symmetry

MDPI

Article

Detection of Key Points in Mice at Different Scales via Convolutional Neural Network

Zhengyang Xu [1], Ruiqing Liu [2], Zhizhong Wang [1], Songwei Wang [1] and Juncai Zhu [1,*]

[1] School of Electrical Information, Zhengzhou University, Zhengzhou 450000, China; 202012182013351@gs.zzu.edu.cn (Z.X.); wzz1982@zzu.edu.cn (Z.W.); wangsongwei@zzu.edu.cn (S.W.)
[2] Department of Mechanical and Electrical Engineering, Henan Light Industry Vocational College, Zhengzhou 450000, China; 202022182013433@gs.zzu.edu.cn
* Correspondence: zhujuncai@gs.zzu.edu.cn

Abstract: In this work, we propose a symmetry approach and design a convolutional neural network for mouse pose estimation under scale variation. The backbone adopts the UNet structure, uses the residual network to extract features, and adds the ASPP module into the appropriate residual units to expand the perceptual field, and uses the deep and shallow feature fusion to fuse and process the features at multiple scales to capture the various spatial relationships related to body parts to improve the recognition accuracy of the model. Finally, a set of prediction results based on heat map and coordinate offset is generated. We used our own built mouse dataset and obtained state-of-the-art results on the dataset.

Keywords: mouse key point detection; scale change; deep learning; CNN

check for updates

Citation: Xu, Z.; Liu, R.; Wang, Z.; Wang, S.; Zhu, J. Detection of Key Points in Mice at Different Scales via Convolutional Neural Network. *Symmetry* **2022**, *14*, 1437. https://doi.org/10.3390/sym14071437

Academic Editor: Chin-Ling Chen

Received: 6 June 2022
Accepted: 6 July 2022
Published: 13 July 2022

1. Introduction

Quantifying behavior is particularly important for neuroscience applications [1]. Video observation and recording is a simple, easy, and effective approach to detect behavior [2–4]. An important aspect of studying animal behavior based on the skeleton is the accurate identification of key points of animal body parts. During video recording to study the behavioral characteristics of mice in the laboratory, we found that scale change greatly affects recognition accuracy [5,6]. Therefore, we improved the accuracy of mouse key point recognition under scale changes by improving the model of the current key point detection algorithm. The dataset is available at https://github.com/Martin-xu-ma/Dataset, accessed on 5 July 2022.

Here, we propose a symmetric method and design a key point detection model as shown in Figure 1. The model uses dilation convolution [7] instead of ordinary convolution for feature extraction, and uses deep and shallow feature fusion to obtain multi-scale information. Superior performance was achieved in scenarios targeting mouse scale variations.

The model body adopts a codec structure, and when the model is input, the image is subtracted from the mean, subtracting the average of the three channels of the image to speed up the training of the network. In the coding section, feature extraction and data dimensionality reduction are performed using dilation convolution, and the sensory field is enlarged to obtain more feature information. In the decoding section, transpose convolution is used to increase the resolution [8–10]. A jump connection method is used to fuse deep and shallow features. Unlike the common heat map-based forecasting methods, we designed a model that uses heatmaps plus coordinate offsets to predict key points, thereby improving the accuracy of key point coordinate prediction [11].

The rest of the organization of this paper is as follows: the second section introduces three key point detection algorithms to compare with the design algorithm of this paper, the third section introduces the two improved parts of the algorithm structure—the dilation

convolution and deep and shallow feature fusion, and the design of the loss function—the fourth section introduces the construction of the data set, the determination of the evaluation index and the ablation experiment verifies the outstanding performance of the proposed algorithm. Finally, we summarize our findings in Section 5.

