Dense prediction of label noise for learning building extraction from aerial drone imagery

Nahian Ahmed[a]*, Rashedur M. Rahman[a], Mohammed Sarfaraz Gani Adnan[b], Bayes Ahmed[c]

[a]*Department of Electrical and Computer Engineering, School of Engineering and Physical Sciences, North South University, Bashundhara, Dhaka 1229, Bangladesh;*

[b]*Department of Urban and Regional Planning, Chittagong University of Engineering and Technology (CUET), Chittagong 4349, Bangladesh;*

[c]*Institute for Risk and Disaster Reduction (IRDR), University College London (UCL), Gower Street, London WC1E 6BT, UK;*

Email: nahian.ahmed@northsouth.edu

Dense prediction of label noise for learning building extraction from aerial drone imagery

Label noise is a commonly encountered problem in learning building extraction tasks; its presence can reduce performance and increase learning complexity. This is especially true for cases where high resolution aerial drone imagery is used, as the labels may not perfectly correspond/align with the actual objects in the imagery. In general machine learning and computer vision context, labels refer to the associated class of data, and in remote sensing-based building extraction refer to pixel-level classes. Dense label noise in building extraction tasks has rarely been formalized and assessed. We formulate a taxonomy of label noise models for building extraction tasks, which incorporates both pixel-wise and dense models. While learning dense prediction under label noise, the differences between the ground truth clean label and observed noisy label can be encoded by error matrices indicating locations and type of noisy pixel-level labels. In this work, we explicitly learn to approximate error matrices for improving building extraction performance; essentially, learning dense prediction of label noise as a subtask of a larger building extraction task. We propose two new model frameworks for learning building extraction under dense real-world label noise, and consequently two new network architectures, which approximate the error matrices as intermediate predictions. The first model learns the general error matrix as an intermediate step and the second model learns the false positive and false negative error matrices independently, as intermediate steps. Approximating intermediate error matrices can generate label noise saliency maps, for identifying labels having higher chances of being mis-labeled. We have used ultra-high-resolution aerial images, noisy observed labels from OpenStreetMap, and clean labels obtained after careful annotation by the authors. When compared to the baseline model trained and tested using clean labels, our intermediate false positive-false negative error matrix model provides Intersection-Over-Union gain of 2.74% and F1-score gain of 1.75% on the independent test set. Furthermore, our proposed models provide much higher recall than currently used deep learning models for building extraction, while providing comparable precision. We show that intermediate false positive-false negative error matrix approximation can improve performance under label noise.

Keywords: label noise, building extraction, dense prediction, deep learning, remote sensing

Introduction

Building extraction involves learning mappings between remotely sensed aerial or satellite images and building labels from freely available vector data. The most commonly used source of labels, OpenStreetMap, though accurate to a large degree, contain various types of label noise (Mnih and Hinton, 2012; Ahmed et al., 2020; Zhang et al., 2020). Pixel-level predictions of building/non-building labels are performed, which is a binary dense prediction task. Label noise occurs when the observed label does not agree with the true label (Frénay and Verleysen, 2013; Frénay and Kabán, 2014) (Fig. 1). Presence of label noise in training data can reduce performance, while noise in testing data can lead to underestimation of model performance (Ahmed et al., 2020). However, most of the existing studies on deep learning-based building extraction do not acknowledge the presence of label noise. In general, complexity of the learning task is also increased under label noise (Garcia et al., 2015; Pelletier et al., 2017). Research on robust method of building extraction considering label noise requires formalization of the sources, processes and effects of noise on large scale freely available labels. Currently, the types of dense label noise processes have not been formalized in a comprehensively and inclusively in research. When building polygons are rasterized, the buildings are represented as superpixels in the prepared dense binary labels. Individual building polygon i.e. superpixel based errors are commonly considered as sources of noisy labels.

Coming from traditional remote sensing terminology, the most common are registration errors, where building polygons are present but not aligned, annotated or registered properly, and omission errors where buildings are left unlabeled (Mnih and Hinton, 2012; Ahmed et al., 2020; Zhang et al., 2020). However, alternative nomenclature has been proposed as well. Pixel-based nomenclature can be used to express label noise processes in multiple scales, and therefore provides a more generalized viewpoint. Even superpixel-based label noise processes are modeled using a composite of pixel-based processes (Mnih and Hinton, 2012; Zhang et al., 2020). This approach assumes that each pixel undergoing label noise is independent of and identical to label noise processes in other (even neighboring) pixels. This scenario is analogous to the use of label noise robust pixel-based building extraction methods such as logistic regression (Maas et al., 2016), random forests (Maas et al., 2019), compared to the use of deep learning-based label noise robust building extraction methods such as fully convolutional networks and U-Nets (Zhang et al., 2020). The primary difference between non-deep learning and deep learning-based building extraction is that the former usually uses features from only the pixel being classified, whereas the latter leverages context to predict dense labels for the entire image at once. Feature representation is an important part of deep learning based remote sensing image processing (Jing et al., 2021; He et al., 2021). Modeling of superpixel based label noise process has been conducted for the general computer vision task of semantic segmentation (Lu et al., 2016), but has largely been left unexplored for remote sensing applications. If building extraction can be modeled using a dense prediction approach, we argue that pixel-based label noise robustness approaches can also be extended to dense prediction-based label noise robustness approaches.

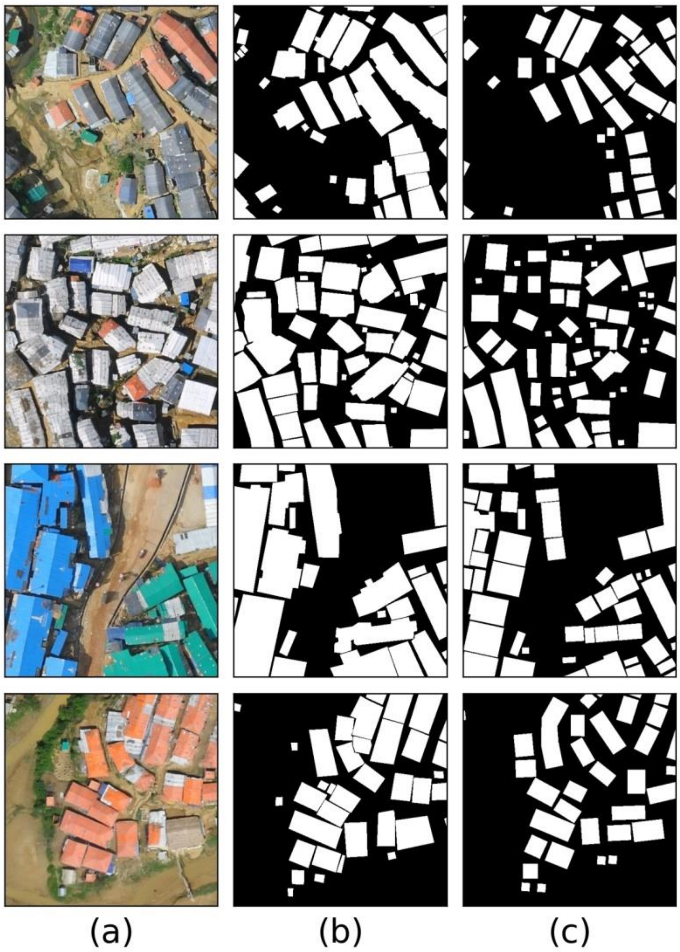


Figure 1. Some examples of large image tiles from our dataset. (a) Image (b) True clean dense labels (c) Observed dense labels from OpenStreetMap with real world noise

There are various aspects of viewing the label noise generation process. Labeling tools used by human annotators also play a role in determining the label noise processes for dense prediction tasks (Frank et al., 2017). Simulated noise is common in label noise robust image classification scenarios (Ghosh et al., 2017; Rolnick et al., 2017; Patrini et al., 2017) and can be extended to dense prediction-based building extraction as well, however, we have access to data with real-world dense label noise. It is also important to acknowledge the limitations of simulated noise when compared to real-world noise (Jiang et al., 2020). Label noise processes can broadly be categorized by their randomness (Frénay, B., & Verleysen, 2013). For example, if certain building superpixels are being omitted in the observed labels, the question arises, are these buildings being selected totally at random, or are certain types of buildings, perhaps newly constructed buildings, being omitted. Randomness characterizes label noise processes. Identifying this randomness is crucial for modeling label noise robust learning systems. Randomness is unique to each dataset and is estimated prior to modeling solutions.

We have quantified the effects of label noise on evaluation regimes for this dataset and found that deep neural networks for semantic segmentation are intrinsically robust to real world random label noise, specially aided if data augmentation and regularization are introduced (Ahmed et al., 2020). However, robustness to label noise is achieved as a by-product of overfitting-reduction schemes, and therefore the modelling of label noise is implicit. In this work, we explicitly model dense label noise as a subtask of building extraction, and show improved performance on independent test set.

The primary objective of this study is to analyze label noise robustness of deep semantic segmentation networks using our proposed evaluation regime. State-of-the-art methods for deep learning-based building extraction from remotely sensed imagery usually perform model evaluation using noisy labels as ground truth, we test the effects of performing model evaluation against noisy labels and clean labels. Our contributions are as follows. We outline approaches for modeling dense label noise and formalize a multi-view and multi-scale taxonomy of label noise. We propose two new model frameworks for building extraction from aerial drone imagery under dense label noise, and consequently two new network architectures. Our network architectures approximate the dense label noise characterizing error matrices as an intermediate step to improve performance. Approximating intermediate error matrices can generate label noise saliency/heat maps. We have made our dataset and method implementations publicly available (https://drive.google.com/uc?id=1UUGeewOaNzv_8kMGXOgEzR8_QKPlPsr8) (https://github.com/nahian-ahmed/dense-label-noise).

Dense label noise models

Preliminaries and definitions

Formulations on label noise in non-dense approaches are well defined and studied (Frénay, B., & Verleysen, 2013; Frénay and Kabán, 2014). Label noise processes are defined based on the nature of the randomness of the process in question. The three types of noisy labels are -

(1) Noisy completely at random (NCAR) labels, where labels are flipped completely independent of features and class label,
(2) Noisy at random (NAR) labels, where labels are flipped independent of features but dependent on class label,
(3) Noisy not at random (NNAR) labels, where labels are flipped depending on features and class label.

