# Exploiting AirSim as a Cross-dataset Benchmark for Safe UAV Landing and Monocular Depth Estimation Models

Jon Ander Iñiguez De Gordoa[a], Javier Barandiaran[b] and Marcos Nieto[c]
*Vicomtech Foundation, Basque Research and Technology Alliance (BRTA)*
*Mikeletegi 57, 20009 Donostia-San Sebastián, Spain*

Keywords: Monocular Depth Estimation, Safe Drone Landing, UAV, Synthetic Dataset, Simulation.

Abstract: As there is a lack of publicly available datasets with depth and surface normal information from a drone's view, in this paper, we introduce the synthetic and photorealistic AirSimNC dataset. This dataset is used as a benchmark to test the zero-shot cross-dataset performance of monocular depth and safe drone landing area estimation models. We analysed state-of-the-art Deep Learning networks and trained them on the SafeUAV dataset. While the depth models achieved very satisfactory results in the SafeUAV dataset, they showed a scaling error in the AirSimNC benchmark. We also compared the performance of networks trained on the KITTI and NYUv2 datasets, in order to test how training the networks on a bird's eye view affects in the performance on our benchmark. Regarding the safe landing estimation models, they surprisingly showed barely any zero-shot cross-dataset penalty when it comes to the precision of horizontal surfaces.

## 1 INTRODUCTION

Autonomous drones must retrieve information from their surroundings in order to ensure a safe flight and landing. On the one hand, embedded safety mechanisms based on ultrasound sensors are limited to short range applications. On the other hand, long distance safety mechanisms, such as LiDAR, are expensive and too heavy for small UAVs to carry them on board. Consequently, novel UAV models are trying to exploit Deep Learning and Computer Vision techniques as an alternative in order to get environment information.

While depth maps and surface normals can be estimated from RGB images using Deep Learning techniques, most depth and surface normal datasets that are publicly available have been directed towards autonomous driving or indoor applications. However, there is a lack of open-source and high resolution 3D datasets from bird's eye view. Generating a dataset with such characteristics is challenging, not only because of the technical difficulty of retrieving data from the viewpoint of a drone, but also due to the safety laws and airspace regulations in each country. Some approaches such as the SafeUAV dataset tried to generate a Google Earth-based semi-synthetic depth and landing area dataset (Marcu et al., 2019), but the reconstructed images are quite coarse and sometimes distorted. Therefore, the is no public benchmark on which UAV-based Deep Learning models can be properly compared.

The main contribution of this paper is the exploitation of the AirSim simulator (Shah et al., 2017) by generating the AirSimNC dataset, which is used as a benchmark in order to compare the zero-shot cross dataset performance of different monocular depth and safe landing area estimation models from a drone view. This dataset contains photorealistic images from bird's eye view at different heights, as well as the ground truth depth map and information about safe landing areas. The dataset is also diverse in effects such as motion blur or different weather conditions. Regarding monocular depth estimation, we analysed the state-of-the-art *AdaBins* (Bhat et al., 2020) and *DenseDepth* (Alhashim and Wonka, 2018) networks. We trained them on the SafeUAV dataset and tested them on the AirSimNC benchmark. We also tested the same networks trained on the bigger KITTI and NYUv2 datasets, in order to check whether if training the networks on a bird's eye view dataset such as SafeUAV (even if is not as photorealistic as KITTI or NYUv2) leads to better results in the AirSimNC benchmark. Regarding the

[a] https://orcid.org/0000-0002-9008-5620
[b] https://orcid.org/0000-0002-8135-0410
[c] https://orcid.org/0000-0001-9879-0992



Iñiguez De Gordoa, J., Barandiaran, J. and Nieto, M.
Exploiting AirSim as a Cross-dataset Benchmark for Safe UAV Landing and Monocular Depth Estimation Models.
DOI: 10.5220/0011562200003332
In *Proceedings of the 14th International Joint Conference on Computational Intelligence (IJCCI 2022)*, pages 454-462
ISBN: 978-989-758-611-8; ISSN: 2184-3236

safe landing area estimation models, we analysed the SafeUAV-Net-Large and Small semantic segmentation networks, also trained in SafeUAV and tested them on the AirSimNC dataset.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 introduces the six estimation models that were fully or partially trained on the SafeUAV dataset in order to obtain robust models that are specialized in bird's eye view (Marcu et al., 2019). Section 4 presents the novel AirSimNC dataset, and the experimental results obtained by the different estimation models. The depth estimation models introduced in Section 3 are compared to state-of-the-art monocular models in Section 4 too. Section 5 summarizes the most relevant conclusions of this paper.

