Completely Self-Supervised Crowd Counting via Distribution Matching

Deepak Babu Sam*[1], Abhinav Agarwalla*[1], Jimmy Joseph[1], Vishwanath A. Sindagi[2], R. Venkatesh Babu[1], and Vishal M. Patel[2]

[1] Indian Institute of Science, Bangalore 560012, India
[2] Johns Hopkins University, Baltimore MD 21218, USA
{deepaksam,venky}@iisc.ac.in, {agarwallaabhinav,jimmyj005}@gmail.com, {vishwanathsindagi,vpatel36}@jhu.edu

Abstract. Dense crowd counting is a challenging task that demands millions of head annotations for training models. Though existing self-supervised approaches could learn good representations, they require some labeled data to map these features to the end task of density estimation. We mitigate this issue with the proposed paradigm of complete self-supervision, which does not need even a single labeled image. The only input required to train, apart from a large set of unlabeled crowd images, is the approximate upper limit of the crowd count for the given dataset. Our method dwells on the idea that natural crowds follow a power law distribution, which could be leveraged to yield error signals for backpropagation. A density regressor is first pretrained with self-supervision and then the distribution of predictions is matched to the prior. Experiments show that this results in effective learning of crowd features and delivers significant counting performance.

Keywords: Self-supervision, unsupervised learning, crowd counting

1 Introduction

The ability to estimate head counts of dense crowds effectively and efficiently serves several practical applications. This has motivated deeper research in the field and resulted in a plethora of crowd density regressors. These CNN based models deliver excellent counting performance almost entirely on the support of fully supervised training. Such a data hungry paradigm is limiting the further development of the field as it is practically infeasible to annotate thousands of people in dense crowds for every kind of setting under consideration. The fact that current datasets are relatively small and cover only limited scenarios, accentuates the necessity of a better training regime. Hence, developing methods to leverage the easily available unlabeled data has gained attention recently.

The classic way of performing unsupervised learning revolves around autoencoders [18, 26, 40, 57]. Autoencoders or its variants are optimized to predict back

* These authors contributed equally.

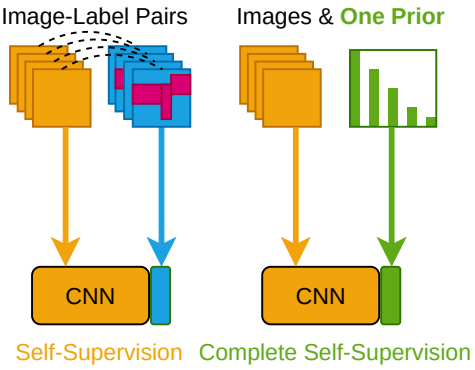

Fig. 1. Though self-supervision methods learn features in an unsupervised manner (in orange), they require labeled training to map these features to the end task (in blue). But complete self-supervision is devoid of any such instance-level supervision, instead relies on matching the statistics of the predictions to a prior distribution (in green).

their inputs, usually through a representational bottleneck. By doing so, the acquired features are generic enough that they could be employed for solving other tasks of interest. These methods have graduated to the more recent framework of self-supervision, where useful representations are learned by performing some alternate task for which pseudo labels can be easily obtained. For example, in self-supervision with colorization approach [28, 29, 70], a model is trained to predict the color image given its grayscale version. One can easily generate grayscale inputs from RGB images. Similarly, there are lots of tasks for which labels are freely available like predicting angle of rotation from an image [14, 15], solving jumbled scenes [44], inpainting [47] etc. Though self-supervision is effective in learning useful representations, they require a final mapping from the features to the end task of interest. This is thought to be essentially unavoidable as some supervisory signal is necessary to aid the final task. For this, typically a linear layer or a classifier is trained on top of the learned features using supervision from labeled data, defeating the true purpose of self-supervision. In the case of crowd counting, one requires training with annotated data for converting the features to a density map. To reiterate, the current unsupervised approaches could capture the majority of its features from unlabeled data, but demand supervision at the end for them to be made useful for any practical applications.

Our work emerges precisely from the above limitation of the standard self-supervision methods, but narrowed down to the case of crowd density estimation. The objective is to eliminate the mandatory final labeled supervision needed for mapping the learned self-supervised features to a density map output. In other words, we mandate developing a model that can be trained without using any labeled data. Such a problem statement is not only challenging, but also ill-posed. Without providing a supervisory signal, the model cannot recognize the task of interest and how to properly guide the training stands as the prime issue.

We solve this in a novel manner by carefully aiding the model to regress crowd density on the back of making some crucial assumptions. The idea relies on the observation that natural crowds tend to follow certain long tailed statistics and could be approximated to an appropriate parametric prior distribution. If a network trained with a self-supervised task is available, its features can be faithfully mapped to crowd density by enforcing the predictions to match the prior distribution. The matching is measured in terms of Sinkhorn distance [12], which is differentiated to derive error signals for supervision. This proposed framework is contrasted against the normal self-supervision regime in Figure 1, with the central difference being the replacement of the essential labeled training at the end by supervision through distribution matching. We show that the proposed approach results in effective learning of crowd features and delivers good performance in terms of counting metrics.

