EasyChair Preprint

№ 10722

Improved Knowledge Distillation for Crowd Counting on IoT Devices

Zuo Huang and Richard Sinnott

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

August 15, 2023

Improved Knowledge Distillation for Crowd Counting on IoT Devices

***Abstract*—Manual crowd counting for real world problems is either impossible and/or results in wildly inaccurate estimations. Deep learning is one area that has been applied to address this issue. However crowd counting is a computationally intensive task. Many crowd counting models employ large-scale deep convolutional neural networks (CNN) to achieve higher accuracy, however these are typically at the cost of performance and inference speed. This makes such approaches difficult to apply in real world settings, e.g., on Internet-of-Things (IoT) devices. To tackle this problem, one method is to compress models using pruning and quantization or use of lightweight model backbones. However, such methods often result in a significant loss in accuracy. To address this, some studies have explored knowledge distillation methods to extract useful information from large state-of-the-art (teacher) models to guide/train smaller (student) models. However, knowledge distillation methods suffer from the problem of information loss caused by hint-transformers. Furthermore, teacher models may have a negative impact on student models. In this work, we propose a method based on knowledge distillation that uses self-transformed hints and loss functions that ignore outliers to tackle real world and challenging crowd counting tasks. Through our approach we achieve a MAE of 77.24 and a MSE of 276.17 using the JHU-CROWD++ [1] test set. This is comparable to state-of-the-art deep crowd counting models, but at a fraction of the original model size and complexity, thus making the solution suitable for IoT devices.**

***Index Terms*—Crowd counting, Deep learning, Knowledge distillation.**

I. Introduction

Crowd counting involves an estimation of the total number of people in a crowd. This can be based on static images, pre-recorded video streams or ideally live video streams. This is useful in many scenarios, e.g., when evaluating and subsequently preventing over-crowding situations or just estimating the size of crowds more generally. In both cases, the inference speed and accuracy of crowd counting models can be critical. Many modern deep learning models used for crowd counting utilize CNN networks as the backbone, then produce density maps that contain a high volume of information. Such maps often have a degree of associated noise as shown in Figure 1.

To increase the inference speed of crowd counting models, many works have employed lightweight CNNs as the backbone, however such models either have low accuracy or they are too large for deployment to constrained IoT devices [2]–[6]. Another approach is compressing pre-trained larger-scale, crowd counting models using approaches such as pruning and quantization. However these can cause a loss in accuracy due to the limited number of parameters. Therefore some crowd counting works have explored knowledge distillation (KD) to reduce the size of models whilst attempting to maintain accuracy. This is often achieved by distilling knowledge/information from larger (teacher) models to smaller (student) models [7], [8].

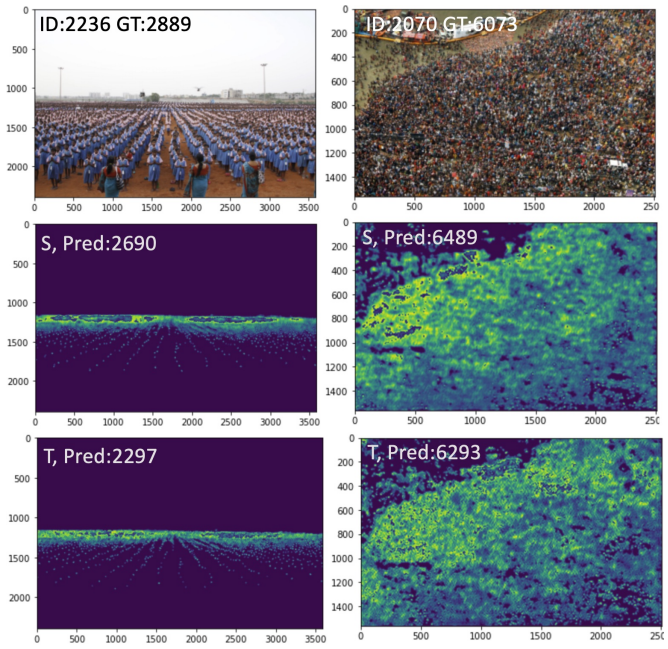

Fig. 1. Examples of crowd counting where ID is the image id in the JHU-CROWD++ data resource; GT is the ground-truth count; Pred is the predicted count; S is the student predicted density map and T the teacher predicted density map.

To distill as much as knowledge as possible from a teacher model, many crowd counting KD methods distill knowledge from multi-layered feature-maps (also known as *hints*) in the backbone of the teacher model, e.g., [7], [8]. Since the hints used in the teacher model are different from the student model, KD methods need to employ transformers to adjust the features map dimensions to address the requirements of the loss functions. However, this can give rise to a loss of information in the KD process [9]. To tackle this problem, KD methods such as [7], [10]–[13] employ convolutions (with 1x1 filters) to increase the number of channels used for student hints to match the teacher's hints. However, transformers employed in a given KD process are not part of a student model, hence the knowledge learned from the teacher model in the transformer is not always useful for inferences made by a student model. One approach to resolve this is to let the transformers form part of the student model. This can eliminate the parameter loss problem, however this can

increase the size of the model. Another problem with hint distillation is that many KD methods choose their hints without apparent reason or justification [9]. Knowledge in the selected hints could therefore have a negative and unexplained impact on a given student model.