# 2 RELATED WORK

## 2.1 Deep Learning-based Normal Estimation

Normal estimation consists of inferring the local orientation of the surfaces from RGB images. Deep Learning-based approaches have achieved state-of-the-art performance in surface normal predictions but they are limited due to the shortage of large and realistic outdoor datasets with ground truth information. Surface normal estimation is relevant to UAV applications as it can be used for safe landing area estimation.

While some approaches tried to estimate the surface normals using depth map predictions (Yang et al., 2018), it has been proved that estimating the surface normals independently leads to a better normal prediction performance (Zhan et al., 2019). Eigen et al. (Eigen and Fergus, 2015) estimated the surface normals using a multi-scale convolutional architecture in order to estimate the normal vector for each image pixel. Li et al. (Li et al., 2015) combined a deep CNN with a post-proccessing refining step based on Conditional Random Fields. Wang et al. (Wang et al., 2015) developed architectures that integrated local and global information from the input image, as well as information used by classical normal estimation methods. The GeoNet++ model, introduced by Qi et al. (Qi et al., 2020), incorporates a two-stream backbone neural network with an edge-aware refinement module for precise boundary detailed outputs.

## 2.2 Deep Learning-based Monocular Depth Estimation

Estimating the depth map from just one view is an ill-posed problem because many 3D scenes can have the same picture representation on the image plane. However, monocular methods require a much cheaper hardware, less computational complexity and no external calibrations or rectifications, at the expense of less accurate results compared to stereo methods.

Many current monocular approaches share a similar encoder-decoder architecture. The most popular encoders are EfficientNets (Bhat et al., 2020), ResNets (Laina et al., 2016), or variants of ResNet such as ResNext (Kim et al., 2020; Yin et al., 2019) or DenseNet (Alhashim and Wonka, 2018; Lee et al., 2019). For the decoder, Laina et al. (Laina et al., 2016) proposes an up-scaling decoder architecture based on up-convolution blocks, while Lee et al. (Lee et al., 2019) uses Local Planar Guidance layers. Skip connections are also popular in monocular depth architectures, which are usually followed by an attention mechanism (Kim et al., 2020), or a dilated residual block (Yin et al., 2019). Aich et al. introduced an architecture with bidirectional attention modules (Aich et al., 2021). Moreover, Zhang et al. exploits temporal consistency among consecutive video frames (Zhang et al., 2019).

## 2.3 Synthetic Datasets

Retrieving fully annotated ground truth from real scenes can be very expensive and complicated, specially from bird's eye view. Some existing datasets such as Okutama-Action (Barekatain et al., 2017) or VisDrone2018 (Zhu et al., 2018) are based on real drone view footage but only contain object detection ground truth (i.e. no depth maps or landing areas). Moreover, the publicly available depth and normal datasets either have sparse ground truth or employ colorization methods. Synthetic datasets are an attractive alternative for dense and accurate ground truth generation. There are different approaches for synthetic dataset generation: from open-source 3D animated films (Butler et al., 2012; Mayer et al., 2016), to vehicle simulators such as CARLA (Sekkat et al., 2022) or AirSim. The AirSim simulator has already been used in order to generate the VirtualCity3D (Liu and Huang, 2021) and UrbanScene3D datasets (Lin et al., 2022), which contain ground truth information about bounding boxes, instance segmentation or 3D pointclouds.

# 3 TRAINING on SafeUAV DATASET

Due to the absence of comparable and publicly available sets, we used the semi-synthetic SafeUAV dataset to train all the networks analysed in this paper. This dataset comprises RGB images from 3D-reconstructed urban and suburban areas retrieved from Google Earth, their corresponding dense depth maps and labelled images with the orientation of each surface. Pixels are labelled as 'horizontal', 'vertical' or 'other', and this task is referred to as HVO segmentation. This labelling is an oversimplification of the world, since not every horizontal surface is actually safe for drone landing (e.g. water or roads).

In order to generate monocular depth and HVO estimation models for UAV applications, different networks were trained on the SafeUAV dataset.

## 3.1 Studied Deep Learning-based Architectures

The monocular depth estimation architectures analyzed in this work are AdaBins and DenseDepth. AdaBins divides a given depth range into $N$ adaptive bins, and at the time of writing this paper, it presented the best compromise among the evaluation metrics in the KITTI Depth and NYUv2 benchmarks (Bhat et al., 2020). On the other hand, DenseDepth shows a strong compromise with its relatively low computational complexity and a marginal performance penalty (Alhashim and Wonka, 2018).

