Article

Handling Incomplete Instance Annotations via Asymmetric Loss Function

Feng Chen [1,*], Michael Pound [1] and Andrew P. French [1]

[1] School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK; feng.chen@nottingham.ac.uk, michael.pound@nottingham.ac.uk, andrew.p.french@nottingham.ac.uk

* Correspondence: feng.chen@nottingham.ac.uk

Abstract: Annotating training data is a time consuming and labor intensive process in deep learning, especially for images with many objects present. In this paper, we propose a method to allow deep networks to be trained on data with reduced numbers of annotations (per image) in heatmap regression tasks (e.g. object detection and counting), by applying an asymmetric loss function. In a real scenario, this reduction of annotations can be imposed by the researchers (e.g. ask the annotators to label only 50% of what they see in each image), or can potentially counteract unintentionally missing labels from the annotators. To demonstrate the effectiveness of our method, we conduct experiments in two domains, crowd counting and wheat spikelet detection, using different deep network architecture. We drop various percentages of instance annotations *per* image in training. Results show that an asymmetric loss function is effective across different models and datasets, even in very extreme cases with limited annotations provided (e.g. 90% of the original annotations reduced). Whilst tuning of the key parameters are required, we find that setting conservative parameter values can help more realistic situations, where only small amounts of data have been missed by annotators.

Keywords: Deep Learning; Reducing Training Annotations per Image; Object Detection; Object Counting; Asymmetric Loss Function

Citation: Chen, F.; Pound, M.; French, A.P. Handling Incomplete Instance Annotations via Asymmetric Loss Function. *Remote Sens.* **2021**, *1*, 0. https://doi.org/

1. Introduction

Training deep networks usually requires a lot of human-annotated data, and this annotation process is time-consuming and expensive, especially for images containing many objects. The fatigue and subjectivity of human annotators can also cause missed annotations, which do not benefit the deep networks. In recent years, research on training deep networks with limited numbers of annotations has become increasingly popular, leading to better robustness in deep learning, and higher efficiency for the annotation stage. Although some methods have achieved promising results, they either tend to focus on image classification and semantic segmentation, or apply some overly-complex or task-specific techniques. They also mostly focus on reducing the size of the entire training set, while our approach explores the scenarios of reducing annotations per training image. In this paper, we propose a simple yet effective approach to allow training of deep networks for heatmap regression tasks using fewer annotations, by replacing the original loss function with an asymmetric loss function (e.g. asymmetric MSE).

Traditional Mean Square Error (MSE) is widely used as a loss function in many object detection and counting tasks, especially those using heatmap regression. A heatmap is normally constructed by many Gaussian-like regions, and MSE can guide the network to learn such regions. However, if some of the ground-truth point annotations are removed, the network will easily overfit to the remaining data, leading to poor performance on the test data. Under this situation, an Asymmetric Mean Square Error (AMSE) has the potential to help the network learn the useful information of the remaining data and generalize to

the full predicted heatmap. Figure **??** shows the key properties of AMSE functions. If we set an appropriate weight (β), which is related to how many annotations we expect are reduced, the AMSE function will amplify the effects of the remaining annotations and at the same time reduce the interference from the areas of the image with instances present, but annotations missing. In essence, an asymetric loss allows us to signal to the network that we know not all positive training samples in the image are labelled, and to act accordingly during training. A detailed comparison of MSE and AMSE will be shown in Section 3.

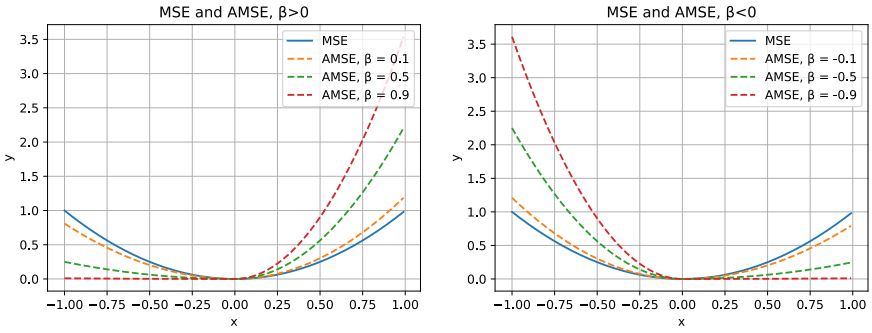


Figure 1. A comparison between MSE and AMSE. In both graphs, the blue solid line represents MSE, and the dashed lines show the behaviour of AMSE under different weights (noted as β). The graph on the left shows AMSE under $\beta > 0$, where the left part of each curve is suppressed, while the graph on the right shows the opposite. In this paper, we set $x = Y_i - \hat{Y}_i$, where Y_i represents ground truth and $\hat{Y}_i$ stands for prediction from the model, and we want the model to overpredict, so $\beta > 0$ (left graph), which penalize more on the scenarios of $\hat{Y}_i < Y_i$, is adopted.

To demonstrate the generalisability and effectiveness of using an AMSE, we use two datasets from very different domains, and two types of neural network architecture. We conduct experiments on a crowd counting dataset (ShanghaiTech [**?**]) using a Context-Aware Network (CAN [**?**]). We also conduct experiments on a wheat grain ("spikelet") detection dataset using a Stacked Hourglass Network [**?**]. We test different use cases for reduced annotations, which we term the "drop rate", which ranges from minor (10% drop) to extreme (90% drop) scenarios.

Our main contributions are: 1) A description and evaluation of a method which uses an AMSE loss function to counteract performance loss from reduced number of annotations per image, 2) An exploration of the sensitivity of the single tuning parameter of AMSE over two diverse datasets and two model architectures 3) Demonstration of competitive performance on public counting and detection datasets in the presence of substantial (from 10% to 90%) reduced annotations. Our results show that when the reduction of annotations per image is less than 50%, our method can achieve comparable performance to the baseline (i.e. training on a fully-annotated dataset with MSE loss); when the reduction is greater than 70%, our method still perform much better than the model trained on the same reduced dataset with MSE loss.

The remainder of this paper is arranged as follows. Section 2 introduces the related work. Section 3 defines the problem and our method. Section 4 presents our experiments and analysis. Section 5 concludes and discusses the paper.

2. Related Work

The quality and quantity of training data are crucial in deep learning. Previous authors have proposed approaches to either reduce the volume of training data, or enable deep-learning models to tolerate noisy and missing data. These experiments and methods impact a large number of research areas including crowdsourcing, semi-supervised learning, meta-learning and loss functions.

Crowdsouring has become a popular method to acquire labels due to the increasing demand for large image datasets, especially after the Imagenet dataset [**?**] was released.

However, imperfect labels are commonly produced via crowdsourcing, which are often produced by non-expert users, leading to a need to suppress or handle noisy labels [**?** **?** **?**].

In recent years, researchers have started to pay more attention to model capacity and generalization. In [**?**], the authors discuss generalization in deep learning. They conduct a series of experiments including swapping and deleting labels on different datasets, including CIFAR10 [**?**] and Imagenet [**?**], and find that deep models can learn the general patterns of the data in early epochs but soon overfit on the local patterns. Krueger [**?**] and Arpit [**?**] also explore experiments to demonstrate generalization and memorization in deep learning. Due to the huge capacity of a deep neural network, in practice all training information can be memorized by the model, including noisy and missing labels. Since then more methods of handling noisy and missing labels have been explored.

Goldberger and Ben-Reuven [**?**] propose a method using the Expectation Maximization (EM) algorithm to estimate the correct labels and retrain the network after a certain number of epochs. Northcutt et al. [**?**] propose to use confident learning to estimate label errors based on the assumption of the dependency between the noise and true labels. These two papers contribute most to recognizing true labels, and estimating the noise level in data.

Other researchers focus on improving training techniques and tuning the model architecture instead of recognizing noise. Jiang et al. [**?**] propose a step-by-step curriculum learning model that learns the easier patterns first and then generalizes to the more difficult cases. Li et al. [**?**] propose a method to perform a meta-learning update before a gradient update, where they generate synthetic noisy labels in the training process to simulate real-life annotation noise. Rebuffi et al. [**?**] utilize techniques in semi-supervised learning and transfer learning to train models on scarcely annotated datasets. They first train the model on various other datasets to initialize weights, and then use a two-phase strategy, one phase to fit the labelled data and the second phase to fit the unlabelled data, to optimize the network. Lu et al. [**?**] propose a method which builds an L1-optimization-based sparse model with an intermediate labelling variable to solve weak and noisy labels in semantic segmentation.