In this work, we propose a KD method that uses self-transformed hints to avoid the need for additional transformers. The motivation is that if the teacher model has multi-layered feature maps (hints) that have the same dimension in each layer and sufficient useful knowledge for distillation, it is possible to design student models that have hints with the same dimensions as teacher model hints, thereby obviating the need for transformers. We call feature-maps that meet the above requirements as *self-transformed hints*.

KD also suffers from outlier problems when applied to regression tasks [8], [14]–[16]. An outlier is a data sample that can for example cause the teacher model to produce a larger loss than the associated student model. The negative impact of outliers in classification tasks is limited, since cross-entropy loss is bounded. Outliers, can however have a significant negative impact on regression tasks since regression loss is unlimited, especially when the output is for example a density map. To tackle this problem [14] ignore outliers in the bounding-box regression branch used for object detection tasks and increase the weight of the soft-loss based on the accuracy of the teacher model in the post-regression task [15]. [16] reduce the weight of the outlier soft-loss in tasks such as sinusoidal fitting, gaze direction prediction and head-pose prediction. For crowd counting, ShuffleCount [8] reduces the weight of the feature loss over time as the number of epochs increases. However, this method does not target outliers, but all data samples. In this work, we propose an improved outlier-tolerant loss for KD that removes the potential (negative) impact of outliers. The main contributions of this work are:

- we propose a method for distilling knowledge using self-transformed hints, which can prevent information loss both during the distillation process and when using the student model for inference;
- we propose an approach that can reduce the impact of outliers on loss, and
- we show how the proposed student model, after distillation, has a high degree of accuracy with far fewer parameters so that it is suitable for IoT devices used for real-time crowd counting inference with results comparable to much larger scale state-of-the-art models.

II. Related Works

A. Deep Learning-based Crowd Counting

Nowadays, most approaches to crowd counting utilize deep convolutional neural networks (CNNs) to generate density maps, which are then supervised by ground-truth pixel-wise density maps [18] and one or more loss functions. A pixel-wise density map can be defined as:

$$D(x_j) = \sum_{i=1}^{N} \mathcal{N}(x_j; y_i, \sigma^2) \tag{1}$$

where x_j is a pixel, y_i is the mean location of a given head i in an image, and $\mathcal{N}(x_j; y_i, \sigma^2)$ represents a 2D Gaussian kernel with mean y_i and variance σ. Then $\sum_{x \in X} D(x)$ is the estimated count of heads in image X which corresponds to the estimated size of the crowd.

Deep learning methods such as [1]–[3], [7], [17], [19]–[26] have achieved significant success in crowd counting tasks. Zhang et al. [19] proposed a convolutional neural network to resolve unseen aspects of crowd counting. A disadvantage of this model however was that it required perspective maps in order to calculate the size of pixels within an image. Work such as [2] use three parallel CNN branches, each with a different size of filter to produce features for images with varying resolutions. CSRNET [21] used VGG16 [27] as the backbone together with a dilated convolution to produce high-resolution density maps. Gao et al. [22] used ResNet-101 [28] for improved results. DSSINet [26] used three VGG16s (the first ten layers) that share parameters to make the backbone larger. The models discussed above use pixel-wise loss functions that measure the distances of every corresponding pixel pair between the ground-truth and the estimated density map. Such an approach is computationally expensive. Ma et al. [17] proposed a Bayesian Loss approach that used the weighted average (the expectation) of counting all pixels instead of pixel-wise counting to reduce the computational cost. During model training, pixels were weighted using Bayes rules based on their distance from the centre of a head.

While deep CNN networks can be used to improve the accuracy of crowd counting models, they typically do so at the cost of inference speeds. That is, such models normally require GPUs with large processing capacities to achieve timely inferences. To make crowd counting models suitable for devices with limited computational resources, works such as [5], [6], [29] adapt light-weight CNNs such as MobileNetV2 [30], ShuffleNetV2 [31] and EfficientNet-lite [32] as the network backbones. In MobileCount [5], a simplified MobileNetV2 model was utilized as the encoder/backbone, and multi-layer feature maps used to exploit fusion methods to increase the efficiency of the model. LigMSANet [29] also used MobileNetV2 to extract multi-layer feature maps, before decoding them to density maps using scale fusion and adaptation. LigMSANet achieved comparable accuracy to SANET and CSRNet but with less than 1 million parameters. EffCC-lite2 [6] used EfficientNet-lite2 as the backbone and adopted and enhanced the Bayesian loss of [17] to include bounding-box annotations instead of pixel-wise density maps for model training. It achieved an accuracy comparable to deep crowd counting models such as [1], [17], [26], but with far fewer parameters.

