Automatic Generation of Large Point Cloud Training Datasets Using Label Transfer

TORBEN PETERS[1] & CLAUS BRENNER[1]

Abstract: In this work, we describe a framework for the automatic annotation of large scale point clouds. Our input data is generated using a mobile mapping system, which features LiDAR as well as camera image acquisition. In a first step, we labelled images of the measurement campaigns using pre-trained CNN (convolutional neural network) models for semantic segmentation. To that end, we ran PSPNet, which was trained on the cityscapes dataset. Then, since all camera poses are known from the mobile mapping GNSS/ IMU system, the image labels were transferred to the 3D points measured by the LiDAR scanners. Using this approach, we are able to automatically generate very large amounts of labelled point clouds, which can be used as training data. However, the dataset contains label noise, mainly because of calibration- and classification-errors, label policy, and occlusions which occur due to platform or object movements. We investigate different types of label noise and show how to recover the erroneous labelled 3d points. We do this by learning to identify wrong patterns in the label- aggregation and by calculating temporal features for every 3d-point by aligning large scale point clouds from different measurement campaigns into a voxel grid.

1 Introduction

Many state-of-the-art solutions to recognition, interpretation or planning problems are based on deep learning techniques. For example, in the domain of autonomous driving, deep learning is used for object segmentation and classification, motion planning, and even end-to-end learning. In classical supervised learning, networks are trained with data of a specific domain for the given task. However, if one domain intersects with another domain, the knowledge can be transferred between tasks. This procedure is called transfer learning. Autonomous cars often use different sensors in order to solve related tasks, for example cameras and LiDAR sensors, which makes the transfer of knowledge a viable approach. In this paper, we are presenting a framework for label transfer between images and 3d points. We are showing to what extend the quality of the labels is degrading by investigating different types of label noise. We do this by comparing the transferred labels with a manually annotated reference dataset. Furthermore, we are improving the label transfer by implementing a full ray tracing. Additionally, by storing aggregated labels in histograms, we are able to show that some types of noise are identifiable in the labels itself. Other types of noise are separable from the true classes by analyzing the temporal recurrence of a 3d-point. We do this by aligning point clouds from different measurement campaigns in a voxel grid. By determining the occupancy count of the voxels, we can calculate a feature that helps us to identify wrongly labeled points. In the end, we are investigating different approaches for label noise reduction and show that we are able to recover the label noise by manually labeling only a small amount of data.

[1] Leibniz University Hannover, Institute of Cartography and Geoinformatics

2 Related work

Semantic segmentation attempts to segment and classify parts of a scene by doing pixel- or pointwise classification. Since the rise of deep learning, popular approaches are using fully convolutional neural networks for the semantic segmentation of images (LONG et al. 2015). More advanced approaches are still relying on neural networks like PSPNet (ZHAO et al. 2017) or Deeplab (CHEN et al. 2018) but improved the performance by modifying their architecture and components. In order to compare different approaches many benchmarks are available. The so called cityscapes benchmark for semantic segmentation was released by CORDTS et al. (2016). They include annotated data from 50 different german cities and annotated 25000 images with up to 30 classes. In our work we are trying to use this information in order to enrich our own dataset.
With **Transferring labels** we refer to the general procedure of mapping label information from one domain (e.g. 2d) into another (e.g. 3d). The transfer between 3d- and 2d-space has been done before, XIE et al. (2016) transferred human annotated point clouds into images in order to create arbitrary amounts of training images and corresponding labels. Transferring labels from 2d images to 3d point clouds is also not completely new. Notable contributions are made by MCCORMAC et al. (2016) and HERMANS et al. (2014). Both are considering static scenes and transfer labels by labeling RGB-D images wich are then mapped into 3d-scenes. Another way of labeling 3d point clouds from images was proposed by BOULCH et al. (2017). They are projecting the point cloud into images which are then classified by convolutional neural networks. In order to classify the point cloud the classified image pixel are remapped from 2d into 3d. Recently in 2018, ZHANG et al. used semantic segmented images in order to project them into point clouds. However, in comparison to our work they used stationary terrestrial laser scanners, which do not exhibit the same amount of label noise as is the case with a mobile mapping system or laser scanners of autonomous cars. The handling of dynamic objects in stereo images was investigated by KOCHANOV et al. (2016). They are using scene flow in order to propagate the semantics of dynamic objects into 3d maps.
Label noise describes the presence of wrong labels in a data set. There are many different types of noise described in literature (NETTLETON et al 2010; DRORY et al. 2018). One type of noise are randomly flipped labels. Some studies are showing that deep neural networks are robust to this type of noise as long as the noise level does not exceeds a certain level (DRORY et al. 2018; ROLNICK et al. 2018). Another type is the flip label-noise. In this case a each label is confused to a certain degree with a different class.
Label noise cleaning deals in general with correcting the label noise in a data set. Today many state of the art procedures are based on deep neural networks (XIAO et al. 2015; LEE et al. 2017; PATRINI et al. 2016; HENDRYCKS et al. 2018). While Lee et al. tried to correct the label noise by demoting or removing wrong instances XIAO et al. (2015) corrected the labels directly. Both rely on a small “clean” dataset that helps them to estimate the noise in order to clean the corrupted labels. Patrini et al. were able to correct the label noise in an unsupervised way by estimating a transition matrix which describes the probability that a class is flipped to another. Hendrycks et al. improved this approach by incorporating supervised information. However the topic of label noise handling has been extensively studied as shown by the survey of FRENAY et al. (2014). Additionally we would like to note that it is not uncommon to deal with label noise cleaning in the

scenario that is presented in this work. Some of the previous mentioned authors used conditional random fields (CRF) in order to improve the quality of the mapped labels in 2d and 3d (Xie et al 2016; McCormac et al 2016; Hermans et al 2014).

3 Dataset

Our dataset was created within fourteen different measurement campaigns in the time between March and October of 2017. We used a Riegl-VMX 250 mobile mapping system mounted on a van and equipped with two full-circle laserscanners, two cameras and an inertial measurement/ global navigation satellite system (IMU/GNSS) unit. Each measurement campaign covered the same route and was done at least once a month in an urban environment, which resulted in ~1 billion points and ~10.000 images per campaign.

We first automatically annotated the (RGB camera) images taken during our measurement campaigns using pre-trained convolutional neural network (CNN) models for semantic segmentation. To that end, we ran PSPNet (Zhao et al.) which was trained on the *cityscapes* dataset (Cordts et al. 2016)). Then, using the known pose of all cameras, we transferred all labels to the 3d points. The dataset constructed in this way contains about 15 billion labelled points.

