Learning to Localise and Count with Incomplete Dot-annotations

Feng Chen[1] Michael P. Pound[1] Andrew P. French[1, 2]
[1]School of Computer Science, University of Nottingham, NG8 1BB, U.K.
[2]School of Biosciences, University of Nottingham, LE12 5RD, U.K.
feng.chen, michael.pound, andrew.p.french@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 localisation and counting), by applying an asymmetric loss function. This reduction of annotations can be imposed by the researchers by asking annotators to intentionally label only 50% of what they see in each image - a form of 'few-click' annotation. Our method also has a secondary benefit of counteracting unintentionally missing labels from the annotators. We conduct experiments on wheat spikelet localisation and crowd counting to assess the effectiveness and robustness of our method. 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 is required, we find that setting conservative parameter values can help more realistic situations, where only small amounts of data have been missed by annotators.

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 using less labeled data, for example, semi-supervised learning, has become increasingly popular. Although some methods have achieved promising results, they tend to focus on image classification, detection and semantic segmentation; regression-based counting and localisation tasks are under explored. Different from semi-supervised learning, which often focuses on reducing the *number* of labelled images, 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 neural networks for heatmap regression tasks using fewer annotation clicks *per image* – replacing the original loss function with an Asymmetric Mean Square Error (AMSE) loss function.

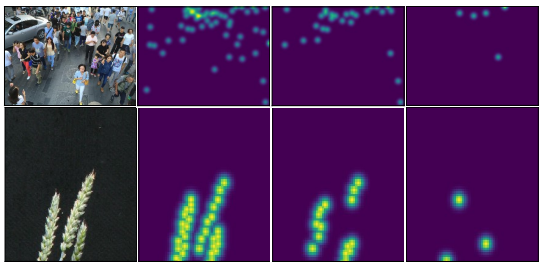


Figure 1. A graphical overview of reduced annotations: upper row: ShanghaiTech Part B [24], bottom row: ACID wheat dataset [17]. (a) input image, (b) original instance heatmap image, (c) and (d) show the same heatmap as (b) but with 50% and 90% of annotations dropped respectively. Our aim is to train a well-performing network using dropped annotations as represented in (c) and (d).

Mean Square Error (MSE) is widely used as a loss function in many heatmap regression tasks, like object localisation and counting. A heatmap is normally constructed by many Gaussian-like regions, and MSE can guide the network to predict such regions. However, if some of the ground-truth point annotations are removed, the network will easily overfit to the remaining area. Under this situation, AMSE can force the network to enhance the response from the false-negative areas (i.e. areas where targets are incorrectly unnanotated) and deal with class imbalance between background and targets, when the annotations are significantly reduced. We conduct experiments of AMSE on annotation-reduced versions of ACID dataset [17] using Stacked Hourglass Network [14] to perform wheat spikelet localisation; a crowd counting dataset (ShanghaiTech [24]) is further explored with Context-aware Network ([12]) to demonstrate the robustness of our method in an even more

crowded case. 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; and we tune the hyperparameter of AMSE (β) to adapt to different drop rates.

Our main contributions are: 1) Introducing a novel and efficient annotation concept, which helps account for accidental or purposeful partial annotation per image, 2) A description and evaluation of the proposed AMSE loss function to counteract performance loss from this reduced number of annotations per image, 3) An exploration of the sensitivity of the single tuning parameter β of AMSE over two diverse datasets and two deep neural network architectures 4) Demonstration of competitive performance on public counting and localisation datasets (both plant- and crowd-related) in the presence of substantially (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 perform better than the model trained on the same reduced dataset with MSE loss. However, as we explore, selection of β and management of the training regime is key.

2. Related Work

The quality and quantity of training data and its annotations are of course crucial in deep learning. The annotation process is expensive and time-consuming. Most existing work focuses on reducing the price of acquiring annotations (e.g. crowdsourcing [22]) and using easier annotations (e.g. weak supervision [25]).

Crowdsourcing has become a popular method to acquire labels due to the increasing demand for large image datasets, especially after the Imagenet dataset [3] was released. However, imperfect labels are commonly generated from crowdsourcing. They are often produced by non-expert users, leading to a need to suppress or handle noisy or missing labels [4, 10, 11].

