Evaluating Supervision Levels Trade-Offs for Infrared-Based People Counting

David Latortue*
david.latortue.1@ens.etsmtl.ca

Moetez Kdayem*
moetez.kdayem.1@ens.etsmtl.ca

Fidel A. Guerrero Peña
fidel-alejandro.guerrero-pena@etsmtl.ca

Eric Granger
eric.granger@etsmtl.ca

Marco Pedersoli
marco.pedersoli@etsmtl.ca

LIVIA, Dept. of Systems Engineering
ETS Montreal, Canada

Abstract

Object detection models are commonly used for people counting (and localization) in many applications but require a dataset with costly bounding box annotations for training. Given the importance of privacy in people counting, these models rely more and more on infrared images, making the task even harder. In this paper, we explore how weaker levels of supervision affect the performance of deep person counting architectures for image classification and point-level localization. Our experiments indicate that counting people using a convolutional neural network with image-level annotation achieves a level of accuracy that is competitive with YOLO detectors and point-level localization models yet provides a higher frame rate and a similar amount of model parameters. Our code is available at: https://github.com/tortueTortue/IRPeopleCounting.

1. Introduction

Intelligent building systems aim to enhance energy efficiency, environmental sustainability, security, and safety while improving user comfort [12]. Various subsystems within the building monitor and control lighting, heating, ventilation and air conditioning (HVAC), and other energy-consuming systems and manage space utilization to promote occupant well-being. These systems may leverage people counting technology to monitor occupancy, optimize energy consumption, and contribute to security and safety. However, privacy is a significant concern when implementing such technology [6]. Standard privacy-preserving measures, including data minimization and infrared cameras, help maintain individual anonymity and ensure privacy in low-light conditions. Therefore, this paper focuses on people counting using images captured with infrared cameras. Additionally, this paper focuses on methods for counting people in sparsely occupied scenes (not dense crowds) with limited overlap among people – a setting we call sparse crowd counting.

Commonly, the preferred approaches for counting people in sparse settings involve using object detectors [8] [43]. However, these approaches require annotating bounding boxes around individuals, which can be time-consuming and expensive. Fortunately, there are more cost-effective methods that can achieve the same goal with a lower cost for annotations. In particular, point-wise based supervision is often used in crow counting techniques [19,37], and relies solely on (x, y) pixel coordinates to localize people. This typically reduces the number of clicks needed for annotation by half, and eliminates the need for adjustments when bounding boxes do not properly fit a person. At the lowest level of supervision, image-level [38] counting relies on a single integer value annotation per image to indicate the number of people in each image. In this study, the level of supervision refers to the information available during the annotation rather than the learning process.

In this paper, we investigate the impact of the three aforementioned levels of supervision on the accuracy of deep learning (DL) models for sparse people counting based on infrared images. Our study compares the accuracy and complexity of YoloV8 and DINO object detection models, P2PNet and PET point-level localization models, as well as ConvNeXt and ViT image-level counting models. Fig. 1 presents a dichotomy of all these models. Additionally, to

[*] Equal contribution.

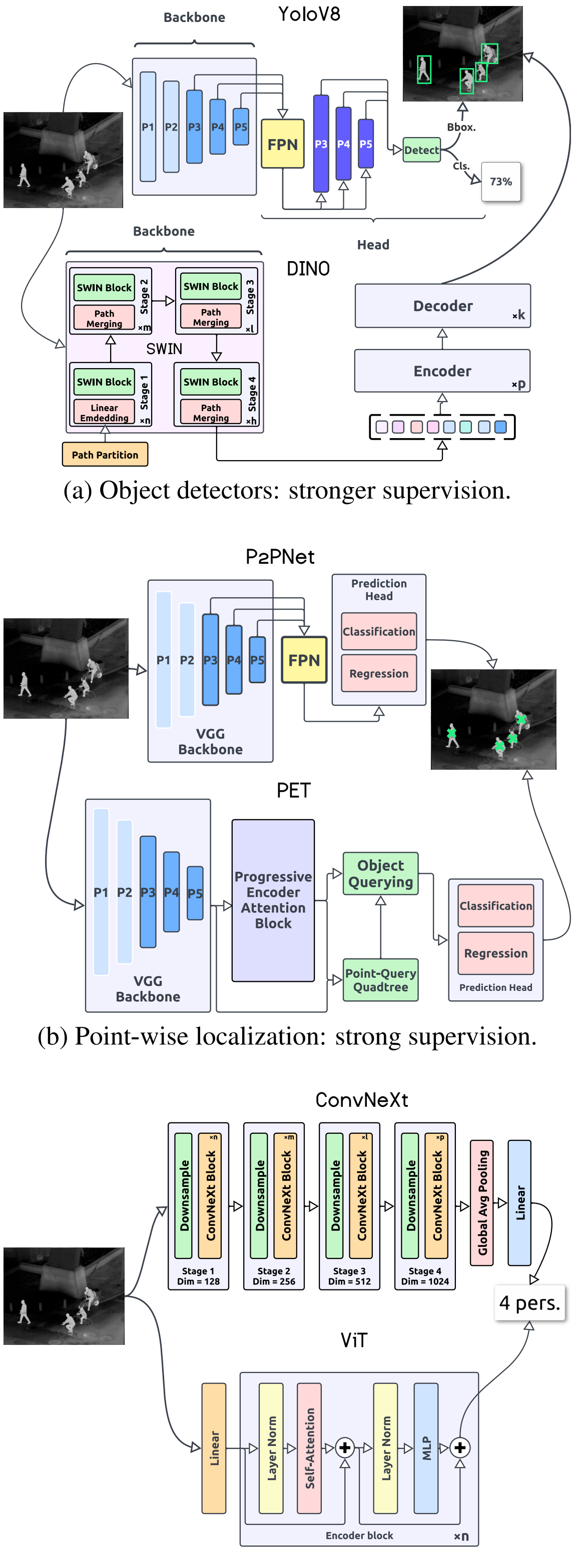
further improve the performance of the image-level models, we also employ the masked autoencoder (MAE) pretraining method to fully exploit the potential of using unlabeled data. We finally explore the effect of the dataset size and localization results, utilizing the most popular and cost-effective models in each category.

