The Influence of Noise on Image Semantic Segmentation

Pengchao Wei

Computer Science
E-mail: pengchao.wei@campus.tu-berlin.de
Student ID: 406089

Bing Li

Geodesy and Geoinformation Science
E-mail: bing.li@campus.tu-berlin.de
Student ID: 404864

Abstract

Noise is ubiquity in this world. Denoising is an essential technology especially in image semantic segmentation, where noises are generally categorized into two main types i.e. feature noise and label noise. The main focus of this paper is aiming at modeling label noise, investigating the behaviors of different types of label noise on image semantic segmentation task using different classifiers. The performance without label noise and with is evaluated and illustrated in this paper. Addition to that, the influence of feature noise on the image semantic segmentation task is researched as well and a feature noise reduction method is applied to mitigate its influence in learning procedure.

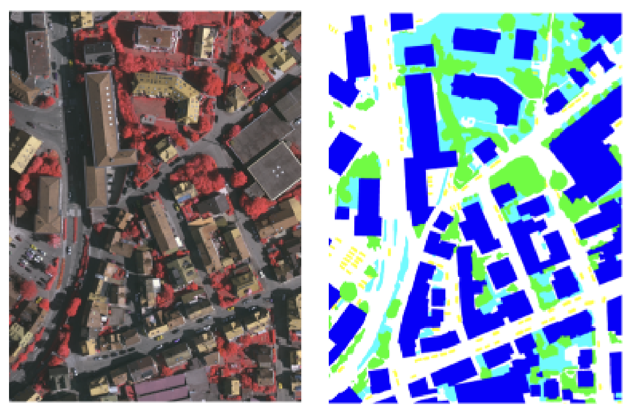
Keywords: Label noise, feature noise, image semantic segmentation

1. Introduction

Image semantic segmentation is one of the most commonly researched areas in computer vision. In that context, the most existing algorithms are fully supervised namely a mapping is supposed to be learned from input feature and reference data in order to assign a semantic label from predefined categories to each pixel in test set i.e. pixel-wise classification.

In supervised learning, there may exist two different types of noise: feature (attribute) noise and label noise (class noise) [7][9]. A large amount of approaches are proposed to cope with feature noise in automatic image analysis. However, the label noise in reference data is seldom investigated and customarily assumed to be noise free.

Since the reference data is involved in learning the mapping, label noise definitely have a significant influence on supervised learning task. Especially label noise has great impact on image semantic segmentation. One reason is that it is easy to be wrong to assign a semantic label to each pixel in the image annotated manually by experts or volunteers e.g. the pixels located at the boundary of different objects within the image or the contour of the objects. Besides, manually annotating each pixel in the image is a tedious work and quite time consuming. For deep learning methods, where large datasets is essential for training to prevent from overfitting, it seems infeasible to manually annotate each image in datasets. Even though there exist some crowd-sourcing platforms for semantic map annotation, the label noise in reference data is still non-negligible [9].



(a) Image scene (b) Reference map

Figure 1: An example image scene from Vaihingen dataset [14] and its corresponding color-coded reference map, where class road, buildings, low vegetation, trees, cars and clutter are coded by white, blue, cyan, green, yellow, and red, respectively. As an example, the pixels located at the boundary of building and road may be mislabelled to each other with high possibility.

In this paper, we investigate the influence of different types of noise on image semantic segmentation with a focus on label noise. For feature noise, a gaussian additive term is implemented to see how the performance changes with feature noise and after applying feature noise reduction technique. For label noise, three different types are considered i.e. spatially correlated, semantically correlated and radiometrically (feature) correlated. The influence of different noises on image semantic segmentation task is evaluated on two different algorithms and their robustness to label noise is also explored and illustrated. K-Nearest-Neighbor classifier is considered as the simple baseline model for comparison due to the fact that KNN is sensitive to label noise [7]. For more sophisticated approach, a Fully Convolutional Network (FCN) classifier, which is to some extend has robustness to label noise [7], is implemented.

2. Related Work

There exists many works researching the impact of different types of noise on different machine learning algorithms.

For feature noise, [24] has conducted experiments regarding to feature noise and systematically investigated the consequences of feature noise on different learning algorithms as well as concluded different feature noise reduction guidelines for readers to enhance data quality from their own perspective. [15] introduces sample noise term and feature noise term (here feature noise specifically refers to injecting feature noise to the existing dataset) as the generalized noise model. Among different feature noises, the most commonly occurring one in image is gaussian additive noise [3], which is frequently used to model thermal noise in natural. A gaussian additive noise can be modelled as follows:

$$ND = D + \epsilon \tag{1}$$

ND (noisy dataset) is the original dataset *D* which is corrupted by gaussian noise with unknown variance term ϵ.

However, label noise is much more harmful than feature noise in supervised learning task [24][9]. In general, label noise originates from a large amount of possibilities. For example, label noise could possibly occur because of insufficient information or evidence in the inference of labeling [4], mistakes caused by experts when labeling, subjectivity of experts (experts may have disagreement) [9], and encoding and communication problems [7].

Except concluding the sources of label noises, [7] also specifies different models of label noise into three taxonomies: NCAR (noisy completely at random model) where label noise is uniformly distributed, NAR (noisy at random model) where the mislabelling probability is depend on its true class and NNAR (noisy not at random model) where samples are more likely to be mislabelled to class that has overlapping with its true class in feature space i.e. feature correlated. [24] proposed a pairwise label noise model where two classes c1 and c2 are selected and each sample in c1 has a probability to be mislabelled as c2 and vice versa i.e. semantic correlated. [1] divides the label noise into two types based on the closed-set scenario: uniform and non-uniform random random label noise. [9] introduces some interesting properties of label noise in image: mislabelling is usually spatially correlated i.e. it is mostly unlikely that a single pixel is mislabelled but its surrounding are all correctly annotated. Normally a whole area in the image are wrongly annotated.

[8] defines the non-uniform types as different flipping probabilities for each class. [13] states that uniform label noise is challenging to be handled instead of non-uniform label noise. At the same time, the-state-of-art technical methods are not simply removing the corrupted data[1] while [5] suggests that it is useful to exploit the advantage of incorrected labeled data for semi-supervised training. No matter supervised learning, weakly supervised learning(semi-supervised), label noise model plays an essential role in a direct way or an indirect way. When learning algorithms have to be embedded directly, it should be of great importance to specify the model of label noise.