As defined in [25], *weak supervision* includes incomplete supervision (i.e. just a few instances are labelled), inexact supervision (i.e. only coarse-grained annotations are provided) and inaccurate supervision (i.e. labels contain errors). *Incomplete* supervision is often explored with semi-supervised learning [19]. For example, Rebuffi et al. [18] combine semi-supervised learning with self-supervision to train models on scarcely-annotated datasets. *Inexact* supervision often reduces the difficulties of acquiring labels, for example, providing only image-level labels (e.g. total count or class labels) instead of dot-annotations on every interesting object to perform object counting [21, 2] or semantic segmentation [13]. The typical scenarios of *inaccurate* supervision are noisy and missing labels, of which some fundamental experiments of how deep neural networks react to them were carried out recently. For example, Zhang et al. [23] swap and delete labels on different datasets including CIFAR10 [7] and Imagenet [3] to demonstrate that deep models can learn the general patterns of the data in early epochs but soon overfit the local patterns. Krueger [8] and Arpit [1] find that deep neural networks can practically memorize all training information including noisy (i.e. misclassified) and missing labels. At the application level, Jiang et al. [6] 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. [9] 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. Goldberger and Ben-Reuven [5] use the Expectation Maximization (EM) algorithm to estimate the correct labels and retrain the network after a certain number of epochs. Based on the assumption of the dependency between the noise and true labels, Northcutt et al. [16] use confident learning to estimate label errors and refine the network.

Our problem, which has not been widely explored, is a form of inexact supervision, while focusing on incomplete or missing annotations *per image*. This is different from what often happens in semi-supervised learning where the number of labelled images is decreased. Attempts to solve a similar problem have been made, for example, Nguyen et al. [15] propose a loss function to dynamically integrate the object and background pixels to perform comic speech balloon segmentation with incomplete labels per image, and Wang et al. [20] propose to propagate the incomplete bounding box per image using a positiveness-focused object detector to perform detection-based object counting. Here we focus on heatmap-based object localisation and counting, and the number of the objects of interest per image is generally much larger than the mentioned approach. Our method is effective yet easy to apply: as we will show, it is architecture-independent and only has one tuning hyper parameter.

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) *accidental* 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 exactly how many annotations are missed on each image as missing annotations are the result of unknown errors. The intentional scenario is more likely to produce more extreme missing data amounts, whereas the accidental scenario is more likely to produce more conservative drops in annotation data. Our method is valuable for both of these scenarios, as we explore using differing data drop amounts.

Our approach is developed to tackle deep learning-based heatmap regression problems, for object localisation and counting, on 2D RGB images. These images are dot-annotated, from which a Gaussian kernel is applied to generate the heatmaps to represent locations and densities. We assume that there can be different drop rates $dr \in [0, 1]$ applied to the labelled 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 (i.e. complete annotations); 2) MSE methods with different drop rates. Figure 1 illustrates heatmaps with different drop rates for comparison.

3.2. AMSE V.S. MSE

In this paper, we focus on adapting the MSE loss function to missing-annotation scenarios, as MSE is commonly used in heatmap regression problems. We believe the asymmetric form may also be applied to other loss functions so that they can adapt to missing annotation scenarios, although we do not examine this here. Mean Square Error (MSE) 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.

We use an Asymmetric MSE (AMSE) to counteract the effect of the reduced (missing) annotations. AMSE is expressed as:

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

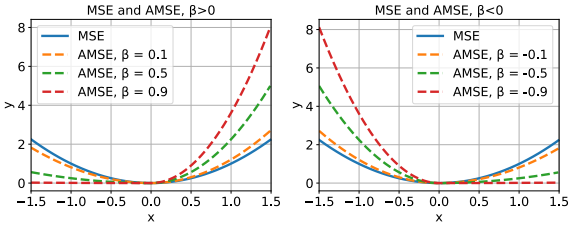


Figure 2. A comparison between MSE and AMSE. In both graphs, the blue solid line represents MSE, and the dashed lines show the behaviour of AMSE under different weights (noted as β). The graph on the left shows AMSE under $\beta > 0$, where the left part of each curve is suppressed, while the graph on 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.

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 2: 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 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. To note, when β in AMSE is set to 0, the AMSE is equivalent to traditional MSE.