3.1 Methodology

Figure 1 shows an overview of our methodology. As mentioned before, our first step is to use PSPnet by Zhao et al. to classify each pixel in every image taken by our mobile mapping van. According to the results reported in Zhao et al., PSPNet reached a mean intersection over union (IoU) of 81.2 % for the cityscapes dataset. For each image we calculated the pose of the camera in UTM coordinates and the view direction vector d. Furthermore, the intrinsic parameters of each camera were available. By projecting the 3d points into all semantically segmented images, their labels can be picked and we are able to map each label to a 3d-point.

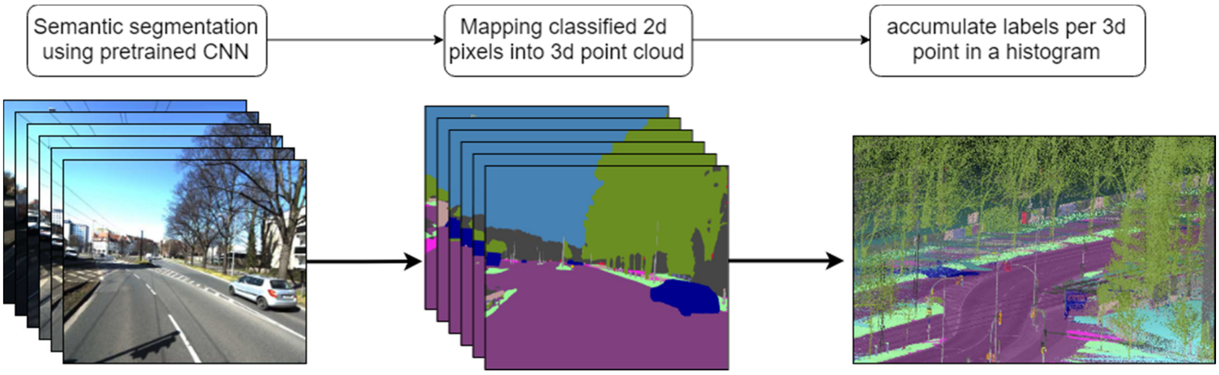

Semantic segmentation using pretrained CNN
Mapping classified 2d pixels into 3d point cloud
accumulate labels per 3d point in a histogram

Fig. 1: Overview methodology (f.l.t.r): camera images, semantic segmented images, point cloud containing transferred labels colored by majority label.

Because a 3d-point may be projected into multiple images, it is also possible that different labels are associated with this point. In order to accumulate these labels, we define one histogram h_i per 3d-point p_i ,with $i \in \{1, ..., n\}$ where n is the number of measured points. Each histogram has 20 bins, and if a class $c = j \in \{1, ..., 20\}$ is observed, we are incrementing the bin j by one. The resulting histogram can contain contradictory information, for example if a car drives through the scene while the images are taken, we are accumulating car and street labels. The following picture shows an example of the transferred labels. We adhere to the color conventions used in the cityscapes project in order to make our results easier comparable.

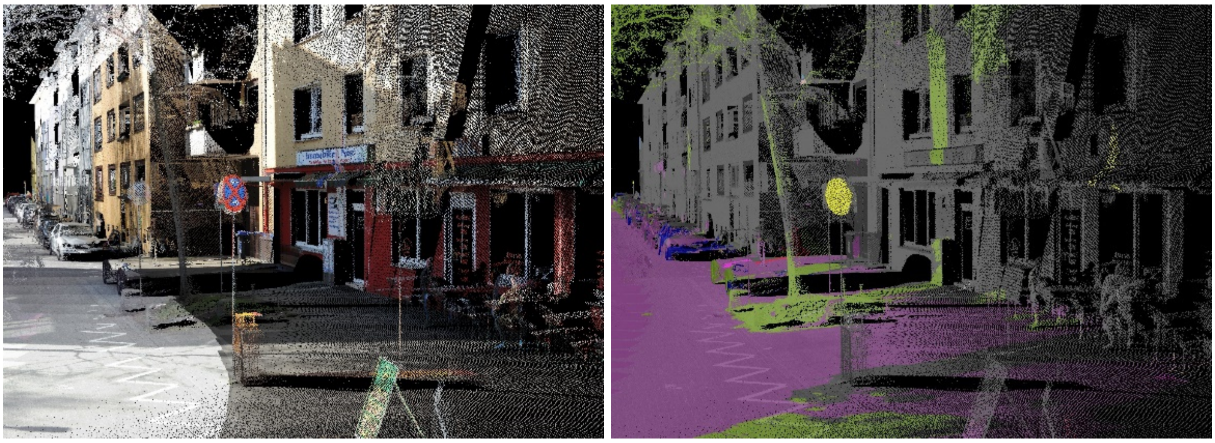


Fig. 2: (a) Point cloud colored by RGB images, (b) Point cloud colored by transferred majority label

3.2 Label noise

By *label noise*, we term the effect that a certain amount of points will have non-correct labels. Although we do not know the exact process that leads to these label flips, we can identify several reasons. Examples are calibration errors or poor prediction quality of PSPNet due to different sensor types or different settings from the original data. Besides those errors we identified several systematic causes for label noise in our dataset. This is noise due to

- the label policy of cityscapes,
- occlusion in 3d-point clouds, and
- the difference in capture time between the laserscanner and camera sensors.

In the following sections we are describing each case in detail.

3.2.1 Noise due to label policy

As shown in figure 3(a) the image label policy of cityscapes requires that surfaces which are visible behind tree crowns, such as building facades, have to be assigned the tree label. However, many of the laser rays will go through the tree crown and thus the points on the facade will be labelled as tree. An example for this kind of label noise is shown in figure 3(b). In the depicted scene, the tree labels are erroneously projected onto the building behind.