There are many works proposed to cope with label noise in literature as well. Generally outlier detection methods can be used to detect mislabeled instances in dataset. The approaches dealing with label noise are basically separated into 3 main branches in [7] i.e. approaches that have robustness to label noise, approaches that remove or relabel mislabelled samples before training procedure, approaches modeling label noise during training procedure. In deep learning, the robustness to label noise can be achieved by applying robust loss functions or regularization techniques [7]. The robust machine learning methods include random forest, logistic regression and some other boost methods. For label noise cleaning, a classifier is applied on the dataset and those pixels that are wrongly classified are simply removed from training for example use K-Nearest-Neighbor(KNN) [20] and SVM [6]. [17] introduced a methods based on directed graphics. [10] proposed to use probabilistic generative model using a kernel fisher discriminant. [18] employed conditional random fields to model the label noise. [12][23] modelled the uncertainty of label noise. [19] addressed this problem using CNNs and [16] provided a possibility to use unsupervised models.

Except dealing with label noise in the semantic segmentation task, some weakly supervised methods are proposed to semantically segment the images, where only the image-level label is given. [21] proposed an algorithm for image semantic segmentation under different forms of weak supervision i.e. image-level tags, bounding boxes and partial labels. [22] proposed a joint conditional random field based method for social image semantic segmentation in the presence of only image-level label even with noise. [11] converts weakly supervised semantic segmentation problem into a label noise reduction problem. Detailly each image is partitioned into a set of segments(super-pixels) and then the weak image-level labels are propagated to the superpixel level.

3. Methodology

For K-Nearest-Neighbor (KNN) classifier, we simply use patch-based feature representation. Namely the spectral information in the local neighborhood of the center pixel is used as its feature characterisation. The patch size of the neighborhood is not supposed to be very large, since the KNN algorithm suffers in curse of dimensionality.

The second method that we used for image semantic segmentation is Fully Convolutional Networks (FCN). We consider the SegNet [2] as it achieves satisfactory performance on a large amount of datasets. In principle, SegNet is an encoder and decoder architecture, which generates an output image or image patch with the same resolution as the input, which exactly fits the desire of image semantic segmentation task. The initial weight parameter is pre-trained from VGG-16 on ImageNet dataset for the purpose of fast convergence. Then the parameter is fine-tuned to make it suitable for remote sensing images. The encoder of SegNet has 5 convolutional blocks and each of them contains convolutional layers of filter 3x3 followed by a Rectified Linear Unit (ReLU) activation function. Each covolutional block is followed by a max pooling layer of size 2x2 to select the most important features. The structure of decoder is exactly symmetric to the encoder. Instead of convolutional layers and pooling layers, the decoder has deconvolutional layers and upsampling.

The gaussian additive noise is the most common type of feature noise in image acquisition. Therefore, we introduce gaussian additive noise with zero mean and user-defined standard deviation as feature noise in our research. For feature noise reduction, we apply non-local mean filter for denoising, which has a powerful performance on gaussian white noise. However, the non-local mean filter also removes some detail in the image.

For label noise in the reference maps, we consider two possibilities. Single pixel is mislabelled by the experts or a group of pixels (super-pixel) are wrongly annotated. In label noise injection, we divide the procedure into two parts. The first step is to decide or select which pixel should be chosen for relabelling. We provide three basic possibilities:

- randomly select pixels or superpixels from the image.
- select pixels or superpixels in the areas of high classification uncertainty based on the classification result of KNN classifier on noise free reference maps.
- select pixels or superpixels that are located at or close to the boundary of different objects.

To relabel those pixels or superpixels that are selected based on above, five possibilities are taken into consideration.

- select any other labels with equal probability (uniform label noise).
- select a new label based on the confusion matrix (semantically correlated) from KNN classification.
- select the label of nearest out-of-class neighbor in feature space i.e. radiometrically correlated.
- relabel the pixel based on its local neighbors.
- relabel the pixel based on the global neighbors in the reference map.

By selecting pixels and relabelling them, different types of label noise are injected to the reference maps

4. Experiments

4.1. Dataset

The dataset used in this work is *Vaihingen* dataset, which is commonly used in remote sensing image semantic segmentation and contains high resolution aerial image scene over Vaihingen city in Germany. In the reference semantic map, six categories are annotated i.e. impervious surface, building, low vegetation, tree, car and clutter (background). The entire dataset contains 33 tiles but only 16 have public ground truth data. Those images consist of three spectral bands: near infrared, red and green bands. Figure 1 gives an example image scene and corresponding reference map, where white, blue, cyan, green, yellow, red color denote the aforementioned classes respectively.

4.2. Metrics

Three widely used metrics (overall accuracy, average accuracy, kappa coefficient) are employed to quantitatively evaluate the generalized classification performance of KNN and FCN classifier. Confusion matrix is applied to summarize the detailed classification performance in each semantic class. Overall accuracy (OA) simply describes how many percentage of samples in the testset are correctly classified by the classifier. Besides overall accuracy, it is also necessary to report the mean accuracy averaged across all classes due to the fact that even if the classifier achieves a better performance in terms of overall accuracy, it might be true that the classifier performs quite worse in a single class.

However, both overall accuracy and average accuracy do not take the fact into consideration that agreements can also happen by chance. Therefore, Cohen's kappa coefficient (also called kappa score) is introduced, which is a more sophisticated and robust metric compared to a simple percent agreement metric. the kappa score κ measures how much better the agreements between two individual raters i.e. ground truth rater and classifier rater are beyond chance agreements. In other words, it measures the degree of inter-rater reliability or agreement between two raters.

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{2}$$

where p_o is the observed agreement among two raters i.e overall accuracy OA, p_e denotes the expected proportion of chance agreements.

The aforementioned overall accuracy, average accuracy or the agreements in Equation 2 can be easily calculated based on confusion matrix.

4.3. Baseline

In this subsection, we evaluate our implemented two classifiers on noisy free data i.e. no feature noise or label noise introduced.

In KNN classifier, we use 2000 samples in each class for training and image patch size of 3 for feature representation. The parameter K is 11. To make the comparison fair, these parameter stay unchanged in following experiments. As can be seen from Figure 2, KNN classifier could achieve overall accuracy 65.6%, 54.8% and kappa score 0.542 respectively.

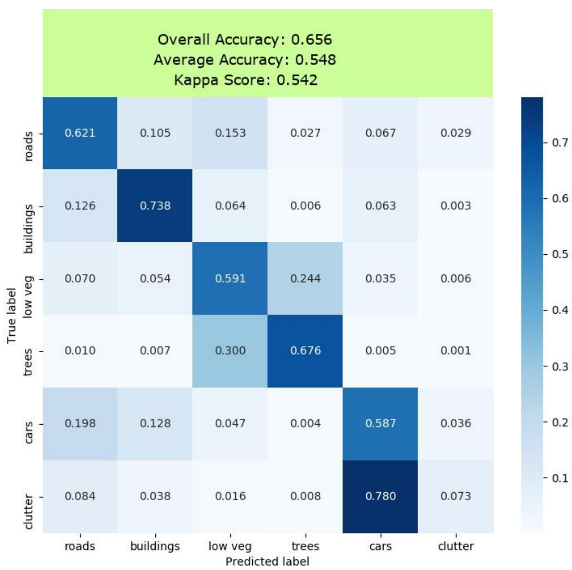
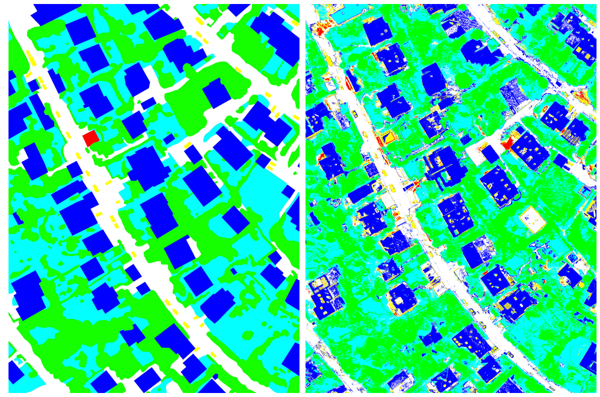


