Bilinear Magic Meets CLIP: Unsupervised Learning in Pig Counting Scenario

Yue Sun, Lili Yang*

College of Information and Electrical Engineering, China Agricultural University
Beijing, China
llyang@cau.edu.cn

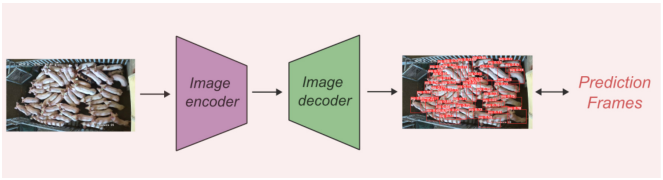
Abstract

Pig counting is an important task in pig marketing and farming. Most of the existing methods use supervised object detection for counting. However, supervised pig counting relies heavily on expensive data labeling, especially in the dense counting scenario. To address this issue, we propose BU-CLIP, which adopts the Contrastive Language Image Pre-training model (CLIP) for unsupervised pig counting. We tailor the image encoder of CLIP and the loss function to make it more suitable for pig counting. Our method replaces CLIP's image encoder with a bilinear model combining ConvNeXt and ResNet50 backbones, while the pooling operation is performed via a multi-head attention module. We reconstruct the loss function by using a multi-modal full-ranking loss, which captures the intrinsic correspondence between text and image. The proposed model is tested on a dense pig counting dataset, and extensive experiments demonstrate that our method outperforms unsupervised state-of-the-art counting methods and achieves almost the same results as supervised state-of-the-art counting methods.

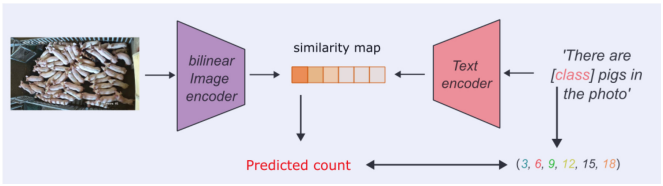
Introduction

With the development of agricultural informatization, the pig farming industry is rapidly transforming to intensification, scale, and automation (Wang et al. 2022). The stock of pigs is the base number for the estimation of fixed assets in the pig farming industry. It is also an important indicator reflecting the pig survival rate and the supply-demand relationship in the market.

At present, pig counting relies heavily on manual inspection, which is time-consuming and labor-intensive (Lee et al. 2022; Xia et al. 2025). In addition, during manual counting, the contact between the handler and the pig increases the risk of zoonotic disease transmission. Although electronic ear marks have been used for counting, contact between pigs can lead to erroneous reporting (Brown-Brandl, Rohrer, and Eigenberg 2013; Yu et al. 2025), and there is a risk of ear marks becoming dislodged or damaged when pigs scratch or in muddy environments. Frequently, counting the number of pigs in grouped barns is a key management task for large pig farms. Pigs are often moved to different barns at different growth stages or sizes. Farmers need to know how many pigs are in each large barn. Recent advancements in surveillance technology and computer vision allow researchers to efficiently count, locate, and track pigs (Hao et al. 2023), eliminating the need for labor-intensive manual methods and ensuring a certain level of accuracy.



(a) Supervised methods: rely on box-level labels



(b) Our BU-CLIP model: without any labeled images

Figure 1: (a) Supervised methods use box-level or point-level labels for learning. (b) Our unsupervised pig counting method uses alignment matching of linguistic knowledge and image patches.

Existing pig counting methods typically rely on variations of YOLO and CNN models, or combined. Therefore, image labeling is necessary for model training. However, when there are too many pigs, labelling them becomes a challenge. Dense pig images may exist with occlusions and overlapping, increasing time consumption and task difficulty.