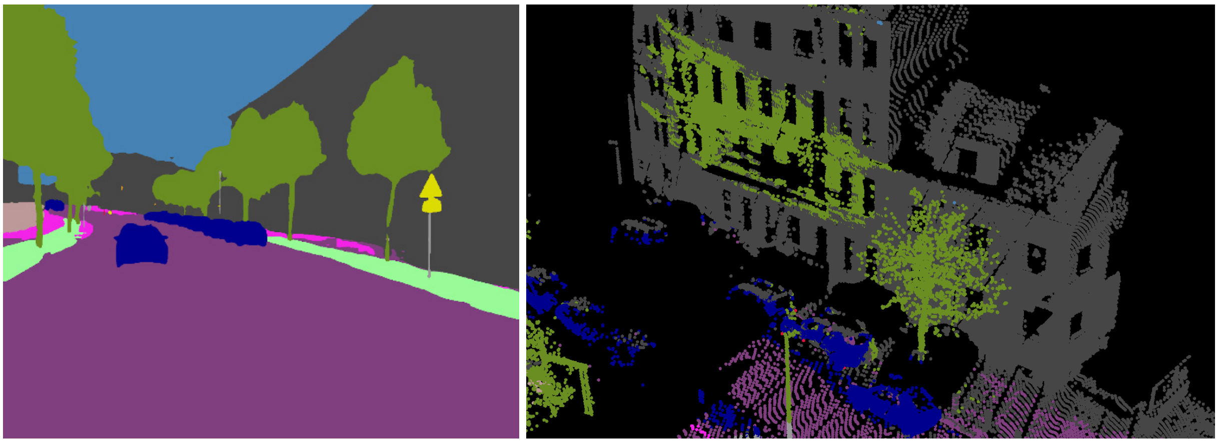


Fig. 3: (a) Semantic segmented image (b) Point cloud colored by transferred majority label

One can see this effect also in figure 2 (b). By storing the aggregated labels in a histogram we are also accumulating labels from the object in the front.

This information can be used in order to segregate the noisy histograms and thus counteract the noise induced by the label policy. We can verify this by clustering the histograms using k-means clustering with $k = 32$, shown in figure 4. In this figure, each histogram is assigned to the nearest centroid. It is colorized according to the cluster number, therefore histograms with the same color belong to the same cluster.

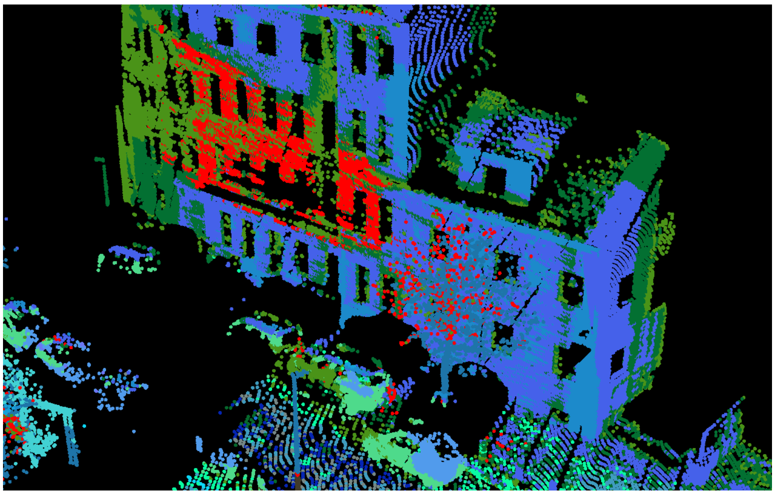


Fig. 4: Histograms clustered using k-means clustering. Classes are randomly colored. Red class on the building visualizes histograms which contains tree and building labels.

By looking at the bright red clusters on the facades it becomes apparent that this kind of noise is separable from pure tree and background classes. To that end, we can directly use the histograms h_i as features in order to learn if the histogram is affected by the noise and therefore the class derived from the peak of the histogram is probably wrong.

3.2.2 Noise due to occlusion

In order to transfer labels from the images to the scan strips, the strips are processed one after the other, using all images that were taken along this strip. Since all single laser points and all captured images are time stamped, the complete geometry can be recovered using the IMU/GNSS data, so that every laser point can be projected to every image, using exterior and interior orientation, as well as lens distortion terms.

However, since the laser and image rays do not coincide, occlusions are quite frequent, which results in 3d points being assigned the label of an occluding object rather than the correct object, which will introduce label noise. This effect is mitigated to a certain extend by the fact that we are accumulating all transferred labels in a histogram, as described above. Therefore, if the point is unoccluded for most of the time, the histogram will still peak at the correct label.

In order to improve this further, we have implemented a full ray tracing for the label transfer. All points of a scan strip are sorted into a voxel grid. Then, when determining the label of a 3d-point, the ray to each camera center is traced in this grid and the point is considered to be occluded if an occupied cell is found along the ray. We used 10*cm* grid cells for this operation. In addition, a voxel pyramid was computed to speed up the computation. Although this process is only a crude approximation of the complex interactions between laser and image rays, it generally improves the label transfer.

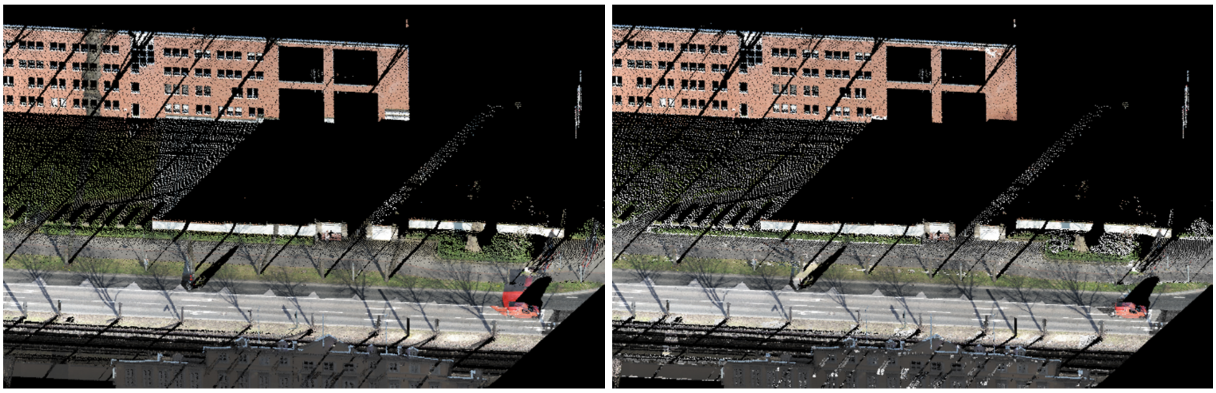


Fig. 5: (a) point cloud colored without raytracing (b) point cloud colored with raytracing

An example is shown in figure 5. Here, the colorization with ray tracing prevents the building in the background from being colored with tree (stem) pixels. The red car on the street is also less scattered to the ground if ray tracing is used. The colorization shown here has direct impact on the label transfer because we are aggregating class labels by mapping classified image pixels to 3d points using the very same mechanism.

