A System for Image-Based Non-Line-Of-Sight Detection Using Convolutional Neural Networks

Clarissa Boker, Joshua Niemeijer, Nicolai Wojke, Cyril Meurie, Yann Cocheril

A System for Image-Based Non-Line-Of-Sight Detection Using Convolutional Neural Networks

Clarissa Böker[1], Joshua Niemeijer[1], Nicolai Wojke[1], Cyril Meurie[2] and Yann Cocheril[2]

Abstract— **The ERSAT GGC project introduces the concept of virtual balises for train localization, which avoids investment and maintenance costs of physical balises. Since this concept relies on the matching of train positions to balise positions stored in a database, it is dependent on placing virtual balises in track areas with unimpeded GNSS reception. One factor majorly contributing to the distortion of GNSS signals is the non-line-of-sight (NLOS) scenario where the direct path between a satellite and the receiver on the train is blocked. As these NLOS situations result in deflections or the total absence of GNSS signals, this paper proposes a system to identify obstacles occluding the visibility of satellites above the tracks traversed by a train. This is achieved by video recording the sky from the roof of the train and segmenting the images into sky and non-sky regions. The line-of-sight status of individual satellites is found through projecting the known satellite locations into the segmented images. Consequently, the information whether a satellite is located in a sky or non-sky segment of the image allows for a determination of the GNSS performance at any observed track area.**

I. INTRODUCTION

The localization of trains is currently accomplished by measuring the traveled distance with an on-board odometer and resetting the accumulating error at intervals through balises located along the tracks. These physical balises are installed on the ties of the trackbed and send out the balise information via telegrams to be received by a counterpart module on the train, the balise transmission module (BTM). In order to avoid the investment and maintenance costs for physical balises, the ERSAT GGC project introduces the concept of virtual balises. While the balise information contained in the virtual balises is identical to the physical balises, the transmission is accomplished with an on-board virtual balise reader (VBR) instead of through the BTM. For this mode of transmission, the position of a GNSS antenna on top of the train is projected onto the track and computed periodically by the VBR. Every time the projected antenna position matches the position of a virtual balise stored in a track database, the balise information and the reference position of the virtual balise is exchanged. For this process the placement of the virtual balises in track areas with unimpeded GNSS service is crucial.

However, conditional on global as well as local influences, the quality of the GNSS service typically varies. While global influences are reducible by augmentation technologies [1], local influences such as multipath, radio interference and NLOS errors strongly depend on the surrounding of the GNSS antenna. This necessitates a classification method for identifying the suitability of railway track areas for placing virtual balises. One local threat to the position accuracy is presented by reflecting obstacles along the tracks, as they commonly result in multipath errors, which are caused by multiple echoes of a GNSS signal arriving with a delay compared to the direct signal path. Since the power level of the signal is very low after having traveled from the satellites to the receiver, an additional threat is posed by unrelated communication transmitted at frequencies close to the GNSS receiver frequency, as this might lead to interference. Another major error source for the position estimation is the NLOS phenomenon, which occurs due to blockage of the direct signal path between the satellites and the receiver on the train. This error mainly originates from obstacles, such as buildings or vegetation, blocking the visibility of relevant satellites, resulting in the reception of either a reflected signal or no signal at all. Therefore, we propose a method focusing on detecting and reducing NLOS errors by identifying the occlusion of the sky above the GNSS receiver, through images of the sky recorded with an upwards oriented omnidirectional camera mounted on top of the train. With the aid of a deep convolutional neural network (CNN), the captured sky images are subsequently segmented into two classes, sky and background, in order to determine whether satellites are located in sky or non-sky regions of the image. Based on the image segmentation process, the distinction between satellite positions in sky or non-sky regions allows for identifying the reception state of the satellites and consequently for forecasting the GNSS service performance.