These label noise models are equally highly apt at expressing label noise processes for classification on tabular data and image data. In image classification, each image is assigned a single label; though the feature is more complex, the target is still a single label and therefore the non-dense label noise models are sufficient in describing the noise processes. However, for dense prediction, tasks the notation and process models for label noise need extension. We have formulated label noise models for our image segmentation task by extending the label noise models presented by Frénay, B., & Verleysen, (2013) and design according to pixel-wise and dense dependencies. Dense label noise models can represent complex non-linear and fully-connected statistical dependencies between the image tensors and label tensors. Fig. 2 shows the conceptual differences between the label generation process for the general classification, image classification, and dense prediction.

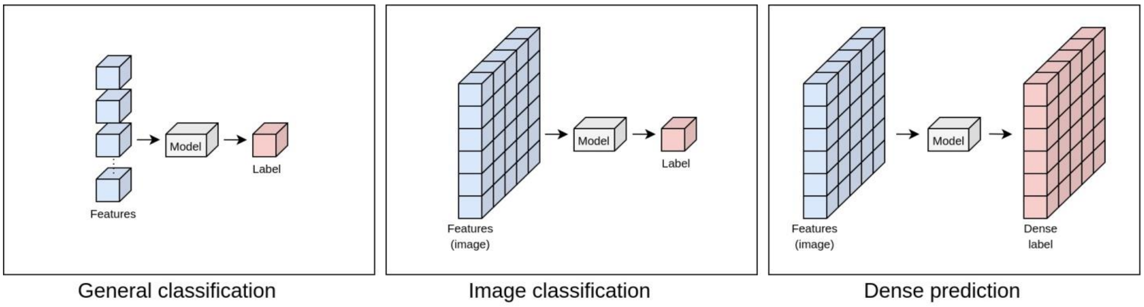


Figure 2. Differences among general classification, image classification and dense prediction

Given an observed noisy dense label $\widetilde{\boldsymbol{Y}} \in \{0,1\}^{n_h \times n_w}$ and its corresponding true clean dense label $\boldsymbol{Y} \in \{0,1\}^{n_h \times n_w}$, where height and width of image tile is n_h and n_w respectively. Indexing n_h by i and indexing n_w by j , $\boldsymbol{Y}_{i,j}$ represents the pixel in i-th row and j-th column of a label tile, $\widetilde{\boldsymbol{Y}}_{i,j}$is considered to be noisy if $\widetilde{\boldsymbol{Y}}_{i,j} \neq \boldsymbol{Y}_{i,j}$. We extend the binary variable random in Frénay and Verleysen (2013) indicating presence of label noise, to dense prediction settings. We define the *error matrix* $\boldsymbol{E} \in \{0,1\}^{n_h \times n_w}$ as the matrix indicating positions of pixels with label noise. Thus, $\boldsymbol{E}_{i,j} = 1$ when $\widetilde{\boldsymbol{Y}}_{i,j} \neq \boldsymbol{Y}_{i,j}$ and $\boldsymbol{E}_{i,j} = 0$ if $\widetilde{\boldsymbol{Y}}_{i,j} = \boldsymbol{Y}_{i,j}$. For binary labels, if the current observed pixel label $\widetilde{\boldsymbol{Y}}_{i,j}$ and its labeling error presence $\boldsymbol{E}_{i,j}$ is known, the true label $\boldsymbol{Y}_{i,j}$ can directly be computed by flipping the observed label when the pixel label in question is deemed to be noisy.

Each element $\boldsymbol{E}_{i,j}$ is a binary random variable indicating if $\boldsymbol{Y}_{i,j}$ is to be noised or not. The relationship among $\boldsymbol{Y}$, $\widetilde{\boldsymbol{Y}}$ and $\boldsymbol{E}$ in matrix form can be defined as

$$\boldsymbol{Y} = \left|\widetilde{\boldsymbol{Y}} - \boldsymbol{E}\right| \tag{1}$$

All operations in Eq. (1) are element-wise matrix operations. Table 1 confirms Eq. (1) and shows the different cases that may arise from combinations of $\boldsymbol{Y}_{i,j}$ and $\widetilde{\boldsymbol{Y}}_{i,j}$. When the true label and observed label are the same (row no. 1 and 2 in Table 1), label noise is absent; when the true label and observed label are not equal (row no. 3 and 4 in Table 1), label noise is present. Given knowledge on the observed noisy label and error matrix, the clean label can directly be computed using Eq. (1).

Table 1. The four possible cases arising from combinations of $\boldsymbol{Y}_{i,j}$ and $\widetilde{\boldsymbol{Y}}_{i,j}$

No	**Case**	**Label noise**	$\boldsymbol{Y}_{i,j}$	$\widetilde{\boldsymbol{Y}}_{i,j}$	$\boldsymbol{E}^{+}_{i,j}$	$\boldsymbol{E}^{-}_{i,j}$	$\boldsymbol{E}_{i,j}$	$\left\|\widetilde{\boldsymbol{Y}} - \boldsymbol{E}\right\|$
1	True negative observed pixel label	No	0	0	0	0	0	0
2	True positive observed pixel label	No	1	1	0	0	0	1
3	False positive observed pixel label	Yes	0	1	1	0	1	0
4	False negative observed pixel label	Yes	1	0	0	1	1	1

The error matrix is the absolute difference between the true and observed labels

$$\boldsymbol{E} = \left|\boldsymbol{Y} - \widetilde{\boldsymbol{Y}}\right| = \left|\widetilde{\boldsymbol{Y}} - \boldsymbol{Y}\right| \tag{2}$$

Let, the error matrix denoting *false positive* observed labels be $\boldsymbol{E}^{+} \in \{0,1\}^{n_h \times n_w}$ and the error matrix denoting *false negative* observed be $\boldsymbol{E}^{-} \in \{0,1\}^{n_h \times n_w}$. Thus, $\boldsymbol{E}$ is the element-wise logical 'or' (expressed as summation) of $\boldsymbol{E}^{+}$ and $\boldsymbol{E}^{-}$ in matrix form,

$$\boldsymbol{E} = \boldsymbol{E}^{+} + \boldsymbol{E}^{-} \tag{3}$$

Fig. 3 shows an example of how label noise arises from disagreements between the true label and observed label, displaying that a few positive pixel labels were missed and a few true negative pixel labels were labeled as positives.

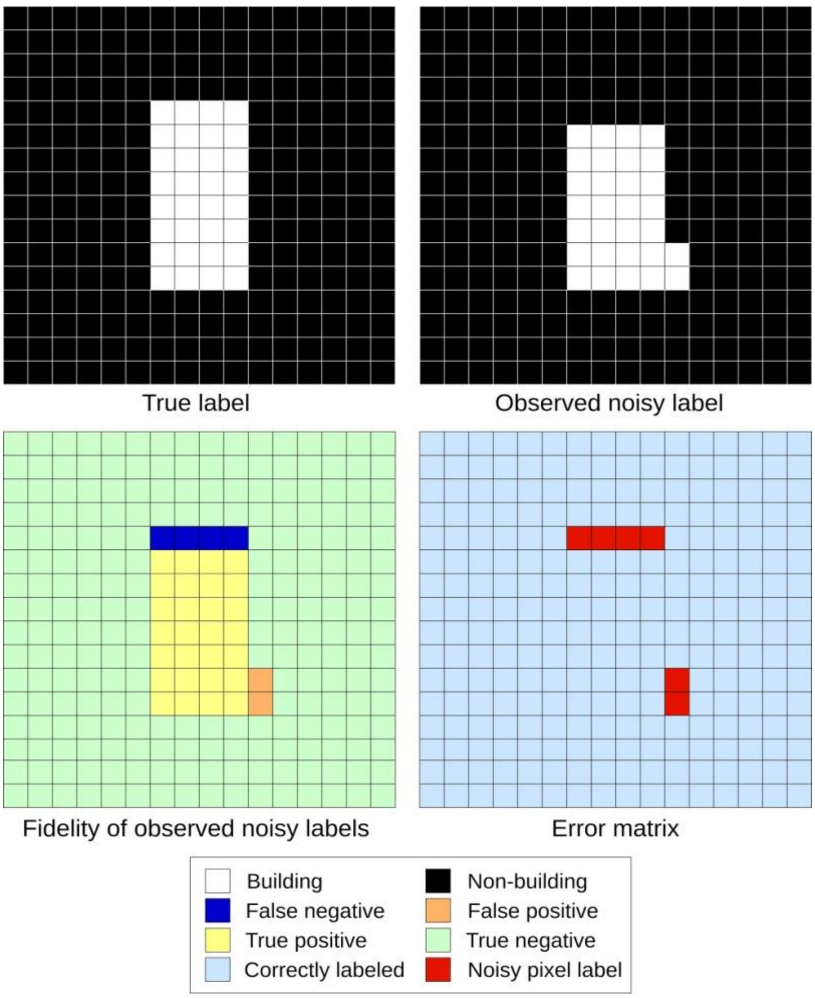


Figure 3. Example of how observed noisy dense labels differ from their corresponding true dense labels. A 16 x 16 pixel image is used for demonstration. The error matrix E is shown in the bottom right subfigure, indicating positions of noisy pixel labels.