2 Related Work

Though there are earlier works like [9] on counting people in sparse crowds, the paradigm of dense crowd counting via density regression plausibly begins with [19]. The initial methods generally employ hand-crafted features and frequency analysis for counting. With the advent of deep learning, many CNN based density regressors have emerged. It ranges from the initial simple models [67] to multi-network/multi-scale architectures designed specifically to address the drastic diversity in crowd images [5, 7, 8, 45, 71]. Regressors with better, deeper and recurrent based deep models [24, 30, 32] are shown to improve counting performance. An alternate line of works enhance density regression by providing auxiliary information through crowd classification [51, 52], scene context [2, 10, 36], perspective data [49, 64], attention [35, 65, 66] and even semantic priors [59]. Models designed to progressively predict density maps and perform refinement is explored in [20, 48, 54]. Works like [34, 53] effectively fuse multi-scale information. Some approaches try to bring flavors of detection to crowd counting [3, 4, 31, 33, 38]. Interestingly, all these works, including more recent ones [39, 58, 60, 61], are fully supervised and leverage annotated data to achieve good performance. The issue of annotation has drawn attention of a few works in the field and is mitigated via multiple means. A count ranking loss on unlabeled images is employed in a multi-task formulation along with labeled data by [37]. Wang et al. [62] train using labeled synthetic data and adapt to real crowd scenario. The autoencoder method proposed in [6] optimizes almost 99% of the model parameters with unlabeled data. However, all of these models require some annotated data (either given by humans or obtained through synthetic means) for training.

Our approach is not only new to crowd counting, but also kindles alternate avenues in the area of unsupervised learning as well. Though initial works on the subject employ autoencoders or its variants [18, 26, 40, 57] for learning useful features, the paradigm of self-supervision with pseudo labels stands out to be superior in many aspects. Works like [28, 29, 70], learn representa-

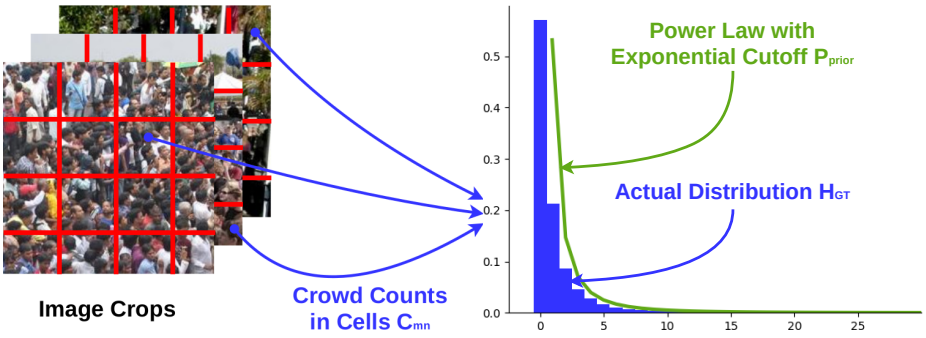

Fig. 2. Computing the distribution of natural crowds: crops from dense crowd images are framed to a spatial grid of cells and crowd counts of all the cells are aggregated to a histogram (obtained on Shanghaitech Part_A dataset [71]). The distribution is certainly long tailed and could be approximated to a power law.

tions through colorizing a grayscale image. Apart from these, pseudo labels for supervision are computed from motion cues [1, 22, 46], temporal information in videos [41, 63], learning to inpaint [47], co-occurrence [21], spatial context [13, 43, 44], cross-channel prediction [69], spotting artifacts [23], predicting object rotation [14, 15] etc. The recent work of Zhang et al. [68] introduce the idea of auto-encoding transformations rather than data. Furthermore, self-supervision is shown effective for unsupervised domain adaptation in [56]. An extensive and rigorous comparison of all major self-supervised methods is available in [27]. All these approaches focus on learning generic features and not the final task. But we extend self-supervision paradigm directly to the downstream task of interest.

3 Our Approach

3.1 Natural Crowds and Density Distribution

As mentioned in Section 1, our objective of training a density regressor without using any annotated data is somewhat ill-posed. The main reason being the absence of any supervisory signal to guide the model towards the task of interest, which is the density estimation of crowd images. But this issue could be circumvented by effectively exploiting certain structure or pattern specific to the problem. In the case of crowd images, restricting to only dense ones, we deduce an interesting pattern on the density distribution. They seem to spread out following a power law. To see this, we sample fixed size crops from lots of dense crowd images and divide each crop into a grid of cells as shown in Figure 2. Then the number people in every cell is computed and accumulated to a histogram. The distribution of these cell counts is quite clearly seen to be long tailed, with regions having low counts forming the head and high counts joining the tail. The number of cell regions with no people has the highest frequency, which then rapidly decays as the crowd density increases. This resembles the

way natural crowds are arranged with sparse regions occurring more often than rarely forming highly dense neighborhoods. Coincidentally, it has been shown that many natural phenomena obey a similar power law and is being studied heavily [11]. The dense crowds also appear conforming to this pattern as evident from multiple works [16, 17, 25, 42] etc. on the dynamics of pedestrian gatherings.

Moving to a more formal description, if $\boldsymbol{D}$ represents the density map for the input image $\boldsymbol{I}$, then the crowd count is given by $C = \sum_{xy} \boldsymbol{D}_{xy}$ (please refer [7, 19, 71] regarding creation of density maps). $\boldsymbol{D}$ is framed into a grid of $M \times N$ (typically set as $M = N = 3$) cells, with C_{mn} denoting the crowd count in the cell indexed by (m, n). Now let H^{GT} be the histogram computed by collecting the cell counts (C_{mn}s) from all the images. We try to find a parametric distribution that approximately follows H^{GT} with special focus to the long tailed region. The power law with exponential cut-off seems to be better suited (see Figure 2). Consequently, the crowd counts in cells C_{mn} could be thought as being generated by the following relation,

$$C_{mn} \sim P_{prior}(\mathrm{c}) \propto c^{\alpha} \exp(-\lambda c), \tag{1}$$

where P_{prior} is the substitute power law distribution. There are two parameters to P_{prior} with α controlling the shape and λ setting the tail length.