Figure 2: Classification performance of KNN classifier on noisy free data

The visualized classification map test image scene is illustrated in Figure 3. From Figure 3b, the pixels at the boundary of different objects seems to be difficult for the KNN classifier to predict.

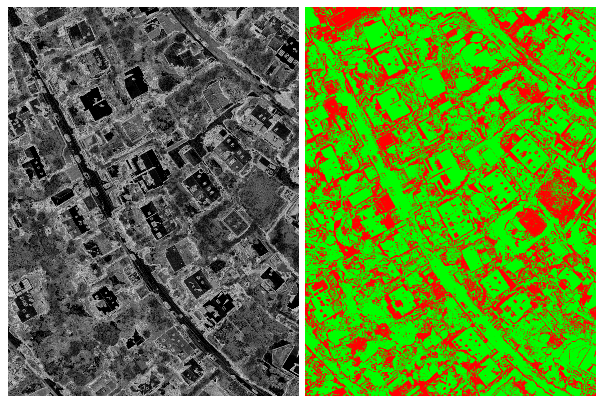
Figure 4 depicts the uncertainty map where the intensity value of pixel indicates the confidence of the classification decision made by the classifier and correct map, in which the pixels in green or red denotes that the pixels are correctly or wrongly classified.

In FCN model, the images and corresponding reference map are cropped to small image patch of size 256x256 fed



(a) Reference map (b) Estimated map

Figure 3: Classification map of KNN classifier



(a) Uncertainty map (b) Correct map

Figure 4: Uncertainty map and correct map of KNN classifier

into the input of the network. The learning rate is 0.01, mini-batch size 10 and epochs number 5.

As illustrated in Figure 5, with only 5 epochs, the FCN model could already achieve a satisfactory performance 80.8%, 64.0%, 0.739 in terms of overall accuracy, average accuracy and kappa score respectively. The visualized classification map is shown in Figure 6, where most samples are correctly classified.

4.4. Feature Noise

In practice, we inject feature noise to both training samples and test samples with zero mean and standard deviation 10 to the original imagery data. Figure 7 visually illustrates the difference between original image content and its corresponding noisy image and denoised image applying non-local mean filter via cropped image patch of size 400x400 to see the detail.

Figure 8 illustrates the estimated classification maps on the images with feature noise and denoised images by NLM

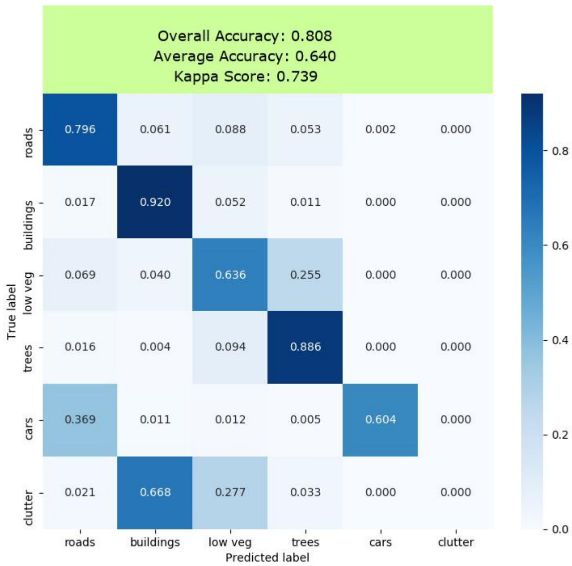
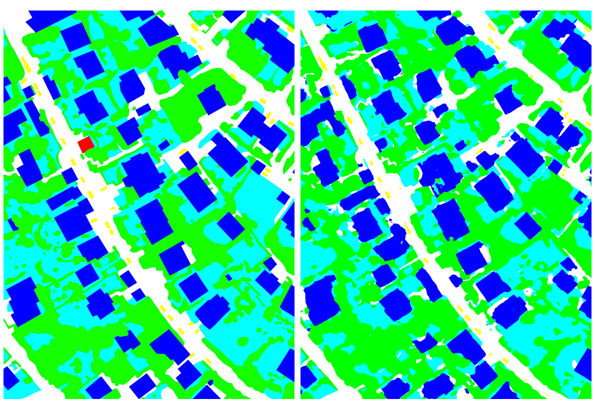
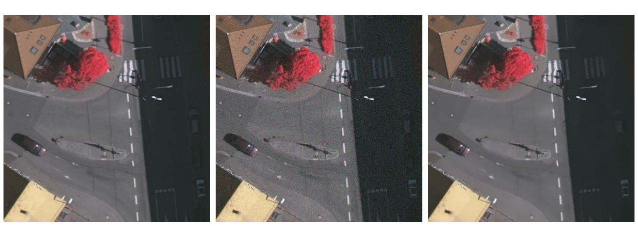


Figure 5: Classification performance of FCN on noisy free data



(a) Reference map (b) Estimated map

Figure 6: Classification map of FCN classifier



(a) Original patch (b) Noisy patch (c) Denoised patch

Figure 7: From left to right: original image patch cropped from Vaihingen dataset, corresponding noisy patch and denoised patch by NLM filter

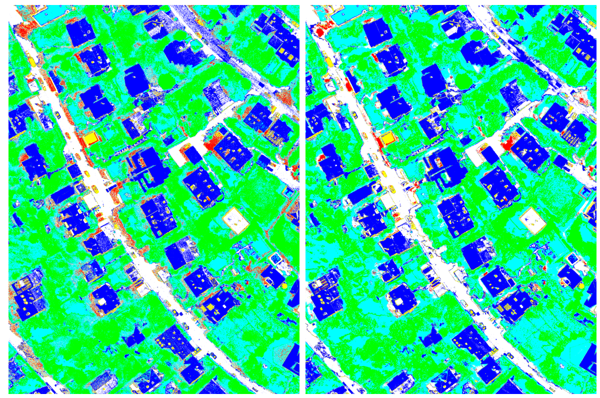


Figure 8: Classification map of KNN classifier on noisy images (left) and denoised images (right) by NLM filter

filter. Due to the limit of the paragraph, the detailed classification performance and class posterior maps will not be depicted. Overall speaking, the KNN classifier could achieve overall accuracy 62.2%, average accuracy 50.8% and kappa score 0.499 on the imagery data with gaussian additive noise. Compared to the performance in Figure 2, all three measurements decreased by 3.4%, 4.0% and 0.043 separately.