We assume M annotations in the original labeled file, while in the case of reduced annotations, $dr * M$ annotations are dropped (deleted) so only $(1 - dr) * M$ annotations are left. A ground-truth heatmap is constructed from the original labelled file (e.g. dot-annotated image) to some Gaussian-like (hot-spot) areas (values within these areas > 0), and the remaining 0-valued background region. Reducing annotations creates a lot of false negatives because some desired areas (i.e. the hot-spot areas) are treated as background. This causes locations to be lost and counts underestimated in object localisation and counting tasks respectively. Therefore, the model is expected to *overestimate* the predictions especially in the false-negative areas. 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 2 equal to $(Y_i - \hat{Y}_i)$, we want to allocate a greater loss for $(Y_i - \hat{Y}_i) > 0$; and when $(Y_i - \hat{Y}_i)$ grows, a larger loss should be applied. This will guide the network to learn a larger $\hat{Y}_i$. 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 2, 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 wheat spikelet localisation dataset, and a crowd counting dataset. The ACID (Annotated Crop Image Database) dataset [17] is constructed from 520 wheat plant images, with their 'spikelets' (individual wheat grains) and ears annotated, while in our experiments, only the annotations on spikelets are adopted. A fixed-size Gaussian kernel is applied to these annotations, to generate the ground-truth heatmaps. ShanghaiTech [24] 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 ([24, 12]), a geometry-adaptive kernel is adopted to generate the density maps for part A and a fixed kernel is used for part B.

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 3. 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 where 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. 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 dropped annotations we want the networks to localise or count all instances if possible. We apply different AMSE weights β to the datasets with various drop rates dr to explore the relationship between β and dr.

We use a Stacked Hourglass Network (SHN [14]) to perform spikelet localisation on ACID dataset [17]. We adopt a similar SHN structure as in [17], which is four-stacked with intermediate supervison, except the classification head. We use the original structure of Context-aware Network (CAN [12]) to perform crowd counting on ShanghaiTech dataset [24] as additional experiments. 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 5 epochs for spikelet localisation and every epoch for crowd counting against test data which is fully-annotated and unseen in training.

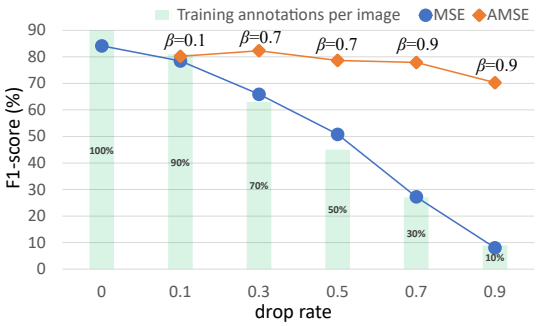
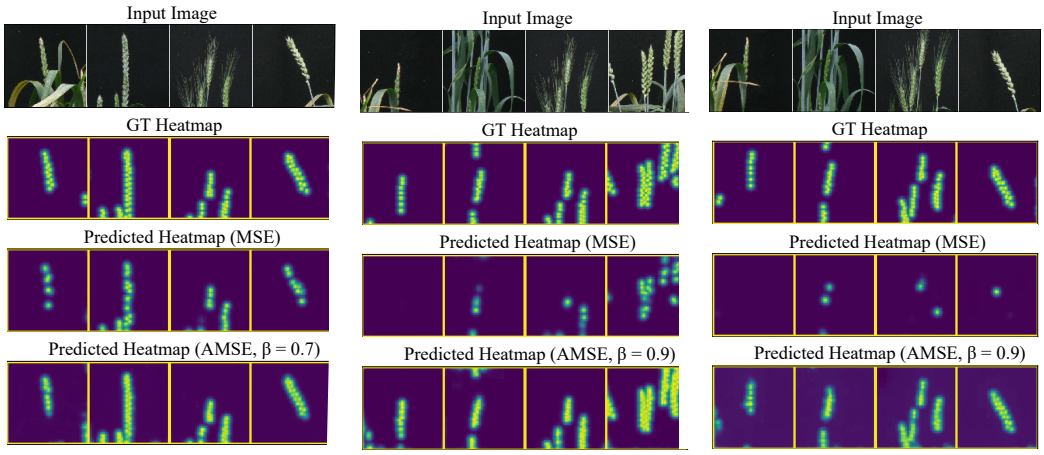