Our main contributions are summarized as follows. (1) We provide an extensive empirical comparison of various people counting techniques across multiple levels of supervision on two infrared image datasets (LLVIP and Distech IR). Our results show that image-level counting architectures deliver comparable performance to detectors, significantly reducing the annotation effort required. (2) The MAE pretraining technique is utilized in the people counting task, resulting in improved performance, especially when dealing with large amounts of unannotated data.

2. Related Work

People counting is a computer vision task that aims to estimate the number of individuals in an image. To address this challenge, various approaches can be employed, ranging from density estimation [4] to object detection [33]. In the literature, these methodologies are categorized as part of the broader group of methods related to crowd counting [8]. For very dense crowds, the techniques proposed in the literature tend to prioritize localizing the head of each individual [21, 28, 40] or density map estimation [9, 29, 44]. In the context of intelligent buildings, sparse crowds are more prevalent. Therefore, our work focuses on people counting for scenarios with 0 to 20 people. For this, DL models for object detection, object localization, and image-level estimations [26] are employed.

Object detectors predict the location and class label for each objection, aiming to identify the position and nature of objects in an image [48]. Various DL models have been developed for this task, significantly improving performance in recent years. In the literature, these methods are often categorized as either single-shot detection or two-shot detection. Single-shot (or single-stage) detection is an approach where the model directly outputs the position of an object, its bounding boxes, and the probability of its category. On the other hand, a two-stage model extracts an ensemble of regions of interest (ROIs) and then classifies each. The most well-known model families for single-shot and two-shot detectors are YOLO [30–32] and R-CNN [10, 11, 34], respectively. A later addition to the single-shot detector category is the DETR [3], a transformer-based detector that eliminates the need for hand-crafted components. DETR Improved Denoising Anchor Boxes (DINO) [49], combines DAB-DETR [20] and Denoising-DETR [18], for improved efficiency. It refines anchor boxes and classifications through a mixed query selection strategy and contrastive denoising training. In this work, for people count-



(a) Object detectors: stronger supervision.

(b) Point-wise localization: strong supervision.

(c) Image-level counting: normal supervision.

Figure 1. A dichotomy of the DL architectures analyzed in this paper sorted according to the level of supervision, from least expensive (bottom) to most expensive (top) annotation process.

ing, we use YOLOs, which have proven to be the best compromise between efficiency and performance, and DINO to investigate whether transformers can achieve the performance of earlier proposed architectures in this domain.

Point-wise localization is a task that involves locating an instance of an object within an image. Traditionally, object localization refers to identifying the coordinates of an object as well as defining bounding boxes around the object [35]. However, there are models exclusively designed to focus solely on the object's position [19, 37], which we refer to as point-level localizes. In the literature, numerous approaches are employed for crowd counting using coordinates like SGANet [41], GauNet [5], LoViTCrowd [39]. However, the two models currently leading in state-of-the-art performance on the popular benchmark ShanghaiTech A [50] are PET [19] and P2PNet [37]. Therefore, we have chosen to use these for our work.

Image-level counting typically involves categorizing the content of single-object images [42]. In this work, we have utilized the two common DL models for image classification/regression that often provide state-of-the-art performance on benchmark datasets: the CNN and Vision Transformer (ViT) [6]. Furthermore, image-level supervision for people counting is straightforward, typically requiring only the count of people in each image. However, we can distinguish two image-level approaches for people counting, through classification and regression. For both ViTs and CNNs, there is a wide range of high-performing models to choose from such as SWIN [22], VOLO [47], RevCol [2], NFNet [1]. In this paper, we have selected ConvNeXt and ViT for their efficiency and simplicity.

3. Methodology

Our study compares different levels of supervision used to train a DL model for the people counting task. In particular, we define three levels of supervision based on the strength of annotations used for people counting:
1) Normal supervision: In this level, the full annotation consists of the people count.
2) Strong supervision: This level involves using point-level localization, i.e., pixel coordinates of the person center, as annotations.
3) Stronger supervision: At this level, bounding boxes enclosing a person are used as annotations.

This section details the architectures and training approaches employed for each supervision strategy for people counting. Fig. 1 depicts the architectures in each case.