The landing area estimation networks analyzed in this paper were proposed in (Marcu et al., 2019). SafeUAV-Net-Large (or *Large model*), is presented as the model with the higher accuracy, while SafeUAV-Net-Small (or the *Small model*) is presented as ther faster and computationally less expensive model.

## 3.2 Implementation Details

We trained all of our networks on a single NVIDIA Tesla V100-SXM2 GPU with 32 GB memory. To avoid over-fitting, we used data augmentation techniques. We used the Adam Optimizer (Kingma and Ba, 2015), with 40 epochs, a learning rate of 0.0001, patience of 3 and a batch size of 8.

AdaBins got the best results in the KITTI and NYUv2 benchmarks when the number of bins was set to 256. We trained the AdaBins network for 256 and 80 bins, generating the *AdaBins-256* and *AdaBins-80* models, respectively.

Since DenseDepth showed worse results than AdaBins on the same benchmarks, we analyzed if its performance could be enhanced by training it on mixed datasets. We trained the DenseDepth network on the SafeUAV dataset, and then in a super-dataset containing the SafeUAV and NYUv2 datasets. This way, we generated two models: *DD-UAV* and *DD-UAV-NYU*.

Table 1 shows the size and inferring speed of the models. The speed values should only be considered as a reference to compare the complexity of each model. There are different tools that allow to reduce the inferring time of networks that have not been used on these models yet.

## 3.3 Experimental Results of Depth Models

Table 2 shows the evaluation metrics of our depth estimation models on the SafeUAV dataset. Overall, the AdaBins-256 models obtained better results than AdaBins-80. DD-UAV-NYU got considerably better error metrics than DD-UAV. In order to obtain a wider visualization of the performance of our models, we calculated two more parameters, as shown in Table 3: the structural similarity SSIM (Wang et al., 2004) and the median relative value (Median Rel) between the estimated and ground truth depth maps. The Median Rel indicates scaling errors in the estimations, and it is more robust against outliers than other mean error metrics. Table 3 shows that both DD-UAV and DD-UAV-NYU have a slight scaling error (the estimated depth values are 7.2% and 3.6% shorter than the ground truth depth values, respectively).

## 3.4 Experimental Results of HVO Models

Table 4 shows the classification evaluation metrics for the Large and Small models. The row 'Horizontal' shows the evaluation metrics when the classification problem is reduced to a 'horizontal/non-horizontal' binary classification model. The 'Other' and 'Vertical' rows were calculated in a similar way. For safe landing applications, the precision at predicting horizontal surfaces stands out in importance. Rather than being able to find all the horizontal areas from one frame, it would be more useful to have a model where all the predicted horizontal areas were actually horizontal. In that sense, the higher the horizontal precision is, the safer it will be to use the model for drone landing applications. In the SafeUAV dataset, the Large model shows a higher horizontal precision than the Small model.

Table 1: Number of parameters, size and inference time of each estimation model.

| Model | Number of parameters (M) | Model size (MB) | Tesla V100 (images/s) |
|---|---|---|---|
| AdaBins-80 | 78 M | 917 | 9.623 |
| AdaBins-256 | 78 M | 918 | 9.583 |
| DD-UAV | 42.6 M | 337 | 28.835 |
| DD-UAV-NYU | 42.6 M | 337 | 28.835 |
| SafeUAV-Net-Large | 24 M | 650 | 27.853 |
| SafeUAV-Net-Small | 1 M | 24 | 101.366 |

Table 2: Quantitative performance of our depth models on SafeUAV. The RMS error is shown in meters. Best results are in **bold**, while the second best results are underlined.

| Model | Accuracy (higher is better) | | | Error (lower is better) | | |
|---|---|---|---|---|---|---|
| | $\delta_1$ | $\delta_2$ | $\delta_3$ | Abs Rel | RMS | $\log_{10}$ |
| AdaBins-80 | 0.952 | 0.987 | 0.992 | 0.084 | 6.425 | 0.033 |
| AdaBins-256 | 0.972 | 0.993 | 0.996 | **0.065** | **5.587** | **0.026** |
| DD-UAV | 0.943 | **0.998** | **1** | 0.130 | 11.118 | 0.052 |
| DD-UAV-NYU | **0.973** | 0.996 | 0.999 | 0.068 | 8.451 | 0.030 |