Figure 3. Test F1 score (higher is better) on spikelet detection under different drop rates. Blue line: results obtained by using MSE loss function (i.e. $\beta = 0$) in training. Orange line: best results obtained by using AMSE loss function in training, with β value above each point showing when the corresponding result is achieved. Green bar: what percentage of annotations per image is used in training (100% stands for fully-annotated data)

4.2. Wheat Spikelet Localisation with AMSE

We explore training an SHN on the spikelet localisation task (abbreviated to spikelet-SHN below) under $dr = 0, 0.1, 0.3, 0.5, 0.7, 0.9$. The location of each spikelet is calculated by non-maximum suppression over the predicted heatmap. The test results are presented in F1 score (higher results are better) as used in [17]. Figure 3 and Table 1 compare the test F1 score between training with MSE and AMSE loss function. From Figure 3 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 1 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. 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. From Table 1 we can also observe that the optimal β tends to increase as dr grows; the nature of the increase is not linear, but here we do see a monotonic relationship.



(a) $dr = 0.5$, best $\beta = 0.7$ (b) $dr = 0.7$, best $\beta = 0.9$ (c) $dr = 0.9$, best $\beta = 0.9$

Figure 4. Inputs, ground truth and predicted heatmaps of spikelet detection under different dr. In each sub-figure: first row: test images, second row: ground-truth heatmaps (fully annotated), third/fourth row: predictions by the model trained on the dropped data using MSE loss and AMSE loss (with optimal β) respectively.

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)

Table 1. 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 localisation aims to obtain the accurate *position* of each spike, so here we present some graphical results in addition to the F1 scores. Figure 4 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 (2nd row versus 3rd row in sub-figures) especially under extreme cases where $dr \geq 0.5$. However, the predictions from the models trained with AMSE loss are surprisingly close to the ground truth (2nd row versus 4th row in sub-figures), even when the drop rate is very extreme at $dr = 0.9$; the performance gap between using MSE and AMSE is huge. In Figure 4 (c), we can also 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.

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 per image; although clearly we must choose a value for β, and observe a suitable training period.

4.3. Crowd Counting with AMSE

We also explore training with AMSE on a general problem, that of crowd counting, using the ShanghaiTech dataset and the Context-aware Network (CAN). The ShanghaiTech dataset contains A and B parts (ShA and ShB); ShA is more crowded than ShB (average count: 501.4 vs 123.6). The images in both ShA and ShB have more complicated background and contains much more objects than ACID (average count: 92.3), so they are useful to demonstrate the robustness and adaptability of AMSE. In the following 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). MAE is defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}_i|, \tag{4}$$

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

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 ($dr = 0$)

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

Table 2. 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.

and RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}, \tag{5}$$

where N denotes the total number of images, Y_i denotes the true number of crowd counts in image i and $\hat{Y}_i$ denotes the predicted crowd counts of the same image.

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 2. 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 2, 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.

We next 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 5 (a) and (b) compares the test MAE score using MSE and AMSE loss function in training on ShA-CAN and ShB-CAN respectively. Table 3 provides the corresponding numerical results (reporting MAE and RMSE), showing the best test MAE/RMSE achieved for each dr and the corresponding β when it was achieved using our method. As a comparison, the results achieved by training with the original MSE loss function are shown in the column labeled $\beta = 0$.