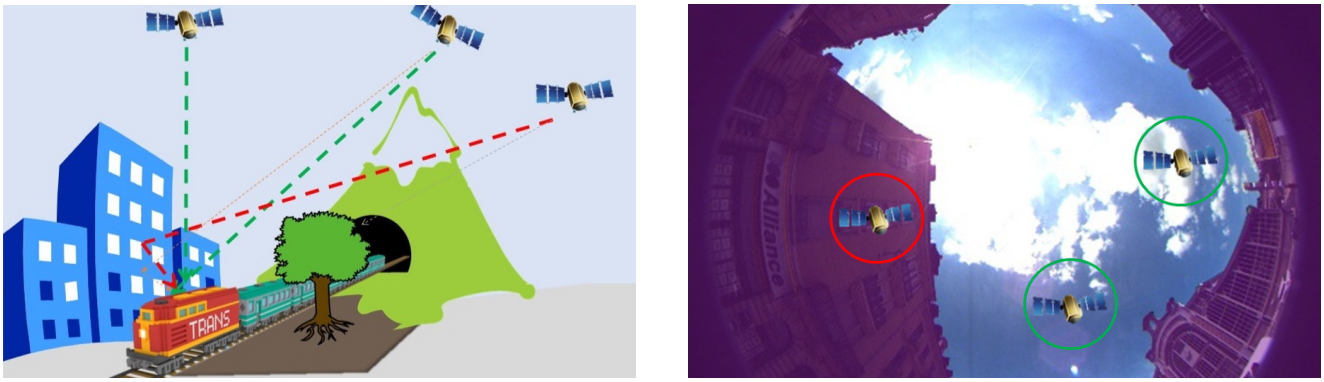
The remainder of this paper is structured as follows. Section II provides an overview of related work on detecting NLOS situations. Section III explains the physical build up used for our NLOS detection system in greater detail, including the camera model and calibration process. Subsequently, Section IV describes our approach to sky segmentation with CNNs, before the conducted experiments are presented in Section V. Finally, the main aspects of the paper are summarized in Section VI.

II. RELATED WORK

In radio localization, the position of a mobile node (in our case a train) is estimated from time-of-arrival measurements of multiple reference stations (in our case GNSS satellites).

[1]Clarissa Böker, Joshua Niemeijer, Nicolai Wojke are with the Institute of Transportation Systems, German Aerospace Center (DLR), 38108 Braunschweig, Germany, firstname.lastname@dlr.de

[2]Cyril Meurie, Yann Cocheril are with Univ Lille Nord de France, F-59000 Lille IFSTTAR, COSYS, LEOST, F-59650 Villeneuve d'Ascq, France, firstname.lastname@ifsttar.fr



(a) (b)

Fig. 1: (a) illustrates the GNSS non-line-of-sight detection system. Satellite positions are projected into the image plane of an omnidirectional camera mounted on the roof of a vehicle. (b) shows an image of the omnidirectional camera with the segmentation overlay in violet. Line-of-sight status is based on image segmentation results. The illustrated scenario contains (green) satellites in line-of-sight (i.e., in sky regions) and (red) satellites that are occluded by surrounding buildings (i.e., in non-sky regions).

Within this context, localization in NLOS scenarios has been well studied [2].

One possibility to deal with NLOS comprises the detection of reference stations that are subject to NLOS in order to discard their measurements and utilize only the remaining measurements for locating the mobile node. Following this framework, Riba and Urruela [3] formulate the search for NLOS paths as a statistical hypothesis test. They model the conditional probability density function of observing a time-of-arrival measurement for the NLOS and LOS case for all reference stations. Based on this criterion, they compute the most likely mapping between LOS/NLOS signals and reference stations.

A practical issue with the approach taken in [3] is the requirement of exact measurement likelihood functions both for the LOS and NLOS case, which are difficult to obtain in real-world applications. Simpler alternatives, which do not require exact likelihood functions, can be implemented within the maximum likelihood framework by weighting measurements based on heuristics and rejecting measurements with large/inconsistent error residuals [4].