B. Knowledge Distillation

One way to increase the inference speed of deep crowd counting models is by compressing the models, e.g., by pruning or quantization, however this can cause a significant loss of accuracy especially if the compression rate is too high. To maintain accuracy, works such as [7], [8] adopt knowledge

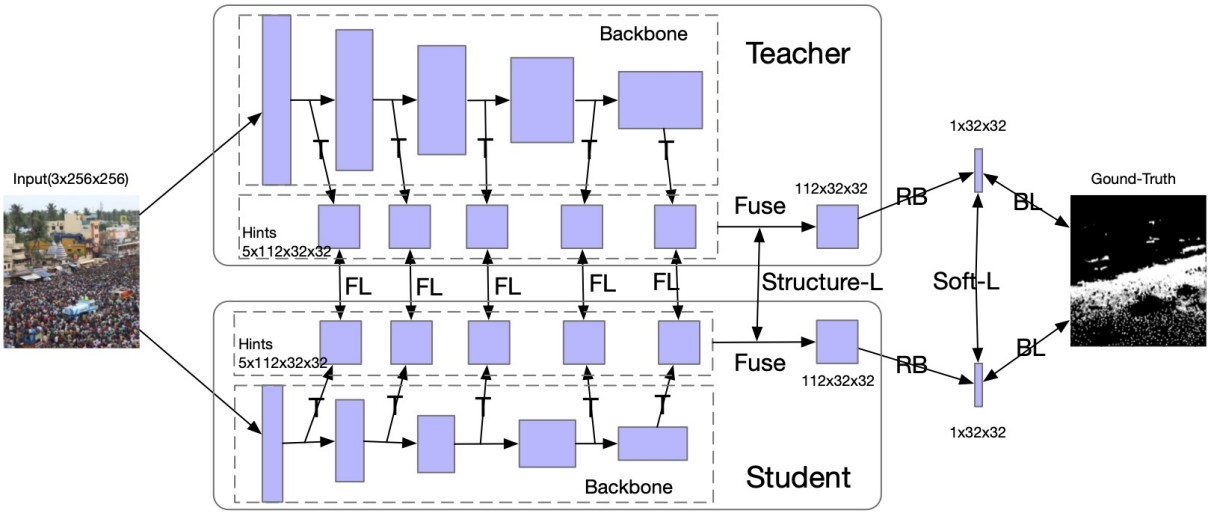

Fig. 2. Proposed Knowledge Distillation Method. Here the input image goes through the teacher and student models. By using self-contained transformers (T), both models produce dimension-adjusted feature maps (hints). The knowledge distillation process applies feature-loss (FL) and structure-loss (Structure-L) on these hints and soft-loss (Soft-L) on density maps to distill knowledge from the teacher model to the student model. For inference of the student model, the input image first goes through the backbone, then the transformers (T) adjust their dimensions, before they are fused into one feature map (summation). This then goes through Regression Blocks (RB) [6] to produce the predicted density map. Bayes-L provides the Bayesian-loss [6], [17] that is then applied for the predicted density map and the ground-truth density map.

distillation (KD) methods to distil knowledge from large crowd counting (teacher) models to train smaller (student) models in order to increase the accuracy of the student models.

Hinton et al. [33] propose KD for neural networks that extract knowledge from the logits of teacher models. These are transferred to a student (distilled) model by minimising the loss or distance between their logits. The distilled student model was able to achieve an accuracy very close to that of the teacher model in the classification task using the MNIST dataset. Romero et al. [11] introduced a KD method based on hints to distil further knowledge. Since the dimensions of hints can be different from layer to layer in the teacher and the student model, they typically need to be adjusted by transformers depending on the requirements of the loss functions [12], [34]. However, such transformations can cause information loss [9]. To reduce such loss, some works increase the dimension of student hints to match the teacher model [7], [12] .

For crowd counting tasks, SKT [7] use a structure distillation method to transfer the structure knowledge of hints from the teacher model to student models. To reduce information loss, the approach employs convolution layers with 1x1 filters as transformers to enlarge the size of hints used in the student model. The student model of SKT achieved a similar accuracy to the teacher model albeit with far fewer parameters. ShuffleCount [8] was the first crowd counting method to consider the negative effects of teacher models. They apply a soft decay factor to the features-loss to gradually reduce the weight of the feature-loss as the number of training epochs increased. ShuffleCount itself has 1.31 million parameters.

III. Proposed Method

A. Teacher Model Selection and Proposed Student Models

To minimize as much information loss as possible during the KD process and improve the accuracy of subsequent inference, the proposed method avoids the use of additional transformers to adjust the dimension of feature maps (hints). Rather both the teacher model and the student model must contain self-transformed hints of the same dimension in their corresponding blocks or layers, as shown in Fig. 2. Furthermore, the hints should contain sufficient useful information (parameters) to enable knowledge distillation. Since the dimensions of the hints in the student model are the same as those in the teacher model, a teacher model with a light-weight backbone is preferred since a heavy backbone, such as VGG16 or Resnet101, may result in over-sized hints for small student models. An over-sized hint not only makes the student model larger but also makes the training process more time consuming.