Our approach is to fix a prior distribution so that it can be enforced on the model predictions. Studies like [16, 42] simulate crowd behaviour dynamics and estimate the exponent of the power law to be around 2. Empirically, we also find that $\alpha = 2$ works in most cases of dense crowds, with the only remaining parameter to fix is the λ. Observe that λ affects the length of the tail and directly determines the maximum number of people in any given cell. If the maximum count C^{max} is specified for the given set of crowd images, then λ could be fixed such that the cumulative probability density (the value of CDF) of P_{prior} at C^{max} is very close to 1. We assume $1/S$ as the probability of finding a cell with count C^{max} out of S images in the given set. Now the CDF value at C^{max} could be set to $1-1/S$, simply the probability for getting values less than the maximum. Note that C^{max} need not be exact as small variations do not change P_{prior} significantly. This makes it practical as the accurate maximum count might not be available in real-world scenarios. Since C^{max} is for the cells, the maximum crowd count of the full image C^{fmax} is related as $C^{max} = C^{fmax}/(MNS_{crop})$, where S_{crop} denotes the average number of crops that make up a full image (and is typically set as 4). Thus, for a given a set of highly dense images, only one parameter, the C^{fmax} is required to fix an appropriate prior distribution.

We make a small modification to the prior distribution P_{prior} as its value range starts from 1. H^{GT} has values from zero with large probability mass concentrated near the low count region. Roughly 30% of the mass is seen to be distributed for counts less than or around 1. So, that much probability mass near the head region of P_{prior} is redistributed to $[0, 1]$ range in a uniform manner. This is found to be better for both training stability and performance.

In short, now we have a prior distribution representing how the crowd density is being allocated among the given set of images. Suppose there exists a CNN

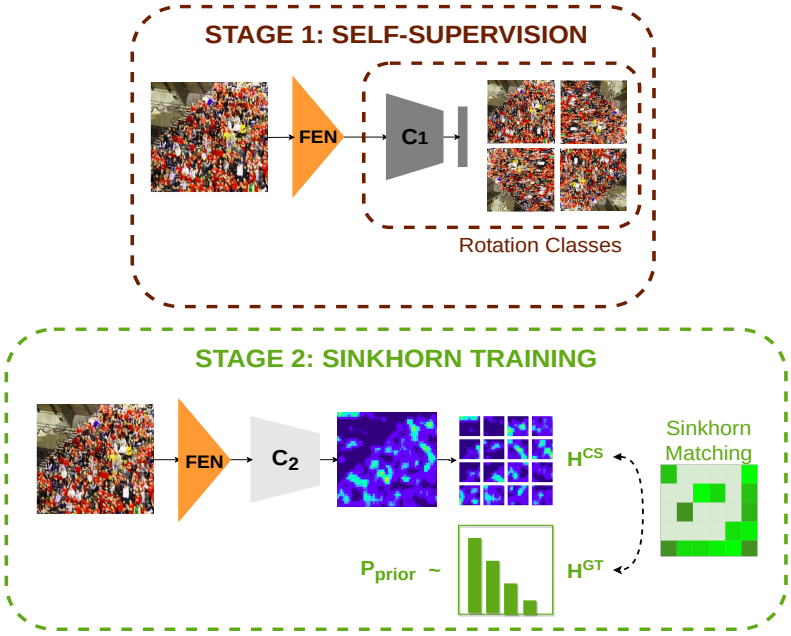

Fig. 3. The architecture of CSS-CCNN is shown. CSS-CCNN has two stages of training: the first trains the base *feature extraction network* in a self-supervised manner with rotation task and the second stage optimizes the model for matching the statistics of the density predictions to that of the prior distribution using optimal transport.

model that can output density maps, then one could try to generate error signals for updating the parameters of the model by matching the statistics of the predictions with that of the prior. But that could be a very weak signal for proper training of the model. It would be helpful if the model has a good initialization to start the supervision by distribution matching, which is precisely what we do by self-supervision in the next section.

3.2 Stage 1: Learning Features with Self-Supervision

We rely on training the model with self-supervision to learn effective and generic features that could be useful for the end task of density estimation. That means the model has to be trained in stages, with the first stage acquiring patterns frequently occurring in the input images. Since only dense crowd images are fed, we hope to learn mostly features relevant to crowds. These could be peculiar edges discriminating head-shoulder patterns formed by people to fairly high-level semantics pertaining to crowds. Note that the model is not signaled to pick up representations explicitly pertinent to density estimation, but implicitly culminate in learning crowd patterns as those are the most prominent part of the input data distribution. Hence, the features acquired by self-supervision could serve as a faithful initialization for the second stage of distribution matching.

Regarding self-supervision, there are numerous ways to generate pseudo labels for training models. The task of predicting image rotations is a simple, but highly effective for learning good representations [27]. The basic idea is to randomly rotate an image and train the model to predict the angle of rotation. By doing so, the network learns to detect characteristic edges or even fairly high-level patterns of the objects relevant for determining the orientation. These features are observed to be generic enough for diverse downstream tasks [27] and hence we choose self-supervision through rotation as our method.