3.2.3 Noise due difference in capture time

Further label transfer errors occur if the recorded object is not static. This ensues from the fact that our cameras are facing backwards while the laserscanners are vertically inclined. We tried to create a feature that gives us a measure of how dynamic a captured 3d-point is. We aligned all points between different measurement campaigns using the strip adjustment approach described by Brenner (2016). By creating a voxelgrid using an edge length of 5 *cm*, we grouped different points and histograms in 3d-space. We counted the number $v_i \in \{1,\dots,14\}$ of how often a voxel was measured in distinct measurement campaigns. This value is assigned to each point that is present in the corresponding voxel. After normalization, a score of $v_i = 0$ means that p_i is highly dynamic and $v_i = 1$ means that p_i is highly static.

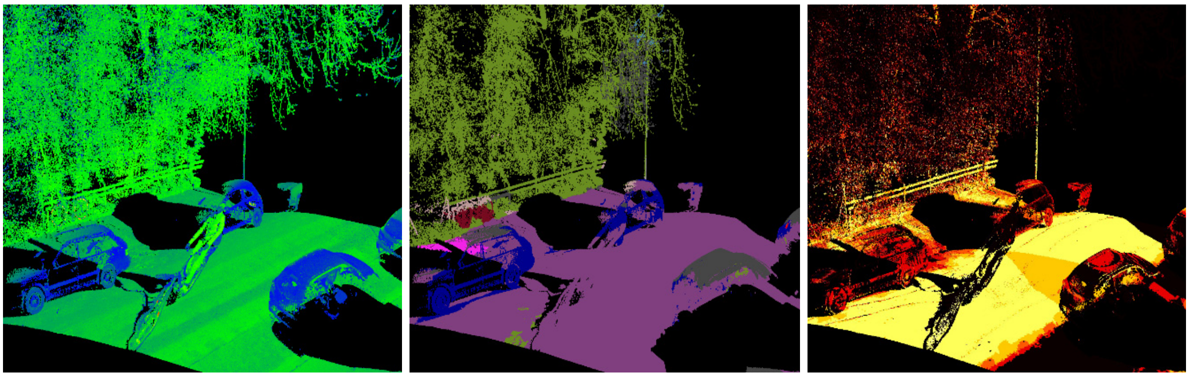

(a) (b) (c)

Fig. 6: Point cloud colored by (a) intensity, (b) class label, and (c) voxel count v_i black color means measured once or never, bright color means measured 14 times.

Figure 6 shows a scan strip with a cyclist on the street and two parking cars. Because of the difference in capture time the cyclist is labelled as street in figure 6(b). By counting the number of occurrences v_i, we can create a feature that can be used to identify dynamic objects. This feature is visualized in figure 6(c), where we can see, that the cyclist who introduced label noise can easily be removed. Furthermore this feature preserves the parking cars in the background because they were measured multiple times.

4 Evaluation of the Quality of the Dataset

In order to measure the quality of our dataset, we labeled a subset of 75283713 3d points manually. However, due to unbalanced occurrence and the different physical size of the classes the ground truth is highly unbalanced.

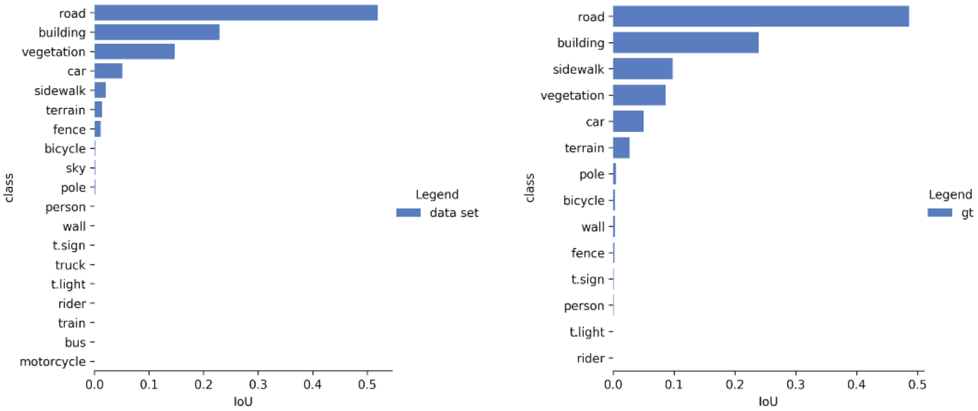


Fig. 7: (a) distribution of classes in the dataset (b) distribution of classes in the ground truth

Figure 7(a) shows the relative frequency of each class in the dataset. It covers all classes that are included in the evaluation of cityscapes. In the ground truth (figure 7(b)) the classes sky, motorcycle, truck, train and bus are missing. It is obvious that we cannot label any points as sky, because this class is not measurable with a laserscanner, therefore it must be label noise and can be excluded from the dataset. As can be seen in figure 7(a), the other four classes are extremely rare, so we didn't find enough (or any) examples and had to exclude these classes from the evaluation.

We measured the quality of the dataset by evaluating the IoU which is calculated from the well-known true/ false positives/ negatives TP, FP, FN as follows:

$$(1) \quad IoU = \frac{TP}{TP+FP+FN}$$

In order to extract a class c from a histogram h_i we used the majority class $c = argmax(h_i)$. The IoU is first calculated per category, which gives us an idea of how well each category intersects with the ground truth. We estimated the overall label noise by calculating the mean IoU. The results are shown in the following figures. In order to compare them to the performance of the pretrained network, we included the IoU results of PSPnet from cityscapes.