3.1. Stronger Annotations: Object Detection

Let us consider a set of training samples $\mathcal{D} = \{(x_i, B_i)\}$ where $x_i \in \mathbb{R}^{W\times H\times C}$ are images with spatial resolution $W \times H$ and C channels. Here, a set of bounding boxes is represented by $B_i = \{b_0, b_1, \ldots, b_N\}$ with $b = (c_x, c_y, w, h)$ being c_x and c_y the coordinates of the bounding box with size $w \times h$. For the sake of simplicity, we omitted the class label of each bounding box from the notation, as the people counting task is concerned exclusively with the person class. Then, in the training process of a neural network-based detector, we aim to learn a parameterized function $f_\theta : \mathbb{R}^{W\times H\times C} \to \mathcal{B}$, being $\mathcal{B}$ the family of sets B_i and θ the parameters vector. For such, the optimization is guided by a loss function, which is a combination of a regression $\mathcal{L}_{reg}$ and a classification $\mathcal{L}_{cls}$ term, i.e., l_2 loss and binary cross-entropy, respectively. Averaging these values for every detected instance and sample in $\mathcal{D}$ will yield a cost function that is a surrogate for the task's objective, counting the number of people y.

$$\mathcal{C}_{det}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,B)\in\mathcal{D}} \mathcal{L}_{cls}(f(x;\theta), B) + \lambda\mathcal{L}_{reg}(f(x;\theta), B) \tag{1}$$

Then, in the context of people counting, the number of people for a given input image is obtained as the number of valid bounding boxes within the network's output.

This study utilizes two kinds of detectors: YOLO and DINO. For YOLO, we use version 8, which is the latest model in this family [33] [30] [31] [32]. Such an architecture is divided into two main components: the backbone and the head. The backbone is a convolutional network that extracts feature maps from the input image. This process generates three scales from the feature maps $P_{i\in(3,5)}$. For each feature map P_i, the remaining feature maps P_j and P_k are either upsample using a bilinear interpolation or downsampled via a convolution layer before concatenation. Finally, a detection block outputs bounding boxes and a class prediction for each resulting feature map.

On the other hand, our second choice of detector is DINO, from the DETR family. It is a fully transformer-based model using SWIN [23] as a backbone. The SWIN transformer backbone is similar to ViT but uses different window sizes at every stage to learn different resolutions. The features yielded by each stage are sent as inputs to the encoder of the DINO detector. This detector is an end-to-end architecture using encoder and decoder blocks coupled with multiple prediction heads to output the prediction boxes.

3.2. Strong Annotations: Point-Level Localization

We employ point-level localization as the second level of supervision. Similar to the previous approach, we define a training set $\mathcal{S} = \{(x_i, P_i)\}$ being x_i the same input images. However, unlike the previous level, the annotation set P_i consists of a collection of pixel coordinates that identify a person's location, along with its confidence score, denoted as $p = (c_x, c_y, s), \forall p \in P_i$. The expected mapping takes the form of $g_\vartheta : \mathbb{R}^{W\times H\times C} \to \mathcal{P}$ where $\mathcal{P}$ represents the family of possible outputs P. Then, the task of finding the optimal parameters vector ϑ is solved using a Gradient Descent-based optimization that uses the cost function:

$$\mathcal{C}_{loc}(\theta) = \frac{1}{|\mathcal{S}|} \sum_{(x,P)\in\mathcal{S}} \mathcal{L}_{cls}(g(x;\vartheta), P) + \lambda \mathcal{L}_{loc}(g(x;\vartheta), P) \tag{2}$$

Here, $\mathcal{L}_{cls}$ is the binary cross entropy loss function, and $\mathcal{L}_{loc}$ is the l_2 loss between the estimated and the ground truth points. Following the previous setup, the number of people is determined as the number of estimated points with a confidence score higher than a given threshold.

Regarding point-level architectures, we employ P2P-Net [37] and Point query Transformer (PET) [19]. In the first approach, the network incorporates the first 13 convolutional layers from VGG-16 to extract deep features. These features are subsequently upsampled by a factor of two using nearest-neighbor interpolation and combined with a feature map from a lateral connection via element-wise addition. This lateral connection reduces the channel dimensions of the feature map after the fourth convolutional block. The merged feature map then undergoes a 3 × 3 convolutional layer, which helps mitigate the aliasing effect caused by the upsampling process. Finally, a prediction head with two branches generates point locations and confidence scores. Here, we use the same architecture for both branches, consisting of three stacked convolutional layers with interleaved ReLU activations.

On the other hand, the PET network employs a decomposable query process by dividing sparse points into new points and selectively querying them, particularly in densely populated areas. The framework comprises two main components: a point query quadtree and a progressive rectangle attention mechanism. It begins with input passed through a VGG16-based feature extraction backbone [36], allowing for scalable quantity estimation. A quadtree, transitioning from sparse to dense, splits each point into 4 points, adaptively partitioning points in crowded scenes. A CNN function encodes pixel localization as a point query, and a transformer decoder decodes the point queries before passing them through a prediction head to obtain crowd predictions. The framework supports different query numbers per image. It employs a progressive pooling approach based on horizontal windows due to its tendency to contain more people than a vertical window, adapting window size according to point density. The decoder's attention focuses on local windows to determine whether a point query corresponds to a person based on context. This approach improves computational efficiency while addressing the crowd-counting challenge.