Here we can argue that the existing methods are either overly-complex (such as requiring more input parameters, or modifying an architecture) or typically only developed for image classification problems. Also, most papers discuss the issues of noisy labels (e.g. randomly assigned labels) instead of labels altogether missing. This is the motivation for our development of our simple method (both computationally and intuitively), with only one tuning parameter, which is architecture-independent and able to handle the problem of missing annotations in counting and detection.

3. Problem Definition and Method

3.1. Problem Definition

The core problem we are addressing is training a deep network with competitive performance using a reduced number of annotations per image. This reduction of annotations can be caused by two possible scenarios: 1) intentional reduction in annotation quality, as a result of having a learning system which is able to cope with quicker, less accurate annotation regimes, or 2) errors and oversight on the part of the annotator trying to achieve "perfect" labelling. The first of these scenarios could be the result of instructing an annotator to only annotate approximately half of the objects they can see in the image, for example. If we can train a network successfully in this regime, we can cut annotator time whilst still acquiring high quality annotations for those that are given. In contrast, in the second scenario we do not know per se exactly how many annotations are missed on each image as missing annotations are the result of errors. However, with more than one annotator we could estimate the amount of lost annotations statistically. Our method is valuable for both of these scenarios.

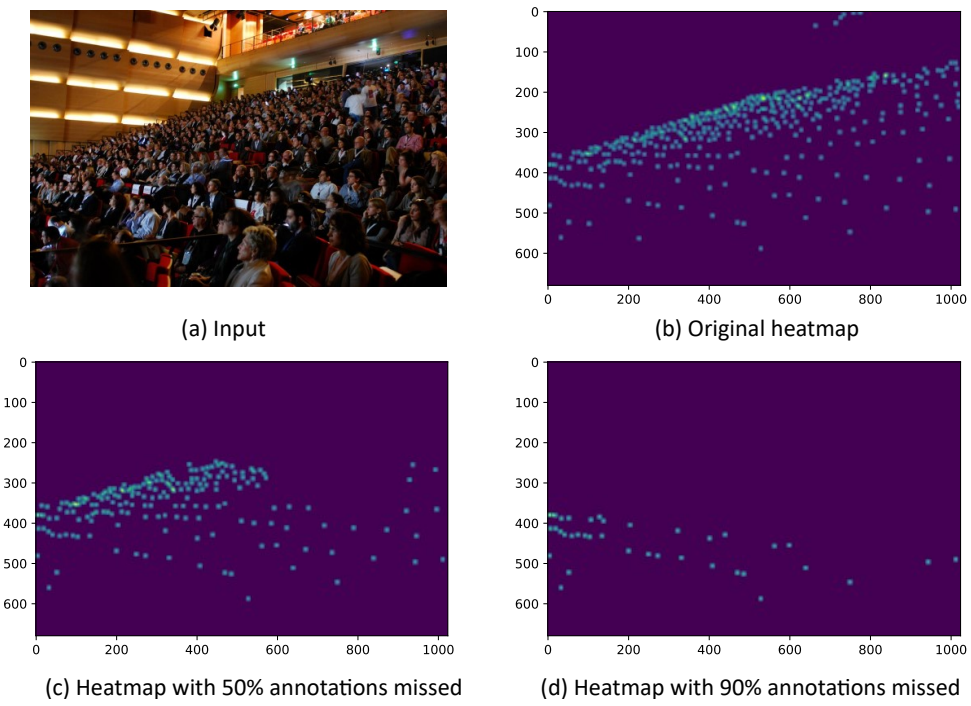


Figure 2. A graphical overview of reduced annotations. (a) an input image from ShanghaiTech crowd counting dataset [**?**]. (b) the original instance heatmap image, (c) and (d) show the same heatmap as (b) but with 50% and 90% of annotations dropped respectively. Our aim is to train a well-performing network using dropped data, as represented in in (c) and (d).

Our method is developed to tackle deep learning-based heatmap regression problems, for object detection and counting, on 2D colour images. These images are dot-annotated, from which a Gaussian kernel is applied to generate the heatmaps to represent locations. We assume that there can be different drop rates $dr \in [0, 1]$ applied to the label dataset. The drop rate dr is applied equally to all the ground truth instances in the dataset; the annotations in each ground-truth heatmap are randomly dropped subject to dr. For example, if $dr = 0$, the ground-truth heatmaps are unchanged, but if $dr = 1$, then there will be no annotation in any ground-truth heatmap of that dataset. When $dr = 0.5$, there are 50% annotations dropped randomly in *each* ground-truth heatmap image. Our task is to examine AMSE performance when training a deep learning model on such a dropped dataset, and compare the performance to 1) baseline MSE methods with zero drop out (ie. "normal" annotation and training); 2) MSE methods with different drop rates (ie. normal loss but missing annotations). Figure **??** illustrates heatmaps with different drop rates for comparison.

3.2. AMSE V.S. MSE

A commonly used loss function in heatmap regression problems is Mean Square Error (MSE), which is expressed as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2, \tag{1}$$

where Y_i is the ground truth and $\hat{Y}_i$ is the predicted value.

In our method, we use an Asymmetric Mean Square Error (AMSE) to counteract the effect of the dropped annotations. AMSE is expressed as:

$$AMSE = \frac{1}{N}\sum_{i=1}^{N}\{[\beta + sign(Y_i - \hat{Y}_i)] * (Y_i - \hat{Y}_i)\}^2, \tag{2}$$

where $\beta \in [-1, 1]$ is a constant weight set manually, and $sign(*)$ is expressed as:

$$sign(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases} \tag{3}$$

The difference between MSE and AMSE is illustrated in Figure **??**. From the two graphs in Figure **??** we can see that MSE maps the input x equally no matter whether $x < 0$ or $x > 0$, while the AMSE amplifies the effect of one side of the input and suppresses that of the other side. This property, as applied as a loss function, is explored below. To note, when β in AMSE is set to 0, the AMSE is equivalent to traditional MSE.

In the case of reduced annotations, we assume that in the original case, there are M annotations appearing, but now $dr * M$ annotations are dropped so only $(1 - dr) * M$ annotations are left. We want to force the network to learn more from the preserved annotations, but have less confidence on the *background* pixels, as these may contain false negative features (i.e. true instances without annotations). A ground truth heatmap is constructed with some Gaussian-like (hot-spot) areas (values within these areas > 0), and the remaining 0-valued background region. In the case of reduced annotations, some of these hot-spot areas are omitted. As the number of the instance hot-spots is underestimated in such a ground truth, the model should prefer to *overestimate* the number of predictions. In other words, we expect the model to predict a larger $\hat{Y}_i$ (compared to the ground truth Y_i). If we set x in Figure **??** equal to $(Y_i - \hat{Y}_i)$, we want to allocate a greater loss for $(Y_i - \hat{Y}_i) > 0$, and the larger the $(Y_i - \hat{Y}_i)$ is, the greater the loss that should be applied. This will guide the network to learn a higher $\hat{Y}_i$. As a result, the model learns to avoid *underestimation*, which in practice cancels out the effects from reduced annotations. Therefore, $\beta > 0$ is adopted in our method to deal specifically with reduced-annotation cases.

This weight, β, is the key (and only) tuning parameter of AMSE. As shown in Figure **??**, when β grows, the curve becomes more unbalanced. It is reasonable to assume from this that the ideal value of β is related to the drop rate dr for a particular dataset; this is explored experimentally in the following section.

4. Experiments and Results

4.1. Datasets and Implementation Details

Our experiments are based on two contrasting datasets: a crowd counting dataset and a wheat spikelet detection dataset. ShanghaiTech [**?**] is a crowd counting dataset released in 2016 consisting of 2 parts: Part_A is constructed mostly of images with a large number of people, and images in Part_B are relatively less crowded. The images in ShanghaiTech dataset are dot-annotated on each human's head and the corresponding ground-truth heatmaps are generated by applying Gaussian kernels on them. Following the same settings as previous researchers ([**? ?**]), a geometry-adaptive kernel is adopted to generate the density maps for part A and a fixed kernel is used for part B. The ACID (Annotated Crop Image Database) dataset [**?**] is constructed from 520 wheat plant images, with their 'spikelets' (individual wheat grains) and ears annotated. A Gaussian kernel is applied to these annotations, to generate the ground-truth heatmaps. In our experiments, only the annotations on spikelets are adopted. Example images from both datasets are shown in Figure **??**.