The label noise process involves the *corruption* of clean labels (Fig. 4). In general learning schemes for building extraction, it is assumed that the observed labels are clean and are directly used for learning/evaluation (Fig. 4(a)). However, acknowledgement of label noise assumes the intermediary distribution of clean labels over the images to be the clean labels and models the label noise process as the distribution of observed noisy labels over the true clean labels (Fig. 4(b)), which means that when label noise is present, the ground truth clean labels are unobserved

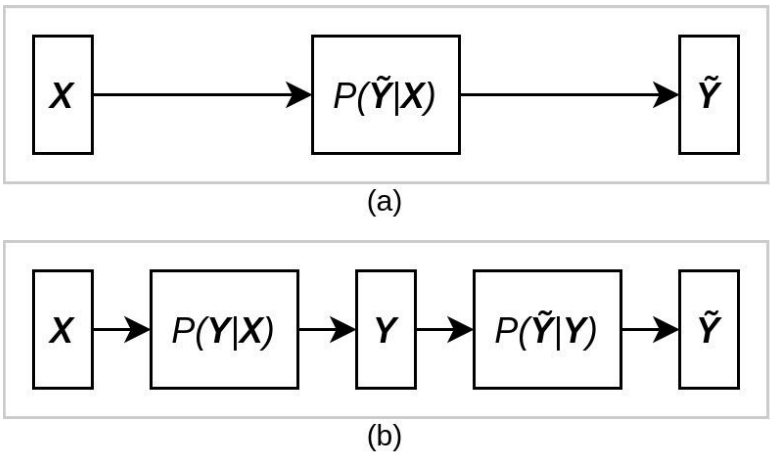


Figure 4. Observed label generation processes (a) Modeled without noise-free labels (b) Modeled through noise-free labels

Having defined the important concepts i.e. $\boldsymbol{Y}$, $\widetilde{\boldsymbol{Y}}$ and $\boldsymbol{E}$, for modeling dense label noise processes, we move on to define the statistical dependencies for learning dense prediction (Fig. 5). There are two main models -

- *Pixel-wise models:* perform pixel classification using features from only the corresponding input pixels (Fig. 5). Therefore, changing tile sizes does not have significant effects if the same pixels are provided for training and testing because only pixel-wise mappings are learned; features from neighboring pixels are not considered. Without context, the rooftop of a building and a road may appear identical to the model. However, learning pixel-wise mapping is common in non-deep learning approaches to building extraction. Given, the input tensor $\boldsymbol{T}^a$ and its dependent output tensor $\boldsymbol{T}^b$, the pixel wise models learn,

$$P(\boldsymbol{T}^b_{i,j}|\boldsymbol{T}^a_{i,j}) \tag{4}$$

- *Dense models:* generates labels for pixels using features from all pixels of the input tensor (Fig. 5). The model estimates each $P(\boldsymbol{T}^b_{i,j}|\boldsymbol{T}^a)$ and then uses the product chain rule to learn $P(\boldsymbol{T}^b|\boldsymbol{T}^a)$,

$$P(\boldsymbol{T}^b|\boldsymbol{T}^a) = \prod_{i=1}^{n_h}\prod_{j=1}^{n_w} P(\boldsymbol{T}^b_{i,j}|\boldsymbol{T}^a) \tag{5}$$

As Fig. 5 shows, we represent fully connected dense mappings using a red full red arrow with continuous line and pixel wise mappings using a blue half-arrow with dotted line.

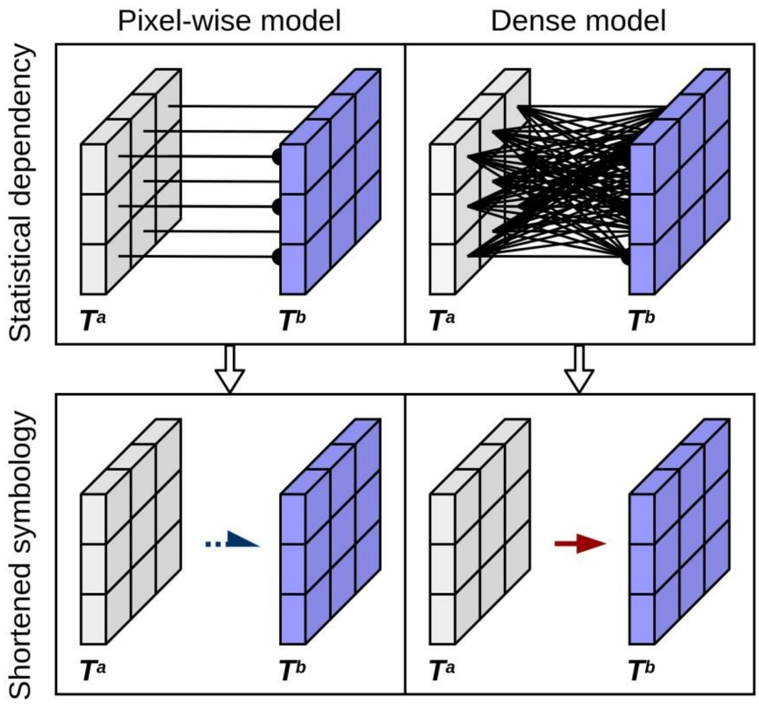


Figure 5. Shortened symbology of statistical dependencies considered in pixel wise models and dense models. In the pixel-wise model, each $\boldsymbol{T}^b_{i,j}$ is only dependent on $\boldsymbol{T}^a_{i,j}$. In the dense model, each $\boldsymbol{T}^b_{i,j}$ is dependent on the the entire matrix $\boldsymbol{T}^a$ indicating fully connectedness.

Taxonomy of dense label noise models

The three types of label noise in Frénay and Verleysen (2013) are categorized according to randomness. We refer to this approach as taxonomy characterized by randomness. However, in the context of dense prediction, structure (spatial information) in dense labels also plays a role in label noise processes. We define the taxonomy of dense label noise models. Given the two types of mapping models (pixel-wise and dense) and the three types of stochasticity defined label noise processes (NCAR, NAR and NNAR), there are six possible models (Fig. 6).

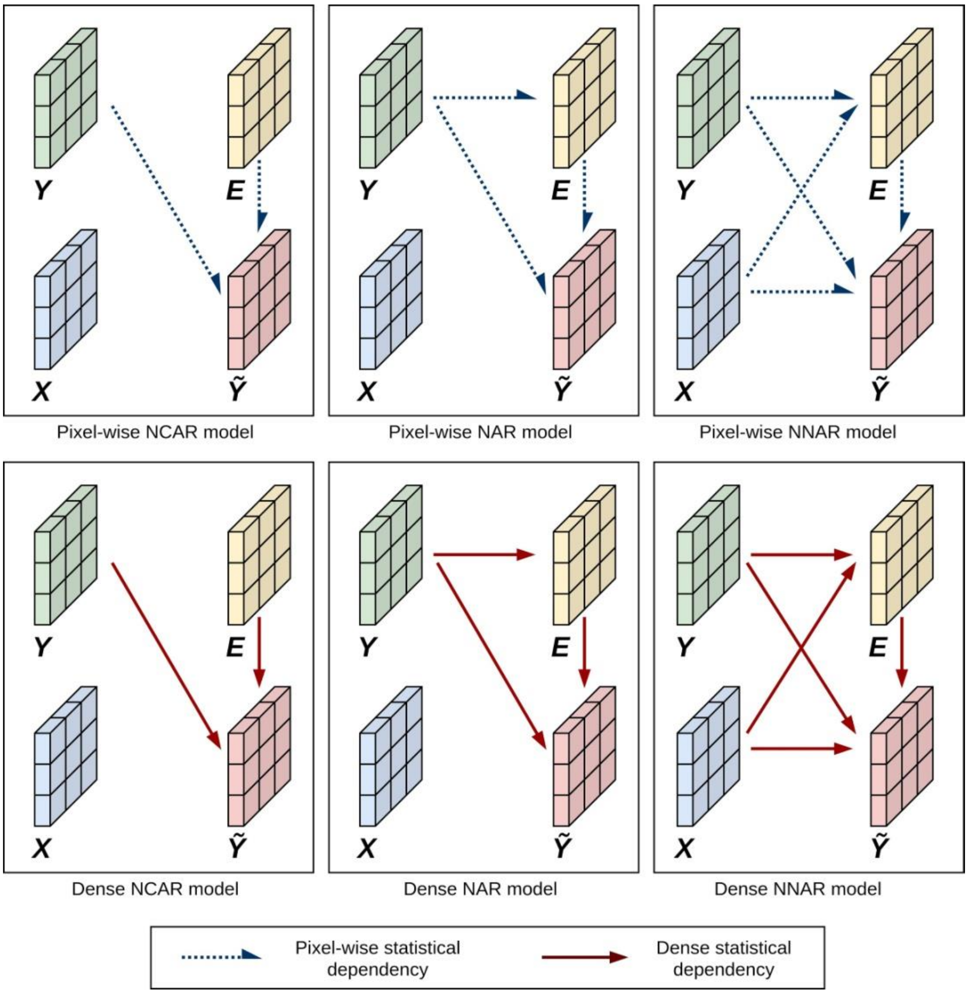


Figure 6. Statistical dependencies of different types of pixel based and dense label noise models. The dependency between $X$and $Y$are not shown for brevity.

(1) *Pixel-wise NCAR model*: NCAR models are class independent, therefore the only noise parameters for a pixel-wise NCAR model would be the *probability of error* $p_e = P(\widetilde{\boldsymbol{Y}}_{i,j} \neq \boldsymbol{Y}_{i,j})$. It is important to note that p_e is constant for all pixels, and therefore NCAR models cannot model non-uniform label noise. All $\boldsymbol{E}_{i,j}$ would have the same values because the probability of a pixel being noisy is constant and not dependent on any variables. The error matrix $\boldsymbol{E}$ is completely independent (pixel-wise NCAR model in Fig. 6). For binary classification (which is our case for the pixel-wise models) having $p_e = 1/2$ would render the labels useless and inadequate to learn from (Angluin

and Laird, 1988). Furthermore, since NCAR models are class independent, asymmetric noise cannot be modeled as well. NCAR models assume that labels of all classes have equal chances of being observed as noisy labels. In real world settings, this is rarely the case. For example, in building extraction tasks, the positive class is much more prone to label noise. Furthermore, the positive class is also the minority class in most imbalanced building extraction datasets.