Table 3: SSIM and Median Rel of our depth estimation models on SafeUAV. Best results are in **bold**, second best results are underlined.

| Model | SSIM | Median Rel |
|---|---|---|
| AdaBins-80 | 0.331 | **0.004** |
| AdaBins-256 | 0.318 | -0.009 |
| DD-UAV | **0.583** | 0.072 |
| DD-UAV-NYU | 0.524 | 0.036 |

# 4 TESTS on AirSim SIMULATOR

Datasets tend to have strong built-in biases. Therefore, rather than training and testing models on subsets from the same datasets, a more faithful real-world performance is obtained by evaluating the networks on a completely different dataset from the one it was trained on. This is known as the zero-shot cross-dataset performance. In order to analyze the cross-dataset performance of our trained models, we generated a synthetic dataset using the AirSim simulator. AirSim is an open-source simulator for drones, cars and other vehicles, and it creates accurate and real-world environments by taking advantage of the rendering, physics and perception computation of Unreal Engine. By exploiting the photorealism of Unreal Engine and the multiple scene settings that are reproducible with AirSim, we generated a 3D photorealistic dataset which is diverse in weather conditions and camera effects such as motion blur or noise.

## 4.1 Synthetic AirSimNC Dataset Generation

Our synthetic dataset, called AirSimNC, was created using the suburban AirSimNH and urban CityEnviron AirSim environments. The synthetic images were retrieved from random drone positions with fixed pitch and roll angles of 45°and 0°, respectively, and a uniformly randomized elevation between 30 and 90 meters. From each perspective, RGB, depth and normal information was retrieved. In order to generate a semantic segmentation map that classifies the image surfaces as horizontal, vertical or other, the normal maps were retrieved using the simulator. Then, the normal vector was computed for each pixel in the normal map. Afterwards, the angle between the horizontal plane and the normal vector was calculated for each pixel. Finally, each pixel was classified as horizontal, vertical or sloped surface by applying empirically set thresholds into its corresponding angle.

## 4.2 Experimental Results of Depth Models

We analyzed the performance of the four depth estimation models introduced on Section 3. We also evaluated the performance of the AdaBins and DenseDepth networks when they are trained on the KITTI and NYUv2 datasets.

The first half of Table 5 shows the evaluation metrics of the models in the AirSimNC dataset (The $\log_{10}$ error could not be calculated for the DD-KITTI and DD-NYU models because of undefined terms). The estimations suffer from a strong scaling error, as shown in the Median Rel values of Table 6 (in

Table 4: Classification metrics of the HVO models on SafeUAV. Row 'Average' shows the average of the metrics in the three classes. Row 'Reported' shows the metrics claimed by (Marcu et al., 2019).

| | SafeUAV | **Accuracy** | **Precision** | **Recall** | **IoU** |
|---|---|---|---|---|---|
| Large model | Horizontal | 0.827 | 0.756 | 0.725 | 0.587 |
| | Other | 0.725 | 0.674 | 0.751 | 0.551 |
| | Vertical | 0.852 | 0.678 | 0.559 | 0.442 |
| | **Average** | **0.801** | **0.703** | **0.678** | **0.527** |
| | **Reported** | **0.846** | **0.761** | **0.748** | **0.607** |
| Small model | Horizontal | 0.817 | 0.703 | 0.804 | 0.600 |
| | Other | 0.733 | 0.715 | 0.675 | 0.532 |
| | Vertical | 0.860 | 0.689 | 0.610 | 0.478 |
| | **Average** | **0.804** | **0.702** | **0.696** | **0.537** |
| | **Reported** | **0.823** | **0.728** | **0.693** | **0.551** |

Table 5: Quantitative performance of different depth models on the AirSimNC dataset. Best results in each category are in **bold**, while the second best results are underlined. Rows with an asterisk represent the metrics for each model, once the estimated depth maps were scaled to the ground truth median depth.