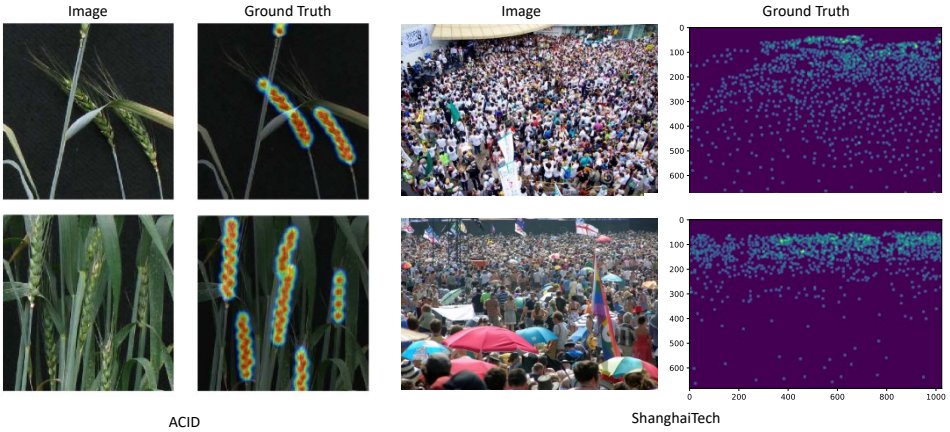


Figure 3. Example images and the corresponding ground truth from the datasets used in this paper. Left: ACID dataset [**?**]. Right: ShanghaiTech dataset [**?**].

To test the scenarios of reduced annotations, we randomly drop some portion of the annotations in each ground-truth image before generating the corresponding heatmap. The dropped portions are decided by the drop rate $dr \in [0, 1]$ as discussed in Section **??**. To note, we drop the annotations directly from the original dot-annotated ground truth, before any pre-processing of the data,for example generating heatmaps and data augmentation. This dropping process is a simulation of the real-life case when the annotators omit some annotations, either as unintentional error, or as part of a "quick but messy" annotation scheme, where annotators may be told that annotating less than 100% of instances is acceptable. In extreme cases, annotators could be told to only annotate a small portion of instances, for example. Importantly, we only drop the annotations on the training sets, while the annotations in the test sets remain the same as the original dataset (i.e. 100% annotated); despite drop out we want the networks to detect or count all instances if possible. We apply different AMSE weights β to the datasets with various drop out rates dr to explore the relationship between β and dr.

In the following experiments, we adopt the original structure of the Context-Aware Network (CAN [**?**]) to perform crowd counting on ShanghaiTech dataset; and we build a 4-stack 2-block Stacked Hourglass Network [**?**], using RGB wheat images as inputs to perform spikelet detection on ACID dataset. The original loss function (MSE) in both models is replaced with AMSE in the experimental condition with a range of β. During training, the model is tested every epoch for crowd counting and every 5 epochs for spikelet detection against test data which is fully-annotated and unseen in training.

4.2. Crowd Counting with AMSE

In the crowd counting experiments, each model is trained on dropped data and tested on unseen and undropped data every epoch to view performance. The predicted counts are calculated by summing up the values through each pixel of the heatmaps. The test results are presented as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). In the following content, 'ShanghaiTech Part A' is abbreviated to 'ShA', 'ShanghaiTech Part B' is abbreviated to 'ShB', and, as introduced before, the Context-Aware Network is abbreviated to 'CAN'. For example, 'ShA-CAN' represents the Context-Aware Network trained and tested on ShanghaiTech Part A.

We first explore the effects of AMSE on the original, fully-annotated dataset ($dr = 0$), to explore performance where we have "perfect" annotation; ideally we want to perform broadly as well as baseline MSE in this scenario. The test results, reported in MAE and RMSE (lower is better), are shown in Table **??**. Generally speaking, we can observe that test errors become bigger as β grows. One exception is $\beta = 0.1$, which obtains even better results than $\beta = 0$ (MSE loss function). This may indicate that the original data has already missed some annotations - although of course this is hard to verify. From Table **??**, we can

conclude that an undropped dataset is indeed sensitive to different β, but for a relatively small β, for example up to $\beta \leq 0.2$ in Part A and $\beta \leq 0.5$ in Part B, the performance is not affected significantly, and even achieves a slight improvement in these datasets.

Table 1: Test MAE and RMSE (lower is better) for $dr = 0$ under different AMSE weights β on ShanghaiTech dataset using CAN: (a) Test results on ShanghaiTech Part A; (b) Test results on ShanghaiTech Part A. Each reported MAE and RMSE is selected as the best among the corresponding list of test results. $\beta = 0$ is equivalent to using MSE loss function and $\beta \neq 0$ stands for using AMSE loss function in training. The best result (i.e. minimum) in each column is highlighted in bold.

(a) ShA-CAN Results ($dr = 0$)

β	Test MAE	Test RMSE
0	67.3	103.5
0.1	**64.72**	**99.6**
0.2	69.41	101.21
0.3	101.11	139.55
0.5	151.21	194.66
0.7	305.88	365.38
0.9	806.57	949.8

(b) ShB-CAN Results ($dr = 0$)

β	Test MAE	Test RMSE
0	8.58	13.72
0.1	**8.36**	**13.26**
0.3	8.62	13.66
0.5	8.86	13.99
0.7	11.22	15.95
0.9	28.51	32.09

We next to create dropped datasets from the original ShanghaiTech dataset using $dr = 0.1, 0.3, 0.5, 0.7, 0.9$, and then apply various values of β to see how they react to different drop rates. Figure **??** (reporting MAE) and Figure **??** (reporting RMSE) compare the test results between using MSE and AMSE loss function in training on ShA-CAN. Figure **??** (reporting MAE) and Figure **??** (reporting RMSE) compare the corresponding test results on ShB-CAN. Table **??** (ShA-CAN) and Table **??** (ShB-CAN) provide numerical results, showing the best test MAE/RMSE achieved for each dr and the corresponding β this was achieved at using our method. As a comparison, the results achieved by training with the original MSE loss function are shown in the column labelled $\beta = 0$.

From Figure **??** which presents test MAE in ShA-CAN, we can see that as drop rate grows, the performance gap between MSE (blue line) and AMSE (orange line) becomes larger; particularly when $dr \geq 0.3$ AMSE significantly outperforms MSE. We can see a similar trend in Figure **??** where test RMSE in ShA-CAN is shown. From Table **??** we can see that when $dr \leq 0.3$, the results achieved by AMSE not only beat those achieved by MSE on the same drop rate but also are marginally better than the baseline. This baseline scenario is the model trained on the *undropped* dataset with MSE loss function, i.e. $dr = 0$ and $\beta = 0$ in the tables. The results for $dr = 0.5$ show that with 50% data dropped, the performance only reduces 6.15% in MAE and 3.76% in RMSE compared to the baseline. Even in the very extreme cases, where 70% and 90% of annotations are dropped per image, the models only experience 14.81% and 34.44% dropped in MAE respectively. Although these extreme cases are unlikely to happen in real life (unless annotators are instructed to behave this way), they still show the robustness and effectiveness of our method.

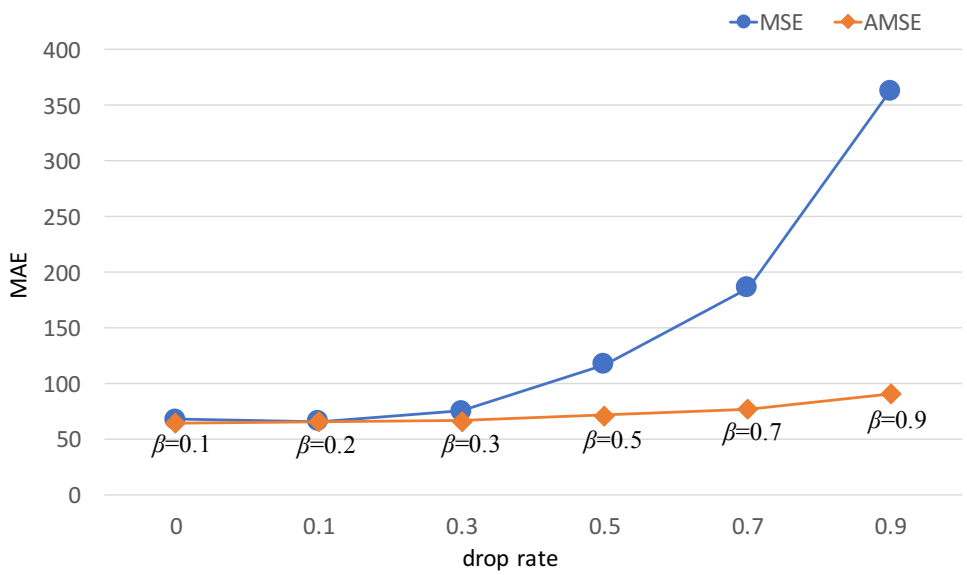


Figure 4. Test MAE (lower is better) on ShA-CAN under different drop rates. Blue line: results obtained by using MSE loss function (i.e. $\beta = 0$) in training. Orange line: best results obtained by using AMSE loss function in training, with β value under each point showing when the corresponding result is achieved.