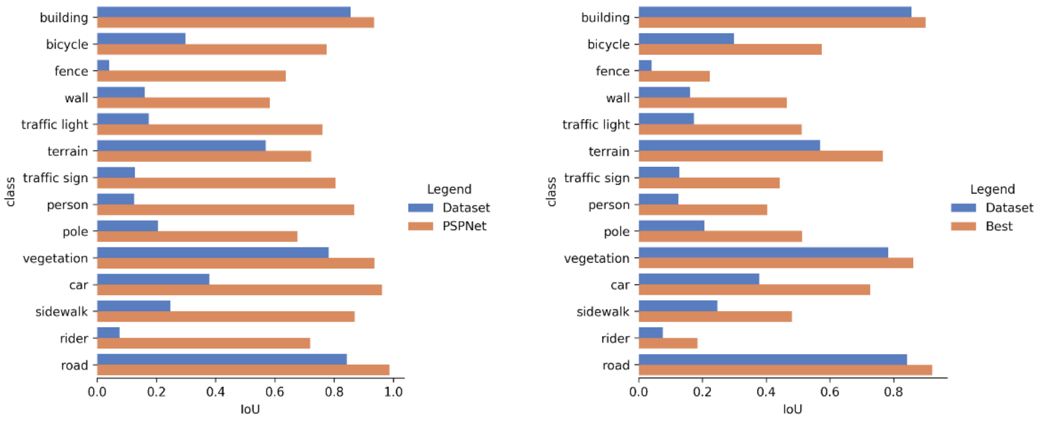
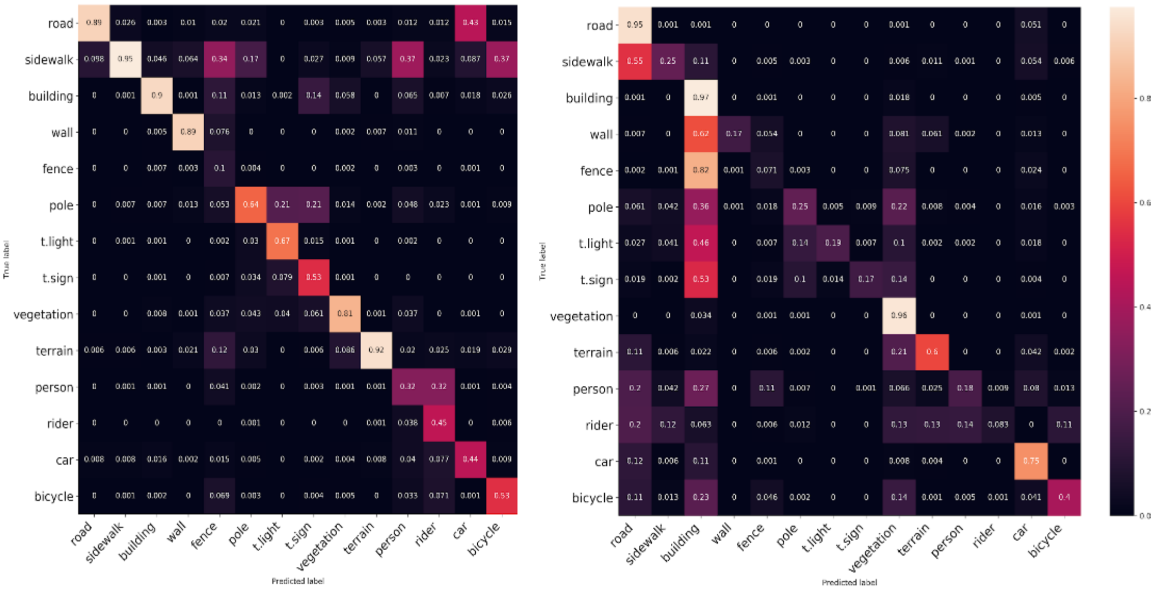


Fig. 8: (a) IoU with majority label

(b) IoU if at least one point in the histogram intersects with the ground truth (best)

Figure 8(a) shows the estimated IoU per category for our dataset compared to the IoU of PSPNet. The mean IoU in our dataset is ~35% and the mean IoU of PSPnet is 81.2%. In figure 8(b) we estimated the "best" case in which at least one value intersects with the ground truth. In contrast to the majority label, a true positive is counted as valid if the histogram contains a value greater zero wherever it intersects with the ground truth class. The mean IoU rises in the "best" case to about ~57%. We would like to note that in this case the IoU for terrain rises above the one of PSPNet. It could be possible that the aggregation through different pictures leads to a denoising effect. In the next step we calculated a confusion matrix for the majority labels.



True label \ Predicted label	road	sidewalk	building	wall	fence	pole	t.light	t.sign	vegetation	terrain	person	rider	car	bicycle
road	0.89	0.026	0.003	0.01	0.02	0.021	0	0.003	0.005	0.003	0.012	0.012	0.43	0.015
sidewalk	0.098	0.95	0.046	0.064	0.34	0.17	0	0.027	0.009	0.057	0.37	0.023	0.087	0.37
building	0	0.001	0.9	0.001	0.11	0.013	0.002	0.14	0.058	0	0.065	0.007	0.018	0.026
wall	0	0	0.005	0.89	0.076	0	0	0	0.002	0.007	0.011	0	0	0
fence	0	0	0.007	0.003	0.1	0.004	0	0	0.002	0	0.003	0	0.001	0
pole	0	0.007	0.007	0.013	0.053	0.64	0.21	0.21	0.014	0.002	0.048	0.023	0.001	0.009
t.light	0	0.001	0.001	0	0.002	0.03	0.67	0.015	0.001	0	0.002	0	0	0
t.sign	0	0	0.001	0	0.007	0.034	0.079	0.53	0.001	0	0	0	0	0
vegetation	0	0	0.008	0.001	0.037	0.043	0.04	0.061	0.81	0.001	0.037	0	0.001	0
terrain	0.006	0.006	0.003	0.021	0.12	0.03	0	0.006	0.086	0.92	0.02	0.025	0.019	0.029
person	0	0.001	0.001	0	0.041	0.002	0	0.003	0.001	0.001	0.32	0.32	0.001	0.004
rider	0	0	0	0	0	0.001	0	0	0	0.001	0.038	0.45	0	0.006
car	0.008	0.008	0.016	0.002	0.015	0.005	0	0.002	0.004	0.008	0.04	0.077	0.44	0.009
bicycle	0	0.001	0.002	0	0.069	0.003	0	0.004	0.005	0	0.033	0.071	0.001	0.53