Based on the selection criteria above, we adopt the light-weight crowd counting model EffCC-lite2 [6] as the basis for our teacher model. EffCC-lite2 employs 5 layers of dimension-adjusted feature maps extracted from the backbone to produce density maps based on fusion and regression blocks [6]. We use these feature maps as self-transformed hints. A second reason is that the information contained in the hints could be sufficient to provide the knowledge needed by the KD process, as it is used to generate density maps. The backbone of the teacher model (EfficientNet Lite2 [32]) provides a light-weight CNN, resulting in hints that are small enough for small student models - of the order of 46.5K parameters. Finally, every layer in the hint has the same dimensions (112x32x32), which makes it possible to distill knowledge of the structure-

information (using Equation 5) from the teacher model to the student model, thereby increasing the accuracy of the student model [7].

Building on this, we design a student model with self-transformed hints with the same dimensions as the teacher model, but with a small backbone network (see Fig. 2). By reducing the number of channels and the depth of the backbone of the teacher model (EfficientNet Lite2), we reduce the number of parameters of the student model to less than one million. We then propose two student models: one with 0.875 million parameters and the other with 0.228 million parameters. The backbones of the student models are described in Table I. The details of EfficientNet can be found in [32]. In the student models, the self-contained transformers use convolutional operations with 1x1 filters to change the channel number of the input tensors to 112. This is then used to interpolate the feature maps to increase their length and width to 32. The fusion and regression blocks of the student model are the same as those of the teacher model.

TABLE I
NUMBER OF PARAMETERS AND CONFIGURATION OF TEACHER AND STUDENT MODELS. SEE [32] FOR DETAILS OF EFFICIENTNET.

Model	#Parameters	Channel multiplier	Depth multiplier
Teacher	4.53M	1.1	1.2
Student-0.5	0.875M	0.55	0.60
Student-0.25	0.228M	0.25	0.30

B. Loss functions

We use a teacher model to distill knowledge from the dimension-adjusted feature maps (hints) using the structure information of hints, and the predicted density map as shown in Fig. 2.

1) Angular Cosine Distance: For feature- and structure-loss, we employ an angular cosine distance to measure the distance between feature maps. The cosine similarity between two feature maps X and Y can be defined as:

$$\cos(X, Y) = \frac{X \cdot Y}{\parallel X \parallel \cdot \parallel Y \parallel} \tag{2}$$

where $\parallel X \parallel$ is a Frobenius Norm of tensor X and $\cdot$ is the element-wise multiplication. From this we define the angular cosine distance as:

$$acos(X, Y) = \arccos\left(\cos\left(X, Y\right)\right)\frac{180}{\pi} \tag{3}$$

where $\arccos$ is the inverse cosine function. The output of the angular cosine distance ranges from 0 to 180 degrees to represent the minimum and maximum distance respectively. The angular cosine distance requires both input tensors to have the same dimensions.

2) Feature-loss: There are five feature-loss points for a teacher or student model that can occur during the training process as shown in Fig. 2 based on one loss point for each layer of the self-transformed hints. Feature-loss is based on the angular cosine distance calculated using Equation 3. Specifically, we define feature-loss as:

$$\mathbf{L_{feat}}^{i}(T, S) = acos(H(T)^{i}, H(S)^{i}) \tag{4}$$

where i represents layer i of the hint, and $H(T)$ and $H(S)$ are the hints associated with the teacher (T) and student models (S) respectively.

3) Structure-loss: Structure-loss is the L2 norm between the structure matrix of the teacher and the student hints. For structure matrix $A(H)$, we define $a_{i,j} \in A(H)$ as an angular cosine distance between layer i and layer j of H. The structure-loss between the teacher model (T) and the student model (S) can then be calculated as:

$$\mathbf{L_{stru}}(T, S) = \parallel \mathbf{A}(H(T)), \mathbf{A}(H(S)) \parallel \tag{5}$$

Noting that structure-loss requires that the hint layers have the same dimensions.

4) Soft-loss: Soft-loss is the L2 norm between the predicted density map of the teacher and student models. A soft-loss can be defined as:

$$\mathbf{L_{soft}}(T, S) = \parallel \hat{D}(T), \hat{D}(S) \parallel \tag{6}$$

where $\hat{D}$ is the predicted density map.