| Model | Accuracy (higher is better) | | | Error (lower is better) | | |
|---|---|---|---|---|---|---|
| | $\delta_1$ | $\delta_2$ | $\delta_3$ | Abs Rel | RMS | $\log_{10}$ |
| AdaBins-80 | **0.388** | **0.623** | **0.804** | 0.608 | **42.592** | **0.191** |
| AdaBins-256 | 0.249 | 0.504 | 0.724 | 0.930 | 51.104 | 0.233 |
| AdaBins-256-KITTI | 0.291 | 0.539 | 0.725 | 0.615 | 52.469 | 0.218 |
| AdaBins-256-NYU | 0.166 | 0.339 | 0.519 | 1.409 | 70.165 | 0.309 |
| DD-UAV | 0.240 | 0.396 | 0.507 | 0.530 | 107.630 | 0.383 |
| DD-UAV-NYU | 0.223 | 0.444 | 0.647 | **0.447** | 60.149 | 0.249 |
| DD-KITTI | 0.208 | 0.407 | 0.585 | 1.041 | 63.518 | - |
| DD-NYU | 0.091 | 0.190 | 0.302 | 3.114 | 76.365 | - |
| * AdaBins-80 | **0.588** | 0.802 | 0.900 | 0.338 | 33.442 | 0.124 |
| * AdaBins-256 | 0.567 | 0.801 | 0.899 | 0.393 | 33.893 | 0.128 |
| * AdaBins-256-KITTI | 0.435 | 0.745 | 0.848 | 0.387 | 42.254 | 0.156 |
| * AdaBins-256-NYU | 0.415 | 0.688 | 0.840 | 0.380 | 46.670 | 0.164 |
| * DD-UAV | 0.496 | 0.762 | 0.891 | 0.338 | 97.095 | 0.135 |
| * DD-UAV-NYU | 0.532 | **0.808** | **0.920** | **0.293** | 52.056 | **0.123** |
| * DD-KITTI | 0.239 | 0.459 | 0.649 | 0.958 | 61.548 | - |
| * DD-NYU | 0.258 | 0.492 | 0.681 | 0.837 | **21.257** | - |

our AdaBins-80-UAV model, for example, the estimated depths are 27.3% higher than the ground truth depths, while the depth maps estimated by our DD-UAV models are 66.2% shorter than the real depth maps).

In a real-world application, a model could be calibrated in a well-known environment in order to diminish the scaling error. To simulate such calibration, we multiplied each estimated depth map with a scaling factor, so that the median of the scaled depth map matches the median of the ground truth (which is similar to the procedure explained in (Alhashim and Wonka, 2018)). The second half of Table 5 shows the evaluation metrics of the depth estimation models, once the scaling correction is performed.

Our four depth models achieved better scaled re-

Table 7: Evaluation metrics of the Large and Small models on the synthetic AirSimNC dataset.

| AirSimNC | | **Accuracy** | **Precision** | **Recall** | **IoU** |
|---|---|---|---|---|---|
| Large model | Horizontal | 0.664 | 0.724 | 0.483 | 0.404 |
| | Other | 0.567 | 0.252 | 0.625 | 0.215 |
| | Vertical | 0.700 | 0.467 | 0.281 | 0.219 |
| | **Average** | **0.640** | **0.481** | **0.463** | **0.279** |
| Small model | Horizontal | 0.634 | 0.738 | 0.366 | 0.324 |
| | Other | 0.558 | 0.251 | 0.629 | 0.220 |
| | Vertical | 0.673 | 0.379 | 0.273 | 0.197 |
| | **Average** | **0.622** | **0.456** | **0.423** | **0.247** |

Table 6: SSIM and Median Rel metrics of depth models on the AirSimNC dataset. Best results are in **bold**, second best results are underlined.

| Model | SSIM | Median Rel |
|---|---|---|
| AdaBins-80 | 0.099 | -0.175 |
| AdaBins-256 | 0.092 | -0.564 |
| AdaBins-256-KITTI | **0.178** | -0.125 |
| AdaBins-256-NYU | 0.057 | -1.185 |
| DD-UAV | 0.121 | 0.524 |
| DD-UAV-NYU | 0.147 | 0.205 |
| DD-KITTI | 0.002 | **-0.044** |
| DD-NYU | -0.001 | -1.880 |

sults than the other four models in almost every metric, even if the later models were trained on much larger and realistic datasets. The best evaluation metrics were obtained by the (scaled) DD-UAV-NYU model in most parameters, followed by AdaBins-80. This shows how monocular depth models can benefit from mixing depth datasets during the training process.

Figure 1 shows estimations of our depth models, compared to the ground truth depth. The estimations in AirSimNH (second column) suffer from outliers: in high-frequency objects in the AdaBins models, and in the upper part of the depth maps in the DenseDepth estimations.