Instead of rejecting NLOS measurements, methods such as the one proposed by [5] are able to incorporate NLOS measurements into the position estimation. This is beneficial in adverse environments without sufficient availability of LOS reference stations. Even when exact error distributions are not needed, knowledge of the LOS/NLOS state of reference stations simplifies the problem and improves localization accuracy.

Compared to above methods, we rely on additional hardware to determine the LOS/NLOS state of reference stations. In this regard, our work is most similar to those of [6], [7], [8], and [9] realized in the context of autonomous vehicles in urban environment. Peyraud *et al.* [6] design an integrated system for localization in NLOS scenarios based on an extended Kalman filter. During prediction, their system utilizes road network information to enhance the precision of the motion model. During update, geometric 3D map data is utilized in order to exclude NLOS reference stations from the measurement correction of the predicted state. Our image segmentation method could be directly integrated into their system, eliminating the need to generate accurate 3D models for the area of deployment and thereby lowering system complexity. From a similar point of view, Attia *et al.* [7] and Marais *et al.* [8] use image processing to classify sky and non-sky regions in images acquired by a vehicle equipped with a GNSS-RTK receiver and an upwards oriented omnidirectional camera. Compared to our work, they use a hand-crafted image processing pipeline for the segmentation [9].

Other integrated systems which model the localization problem as a recursive Bayes filter have been proposed in [10] and [11]. Huerta *et al.* [10] use a combination of particle and unscented Kalman filter to jointly estimate the mobile node location and the LOS/NLOS state of reference stations. Yousefi *et al.* [11] design an unscented Kalman filter, which projects sigma points of the unscented transformation onto the feasible region. In contrast to a conventional unscented Kalman filter formulation, their algorithm is more efficient and has a better numerical stability. Our image based segmentation method could be integrated into the above mentioned systems as an additional information source, e.g., to fix the LOS/NLOS state of individual reference stations.

III. GNSS PERFORMANCE ESTIMATION

In order to determine the quality of the GNSS system at any given location within the railway map, it is necessary to identify GNSS satellites which deliver unimpaired signals. In the following, we formulate our pipeline for detecting NLOS satellites by using camera images, in order to find locations suitable for virtual balise placement.

The physical sensor setup of our NLOS detection system is illustrated in Fig. 1. An omnidirectional camera with an opening angle of approximately 180° is mounted on top of

a moving train. This camera takes images at fixed intervals of up to 30 frames per second. For the NLOS detection task, we define a function $f : \mathbb{R}^3 \rightarrow \{0,1\}$ which outputs for any location in a reference world coordinate system $\mathbf{x}_w$ whether the satellite is subject to NLOS or LOS:

$$f(\mathbf{x}_w) = \begin{cases} 1 & \text{if } p(l = \text{Sky} \mid \mathbf{R}\mathbf{x}_w + \mathbf{t}, \mathbf{I}) > \theta_{\min}, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $l \in \{\text{Sky}, \text{Object}\}$ is a binary segmentation label that classifies locations as either sky or object, $(\mathbf{R}, \mathbf{t})$ are the rotation matrix and translation vector of the train pose at image acquisition time, $\mathbf{I}$ is the acquired camera image, and $\theta_{\min}$ is a probability threshold. For this, we assume the alignment of the camera mounted on the train to be known.

Our implementation of the NLOS detector in Equation 1 is a two-step process. First, a binary image segmentation mask is generated from the raw camera image using a CNN as detailed in Section IV. Using this method, we obtain a mask which outputs a probability for the label assignment l at every pixel location in $\mathbf{I}$. Secondly, we transform all satellite positions into camera coordinates according to $\mathbf{R}\mathbf{x}_w + \mathbf{t}$ based on the current train pose. Subsequently, we project the satellites into the image using known intrinsic camera calibration. In the image we look up the label probability and perform thresholding. For the omnidirectional camera utilized in our physical sensor setup, we rely on camera parameters obtained with the omnidirectional camera calibration toolbox of Scaramuzza *et al.* [12]. We use $\theta_{\min} = 0.5$ in our implementation as it is the maximum a posteriori label assignment for the two-class problem.