To address this issue, we propose BU-CLIP, a method for unsupervised pig counting based on CLIP (Radford et al. 2021) with bilinear image encoders, as shown in Fig. **??**. We also innovatively tailor a loss function in CrowdCLIP (Liang et al. 2023) by calculating the maximum value of the upper and lower triangular loss, ensuring better alignment between the images and text prompts. Furthermore, we tailor the text prompts to better suit the pig distribution. Extensive

*Corresponding author
Accepted at the First International Workshop on AI in Agriculture (Agri AI), co-located with AAAI 2026.

experiments show that our method outperforms state-of-the-art methods, with an RMSE of 6.95. Compared to unsupervised models like CrowdCLIP (Liang et al. 2023) and CSS-CCNN (Sam et al. 2020), our RMSE improves by 32.5% and 64.1%, respectively. Our main contributions are as follows:

- We propose a CLIP-based unsupervised pig counting method that innovatively tailors the text prompts and the loss function for pig counting. To our knowledge, this is the first unsupervised model in pig counting task.
- We collect and produce a dense dataset of pigs from the top-view angle, which is publicly available for academic purposes.
- We conduct extensive experiments on our proposed model, demonstrating that our model outperforms unsupervised state-of-the-art models.

Related works

Counting method

Fully-supervised counting. Fully-supervised counting methods often involve regressing density maps, which are generated from labeled dot maps (Liang et al. 2022). Convolutional Neural Networks (CNNs) (Babu Sam, Surya, and Venkatesh Babu 2017; Walach and Wolf 2016; Zhang et al. 2024; Tian et al. 2025) are used for the task of counting through density maps. By optimizing basic CNN, it's possible to improve the counting effectiveness for multi-scale targets (Zhang et al. 2016; Babu Sam, Surya, and Venkatesh Babu 2017), or to develop a dense detection model for crowd counting. Alternatively, fully-supervised models like YOLO can also be used to delineate targets, transforming image recognition into a regression problem for counting tasks (Redmon et al. 2016).

Weakly-/semi-supervised counting. To reduce the workload of manual labeling (Liu et al. 2022a; Liang et al. 2022; Bellocchio et al. 2019) or when precise annotation is not feasible (Lin and Chan 2023; Lin et al. 2022), weakly-/semi-supervised counting methods have been proposed. These methods add annotation algorithms into existing models and generate hard pseudo-labels for unlabeled images (Lin and Chan 2023; Li et al. 2023). Alternatively, new modules are used to supervise unlabeled regions directly by propagating features based on representative regions (Liu et al. 2022a). These methods simplify the annotation process, but they still can't completely avoid the time-&labor-consuming image annotation.

Unsupervised counting. Unsupervised counting models include CSS-CCNN (Sam et al. 2020) and CrowdCLIP (Liang et al. 2023). CSS-CCNN assumes that natural crowds follow a power-law distribution, which can be used to generate error signals for backpropagation. CrowdCLIP constructs ranking text prompts to match the size-sorted image patches and utilizes multi-modal ranking loss to guide image encoder learning. Alternatively, DSSINet (Liu et al. 2020) based on regression-detection bi-knowledge transfer, utilizes a pedestrian detector based on center and scale prediction (CSP) as the detection model.

Pig counting method

For pig counting, common supervised counting methods include YOLO models (Huang et al. 2023; Redmon et al. 2016) and Convolutional Neural Networks (CNN) (Chen et al. 2020; Zhang et al. 2015). Some approaches optimize existing models, such as YOLOv5, which embeds two different feature block sizes of SPP networks into the network (Huang et al. 2023). Alternatively, modifications can be made to CNN using new modules (Tian et al. 2019) or network structures to achieve accurate pig counting (Feng, Wang, and Zhou 2023). However, annotating images is necessary for all these models, which undoubtedly results in a significant workload. Meanwhile, lower-quality datasets can increase the difficulty of annotation as well. No unsupervised method has been proposed for pig counting.