### 4.3 Experimental Results of HVO Models

Table 7 shows the evaluation metrics for the HVO models on the AirSimNC dataset. For both models, the precision for horizontal surfaces is higher than 72%, with barely no cross-dataset penalty. (the Small model shows a slightly higher precision). Out of the estimated 'horizontal' pixels that were incorrectly labelled, most belonged to the 'other' class.

Figure 2 shows a visual comparison of the HVO estimation models. The number of 'other' pixels is higher in the estimations than in the ground truth (the HVO models have learned to 'play it safe' around uncertain or borderline surfaces). The Large model provides more detailed estimations (the edges of surfaces and objects are more discernible in the Large estimations), while the estimations of the Small models are more vague overall.

## 5 CONCLUSION

This paper introduced the synthetic and photorealistic AirSimNC dataset, which was employed as a benchmark to compare the cross-dataset performance of depth and HVO estimation models. Regarding the depth estimation models, the ones trained on SafeUAV beat other state-of-the-art models in our benchmark. However, we also checked that mixing a semi-synthetic dataset with bird's eye view (SafeUAV) and a real dataset with more realistic images (NYUv2) leads to a great improvement in the performance of the DenseDepth network. Overall, the models trained of bird's eye view obtained decent results in our benchmark, up to a scaling error. Regarding the safe landing area estimation models, they showed barely any cross-dataset penalty at the precision of horizontal surfaces.

During this paper, we have regarded depth and landing area estimation as independent tasks. The geometric relationship between depth and surface nor-