(2) *Pixel-wise NAR model:* NAR models are able to model asymmetric and non-uniform label noise processes. Each $\boldsymbol{E}_{i,j}$ is dependent on each $\boldsymbol{Y}_{i,j}$, which in turn affects each $\widetilde{\boldsymbol{Y}}_{i,j}$ (pixel-wise NCAR model in Fig. 6). The probability of a specific label being observed as another label is modelled using the transition matrix (Lawrence and Schölkopf, 2001; Pérez et al., 2007). We define the *transition matrix* for noisy dense binary labels as

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_{0,0} & \gamma_{0,1} \\ \gamma_{1,0} & \gamma_{1,1} \end{bmatrix}$$

$$= \begin{bmatrix} P(\widetilde{\boldsymbol{Y}}_{i,j} = 0|\boldsymbol{Y}_{i,j} = 0) & P(\widetilde{\boldsymbol{Y}}_{i,j} = 0|\boldsymbol{Y}_{i,j} = 1) \\ P(\widetilde{\boldsymbol{Y}}_{i,j} = 1|\boldsymbol{Y}_{i,j} = 0) & P(\widetilde{\boldsymbol{Y}}_{i,j} = 1|\boldsymbol{Y}_{i,j} = 1) \end{bmatrix} \quad (6)$$

The conditional probabilities in Eq. (6) can be estimated from the observed and corresponding clean labels. It is important to note that, the transition matrix is the same for all $\widetilde{\boldsymbol{Y}}_{i,j}$ (and hence for all $\boldsymbol{Y}_{i,j}$). For uniform noise in dense binary labels, the transition matrix becomes

$$\boldsymbol{\gamma} = \begin{bmatrix} 1 - p_e & p_e \\ p_e & 1 - p_e \end{bmatrix} \tag{7}$$

(3) *Pixel-wise NNAR model*: In the case of NNAR models, the error matrix $\boldsymbol{E}$ is dependent on the features as well (pixel-wise NNAR model in Fig. 6). The observed pixel label $\widetilde{\boldsymbol{Y}}_{i,j}$ is dependent on $\boldsymbol{E}_{i,j}$ and $\boldsymbol{Y}_{i,j}$; if $\boldsymbol{E}_{i,j} = 1$, $\boldsymbol{Y}_{i,j}$ is flipped to get $\widetilde{\boldsymbol{Y}}_{i,j}$, otherwise $\widetilde{\boldsymbol{Y}}_{i,j} = \boldsymbol{Y}_{i,j}$. The probability of error is a function of the pixel-wise feature and pixel-wise true label,

$$p_e\left(\boldsymbol{X}_{i,j}, \boldsymbol{Y}_{i,j}\right) = P(\boldsymbol{E}_{i,j} = 1 | \boldsymbol{X}_{i,j} = x, \boldsymbol{Y}_{i,j} = y) \tag{8}$$

(4) *Dense NCAR model:* In the dense NCAR model, every $\widetilde{\boldsymbol{Y}}_{i,j}$ is affected by the entire error matrix $\boldsymbol{E}$, and not just $\boldsymbol{E}_{i,j}$ (which is the case for the pixel-wise NCAR model). Spatial information about label noise in terms of context (as opposed to pixel-based information) can be modeled. Every $\boldsymbol{E}_{i,j}$ need not be constant; however, they are still completely independent (of each other and of any other random variable) and thus completely random (dense NCAR model in Fig. 4).

(5) *Dense NAR model*: The dense NAR model allows modeling asymmetric dense label noise, which is not possible using the dense NCAR model. Unlike the pixel-wise NAR model, the transition matrix for each $\widetilde{\boldsymbol{Y}}_{i,j}$ can be distinct and independent of each other. The transition matrix for $\widetilde{\boldsymbol{Y}}_{i,j}$ in a dense NAR model can be defined as

$$\boldsymbol{\gamma}^{(i,j)} = \begin{bmatrix} \gamma_{0,0}^{(i,j)} & \gamma_{0,1}^{(i,j)} \\ \gamma_{1,0}^{(i,j)} & \gamma_{1,1}^{(i,j)} \end{bmatrix} \tag{9}$$

The error matrix $\boldsymbol{E}$ is directly dependent on the true dense label $\boldsymbol{Y}$ (dense NAR model in Fig. 6), but independent of the dense features $\boldsymbol{X}$.

(6) *Dense NNAR model:* In the dense NNAR model all pixels from the image affect the probabilities of label noise in certain observed pixels (dense NNAR model in Fig. 6). Every $\boldsymbol{E}_{i,j}$ is affected by the entire image tensor $\boldsymbol{X}$, and every $\widetilde{\boldsymbol{Y}}_{i,j}$ is affected by the entire error matrix $\boldsymbol{E}$. The error matrix can be estimated based on the observed dense label $\widetilde{\boldsymbol{Y}}$ and dense feature tensor $\boldsymbol{X}$. We essentially model the conditional distribution of the error matrix $\boldsymbol{E}$, given the feature tensor $\boldsymbol{X}$ and the observed dense label $\widetilde{\boldsymbol{Y}}$ (Eq. (10)). This estimated error matrix can then be used for generating the true labels using Eq. (1).

$$P\left(\boldsymbol{E} \middle| \widetilde{\boldsymbol{Y}}, \boldsymbol{X}\right) = \prod_{i=1}^{n_h} \prod_{j=1}^{n_w} P(\boldsymbol{E}_{i,j} | \widetilde{\boldsymbol{Y}}, \boldsymbol{X}) \tag{10}$$

Materials and methods

Data

The dataset consists of 258 large 512x512 ultra-high-resolution aerial image tiles over the Kutupalong mega camp collected by the United Nations International Organization for Migration on September 17, 2018. Kutupalong is the largest of the camps, comprised of several sub-camps, situated in the south-eastern border region of Bangladesh which acted as the corridor for the Rohingya refugees migrating from

Myanmar. For our case, $n_h = n_w = 512$. The observed noisy labels are collected from OpenStreetMap. The true clean labels are obtained by relabeling performed by the authors. The dataset is randomly split in half for denoting training and testing data. Images have three channels/bands — Red, Green and Blue — with a spatial resolution of 10 cm. These images have very high data quality i.e. without cloud or shadow cover being collected by low flying unmanned aerial vehicles (UAVs) and capture fine-grained details of the physical environments where the buildings are located. The general error matrices are computed using Eq. (2), whereas the FP and FN error matrices are computed without taking the absolute value, rather using the signed/un-signedness of the difference matrix. Our dataset is relatively smaller than most commonly used datasets for building extraction (such as Massachusetts, Potsdam and Vaihingen datasets), this is because we have had to re-label all of our training and test data by hand for obtaining the noise-free true clean labels, which is very time-consuming. Moreover, datasets for semantic segmentation/dense prediction with the corresponding observed labels (with real-world label noise) and counterpart clean labels are virtually non-existent. Our dataset is unique in that aspect, since, having access to the observed noisy labels and clean labels is crucial for obtaining ground truth error matrices (Eq. (1)). It is important to note that the error matrices are only required for pretraining the dense label noise prediction models, during testing/evaluation the models directly output building maps corrected by error matrices.

Model frameworks

The true clean dense label is solely dependent on the feature tensor in all six noise models (caption of Fig. 6). The features (from satellite/aerial images), used for approximating true labels, can be compared to the observed noisy label to obtain the error matrix; the features have an important role in determining the observed label.

Therefore, the dense NNAR model is most suitable for expressing commonly observed registration errors. Currently, deep learning is the state-of-the-art system for automated building extraction (Vakalopoulou et al., 2015; Huang et al., 2016; Chen et al., 2017; Yuan, 2017; Yang et al., 2018; Ji et al., 2018; Xu et al., 2018; Shrestha and Vanneschi, 2018; Boonpook et al., 2021; Sun et al., 2021). Fig 7(a) and 7(b) show the generally used learning systems for deep learning-based building extraction i.e. with clean labels (Fig. 7(a)) and with noisy labels (Fig. 7(b)). We propose two new models for automated building extraction, and consequently, two novel network architectures, where error matrices are approximated as an intermediate step (Fig 7(c) and Fig. 7(d)). As discussed later, we draw from the dense NNAR model in modelling our learning frameworks. The formulated dense noise models ultimately determine the architecture of the neural networks. The base network in Fig. 7(a) represents the statistical dependency between the feature and label tensors in the dense NNAR model (Fig. 6). Similarly, the error matrix network in Fig. 7(a) represents the statistical dependency between the feature and error matrix tensors in the dense NNAR model (Fig. 6). We elaborate on the model frameworks, network architectures, learning and evaluation approaches.

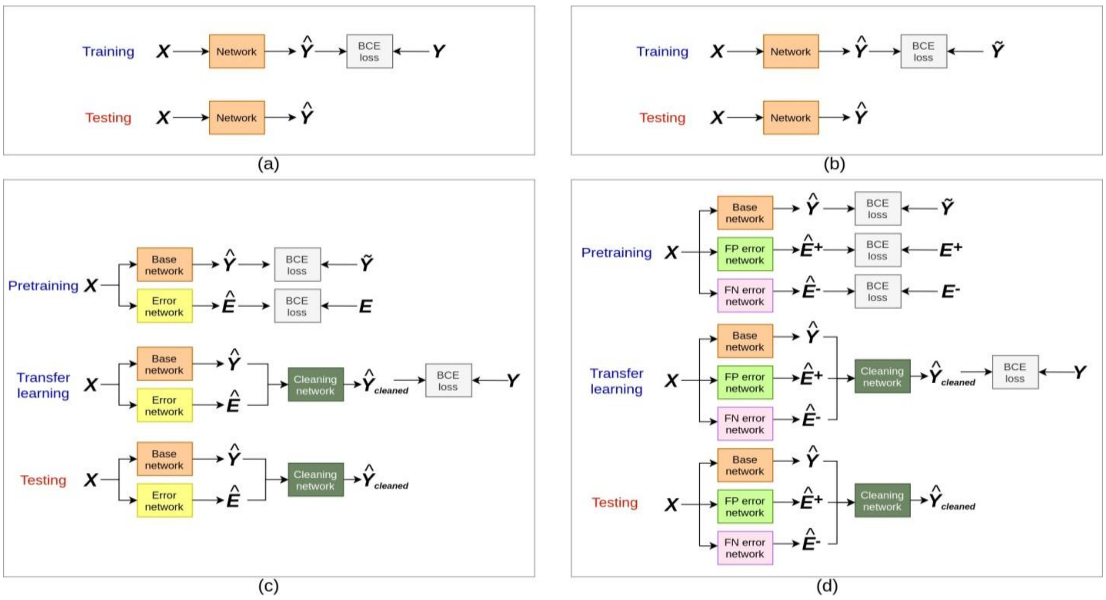


Figure 7. Training and testing approaches (a) With clean labels - control, CL model (b) With noisy labels - NL model (c) With intermediate error matrix approximation - I-EM model (d) With intermediate FP and FN error matrices approximation - I-FPFN-EM model; BCE - binary cross entropy; FP - false positive, FN - false negative