Figure 3 shows the architecture of our density regressor, named the CSS-CCNN (for *Completely Self-Supervised Counting CNN*). It has a base *Feature Extraction Network* (FEN), which is composed of three VGG [50] style convolutional blocks with max poolings in-between. This is followed by two task heads: C_1 for the first training stage of self-supervision, and C_2 for regressing crowd density at second stage. The first stage branch has two more convolutions and a fully connected layer to finally classify the input image to one of the rotation classes. We take 112×112 crops from crowd images and randomly rotate the crop by one of the four predefined angles (0, 90, 180, 270 degrees). The model is trained with cross-entropy loss between the predicted and the actual rotation labels. The optimization runs till saturation as evaluated on a validation set.

Once the training is complete, the FEN has learned useful features for density estimation and the rotation classification head is removed. Now the parameters of FEN are frozen and is ready to be used in the second stage of training.

3.3 Stage 2: Sinkhorn Training

After the self-supervised training stage, FEN is extended to a density regressor by adding two convolutional layers as shown in Figure 3. We take features from both second and third convolution blocks for effectively mapping to crowd density. This aggregates features from slightly different receptive fields and is seen to deliver better performance. The layers of FEN are frozen and only a few parameters in the freshly added layers are open for training in the second stage of distribution matching. This particularly helps to prevent over-fitting as the training signal generated could be weak for updating large number of parameters. Now we describe the details of the exact matching process.

The core idea is to compute the distribution of crowd density predicted by CSS-CCNN and optimize the network to match that closely with the prior P_{prior}. For this, a suitable distance metric between the two distributions should be defined with differentiability as a key necessity. Note that the predicted distribution is in the form of an empirical measure (an array of cell count values) and hence it is difficult to formulate an easy analytical expression for the computing similarity. The classic Earth Mover's Distance (EMD) measures the amount of probability mass that needs to be moved if one tries to transform between the distributions (also described as the optimal transport cost). But this is not a differentiable operation and cannot be used directly in our case. Hence, we choose the Sinkhorn distance formulation proposed in [12]. Sinkhorn distance between two empirical measures is proven to be an upper bound for EMD and

has a differentiable implementation. Moreover, this method performs favorable in terms of efficiency and speed as well.

Let $\boldsymbol{D}^{CS}$ represent the density map output by CSS-CCNN and $\boldsymbol{C}^{CS}$ hold the cells extracted from the predictions. To make the distribution matching statistically significant, a batch of images are evaluated to get the cell counts ($\boldsymbol{C}^{CS}_{mn}$s), which are then formed into an array H^{CS}. We also sample the prior P_{prior} and create another empirical measure H^{GT} to act as the ground truth. Now the Sinkhorn loss $\mathcal{L}_{sink}$ is computed between H^{GT} and H^{CS}. It is basically a regularized version of optimal transport (OT) distance for the two sample sets. Designate $\boldsymbol{h}^{GT}$ and $\boldsymbol{h}^{CS}$ as the probability vectors (summing to 1) associated with the empirical measures H^{GT} and H^{CS} respectively. Now a transport plan $\boldsymbol{P}$ could be conceived as the joint likelihood of shifting the probability mass from $\boldsymbol{h}^{GT}$ to $\boldsymbol{h}^{CS}$. Define $\boldsymbol{U}$ to be the set of all such valid candidate plans as,

$$\boldsymbol{U} = \{\boldsymbol{P} \in \mathbb{R}_{+}^{d \times d} \mid \boldsymbol{P}\mathbf{1} = \boldsymbol{h}^{GT}, \boldsymbol{P}^{T}\mathbf{1} = \boldsymbol{h}^{CS}\}. \tag{2}$$

There is a cost $\boldsymbol{M}$ associated with any given transport plan, where M_{ij} is the squared difference between the counts of ith sample of H^{GT} and jth of H^{CS}. Closer the two distribution, lower would be the cost for transport. Hence, the Sinkhorn loss $\mathcal{L}_{sink}$ is defined as the cost pertinent to the optimal transportation plan with an additional regularization term. Mathematically,

$$\mathcal{L}_{sink}(H^{GT}, H^{CS}) = \underset{\boldsymbol{P} \in \boldsymbol{U}}{\arg\min} \langle \boldsymbol{P}, \boldsymbol{M} \rangle_{F} - \frac{1}{\beta} E(\boldsymbol{P}), \tag{3}$$

where $<>_F$ stands for the Frobenius inner product, $E(\boldsymbol{P})$ is the entropy of the joint distribution $\boldsymbol{P}$ and β is a regularization constant (see [12] for more details). It is evident that minimizing $\mathcal{L}_{sink}$ brings the two distributions closer in terms of how counts are allotted.

The network parameters are updated to optimize $\mathcal{L}_{sink}$, thereby bringing the distribution of predictions close to that of the prior. At every iteration of the training, a batch of crowd images are sampled from the dataset and empirical measures for the predictions as well as prior are constructed to backpropagate the Sinkhorn loss. The value of $\mathcal{L}_{sink}$ on a validation set of images is monitored for convergence and the training is stopped if the average loss does not improve over a certain number of epochs. Note that we do not use any annotated data even for validation. The counting performance is evaluated at the end with the model chosen based on the best mean validation Sinkhorn loss.