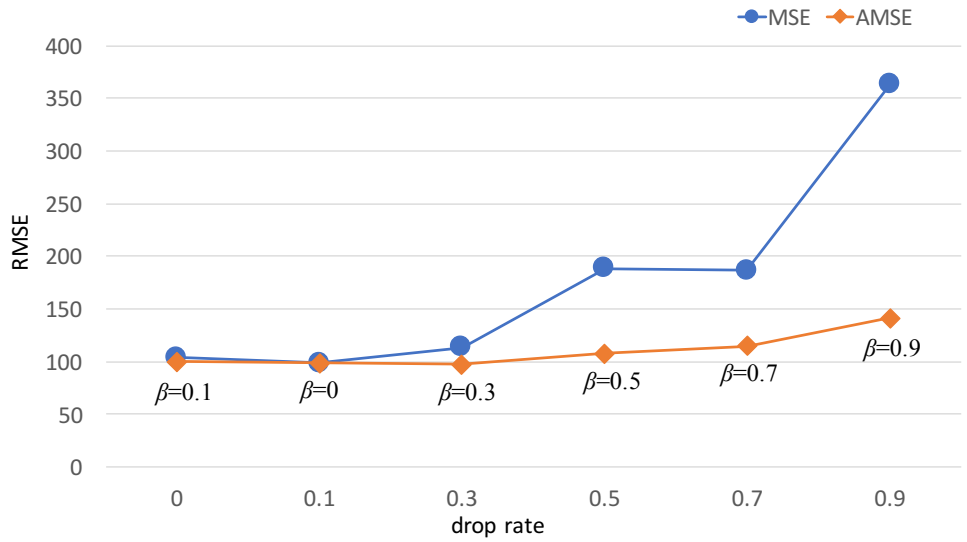


Figure 5. Test RMSE (lower is better) on ShA-CAN under different drop rates. Blue line: results obtained by using MSE loss function (i.e. $\beta = 0$) in training. Orange line: best results obtained by using AMSE loss function in training, with β value under each point showing when the corresponding result is achieved, where $\beta = 0$ means under that specific drop rate, the best result is achieved by training with MSE loss function.

Table 2: Test MAE/RMSE (lower is better) on ShA-CAN under different drop rates. Columns under $\beta = 0$: results obtained by training with MSE loss function. Columns under $\beta \neq 0$: best results obtained by training with AMSE loss function, with β value in bracket showing when the corresponding result is achieved.

ShA-CAN Test MAE/RMSE (various *dr*)

dr	Test MAE		Test RMSE	
	β=0	$\beta\neq0$ (best β)	β=0	$\beta\neq0$ (best β)
0	67.3	**64.72** (0.1)	103.5	**99.6** (0.1)
0.1	65.71	**65.66** (0.2)	98.91	**98.91** (0)
0.3	75.34	**66.65** (0.3)	113.56	**97.63** (0.3)
0.5	116.87	**71.44** (0.5)	188.52	**107.39** (0.5)
0.7	186.26	**77.27** (0.7)	186.26	**114.82** (0.7)
0.9	363.15	**90.48** (0.9)	363.15	**141.39** (0.9)

Similar trends are also shown in ShB-CAN test results. From Figure **??** and Figure **??** it can be observed that when $dr \geq 0.3$ the test MAE and RMSE achieved by AMSE exceed those obtained by MSE loss function; and the gap becomes wider when $dr \geq 0.5$. From Table **??** we can see that when $dr \leq 0.3$ the test results obtained by AMSE are better than those achieved by MSE on the same drop rate, and also very close to the baseline (i.e. $dr = 0, \beta = 0$). When $dr \geq 0.5$, the performance of models trained with AMSE is far better than the ones trained with MSE. We can also observe a general preference for increasing β with dr, although this pattern is slightly less consistent for ShB-CAN.

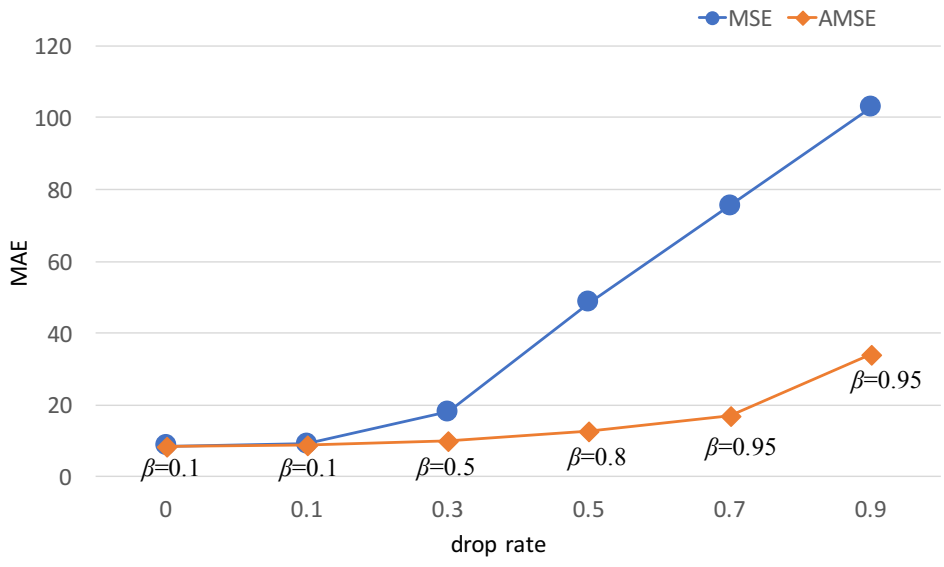


Figure 6. Test MAE (lower is better) on ShB-CAN under different drop rates. Blue line: results obtained by using MSE loss function (i.e. $\beta = 0$) in training. Orange line: best results obtained by using AMSE loss function in training, with β value under each point showing when the corresponding result is achieved.

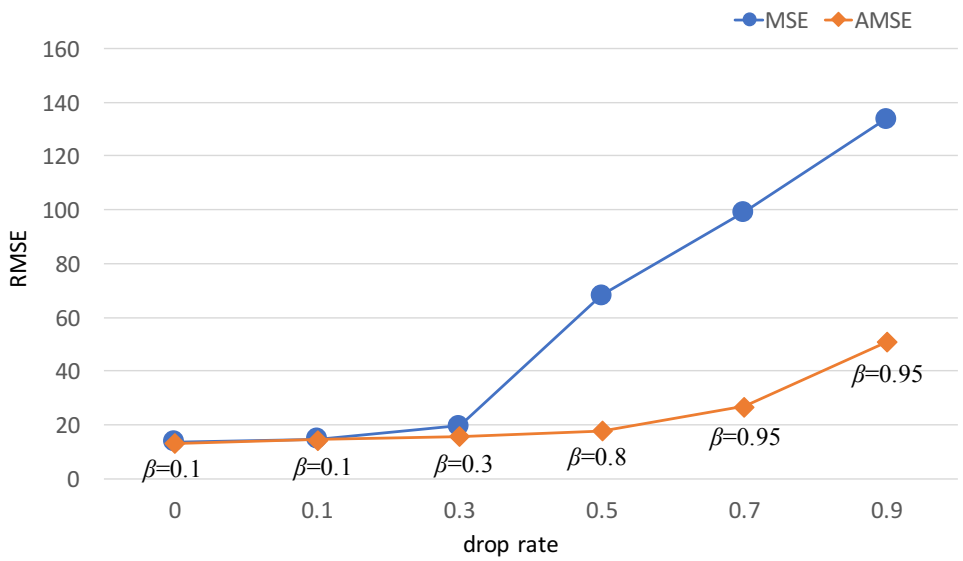


Figure 7. Test RMSE (lower is better) on ShB-CAN under different drop rates. Blue line: results obtained by using MSE loss function (i.e. $\beta = 0$) in training. Orange line: best results obtained by using AMSE loss function in training, with β value under each point showing when the corresponding result is achieved.

Table 3: Test MAE/RMSE (lower is better) on ShB-CAN under different drop rates. Columns under $\beta = 0$: results obtained by training with MSE loss function. Columns under $\beta \neq 0$: best results obtained by training with AMSE loss function, with β value in bracket showing when the corresponding result is achieved.