True label \ Predicted label	road	sidewalk	building	wall	fence	pole	t.light	t.sign	vegetation	terrain	person	rider	car	bicycle
road	0.95	0.001	0.001	0	0	0	0	0	0.001	0	0	0	0.051	0
sidewalk	0.55	0.25	0.11	0	0.005	0.003	0	0	0.006	0.011	0.001	0	0.054	0.006
building	0.001	0	0.97	0	0.001	0	0	0	0.018	0	0	0	0.005	0
wall	0.007	0	0.62	0.17	0.054	0	0	0	0.081	0.061	0.002	0	0.013	0
fence	0.002	0.001	0.82	0.001	0.071	0.003	0	0	0.075	0	0	0	0.024	0
pole	0.061	0.042	0.36	0.001	0.018	0.25	0.005	0.009	0.22	0.008	0.004	0	0.016	0.003
t.light	0.027	0.041	0.46	0	0.007	0.14	0.19	0.007	0.1	0.002	0.002	0	0.018	0
t.sign	0.019	0.002	0.53	0	0.019	0.1	0.014	0.17	0.14	0	0	0	0.004	0
vegetation	0	0	0.034	0	0.001	0.001	0	0	0.96	0	0	0	0.001	0
terrain	0.11	0.006	0.022	0	0.006	0.002	0	0	0.21	0.6	0	0	0.042	0.002
person	0.2	0.042	0.27	0	0.11	0.007	0	0.001	0.066	0.025	0.18	0.009	0.08	0.013
rider	0.2	0.12	0.063	0	0.006	0.012	0	0	0.13	0.13	0.14	0.083	0	0.11
car	0.12	0.006	0.11	0	0.001	0	0	0	0.008	0.004	0	0	0.75	0
bicycle	0.11	0.013	0.23	0	0.046	0.002	0	0	0.14	0.001	0.005	0.001	0.041	0.4

Fig. 9: (a) Confusion matrix normalized per column, (b) Confusion matrix normalized per row

The table in Figure 9(a) is normalized per column therefore the diagonal axis contains the precision of the dataset. Figure 9(b) is normalized per row, thus the diagonal axis contains the recall of the dataset per class. As can be seen in Figure 9(a), moving objects are often confused with their surroundings. We believe that this type of noise is due to the difference in capture time. For example the prediction of the class car is to 44% true but in 43% of the cases it is road. Also the classes person and bicycle are often confused with sidewalk. Except fence, most static object are predicted with a high precision. As shown in Figure 9(b), the recall differs in some cases. In many cases bigger objects are having a high recall for example road, buildings, vegetation, terrain and cars. Smaller objects like poles, traffic lights, persons and bicycles are lower in recall, which makes sense because it is harder to hit the right point in 3d if the object is small or moving. Additionally, classes with a low IoU like wall, fence and rider are generally having a bad recall.

5 Label noise cleaning

In order to rise the IoU of the dataset we are presenting different approaches. One way of dealing with label noise is to remove possibly wrong labels. One potential drawback is that this approach induces sparsity into the dataset which could lead to worse outcomes in a supervised learning task because spatial information are getting lost.

5.1 Filtering based on entropy

A possible measure for the uncertainty of labels is the entropy of the label histogram. A high entropy means that the histogram contains strong votes for different labels which could be an indicator for the presence of label noise. Histograms are removed if the entropy is above a certain threshold. The best IoU was calculated for zero entropy which means that there is only one peak in the histogram. Therefore we are removing histograms if the entropy is not equal to zero. As shown in the following figure it worked well for some classes.

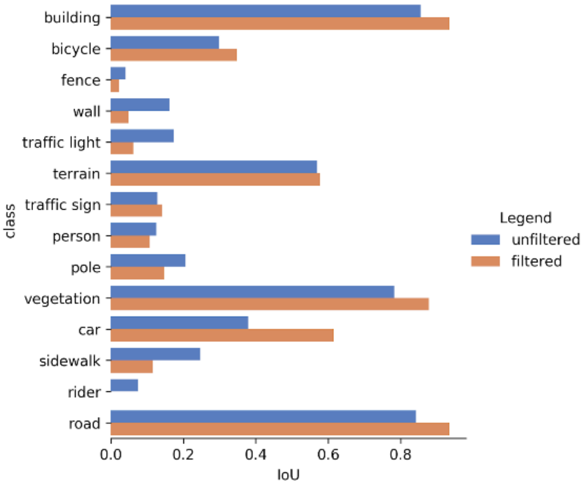


Fig. 10: Comparison of the IoU after the labels were filtered by an entropy threshold

The orange bars are showing that the IoU raised in some cases. The most significant difference was measured for the class cars. This class went from 37.9% to 61.5% IoU. However, many classes that were problematic in the first case dropped in IoU after the dataset was filtered. Furthermore by removing these histograms 27.5 % of the data in the test set was removed. The mean IoU raised from 34.9% to 35.2%. By using this filter only on classes that did raise in the IoU we are able to get an estimated mean IoU of 38.98%.

5.2 Filtering labels by learning the noise

Another way of estimating the noise is by learning it in a supervised manner. In order to separate the noisy labels from the good ones we introduced two classes $d = \{0,1\}$

$$d = \begin{cases} 1, \text{if } \mathrm{l_i} = argmax(h_i) \\ 0, \text{if } \mathrm{l_i} \neq argmax(h_i) \end{cases} \tag{2}$$

A histogram is labelled as 1 if the ground truth l_i is equal to the majority label in the histogram. Otherwise it is marked as zero and thus removed from the dataset. In order to learn the noise in the dataset we used a gradient boosting decision tree (GBDT) algorithm, LightGBM by Guolin, et al.. We separated 15% of ground truth as test set by picking the samples randomly. The hyperparameter tuning was done on the training set by using 3-fold cross validation. We optimized the parameter with random search by maximizing the area under the receiver operating characteristic curve (ROC AUC). The ROC AUC also known as c-statistic is a measure of goodness of fit for binary outcomes which is not sensitive to unbalanced classes. As shown in the following figure, we were able to improve the IoU for nearly every class.

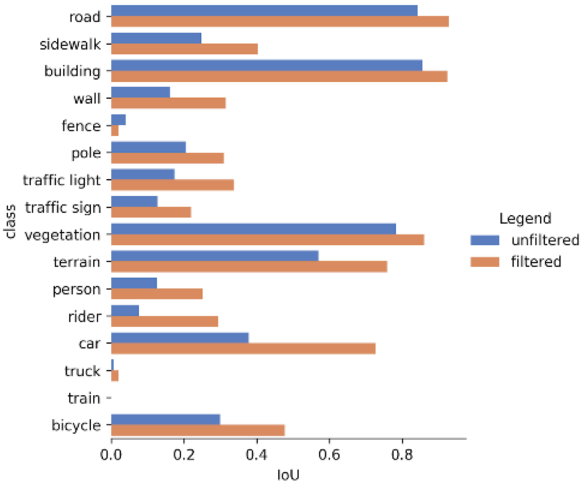


Fig. 11: Comparison of the IoU after the labels were filtered by GBDT

We would like to note that the outcome for the training and test set were nearly identical - the mean IoU differs by only 0.2 %. By using the classifier we can raise the IoU to 48.77 %. In order to increase the IoU even further, more information is needed. We chose to add the voxel count v_i in order to distinguish between dynamic and static objects. We believe that the information of dynamics paired with a typical histogram can help to remove noisy labels. The training procedure is done in same manner as before. The following images show the difference between both classifiers.