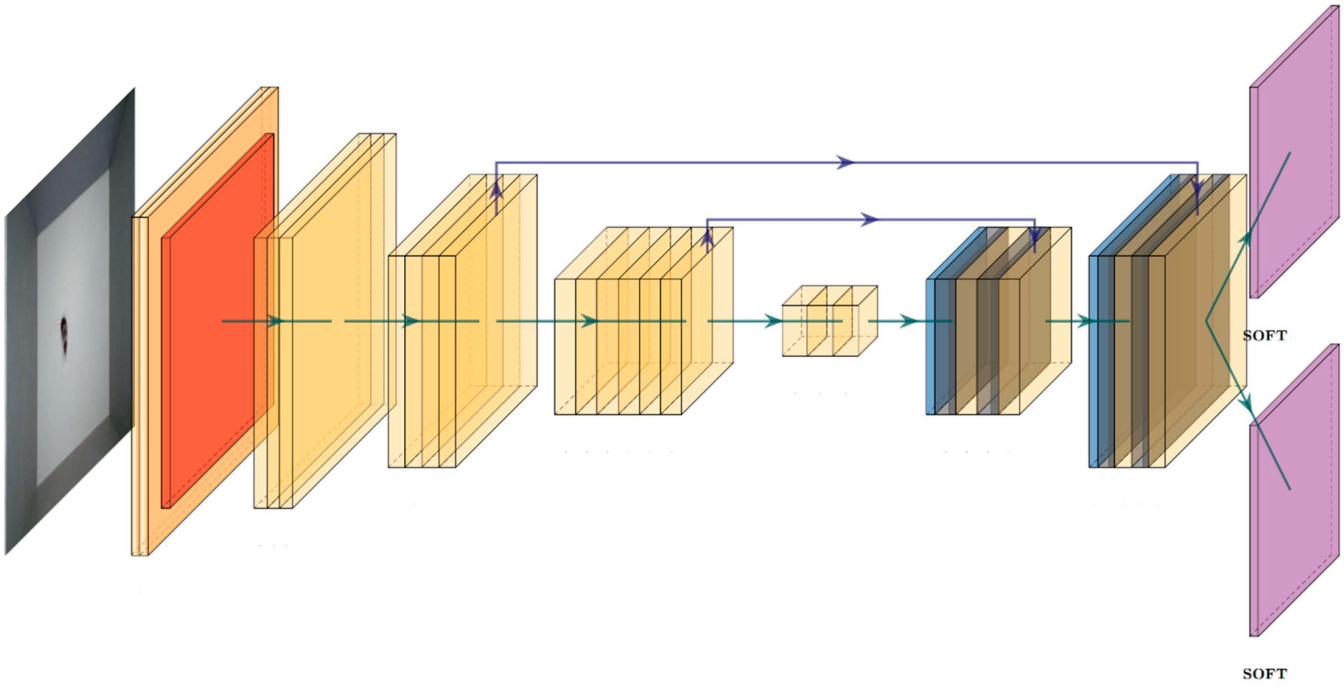

Figure 1. The network architecture of this paper. The network first uses a conventional convolution to connect a pooling layer for four-times down-sampling, followed by four blocks for two-times down-sampling, and then two transposed convolutions for two-times up-sampling, respectively, and finally completes the output of the prediction graph and offset graph.

2. Related Work

Before designing the model, we disassembled several typical single target key point detection algorithms in detail to obtain inspiration for better design. The key point detection algorithm is popular for key point detection of human bones. In the present study, key point detection is used for mouse recognition based on the recognition of its behavior. DeepPose [12,13] is the first application of deep learning to human pose estimation. The author defines human pose estimation as the topic of regression at key points and uses a convolutional neural network (CNN) for regression. The model includes Alexnet [14] and an additional output layer for regression node coordinates, and it is trained using L2 loss function. Further, the model uses cascaded regressors to refine the predictions. The images are sheared around the predicted junctions and fed into the next stage, so that subsequent posture regressors can see higher resolution images and learn their fine features.

CPM [15,16] uses a serialized convolution architecture to express spatial information and texture information. Its network structure is divided into multiple stages. The initial stage uses the original image as the input, while the latter stage combines the feature maps of the previous stage as the input, mainly integrating spatial information, texture information, and center constraints. For the problem where the network is difficult to train, each stage of CPM outputs a confidence map and calculates the corresponding loss to solve the problem of gradient disappearance through relay supervision.

The Stacked Hourglass network [17,18] was the top network in the MPII pose estimation competition in 2016. A single hourglass module is the smallest design to capture information at various scales, and it integrates information at different scales. By stacking a single hourglass module, the latter hourglass modules can further process the advanced features, thus obtaining the spatial structure relationship between the joint points. Likewise,

relay supervision is applied to each hourglass stage to eliminate the vanishing gradient problem caused by the very deep network.

From the perspective of the development trend of the key point detection algorithm for obtaining a large receptive field and high resolution, a large receptive field implies that the network can learn the space between the key point, while high resolution can help to improve prediction accuracy; moreover, the characteristics of the fusion between different scales at different levels to predict the key can also be very supportive [19]. On the basis of the above ideas, we designed the mouse key point detection model.