ShB-CAN Test MAE/RMSE (various *dr*)

dr	Test MAE		Test RMSE	
	β=0	$\beta\neq$0 (best β)	β=0	$\beta\neq$0 (best β)
0	8.58	**8.36** (0.1)	13.72	**13.26** (0.1)
0.1	9.21	**8.78** (0.1)	14.79	**14.39** (0.1)
0.3	18.13	**10.04** (0.5)	19.71	**15.86** (0.3)
0.5	48.42	**12.57** (0.8)	68.16	**17.89** (0.8)
0.7	75.5	**16.87** (0.95)	99.06	**26.81** (0.95)
0.9	102.72	**34.04** (0.95)	133.75	**50.8** (0.95)

From the test results of ShA/ShB-CAN on various dr, we can conclude that AMSE outperforms MSE when there is annotations dropped in training. Particularly, if one is asked to annotate 70% of points per image (ie. dr=0.3), using AMSE with a proper β can still obtain performance on par with training on 100%-annotated data using MSE. If one wishes to further reduce annotations, the results of AMSE will still fall within an acceptable area until dr becomes extreme (i.e. $dr > 0.7$). If we plot dr against β where the best test MAE is achieved under different dr on ShA/ShB-CAN (Figure **??**), we can note in general that the best β value increases as dr grows larger. This can provide some hints in how to select a proper β in real life scenarios. MAE is adopted here because it is a more-commonly used metric than RMSE in crowd counting.

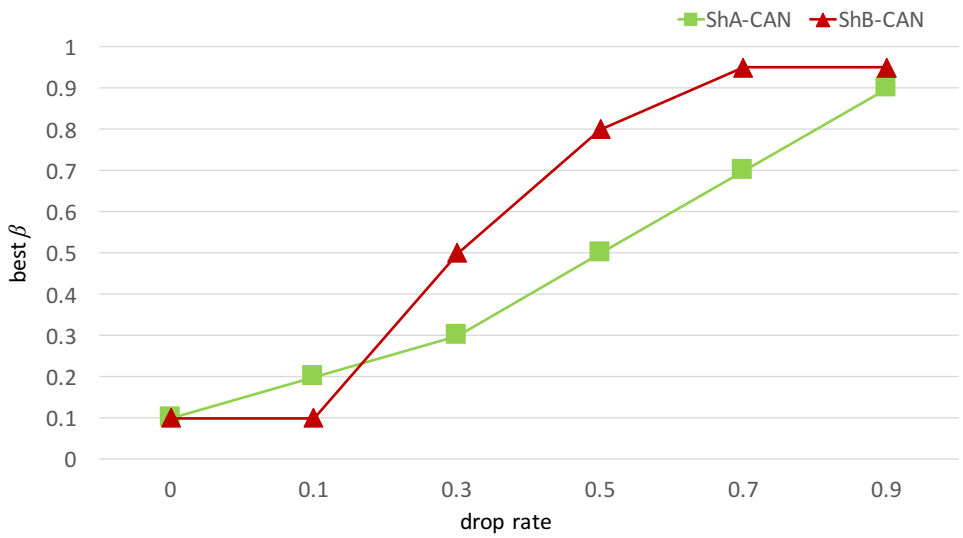
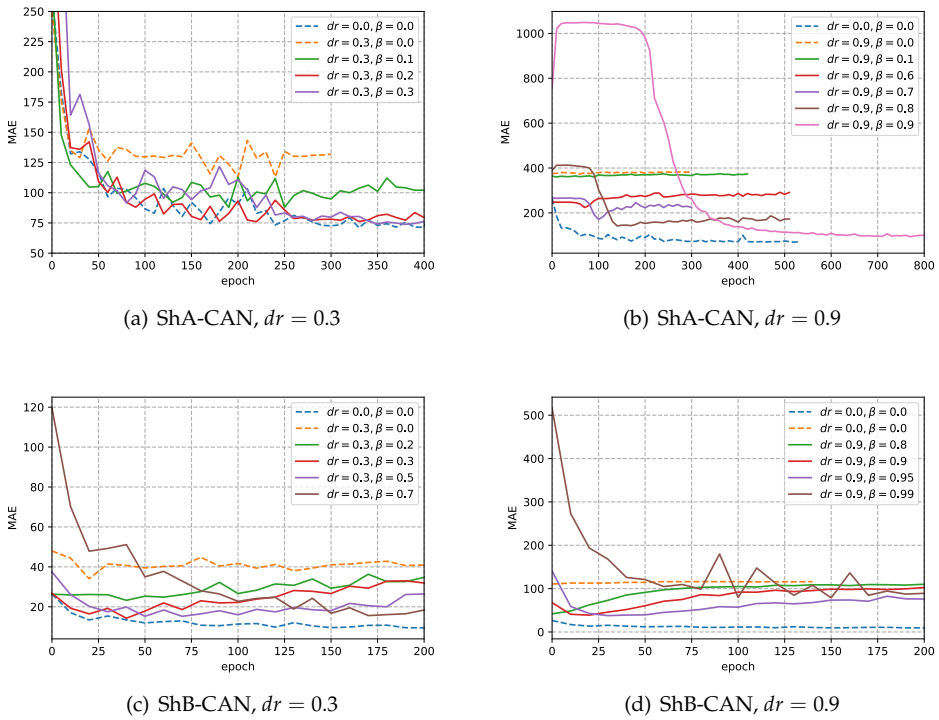


Figure 8. β value where best test MAE is achieved under each drop rate on ShA/ShB-CAN.

As we have revealed the effectiveness and robustness of our method on crowd counting datasets, now let us explore the mechanism of AMSE through some specific scenarios. Figure **??** presents test MAE curves of ShA/ShB-CAN under dr=0.3 and 0.9, which mimics a realistic case of noisy annotation, and an extreme case of intentionally-dropped annotations respectively. The models are tested on unseen images with a fully-annotated ground truth after each training epoch, and the results are shown as averaged every 10th epoch. The complete version of test graphs, which include $dr = 0.1, 0.3, 0.5, 0.7, 0.9$ on both ShA-CAN and ShB-CAN with more β values, is attached in Appendix **??** (see Figure **??** and Figure **??**). From Figure **??** we observe:

- Almost every β improves the test results, while there is an optimal choice of β among all values for each dr.
- Under a given dr, a relatively high (not necessarily the highest) β causes the network start from a larger test error and converge slower. However, the network can be trained longer before overfitting the reduced annotations and hence converge to a better point. For example in Figure **??** (b), which shows test results of ShA-CAN under $dr = 0.9$, the pink solid line ($\beta = 0.9$) clearly converges later than the other parameters, but to a lower MAE.
- Under a given dr, choosing a too high β may reduce performance, though it is still better than not using the AMSE loss function (orange dashed line). For example, in Figure **??** (d), which shows test results of ShB-CAN under $dr = 0.9$, the brown solid line ($\beta = 0.99$) performs worse than the purple solid line ($\beta = 0.95$), while it is still better than the orange dashed line (using MSE loss function).



(a) ShA-CAN, $dr = 0.3$ (b) ShA-CAN, $dr = 0.9$

(c) ShB-CAN, $dr = 0.3$ (d) ShB-CAN, $dr = 0.9$

Figure 9. Test MAE of ShA/ShB-CAN under various dr across different training stages: (a) Test MAE of ShA-CAN trained on $dr = 0.3$; (b) Test MAE of ShA-CAN trained on $dr = 0.9$; (c) Test MAE of ShB-CAN trained on $dr = 0.3$; (d) Test MAE of ShB-CAN trained on $dr = 0.9$. The results are averaged every 10th epoch. In each sub-graph, the blue dashed line (trained on fully-annotated data with MSE loss function) and the orange dashed line (trained on reduced data with MSE loss function) delineate the baseline boundaries of the experiments. The solid lines (trained on reduced data with AMSE loss function) need to fall between the dashed lines to show the effectiveness of AMSE and those that are closer to the blue dashed line show better performance.

To summarize, we can see that AMSE is able to compensate for reduced annotations by allowing the network to train longer and learn better before overfitting. Choosing an exact β is more complex because it depends on different dr and data.

4.3. Wheat Spikelet Detection with AMSE

Spikelet detection is a more complicated task than crowd counting because it aims to *localize* each spikelet instead of directly counting the targets. We construct a 4-stack 2-block Stacked Hourglass Network to perform the spikelet detection task. In the following experiments, each model is trained on data of reduced annotations and tested on the fully-annotated test set every 5 training epochs. The location of each spikelet is calculated by non-maximum suppression over the predicted heatmap. The test results are presented in the F1 score used in the paper where the ACID dataset is explored [**?**]. To note, as F1 score is used here, higher results are better (opposite to the previous crowd counting experiments).

