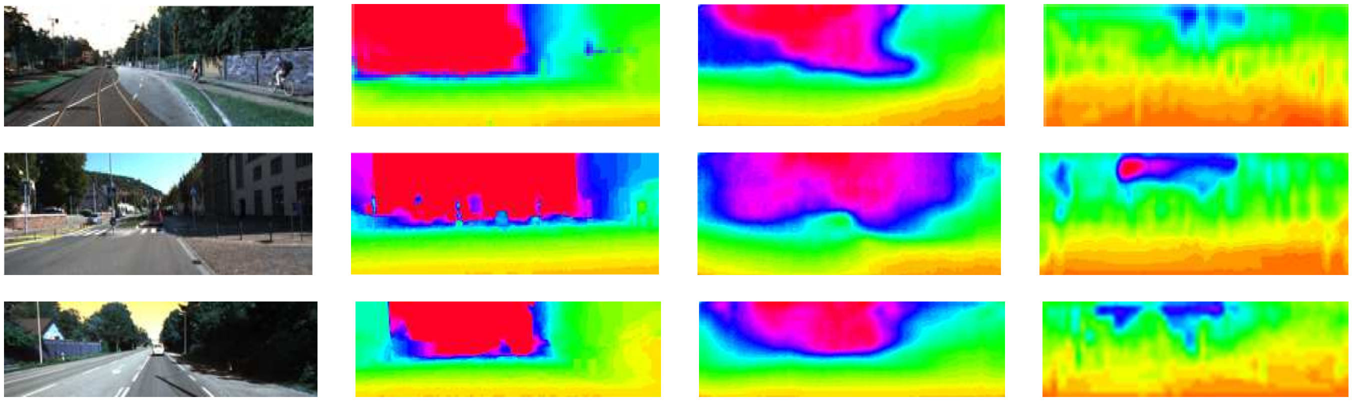

Fig. 5: Qualitative results on the KITTI dataset. On the first column RGB input images are depicted. The second and the third columns show the dense ground truths and MIX_FCN predictions, respectively. The fourth column shows MIX_EIGEN network prediction. Maximum depth range has been trimmed to 40 meters.

	OPTFLOW_FCN [12]	FVD_FCN	MIX_FCN	FVD_LSTM	MIX_LSTM	MIX_EIGEN	
thr. $\delta < 1.25$	0.096	0.106	0.149	0.126	**0.336**	0.111	Higher is better
thr. $\delta < 1.25^2$	0.202	0.231	0.316	0.269	**0.561**	0.246	
thr. $\delta < 1.25^3$	0.295	0.380	0.520	0.439	**0.707**	0.436	
RMSE	10.642	9.986	9.292	**9.126**	9.746	10.673	Lower is better
Log RMSE	1.133	1.007	0.856	0.908	**0.768**	0.960	
Scale Inv. MSE	0.646	0.527	0.402	0.523	0.439	**0.357**	
Abs.Rel.Diff.	2.127	1.797	1.378	1.427	**1.272**	1.777	

TABLE IV: Results on Zurich Forest dataset. Both MIX_FCN and MIX_LSTM outperform MIX_EIGEN in most of the metrics.

It is not surprising that the MIX_LSTM network does not achieve the best performance with respect to this dataset: the image frames of the test set are not always sequential and, thus, the LSTM model could not fully exploit its recurrent structure.

B. *Zurich Forest Dataset*

We gathered a new dataset in order to test the generalization of our networks on a real-world forest environment. The three sequences in the dataset consist of camera images captured while moving through a forested area at a walking pace of around 1 m/s. Each sequence lasted approximately 60 seconds and covered approximately 50 m of distance. These sequences include a variety of forest densities, tree sizes, and realistic lighting conditions. The original images in this dataset were captured with a pair of time-synchronized MatrixVision mvBlueFOX-MLC200w monochrome cameras with 752×480 resolution in stereo configuration with a baseline of 20 cm. Both cameras were recorded at 50 Hz, resulting in sequences with approximately 3000 stereo pairs each. Stereo matching was performed on these image pairs using OpenCV's Semi-global Block Matching algorithm to generate ground truth depth for validation of the monocular depth produced by our networks [31].

We tested our architectures on the three sequences, for a total of 9846 images. We resize the images on a resolution of 256x160 pixels before feeding them into our networks. We report results for our baseline method OPTFLOW_FCN and all the networks trained on FVD and MIX dataset. We report results on Table IV.