Method

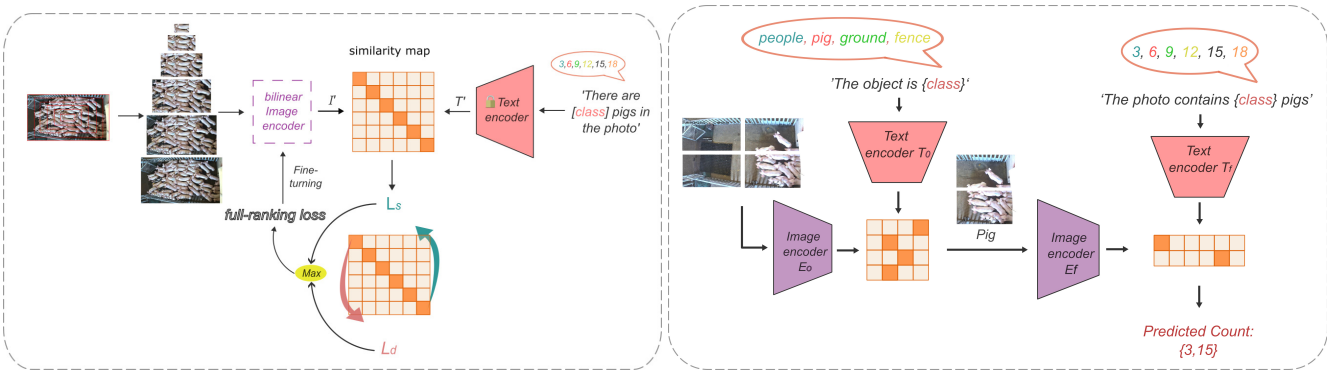
In this paper, we propose BU-CLIP for the pig counting task. An overview of our method is shown in Fig. 2. We first introduce a bilinear model as the image encoder of CLIP and add attention mechanisms.

Bilinear model. The bilinear model was initially proposed in (Lin, RoyChowdhury, and Maji 2015) for fine-grained image recognition tasks (Lin and Maji 2017; Wu and Wang 2017). Later, it was widely used in computer vision. Its strong feature extraction capability results in good performance in fine-grained recognition, image classification, and other related fields. In this study, we use an image encoder based on bilinear model to enhance the expression of image features in the CLIP model.

Bilinear Image Encoder

Image encoder. In dense counting, fine-grained feature extraction is necessary due to the large number of subjects and small targets. The structure of the bilinear image encoder used in our model is shown in Fig. 3. This encoder consists of two parallel backbone networks, an attention module, and a Multi-head Attention Pooling module. The images are resized to 224×224 before being fed into the convolutional network. For the features $I \in \mathbb{R}^{N \times C \times H \times W}$ fed into the bilinear image encoder, two sets of feature mappings I_1 and I_2 are obtained after downsampling by two parallel backbone networks, ConvNeXt (Liu et al. 2022b) and ResNet50 (He et al. 2015). The two backbone networks update their parameters independently to provide a multi-form representation of the fine-grained features.

The two feature mappings are then fused using a matrix multiplication operation. To improve the feature representation, the fused feature mappings are fed into the Global Attention Module (GAM) (Liu, Shao, and Hoffmann 2021) for deep feature extraction, where GAM uses large kernel convolution to increase the receptive field and perceive fine-grained features in both channel and spatial dimensions. Finally, the feature vector $I' \in \mathbb{R}^{M \times C}$ is output by the Multi-head Attention Pooling (MAP) module.



(a) Training phase (b) Testing phase

Figure 2: Overview of our model. (a) During the training phase, we use image patches and text prompts to generate a similarity map and compute the full-ranking loss to fine-tune the image encoder. (b) During the testing phase, the images are divided into four patches, and a simple filtering strategy is used.

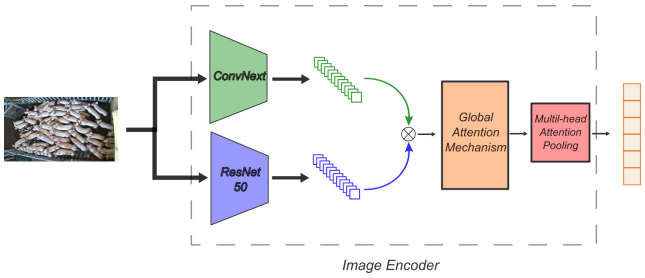


Figure 3: The structure of the bilinear model in image encoder. This module contains ConvNeXt and ResNet50 backbones, Global Attention Mechanism and Multi-head Attention Pooling.