3. Network Architecture

3.1. Dilated Convolution

Our designed model adopts dilated convolution to expand the network receptive field. Dilated convolution was first proposed to solve the problem of increasing receptive field in image segmentation while retaining the original size of the featured image. Unlike normal convolution kernels, dilated convolution between convolution kernels places them into the "empty" area, as shown in Figure 2; in this way, in the case of no additional parameters being introduced, the receptive field of the network is expanded, providing the figure size because of the receptive field increasing and decreasing. At the same time, the dilated convolution retains the internal space of the data structure, which helps in the acquisition of long-distance information.

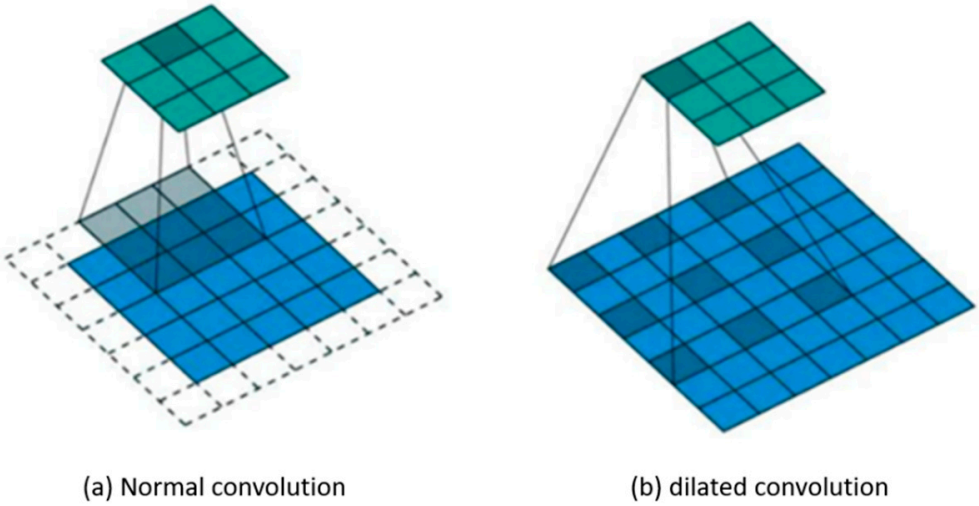

Figure 2. Realization of normal convolution and empty convolution.

Multiscale information can be obtained by setting the dilated convolution to different dilation rates. The Atrous Spatial Pyramid Pooling (ASPP) module [20–22] is designed based on this idea in DeepLabV3 to merge multiscale context information. This module is also adopted in our designed model. By using dilated convolution with different dilation rates, the receptive field is expanded, and multiscale information is captured to implicitly establish the spatial connection between mouse key points. Figure 3 shows the application of the ASPP module in our designed model.