Similar to the experiments in crowd counting, we explore training a Stacked Hourglass Network (SHN) on the spikelet detection task (abbreviated to spikelet-SHN below) under $dr = 0.1, 0.3, 0.5, 0.7, 0.9$ with an AMSE loss function. Figure **??** and Table **??** compare the test F1 score between training with MSE and AMSE loss function. Here we did not apply AMSE to the fully-annotated dataset (i.e. $dr = 0$) because we assume the effect is similar to the crowd counting tasks.

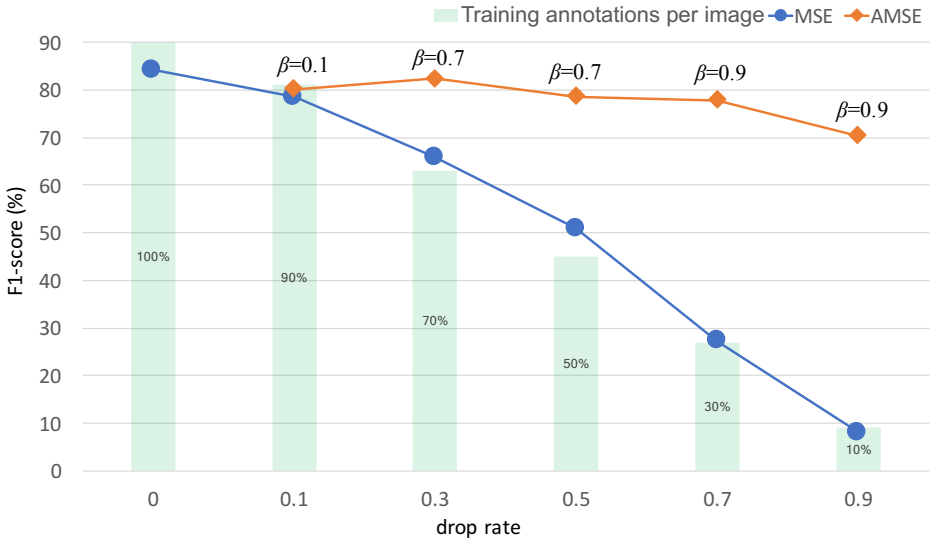


Figure 10. Test F1 score (higher is better) on spikelet detection under different drop rates. Blue line: results obtained by using MSE loss function (i.e. $\beta = 0$) in training. Orange line: best results obtained by using AMSE loss function in training, with β value above each point showing when the corresponding result is achieved. Green bar: what percentage of annotations per image is used in training (100% stands for fully-annotated data)

From Figure **??** we can clearly see that the test F1 score of the model trained with MSE (blue line) worsens as the drop rate increases, while the performance of the model trained with AMSE (orange line) still holds steady. From Table **??** we can calculate that with 30% and 50% annotations dropped, training with AMSE only experiences a 2.2% and 6.5% performance drop compared to the baseline ($dr = 0, \beta = 0$); even under the extreme case where 10% annotations are left (dr=0.9), the model trained with AMSE can still achieve 83.5% of the performance of the baseline, while the model trained with MSE shows far below acceptable performance. From Table **??** we can also observe that the optimal β tends to increase as dr grows, in a similar way to how it behaved in the crowd counting experiments. The nature of the increase is not linear, but here we do see a monotonic relationship.

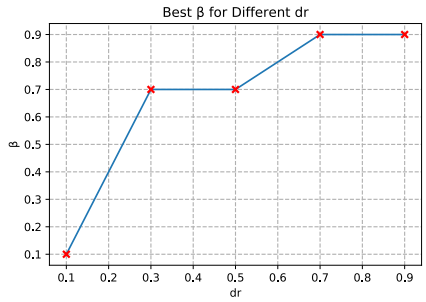
Table 4: Test F1 score(higher is better) on spikelet-SHN under different drop rates. Column under $\beta = 0$: results obtained by training with MSE loss function. Column under $\beta \neq 0$: best results obtained by training with AMSE loss function, with β value in bracket showing when the corresponding result is achieved.

Spikelet-SHN Test F1-score (various *dr*)

dr	Test F1-score (%)	
	β=0	$\beta\neq$0 (best β)
0	84.15	\
0.1	78.45	**80.19** (0.1)
0.3	65.89	**82.3** (0.7)
0.5	50.8	**78.65** (0.7)
0.7	27.24	**77.88** (0.9)
0.9	8.07	**70.29** (0.9)

We also show the test curve behaviour throughout the training process in Figure **??**. The models are tested after every 5 training epoch on unseen images with a fully-annotated

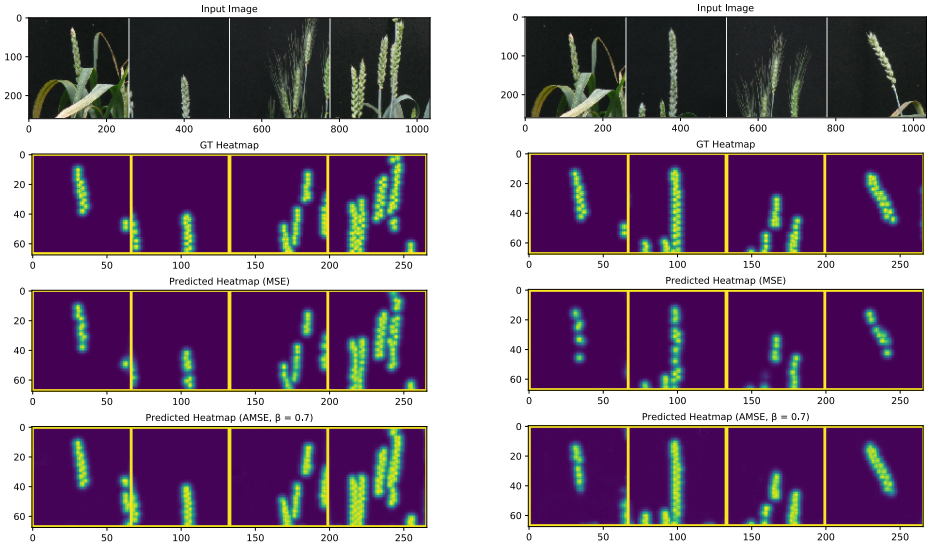
ground truth, and the presented curves are smoothed by averaging every 5 results. Figure **??** again confirms that AMSE is able to delay overfitting, and helps the models learn better on dropped annotations, with performance nearly always falling between the baseline best case ($dr = 0$, MSE) and baseline worse case (dropped data, MSE).



(a) $dr = 0.1$ (b) $dr = 0.3$ (c) $dr = 0.5$ (d) $dr = 0.7$ (e) $dr = 0.9$ (f) best β under different dr

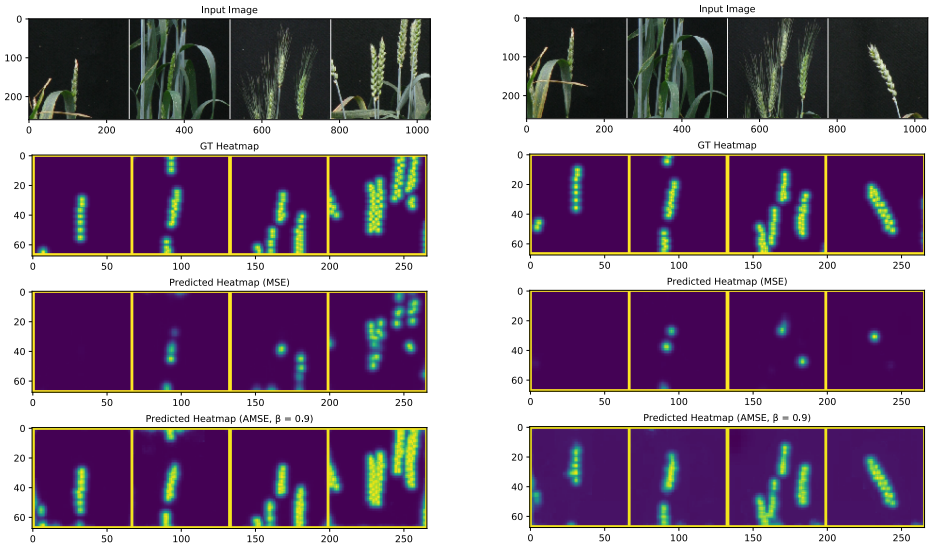
Figure 11. (a) – (e): test F1 scores of spikelet-SHN under $dr = 0.1, 0.3, 0.5, 0.7, 0.9$ respectively throughout different training stages. Each curve is smoothed by averaging every 5 test F1 scores. In each sub figure, the blue dashed line (trained on fully-annotated data with MSE loss function) and the orange dashed line (trained on reduced data with MSE loss function) delineate the baseline boundaries of the experiments. The solid lines are trained on reduced data with AMSE of different values of β (f): The best value of β for each drop rate.