After applying NLM filter for reducing gaussian additive noise, KNN classifier achieves overall accuracy 65.0%, average accuracy 52.7% and kappa score 0.533, which are competitive to the performance on noisy free data in Figure 2 and better than the performance on noisy image data.

When applying FCN model to the noisy and denoised image data, the estimated classification maps are illustrated in Figure 9. The FCN model is able to achieve overall accuracy 80.4%, average accuracy 64.6% and kappa score 0.734 on image data with feature noise. Compared to the performance on noisy free data in Figure 5, the performance only decreases by 0.4%, 0.5% in terms of overall accuracy and kappa score. The average accuracy even increases by 0.6%. On denoised image data by NLM filter, the overall accuracy, average accuracy and kappa score reach to 80.2%, 60.2% and 0.731. Generally speaking, the FCN model seems to be robust to the feature noise in the imagery data.

4.5. Label Noise

As stated before, we consider that there mainly exists two types of label noise, single pixel label noise or superpixel label noise. In each of them, we have three possible ways to select pixels or superpixels for relabelling and three possibilities to assign a new label to selected samples.

Two experiments are conducted to investigate the influence of label noise on the image semantic segmentation task. One is that the training samples have label noise but

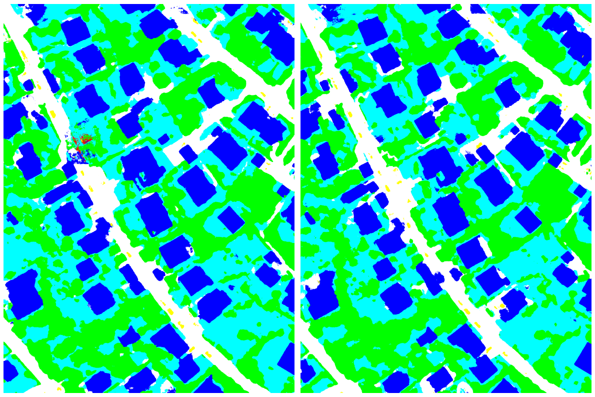


Figure 9: Classification map of FCN classifier on noisy images (left) and denoised images (right) by NLM filter

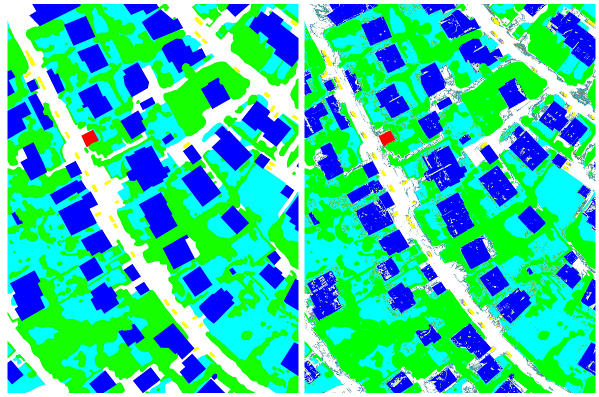
test samples are noise free. We try to measure to what extend the label noise affects the classification performance under usage of different classifiers. Another is to introduce label noise to test set however train set is not injected with any label noise. We measure and evaluate to what degree we can rely on the estimated performance of the classifier on the dataset.

4.5.1 Single-pixel

In pixels selection procedure, we inject 10% random distributed label noise to the reference maps in case of randomly sampling. When selecting pixel for relabelling based on uncertainty map from KNN classifier, we select 10% of total number of pixels with high uncertainty value. If selecting the pixels in close proximity to the boundary of different objects, we model a small gaussian distribution with kernel size five centered at the pixels in the contour of objects. In order to make the experiments in different cases comparable, we randomly select 10% of total number of pixels from objects boundary.

As an example, Figure 10 illustrates the noisy reference map selecting the pixels based on uncertainty map and relabelling based on radiometric characteristic (out of class neighbor in feature space). It seems that the KNN classifier is confused in its decision at the boundary pixel of different object as shown in Figure 10b.

Table 1 depicts the overall accuracy training KNN classifier on noisy training samples with different types of label noise. Interestingly, randomly injecting uniform label noise does not lead to a decrease of the performance, which indicates that the hypothesis of traditional label noise model might not be realistic. Injecting uniform label noise to the uncertainty area seems to have a significant decrease of performance from 65.6% to 63.0%.



(a) Reference map (b) Noisy reference map

Figure 10: Original reference map and exemplary noisy reference map selecting the pixels in uncertainty areas and relabelling them by out-of-class neighbor in feature space

Relabel \ Selection	random	uncertainty	boundary
uniform	65.7%	63.0%	65.7 %
confusion matrix	65.2%	63.9%	65.1%
local neighbor	65.1%	65.1%	65.5%
global neighbor	65.0%	63.9%	65.4%
radiometry	64.6%	65.0%	65.2 %

Table 1: Overall accuracy injecting different types of label noise to training samples applying KNN classifier

Table 2 depicts the estimated overall accuracy on noisy reference maps. In case of randomly selecting pixels for relabelling, the model achieves relatively worse performance than selecting pixels in uncertainty area or objects boundary, which is understandable, because the pixels in uncertainty area or objects is naturally difficult for KNN classifier to predict and relabelling them would have too much effect on the final classification performance. In other words, the uniform label noise seems to have less influence on training a KNN classifier but have a serious influence on performance evaluation of test images. If there exists uniform label noise in test samples, we probably underestimate the classification performance of the classifier.

Table 3 shows the classification accuracy using FCN network training with different types of label noise. Similarly, the classifier achieves worst performance 82.4% when selecting pixels in uncertainty area and adding uniform label noise as using KNN classifier in Table 1. Overall speaking, the FCN network is robust to the label noise in the

Relabel \ Selection	random	uncertainty	boundary
uniform	59.9%	63.6%	65.0%
confusion matrix	60.8%	64.2%	65.2%
local neighbor	60.9%	65.1%	65.7%
global neighbor	60.5%	63.8%	65.3%
radiometry	60.2%	64.0%	64.6%

Table 2: Overall accuracy injecting different types of label noise to testing samples applying KNN classifier

training samples. Figure 11 illustrates an exemplary example in training phase using FCN. As we can see, in the ground truth data, some randomly distributed samples are relabelled. But the output of the network still correctly predicts the pixels in the homogeneous region.