Thus, our Sinkhorn training procedure does not rely on instance-level supervision, but exploits matching the statistics computed from a set of inputs to that of the prior. One criticism regarding this method could be that the model need not learn the task of crowd density estimation by optimizing the Sinkhorn loss. It could learn any other arbitrary task that follows a similar distribution. The counter-argument stems from the semantics of the features learned by the base network. Since the initial training mostly captures features related to dense crowds, the Sinkhorn optimization has only limited flexibility in what it can do other than map them through a fairly simple function to crowd density. This

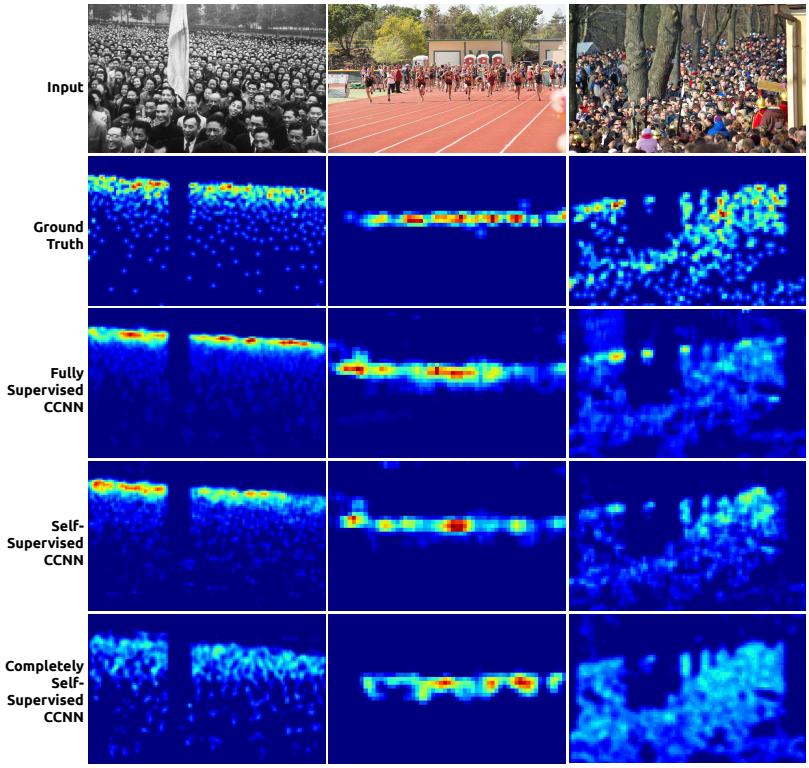

Fig. 4. Density maps estimated by CSS-CCNN along with that of baseline methods. Despite being trained without a single annotated image, CSS-CCNN is seen to be quite good at discriminating the crowd regions as well as regressing the density values.

is especially true as there is only a small set of parameters being trained with Sinkhorn. It is highly likely and straightforward to map the frequent crowd features to its density values, whose distribution is signaled through the prior. Moreover, we show through extensive experiments that CSS-CCNN ends up learning crowd density estimation.

4 Experiments and Analysis

Any crowd density regressor is evaluated mainly for the standard counting metrics. There are two metrics widely being followed by the community. The first is the MAE or Mean Absolute Error, which directly measures the counting performance. It is the absolute difference of the predicted and actual counts averaged over the test set or simply expressed as MAE $= (1/S_{test}) \sum_{i=1}^{N} |C_i - C_i^{GT}|$, where C_i is the count predicted by the model for ith image and C_i^{GT} denotes the actual count. Note that S_{test} is the number of images in the test set. Coming to the second metric, the Mean Squared Error or MSE is defined as

Table 1. Performance comparison of CSS-CCNN with other methods. Our model outperforms all the baselines.

Method	ST PartA		UCF-QNRF		UCF-CC-50		JHU-CRWD	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
WTA-CCNN [6]	154.7	229.4	-	-	433.7	583.3	-	-
CCNN Self-Super	121.2	197.5	196.8	309.3	348.8	484.3	147.5	436.2
CCNN Random	431.1	559.0	718.7	1036.3	1279.3	1567.9	320.3	793.5
CCNN Mean	282.8	359.9	567.1	752.8	771.2	898.4	316.3	732.3
CCNN P_{prior}	272.2	372.5	535.6	765.9	760.0	949.9	302.3	707.6
CSS-CCNN (*0 labels*)	197.3	295.9	437.0	722.3	564.9	959.4	217.6	651.3

MSE = SQRT$((1/S_{test})\sum_{i=1}^{N}(C_i - C_i^{GT})^2)$, a measure of the variance of count estimation and it represents the robustness of the model.

Our completely self-supervised framework is unique in many ways that the baseline comparisons should be different from the typical supervised methods. It is not fair to compare CSS-CCNN with other approaches as they use the full annotated data for training. Hence, we take a set of solid baselines for our model to demonstrate its performance. The *CCNN Random* experiment refers to the results one would get if only *Stage 1* self-supervision is done without the subsequent Sinkhorn training. This is the random accuracy for our setting and helpful in showing whether the proposed complete self-supervision works. Since our approach takes one parameter, the maximum count value of the dataset (C^{fmax}) as input, *CCNN Mean* baseline indicates the counting performance if the regressor blindly predicts the given value for all the images. We choose mean value as it makes for sense in this setting than the maximum (which anyway has worse performance than mean). Another important validation for our proposed paradigm is the *CCNN* P_{prior} experiment, where the model gives out a value randomly drawn from the prior distribution as its prediction for a given image. The counting performance of this baseline tells us with certainty whether the *Stage 2* training does anything more than that by chance. Note that we do not initialize CCNN with any pretrained weights as is typically done for supervised counting models. *CCNN Self-Super* runs the *Stage 1* training to learn the FEN parameters and is followed by labeled optimization for updating the regressor layers. These self-supervised or fully supervised methods are not directly comparable to our approach as we do not use any annotated data for training, but are shown for completeness. Also note that only the train/validation set images are used for optimizing CSS-CCNN and the ground truth annotations are never used. The counting metrics are computed on the labeled test set.