5) KD-loss: Using the above equations, the loss function for the KD process can be defined as:

$$\mathbf{L_{KD}}(T, S) = \sum_{i=1}^{5} \mathbf{L_{feat}}^{i}(T, S) + \mathbf{L_{stru}}(T, S) + \mathbf{L_{soft}}(T, S) \tag{7}$$

6) Hard-loss: We employ the improved Bayesian loss [6] to calculate the hard-loss between the prediction density map and ground-truth count. The improved Bayesian loss is inspired by the Bayesian loss [17]. This is defined as:

$$\mathbf{L_{bayes}} = \sum_{i=1}^{N} |1 - E(c_i)| \tag{8}$$

where $E(c_i)$ is the weighted average when counting all pixels for a head annotation i, and 1 is the ground-truth count. Because the proposed models are trained based on cropped image patches, it is possible that part of a head is located at the edge of a given image patch, which leads to inaccurate ground-truths. Improved Bayesian loss [6] uses a portion of the bounding-box inside the image patch T_i instead of 1 to solve this problem. The improved Bayesian Loss can then be represented as follows:

$$\mathbf{L_{bayes\_improved}} = \sum_{i=1}^{N} |T_i - E(c_i)| \tag{9}$$

C. Outlier-Tolerant Loss

To reduce the potential negative impact of the teacher model caused by outliers, we propose outlier-tolerant loss that can mitigate the KD-loss caused by outliers in training data. We define W as the weight list for the KD-loss of all data samples in a training batch. We then define c_i^s,c_i^t,c_i^{gt} as the predicted

count from the student model, the predicted count of the teacher model, and the ground-truth count of input data sample i, respectively. The weight w_i in W can then be defined as:

$$w_i = \begin{cases} 1, & \text{if } |c_i^s - c_i^{gt}| \geq |c_i^t - c_i^{gt}| \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

when $w_i = 0$, we identify that data sample i is an outlier. The Outlier-Tolerant Loss can then be defined as:

$$\mathbf{L_{KD}^{w}}(T, S) = W \otimes \mathbf{L_{KD}}(T, S) \tag{11}$$

where $\otimes$ is data-sample-wise multiplication in a given training batch. Hence the training process can eliminate the negative impact of outliers in the teacher model.

D. Full Loss Function

Finally, the full loss function of the proposed distillation process can be given as:

$$\mathbf{L} = \mathbf{L_{KD}^{w}}(T, S) + \mathbf{L_{Bayes\_improved}} \tag{12}$$

IV. Experiments and Results

A. Dataset

We train the student models using the same data set (JHU-CROWD++ [1]) as the teacher model [6]. This data set contains 4,372 images with 1.51 million annotations. These are categorized into four groups: low-density, medium-density, high-density and weather. Taking advantage of this, we analyze the impact of our proposed methods on various crowd scenes and situations.

B. Evaluation Metrics

Two of the most common evaluation metrics for crowd counting are the mean absolute error (MAE) and the square root of the mean square error (RMSE). These are given as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|c_i^{gt} - \hat{c}_i| \tag{13}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|c_i^{gt} - \hat{c}_i|^2} \tag{14}$$

where c_i^{gt} and $\hat{c}_i$ denote the ground-truth and the predicted count of people in image i respectively.

C. Training, Validation and Testing

The models were trained using a single P100 Nvidia GPU with 6 cores comprising Intel(R) Xeon(R) CPU E5-2650 cpu and 8G memory. We employ the Adam optimiser and set the learning rate and weight_decay to 10^{-5} and 10^{-4} respectively.

We use the pre-trained EffCC-lite2 [6] as the teacher model, and the JHU-CROWD++ data set [1] to train and distill knowledge for the student models using the configuration shown in Table I. During training and validation, we crop the images and annotations in the training and validation data set to 256x256. We set the batch sizes to 16 and 1 for the training and validation stages respectively.

Test images are not cropped during the testing phase, and the batch size is set to 1. When the length or width of the input image is less than 512, it will be resized to 512 with fixed aspect ratio. If the image is greater than 256*14, it will be resized to 256*14 with the fixed aspect ratio. This was based on initial experiments, where we discovered that increasing the size of a small image (length or width smaller than 512 pixels) will improve the accuracy of the teacher and the associated student model, whilst a large image (length or width larger than 512*14 pixels) will consume too much GPU memory.

D. Results

We compare the MAE and RMSE of the student models to other state-of-the-art crowd counting models based on the JHU-CROWD++ [1] data set. The results of the teacher model are given in [6]. Other model results are explored in [1]. Table II shows that the RMSEs of our best student model (EffCC-lite0.5) are smaller than most state-of-the-art models in the overall category and it achieves the smallest MAE and RMSE in the high-density category. Our smallest student model (EffCC-lite0.25) has a better RMSE than the teacher model in the high-density category and in the weather category. However, both student models have lower accuracy than the teacher model based on other state-of-the-art models such as BCC [17], CG-DRCNs [1], LSCCNN [35] and SA-Net [3] in the low and medium-density category.

The improved accuracy of our student model in the high-density category can be attributed to the proposed outlier-tolerant loss approach. Table VI shows that if we remove the negative impact of outliers, the MAE and RMSE of the student model decreases significantly to less than the teacher model.