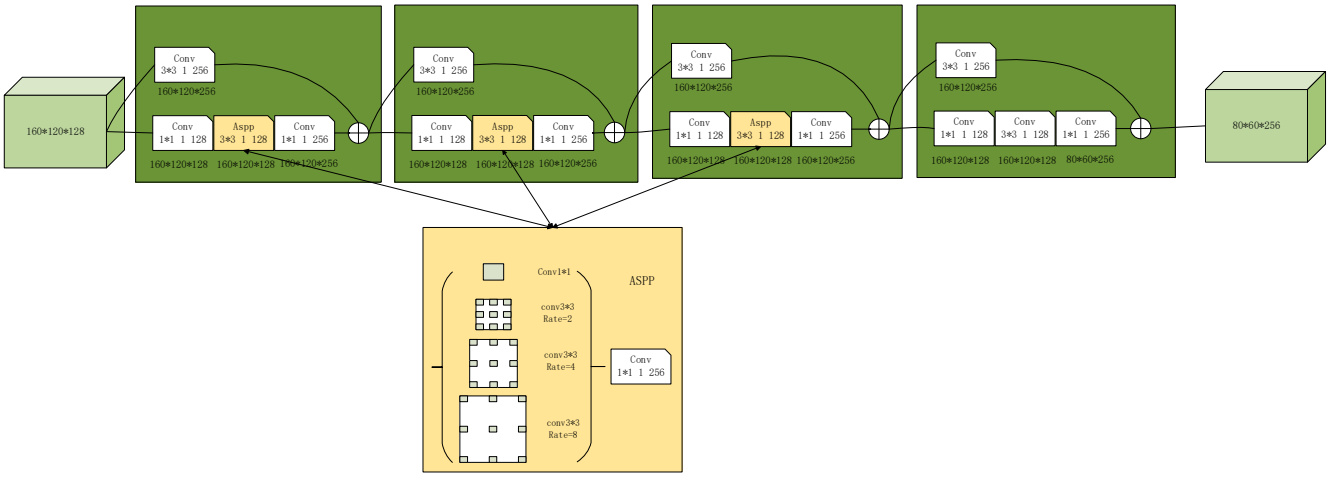

Figure 3. Application of the ASPP module in the designed model. Here, we take block2 as an example to show where empty convolution replaces ordinary convolution. We show the size of the feature graph in block2, the setting of the convolution kernel, and the position of the aspp module to replace the ordinary convolution.

In our designed model, the 3×3 convolution of each residual unit in the block3 module is replaced by the ASPP module. When the ASPP module is added at this position and the expansion rate is (1,2,4), multiscale information can be captured to the maximum extent. The ASPP module includes one 1×1 convolution, three 3×3 dilated convolutions with different expansion rates, and a global average pooling. The purpose of this arrangement is to obtain an image-level feature and better capturing of global information. The last 1×1 convolution is used to fuse multiscale features and output prediction results.

3.2. Encoding and Decoding Structure

The CNN extracts target features through layer-by-layer abstraction. Shallow features have a higher resolution and contain more location and detail information, while deep features have stronger semantic information due to pooling and other down-sampling operations [23–25]. While designing the key point detection model in mice, we considered the simultaneous use of both shallow and deep features in the network, so that we can fully obtain the location of mouse key points and semantic information.

The codec structure is a commonly used structure that combines deep and shallow features, and a typical representative of the codec structure is the UNet model. In the encoder–decoder structure, encoding is responsible for reducing the spatial dimension of feature maps to obtain deeper semantic information, while decoding is responsible for restoring image details and dimensions [26,27]. Skip connections are used to fuse deep and shallow features during encoding and decoding [28,29]. In the mouse key point detection task, it is necessary to integrate features of different levels, and a certain resolution is required for the output. However, to obtain deeper semantic information, the network needs a certain depth; hence, the structure of encoding first and then decoding is very suitable for constructing a key point detection network. The model with the encoder–decoder structure implemented here is shown in Figure 4.