4.1 Crowd Datasets

The Shanghaitech Part_A [71] is a popular dense crowd counting dataset, containing 482 images randomly crawled from the Internet. It has images with crowd counts as low as 33 to as high as 3139, with an average of 501. The train set has

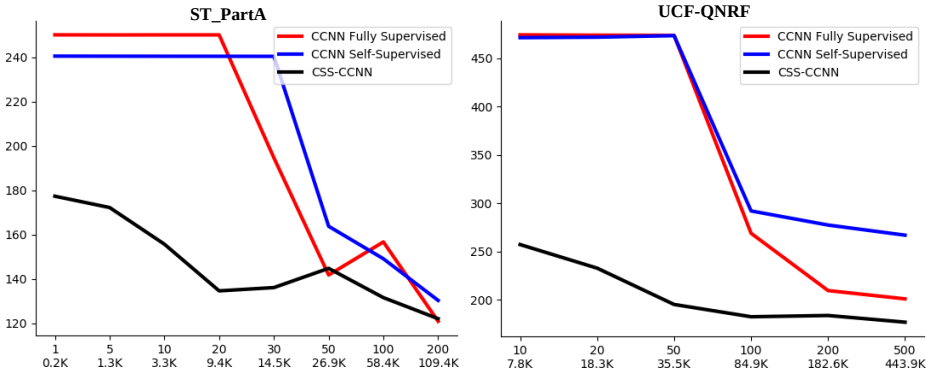

Fig. 5. Comparing our completely self-supervised method to fully supervised and self-supervised approaches under a limited amount of labeled training data. The x-axis denotes the number of training images along with the count (in thousands) of head annotations available for training, while the y-axis represents the MAE thus obtained. At low data scenarios, CSS-CCNN has significantly superior performance.

300 images, out of which 10% is held out for validation. There are 182 images testing. The hyper-parameter used for this is $C^{fmax} = 3000$. We compare the performance of CSS-CCNN with the baselines listed earlier and other competing methods in Table 1. It is clear that CSS-CCNN outperforms all the baselines by a significant margin. This shows that the proposed method works better than any naive strategies that do not consider the input images. Figure 4 visually compares density predictions made by CSS-CCNN and other models. The predictions of our approach are mostly on crowd regions and closely follows the ground truth, emphasizing its ability to discriminate crowds well.

UCF-QNRF dataset [20] is a large and diverse collection of crowd images with 1.2 million annotations. There are 1535 images with crowd count varying from 49 to 12865, resulting in an average of 815 individuals per image. The dataset offers very high-resolution images with an average resolution of 2013 $\times$ 2902. The max count hyper-parameter is set to $C^{fmax} = 12000$. We achieve similar performance trends on UCF-QNRF dataset as well. CSS-CCNN outperforms all the unsupervised baselines in terms of MAE and MSE as evident from Table 1.

UCF_CC_50 dataset [19] has just 50 images with extreme variation in crowd density ranging from 94 to 4543. The small size and diversity together makes this dataset the most challenging. We follow the standard 5-fold cross-validation scheme suggested by the creators of the dataset to report the performance metrics. Since the number of images is quite small, the assumption taken for setting the prior distribution gets invalid to certain extent. But a slightly different parameter to the prior distribution works. We set $\alpha = 1$ and $C^{fmax} = 4000$. Despite being a small and highly diverse dataset, CSS-CCNN is able to beat all the baselines. The self-supervised MAE is also better than [6]. These results evidence the effectiveness of our method.

Table 2. Evaluating CSS-CCNN in a true practical setting: the model is trained on images crawled from the web, but evaluated on crowd datasets.

Train on web images	MAE	MSE
Test on ST_PartA	208.8	309.5
Test on UCF-QNRF	450.7	755.9
Test on JHU-CROWD++	241.2	706.8

JHU-CROWD++ [54, 55] is a comprehensive dataset with 1.51 million head annotations spanning 4372 images. The crowd scenes are obtained under various scenarios and weather conditions, making it one of the challenging dataset in terms of diversity. Furthermore, JHU-CROWD++ has a richer set of annotations at head level as well as image level. The maximum count is fixed to $C^{fmax} = 8000$. The performance trends are quite similar to other datasets, with our approach delivering better MAE than the baselines as evident from Table 1. This indicates the generalization ability of CSS-CCNN across different datasets.

4.2 Performance with Limited Data

Here we explore the proposed algorithm along with fully supervised and self-supervised approaches when few annotated images are available for training. The analysis is performed by varying the number of labeled samples and the resultant counting metrics are presented in Figure 5. For training CSS-CCNN with data, we utilise the available annotated data to compute the optimal Sinkhorn assignments $\boldsymbol{P}^*$ and then optimize the $\mathcal{L}_{sink}$ loss. This way both the labeled as well as unlabeled data can be leveraged for training by alternating respective batches (in a 5:1 ratio). It is clear that, at very low data, scenarios CSS-CCNN beats the supervised as well as self-supervised baselines by a significant margin. The Sinkhorn training shows 13% boost in MAE (for Shanghaitech Part_A) by using just one labeled sample as opposed to no samples. This indicates that CSS-CCNN can perform well in extremely low data regimes. It takes about 20K head annotations for the supervised model to perform as well as CSS-CCNN. Also, CSS-CCNN has significantly less number of parameters to learn using the labeled samples as compared to a fully supervised network. These results suggests that our complete self-supervision is the right paradigm to employ for crowd counting when the amount of available annotated data is less.