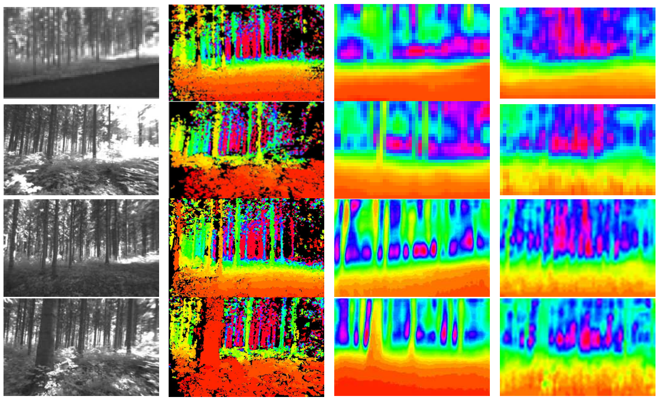

Fig. 6: Qualitative results on the Zurich Forest dataset. On the first column RGB input images are depicted. The second and the third columns show the dense ground truths and MIX_LSTM net predictions, respectively. The fourth column shows MIX_EIGEN network prediction. Maximum depth range has been trimmed to 40 meters. Black pixels in the ground truth represent missing depth measurements.

In this experiment, the LSTM architecture outperforms in almost all metrics the FCN architecture on both training datasets. In particular, we observe significant improvements on global scale-dependent metrics like threshold errors, LogRMSE and the Absolute Relative Difference. This confirms our intuition: LSTM layers helps to improve global scale estimation by using past information to refine current estimations. This comes at a very low computational additional cost, as depicted on Table V. As for the experiments on the KITTI dataset, both the FCN and LSTM architectures perform better than the MIX_EIGEN model.

	FPS (K40)	FPS (TK1)
FCN nets	58.8	2.7
LSTM nets	35.3	2.4

TABLE V: FPS (frame per second) for FCN and LSTM networks on 256x160 pixel inputs. Tested hardware: Tesla K40 and Jetson TK1 (for MAV onboard deploying)

C. Perugia Countryside Dataset

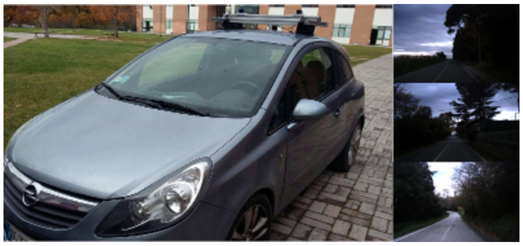

Fig. 7: Car setup used for collecting the Perugia Countryside Dataset. On the right, some sample images of the recorded sequences are shown.

To further evaluate the generalization capabilities of our approach, we collected a second dataset in the countryside area that surrounds the city of Perugia in Italy. Since the MIX_FCN and MIX_LSTM models are trained in forest and urban contexts, this new dataset has been specifically gathered to test whether our networks are able to generalize with respect to domains different from the training set ones or not. Images were collected using a stereo camera rig mounted on a car driven at around 14 m/s (see Figure 7). The sequences cover many kilometers of distance and contain different scenarios, elements (*e.g.*, small town buildings, sparse tree landscapes, moving cars and others) and light conditions.

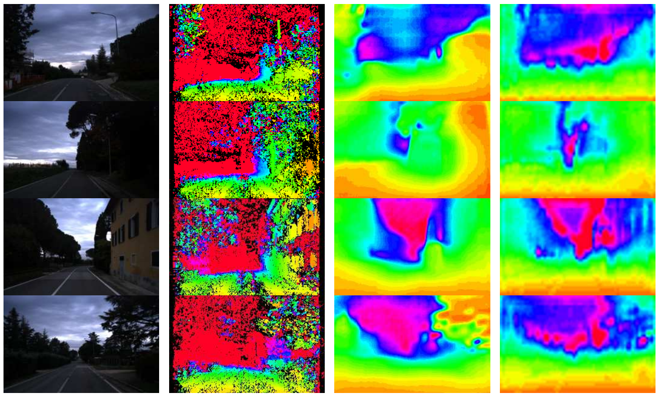

Fig. 8: Qualitative results on the Perugia Countryside dataset. On the first column RGB input images are depicted. The second and the third columns show the dense ground truths and the MIX_LSTM predictions, respectively. The fourth column shows MIX_EIGEN network prediction. Maximum depth range has been trimmed to 40 meters. Black pixels in the ground truth represent missing depth measurements.

The dataset was gathered with a pair of time-synchronized MatrixVision mvBlueFOX3 RGB cameras with 1280 × 960 resolution. In order to be able to compute the ground truth at higher ranges, we set up a stereo rig with a baseline of 60 cm. Both cameras recorded at 10 Hz, resulting in sequences with approximately 1600 stereo pairs each. Stereo matching was performed using the same strategy described in Section IV-B.

We compare our MIX_FCN and MIX_LSTM architectures (which showed good generalization capabilities in the previous experiments) and the baselines with respect to three sequences (5072 images). As the LSTM network and the Eigen's approach require input images with 256 × 160, we crop and resize them accordingly.