One critical component in the implementation of the image-based NLOS detection system is the acquisition of reliable train and camera pose estimates $(\mathbf{R}, \mathbf{t})$, which are necessary to project satellite positions into the camera coordinate frame. Within the scope of the ERSAT GGC project, mapping is performed using a train equipped with specialized high precision hardware that performs well even in NLOS scenarios[1]. Thus, sufficiently reliable pose estimates are available to us at all times. However, recursive state estimation such as in [10], [11] can be used to alleviate temporal NLOS situations, for more relaxed deployment conditions.

IV. SKY SEGMENTATION

In the following section we provide an overview of the sky segmentation process entailing a CNN as its core component. Therefore, we give a brief overview on the segmentation method and two different backbone architectures which have been implemented and evaluated in this work. We also discuss our data augmentation and post processing steps.

A. Sky segmentation with convolutional neural networks

CNNs are a specialized form of neural networks tailored to perform advantageous on image and audio data. Contrary to regular neural networks, the neurons in CNNs are arranged in multidimensional layers, which in turn create several feature maps. All neurons within a feature map share the same weights and hence perform identical operations on different image regions, which increases the invariance of the network towards geometrical rearrangements of features [14] Given a sufficient amount of training images, CNNs show advantageous results for processing images captured in different surroundings as well as different weather and lighting conditions. Since this quality proves very beneficial for our use case of distinguishing between sky and non-sky pixels in the recorded omnidirectional images, we implement the segmentation in form of a CNN.

a) DeepLab: We utilize the DeepLab architecture proposed by [15], as we aspire a dense pixelwise semantic segmentation. We chose this model, because it extends the context observed by the convolution filters without increasing the number of parameters or the computational effort. This is achieved through replacing max-pooling layers with dilated kernels in convolutional layers, also known as atrous convolution. Additionally atrous convolution enlarges the field of view in the filters and offers us control over the conflicting priorities of accurate localization and increased context assimilation. DeepLab further introduces atrous spatial pyramid pooling (ASPP) to explicitly account for the segmentation of objects at multiple scales, which in practice amounts to employing multiple parallel atrous convolutional layers with different sampling rates and fusing the extracted features to generate a final result.

b) ResNet: We utilize ResNet-101 as a pre-trained network backbone with convolutional instead of the fully connected layers, as suggested by [15]. For additional improvement Chen *et al.* [16] extend this DeepLab network by adding a decoder module to refine the segmentation along the object boundaries, while the encoder module captures rich semantic information.

c) MobileNet: Changing the backbone for further enhancement of speed and memory efficiency [17] suggest MobileNet as an alternative. MobileNet is especially efficient in decreasing the required computational resources during inference. The approach is based on adding residual bottleneck layers to a convolutional neural network, which consist of a linear bottleneck with depthwise separable convolution and are directly connected via shortcuts similar to classical residual connections.

In order to achieve the best trade off between computational efficiency and the labeling precision we compare both the ResNet and the MobileNet architectures described above and report our results on the dataset obtained by our omnidirectional camera in Section V.

B. Data augmentation

While many existing semantic segmentation datasets include sky in their label set (e.g., [18], [19], [13]) and thus provide rich training resources on still images, during development we found that the system performance degrades when applied to our omnidirectional images. In order to

[1]However, due to cost effectiveness constraints, we cannot expect this hardware to be available on all future trains.