Ranking-based Contrastive Fine-tuning

This section describes how BU-CLIP can be fine-tuned to improve the ability to extract pig semantics. In unsupervised counting, images are unlabelled, which means that fine-tuning lacks corresponding supervision. To address this issue, we refer to CrowdCLIP to construct matching size-sorted images and text prompts. Then the full-ranking loss is used for fine-tuning.

The training phase is shown in Fig. **??**(a). Firstly, the input image is centrally cropped to create size-sorted patches. Then, ranking text prompts are created using count-interval text to match the size-sorted patches. During training, the text encoder is frozen, and the image encoder is fine-tuned using full-ranking loss.

Construction of size-sorted patches. When inputted with an image of pigs, we crop a set of patches, denoted as O_m, where m represents the predetermined number of patches. These patches are subject to the following rules: For any two cropped image patches ($\mathcal{O}_i$ and $\mathcal{O}_j$, $0 \leq i \leq j \leq M-1$), the size of patch $\mathcal{O}_i$ is smaller than the size of patch $\mathcal{O}_j$.

Notice that the patches from the same image have the same center. During training, all patches are resized uniformly and fed into the image encoder to generate the image ranking embeddings $I' = \left[I'_0, I'_1, \ldots, I'_{M-1}\right]$, where $I' \in \mathbb{R}^{M \times C}$ as introduced in the previous section.

So we are guaranteed that the number of pigs contained in patch $\mathcal{O}_i$ will not exceed the number of pigs contained in patch $\mathcal{O}_j$.

Construction of ranking text prompt. Clearly, the number of pigs in different patches is sorted, with larger patches corresponding to a larger or equal number of pigs. Therefore, we have constructed ranking text prompts to describe the sorted relationship of image patches. The text prompt is defined as 'There are [class] pigs in the photo', where [class] represents a set of sequential numbers $T = [T_0, T_0 + K, \ldots, T_0 + (N-1)K]$, where T_0, K, and N denote the basic reference count, counting interval, and several classes.

We input text prompts into the text encoder to obtain the output text ranking embeddings $T' = \left[T'_0, T'_1, \ldots, T'_{N-1}\right]$, where $T' \in \mathbb{R}^{N \times C}$.

Image encoder optimization. Our tailored loss function is calculated as shown in Fig. 4. Suppose we have a set of image ranking embeddings $I' \in \mathbb{R}^{M \times C}$ and text ranking embeddings $T' \in \mathbb{R}^{N \times C}$ obtained from image and text encoders as obtained from the previous sections.

For the image-language matching pipeline, we calculate the similarity scores between I' and T' using the inner product and obtain the similarity matrix $S = [s_{i,j}]$, where $S \in \mathbb{R}^{M \times N}$:

$$s_{i,j} = I'_i \cdot T'^T_j \tag{1}$$

where $I'_i \in \mathbb{R}^{1 \times C}, T'_j \in \mathbb{R}^{1 \times C}, 0 \leq i \leq M-1$ and $0 \leq j \leq N-1$. Our goal is to make the similarity matrix S into a specific sorting matrix (Fig. **??**(a)) and inherit the ranking relationships between the image and text prompts.

$$s_{i',i} \leq s_{i,i} \tag{2}$$

where $i' \neq i$. In order to preserve the order of the image-text pair in the latent space, we optimize the image encoder us-

ing the full-ranking loss. Specifically, we compute the upper ranking loss L_s top-down, based on the main diagonal of the similarity matrix:

$$L_s = \max\left(0, s_{i',i} - s_{i,i}\right), 0 \leq i' \leq i \leq M-1 \quad (3)$$