3.3. Normal Annotations: Image-Level Counting

At the lowest level of supervision, we focus on image-level tasks. This task operates on a training dataset, denoted as $\mathcal{T} = \{(x_i, y_i)\}$, containing pairs of images, x_i, and the respective count of people present in each image, y_i. The parameterized mapping is defined as $h_\phi\colon \mathbb{R}^{W\times H\times C} \to \mathcal{Y}$, where $\mathcal{Y} = 0, 1, 2, \ldots, 20$. The optimization problem at this level aims to minimize a classification cost using the cross-entropy loss function or a regression cost with the l_2 loss. The estimated number of people corresponds to the appropriate class in the classification scenario.

At this level of annotation, the first architecture we consider is the Vision Transformer (ViT) [6]. The input images are divided into patches and injected with positional embeddings, later fed to the network. ViT consists of multiple encoder blocks arranged in sequence, followed by a classification head represented by a simple multi-layer perceptron. Each encoder block comprises a Self-Attention layer and a residual connection to a feedforward network. Such a Self-Attention layer's primary purpose is to model the relationship between different features and prioritize the most relevant ones. This is achieved by having the layer learn a representation for each feature, allowing it to compute the attention score –score of importance– for every pair of features using dot products of their respective vectors. The goal is to enable the model to capture long and short-range relationships between features. One drawback of ViT is its requirement for a substantial amount of data to converge. Hence, we chose to experiment with a pretraining approach.

Despite the success of ViT-based methods, the Convolutional Neural Networks have been the default deep learning model for vision-related tasks for a long time. Their success can be attributed to the induced bias introduced by the convolution layers, which encourages the model to learn short-range feature maps. This strong prior has helped the field of vision to reach several success stories in the last decade. The powerful track record of this model and the recent advances with the transformer have inspired researchers to build a more modern version of the CNN following the new architecture designs often used for Transformers. ConvNeXt [24], a member of the ResNets [15] and ResNexts [46] model families builds on the concept of residual layers and the aggregation of multiple transformation paths within a block. Like its predecessors, ConvNeXt is divided into four stages, where at the entrance of each, the inputs are down-sampled before going through a series of depthwise convolutions. The full overview of the ConvNeXt architecture can be viewed at Fig. 1.

3.3.1 Masked Autoencoder Pretraining

Unsupervised pretraining is a common practice in image classification. One promising technique employed for both

ViT and ConvNeXt is the Masked Autoencoder [14] [45] pretraining method. It is an unsupervised training objective to reconstruct a full image from a small, random portion of it. This strategy helps image classifiers learn more robust representations to improve classification accuracy.

Similar to standard Autoencoders, this method divides the model into two components: an encoder and a decoder. During pretraining, the classifier functions as the encoder, and the decoder is discarded during fine-tuning. In the training process, the encoder takes a random portion of the image as input, applies a mask to complete the missing parts, and then sends the output to the decoder to generate the full image. The mean squared error loss is computed by comparing the generated image with the original full image. Since this pretraining method was initially developed for image classification, its effectiveness was uncertain for people counting.

4. Results and Discussion

4.1. Experimental Methodology

1) Datasets: In our study, we employed two IR datasets. The first, LLVIP [16], comprises 15488 pairs of both RGB and IR versions of the same images. Although LLVIP offers both modalities, our focus in this work centers exclusively on IR video surveillance scenarios. As a result, we only used the IR images from the dataset. These IR images are captured by fixed outdoor cameras, providing frames from various street settings and with diverse perspectives. It's worth noting that the camera perspectives in the original training and testing sets differ. We merged and then re-split the training and testing sets to address the out-of-distribution scenario caused by the differences in camera perspectives between both sets. The dataset is split into 12025 images for training and 3463 images for testing. Each image has annotations consisting of bounding boxes enclosing each person, and the number of people per image can range from 0 to 13. Some examples of the images in the dataset can be seen in Fig. 2a.

The second dataset is the Distech IR [7]. This dataset consists of IR images capturing individuals in indoor settings across ten office rooms. It encompasses 2536 images, divided into their subsets: 1798 images in the training set, 483 images in the validation set, and 255 images in the testing set. The cameras are positioned statically, providing a top-view perspective. In addition, the dataset contains 2536 sequences, each comprising 64 frames, with only one frame annotated in each sequence. Each image is annotated with bounding boxes around a person. Unlike LLVIP, these frames only include temperature values. To reduce the effect of abnormal values, we applied Winsorization [13], a normalization technique that trims outlier values below 5% of the standard deviation and above 95% of the standard deviation. Distech IR images contain some people ranging between 0 and 12. Fig. 2b displays examples of office scenarios captured with Distech's cameras.

2) Experimental setup:

Image-level counters: Every image-level counting method in this study was trained using a consistent pipeline. We trained all models for 400 epochs using the AdamW optimizer and a batch size of 64. All additional training details specific to each model variation are provided in supplementary material. These parameters remained uniform for both LLVIP and Distech IR datasets. We used the ConvNeXt-Tiny and ConvNeXt-Micro architectures for both classification and regression. Additionally, our experiments involved the ViT-4L and ViT-3L networks. All models used the same data split to ensure a fair benchmarking process. We rounded the output to the nearest integer to determine the people count in the regression-based approaches. We applied the maximum-a-posteriori decision rule in classification-based approaches, matching the people count with the corresponding class.