(a) (b) (c) (d)

Fig. 2: Exemplary output on SkyFinder [13]: (a) shows the ground truth label for all images of camera 4801, (b) and (c) show segmentation results of two corresponding training images, and (d) shows segmentation results of a model trained on ADE20K. As can be seen in (b) and (c), the network that has been trained on SkyFinder memorizes the skyline.

render images from public datasets closer to our target application domain, we distort every image presented to the network during training according to an omnidirectional lens effect with random model parameters as follows.

Let (x_c, y_c) denote a pixel coordinate in the original undistorted camera image and (x_s, y_s) its location in the corresponding synthetic omnidirectional camera image. We assume both images to share a common principal point (x_h, y_h) and obtain the polar coordinates of (x_s, y_s) by

$$\begin{pmatrix} r_s \\ \theta_s \end{pmatrix} = \begin{pmatrix} \sqrt{(x_s - x_h)^2 + (y_s - y_h)^2} \\ \tan^{-1}(y_s / x_s) \end{pmatrix} \tag{2}$$

and likewise (r_c, θ_c) the polar coordinates of (x_c, y_c). Subsequently we map between the two images using an equidistance projection [20]:

$$r_c = f \cdot \tan^{-1}\left(\frac{r_s}{f}\right), \tag{3}$$

$$\theta_c = \theta_f, \tag{4}$$

where f is a focal length parameter sampled uniformly from $[0.7, 3]$ for normalized coordinates $x_c, y_c, x_s, y_s \in [-1, 1]$.

Additional to the omnidirectional lens effect, we also apply random rotation and image flipping to enhance the overall variety of training data.

C. Conditional Random Field post processing

Naturally the spatial accuracy of CNNs is limited due to their built-in invariance to spatial transformations, which is necessary for learning increasingly abstract data representations. Since this inhibits dense prediction tasks such as semantic segmentation, the post processing step aims to further improve the level of detail in the segmentation, especially along object boundaries. This is achieved by applying a fully connected conditional random field (CRF) on top of the CNN, as proposed by [15].

In our implementation, we utilize the fully connected pairwise CRF of Krahenbuhl *et al.* [21], with the unary potential build from the negative log-probability of the CNN pixel classifier and a bilateral pairwise potential to include information about the neighborhood structure. The standard deviation of the position and color terms in the bilateral kernel have both been fixed to 13 throughout all experiments.

V. EXPERIMENTS

In this section, we perform a series of experiments with different network architectures for the purpose of sky segmentation, which is the core methodology of our NLOS detection system.

A. Dataset

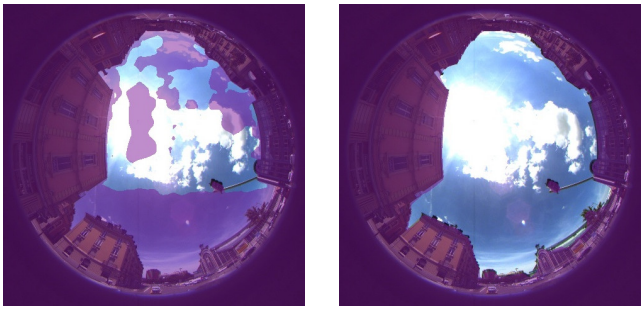
We utilize three datasets in our evaluation. The CAPLOC dataset [8] consists of 150 labeled images collected in the city of Belfort in France. This dataset is our target domain dataset as it has been collected with the same image acquistion system that is deployed in the ERSAT GGC project. In addition, we use images of SkyFinder [13] and ADE20K [18], [19]. SkyFinder is a sky segmentation dataset that contains images collected from public weather cameras. We use a subset of 38 423 images which has been extracted by simple filtering operations. More specifically, we only use images that are attributes with weather conditions clear, partly cloudy, scattered clouds, overcast, and mostly cloudy. In addition, we have filtered out images where the intensity in the upper half of the image is less than 30% in order to exclude images captured at night. We use 23098 images for training and 7667 images each for validation and testing. Both splits contain images from non-overlapping camera sets. ADE20K is a general scene parsing dataset containing indoor and outdoor scenes. Out of all images that contain sky regions, we use a random subset of 8240 images for training and 800 images for validation.