At the same time, we compute the lower ranking loss L_d bottom-up, based on the main diagonal of the similarity matrix:

$$L_d = \max\left(0, s_{i',i} - s_{i,i}\right), 0 \leq i \leq i' \leq M-1 \quad (4)$$

where L_d and L_s represent the lower and upper ranking losses of the similarity matrix, respectively. For the convergence of our model, the final loss is the maximum of L_d and L_s:

$$L = \max\left(L_d, L_s\right) \quad (5)$$

In practice, we set $M = N$ to guarantee that S is a square matrix. Eq. 2 and Eq. 3 establish an inherent correspondence between size-sorted patches and ranking text prompts, producing the similarity matrices. During fine-tuning, we freeze the weights of the text encoder, allowing the image embedding to align with the fixed language space. Text embeddings are confined within the learned language space, resulting in robust generalization.

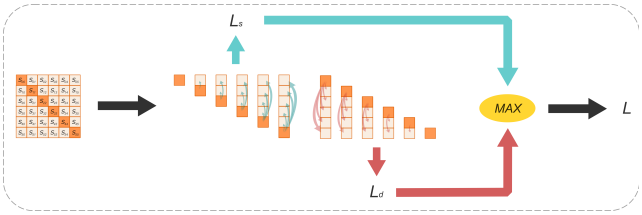


Figure 4: Calculate the upper and lower triangular ranking loss of the similarity map and take the maximum of them as the final loss value.

We use a different ranking loss than CrowdCLIP. They only compute the upper ranking loss L_s. This loss function means that image patches with fewer pigs will match larger text prompts, and not all cases will be covered. By separately computingL_s and L_d and comparing their values, we better match image patches and text prompts and improve fine-tuning convergence.

Prediction Process Based on Filtering Strategy

This section introduces the prediction phase of our model. For the input images, grid cropping is performed, and a filtering strategy is used. The strategy has two stages for selecting real pig patches and mapping them to proper count intervals, as shown in Fig. 2(b).

For convenience, we name the original image encoder and fine-tuned image encoder as E_0 and E_f, respectively. E_0 is used to select high-confidence pig patches, while E_f is used for the final counting.

Given an input image, we first divide it into a grid of $N \times N$ patches. These patches and their text prompts are then fed into E_0 and the text encoder T_0 to generate similarity scores for coarse classification. The text prompt used for T_0 is 'The object is [class]', aiming to classify the patches into different categories with clear distinction, such as 'pig', 'farmer', or 'fence'.

In the second stage, we use the fine-tuned E_f as the image encoder, and the text prompts are defined as 'The photo contains [class] pigs', where [class] is the pre-defined ranking number r. The final count can be obtained by selecting the most similar image-text pair based on the image embedding and class embedding from E_f and T_f.

Experiments and Analysis

Dataset and Evaluation Metric

Dataset: In this paper, we independently construct a clear and dense pig counting dataset. We use a camera on the farm to take distant top-view shots, avoiding the problem of close-up occlusion. Furthermore, we selected the shooting time of the pigs being driven out of the pen, i.e., when the pig density is the highest and the morphology of the subjects has changed the most. The video recording resolution is 2560 × 1440 with a frame rate of 25 FPS. We extract image information every 5 frames, eliminate similar images, and finally obtain 622 images. The number of pigs in the images ranges from 1 to 70, as shown in Fig. 5. The data set is divided into a training set and a testing set in an 8:2 ratio.

Evaluation Metric: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are used as evaluation metrics to assess counting performance.