We also compared the computational cost including the number of parameters, GMac, GFlops, and inference speed of the proposed models with other state-of-the-art models given in [7], [8]. In Table III, we see that our best student model has 0.875M parameters, which is smaller than other distilled models such as ShuffleCount [8] and 1/4-BL+SKT [7], and it has a much lower GMac and GFlops dependency. For a 720p video, our best student model can infer at 16 FPS using a Jetson NX with no compression. Even the smallest student model reaches 28 FPS and its RMSE is smaller than most other state-of-the-art models, with the exception of the CG-DRCN-Res101 model although this has a much larger model size.

E. Ablation Study

This ablation study confirms that the improvement of our best student model is due to the proposed self-transformed knowledge distillation method combined with the outlier-tolerant approach.

1) Feature-loss and Structure-loss: We designed experiments to establish whether KD gives rise to improvements in student models. We first train a baseline student model without KD, and then train other student models with KD using only

TABLE II

COMPARISON WITH OTHER MODELS [1]. LOW, MEDIUM AND HIGH DENOTES THE SIZE OF CROWDS AND WEATHER INDICATES WHETHER THE IMAGE WAS TAKEN IN BAD WEATHER CONDITIONS, E.G., SNOW OR RAIN

	Category	Low(0-50)		Medium(51-500)		High(501+)		Weather		Overall	
	Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Others	MCNN [19]	97.10	192.30	121.40	191.30	618.60	1166.70	330.60	852.10	188.90	483.40
	CMTL [20]	58.50	136.40	81.70	144.70	635.30	1225.30	261.60	816.00	157.80	490.40
	CSR-Net [21]	27.10	64.90	43.90	71.20	356.20	784.40	141.40	640.10	85.90	309.20
	SA-Net [3]	17.30	37.90	46.80	69.10	397.90	817.70	154.20	685.70	91.10	320.40
	CACC [25]	37.60	78.80	56.40	86.20	384.20	789.00	155.40	617.00	100.10	314.00
	SFCN [23]	16.50	55.70	38.10	59.80	341.80	758.80	**122.80**	**606.30**	77.50	297.60
	DSSI-Net [26]	53.60	112.80	70.30	108.60	525.50	1047.40	229.10	760.30	133.50	416.50
	MBTTBF [24]	19.20	58.80	41.60	66.00	352.20	760.40	138.70	631.60	81.80	299.10
	LSCCNN [35]	**10.60**	**31.80**	**34.90**	55.60	601.90	1172.20	178.00	744.30	112.70	454.40
	CG-DRCN-VGG [1]	19.50	58.70	38.40	62.70	367.30	837.50	138.60	654.00	82.30	328.00
	CG-DRCN-Res [1]	14.00	42.80	35.00	**53.70**	314.70	**712.30**	**120.00**	**580.80**	**71.00**	**278.60**
	BCC [17]	**10.10**	32.70	**34.20**	**54.50**	352.00	768.70	140.10	675.70	75.00	299.90
Teacher	EffCC-Lite2	13.08	34.08	39.96	74.80	**309.06**	726.18	131.21	679.36	**72.67**	286.35
Students	EffCC-Lite0.5	16.14	43.53	46.74	82.15	**307.86**	**694.99**	133.18	623.00	77.24	**276.17**
	EffCC-Lite0.25	19.62	53.96	53.23	84.69	338.72	720.23	152.15	667.63	86.54	286.57

TABLE III

COMPARISON OF MODELS BASED ON NUMBER OF PARAMETERS, COMPUTATIONAL COMPLEXITY AND SPEED OF INFERENCE. GMAC IS BASED ON 1920X1080 IMAGES, GFLOPS BASED ON 2032X2912 IMAGES. HERE FPS IS THE INFERENCE SPEED OF THE PYTORCH MODEL BASED ON 1280X720-BASED VIDEO USING A JETSON NX.

	Models	#Parameters	GMac@1920x1080	GFLOPs@2032x2912	FPS@720p
Others-full	CSR-Net [21]	16.26M	857.8	2447.91	-
	DSSI-Net [26]	8.85M	-	8670.09	-
	BCC [17]	21.5M	-	2441.23	-
	CG-DRCN-VGG [1]	21.5M	-	-	-
Others-distilled	ShuffleCount(KD)	1.31M	37.17	-	-
	1/4-BL + SKT	1.35M	-	155.30	-
Teacher	EffCC-Lite2	4.53M	27.3	76.7	6
Student(Ours)	EffCC-Lite0.5	0.875M	8.23	23.04	16
	EffCC-Lite0.25	0.228M	4.69	13.11	28

TABLE IV

ABLATION STUDY OF KNOWLEDGE DISTILLATION. HERE BASELINE IS THE STUDENT MODEL TRAINED WITHOUT KNOWLEDGE DISTILLATION. F IMPLIES THE MODEL IS TRAINED USING FEATURE-LOSS AND S IMPLIES THE MODEL IS TRAINED USING STRUCTURE-LOSS.