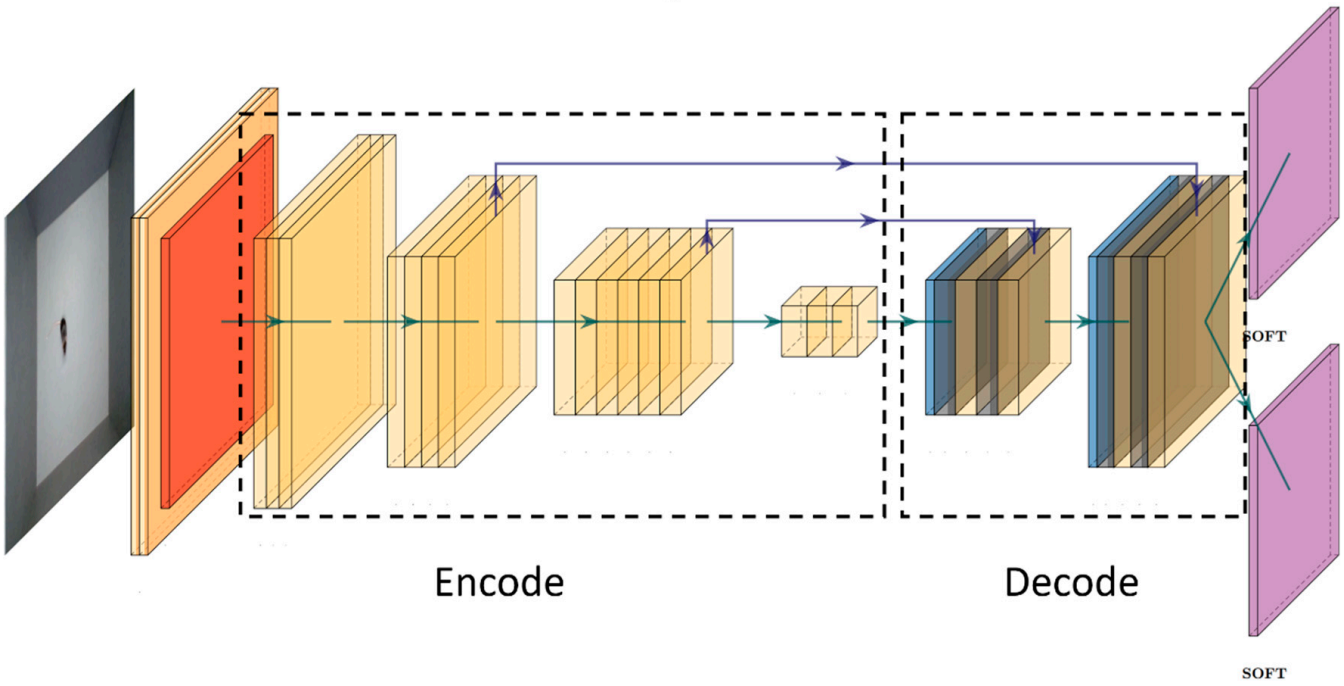

Figure 4. The codec structure of the designed model. The dotted box represents the part of the structure that is encoded and decoded in the model structure.

In our designed model, the residual network is used as the encoding network, and four residual modules are selected to extract features. Each residual module contains residual units of 3, 4, 6, and 3. The first residual block does not perform down-sampling, and the remaining three residual modules perform two-times down-sampling. After encoding, the designed model uses transposed convolution to decode and restore high-resolution images. To achieve a high-resolution output, the designed model uses two transposed convolutions, wherein each transposed convolution performs two-times up-sampling. Thus, the overall network performs 8 times down-sampling, and the output size of the confidence map is 80×60. Moreover, in the process of encoding and decoding, skip connections are used to combine shallow features and deep features, so that the network can simultaneously utilize deep semantic information and shallow spatial information.

3.3. Loss Function

The fused feature map outputs class probabilities and coordinate offsets through 1×1 convolution. The designed module uses the heatmap coordinate deviation method proposed by Google at IEEE conference on Computer Vision and Pattern Recognition (CVPR) 2017 to predict key points. Compared to simple heatmap-based prediction methods, coordinate shifting provides more accurate location predictions, but heatmaps suffer from theoretical errors due to down-sampling.

Finally, the losses of two parts need to be calculated when calculating the loss function. Sigmoid Cross Entropy loss is used for the category probability part, and Huber loss is used for the coordinate prediction part. Huber loss is less sensitive to outliers in the data than square error loss, which can solve the problem of singularities data biased model training in regression problems, and is more robust to outliers. The complete loss calculation formula is as follows:

$$\begin{aligned} loss_{sigmoid_{cross_{entropy}}} &= y_{cla} \times -\log\left(\frac{1}{1+e^{-x_{cla}}}\right) - \log\left(\frac{e^{-x_{cla}}}{1+e^{-x_{cla}}}\right) \times (1 - y_{cla}) \\ &= x_{cla} - x_{cla} \times y_{cla} + \log(1 + e^{-x_{cla}}) \end{aligned} \tag{1}$$

$$loss_{huber} = \begin{cases} \frac{1}{2}(y_{reg} - x_{reg})^2 & for|y_{reg} - x_{reg}| \leq \delta \\ \delta|y_{reg} - x_{reg}| - \frac{1}{2}\delta^2 & otherwise \end{cases} \quad (2)$$

where y_{cla} is the category label, y_{reg} is the real coordinate offset, x_{cla} is the predicted category probability, x_{reg} is the predicted coordinate offset, and δ is a hyperparameter used to judge whether it is a relatively singular data point. When the prediction deviation is less than δ, the mean square error (MSE) is adopted. When the prediction error is greater than δ, the linear error is used.

4. Experiments

4.1. Preparation of Datasets

For the mouse key point detection task under scale changes, a related dataset is produced here. By analyzing the action characteristics of mouse behavior and in combination with the need to measure mouse movement parameters and behavioral parameters in an open field experiment, we determined that the mouse nose tip, left ear, right ear, and tail root could be used as key points to prepare the dataset. To enable the designed algorithm to process images of different scales, we set the camera height to three levels at 60, 70, and 80 cm when shooting videos so as to collect mouse images of different scales. To prevent the convolutional neural network from overfitting a certain background color during training and to verify the robustness of the algorithm under different background colors, we used an experiment box with four background colors of white, light gray, dark gray, and black. An example dataset is shown in Figure 5.