The results (see Table VI) confirm that the recurrent structure provides better performance with respect to both the standard FCN network and the Eigen's approach. Depth estimates (shown in Figure 8) are coherent with the actual scene depths. Thus, this suggests that our models (trained with images from different contexts, *e.g.*, dense forest and urban) are able to generalize with respect to different domains, considerably extending the applications contexts of depth estimation techniques.

We can also observe that the errors are higher with respect to the KITTI and Zurich forest dataset. However, this could be explained by the difference of camera intrinsics between the test and the train setup. Our networks are still able to provide reliable estimate when processing images with different focal lengths up to a scale factor. Despite the absolute metric errors are higher, the relative estimation are consistent (see Figure 8).

	MIX_FCN	MIX_LSTM	MIX_EIGEN
thr. $\delta < 1.25$	0.204	**0.209**	0.197
thr. $\delta < 1.25^2$	0.396	**0.405**	0.389
thr. $\delta < 1.25^3$	0.567	**0.576**	0.564
RMSE	13.003	**12.766**	12.925
Log RMSE	**0.802**	0.811	0.820
Scale Inv. MSE	0.583	**0.542**	0.640
Abs.Rel.Diff.	0.678	**0.631**	0.720

TABLE VI: Results on Perugia Countryside dataset.

V. Conclusion and Future Work

We propose a novel, Deep Learning based monocular depth estimation method, aimed at micro aerial vehicles tasks, such as autonomous obstacle avoidance and motion planning. We demonstrate how, using solely synthetic datasets, we can train a generalizable model that is capable of robust performance in real world scenarios. We obtained results that are comparable with the state of the art on the KITTI dataset without any fine-tuning. We also tested our algorithm in two other challenging scenario we gathered in a dense forest and a countryside, additionally showing how LSTM layers effectively help to improve estimation quality on typical MAV operating scenarios with a low added computational overhead. Future works will explore the possibility of integrating information coming from different sensors and/or modules (eg. IMU, semantic segmentation) to gain a better understanding of the surroundings and implement an effective reactive control for obstacle avoidance over it.

References

[1] H. Oleynikova, D. Honegger, and M. Pollefeys, “Reactive avoidance using embedded stereo vision for mav flight,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 50–56.

[2] C. Richter, A. Bry, and N. Roy, “Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments,” in *Robotics Research*. Springer, 2016, pp. 649–666.

[3] P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester, “Know your limits: Accuracy of long range stereoscopic object measurements in practice,” in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 96–111.

[4] E. R. Davies, *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.

[5] P. Pinggera, U. Franke, and R. Mester, “High-performance long range obstacle detection using stereo vision,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1308–1313.

[6] A. Harakeh, D. Asmar, and E. Shammas, “Ground segmentation and occupancy grid generation using probability fields,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 695–702.

[7] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[8] F. Liu, C. Shen, G. Lin, and I. Reid, “Learning depth from single monocular images using deep convolutional neural fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024–2039, Oct 2016.

[9] A. Roy and S. Todorovic, “Monocular depth estimation using neural regression forest,” 2016.

[10] A. Saxena, M. Sun, and A. Y. Ng, “Make3d: Learning 3d scene structure from a single still image,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 824–840, 2009.

[11] R. Garg and I. Reid, “Unsupervised cnn for single view depth estimation: Geometry to the rescue,” *arXiv preprint arXiv:1603.04992*, 2016.

[12] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, “Fast robust monocular depth estimation for obstacle detection with fully convolutional networks,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2016.

[13] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[14] M. Pizzoli, C. Forster, and D. Scaramuzza, “Remode: Probabilistic, monocular dense reconstruction in real time,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2609–2616.

[15] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[16] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “Dtam: Dense tracking and mapping in real-time,” in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[17] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in *Advances in Neural Information Processing Systems*, 2005, pp. 1161–1168.

[18] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[19] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” *arXiv preprint arXiv:1504.06852*, 2015.

[20] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” *arXiv preprint arXiv:1505.07293*, 2015.

[21] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014.

[24] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[25] S. Sharma, R. Kiros, and R. Salakhutdinov, “Action recognition using visual attention,” *arXiv preprint arXiv:1511.04119*, 2015.

[26] N. McLaughlin, J. Martinez del Rincon, and P. Miller, “Recurrent convolutional network for video-based person re-identification,” in *Computer Vision and Pattern Recognition, 2016. CVPR 2016. IEEE Conference on*, 2016.

[27] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, p. 0278364913491297, 2013.

[29] http://sira.diei.unipg.it/supplementary/ral2016/extra.html.

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 746–760.

[31] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.