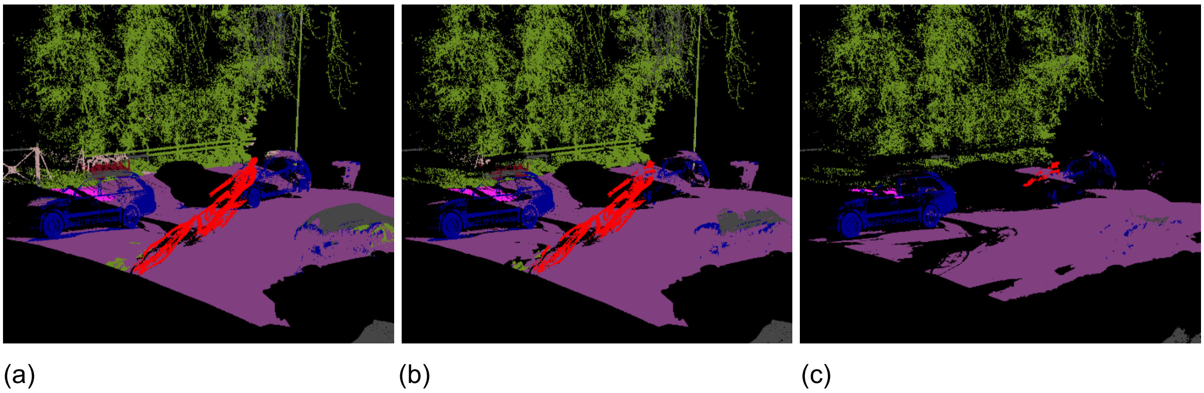


Fig. 12: Comparison of different label noise removal techniques. (a) Shows the depicted scene from Figure 3 with no changes, we marked the bicycle driver as red. (b) Shows the noise removal using only the histogram as feature. In (c) we additionally used the voxel count v_i as feature

We used the scene from figure 6 to show the effect of using the voxel count v_i. A perfect classifier would remove the red marked cyclist completely. As shown in figure 12 (b), the classifier is able to remove some noisy labels but it fails to delete the cyclist when only the histogram information is available. As can be seen in figure 12 (c), we were able to remove almost all of the red points by incorporating the voxel count v_i. Still the classifier is able to preserve the correct classified (blue) cars. Furthermore, we were able to raise the mean IoU to about 51.22%. The following figure shows how the IoU improved per class by using GBDT with voxel count against the filter based on the entropy.

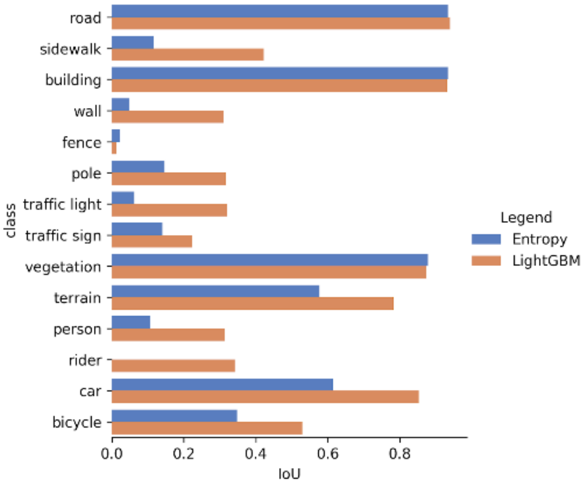


Fig. 13: IoU per category after removing the faulty labels. We compared the cleaning by using the entropy (blue) vs. by learning the noise (orange).

5.3 Correction labels by learning the noise

Lastly we tried to "flip" the labels to the correct class instead of removing faulty histograms. We did this in the same manner as before by using GBDT. The hyperparameter tuning was done by maximizing the weighted F1-Score. Additionally to the histogram we included the voxel count as feature. By flipping the labels we were able to increase the mean IoU to about 63.79% which is only 18% worse than PSPNet performed on the testset in cityscapes. Interestingly, it is also better than the "best case" we presented in figure 8(b), which means that we can recover labels that are not even present in the histogram. In the following picture we are showing the qualitative improvement over the original label transfer.

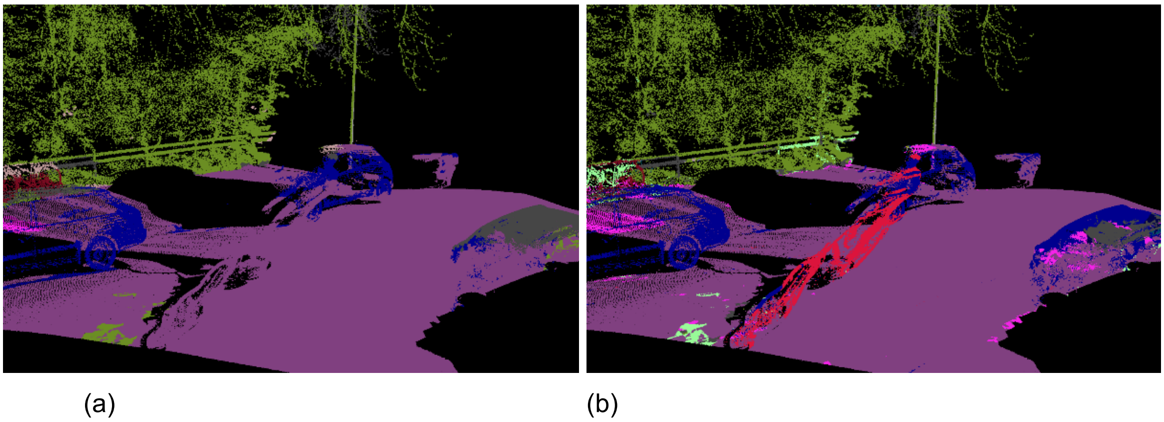


Fig. 14: (b) shows that we were able to recover information that would otherwise be lost if we used the majority label (Fig. 14 (a)). As can be seen, the cyclist and some other points were flipped to the correct class.