Relabel \ Selection	random	uncertainty	boundary
uniform	80.4%	77.6%	80.4%
confusion matrix	79.2%	75.8%	81.6%
local neighbor	80.6%	78.2%	80.3%
global neighbor	76.2%	81.7%	70.8%
radiometry	79.0%	72.2%	76.1%

Table 3: Overall accuracy injecting different types of label noise to training samples applying FCN classifier

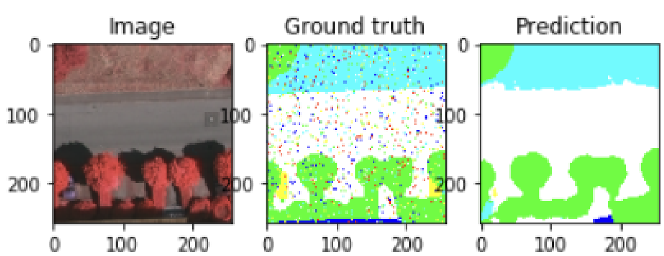


Figure 11: Training example using FCN

Table 4 illustrates the evaluated performance on noisy reference map but trained on noisy free data. It seems that given a specific pixel selection methods the FCN achieves similar overall accuracy even if the relabelling methods is different.

4.5.2 Super-pixel

4.5.2.1 Pre-processing

Relabel \ Selection	random	uncertainty	boundary
uniform	71.9%	72.3%	76.1%
confusion matrix	71.8%	73.8%	72.4%
local neighbor	77.0%	77.2%	76.0%
global neighbor	73.4%	78.0%	73.7%
radiometry	73.8%	73.4%	74.8%

Table 4: Overall accuracy injecting different types of label noise to testing samples applying FCN classifier

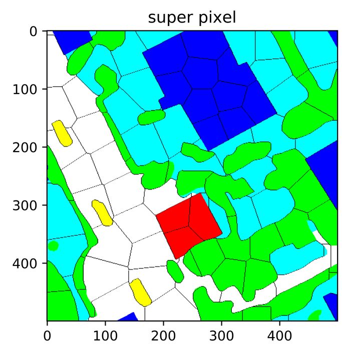


Figure 12: super-pixel in reference map using SLIC

Pre-processing For super pixel procedure,we utilize **SLIC**(Simple Linear Iterative Clustering) algorithm to generate a sequence of superpixels.This algorithm will guarantee that the size of each super pixel will be more or less identical.Here the size of super-pixel can be simply calculated as follows:

$$S = \sqrt{N/K} \tag{3}$$

In Equation 3, N is the total number of super pixels and K is the pixel number of each super pixels. Figure 12 illustrates the shape of super pixel in reference map and the super-pixel can well fit the boundary of corresponding classes. The label of each super should be defined, for simplicity, the label can be defined by the majority of single pixel in each super pixel.

4.5.2.2 Injecting super pixel label noise As mentioned in section 4.5, the label noise injecting will follow two steps as follow:

1) The selection of super pixel.There are three selection types: Random Selection, Uncertainty Area Selection, Boundary Selection. Before selecting super pixel, the probability of comparison between different type of label noises need to be taken into account as well. In this project, the noise level will be set, for simplicity, the size of each super pixel is regarded as the identical

number,

$$M = N * r/s^2 \tag{4}$$

In Equation 4, M is the noise super pixel number, N is the total number of single pixel,r is the noise level, s is the distance of super pixel.

2) Relabel function of each super pixel.There are five relabel function: Random Relabel, Confusion Matrix Relabel, Global Relabel, Local Relabel, Nearest Relabel(feature space).

4.5.2.3 super pixel selection

Selection of super pixel: Random All selected super pixels are generated in a random distribution.

Selection of super pixel: Uncertainty Area According to the uncertainty map, the higher uncertainty area will be selected. All the super pixels will be selected from the highest to the lower until to the limitation of noise level. Based on the entropy map in Figure 4, the uncertainty area will be selected the brighter area.

Selection of super pixel: Boundary Here only the super pixel which is touching or intersecting the boundary of each classes will be selected. However, in order to guarantee the noise level, there are three situations will be taken into account, in Figure 13 illustrates how the super pixel will be selected under different noise levels:

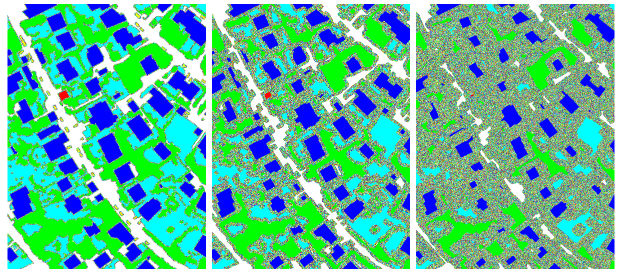
a) The number of entire super pixel located on the boundary exactly meets the noise level, so all super pixel will be selected;

b) The number of entire super pixel located on the boundary can not meet the noise level, so extra super pixel of its neighbors will be selected until to the limitation of noise level;

c) The number of entire super pixel located on the boundary exceeds the noise level, so super pixel will be randomly selected until to the limitation of noise level;

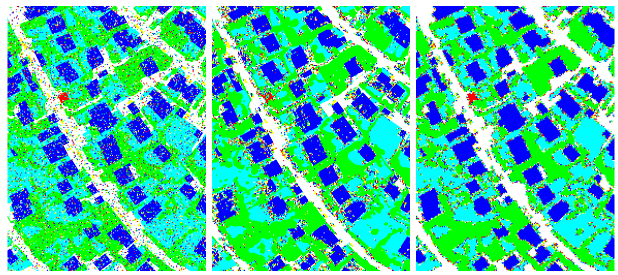
4.5.2.4 super pixel relabel

super pixel relabel: Random The entire selected super pixels will be relabelled randomly under all types of classes.In this project, there are six classes, so each super pixel will be relabelled within the six types.Figure 14 illustrates the randomly relabelled examples.



(a) noise level: 0.1 (b) noise level: 0.5 (c) noise level: 0.75

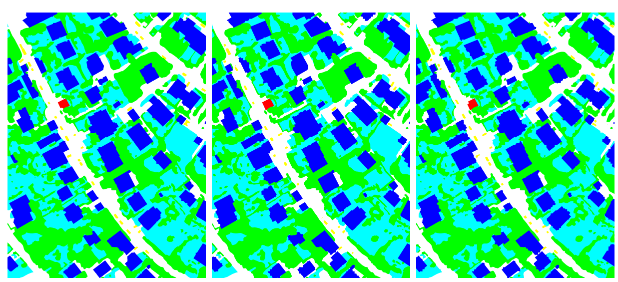
Figure 13: Super pixel selection base on Boundary, Super pixel size: 3



(a) Random (b) Uncertainty (c) Boundary

Figure 14: Super pixel relabel:Random base on noise level:0.2, Super pixel size: 10

super pixel relabel: Local neighborhood The super pixels will be relabelled by its local neighbour, the local neighbors will be counted if one class has the most neighbors to the corresponding super pixel. So in general only the super pixel closed to the boundary will be relabelled.Figure 15 illustrates only between two two different classes there are some changes.