Intermediate error matrix (I-EM) model

The first proposed intermediate error matrix (I-EM) model approximates error matrices as an intermediate step of approximating building/non-building predictions. The noisy observed labels are learned by the base network in Fig. 7(c) approximated as the mean of the distribution in Eq. (11). The noisy observed labels are learned by the error network in Fig. 7(c) approximated as the mean of the distribution in Eq. (12). Finally, the outputs from the error matrix (EM) model and the observed label model are used together by the cleaning network in Fig. 7(c) to learn noise free label approximation in Eq. (13). Viewing the model framework from an end-to-end fashion in terms of testing indicates (Testing in Fig. 7(c)) in Eq. (14).

$$P\left(\widetilde{\boldsymbol{Y}}\middle|\boldsymbol{X}\right) = \prod_{i=1}^{n_h}\prod_{j=1}^{n_w} P(\widetilde{\boldsymbol{Y}}_{i,j}|\boldsymbol{X}) \tag{11}$$

$$P(\boldsymbol{E}|\boldsymbol{X}) = \prod_{i=1}^{n_h}\prod_{j=1}^{n_w} P(\boldsymbol{E}_{i,j}|\boldsymbol{X}) \tag{12}$$

$$P\left(\boldsymbol{Y}\middle|\widetilde{\boldsymbol{Y}},\boldsymbol{E}\right)=\prod_{i=1}^{n_h}\prod_{j=1}^{n_w}P(\boldsymbol{Y}_{i,j}|\widetilde{\boldsymbol{Y}},\boldsymbol{E}) \tag{13}$$

$$P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{E})=\prod_{i=1}^{n_h}\prod_{j=1}^{n_w}P(\boldsymbol{Y}_{i,j}|\boldsymbol{X},\boldsymbol{E}) \tag{14}$$

Intermediate FP and FN error matrix (I-FPFN-EM) model

The second proposed intermediate FP and FN error matrix (I-FPFN-EM) model approximates the FP and FN error matrices separately as an intermediate step of approximating building/non-building predictions. The noisy observed labels are learned by the base network in Fig. 7(d) approximated as the mean of the distribution in Eq. (11). The FP (false positive) error matrix is learned by the FP error network in Fig. 7(d) approximated as the mean of the distribution in Eq. (15). The FN (false negative) error matrix is learned by the FNM error network in Fig. 7(d) approximated as the mean of the distribution in Eq. (16). Finally, the outputs from the FP and FN error matrix models, and the observed label model are used together by the cleaning network in Fig. 7(d) to learn noise free label approximation in Eq. (17). We refer to the FP error matrix model as the FP-EM model and the FN error matrix model as the FN-EM model. Viewing the model framework from an end-to-end fashion in terms of testing indicates (Testing in Fig. 7(d)) in Eq. (18).

$$P(\boldsymbol{E}^{+}|\boldsymbol{X})=\prod_{i=1}^{n_h}\prod_{j=1}^{n_w}P(\boldsymbol{E}_{i,j}^{+}|\boldsymbol{X}) \tag{15}$$

$$P(\boldsymbol{E}^{-}|\boldsymbol{X}) = \prod_{i=1}^{n_h}\prod_{j=1}^{n_w} P(\boldsymbol{E}^{-}_{i,j}|\boldsymbol{X}) \tag{16}$$

$$P\left(\boldsymbol{Y}\middle|\widetilde{\boldsymbol{Y}}, \boldsymbol{E}^{+}, \boldsymbol{E}^{-}\right) = \prod_{i=1}^{n_h}\prod_{j=1}^{n_w} P(\boldsymbol{Y}_{i,j}|\widetilde{\boldsymbol{Y}}, \boldsymbol{E}^{+}, \boldsymbol{E}^{-}) \tag{17}$$

$$P(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{E}^{+}, \boldsymbol{E}^{-}) = \prod_{i=1}^{n_h}\prod_{j=1}^{n_w} P(\boldsymbol{Y}_{i,j}|\boldsymbol{X}, \boldsymbol{E}^{+}, \boldsymbol{E}^{-}) \tag{18}$$

Network architectures

Each intermediate network has four downsampling blocks and four upsampling blocks. We use vanilla U-Nets with approximately 0.5 million parameters for intermediate learning steps. The U-Net/autoencoder architecture is common for building extraction tasks (Wang et al., 2020; Guo et al., 2020). The use of step-wise concatenation of models has been employed for building extraction (Shao et al., 2020). Each downsampling block has two convolutional layers punctuated by a single dropout layer, which is then downsampled to half the output row and column size using max pooling. Each upsampling block also has two convolutional layers punctuated by a single dropout layer, which is then upsampled to double the output row and column size using interpolation. We use the binary cross entropy loss function as it is commonly used for most binary building extraction tasks (Ahmed et al., 2020). For the I-EM model (Fig. 8(a)) the outputs of the base network and error network are concatenated and fed to the cleaning network. For the I-FPFN-EM network the outputs of the base network, FP error matrix network and FN error matrix network are all fed into the cleaning network. Please note that intermediate predictions of observed labels and error matrices (general, FP and FN) are in the form of soft pixel level labels i.e. they are not converted to hard labels based on threshold values. The I-EM model and I-FPFN-EM models have approximately 1.5 million and 2 million parameters respectively. S1 details the network architecture for NL, CL, EM, FP-EM and FN-EM models, Fig. S2 and Fig. S3 in supplementary material contains the detailed network architectures of the I-EM and I-FPFN-EM model respectively.

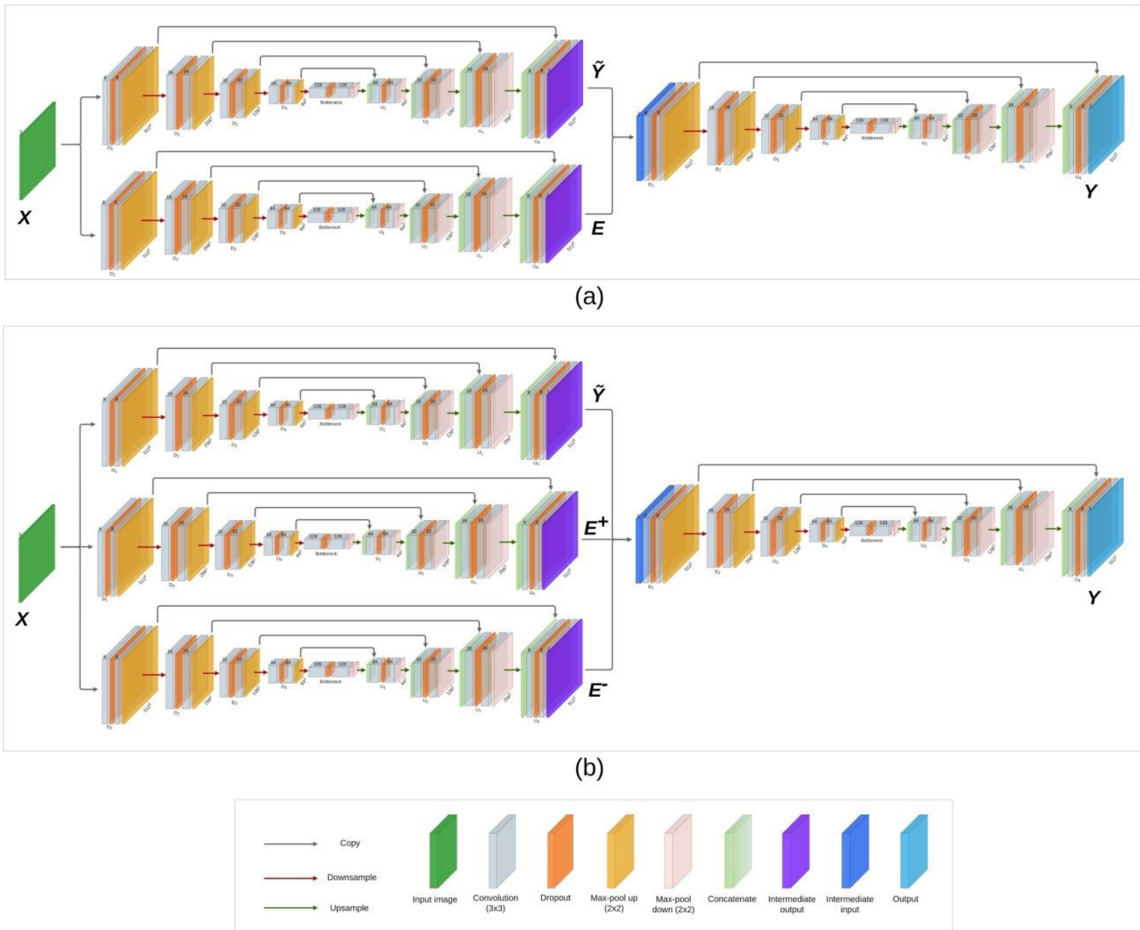


Figure 8. Proposed network architectures for building extraction under label noise (a) I-EM model (b) I-FPFN-EM model

Learning

The I-EM model and I-FPFN-EM model are trained in two steps.

- *Step 1 - Pre-training*: For learning the parameters of the base and error networks. Individual auto-encoders with skip connections are trained. For the I-EM model, the base network is trained using the images $\boldsymbol{X}$ as features and $\widetilde{\boldsymbol{Y}}$ as targets, the error network is trained using the images $\boldsymbol{X}$ as features and $\boldsymbol{E}$ as targets. For the I-FPFN-EM model, the base network is also trained using the images $\boldsymbol{X}$ as features and $\widetilde{\boldsymbol{Y}}$ as targets, the FP error network is trained using the images $\boldsymbol{X}$ as features and $\boldsymbol{E}^{+}$ as targets, and the FN error network is trained using the images $\boldsymbol{X}$ as features and $\boldsymbol{E}^{-}$ as targets.

- *Step 2 - Transfer learning*: After the base networks and error networks (general for I-EM; FP and FN for I-FPFN-EM) are trained, their outputs are concatenated and fed into the cleaning networks. In order to train the cleaning network, the layers in the base and error networks are frozen i.e. they are set as non-trainable. In this second step of training, the entire network is trained in an end-to-end fashion against clean labels.