Point-wise localization: For both P2P-Net and PET, each image undergoes a feature extraction process using a pre-trained VGG-16 backbone. The optimization is carried out using Adam optimizer [25] with a weight decay of 5×10^{-4}. P2P-Net, trained for 3500 epochs as suggested by its authors, utilizes specific parameters critical to its operation, including a window stride of 8, 4 reference points, and a backbone learning rate of 10^{-5}. The batch size for both models was set to 8. In the case of PET, which is typically trained for 1500 epochs, the CNN backbone (VGG16) and the transformer have learning rates of 10^{-5} and 10^{-4}, respectively. The point-query quadtree has a maximum depth of 2, and the initial sparse point query stride is $K = 8$. The transformer encoder and decoder layers share a common quadtree decoder. Window parameters are $s_e = 16$ and $r_e = 2$. Loss coefficients are $\lambda_1 = 5.0$ and $\lambda_2 = 0.1$, which help balance the loss function terms.

Object detectors: YoloV8 post-processing relies on two hyper-parameters: the confidence and non-maximal suppression thresholds, both within the range of 0 to 1. We tuned these thresholds for all detectors by selecting the values that resulted in the highest count accuracy on the validation set. Our experiments use YoloV8-L, YoloV8-M, and YoloV8-S architectures. The specific values chosen for each network are in the supplementary materials. For DINO, we train it for 12 epochs with a learning rate of 10^{-4}, a batch size of 4, and a weight decay of 10^{-4} using the SWIN-TINY backbone. The confidence threshold for DINO was tuned following the same procedure as before.

4.2. Main Comparative Results

In this section, we compare the three levels of supervision for people counting on LLVIP and Distech, and then

(a) LLVIP

(b) Distech IR

Figure 2. Examples of infrared images taken respectively from the Distech IR and LLVIP datasets.

Table 1. Count accuracy results on LLVIP

Model	Acc↑	MSE↓	MAE↓
YoloV8-L	**87.86 %**	0.191	0.160
YoloV8-M	87.80 %	0.182	0.156
YoloV8-S	86.36 %	0.215	0.180
DINO	87.38 %	**0.150**	**0.140**
P2PNet	56.22 %	0.955	0.578
PET	59.19 %	0.776	0.515
ConvNeXt-Tiny	80.13 %	0.239	0.211
ViT-4L	63.89 %	0.446	0.383
ConvNeXt-Micro	80.59 %	0.227	0.204
ViT-3L	61.29 %	0.491	0.421

Table 2. Count accuracy results on Distech IR

Model	Acc↑	MSE↓	MAE↓
YoloV8-L	90.02 %	0.173	0.123
YoloV8-M	88.15 %	0.154	0.133
YoloV8-S	**91.06 %**	0.185	0.123
DINO	90.98 %	**0.114**	**0.098**
P2PNet	76.47%	0.271	0.247
PET	84.70%	0.282	0.196
ConvNeXt-Tiny	88.24 %	0.721	0.266
ViT-4L	79.61 %	1.323	0.485
ConvNeXt-Micro	88.53 %	0.790	0.266
ViT-3L	78.04 %	1.218	0.442

the three levels of supervision on LLVIP with different amounts of training data. In addition, we analyze the impact of the counting model normal annotations of regression and classification heads, and the use of a masked autoencoder pre-training. Finally, we compare the computational and localization performance of methods.

1) Counting with different levels of supervision: Tables 1 and 2 present count accuracy, mean squared error, and mean absolute error results for each model on LLVIP and Distech IR, respectively. Based on count accuracy results, modern image-level approaches are beginning to narrow the performance gap with object detection techniques, even surpassing some state-of-the-art crowd counting methods such as P2PNet [37] and PET [19]. In the case of Distech IR, the ConvNeXt-Micro image classifier with Masked Autoencoder pretraining achieved highly competitive results across all YoloV8 sizes. For LLVIP, both ViT and ConvNeXt-based classifiers outperformed P2PNet and PET. However, it's important to note that, in LLVIP, the best classifier, ConvNeXt-Micro, achieved a count accuracy of 80.59%, while the best detector, YoloV8-L, reached 87.86%. Depending on the application and labeling budget, this may represent a reasonable trade-off. It is also worth noting that, despite not having the best accuracy, DINO still manages to have the lowest MSE and MAE for both datasets. This could mean that DINO's mistakes are closer to the correct count than the other models. Another interesting observation is that, despite having stronger supervision, localization models do not outperform the classifiers on LLVIP. This might be attributed to the fact that these models were initially designed to handle denser crowds.