(a) Random (b) Uncertainty (c) Boundary

Figure 15: Super pixel relabel:local neighbor base on noise level:0.1, Super pixel size: 20

super pixel relabel: Global neighborhood The super pixels will be relabelled according to each class has how many touching other classes based on the whole reference map. For the reference map Figure 3 , the global relationship matrix Figure 16 can be built to reveal the the relationship.In each row the true class can be found which another is the target. In Figure 17,the most neighbors feature can be found.

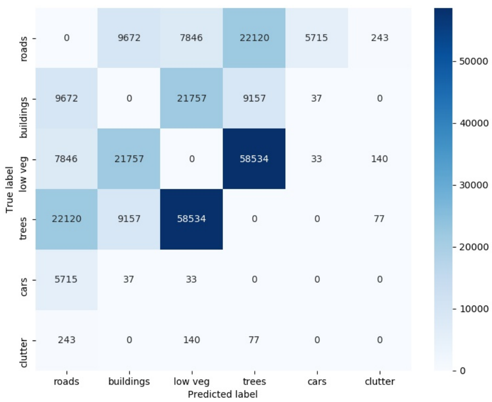
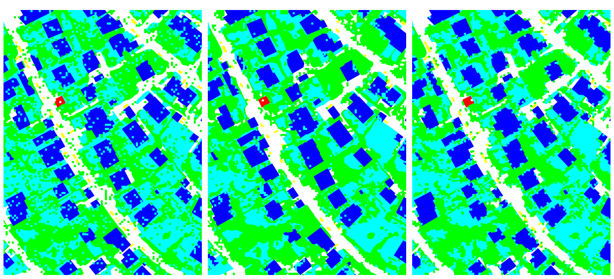


Figure 16: The relationship between each class based on pixel level

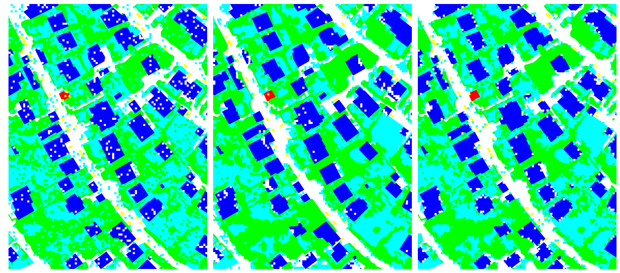


(a) Random (b) Uncertainty (c) Boundary

Figure 17: Super pixel relabel:global relationship base on noise level:0.1, Super pixel size: 20

super pixel relabel:Confusion Matrix Another relabel type can be deduced from confusion matrix. in the confusion matrix, the most confused classed can be obtained and the super pixel can be relabelled to the corresponding most confused labels.The relabelled results illustrates in Figure 18

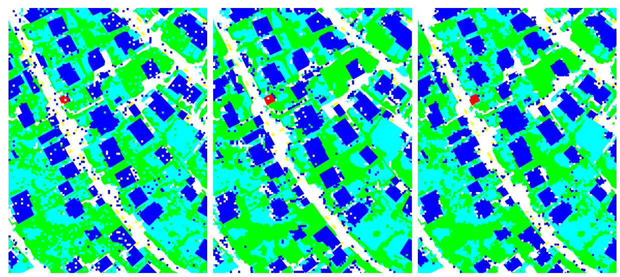
super pixel relabel: Nearest neighbors in feature space Besides the above four relabel type in reference space, the relabel function can be also considered in the feature space, in the feature space, the nearest neighbors will be selected based on KNN.For each super pixel, its corresponding nearest neighbors can be calculated and the super



(a) Random (b) Uncertainty (c) Boundary

Figure 18: Super pixel relabel :Confusion Matrix base on noise level:0.1, Super pixel size: 20

pixel will be relabelled to the label. Figure 19 depicts the results of relabelled function in the feature space.

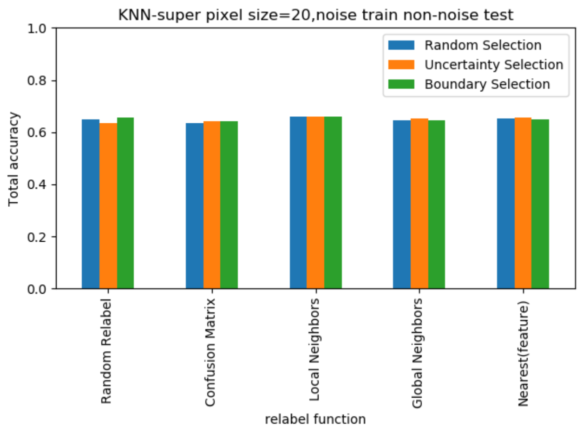


(a) Random (b) Uncertainty (c) Boundary

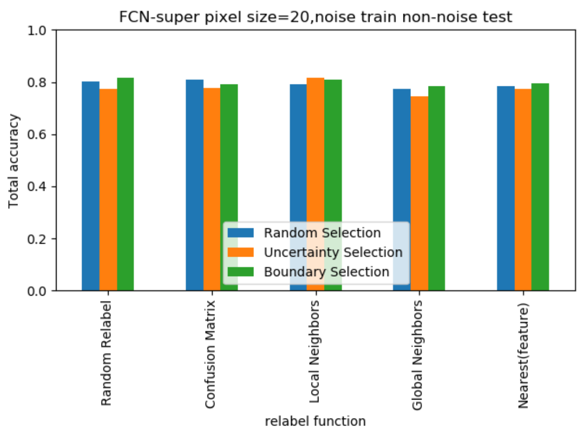
Figure 19: Super pixel relabel :Nearest neighbores in feature space base on noise level:0.1, Super pixel size: 20

4.5.2.5 Training and testing the injected noise data

General training and testing In order to evaluate the result of injected label noise data, two different classifiers will be applied: one is the simple classifier: KNN, another one will exploit the current popular classifier: FCN. For each classifier, as mentioned in the beginning of this section, each training dataset and testing dataset will be considered in two types: the first one is: the training dataset will be injected super pixel label noise and there is no injected label noise in the testing dataset . The another is versus: the testing dataset will contains injected super pixel label noise and the training dataset does not contain injected super pixel noise at all. Thirdly, for super pixel, the size is another important factor that should be considered. Here, the super pixel size will be tested in two situation: one is a small super pixel size: 3, another is relatively larger: 20. In Appendix A, there listed all results in this project for super pixel. In total, there are twenty four situations.