The baseline CL model and NL model both have approximately 0.5 million parameters. The I-EM model and I-FPFN-EM models have approximately 1.5 million and 1.5 million parameters respectively. This larger number of parameters are due to the error matrix networks and the cleaning networks used in the I-EM model and the I-FPFN-EM models. The general error matrix sub-model in the I-EM model, and each of the false positive error matrix model and the false negative error matrix models all have approximately 0.5 million parameters. The time complexity of the I-EM model and I-FPFN-EM model are also increased proportional to the increase of number of parameters with respect to the CL and NL models. The total time needed for training the sub-models of the I-EM model is triple that of the CL or NL models, and the total time needed for training the I-FNFN-EM models is quadruple that of the CL or NL models.

Method comparison

In order to assess the qualitative and quantitative advantages/disadvantages of our two proposed models, we also compare against generally used model frameworks for automated building extraction. We compare four different deep learning-based building segmentation models,

(1) Noisy label (*NL*) model (Ahmed et al., 2020): Dense building extraction with noisy labels.
(2) Clean label (*CL*) model (Ahmed et al., 2020): Dense building extraction with clean labels (control).
(3) *I-EM model*: The first proposed model described above.
(4) *I-FPFN-EM* model: The second proposed model described above.

Other than the CL and NL models in Ahmed et al., (2020), no other study presents dataset/methods for dense prediction of label noise using clean and noisy labels with real world noise. The threshold value determines the boundary value and consequently the binary class label of each pixel. We vary the threshold for each model with low (0.25), medium (0.5) and high (0.75) values to convert the soft labels (between 0 and 1 inclusive) to hard labels (0 or 1).

Performance evaluation metrics

We calculate the total number of true positives (TP), true negatives (TN), false positive (FP) and false negative (FN) predictions on the approximately 33 million pixels of testing data. Concurring to most building extraction scenarios, our dataset is also quite imbalanced, being negative heavy. Therefore, we calculate the precision (Eq. (19), recall (Eq. (20)), F1-score (Eq. (21) and Intersection-over-Union (*IoU*) (Eq. (22)).

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \tag{21}$$

$$IoU = \frac{TP}{TP + FP + FN} \tag{22}$$

Results and discussion

Quantitative evaluation of performance

The CL model provides the control/baseline against which we compare our two proposed models since it represents the ideal scenario when the investigator has access to both images and clean labels. Our I-FPFN-EM model at 0.5 medium threshold (row no. 11 in Table 2) has the highest *IoU* score (0.78514), which provides a gain of 2.74% over the traditional CL model trained on clean labels (0.75768) and a gain of 25.65% over the observed noisy labels with *IoU* score of 0.52857. Similarly, our I-FPFN-EM model at 0.5 threshold has the highest F1-score (0.87964), which provides a gain of

1.75% over the traditional model trained on clean labels with an F1-score of 0.86214, and gain of 18.8% over the observed noisy labels with an F1-score of 0.69159. Compared to the idealistic CL model, our I-FPFN-EM model has a better F1-score and *IoU* score for high threshold value (0.75) as well, and has comparable/nearly identical performance for low threshold value (0.25). At a threshold value of 0.75, the I-FPFN-EM model (row no. 12 in Table 2) has an F1-score of 0.86009 which is 3.45% higher than the F1-score of the CL model (0.8255) at a threshold value of 0.75. The I-FPFN-EM model at a threshold value of 0.75, achieves an *IoU* score of 0.75453, providing a gain of 5.16% over the CL model with an *IoU* score of 0.70285, at a threshold value of 0.75. Our I-FPFN-EM model provides better performance over traditional methods, for the general threshold of 0.5 and the high threshold of 0.75.

The I-EM has slightly poorer/comparable performance to the CL model. This indicates the importance of differentiating FP and FN error matrices as features, instead of approximating an intermediate general error matrix, since that is the primary conceptual difference between the I-EM model and I-FPFN-EM model. A lower threshold means higher recall and lower precision. A higher threshold means higher precision and lower recall. The threshold value determines the precision recall trade-off. However, both the I-EM and I-FPFN-EM models have much higher recall and slightly lower precision for corresponding threshold values when compared to the CL model. In our case of highly imbalanced data, higher recall is preferred over higher precision.

Table 2. Performance of the four compared models for building extraction under label noise and the fidelity of observed labels

No.	**Model**	**Threshold**	**Precision**	**Recall**	**F1-score**	**IoU**

1	NL	0.25	0.79584	0.82862	0.8119	0.68336
2		0.50	0.91337	0.56184	0.69572	0.53342
3		0.75	0.98292	0.0948	0.17291	0.09464
4	CL	0.25	0.79586	0.91111	0.84959	0.73851
5		0.50	0.88502	0.84041	0.86214	0.75768
6		0.75	0.93973	0.73603	0.8255	0.70285
7	I-EM	0.25	0.74541	0.93536	0.82965	0.70889
8		0.50	0.84473	0.85928	0.85194	0.74207
9		0.75	0.89968	0.76947	0.8295	0.70867
10	**I-FPFN-EM**	0.25	0.76109	0.94634	0.84366	0.7296
11		**0.50**	0.86551	0.89424	**0.87964**	**0.78514**
12		0.75	0.92819	0.80131	0.86009	0.75453
13	OBSERVED	-	0.82165	0.59708	0.69159	0.52857

Separated error matrices in the form of FP error matrix and FN error matrix is crucial to surpassing the baseline CL model performance, as our I-EM model has significantly poorer quantitative performance compared to the I-FPFN-EM model. Comparing the I-EM model and the I-FPFN-EM model performances at the three threshold values, the I-FPFN-EM model provides an F1-score increase of 1.4% (0.84366 compared to 0.82965) and *IoU* score increase of 2.071% (0.7296 compared to 0.70889) at a threshold value of 0.25, F1-score increase of 2.77% (0.87964 compared to 0.85194) and *IoU* score increase of 4.307% (0.78514 compared to 0.74207) at a threshold value of 0.5 and F1-score increase of 3.059% (0.86009 compared to 0.8295) and *IoU* score increase of 4.586% (0.75453 compared to 0.70867) at a threshold value of 0.75.

The traditional model trained against noisy labels (NL model), quite obviously has the poorest performance of the four tested models (row no. 1-3 in Table 2). At high threshold values (0.75) the NL model (row no. 3 in Table 2) predictions become practically useless, yielding an F1-score of 0.17291 and *IoU* score of 0.09464, whereas the CL, I-EM and I-FPFN-EM model have much better performance at a high threshold value of 0.75. The fidelity of noisy labels is also evaluated against the true clean labels (row no. 13 in Table 2). Though the NL model has the poorest performance among four tested models, predictions from the NL model have higher fidelity than the observed labels with real world noise. This is commonly observed for building extraction under real-world noisy conditions (Ahmed et al., 2020).

Qualitative evaluation

From a qualitative viewpoint, the predictions from the four models seem quite similar prior to intensive inspection and photo-interpretation. We show some examples of predictions on image tiles from the test set (Fig. 9). The CL model predictions (Fig. 9(d)) have the best qualitative properties, followed by the I-FPFN-EM model predictions (Fig. 9(g)) which sometimes suffers from salt and pepper noise (all predictions in Fig. 9 were made at a threshold value of 0.5 and can be remedied using lower threshold values). Particularly, the I-FPFN-EM model predictions and I-EM model predictions (Fig. 9(f)) for buildings with rare colored roofs (orange painted corrugated metal roofs) contain salt and peppering. Rare colored building rooftops can be challenging to learn due to the comparatively small number of examples in the training set. The NL model predictions completely miss out on entire buildings with orange-colored rooftops (Fig. 9(e)). The last row in Fig. 9 shows the issues of one-storied building rooftops being obstructed partly or completely by vegetation. Building rooftops obstructed by trees and vegetation are not easily detected, as the vegetation over the rooftop is easily confused as non-building regions by the models (last row in Fig. (9)). However, for buildings with vegetation on the rooftops, the I-FPFN-EM model provides less peppering and errors compared to even the CL model (last row in Fig. (9))