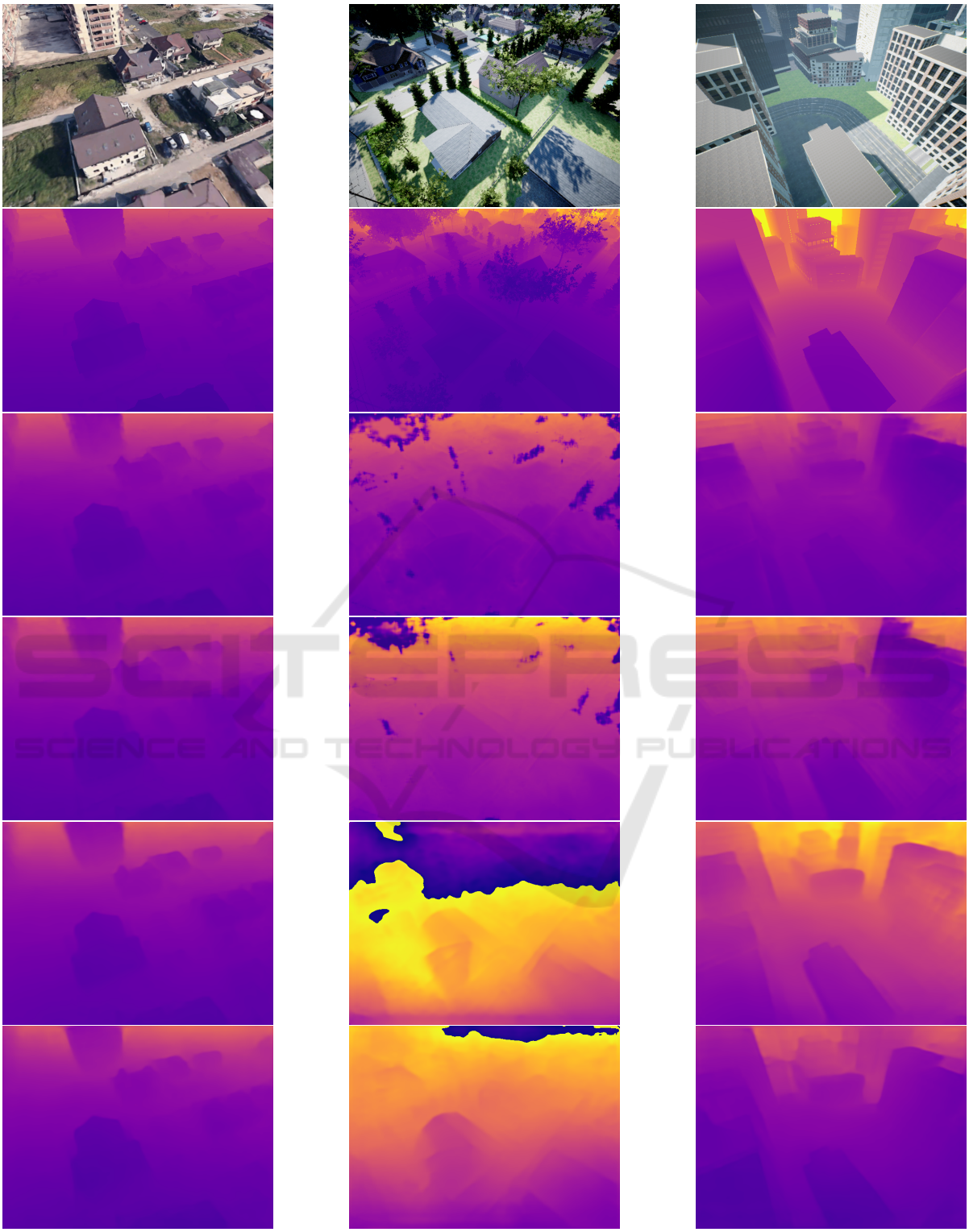

Figure 1: Qualitative comparison of our depth estimation models. First row: RGB images from SafeUAV, AirSimNH and CityEnviron, respectively. Second row: ground truth depth maps. Rows 3 to 6: estimations by the AdaBins-80, AdaBins-256, DD-UAV and DD-UAV-NYU.