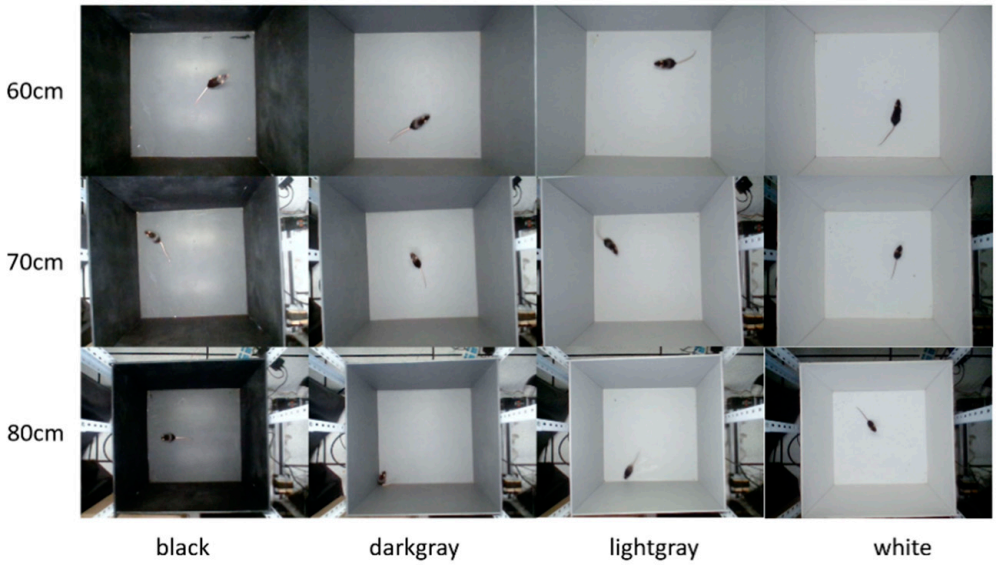

Figure 5. Images of mice at different scales.

In summary, the mouse key point detection dataset prepared here contains labels of three heights and four backgrounds. In the training set, only the 70 cm shooting height was selected, and 675 images of each background color were selected, resulting in a total of 2700 images. In the test set, 100 images of each combination were selected, resulting in a total of 1200 images. The quantity statistics of key points in the data set are shown in Table 1.

Table 1. Distribution of key points in the dataset.

Key Points	Training Sets	Test Sets			
		60 cm	70 cm	80 cm	Sum
Nose	2291	381	375	375	1131
Left ear	2698	400	400	400	1200
Right ear	2699	400	400	400	1200
Tailbase	2598	400	400	398	1198

4.2. Evaluation Index

In the present experiment, the evaluation of the key point detection effect in mice refers to the performance indices of the human bone key point detection algorithm. The performance indices of the prevalent human key point detection algorithms include Object Keypoint Similarity (OKS) for multiple people and Percentage of Correct Keypoint (*PCK*) for a single person. The algorithm used in the designed model performs key point detection on a single mouse; therefore, *PCK* index is adopted.

The *PCK* index calculates the proportion by which the normalized distance between the detected key points and their corresponding labels is smaller than the set threshold. The formula for calculating the *PCK* index is as follows:

$$PCK_i^k = \frac{\sum_p \delta\left(\frac{d_{pi}}{d_p^{def}} \le T_k\right)}{\sum_p 1} \tag{3}$$

$$PCK_{mean}^k = \frac{\sum_p \sum_i \delta\left(\frac{d_{pi}}{d_p^{def}} \le T_k\right)}{\sum_p \sum_i 1} \tag{4}$$

where i represents the key point of type i, k represents the Kth threshold T_k, p represents the Pth image, d_{pi} represents the Euclidean distance between the predicted value and the true value of the key point of type i in the ith image, and d_p^{def} represents the scale factor of the pth image. The calculation method used for this factor is different according to different datasets. PCK_i^k indicates the *PCK* indicator for key points of category i under threshold T_k, and PCK_{mean}^k indicates the average *PCK* indicator for all key points under threshold T_k.