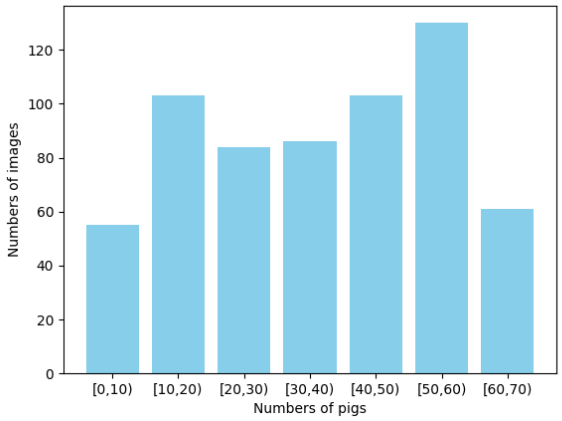


Figure 5: Population distribution of pigs.

Experimental setup

All experiments are conducted with an NVIDIA A4000 GPU. During the fine-tuning phase, the text encoder's parameters are frozen, and the RAdam optimizer (Liu et al. 2019) is used to optimize the image encoder with a learning rate of 1e-5 and momentum of 0.9. The training lasts for 50 epochs, and both the text and image feature vector lengths M and N were set to 6, with 512 channels. For the ranking

text prompts, we set the base reference count R_0 to 3 and the count interval K to 3. This means that the text prompt statement is 'There are [3/6/9/12/15/18] pigs in the photo'. During the testing phase, we set the number of image segmentations P to 2.

Result

Tab. 1 shows the comparison results of the performances of our method and other state-of-the-art counting methods on our pig counting dataset. MAE and RMSE are used for performance evaluation.

Table 1: Comparison of the counting performance on our pig counting dataset. Random denotes that we randomly select a value between the range from 0 to the maximum number of pigs.

Method	Label	MAE	RMSE
Random	–	21.85	26.62
CSS-CCNN (Sam et al. 2020)	None	15.97	19.34
CrowdCLIP (Liang et al. 2023)	None	8.19	10.30
YOLOv8	Box	5.28	7.55
BU-CLIP (Ours)	**None**	**5.60**	**6.95**

A series of YOLO algorithms have been used in pig counting tasks with good performances, so we choose the state-of-the-art YOLOv8 model as the baseline for the fully-supervised counting method. CrowdCLIP (Liang et al. 2023) is also based on the CLIP model for unsupervised learning, while CSS-CCNN (Sam et al. 2020) utilizes the power-law distribution to count subjects. So we chose these two methods as our baseline for unsupervised counting. On the pig counting dataset, our method achieves an MAE of 5.60 and an RMSE of 6.95, improving by 32.5% and 64.1% compared to CrowdCLIP and CSS-CCNN in RMSE, respectively. It is noteworthy that our model even performs comparably to the fully-supervised YOLOv8 model. Through experiments conducted in dense pig populations, we validate the effectiveness of our proposed method. This method achieves accurate counting in pig farming environments and significantly reduces the cost of manual labeling.

Ablation study

The following experiments are conducted on our dataset, which contains a range of 1 to 70 pigs per image. Some images have dense and large numbers of pigs, while others have smaller numbers, making them suitable for ablation experiments in various situations.

Validity analysis of the bilinear model. We first test the validity of the bilinear model. The bilinear model is then replaced with a single model, and the results are shown in Tab. 2. It can be observed that after replacing the bilinear model with ConvNeXt (Liu et al. 2022b), the RMSE rises to 14.49, and after replacing the bilinear model with ResNet50 (He et al. 2015), the RMSE rises to 8.35. Using two different backbone networks can get a different granularity of feature mapping for multi-scale feature fusion, which can express the more complex pig features. This result demonstrates that the bilinear model can produce superior outcomes when replacing the image encoder of CLIP.

Table 2: Effect of different backbones as image encoder on counting performance

Image encoder	MAE	RMSE
ConvNeXt (Liu et al. 2022b)	9.82	14.49
ResNet50 (He et al. 2015)	5.93	8.35
Bilinear model	**5.60**	**6.95**

Validity analysis of loss functions. We further test the impact of choosing different loss functions when fine-tuning the image encoder. We select the ranking loss L_s, L_d, and the sum of them, as well as the MAX loss function used in our model. The results are shown in Tab. 3, where it can be observed that the MAX loss function is most effective. L_s can only address the ranking loss between prompts describing a big number and image patches containing a small number of pigs, while L_d is designed to cope with matching errors between prompts describing a small number and image patches containing a big number of pigs.