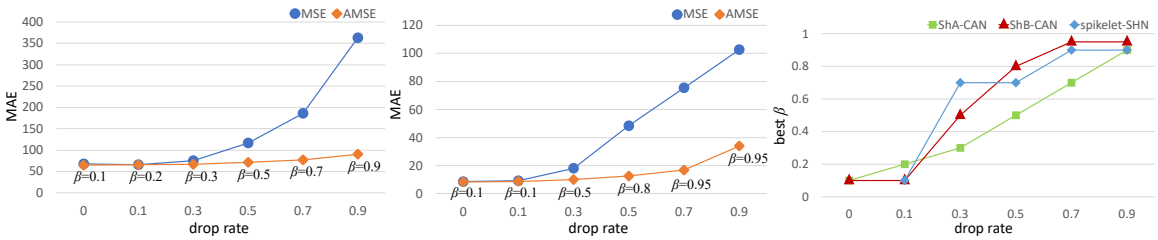
dr	Test MAE				Test RMSE			
	β=0		$\beta\neq0$ (best β)		β=0		$\beta\neq0$ (best β)	
	ShA	ShB	ShA	ShB	ShA	ShB	ShA	ShB
0	67.3	8.58	**64.72** (0.1)	**8.36** (0.1)	103.5	13.72	**99.6** (0.1)	**13.26** (0.1)
0.1	65.71	9.21	**65.66** (0.2)	**8.78** (0.1)	98.91	14.79	**98.91** (0)	**14.39** (0.1)
0.3	75.34	18.13	**66.65** (0.3)	**10.04** (0.5)	113.56	19.71	**97.63** (0.3)	**15.86** (0.3)
0.5	116.87	48.42	**71.44** (0.5)	**12.57** (0.8)	188.52	68.16	**107.39** (0.5)	**17.89** (0.8)
0.7	186.26	75.5	**77.27** (0.7)	**16.87** (0.95)	186.26	99.06	**114.82** (0.7)	**26.81** (0.95)
0.9	363.15	102.72	**90.48** (0.9)	**34.04** (0.95)	363.15	133.75	**141.39** (0.9)	**50.8** (0.95)

Table 3. Test MAE/RMSE (lower is better) on ShA/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.

From Figure 5 (a) 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. From Table 3 (columns under ShA) 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 ($dr = 0, \beta = 0$). 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.

Similar trends are also shown in ShB-CAN test results. From Figure 5 (b) it can be observed that when $dr \geq 0.3$ the test MAE achieved by AMSE exceed those obtained by MSE loss function; and the gap becomes larger when $dr \geq 0.5$. From Table 3 (columns under ShB) 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.

From the test results of ShA/ShB-CAN on various dr, we can conclude that AMSE outperforms MSE in the scenario of reduced annotations, even on a more crowded dataset. As an example, on this dataset, 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 can still fall within an acceptable area until dr becomes extreme (i.e. $dr > 0.7$). If we plot dr against β of the best test MAE achieved (Figure 5 (c)), we can note in general that the best β value increases as dr grows larger in ShA/ShB-CAN; similar trend is also observed in the previous spikelet



(a) ShA-CAN (b) ShB-CAN (c) optimal β under different dr

Figure 5. (a, b) Test MAE (lower is better) on ShA-CAN and ShB-CAN respectively under different drop rates. Blue line: results obtained by using MSE loss (i.e. $\beta = 0$). Orange line: best results obtained by using AMSE loss, with β value under each point showing when the corresponding result is achieved. (c) The best value of β under each drop rate for ShA/ShB-CAN and Spikelet-SHN

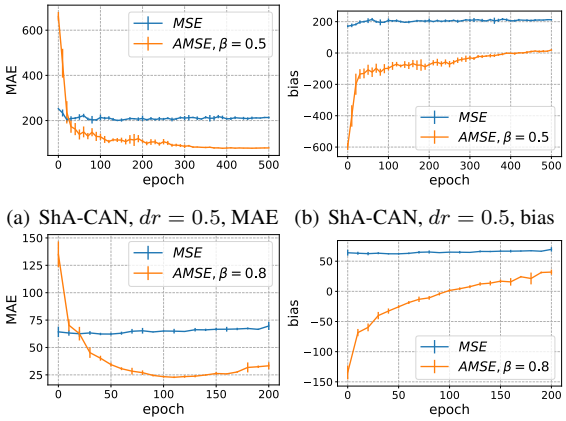
experiments.

To further explore the robustness of our method in crowded scenes, and to reveal a potential mechanism of AMSE when dealing with a dropped dataset, we conduct repeated experiments on ShA/ShB-CAN for a fixed drop rate, $dr = 0.5$. First we generate 5 "folds" of dropped dataset using different random seeds, from ShA and ShB respectively. For each of these folds, we essentially drop a different random subset of the annotations per image. We then train a model using MSE and AMSE with the best performing β noted from earlier experiments above (i.e. $\beta = 0.5$ for ShA and $\beta = 0.8$ for ShB) on these folds. Finally, we test the models on the corresponding unseen data and report their average error, bias and standard deviation. These results are shown in Figure 6. We introduce *bias*, to better reveal the mechanism of AMSE. Bias is defined as:

$$bias = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i), \tag{6}$$

where N denotes the total number of images, Y_i denotes the true number of crowd counts in image i and $\hat{Y}_i$ denotes the predicted crowd counts of the same image. Essentially this gives us a measure of over or under-estimation.