Before using the *PCK* indicator, it is necessary to determine the scale factor and threshold value to be used. The position error of the same degree of angle error in images of different scales is different, and therefore, the scale factor is used to solve this problem. Figure 6 shows the histograms of the interaural distances used at different scales in the dataset. The interaural distances at different scales show obvious differences, and they are normally distributed at the same scale, which is relatively stable and meets the requirements of the scale factor. Finally, there is an issue regarding the non-visibility of mouse ears in the images; in this case, the median of the inter-ear distance at this camera height is used as a scale factor, i.e., 15.29 at 60 cm, 12.85 at 70 cm, and 10.81 at 80 cm.

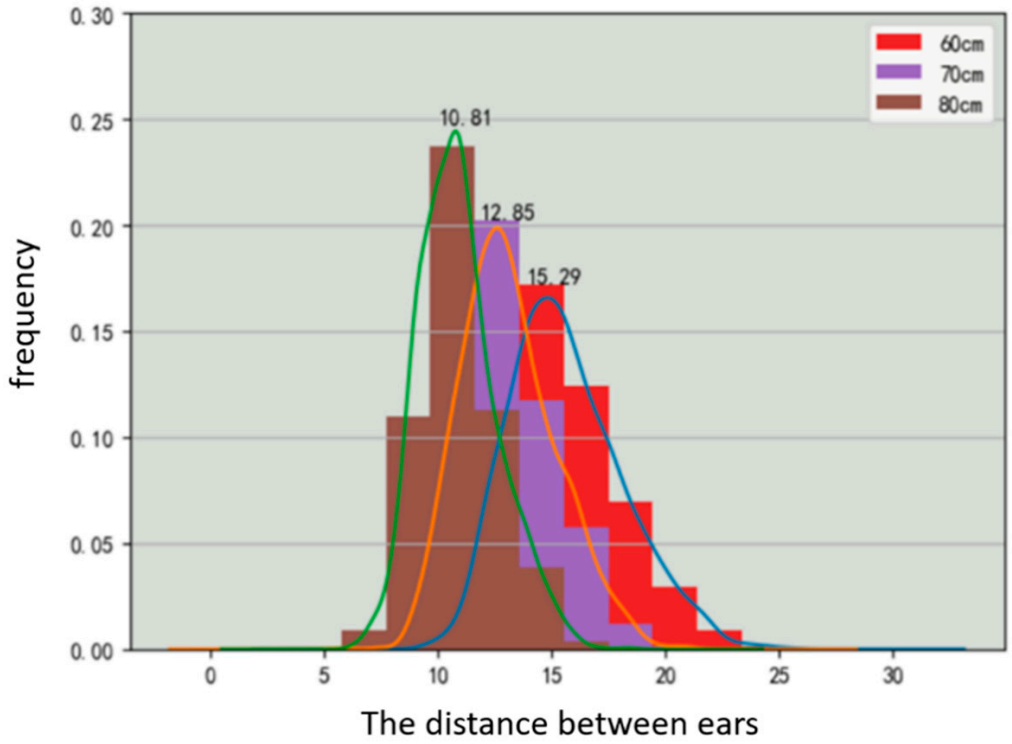

Figure 6. Histogram of the distance between the ears at different scales.

The size of the threshold reflects the tolerance of the error. The size of the threshold in the *PCK* indicator should change with the change in the scale factor. The histogram shown in Figure 6 reveals that the scale factor is concentrated in [10,15] pixels. The minimum unit is 1 pixel, and the corresponding distinguishable threshold is 0.1. When the threshold exceeds 0.5, the left ear or the right ear cannot be distinguished; therefore, the allowable threshold range is [0.0, 0.5]. In summary, the designed model adopts 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5 as thresholds.

4.3. Training Parameters

In order to ensure the objective consistency of the experiment, this experiment adopts a consistent training platform, and the experimental configuration is shown in Table 2:

Table 2. Experimental environment configuration table.

The Project Type	Description
CPU	Inter Core I9-8950HK
GPU	NVIDIA GeForce GTX 1080 8G
Memory	32GB
Operating environment	Windows 10
Development environment	Spyder

This algorithm directly uses the original map for training, and uses random scaling and rotating for each image before training for data augmentation, so each iteration only trains 1 image, that is, batch size = 1, and a total of 1,030,000 iterations are iterated. The learning rate for the first 10,000 iterations is 0.005, the learning rate for 10,000 to 430,000 iterations is 0.02, the learning rate for 430,000 to 730,000 iterations is 0.002, and the learning rate for 730,000 to 1,030,000 iterations is 0.001.