5.4 Results

Lastly we are showing a table that compares all results we presented in this work.

Tab 1: Comparison of different noise reductions to PSPNet. The column „Original“ contains the unchanged 3d data set. The next column „Entropy“ shows the IoU if histograms without zero entropy are removed. „Removed“ and „Flipped“ are the learned approaches with GBDT including the voxel count.

IoU	*Original*	*Entropy*	*Removed*	*Flipped*	*PSPNet*
road	84.3	93.41	94.01	91.96	98.68
sidewalk	24.71	11.64	42.25	55.35	86.92
building	85.59	93.45	93.08	91.85	93.46
wall	16.15	4.89	31.13	52.5	98.68
fence	4.09	2.3	1.32	23.4	63.67
pole	20.6	14.65	31.62	45.15	67.67
traffic light	17.37	6.22	32.0	55.82	76.12
traffic sign	12.8	14.1	22.25	50.53	80.47
vegetation	78.23	87.85	87.4	86.51	93.63
terrain	56.99	57.73	78.22	75.34	72.2
person	12.51	10.74	31.47	59.56	69.32
rider	7.55	0.0	34.18	75.41	69.32
car	37.88	61.45	85.25	76.16	90.27
bicycle	29.91	34.79	52.91	53.48	63.5
Avg. IoU	**34.91**	**35.23**	**51.22**	**63.79**	**81.19**

We have shown that we were able to greatly improve the average IoU by learning the noise in the data set. The class “building” even reached nearly the original IoU. Some classes are better when the histograms are removed instead of being flipped. We could therefore imagine that a classifier that decides whether to remove the label or flip it could even improve the IoU further by taking the best of both worlds.

6 Conclusion

We have presented a framework for label-transfer from the image domain to point clouds. After identifying origins of label noise, we created new features in order to improve the IoU of our dataset. Our Framework has the potential to create very large amounts of labelled point cloud data. This data can be used to train deep neural networks in order to learn semantic segmentation in 3d. For future work, we would like to improve the IoU by using more advanced techniques for noise detection. We can imagine that by using a spatial neighborhood, the label cleaning can be improved. Additionally it could be helpful to adapt the pre trained neural network to our cameras by using adversarial discriminative domain adaptation (adda) (Tzeng et al. 2017). The improved classification error would minimize the label noise in an early stage of our framework.

7 Acknowledgements

This work was funded by the German Research Foundation (DFG) as a part of the Research Training Group GRK2159, `Integrity and collaboration in dynamic sensor networks' (i.c.sens).

8 References

BOULCH, A., GUERRY, J., LE SAUX, B. & AUDEBERT, N., 2018: SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. Computers & Graphics, **71**, 189-198.

BRENNER, C., 2016: Scalable estimation of precision maps in a MapReduce framework. In International Conference on Advances in Geographic Information Systems, 27.

CHEN, L.C., PAPANDREOU, G., KOKKINOS, I., MURPHY, K. & YUILLE, A.L., 2018: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence, **40**(4), 834-848.

CORDTS, M., OMRAN, M., RAMOS, S., REHFELD, T., ENZWEILER, M., BENENSON, R., FRANKE, U., ROTH, S. & SCHIELE, B., 2016: The cityscapes dataset for semantic urban scene understanding. IEEE Conference on Computer Vision and Pattern Recognition, 3213-3223.

DRORY, A., AVIDAN, S. & GIRYES, R., 2018: On the Resistance of Neural Nets to Label Noise. arXiv preprint arXiv:1803.11410.

FRÉNAY, B. & VERLEYSEN, M., 2014: Classification in the presence of label noise: a survey. IEEE Transactions on Neural Networks and Learning Systems, **25**(5), 845-869.

HENDRYCKS, D., MAZEIKA, M., WILSON, D. & GIMPEL, K., 2018: Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise. arXiv preprint arXiv:1802.05300.

HERMANS, A., FLOROS, G. & LEIBE, B., 2014: Dense 3d semantic mapping of indoor scenes from rgb-d images. IEEE International Conference on Robotics and Automation, 2631-2638.

KE, G., MENG, Q., FINLEY, T., WANG, T., CHEN, W., MA, W., QIWEI YE, Q. & LIU, T. Y., 2017: Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 3146-3154.

KOCHANOV, D., OŠEP, A., STÜCKLER, J. & LEIBE, B., 2016: Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. IEEE/RSJ International Conference on Intelligent Robots and Systems, 1785-1792.

LEE, K.H., HE, X., ZHANG, L. & YANG, L., 2017: CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise. arXiv preprint arXiv:1711.07131.

LONG, J., SHELHAMER, E. & DARRELL, T., 2015: Fully convolutional networks for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, 3431-3440.

MCCORMAC, J., HANDA, A., DAVISON, A. & LEUTENEGGER, S., 2017: Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. IEEE International Conference on Robotics and Automation, 4628-4635.

NETTLETON, D.F., ORRIOLS-PUIG, A. & FORNELLS, A., 2010: A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial intelligence review, **33**(4), 275-306.

PATRINI, G., ROZZA, A., MENON, A.K., NOCK, R. & QU, L., 2017: Making deep neural networks robust to label noise: A loss correction approach. IEEE Conference on Computer Vision and Pattern Recognition, 2233-2241.

ROLNICK, D., VEIT, A., BELONGIE, S. & SHAVIT, N., 2017: Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694.

TZENG, E., HOFFMAN, J., SAENKO, K. & DARRELL, T., 2017: Adversarial discriminative domain adaptation. Computer Vision and Pattern Recognition, **1**(2), 4.

XIAO, T., XIA, T., YANG, Y., HUANG, C. & WANG, X., 2015: Learning from massive noisy labeled data for image classification. IEEE Conference on Computer Vision and Pattern Recognition, 2691-2699.

XIE, J., KIEFEL, M., SUN, M.T. & GEIGER, A., 2016: Semantic instance annotation of street scenes by 3d to 2d label transfer. IEEE Conference on Computer Vision and Pattern Recognition, 3688-3697.

ZHANG, R., LI, G., LI, M. & WANG, L., 2018. Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning. ISPRS Journal of Photogrammetry and Remote Sensing, **143**, 85-96.

ZHAO, H., SHI, J., QI, X., WANG, X. & JIA, J., 2017: Pyramid scene parsing network. IEEE Conference on Computer Vision and Pattern Recognition, 2881-2890.