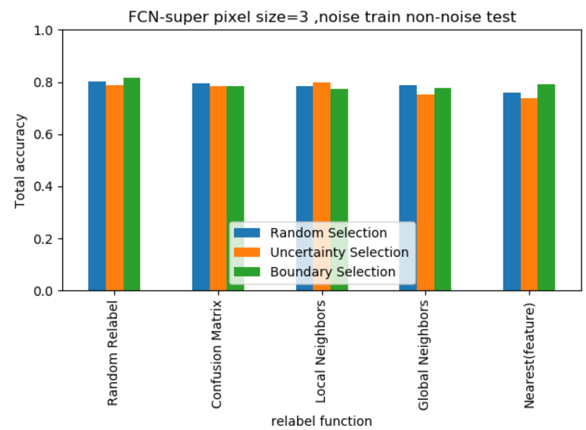
(a) KNN



(b) FCN

Figure 20: Different classifiers: KNN and FCN

(a) KNN



(b) FCN

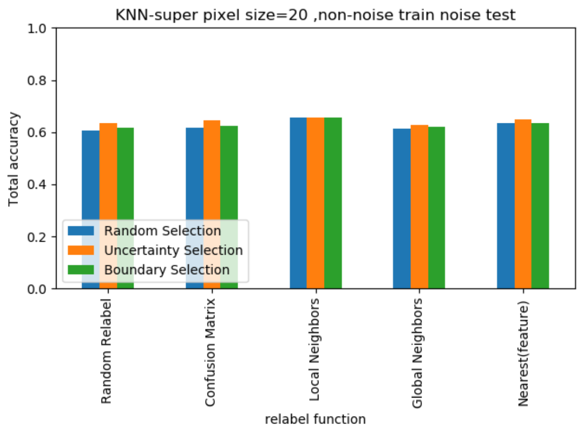
Figure 21: Different superpixel size with Figure 20: KNN and FCN

Different classifiers: KNN and FCN Figure 20 illustrates that for classifier KNN, under a special selection approach, the total accuracy does not change much with the relabel function except the randomly relabelled function.I n general, the local neighbor label noise affects slightly the total accuracy because only between two classes, the super pixels' label will be changed. For FCN, due to the properties of FCN, in general, the accuracy of classification is better than KNN. However, the accuracy of each situation fluctuates much more greater than KNN. Similar to KNN, injecting local neighbor label noise has a little higher total accuracy than others, and based on boundary selection approach, except based on confusion matrix situation, all the results has a better performance. Based on the random selection, the performance behaves randomly.

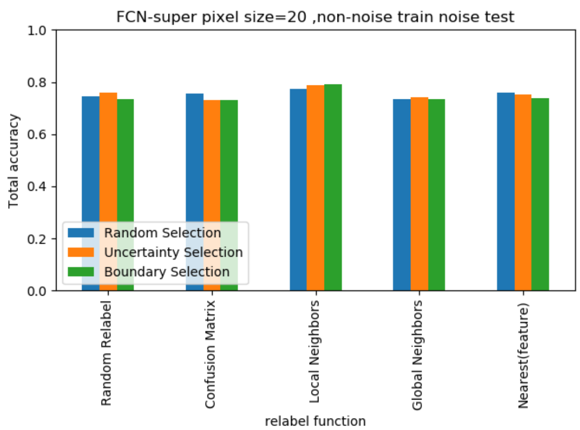
Different super pixel size Figure 20 and Figure 21 illustrate how the corresponding classifiers perform with different size of super pixel. Single pixel can be regarded as a special super pixel: single pixel is a kind of super pixel which size is equal to one. For the classifier KNN, when the size of super pixel is set to three, the result injected label noise based on nearest neighbor in feature space performs a little better than other cases. But there is not much difference except the cases: randomly super pixel selection or randomly super pixel relabel function. For classifier FCN, there is much more change using different size of super pixels.

Different training and testing approaches

Different super pixel size Here two different training and testing approaches are applied as mentioned in the beginning of this section. Figure 22 illustrates that, for classifier KNN, this situation does not affect the performance a lot compared with the classifier FCN. Both cases, injecting la-



(a) KNN



(b) FCN

Figure 22: Different training and testing approaches Figure 20: KNN and FCN

bel noise based on local neighbors has a better performance.

5. Conclusion

To sum up, in this paper we investigate the influence of feature noise and label noise on image semantic segmentation task. The influence is researched on K-Nearest-Neighbor (KNN) classifier and Fully Convolutional Networks (FCN) to show their robustness to feature noise and different types of label noise.

We consider the gaussian additive noise as the feature noise in image acquisition. For label noise, totally 15 possibilities are implemented. The experiments indicate that FCN is more robust to the label noise in training data than KNN classifier. However, the generalization performance is highly degraded by the label noise in test data for both of the classifiers especially FCN network.

References

[1] Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*, June 2019. 2

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 3

[3] Charles Boncelet. Image noise models. In *The Essential Guide to Image Processing*, pages 143–167. Elsevier, 2009. 2

[4] Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167, 1999. 2

[5] Y. Ding, L. Wang, D. Fan, and B. Gong. A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1215–1224, 2018. 2

[6] Sergiy Fefilatyev, Matthew Shreve, Kurt Kramer, Lawrence Hall, Dmitry Goldgof, Rangachar Kasturi, Kendra Daly, Andrew Remsen, and Horst Bunke. Label-noise reduction with support vector machines. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3504–3508. IEEE, 2012. 2

[7] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013. 1, 2

[8] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10456–10465. Curran Associates, Inc., 2018. 2

[9] R. Hänsch and O. Hellwich. The truth about ground truth: Label noise in human-generated reference data. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5594–5597, 2019. 1, 2

[10] Neil Lawrence and Bernhard Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. pages 306–313, 05 2001. 2

[11] Zhiwu Lu, Zhenyong Fu, Tao Xiang, Peng Han, Liwei Wang, and Xin Gao. Learning from weak and noisy labels for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):486–500, 2016. 2

[12] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. 10 2019. 2

[13] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[14] Franz Rottensteiner, Gunho Sohn, Markus Gerke, and Jan D Wegner. Isprs semantic labeling contest. *ISPRS: Leopoldshöhe, Germany*, 2014. 1

[15] Arun Saseendran, Lovish Setia, Viren Chhabria, Debrup Chakraborty, and Aneek Barman Roy. Impact of noise in dataset on machine learning algorithms. 02 2019. 2

[16] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. *CoRR*, abs/1803.11364, 2018. 2

[17] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2699, 2015. 2

[18] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5596–5605. Curran Associates, Inc., 2017. 2

[19] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J. Belongie. Learning from noisy large-scale datasets with minimal supervision. *CoRR*, abs/1701.01619, 2017. 2

[20] D Randall Wilson and Tony R Martinez. Reduction techniques for instance-based learning algorithms. *Machine learning*, 38(3):257–286, 2000. 2

[21] Jia Xu, Alexander G Schwing, and Raquel Urtasun. Learning to segment under various forms of weak supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3781–3790, 2015. 2

[22] Wei Zhang, Sheng Zeng, Dequan Wang, and Xiangyang Xue. Weakly supervised semantic segmentation for social images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2718–2726, 2015. 2