4.3 CSS-CCNN in True Practical Setting

The complete self-supervised setting is motivated for scenarios where no labeled images are available for training. But till now we have been using images from crowd datasets with the annotations being intentionally ignored. Now consider crawling lots of crowd images from the Internet and employing these unlabeled data for training CSS-CCNN. For this, we use textual tags related to dense

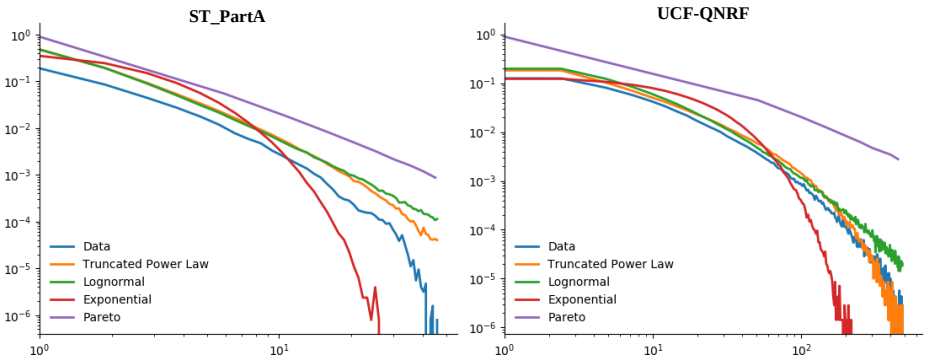

Fig. 6. Double logarithmic representation of maximum likelihood fit for the crowd counts from different datasets.

crowds and similarity matching with dataset images to collect approximately 5000 dense crowd images. No manual pruning of undesirable images with motion blur, perspective distortion or other artifacts is done. CSS-CCNN is trained on these images with the same hyper-parameters as that of Shanghaitech Part_A and the performance metrics are computed on the datasets with annotations. From Table 2, it is evident that our model achieves very competitive MAE on the crowd datasets (compared to Table 1), despite not using images from those datasets for training. This further demonstrates the generalization ability of CSS-CCNN to learn from less curated data, emphasizing the practical utility.

4.4 Analysis of the Prior Distribution

The proposed Sinkhorn training requires a prior distribution of crowd counts to be defined and the choice of an appropriate prior is essential for the best model performance as seen from Table 3. Here we analyze the crowd data more carefully to see why the truncated power law is the right choice of prior. For this, the counts from crowd images are extracted as described in Section 3.1 and a maximum likelihood fit over various parametric distributions is performed. The double logarithmic visualization of the probability distribution of both the data and the priors are available in Figure 6. Note that the data curve is almost a straight line in the logarithmic plot, a clear marker for power law characteristic. Both truncated power law and lognormal tightly follow the distribution. But on close inspection of the tail regions, we find truncated power law to best represent the prior. This further validates our choice of the prior distribution.

4.5 Sensitivity Analysis for the Crowd Parameter

As described in Section 3.1, CSS-CCNN requires the maximum crowd count (C^{fmax}) for the given set of images as an input. This is necessary to fix the prior distribution parameter λ. One might not have the exact max value for

Table 3. Ablating the effect of hyper-parameters on CSS-CCNN. Our model is robust to fairly large change in the maximum count parameter.

Param	MAE	MSE
Uniform Prior	261.8	406.0
Pareto Prior	248.3	386.2
Lognormal Prior	239.5	345.8
Truncated Power Law Prior	197.3	295.9
$C^{fmax} = 2000$	204.2	316.4
$C^{fmax} = 2500$	197.9	304.6
$C^{fmax} = 3000$	197.3	295.9
$C^{fmax} = 3500$	191.9	288.5
$\alpha = 1.9$	202.9	303.3
$\alpha = 2.0$	197.3	295.9
$\alpha = 2.1$	200.7	305.6

the crowds in a true practical setting; an approximate estimate is a more reasonable assumption. Hence, we vary C^{fmax} around the actual value and train CSS-CCNN on Shanghaitech PartA [71] and UCF-QNRF[20]. The performance metrics in Table 3 show that changing C^{fmax} to certain extent does not alter the performance significantly. The MAE remained roughly within the same range, even though the max parameter is being changed in the order of 500. These findings indicate that the our approach is insensitive to the exact crowd hyper-parameter value, increasing its practical utility. We also check the sensitivity of our approach on the power law exponent α. Varying α around 2 results in similar performances, in agreement with our design choices (see Section 3.1).

5 Conclusions

We show for the first time that a density regressor can be fully trained from scratch without using a single annotated image. This new paradigm of complete self-supervision relies on optimizing the model by matching the statistics of the distribution of predictions to that of a predefined prior. Though the counting performance of the model stands better than other baselines, there is a performance gap compared to fully supervised methods. Addressing this issue could be the prime focus of future works. For now, our work can be considered as a proof of concept that models could be trained directly for solving the downstream task of interest, without providing any instance-level labeled data.

6 Acknowledgments

This work was supported by MeitY (Ministry of Electronics and Information Technology) project (No. 4(16)2019-ITEA), Govt. of India. VMP was supported by an ARO grant W911NF-21-1-0135.