2) Counting with different amounts of data: Based on the count accuracy results in LLVIP, it is evident that with even the proper training regimen, ConvNeXt-Tiny can't entirely close the performance gap with a model trained with stronger supervision like YoloV8-L for the same amount of data. Our results show that ConvNeXt-Tiny achieved an 80.13 % count accuracy, whereas YoloV8-L achieves 87.86 %. Considering the cost difference of labeling, we believe that those performances nonetheless establish ConvNeXt-Tiny as a potentially viable solution for specific applications. However, we were keen to understand how much training data ConvNeXt would need to reach the performance of YoloV8. To gain a better perspective, we trained both models using various portions of the training data, ranging from 10 % to 100 % of LLVIP with 10 % increments. This experiment allowed us to find a relationship between normal and stronger annotations. We found that, in practice, YoloV8-L and YoloV8-M required respectively an approximately 16 % and 17 % of the amount of labeled data used for ConvNeXt-Tiny to achieve equivalent performance. On the other hand, as mentioned earlier in this paper, bounding box annotations are considerably more time-consuming and challenging to produce than image-level an-

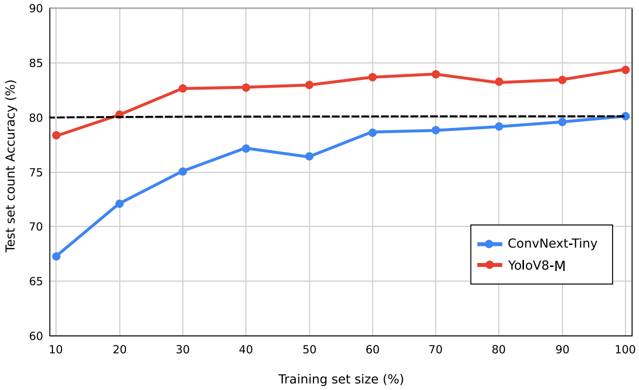


Figure 3. Count accuracy on the test set depending on the training set size for LLVIP

notations, which involve specifying the number of people in the image. A graph displaying count accuracy on the testing set as a function of the amount of data can be seen in Fig. 3.

3) Ablations on image-level counting: Tab. 3 and 4 show the obtained result over LLVIP and Distech IR datasets for image-level settings. Let's start by looking at the performances of both architectures: ViT and ConvNeXt. Contrary to our expectations, considering the impressive performances of Transformers on challenging benchmarks like ImageNet, the ConvNeXt significantly outperforms ViT by a substantial margin of at least 14% in LLVIP and 3% in the case of Distech IR. This outcome can be attributed to two key factors. First, we utilized the full ConvNeXt model pre-trained on ImageNet, providing a competitive parameter count advantage compared to using only the first four layers of ViT-S. Second, the Transformer, as indicated in the literature [27], tends to excel with large and diverse datasets, which is not the case in our settings. Following with using pretraining to improve accuracy, our data shows that the masked auto-encoding method consistently yields better results for all classifier configurations, suggesting that MAE pretraining is also a suitable approach for people counting.

As discussed, we also aim to determine the optimal modeling approach – classification or regression – for image-level counting. Our results show that, in the case of LLVIP, regression consistently outperforms classification, regardless of whether pretraining is used. However, the opposite holds for Distech IR, where classification outperforms regression. Upon analyzing the class distribution of both datasets, we observed that the LLVIP dataset exhibits a significantly more balanced distribution, with a standard deviation of 9.86% across class occurrences, as opposed to 16.03% for the other dataset.

Table 3. Count accuracy for different configurations of ConNext and ViT for LLVIP.

Model	**Pretrain**	**Head**	**Acc↑**	**MSE↓**	**MAE↓**
ConvNeXt-Tiny	None	Class. Regr.	78.74 % 79.84 %	0.251 0.238	0.224 0.213
	MAE	Class. Regr.	79 .69 % 80.13 %	0.249 0.239	0.215 0.211
ViT-4L	None	Class. Regr.	62.36 % 62.94 %	0.480 0.478	0.408 0.403
	MAE	Class. Regr.	63.20 % 63.89 %	0.505 0.446	0.409 0.383
ConvNeXt-Micro	None	Class. Regr.	77.99 % 79.03 %	0.264 0.253	0.233 0.223
	MAE	Class. Regr.	78.14 % 80.59 %	0.274 0.227	0.235 0.204
ViT-3L	None	Class. Regr.	58.15 % 61.06 %	0.614 0.496	0.479 0.420
	MAE	Class. Regr.	61.70 % 61.29 %	0.515 0.491	0.424 0.421

Table 4. Count accuracy for different configurations of ConNext and ViT for Distech IR.

Model	**Pretrain**	**Head**	**Acc↑**	**MSE↓**	**MAE↓**
ConvNeXt-Tiny	None	Class. Regr.	82.75 % 82.35 %	0.787 0.771	0.294 0.325
	MAE	Class. Regr.	88.24 % 86.28 %	0.721 0.697	0.266 0.274
ViT-4L	None	Class. Regr.	79.22 % 76.08 %	1.072 1.556	0.407 0.545
	MAE	Class. Regr.	79.61 % 67.06 %	1.323 1.429	0.485 0.607
ConvNeXt-Micro	None	Class. Regr.	85.10 % 78.82 %	0.791 0.940	0.290 0.392
	MAE	Class. Regr.	88.24 % 86.67 %	0.790 0.678	0.266 0.254
ViT-3L	None	Class. Regr.	83.53 % 70.98 %	0.936 1.400	0.341 0.545
	MAE	Class. Regr.	78.04 % 67.84 %	1.218 1.194	0.442 0.529