Category	Low(0-50)		Medium(51-500)		High(501+)		Weather		Overall	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Baseline	24.86	61.15	56.11	97.02	389.67	856.93	173.07	710.17	97.21	340.14
Baseline(S)	22.15	71.33	51.71	94.40	369.61	795.55	157.68	686.62	90.74	317.68
Baseline(F)	16.71	44.45	46.10	78.41	313.80	732.01	141.04	678.91	77.91	289.48
EffCC-Lite0.5(S+F)	16.14	43.53	46.74	82.15	307.86	694.99	133.18	623.00	77.24	276.17

feature-loss or only structure-loss. Table IV shows that the baseline with structure-loss and the baseline with feature-loss both have improvements in accuracy compared to the baseline model without KD. The best model has lower MAE and RMSE compared to all baselines, with the exception of the medium density category of the baseline with feature-loss.

2) Self-transformed Hints: In order to test whether the proposed self-transformed hints can increase the accuracy of student models, we created a baseline student model (based on student-0.5 in Table I) that did not include self-transformed hints. In the baseline, we use external transformers to increase the dimensions of the feature maps extracted from the student model backbones in order to match the dimensions of the corresponding feature maps of the associated teacher models. In this experiment, the best model using self-transformed hints was based on our EffCC-lite0.5 student model. Since the hints in the baseline model have different numbers of channels in every layer, structure-loss makes them unsuitable for training, therefore we train the best model without structure-loss. Table V shows that the MAE and RMSE of the best model decreases significantly in the high density, weather and overall categories, with only a slight increase in low and medium density categories compared to baseline. This indicates that the proposed self-transformed hints contribute to the accuracy increase of the student model.

3) Outlier-Tolerant Loss: We also trained a student model without using outlier-tolerant loss as a baseline to confirm the negative impact of outliers. Table VI shows that the baseline

TABLE V
ABLATION STUDY OF INFORMATION LEAKAGE, WHERE F IMPLIES ALL MODELS ARE TRAINED WITH FEATURE-LOSS.

Category	Low(0-50)		Medium(51-500)		High(501+)		Weather		Overall	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Baseline(F)	14.59	33.50	46.13	73.71	358.93	775.43	149.83	699.12	84.06	304.78
EffCC-Lite0.5(F)	16.71	44.45	46.10	78.41	313.80	732.01	141.04	678.91	77.91	289.48

TABLE VI
ABLATION STUDY OF OUTLIER-TOLERANT LOSS. HERE BASELINE IS THE STUDENT MODEL TRAINED WITHOUT OUTLIER-TOLERANT LOSS (OTL)

Category	Low(0-50)		Medium(51-500)		High(501+)		Weather		Overall	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Teacher	13.08	34.08	39.96	74.80	309.06	726.18	131.21	679.36	72.67	286.35
Baseline w/o OTL	15.77	47.92	44.08	81.02	322.42	756.35	140.87	678.87	77.77	299.20
EffCC-Lite0.5(F+S)	16.14	43.53	46.74	82.15	**307.86**	**694.99**	133.18	**623.00**	77.24	**276.17**

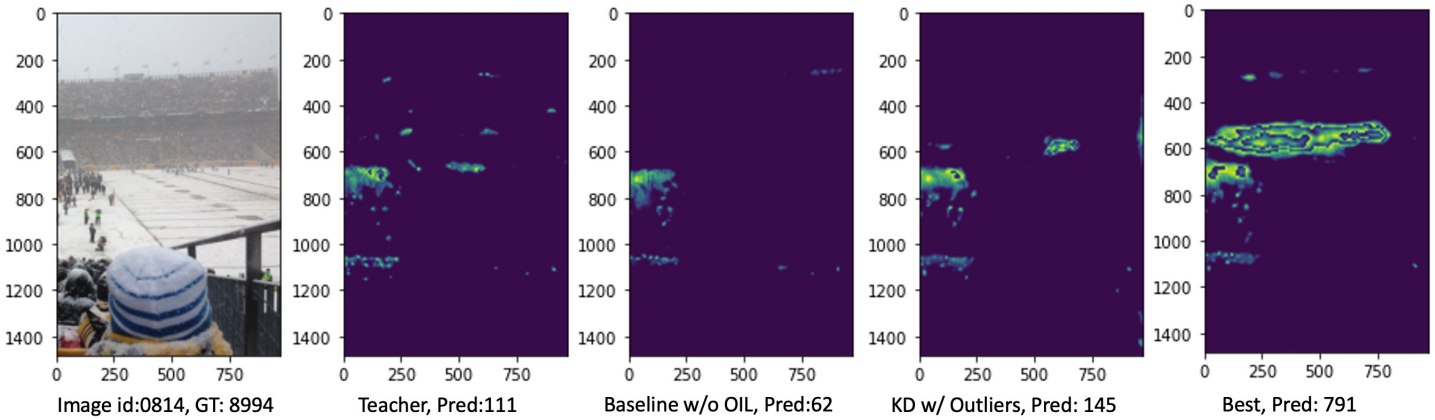

Fig. 3. An example of one of the worst cases. Here the input image id is 0814 in the test data set from JHU-CROWD++; GT is the ground-truth count; Pred is the predicted count, Baseline without KD is the student model (EffCC-lite0.5) trained without KD. KD without OTL means the student model trained without outlier-tolerant loss. The best model is the EffCC-lite0.5(S+F) model trained with outlier-tolerant loss.

model has higher MAE and RMSE in high-density, weather and overall categories compared to our best model trained using outlier-tolerant loss. However, the results do not differ significantly between these two models in the low and medium density categories.