Bibliography

[1] Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)

[2] Babu Sam, D., Babu, R.V.: Top-down feedback for crowd counting convolutional neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)

[3] Babu Sam, D., Peri, S.V., Sundararaman, M.N., Babu, R.V.: Going beyond the regression paradigm with accurate dot prediction for dense crowds. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2020)

[4] Babu Sam, D., Peri, S.V., Sundararaman, M.N., Kamath, A., Babu, R.V.: Locate, size and count: Accurately resolving people in dense crowds via detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)

[5] Babu Sam, D., Sajjan, N.N., Babu, R.V., Srinivasan, M.: Divide and grow: Capturing huge diversity in crowd images with incrementally growing CNN. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

[6] Babu Sam, D., Sajjan, N.N., Maurya, H., Babu, R.V.: Almost unsupervised learning for dense crowd counting. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)

[7] Babu Sam, D., Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

[8] Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)

[9] Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)

[10] Cheng, Z.Q., Li, J.X., Dai, Q., Wu, X., Hauptmann, A.G.: Learning spatial awareness to improve crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)

[11] Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. SIAM review (2009)

[12] Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems (NIPS) (2013)

[13] Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)

[14] Feng, Z., Xu, C., Tao, D.: Self-supervised representation learning by rotation feature decoupling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[15] Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
[16] Helbing, D., Johansson, A., Al-Abideen, H.Z.: Dynamics of crowd disasters: An empirical study. Physical review E (2007)
[17] Helbing, D., Kühnert, C., Lämmer, S., Johansson, A., Gehlsen, B., Ammoser, H., West, G.B.: Power laws in urban supply networks, social systems, and dense pedestrian crowds. In: Complexity Perspectives in Innovation and Social Change, pp. 433–450. Springer (2009)
[18] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science (2006)
[19] Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
[20] Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
[21] Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Learning visual groups from co-occurrences in space and time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
[22] Jayaraman, D., Grauman, K.: Learning image representations tied to egomotion. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
[23] Jenni, S., Favaro, P.: Self-supervised feature learning by learning to spot artifacts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
[24] Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., Shao, L.: Crowd counting and density estimation by trellis encoder-decoder networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[25] Karamouzas, I., Skinner, B., Guy, S.J.: A universal power law governing pedestrian interactions. APS (2015)
[26] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the International Conference on Learning Representations (ICLR) (2013)
[27] Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[28] Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)

[29] Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
[30] Li, Y., Zhang, X., Chen, D.: CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
[31] Lian, D., Li, J., Zheng, J., Luo, W., Gao, S.: Density map regression guided detection network for rgb-d crowd counting and localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[32] Liu, C., Weng, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[33] Liu, J., Gao, C., Meng, D., Hauptmann, A.G.: DecideNet: Counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
[34] Liu, L., Qiu, Z., Li, G., Liu, S., Ouyang, W., Lin, L.: Crowd counting with deep structured scale integration network. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
[35] Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H.: ADCrowdNet: An attention-injective deformable convolutional network for crowd understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[36] Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[37] Liu, X., Van De Weijer, J., Bagdanov, A.D.: Exploiting unlabeled data in CNNs by self-supervised learning to rank. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
[38] Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[39] Ma, Z., Wei, X., Hong, X., Gong, Y.: Learning scales from points: A scale-aware probabilistic model for crowd counting. In: Proceedings of the 28th ACM International Conference on Multimedia (2020)
[40] Makhzani, A., Frey, B.J.: Winner-take-all autoencoders. In: Advances in Neural Information Processing Systems (NIPS) (2015)
[41] Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
[42] Moussaïd, M., Helbing, D., Theraulaz, G.: How simple rules determine pedestrian behavior and crowd disasters. Proceedings of the National Academy of Sciences (2011)
[43] Nathan Mundhenk, T., Ho, D., Chen, B.Y.: Improvements to context based self-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

[44] Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
[45] Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
[46] Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
[47] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
[48] Ranjan, V., Le, H., Hoai, M.: Iterative crowd counting. In: Proceedings of the European Conference on Computer Vision (2018)
[49] Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting perspective information for efficient crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[50] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
[51] Sindagi, V.A., Patel, V.M.: CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In: Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (2017)
[52] Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid CNNs. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
[53] Sindagi, V.A., Patel, V.M.: Multi-level bottom-top and top-bottom feature fusion for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
[54] Sindagi, V.A., Yasarla, R., Patel, V.M.: Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
[55] Sindagi, V.A., Yasarla, R., Patel, V.M.: JHU-CROWD++: Large-scale crowd counting dataset and a benchmark method. Technical Report (2020)
[56] Sun, Y., Tzeng, E., Darrell, T., Efros, A.A.: Unsupervised domain adaptation through self-supervision. arXiv preprint arXiv:1909.11825 (2019)
[57] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the International Conference on Machine Learning (ICML) (2008)
[58] Wan, J., Chan, A.: Modeling noisy annotations for crowd counting. Advances in Neural Information Processing Systems (2020)
[59] Wan, J., Luo, W., Wu, B., Chan, A.B., Liu, W.: Residual regression with semantic prior for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

[60] Wan, J., Wang, Q., Chan, A.B.: Kernel-based density map generation for dense object counting. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
[61] Wang, Q., Gao, J., Lin, W., Li, X.: Nwpu-crowd: A large-scale benchmark for crowd counting and localization. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
[62] Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[63] Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
[64] Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., Ding, E.: Perspective-guided convolution networks for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
[65] Zhang, A., Shen, J., Xiao, Z., Zhu, F., Zhen, X., Cao, X., Shao, L.: Relational attention network for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
[66] Zhang, A., Yue, L., Shen, J., Zhu, F., Zhen, X., Cao, X., Shao, L.: Attentional neural fields for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
[67] Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
[68] Zhang, L., Qi, G.J., Wang, L., Luo, J.: AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
[69] Zhang, R., Isola, P., Efros, A.: Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
[70] Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016)
[71] Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)