4.3. Models size and computational cost

The inference time required for each model becomes crucial when selecting the right model for a solution. While the parameter count can indicate the expected speed performance among models within the same family (e.g., YoloV8), it may not effectively characterize the speed of a model relative to another model type with a similar parameter count. We conducted a frame-per-second benchmark on GPU and CPU for every model to facilitate a fair comparison of models across different types. All benchmarks were performed under the same conditions. The GPU used for benchmarking is an NVIDIA A100 SXM4 with 40GB of memory, and the CPU is AMD EPYC 7413 24-core Processor. Each benchmark started with 100 warm-up steps, followed by 10000 inferences with a batch size of 1. We report the mean of all repetitions. The results can be viewed at Tab. 5. As observed in the table, image-level counting achieves the highest efficiency on the CPU, as expected. This is a desirable feature for intelligent building applications that require close-to-real-time responses.

Table 5. Number of parameters and frames-per-second (FPS) on GPU and CPU for all evaluated models.

Model	Parameters	FPS (GPU)	FPS (CPU)
YoloV8-L	44 M	103.16	14.07
YoloV8-M	26 M	125.33	17.21
YoloV8-S	11 M	162.5	28.46
DINO-SWIN-T	48 M	67.72	22.17
P2PNet	22 M	365.41	27.5
PET	21 M	70.38	9.32
ConvNeXt-Tiny	29 M	127.92	30.45
ConvNeXt-Micro	24 M	160.35	32.84
ViT-4L	30 M	316.3	51.88
ViT-3L	23 M	404.54	60.21

4.4. Localization

Additionally, the counting capabilities of a model, we also want to evaluate its capability to localize an object. For this, we evaluate the capabilities of our models to localize the center of the objects in an image by measuring the mean Average Euclidean Distance (mAED) between the ground truth and the closest estimated localization points. The details of the measurement are given in supplementary materials. To find the best configuration of matched points, we use the Hungarian algorithm [17].

For the point-wise localization methods, we obtain directly the object center's x, y coordinates. In detection, the localization coordinates are considered as the center of the detected objects. However, a classifier trained without explicit localization information does not output the objects' x, y coordinates. Therefore, we employ Class Activation Maps [51] to extract the objects' center. Additional details regarding the algorithm used to find the positions from the activation maps can be found in the supplementary material.

In Tab. 6, we present the mAED results for our most promising models: YoloV8-L for detection, PET for point-wise localization, and ConvNeXt for image-level counting. Despite the lack of location information in the labeling, ConvNeXt-Tiny achieved comparable results to YoloV8-L and outperformed PET. ConvNeXt localization excels when individuals are widely separated (Fig. 4). However, the localization becomes imprecise when multiple people close, as evidenced in the first two images from ConvNeXt-Tiny. Such an issue is related to the low resolution of the activation maps, which limits the algorithm to delineate each maximum within closely clustered points accurately.

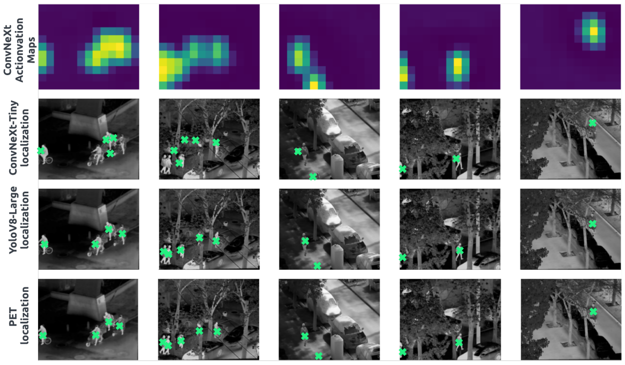

ConvNeXt Actionvation Maps
ConvNeXt-Tiny localization
YoloV8-Large localization
PET localization

Figure 4. Examples of localizations of ConvNeXt, YoloV8 and PET

Table 6. Mean average Euclidian Distance between ground truth points and estimated localization points for our algorithms with different level of supervision

Model	Supervision	mAED
YoloV8-L	Stronger	0.12582
PET	Strong	0.17054
ConvNeXt	Normal	0.16857

5. Conclusions

In this study, we present key insights into people counting. Our analysis highlights several significant findings. First, regression heads for classifiers enhance performance, especially when dealing with balanced people counts, emphasizing the importance of data distribution in model outcomes. In addition, the Masked Autoencoder pretraining technique demonstrates its adaptability, proving effective not only in identifying object classes but also inaccurate people counting. Thirdly, our experiments reveal limitations in top-performing crowd counting methods based on point-level localization, particularly in sparse crowd scenarios, suggesting the need for further exploration in optimizing their efficiency. Moreover, this research underscores the feasibility of employing cost-effective annotation methods without compromising count accuracy, making people counting more accessible and affordable. Lastly, the study challenges the common belief that more supervision inherently translates to better results, as point-level methodologies under-performed compared to level supervision classifiers, underscoring the importance of aligning model design with crowd density for optimal outcomes.

Acknowledgements: This work was supported by Distech Controls Inc., NSERC, and MITACS. The authors thanks Heitor Rapela Medeiros and Masih Aminbeidokhti for the valuable discussion.