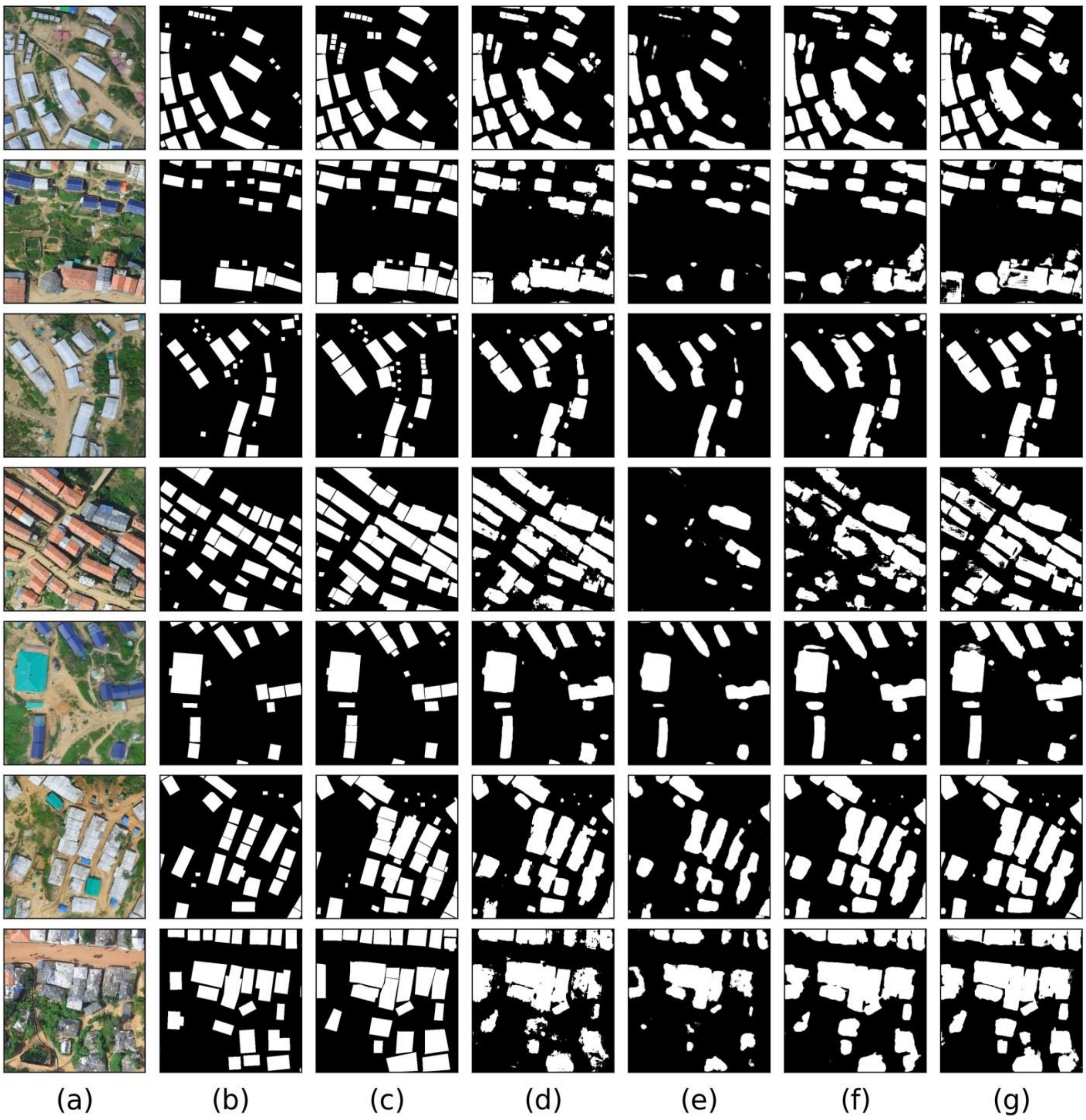


Figure 9. Examples of building predictions made by different models (a) Image (b) Noisy label (c) Clean label (d) Predictions from CL model (e) Predictions from NL model (f) Predictions from I-EM model (g) Predictions from I-FPFN-EM model

Some examples of error matrices predicted during the intermediate step are shown in Fig. 10. The error matrices are sparse, and weakly correlated to the images as the real world label noise can be random at times. However, they can provide insights about location having higher probabilities of being mislabeled. The ground truth FP error matrix is shown in Fig. 10(b) and the predicted FP error matrix is shown in Fig. 10(c). FP pixels are usually pixels adjacent to the clean building label boundary, but falling outside the boundary; this intuition is captured by the FP error matrix model as indicated by the predictions in Fig. 10(c). i.e. the regions adjacent to actual/clean boundaries have higher activations than other regions in the images, and thus have a higher probability of being an observed FP pixel. The predicted FP error matrix (non-thresholded) provides a heat map indicating the probability of each observed positive pixel label actually being true negative pixels.

FN pixels are less sparse than FP pixels since a major source of label noise in building extraction datasets comes from omitted/missed out buildings and shrunk label polygons. Fig. 10(d) shows the actual FN error matrix and Fig. 10(e) shows the predicted by the FN error matrix model. FN pixels are pixels within the clean building boundaries which are observed as non-building in the noisy labels, therefore regions in close proximity to the clean building boundaries but on the inner side have the highest probability of being observed as FN pixels, this is shown in Fig. 10(e). It is interesting to note that all pixels with significantly high FP error matrix activations lie outside and adjacent to the clean building boundaries whereas all pixels with significantly high FN error matrix activations lie inside the clean building boundaries; the modeling intuition is expressed in the qualitative results.

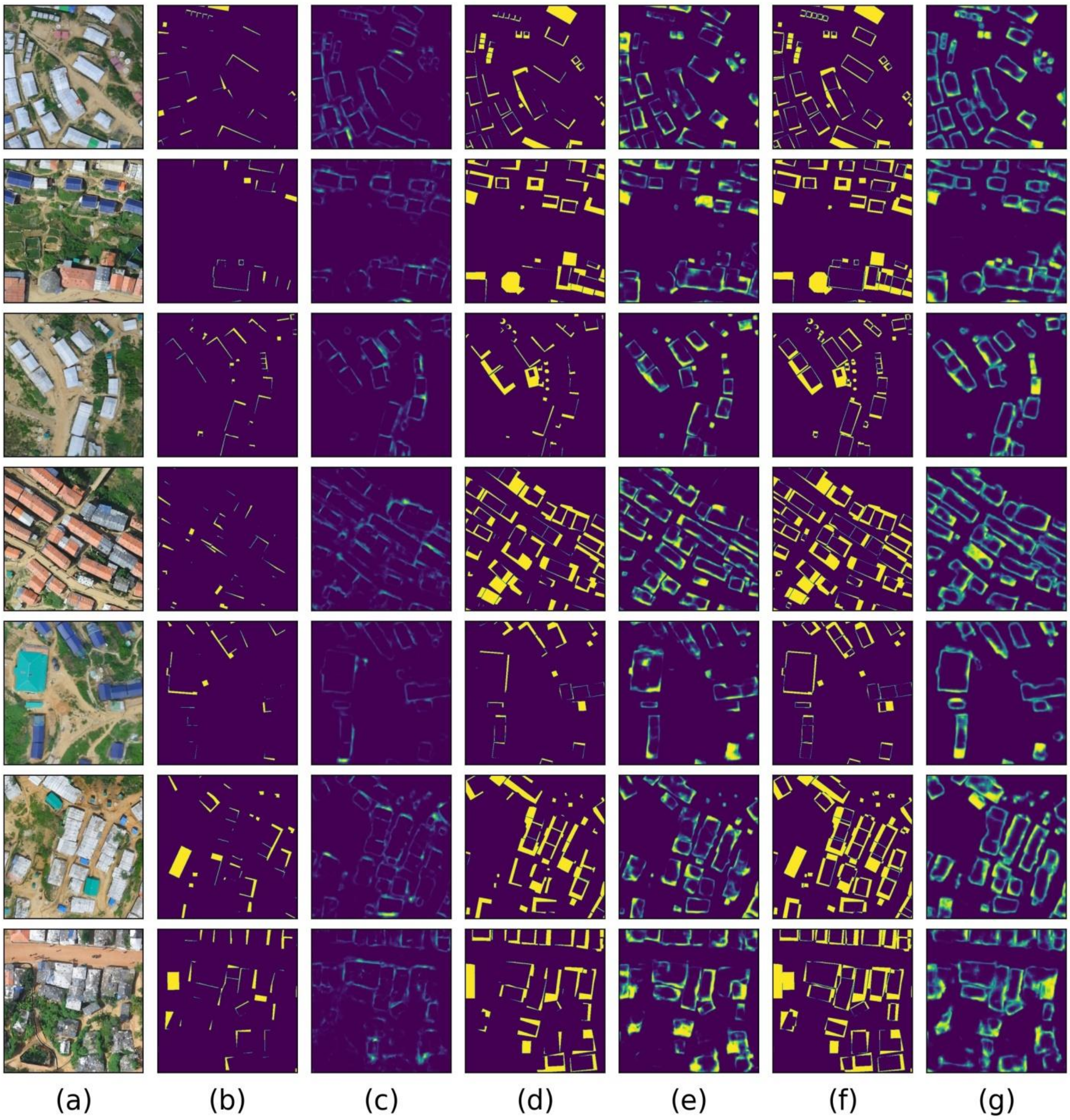
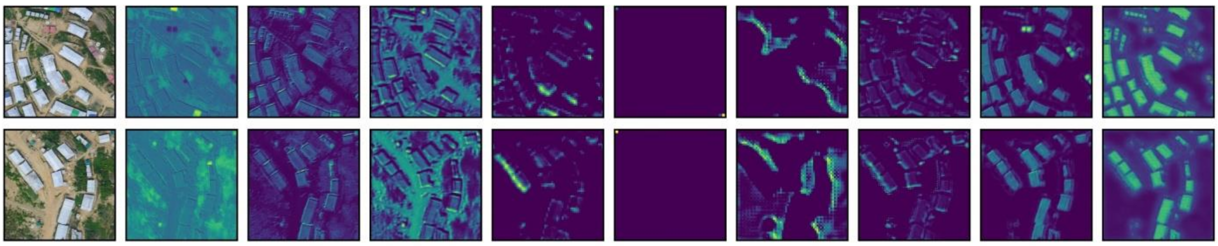
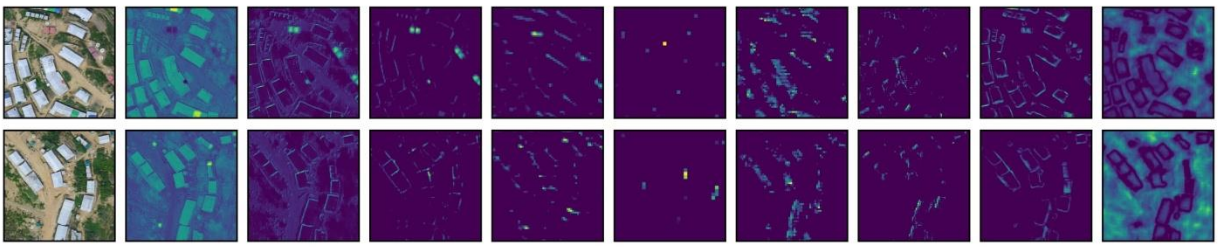


Figure 10. Examples of error matrix predictions (a) Image (b) FP error matrix (c) Predicted FP error matrix (d) FN error matrix (e) Predicted FN error matrix (f) General error matrix (g) Predicted general error matrix

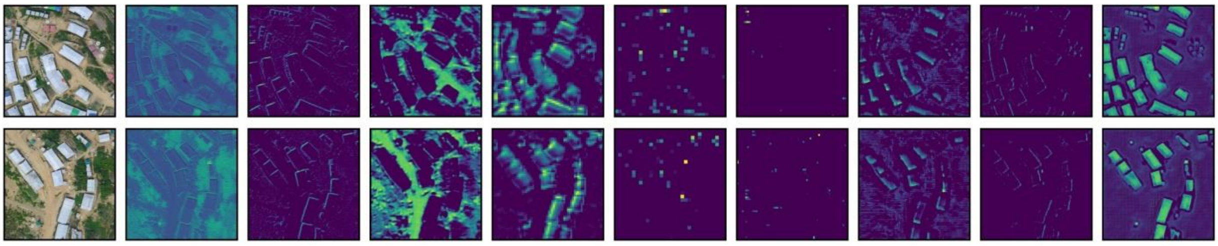
The general error matrix predictions are shown in Fig. 10(f) and the predicted general error matrix predictions are shown in Fig. 10(g). Among the three types of error matrices (general, FP and FN) the general error matrices are least sparse, since they are the element wise addition of the FP and FN error matrices. The extra information provided by separated FP and FN matrices are crucial to approximating useful noise features. Experimental results on our dataset confirm this statement. The I-EM model results are poorer than the CL model (albeit providing higher recall values at all thresholds) qualitatively and quantitatively (in terms of F1-score and IoU score on the independent test set). The predicted intermediate observed label also affects the predicted true label. The outputs of hidden blocks of different models are shown in Fig. 11, feature maps for learning error matrices (Fig. 11(b), 11(c), 11(d) are quite different from feature maps for learning base level building extraction (Fig. 11(a)). The activation maps in Fig. 11 are outputs of the blocks for each model architecture. The first feature map for each output is shown. The block outputs in Fig. 11 (U_1-U_4. the bottleneck and D_1-D_4) show discriminative properties of the learned mappings in terms of resolution and separability.