Table 3: Ablation experiments on the ranking loss.

Loss	MAE	RMSE
L_s	10.23	12.12
L_d	7.13	9.20
SUM	8.89	11.67
MAX	**5.60**	**6.95**

The influence of patch number. We verify the effect of using different numbers of image patches during the testing phase. During the testing phase, the number of patches P is set to 1, 2, 3, and 4. The results are shown in Tab. 4. It can be observed that when P is set to 1 and 4, the results are not good, while P is set to 3, the results are better, and the RMSE is 10.99. Our explanation for this is that when P is set to 1, the range of pigs contained in one patch is too large for prediction using text prompts. When P=4, a single image is cut into 16 patches, which is excessive.

Table 4: Ablation experiments on the patch number design.

Patch Number	MAE	RMSE
1	24.57	28.35
2	**5.60**	**6.95**
3	8.98	10.99
4	22.31	25.84

Conclusion

In this paper, we propose a novel framework, BU-CLIP, for the unsupervised pig counting task based on CLIP. We use a bilinear model combining ConvNeXt and ResNet50 backbones to replace CLIP's image encoder. Then Global Attention Mechanism (GAM) and Multi-head Attention Pooling (MAP) are used to perform fine-grained feature extraction

on the data to enhance the expression of image features. To fine-tune the image encoder without labels, we constructed size-sorted text and images, and innovatively tailored a full-ranking loss to get better alignment of image and text. We conduct experiments on dense pig counting datasets and demonstrate the superiority of BU-CLIP over the state-of-the-art unsupervised models.

References

Babu Sam, D.; Surya, S.; and Venkatesh Babu, R. 2017. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5744–5752.

Bellocchio, E.; Ciarfuglia, T. A.; Costante, G.; and Valigi, P. 2019. Weakly supervised fruit counting for yield estimation using spatial consistency. *IEEE Robotics and Automation Letters*, 4(3): 2348–2355.

Brown-Brandl, T.; Rohrer, G.; and Eigenberg, R. 2013. Analysis of feeding behavior of group housed growing–finishing pigs. *Computers and Electronics in Agriculture*, 96: 246–252.

Chen, G.; Shen, S.; Wen, L.; Luo, S.; and Bo, L. 2020. Efficient pig counting in crowds with keypoints tracking and spatial-aware temporal response filtering. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 10052–10058. IEEE.

Feng, W.; Wang, K.; and Zhou, S. 2023. An efficient neural network for pig counting and localization by density map estimation. *IEEE Access*.

Hao, W.; Zhang, L.; Han, M.; Zhang, K.; Li, F.; Yang, G.; and Liu, Z. 2023. YOLOv5-SA-FC: A Novel Pig Detection and Counting Method Based on Shuffle Attention and Focal Complete Intersection over Union. *Animals*, 13(20).

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385.

Huang, Y.; Xiao, D.; Liu, J.; Tan, Z.; Liu, K.; and Chen, M. 2023. An Improved Pig Counting Algorithm Based on YOLOv5 and DeepSORT Model. *Sensors*, 23(14): 6309.

Lee, G.; Ogata, K.; Kawasue, K.; Sakamoto, S.; and Ieiri, S. 2022. Identifying-and-counting based monitoring scheme for pigs by integrating BLE tags and WBLCX antennas. *Computers and Electronics in Agriculture*, 198: 107070.

Li, C.; Hu, X.; Abousamra, S.; and Chen, C. 2023. Calibrating uncertainty for semi-supervised crowd counting. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 16685–16695. IEEE.

Liang, D.; Chen, X.; Xu, W.; Zhou, Y.; and Bai, X. 2022. Transcrowd: weakly-supervised crowd counting with transformers. *Science China Information Sciences*, 65(6): 160104.