B. Evaluation Protocol

All backbone architectures have been initialized with models pre-trained on ImageNet. For fine-tuning to the segmentation task, we used the AMSgrad optimizer [22] with initial learning rate set to 1×10^{-4}. The learning rate was decayed by a factor of 0.5 every 5 epochs. For regularization, weight decay of 1×10^{-10} was used. These parameters have been found empirically through a series of experimental runs. We used a separate validation set to monitor model performance and early stopping to prevent overfitting. The fisheye data augmentation technique described in Section IV-B was used for numbers reported on CAPLOC. On all datasets, we report performance in terms of the mean intersection-over-union (Mean IOU) metric and pixel accuracy.

TABLE I: Comparison of segmentation accuracy on SkyFinder [13] for training on different datasets.

Architecture	Training Set	Mean IoU	Pixel Accuracy
MobileNet	SkyFinder	0.9792	0.9896
MobileNet	ADE20K	0.9485	0.9739



(a) (b)

Fig. 3: Segmentation results obtained with the MobileNet model. (a) shows inference results from the model trained on SkyFinder [13], (b) shows inference results from the model trained on ADE20K [18].

C. Results on SkyFinder

SkyFinder is the largest out of the three datasets considered in this paper. With roughly 38 000 images and large variety in weather conditions, it seems well suited for training and evaluation of sky segmentation CNNs. Yet, we found SkyFinder mostly unsuitable for our application.

In Table I we compare the performance on SkyFinder for two MobileNet models trained on SkyFinder and ADE20K. The difference in Mean IOU is roughly 0.03, which is surprisingly small given the difference in dataset domains and sizes. At the same time, we found networks that have been trained on SkyFinder to generalize poorly to other datasets. Applying a network trained on SkyFinder to CAPLOC results in a reduction of mean intersection-over-union from 0.9430 to 0.5051 compared to a model trained on ADE20K (cf. Table II and Fig. 3).

We suspect the poor generalization of SkyFinder is due to a very specific annotation procedure. In order to reduce the annotation effort, a single label image has been created for each of the 53 cameras. Depending on weather conditions, the annotated skyline is of varying accuracy for individual camera images. This specific dataset characteristic yields models that occasionally memorize the skyline of individual camera images. Fig. 2 shows an example of such a scenario. For our application this behavior is problematic as it hinders generalization to unseen sceneries and also renders evaluation metrics less accurate.

D. Results on CAPLOC

In the second experiment we evaluate model performance on our target domain dataset. CAPLOC itself contains too few images for training a neural network. Therefore, we train models on SkyFinder and ADE20K and use CAPLOC images for evaluation only. Table II provides results of a shoot-out of different training datasets and CNN architectures. The training set *ADE20K + SkyFinder* refers to a custom dataset containing all training images of ADE20K and 2000 manually selected images from SkyFinder that increase the variability of sky appearance. We also add a number of results from previous work [9] which is based on a hand-crafted image processing pipeline.