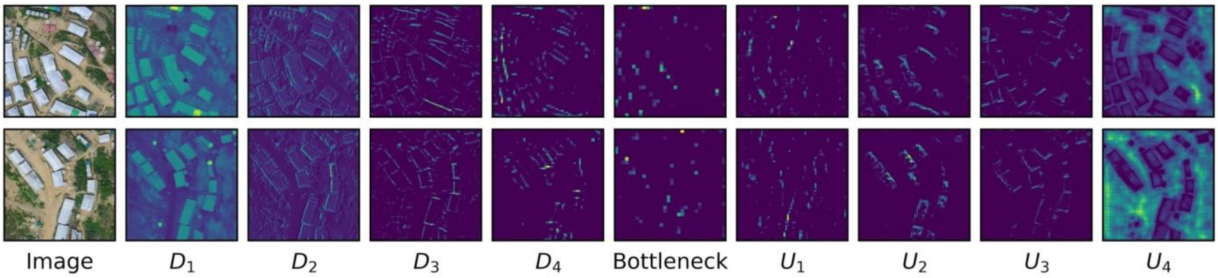
(a)



(b)



(c)



(d)

Figure 11. Outputs learned by hidden convolutional layers for building extraction and for error matrix approximation. (a) Clean label network (b) Error network (c) FP error network (d) FN error matrix network

Conclusion

In this work, we have provided a comprehensive taxonomy of label noise, in which the six formulated label noise models can be used to express any kind of label noise in building extraction tasks. Dense models are more apt than pixel-wise models for building extraction. We propose two new model frameworks for dense prediction based building extraction under label noise. The first model approximates the general error matrix as an intermediate step, but has poor performance improvements compared to the clean model. However, approximating the FP error matrix and the FN error matrix separately greatly improves performance over the idealistic scenario presented in the form of the CL model. Therefore, it is important to model the false positives and false negatives independently rather than using a general model for both types of pixel-level observed labels. Label noise in most building extraction cases is asymmetric, as also observed for our case; there is a massive imbalance in the pixel-level label noise i.e. there are much more false negatives than false positives. Therefore, a general model is not sufficient in modeling the FP and FN noise processes to a degree that can aid the larger task of noise-free building extraction. Qualitative results show that the error matrix models (FP, FN and general) all capture the intuition behind the model framework. The FP error matrix dense model has higher activations for regions right outside and adjacent to the actual clean building boundaries. Similarly, FN error matrix dense model has higher activations for regions inside and adjacent to the actual clean building boundaries. Clean labels and corresponding observed labels with real-world label noise are rarely available in conjunction with each other, which are essential for obtaining the error matrices outlined in our proposed methodologies, and thus limit the applicability.

Acknowledgments

This work is supported by Faculty Research Grant (CTRG-20-SEPS-14), North South University, Bashundhara, Dhaka 1229, Bangladesh.

Disclosure statement

No potential conflict of interest was reported by the authors.

Data and Codes Availability Statement

The data and codes that support the findings of this study are available at dedicated GitHub repository (https://github.com/nahian-ahmed/dense-label-noise).

Funding

This work is supported by Faculty Research Grant (CTRG-20-SEPS-14), North South University, Bashundhara, Dhaka 1229, Bangladesh.

References

1. Mnih, V., & Hinton, G. E. (2012). Learning to label aerial images from noisy data. In Proceedings of the 29th International conference on machine learning (ICML-12) (pp. 567-574).

2. Ahmed, N., Mahbub, R. B., & Rahman, R. M. (2020). Learning to extract buildings from ultra-high-resolution drone images and noisy labels. International Journal of Remote Sensing, 1-22.

3. Zhang, Z., Guo, W., Li, M., & Yu, W. (2020). GIS-supervised building extraction with label noise-adaptive fully convolutional neural network. IEEE Geoscience and Remote Sensing Letters.

4. Frénay, B., & Kabán, A. (2014, April). A comprehensive introduction to label noise. In ESANN.

5. Frénay, B., & Verleysen, M. (2013). Classification in the presence of label noise: a survey. IEEE transactions on neural networks and learning systems, 25(5), 845-869.

6. Lu, Z., Fu, Z., Xiang, T., Han, P., Wang, L., & Gao, X. (2016). Learning from weak and noisy labels for semantic segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(3), 486-500.

7. Maas, A., Rottensteiner, F., & Heipke, C. (2016). Using label noise robust logistic regression for automated updating of topographic geospatial databases. In XXIII ISPRS Congress, Commission VII 3 (2016), Nr. 7 (Vol. 3, No. 7, pp. 133-140). Göttingen: Copernicus GmbH.

8. Maas, A. E., Rottensteiner, F., & Heipke, C. (2019). A label noise tolerant random forest for the classification of remote sensing data based on outdated maps for training. Computer Vision and Image Understanding, 188, 102782.

9. Ghosh, A., Kumar, H., & Sastry, P. S. (2017, February). Robust loss functions under label noise for deep neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).

10. Rolnick, D., Veit, A., Belongie, S., & Shavit, N. (2017). Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694.

11. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., & Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1944-1952).

12. Jiang, L., Huang, D., Liu, M., & Yang, W. (2020, November). Beyond synthetic noise: Deep learning on controlled noisy labels. In International Conference on Machine Learning (pp. 4804-4815). PMLR.

13. Garcia, L. P., de Carvalho, A. C., & Lorena, A. C. (2015). Effect of label noise in the complexity of classification problems. Neurocomputing, 160, 108-119.

14. Pelletier, C., Valero, S., Inglada, J., Champion, N., Marais Sicre, C., & Dedieu, G. (2017). Effect of training class label noise on classification performances for land cover mapping with satellite image time series. Remote Sensing, 9(2), 173.

15. Frank, J., Rebbapragada, U., Bialas, J., Oommen, T., & Havens, T. C. (2017). Effect of label noise on the machine-learned classification of earthquake damage. Remote Sensing, 9(8), 803.

16. Angluin, D., & Laird, P. (1988). Learning from noisy examples. Machine Learning, 2(4), 343-370.

17. Lawrence, N., & Schölkopf, B. (2001, July). Estimating a kernel fisher discriminant in the presence of label noise. In 18th International Conference on Machine Learning (ICML 2001) (pp. 306-306). Morgan Kaufmann.

18. Pérez, C. J., Girón, F. J., Martín, J., Ruiz, M., & Rojano, C. (2007). Misclassified multinomial data: a Bayesian approach. RACSAM, 101(1), 71-80.

19. Xu, Y., Wu, L., Xie, Z., & Chen, Z. (2018). Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. Remote Sensing, 10(1), 144.

20. Yuan, J. (2017). Learning building extraction in aerial scenes with convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 40(11), 2793-2798.

21. Chen, K., Fu, K., Gao, X., Yan, M., Sun, X., & Zhang, H. (2017, July). Building extraction from remote sensing images with deep learning in a supervised manner. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 1672-1675). IEEE.

22. Yang, H., Wu, P., Yao, X., Wu, Y., Wang, B., & Xu, Y. (2018). Building extraction in very high resolution imagery by dense-attention networks. Remote Sensing, 10(11), 1768.

23. Ji, S., Wei, S., & Lu, M. (2018). Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Transactions on Geoscience and Remote Sensing, 57(1), 574-586.

24. Vakalopoulou, M., Karantzalos, K., Komodakis, N., & Paragios, N. (2015, July). Building detection in very high resolution multispectral data with deep learning features. In 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 1873-1876). IEEE.

25. Huang, Z., Cheng, G., Wang, H., Li, H., Shi, L., & Pan, C. (2016, July). Building extraction from multi-source remote sensing images via deep deconvolution neural networks. In 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 1835-1838). IEEE.

26. Shrestha, S., & Vanneschi, L. (2018). Improved fully convolutional network with conditional random fields for building extraction. Remote Sensing, 10(7), 1135.

27. Boonpook, W., Tan, Y., & Xu, B. (2021). Deep learning-based multi-feature semantic segmentation in building extraction from images of UAV photogrammetry. International Journal of Remote Sensing, 42(1), 1-19.

28. Sun, S., Mu, L., Wang, L., Liu, P., Liu, X., & Zhang, Y. (2021). Semantic Segmentation for Buildings of Large Intra-Class Variation in Remote Sensing Images with O-GAN. Remote Sensing, 13(3), 475.

29. Wang, S., Hou, X., & Zhao, X. (2020). Automatic building extraction from high-resolution aerial imagery via fully convolutional encoder-decoder network with non-local block. IEEE Access, 8, 7313-7322.

30. Guo, M., Liu, H., Xu, Y., & Huang, Y. (2020). Building extraction based on U-Net with an attention block and multiple losses. Remote Sensing, 12(9), 1400.

31. Shao, Z., Tang, P., Wang, Z., Saleem, N., Yam, S., & Sommai, C. (2020). BRRNet: A fully convolutional neural network for automatic building extraction from high-resolution remote sensing images. Remote Sensing, 12(6), 1050.

32. Jing, H., Sun, X., Wang, Z., Chen, K., Diao, W., & Fu, K. (2021). Fine Building Segmentation in High-Resolution SAR Images via Selective Pyramid Dilated Network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

33. He, Q., Sun, X., Yan, Z., & Fu, K. (2021). DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images. IEEE Transactions on Geoscience and Remote Sensing.