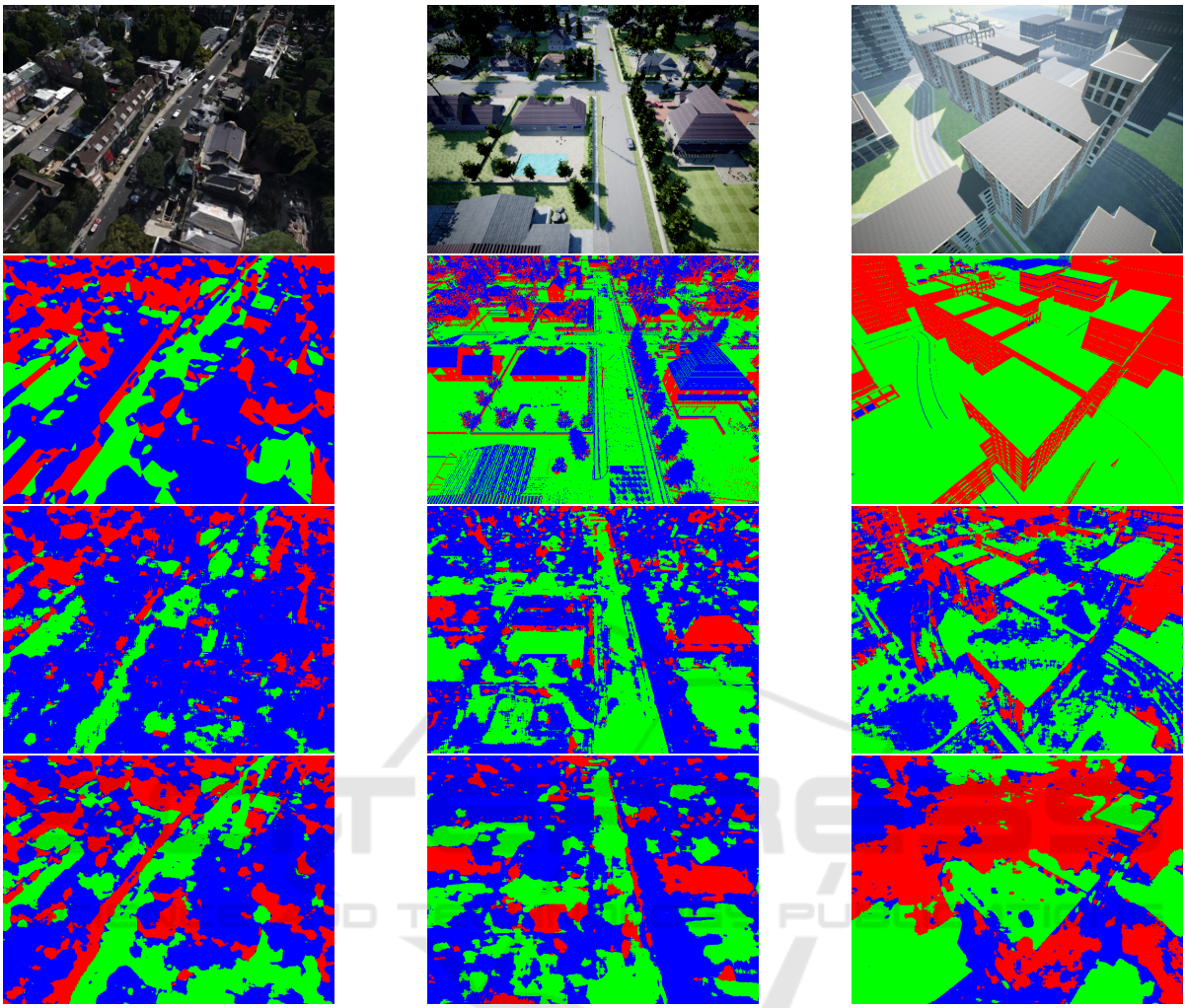

Figure 2: Qualitative comparison of the HVO estimation models. First row: RGB images from SafeUAV, AirSimNH and CityEnviron, respectively. Second row: ground truth HVO labels. Rows 3 to 4: Large and Small model predictions.

mals could be exploited so that a single model performs both estimation tasks simultaneously. The development of such network is left as future work.

## ACKNOWLEDGEMENTS

This work has received funding from Basque Government under project CODISAVA of the program ELKARTEK (ETORTEK)-2020.

## REFERENCES

Aich, S., Uwabeza Vianney, J. M., Amirul Islam, M., and Bingbing Liu, M. K. (2021). Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11746–11752.

Alhashim, I. and Wonka, P. (2018). High quality monocular depth estimation via transfer learning. *arXiv e-prints*, abs/1812.11941.

Barekatain, M., Marti, M., Shih, H., Murray, S., Nakayama, K., Matsuo, Y., and Prendinger, H. (2017). Okutama-action: An aerial view video dataset for concurrent human action detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2153–2160, Los Alamitos, CA, USA. IEEE Computer Society.

Bhat, S. F., Alhashim, I., and Wonka, P. (2020). Adabins: Depth estimation using adaptive bins. *arXiv*, pages 1–13.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intel-*

*ligence and Lecture Notes in Bioinformatics)*, 7577 LNCS(PART 6):611–625.

Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2650–2658.

Kim, D., Lee, S., Lee, J., and Kim, J. (2020). Leveraging Contextual Information for Monocular Depth Estimation. *IEEE Access*, 8:147808–147817.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15.

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 239–248.

Lee, J. H., Han, M.-K., Ko, D. W., and Suh, I. H. (2019). From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*.

Li, B., Shen, C., Dai, Y., van den Hengel, A., and He, M. (2015). Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1119–1127.

Lin, L., Liu, Y., Hu, Y., Yan, X., Xie, K., and Huang, H. (2022). Capturing, reconstructing, and simulating: the urbanscene3d dataset. In *ECCV*.

Liu, Y. and Huang, H. (2021). Virtualcity3d: A large scale urban scene dataset and simulator.

Marcu, A., Costea, D., Licăreţ, V., Pîrvu, M., Sluşanschi, E., and Leordeanu, M. (2019). SafeUAV: learning to estimate depth and safe landing areas for UAVs from synthetic data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11130 LNCS:43–58.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:4040–4048.

Qi, X., Liu, Z., Liao, R., Torr, P. H., Urtasun, R., and Jia, J. (2020). GeoNet++: Iterative Geometric Neural Network with Edge-Aware Refinement for Joint Depth and Surface Normal Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Sekkat, A. R., Dupuis, Y., Kumar, V. R., Rashed, H., Yogamani, S., Vasseur, P., and Honeine, P. (2022). Synwoodscape: Synthetic surround-view fisheye camera dataset for autonomous driving. *IEEE Robotics and Automation Letters*, 7(3):8502–8509.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2017). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*.

Wang, X., Fouhey, D. F., and Gupta, A. (2015). Designing deep networks for surface normal estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June(Figure 2):539–547.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Yang, Z., Wang, P., Xu, W., Zhao, L., and Nevatia, R. (2018). Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 1(c):7493–7500.

Yin, W., Liu, Y., Shen, C., and Yan, Y. (2019). Enforcing geometric constraints of virtual normal for depth prediction. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:5683–5692.

Zhan, H., Weerasekera, C. S., Garg, R., and Reid, I. (2019). Self-supervised learning for single view depth and surface normal estimation. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:4811–4817.

Zhang, H., Shen, C., Li, Y., Cao, Y., Liu, Y., and Yan, Y. (2019). Exploiting temporal consistency for real-time video depth estimation. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:1725–1734.

Zhu, P., Wen, L., Bian, X., Ling, H., and Hu, Q. (2018). Vision Meets Drones: A Challenge. pages 1–11.