Liang, D.; Xie, J.; Zou, Z.; Ye, X.; Xu, W.; and Bai, X. 2023. CrowdCLIP: Unsupervised Crowd Counting via Vision-Language Model. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2893–2903.

Lin, H.; Ma, Z.; Hong, X.; Wang, Y.; and Su, Z. 2022. Semi-supervised crowd counting via density agency. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1416–1426.

Lin, T.; and Maji, S. 2017. Improved Bilinear Pooling with CNNs. *CoRR*, abs/1707.06772.

Lin, T.; RoyChowdhury, A.; and Maji, S. 2015. Bilinear CNN Models for Fine-grained Visual Recognition. *CoRR*, abs/1504.07889.

Lin, W.; and Chan, A. B. 2023. Optimal transport minimization: Crowd localization on density maps for semi-supervised counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21663–21673.

Liu, L.; Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; and Han, J. 2019. On the Variance of the Adaptive Learning Rate and Beyond. *CoRR*, abs/1908.03265.

Liu, Y.; Ren, S.; Chai, L.; Wu, H.; Xu, D.; Qin, J.; and He, S. 2022a. Reducing spatial labeling redundancy for active semi-supervised crowd counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, Y.; Shao, Z.; and Hoffmann, N. 2021. Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions. *CoRR*, abs/2112.05561.

Liu, Y.; Wang, Z.; Shi, M.; Satoh, S.; Zhao, Q.; and Yang, H. 2020. Towards unsupervised crowd counting via regression-detection bi-knowledge transfer. In *Proceedings of the 28th ACM International Conference on Multimedia*, 129–137.

Liu, Z.; Mao, H.; Wu, C.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022b. A ConvNet for the 2020s. *CoRR*, abs/2201.03545.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Sam, D. B.; Agarwalla, A.; Joseph, J.; Sindagi, V. A.; Babu, R. V.; and Patel, V. M. 2020. Completely Self-Supervised Crowd Counting via Distribution Matching. *CoRR*, abs/2009.06420.

Tian, M.; Guo, H.; Chen, H.; Wang, Q.; Long, C.; and Ma, Y. 2019. Automated pig counting using deep learning. *Computers and Electronics in Agriculture*, 163: 104840.

Tian, Y.; Cheng, K. M.; Zhang, Z.; Zhang, T.; Li, S.; Yan, D.; and Xu, B. 2025. A Novel Modeling Framework and Data Product for Extended VIIRS-like Artificial Nighttime Light Image Reconstruction (1986-2024). *arXiv preprint arXiv:2508.00590*.

Walach, E.; and Wolf, L. 2016. Learning to count with cnn boosting. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 660–676. Springer.

Wang, S.; Jiang, H.; Qiao, Y.; Jiang, S.; Lin, H.; and Sun, Q. 2022. The Research Progress of Vision-Based Artificial Intelligence in Smart Pig Farming. *Sensors*, 22(17).

Wu, L.; and Wang, Y. 2017. Where to Focus: Deep Attention-based Spatially Recurrent Bilinear Networks for Fine-Grained Visual Recognition. *CoRR*, abs/1709.05769.

Xia, W.; Peng, R.; Chu, H.; Zhu, X.; Yang, Z.; Zhao, Y.; and Yang, L. 2025. An Overall Real-Time Mechanism for Classification and Quality Evaluation of Rice. arXiv:2502.13764.

Yu, T.; Zhang, Z.; Lyu, Z.; Gong, J.; Yi, H.; Wang, X.; Zhou, Y.; Yang, J.; Nie, P.; Huang, Y.; et al. 2025. BrowserAgent: Building Web Agents with Human-Inspired Web Browsing Actions. *arXiv preprint arXiv:2510.10666*.

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

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

Zhang, Z.; Xiao, Y.; Chen, Z.; Chen, X.; and Li, X. 2024. Gait recognition for farm workers via multi-scale temporal feature perception. *Computers and Electronics in Agriculture*, 226: 109353.