From Figure 6 panels (a) and (c) which report Test MAE, we can observe that our method (AMSE, orange line) is robust to different random samples of annotations as the standard deviation of MAE is small; our method also consistently outperforms MSE (blue line), after sufficient training, using the same dropped dataset. To understand Figure 6 panels (b) and (d) we need to introduce what bias represents. We define bias as Equation (6); $bias < 0$ means the predicted count is greater than the true count (i.e. prediction is *overestimated*) while $bias > 0$ represents the opposite (*underestimation*). Therefore, from Figure 6 panels (b) and (d) which report Test bias, we observe that: our method (AMSE, orange line) starts from predicting overestimated counts and gradually forces the model to predict true counts during training, before it overfits to the dropped training data; while the model trained with MSE (blue line) overfits



(a) ShA-CAN, $dr = 0.5$, MAE (b) ShA-CAN, $dr = 0.5$, bias

(c) ShB-CAN, $dr = 0.5$, MAE (d) ShB-CAN, $dr = 0.5$, bias

Figure 6. Average test MAE/bias of ShA/ShB-CAN on $dr = 0.5$ from repeated experiments. Under each parameter set, the curves show the mean and standard deviation (error bars) of 5 repeated experiments, with various random seeds to generate different dropped training data. All the curves are smoothed after every 10 epochs. (a)/(b) Average test MAE/bias of ShA-CAN on $dr = 0.5$ with MSE and AMSE of $\beta = 0.5$ (i.e. optimal *beta*, below as the same); (c)/(d) Average test MAE/bias of ShB-CAN on $dr = 0.5$ with MSE and AMSE of $\beta = 0.8$.

to the dropped data at the very beginning, and consequently predicts steady underestimated counts throughout training. We can conclude from this exploration that AMSE operates by helping to reduce overestimates over time, as training progresses. However, what is key to note is that over training can lead to underestimation (as with MSE) when the model is allowed to overfit the training data. Therefore an important point is raised: as shown in our experiments, training time, as well as an appropriate *beta*, needs to be optimised per scenario. Training should be monitored for a sufficient amount of time, then select the model with lowest validation error - rather than simply training for a fixed number of epochs and selecting the final model as the best. This can be used to identify the zero-crossing point for the

bias - where the model is optimally trained.

We have demonstrated the mechanism and potential robustness of our method on crowd counting datasets, now we explore the properties of AMSE through some specific scenarios. Figure 7 presents test MAE curves of ShA/ShB-CAN under dr=0.3 and 0.9, which mimics a realistic case of missing annotations, 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. From Figure 7 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 7 (b), which shows test results of ShA-CAN under $dr = 0.9$, the pink solid line ($\beta = 0.9$) clearly converges slower 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 7 (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 still generally better than the orange dashed line (using MSE loss function).

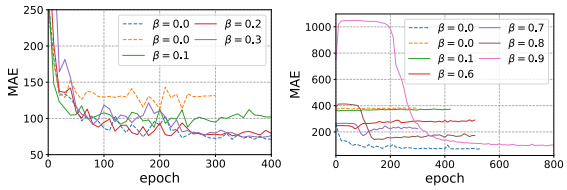
To summarize, we see that AMSE is able to compensate for reduced annotations under very crowded scenes and even with very extreme drop rates (e.g. $dr = 0.7, 0.9$), by allowing the network to train longer and learn better before overfitting.

5. Conclusion

The results of our experiments on spikelet localisation and crowd counting demonstrate the effectiveness and robustness of our method, even when missing 90% of annotations. When $dr \leq 0.5$ using AMSE in training can achieve comparable performance as using a fully-annotated dataset with MSE, demonstrating AMSE could be used to reduce annotation effort. The approach requires one parameter to be chosen – β - which we show is positively related to the drop rate of the data, though the nature of this relationship does vary depending on dataset. It also requires an appropriate training time regime to be set per-domain.