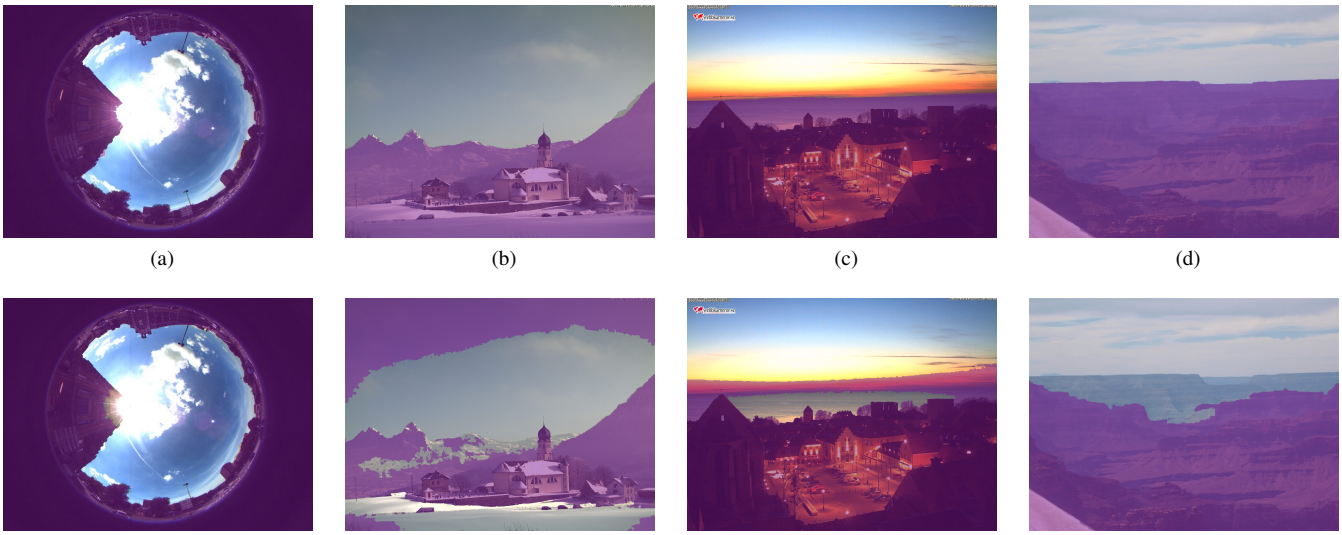
With respect to CNN architectures, we find that the ResNet 101 backbone provides a small improvement over MobileNet. This improvement comes at the cost of additional computational complexity: MobileNet runs at approximately 15 ms per image of 640x480 resolution on a Nvidia 1080 GTX, the ResNet 101 takes almost twice as long, about 27 ms. These results suggest that light-weight architectures provide decent sky segmentation results at low computational cost. Potentially, sky segmentation does not require very deep architectures with deep semantic features as common in other segmentation tasks. This hypothesis is further underlined by the fact that baseline methods of [9] achieve competitive results on CAPLOC.

Second, we find that the CRF post processing provides a consistent improvement in terms of classification accuracy over the bare CNN results. While this improvement is small in terms of figures provided in Table II, we observe much more detailed contours in the final segmentation output (cf. Fig. 2 for raw CNN output and Fig. 4 for output with CRF post processing). The CRF post processing takes approximately 600 ms per image.

The baseline methods of [9] perform very favorable in terms of a classification-vs-runtime trade-off on CAPLOC, suggesting that hand-crafted image processing algorithms perform well in relatively controlled environments (i.e., good weather and clearly visible skyline). On the other hand, CNNs might improve performance when these conditions are not met, e.g., due to overexposure or poorly visible skyline. To underline this hypothesis, we show further challenging examples in which our CNN performs well *in the wild* in Fig. 4. Note that for a fair comparison with CNNs, we used the CAPLOC parameter set when applying baseline methods to these images. Performance gains can be expected when tuning them on a per-image or per-dataset basis.

VI. CONCLUSION

We have presented a camera-based NLOS detection system which, at the core, utilizes CNNs to obtain an image segmentation mask for satellite line-of-sight determination. As a result of the data-driven approach, performance of our system largely depends on the availability of high-quality datasets. In our experiments we find that CNNs potentially broaden the applications from relatively controlled environments regarded in previous work to more challenging scenarios. In future work, we will investigate the combination of complementary methods to further increase overall system performance.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4: Overview of segmentation results: (a)-(d): MobileNet with CRF. (e)-(h): Unsupervised baseline methods [9].

TABLE II: Comparison of sky segmentation accuray on CAPLOC [8] for MobileNet [17] and ResNet [16] models trained on different datasets. Additional comparison with unsupervised classifiers provided by [9] as baseline methods.