As mentioned before, spikelet detection aims to obtain the accurate *position* of each spike (rather than simply a count), so here we present some graphical results in addition to the F1 scores. Figure **??** compares the predicted heatmaps on test data from the models trained with MSE and AMSE loss respectively, when $dr = 0.3, 0.5, 0.7, 0.9$. This figure shows that the models trained with MSE loss predict much fewer spikeliets compared to the ground truth especially under extreme cases where $dr \geq 0.5$ (2nd row versus 3rd row in sub figures. However, the predictions from the models trained with AMSE loss are surprisingly close the ground truth, even when the drop rate is very extreme at $dr = 0.9$ (2nd row versus 4th row in sub figures). The performance gap between using MSE and AMSE is clearly large when applying a relatively large dr to the data. In Figure **??** (d), we also can observe that the predicted heatmaps by the AMSE model (4th row) have a noticeably brighter background. This is a side effect when a large β is applied (e.g. $\beta = 0.9$), because the model tends to predict higher pixel values globally. However, it does not affect the predicted results. Applying a simple threshold method, or looking for local maximum, would be able to counteract the background, if necessary.



(a) $dr = 0.3$, best $\beta = 0.7$

(b) $dr = 0.5$, best $\beta = 0.7$



(c) $dr = 0.7$, best $\beta = 0.9$

(d) $dr = 0.9$, best $\beta = 0.9$

Figure 12. Inputs, ground truth and predicted heatmaps of spikelet detection under different dr. The first row in each sub figure shows the test images. The second row in each sub figure shows the ground truth (fully annotated). The third row in each sub figure shows the predictions by the model trained using MSE loss on the dropped data under the given dr. The fourth row in each sub figure shows the predictions by the model trained using AMSE loss with the optimal β under the given dr.

Based on the F1 scores and the graphical results, we conclude that training with AMSE can significantly improve the performance of models trained on the spikelet dataset of reduced annotations.

5. Conclusion and Discussion

This paper introduces a simple to understand, yet effective method to deal with missing annotations in object counting and detection by replacing the original loss function with AMSE using a weight parameter β. Missing annotations are a common complaint of annotation data in deep learning, but our method allows people to improve their models without modifying the fundamental structure of them when the ground truth is, or is

likely to be, incomplete. Moreover, our method may be able to simplify the annotation process itself because fewer annotations are needed for training a good model; in some case, annotators could be advised to annotate half of the data, for example, saving time and effort. Of course, the danger here would be that annotators only annotate clear, easy-to-see instances whilst not annotate hard examples, which would skew the training set, but this is not an issue in our experiments, as we randomly drop annotations from the original datasets. Future work should consider the effect of this skewing.

There is only one parameter to be chosen in this method – β - which is positively related to the drop rate of the data. Although the nature of this relationship does seem to vary depending on dataset, it appears reasonable to assume as dr increases, β should increase too. On a intentionally dropped dataset (i.e. the annotators are instructed to do so), dr is known, which simplifies the case; while in the cases where dr is not known (e.g. the annotators miss some annotations due to fatigue), dr could be estimated empirically by careful annotation of a small subset of the training data, and comparing this with the existing annotations. If the dr still cannot be measured, results indicate that a low β value will likely account for naturally missing annotations, without significant danger of making the approach perform worse. The downside of using the AMSE approach would seem to be a longer training time when β is large.

The results of our experiments on crowd counting and spikelet detection also demonstrate the effectiveness and robustness of our method, even when deleting 90% of annotations. When $dr \leq 0.5$ using AMSE in training can achieve comparable performance as training on fully-annotated dataset with MSE, which shows AMSE can potentially reduce annotation efforts. We also believe that our method is not restricted to object counting and detection; it could be expanded to other learning tasks; although this is left for future work.

Despite the effectiveness of the AMSE method, there are two aspects that could be developed further. First, estimating the true drop rate dr of a "fully-annotated" dataset is currently challenging. It may be possible to estimate for a given dataset by using statistical analysis or machine learning methods to interrogate the properties of the training data. Second, estimating the optimal β in a given domain is left to the user. One could empirically determine the best β for the AMSE loss by trial and error on a subset of a dataset. We already can see that the best β seems positively related to dr, but the value of *optimal* β varies in different data and models. This may require further mathematical exploration on the patterns of AMSE under different drop rates. At the current stage, we advise interested readers to start from a relatively small β (e.g. $\beta = 0.1$) on a real dataset (i.e. the annotators are instructed to fully annotate the data but they miss some annotations due to fatigue), and then increase the value gradually to improve a deep learning model. We have shown that a small β is able to benefit the model even if the annotations are not dropped manually in Section **??**. This is likely because a small number of missing annotations is intuitively common in real world data; and of course we do not know the true baseline drop rate of the dataset we make use of in this work.

To address the challenge of missing annotation data then, we have shown that an AMSE loss can, when properly tuned, achieve close to or even better than no-dropout baseline results even in the presence of substantial dropped annotations. Even without tuning, a small value of β still leads to better performance than using MSE on dropped dataset. This result provides tolerance to missing annotations, and potentially more practical annotation guidelines, where 100% annotation coverage is not required.

Author Contributions: Conceptualization and methodology, F.C., M.P. and A.F.; conducting experiments and validation, F.C.; formal analysis, F.C.; writing—original draft preparation, F.C.; writing—review and editing, M.P. and A.F.; supervision, M.P. and A.F. All authors have read and agreed to the published version of the manuscript.

Funding: F.C. was funded by a School of Computer Science (University of Nottingham) studentship.

Data Availability Statement: The ShanghaiTech crowd counting dataset and the ACID wheat dataset can be found in [**?**] and [**?**] respectively.

Acknowledgments: We want to thank School of Computer Science, University of Nottingham for providing computational resources. We also want to thank the authors of [**? ? ? ?**] for making their data or codes public.

Conflicts of Interest: The authors declare no conflict of interest.

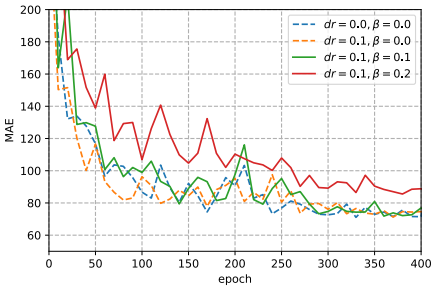
Abbreviations

The following abbreviations are used in this manuscript:

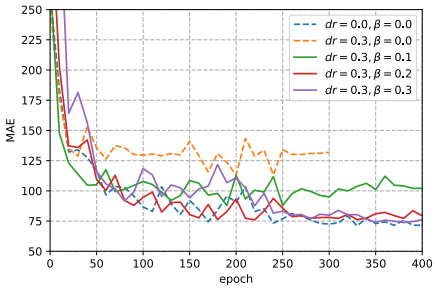
AMSE	Asymmetric Mean Squared Error (as loss function)
MSE	Mean Squared Error (as loss function)
MAE	Mean Absolute Error (as metric)
RMSE	Root Mean Squared Error (as metric)
CAN	Context-Aware Network
SHN	Stacked Hourglass Network
ShA	ShanghaiTech Part A (crowd counting dataset)
ShB	ShanghaiTech Part B (crowd counting dataset)

16

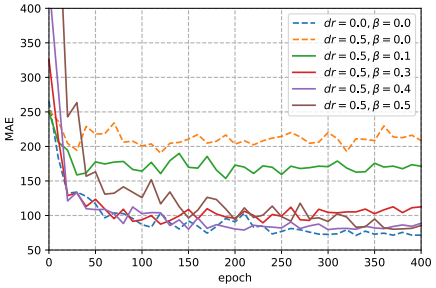
Appendix A. Full Test Curves of ShA/ShB-CAN