To demonstrate the improvement caused by outlier-tolerant loss, we show an image that causes the largest errors in the test data set. As shown in Fig. 3, both the teacher model and the baseline model (student model trained without KD) produce poor results. When the baseline is trained without outlier-tolerant loss, the student model achieves a similar result to the teacher model. If we train the student model using outlier-tolerant loss, the accuracy improves significantly, and the student model is able to capture the crowd at the far end of the view (right hand image) which the teacher model was unable to capture.

V. CONCLUSIONS

In this work, we have shown how self-transformed hints and outlier-tolerant loss can significantly improve the accuracy of knowledge distillation for crowd counting tasks appropriate for IoT devices. Our best student model has a similar MAE and smaller RMSE compared to state-of-the-art models. However, it has less than 1 million parameters and 8.23 GMac for a 1920x1080 image allowing it to infer 2.6 times faster (16 FPS) than other light-weight teacher models (EffCC-lite2) using 720p video on a Jetson Nx. As our distilled model is only 3.5 Megabytes in size and hence it can be deployed directly onto lightweight IoT crowd-counting devices.

While the proposed knowledge distillation method requires the teacher model to have a sufficient number of feature maps that have the same dimension, we believe that such feature maps can help CNN models increase their accuracy and efficiency. We also believe that other regression tasks can benefit from the proposed knowledge distillation methods presented in this work.

Future work will focus on the practical deployment of these models onto diverse IoT devices in real world crowd-counting scenarios.

REFERENCES

[1] V. Sindagi, R. Yasarla, and V. Patel, "Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 11 2020.

[2] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 589–597, 2016.

[3] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 734–750, 2018.

[4] X. Wu, B. Xu, Y. Zheng, H. Ye, J. Yang, and L. He, "Video crowd counting via dynamic temporal modeling," *CoRR*, 2019.

[5] P. Wang, C. Gao, Y. Wang, H. Li, and Y. Gao, "Mobilecount: An efficient encoder-decoder framework for real-time crowd counting," *Neurocomputing*, vol. 407, pp. 292–299, 2020.

[6] Z. Huang, R. Sinnott, and Q. Ke, "Crowd counting using deep learning in edge devices," in *2021 IEEE/ACM 8th International Conference on Big Data Computing, Applications and Technologies (BDCAT'21)*, pp. 28–37, 2021.

[7] L. Liu, J. Chen, H. Wu, T. Chen, G. Li, and L. Lin, "Efficient crowd counting via structured knowledge transfer," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2645–2654, 2020.

[8] M. Jiang, J. Lin, and Z. J. Wang, "Shufflecount: Task-specific knowledge distillation for crowd counting," in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 999–1003, IEEE, 2021.

[9] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[10] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

[11] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[12] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1921–1930, 2019.

[13] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3779–3787, 2019.

[14] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," *Advances in neural information processing systems*, vol. 30, 2017.

[15] M. R. U. Saputra, P. P. De Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni, "Distilling knowledge from a deep pose regressor network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 263–272, 2019.

[16] M. Takamoto, Y. Morishita, and H. Imaoka, "An efficient method of training small models for regression problems with knowledge distillation," in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 67–72, IEEE, 2020.

[17] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6142–6151, 2019.

[18] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in neural information processing systems*, pp. 1324–1332, 2010.

[19] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 833–841, 2015.

[20] V. A. Sindagi and V. M. Patel, "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.

[21] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1091–1100, 2018.

[22] J. Gao, W. Lin, B. Zhao, D. Wang, C. Gao, and J. Wen, "Cˆ 3 framework: An open-source pytorch code for crowd counting," *arXiv preprint arXiv:1907.02724*, 2019.

[23] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8198–8207, 2019.

[24] V. A. Sindagi and V. M. Patel, "Multi-level bottom-top and top-bottom feature fusion for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1002–1012, 2019.

[25] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5099–5108, 2019.

[26] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin, "Crowd counting with deep structured scale integration network," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1774–1783, 2019.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[29] G. Jiang, R. Wu, Z. Huo, C. Zhao, and J. Luo, "Ligmsanet: Lightweight multi-scale adaptive convolutional neural network for dense crowd counting," *Expert Systems with Applications*, vol. 197, p. 116662, 2022.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[31] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 116–131, 2018.

[32] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.

[33] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[34] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," *Advances in neural information processing systems*, vol. 31, 2018.

[35] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, and R. V. Babu, "Locate, size and count: Accurately resolving people in dense crowds via detection," *arXiv preprint arXiv:1906.07538*, 2019.