References

[1] Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *CoRR*, abs/2102.06171, 2021. 3

[2] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. Reversible column networks, 2023. 3

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. 2

[4] Zhuojun Chen, Junhao Cheng, Yuchen Yuan, Dongping Liao, Yizhou Li, and Jiancheng Lv. Deep density-aware count regressor. *CoRR*, abs/1908.03314, 2019. 2

[5] Zhi-Qi Cheng, Qi Dai, Hong Li, Jingkuan Song, Xiao Wu, and Alexander G. Hauptmann. Rethinking spatial invariance of convolutional networks for object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19638–19648, June 2022. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 1, 3, 4

[7] Thomas Dubail, Fidel Alejandro Guerrero Peña, Heitor Rapela Medeiros, Masih Aminbeidokhti, Eric Granger, and Marco Pedersoli. Privacy-preserving person detection using low-resolution infrared cameras, 2022. 5

[8] Zizhu Fan, Hong Zhang, Zheng Zhang, Guangming Lu, Yudong Zhang, and Yaowei Wang. A survey of crowd counting and density estimation based on convolutional neural network. *Neurocomputing*, 472:224–251, 2022. 1, 2

[9] Luca Fiaschi, Ullrich Koethe, Rahul Nair, and Fred A. Hamprecht. Learning to count with regression forest and structured labels. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 2685–2688, 2012. 2

[10] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. 2

[11] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. 2

[12] Riadh Habash. 1 - building as a system. In Riadh Habash, editor, *Sustainability and Health in Intelligent Buildings*, Woodhead Publishing Series in Civil and Structural Engineering, pages 1–32. Woodhead Publishing, 2022. 1

[13] Cecil Hastings Jr, Frederick Mosteller, John W Tukey, and Charles P Winsor. Low moments for small samples: a comparative study of order statistics. *The Annals of Mathematical Statistics*, 18(3):413–426, 1947. 5

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021. 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 4

[16] Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. LLVIP: A visible-infrared paired dataset for low-light vision. *CoRR*, abs/2108.10831, 2021. 5

[17] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955. 8

[18] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M. Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13619–13627, June 2022. 2

[19] Chengxin Liu, Hao Lu, Zhiguo Cao, and Tongliang Liu. Point-query quadtree for crowd counting, localization, and more. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1676–1685, October 2023. 1, 3, 4, 6

[20] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022. 2

[21] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. *CoRR*, abs/1811.10452, 2018. 2

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021. 3

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021. 3

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CoRR*, abs/2201.03545, 2022. 4

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[26] Chen Change Loy, Ke Chen, Shaogang Gong, and Tao Xiang. *Crowd Counting and Profiling: Methodology and Evaluation*, volume 11. 10 2013. 2

[27] Zhiying Lu, Hongtao Xie, Chuanbin Liu, and Yongdong Zhang. Bridging the gap between vision transformers and convolutional neural networks on small datasets, 2022. 7

[28] Yiming Ma, Victor Sanchez, and Tanaya Guha. Fusioncount: Efficient crowd counting via multiscale feature fusion. In *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, oct 2022. 2

[29] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3253–3261, 2015. 2

[30] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015. 2, 3

[31] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016. 2, 3

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 2, 3

[33] Peiming Ren, Wei Fang, and Soufiene Djahel. A novel yolo-based real-time people counting approach. 09 2017. 2, 3

[34] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. 2

[35] Anthony D. Rhodes, Max H. Quinn, and Melanie Mitchell. Fast on-line kernel density estimation for active object localization. *CoRR*, abs/1611.05369, 2016. 3

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4

[37] Qingyu Song, Changan Wang, Zhengkai Jiang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yang Wu. Rethinking counting and localization in crowds: A purely point-based framework. 2021. 1, 3, 4, 6

[38] Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang. Fishnet: A versatile backbone for image, region, and pixel level prediction. In *Advances in Neural Information Processing Systems*, pages 760–770, 2018. 1

[39] Nguyen Hoang Tran, Ta Duc Huy, Soan T. M. Duong, Phan Nguyen, Dao Huu Hung, Chanh D Tr Nguyen, Trung Bui, and QUOC HUNG TRUONG. Improving local features with relevant spatial information by vision transformer for crowd counting. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. 3

[40] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. Distribution matching for crowd counting. *CoRR*, abs/2009.13077, 2020. 2

[41] Q. Wang and T.P. Breckon. Crowd counting via segmentation guided attention networks and curriculum loss. *IEEE Trans. Intelligent Transportation Systems*, 2022. to appear. 3

[42] Shuai Wang and Zhendong Su. Metamorphic testing for object detection systems. *CoRR*, abs/1912.12162, 2019. 3

[43] Yi Wang, Junhui Hou, Xinyu Hou, and Lap-Pui Chau. A self-training approach for point-supervised object detection and counting in crowds. *CoRR*, abs/2007.12831, 2020. 1

[44] Yi Wang and Yuexian Zou. Fast visual object counting via example-based density estimation. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3653–3657, 2016. 2

[45] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders, 2023. 5

[46] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. 4

[47] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. VOLO: vision outlooker for visual recognition. *CoRR*, abs/2106.13112, 2021. 3

[48] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. *CoRR*, abs/2106.11342, 2021. 2

[49] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[50] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016. 3

[51] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015. 8