[23] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. 03 2020. 2

[24] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artif. Intell. Rev.*, 22:177–210, 11 2004. 2

A. Super Pixel training and testing results

CM: Confusion Matrix LN:Local Neighbor GL:Global Neighbor NN:Nearest Neighbor(Feature) RN:Random

Noise level: 0.1, Classifier:FCN, Super pixel size:3					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	70.09%	78.59%	73.68%	74.36%	70.47%
Average Accuracy	58.22%	62.65%	55.52%	70.93%	51.17%
Kappa Score	59.51%	71.19%	63.82%	65.40%	61.12%

Table 1: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:3					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	75.64%	78.62	77.24%	76.74%	74.10%
Average Accuracy	59.08%	57.34%	64.42	58.76%	51.92%
Kappa Score	66.62%	71.15%	68.73%	68.37%	65.06%

Table 2: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:3					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	71.85%	74.95%	70.14%	74.22%	73.40%
Average Accuracy	55.14%	58.62%	55.44%	56.15%	53.13%
Kappa Score	61.82%	65.99%	58.80%	65.04%	64.47%

Table 3: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:20					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	75.38%	77.23%	73.28%	76.04%	74.39%
Average Accuracy	57.28%	59.19%	68.13%	64.60%	53.31%
Kappa Score	66.59%	69.36%	63.65%	67.77%	65.84%

Table 4: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:20					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	73.06%	78.87%	73.96%	75.35%	75.97%
Average Accuracy	60.52%	64.08%	59.93%	59.30%	54.64%
Kappa Score	63.49%	71.49%	64.31%	66.53%	67.59%

Table 5: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:20					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	72.98%	79.13%	73.44%	73.77%	73.27%
Average Accuracy	55.76%	63.77%	60.03%	59.21%	53.65%
Kappa Score	63.31%	71.93%	63.86%	64.72%	64.39%

Table 6: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:3					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	78.51%	77.41%	77.80%	78.98%	81.54%
Average Accuracy	54.30%	55.52%	55.22%	59.51%	63.26%
Kappa Score	70.83%	69.19%	69.90%	71.39%	74.94%

Table 7: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:3					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	78.50%	79.99%	75.05%	73.93%	78.73%
Average Accuracy	65.19%	62.86%	58.45%	53.38%	64.41%
Kappa Score	70.97%	73.00%	65.62%	64.68%	71.51%

Table 8: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:3					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	79.30%	78.24%	78.83%	75.99%	80.24%
Average Accuracy	60.62%	62.10%	59.94%	56.84%	79.11%
Kappa Score	71.79%	70.56%	71.39%	67.41%	73.34%

Table 9: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:20					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	80.88%	79.27%	77.17%	78.57%	80.16%
Average Accuracy	59.51%	63.30%	58.65%	59.84%	60.03%
Kappa Score	74.02%	72.14%	69.13%	71.04%	73.02%

Table 10: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:20					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	77.72%	81.58%	74.54%	77.48%	77.43%
Average Accuracy	63.39%	64.86%	55.09%	62.12%	58.87%
Kappa Score	69.85%	75.03%	64.94%	69.69%	69.42%

Table 11: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:FCN, Super pixel size:20					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	79.08%	80.73%	78.26%	79.59%	81.44%
Average Accuracy	60.95%	61.40%	58.78%	63.37%	63.75%
Kappa Score	71.69%	73.85%	70.21%	72.53%	74.85%

Table 12: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:3					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	61.55%	66.10%	61.04%	62.74%	60.72%
Average Accuracy	52.48%	54.51%	52.24%	52.90%	44.76%
Kappa Score	48.64%	54.77%	47.73%	50.35%	48.20%

Table 13: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:3					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	64.55%	65.42%	63.69%	65.46%	64.15%
Average Accuracy	54.77%	54.76%	55.14%	55.24%	49.10%
Kappa Score	52.48%	54.00%	51.12%	53.97%	52.61%

Table 14: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:3					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	65.59%	62.17%	64.83%	59.95%	
Average Accuracy	54.54%	54.90%	54.56%	44.25%	
Kappa Score	54.15%	49.02%	52.97%	47.75%	

Table 15: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:20					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	62.54%	65.53%	62.00%	63.53%	61.62%
Average Accuracy	53.32%	54.80%	52.58%	54.01%	45.51%
Kappa Score	49.96%	54.04%	49.10%	51.45%	49.42%

Table 16: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:20					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	64.45%	65.74%	62.72%	64.96%	63.50%
Average Accuracy	55.45%	55.78%	54.76%	55.40%	47.75%
Kappa Score	52.22%	54.27%	49.88%	53.32%	51.64%

Table 17: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:20					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	61.64%	65.57%	61.19%	63.39%	60.77%
Average Accuracy	51.59%	54.48%	52.42%	53.24%	44.80%
Kappa Score	48.68%	54.08%	48.00%	51.24%	48.31%

Table 18: Training data: non-injected label noise, Testing data: injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:3					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	63.11%	65.69%	65.03%	65.04%	65.28%
Average Accuracy	50.21%	54.08%	54.54%	54.05%	52.11%
Kappa Score	51.05%	54.28%	53.48%	53.41%	53.70%

Table 19: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:3					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	64.46%	65.81%	65.04%	65.65%	63.41%
Average Accuracy	52.30%	54.03%	53.72%	54.22%	52.32%
Kappa Score	52.88%	54.41%	53.23%	54.13%	52.00%

Table 20: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:3					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	63.76%	65.55%	65.17%	66.28%	63.64%
Average Accuracy	50.44%	54.74%	52.97%	53.83%	49.79%
Kappa Score	52.02%	54.15%	53.57%	54.93%	52.20%

Table 21: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:20					
Select Type	Boundary				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	64.28%	65.79%	64.69%	64.80%	65.44%
Average Accuracy	52.15%	54.57%	53.86%	53.90%	52.39%
Kappa Score	52.61%	54.43%	52.98%	53.13%	54.07%

Table 22: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:20					
Select Type	Uncertainty Area				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	64.05%	65.86%	65.08%	65.54%	63.36%
Average Accuracy	51.82%	54.68%	53.96%	54.16%	52.02%
Kappa Score	52.42%	54.43%	53.36& 53.99%	51.81%	

Table 23: Training data: injected label noise, Testing data: non-injected label noise

Noise level: 0.1, Classifier:KNN, Super pixel size:20					
Select Type	Random				
Relabel Type	CM	LN	GN	NN	RN
Total Accuracy	63.43%	66.03%	64.66%	65.06%	65.03%
Average Accuracy	51.00%	54.62%	53.91%	54.06%	52.06%
Kappa Score	51.50%	54.70%	52.98%	53.42%	53.40%

Table 24: Training data: injected label noise, Testing data: non-injected label noise