Architecture	Training Set	Mean IoU	Pixel Accuracy
MobileNet	ADE20K	0.9430	0.9797
MobileNet + CRF	ADE20K	0.9557	0.9843
MobileNet	ADE20K + SkyFinder	0.9618	0.9865
MobileNet + CRF	ADE20K + SkyFinder	0.9741	0.9909
ResNet 101	ADE20K	0.9518	0.9827
ResNet 101 + CRF	ADE20K	0.9655	0.9877
ResNet 101	ADE20K + SkyFinder	0.9464	0.9806
ResNet 101 + CRF	ADE20K + SkyFinder	0.9676	0.9885
Kmeans + Median filtering [9]		0.9676	0.9885
Otsu + Median filtering [9]		0.9683	0.9887
Fisher + Median filtering [9]		0.9685	0.9888

ACKNOWLEDGMENT

This work is funded by the EU H2020 GSA ERSAT GGC project no. 776039.

REFERENCES

[1] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[2] I. Guvenc and C.-C. Chong, "A survey on toa based wireless localization and nlos mitigation techniques," *Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, 2009.

[3] J. Riba and A. Urruela, "A non-line-of-sight mitigation technique based on ml-detection," in *IEEE ICASSP*, 2004, pp. ii–153.

[4] P. D. Groves and Z. Jiang, "Height aiding, c/n 0 weighting and consistency checking for gnss nlos and multipath mitigation in urban areas," *J NAVIGATION*, vol. 66, no. 5, pp. 653–669, 2013.

[5] G. Wang, H. Chen, Y. Li, and N. Ansari, "Nlos error mitigation for toa-based localization via convex relaxation," *Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4119–4131, 2014.

[6] S. Peyraud, D. Bétaille, S. Renault, M. Ortiz, F. Mougel, D. Meizel, and F. Peyret, "About non-line-of-sight satellite detection and exclusion in a 3d map-aided localization algorithm," *Sensors*, vol. 13, no. 1, pp. 829–847, 2013.

[7] D. Attia, C. Meurie, Y. Ruichek, and J. Marais, "Counting of satellites with direct gnss signals using fisheye camera : A comparison of clustering algorithms," in *IEEE ITSC*, 2011, pp. 7–12.

[8] J. Marais, C. Meurie, D. Attia, Y. Ruichek, and A. Flancquart, "Toward accurate localization in guided transport: Combining gnss data and imaging information," *TRANSPORT RES C-EMER*, vol. 43, no. 2, pp. 188–197, June 2014.

[9] J. Marais, S. Tay, A. Flancquart, and C. Meurie, "Weighting with the pre-knowledge of gnss signal state of reception in urban areas," in *European Navigation Conference*, 2015, p. 7.

[10] J. M. Huerta, J. Vidal, A. Giremus, and J.-Y. Tourneret, "Joint particle filter and UKF position tracking in severe non-line-of-sight situations," *J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 874–888, 2009.

[11] S. Yousefi, X.-W. Chang, and B. Champagne, "Mobile localization in non-line-of-sight using constrained square-root unscented kalman filter," *Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2071–2083, 2015.

[12] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *IEEE ICVS*, 2006, pp. 45–45.

[13] R. P. Mihail, S. Workman, Z. Bessinger, and N. Jacobs, "Sky segmentation in the wild: An empirical study," in *IEEE WACV*, 2016, pp. 1–6.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *PAMI*, vol. 40, no. 4, pp. 834–848, 2017.

[16] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *arXiv preprint arXiv:1802.02611*, 2018.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[18] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *IJCV*, pp. 1–20, 2016.

[19] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017, pp. 633–641.

[20] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, "Cnn based semantic segmentation for urban traffic scenes using fisheye camera," in *IEEE IV*, 2017, pp. 231–236.

[21] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011, pp. 109–117.

[22] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *ICLR*, 2018, pp. 1–8.