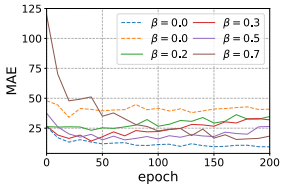
On an intentionally-dropped dataset (i.e. the annotators are instructed to do so), dr is known, which simplifies the case; while in the cases where dr is not known (e.g. the annotators miss some annotations due to fatigue), dr could be estimated empirically by careful annotation of a small subset of 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 realistically-missing annotations, without significant danger of adversely affecting performance. We believe that our method is not restricted to object localisation and counting; it could be expanded to other learning tasks. The idea of intentionally reducing annotations *per image* rather than reducing labeled images as used in common semi-supervised learning is also interesting to explore, particularly when the appearance or pattern of the interested objects is similar (e.g. spikelets and human heads) but the number is large – this can potentially create a more practical and efficient annotation guideline.

In conclusion, with careful selection of β and a suitable training regime, AMSE can reduce the annotation cost in counting and localisation scenarios.

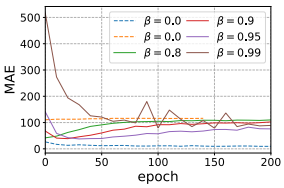


(a) ShA-CAN, $dr = 0.3$

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



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



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

Figure 7. Test MAE (lower is better) of ShA-CAN (upper row) and ShB-CAN (bottom row) under $dr = 0.3, 0.9$ respectively throughout different training stages (y-axis: MAE, x-axis: epoch). 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) 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.

6. Acknowledgements

This work was supported by School of Computer Science, University of Nottingham, U.K., via studentship funding.

References

[1] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. 2

[2] Enrico Bellocchio, Thomas A. Ciarfuglia, Gabriele Costante, and Paolo Valigi. Weakly supervised fruit counting for yield estimation using spatial consistency. *IEEE Robotics and Automation Letters*, 4(3):2348–2355, 2019. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

[4] M Valerio Giuffrida, Feng Chen, Hanno Scharr, and Sotirios A Tsaftaris. Citizen crowds and experts: observer variability in image-based plant phenotyping. *Plant methods*, 14(1):12, 2018. 2

[5] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *Proc. International Conference on Learning Representations (ICLR)*, 2017. 2

[6] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2304–2313. PMLR, 2018. 2

[7] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 2

[8] David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, and Aaron Courville. Deep nets don't learn via memorization. In *Proceedings of International Conference on Learning Representations (ICLR) Workshops*, 2017. 2

[9] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[10] S. Li and W. Deng. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Transactions on Image Processing (T-IP)*, 28(1):356–370, Jan 2019. 2

[11] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[12] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 4

[13] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao. Learning from weak and noisy labels for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 39(3):486–500, March 2017. 2

[14] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the Springer European Conference on Computer Vision (ECCV) Workshops*, 2016. 1, 4

[15] Nhu-Van Nguyen, Christophe Rigaud, Arnaud Revel, and Jean-Christophe Burie. A learning approach with incomplete pixel-level labels for deep neural networks. *Neural Networks*, 130:111–125, Oct. 2020. 2

[16] Curtis G Northcutt, Lu Jiang, and Isaac L Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 2021. 2

[17] Michael P Pound, Jonathan A Atkinson, Darren M Wells, Tony P Pridmore, and Andrew P French. Deep learning for multi-task plant phenotyping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2017. 1, 4

[18] Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Semi-supervised learning with scarce annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 762–763, 2020. 2

[19] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020. 2

[20] Jianfeng Wang, Rong Xiao, Yandong Guo, and Lei Zhang. Learning to count objects with few exemplar annotations. *CoRR*, abs/1905.07898, 2019. 2

[21] Yifan Yang, Guorong Li, Zhe Wu, Li Su, Qingming Huang, and Nicu Sebe. Weakly-supervised crowd counting learns from sorting rather than locations. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, pages 1–17. Springer, 2020. 2

[22] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. A survey of crowdsourcing systems. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 766–773, 2011. 2

[23] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017. 2

[24] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 4

[25] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 08 2017. 2