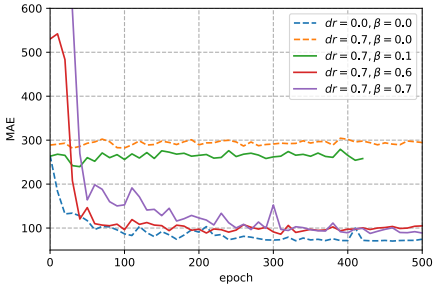
(a) $dr = 0.1$



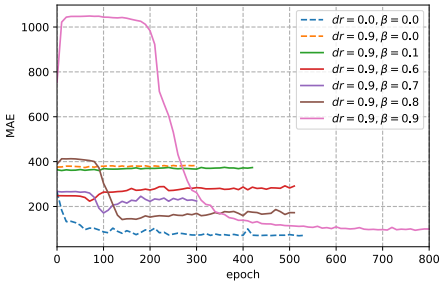
(b) $dr = 0.3$



(c) $dr = 0.5$

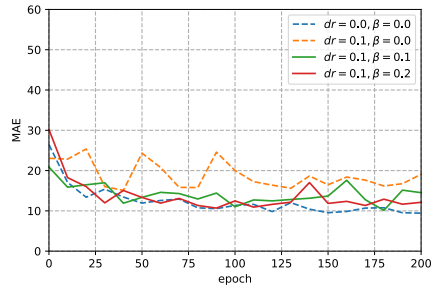


(d) $dr = 0.7$

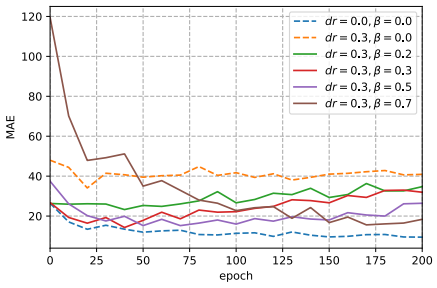


(e) $dr = 0.9$

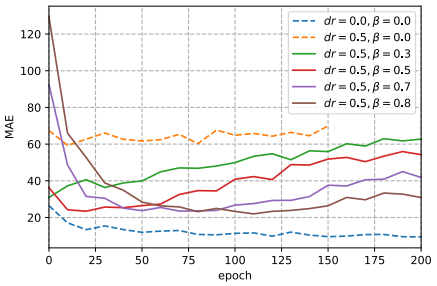
Figure A1. Test MAE of ShA-CAN under various dr across different training stages: (a) Test MAE of ShA-CAN trained with $dr = 0.1$; (b) Test MAE of ShA-CAN trained with $dr = 0.3$; (c) Test MAE of ShA-CAN trained with $dr = 0.5$; (d) Test MAE of ShA-CAN trained with $dr = 0.7$; (e) Test MAE of ShA-CAN trained with $dr = 0.9$. The results are averaged every 10th epoch. In each sub-graph, the blue dashed line (trained on fully-annotated data with MSE loss function, i.e. $\beta = 0$) and the orange dashed line (trained on reduced data with MSE loss function, i.e. $\beta = 0$) delineate the baseline boundaries of the experiments. The solid lines (trained on reduced data with AMSE loss function, i.e. $\beta \neq 0$) need to fall between the dashed lines to show the effectiveness of AMSE and those that are closer to the blue dashed line show better performance.



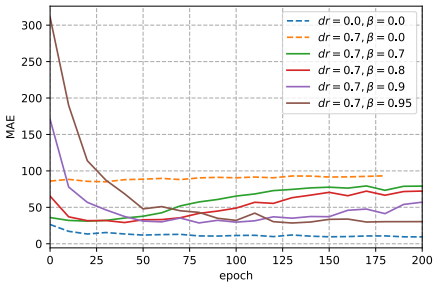
(a) $dr = 0.1$



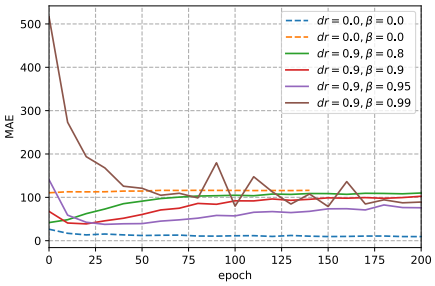
(b) $dr = 0.3$



(c) $dr = 0.5$



(d) $dr = 0.7$



(e) $dr = 0.9$

Figure A2. Test MAE of ShB-CAN under various dr across different training stages: (a) Test MAE of ShB-CAN trained with $dr = 0.1$; (b) Test MAE of ShB-CAN trained with $dr = 0.3$; (c) Test MAE of ShB-CAN trained with $dr = 0.5$; (d) Test MAE of ShB-CAN trained with $dr = 0.7$; (e) Test MAE of ShB-CAN trained with $dr = 0.9$. The results are averaged every 10th epoch. In each sub-graph, the blue dashed line (trained on fully-annotated data with MSE loss function, i.e. $\beta = 0$) and the orange dashed line (trained on reduced data with MSE loss function, i.e. $\beta = 0$) delineate the baseline boundaries of the experiments. The solid lines (trained on reduced data with AMSE loss function, i.e. $\beta \neq 0$) need to fall between the dashed lines to show the effectiveness of AMSE and those that are closer to the blue dashed line show better performance.

References

. Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; Ma, Y. Single-image crowd counting via multi-column convolutional neural network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

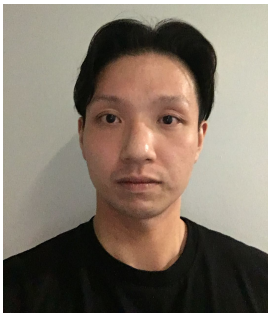
. Liu, W.; Salzmann, M.; Fua, P. Context-aware crowd counting. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

. Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. Proceedings of the Springer European Conference on Computer Vision Workshop (ECCV), 2016.

. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

. Giuffrida, M.V.; Chen, F.; Scharr, H.; Tsaftaris, S.A. Citizen crowds and experts: observer variability in image-based plant phenotyping. *Plant methods* **2018**, *14*, 12.
. Li, S.; Deng, W. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Unconstrained Facial Expression Recognition. *IEEE Transactions on Image Processing (T-IP)* **2019**, *28*, 356–370. doi:10.1109/TIP.2018.2868382.
. Li, S.; Deng, W.; Du, J. Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization, 2016, [arXiv:cs.LG/1611.03530].
. Krizhevsky, A.; Nair, V.; Hinton, G. CIFAR-10 (Canadian Institute for Advanced Research).
. Krueger, D.; Ballas, N.; Jastrzebski, S.; Arpit, D.; Maharaj, T.; Bengio, E.; Fischer, A.; Courville, A. Deep Nets Don't Learn via Memorization. Proceedings of International Conference on Learning Representations (ICLR) Workshops, 2017.
. Arpit, D.; Jastrzębski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M.S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; Lacoste-Julien, S. A Closer Look at Memorization in Deep Networks, 2017, [arXiv:stat.ML/1706.05394].
. Goldberger, J.; Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. Proc. International Conference on Learning Representations (ICLR), 2017.
. Northcutt, C.G.; Jiang, L.; Chuang, I.L. Confident Learning: Estimating Uncertainty in Dataset Labels, 2019, [arXiv:stat.ML/1911.00068].
. Jiang, L.; Zhou, Z.; Leung, T.; Li, L.J.; Fei-Fei, L. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels, 2017, [arXiv:cs.CV/1712.05055].
. Li, J.; Wong, Y.; Zhao, Q.; Kankanhalli, M.S. Learning to learn from noisy labeled data. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
. Rebuffi, S.A.; Ehrhardt, S.; Han, K.; Vedaldi, A.; Zisserman, A. Semi-Supervised Learning with Scarce Annotations, 2019, [arXiv:cs.CV/1905.08845].
. Lu, Z.; Fu, Z.; Xiang, T.; Han, P.; Wang, L.; Gao, X. Learning from Weak and Noisy Labels for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)* **2017**, *39*, 486–500. doi:10.1109/TPAMI.2016.2552172.
. Pound, M.P.; Atkinson, J.A.; Wells, D.M.; Pridmore, T.P.; French, A.P. Deep learning for multi-task plant phenotyping. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCV), 2017.

Short Biography of Authors



Feng Chen is a PhD student in Computer Science, University of Nottingham, supervised by Prof. Andrew French and Prof. Tony Pridmore. His research has included low-cost deep learning, active learning, crowd sourcing, and image-based plant phenotyping.

Michael P. Pound is an Assistant Professor in Computer Science, University of Nottingham. He graduated with a PhD in Computer Science in 2011, and since then has worked on the application of computer vision to plant-phenotyping problems. His recent research has included work on 3D reconstruction of plants, and deep learning solutions for the segmentation of 2D and 3D data.

Andrew P. French is an Associate Professor at the University of Nottingham, UK. He is a computer scientist by training, graduating with a PhD in Computer Science in 2005, who now develops AI-based computer vision solutions to biological problems. Specifically, he contributes to a number of plant phenotyping projects, developing automated ways of measuring plant traits. Currently this involves approaches from computer vision and deep learning.