Runtime: on the published mouse dataset, the *PCK* index of our algorithm is increased by 29% higher than stackedhourglass, 17% higher than CPM and 10% higher than deeplabcut, but due to the addition of jump connections in the algorithm architecture to fuse deep

and shallow feature information, the amount of parameters increases, and the training time is deeper than deeplabcut increased by 1×.

4.4. Ablation Experiments

The main body of our network structure adopts the residual network structure; however, to maintain the recognition accuracy at different scales, we added the ASPP module to the residual module to expand the receptive field and learn from the Unet structure for deep and shallow feature fusion to achieve task requirements. In this section, we will show the performance improvement after adding modules to the network.

a. Experimental results and analyses under different dilated convolution parameters.

We first verify the impact of dilated convolution on the performance of the algorithm. For this purpose, the ASPP modules with different dilated convolution expansion rates and position parameters are integrated into the network for comparison. Specifically, the ASPP modules are integrated into block2 and block3 of the network to verify the effect of dilated convolution location on model performance. We also design three dilation rate combinations for each ASPP module, namely (1, 2, 4), (2, 4, 8), and (4, 8, 12), to verify the dilation rate pair of the dilated convolution impact on model performance.

Table 3 shows that when the ASPP module is located in block2, the performance gradually improves with the increase in the expansion rate. This may be because the receptive field is small at the bottom layer of the network, and the dilated convolution adopts sparse sampling. When the expansion rate increases, the feature information of a large receptive field can be obtained; thus, the performance of the algorithm is gradually improved. When the ASPP module is located in block3 and the inflation rate is (1, 2, 4), the algorithm performance is significantly improved; however, when the inflation rate is (2, 4, 8) and (4, 8, 12), the performance of the *PCK* indicator begins to decline. The reason may be that in block3, the receptive field has reached a large size and the feature map has become very small. The mouse key points are concentrated in the adjacent positions on the feature map, and the dilated convolution adopts sparse sampling; thus, when the expansion rate is too large, the spatial connection is broken.

Table 3. Performance comparison of the ASPP modules with different parameters and algorithms at different thresholds. B2 refers to the aspp module added in block2, r1 refers to the void rate (1, 2, 4), r2 refers to the void rate (2, 4, 8), r3 refers to the void rate (4, 8, 12).

	0	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Cpm	0	0.2404	0.5874	0.7327	0.7661	0.7739
Hourglass	0	0.2007	0.4506	0.5284	0.5409	0.5437
Deeplabcut	0	0.0944	0.3683	0.6598	0.8344	0.9111
B2r1	0	0.1113	0.4395	0.7064	0.8386	0.9021
B2r2	0	0.1364	0.4707	0.7501	0.8896	0.9444
B2r3	0	0.1842	0.5576	0.8088	0.9224	0.9571
B3r1	0	0.229	0.6204	0.8374	0.9302	0.9628
B3r2	0	0.1954	0.5428	0.7957	0.9101	0.955
B3r3	0	0.1903	0.5363	0.7821	0.9023	0.9494

To summarize, when the ASPP module is located in block3 and the expansion rate is (1, 2, 4), the network achieves the highest *PCK* index. Moreover, the *PCK* index at each scale is improved, which confirms that the ASPP module can effectively improve prediction accuracy for many scales.

We have determined the best parameters and positions to add the ASPP module. Compared with several other algorithms, our model has achieved corresponding improvements at different scales. Table 3 shows the specific *PCK* performance index when the threshold is 0.4.

b. Experimental results and analysis after deep and shallow feature fusion.

After adding the ASPP module to the designed model, the performance of the algorithm significantly improved; however, the performance improvement in terms of high-accuracy recognition was still not obvious. Therefore, we considered the fusion of deep and shallow features to obtain shallow position information and deep semantic information. We believed that the fusion could further improve the performance of the algorithm.

We first included the ASPP module in the network structure. According to the experimental results, when the ASPP module is added in block3 and the expansion rate is (1, 2, 4), the model performance is the best; however, when the fault tolerance threshold is small, there is no obvious improvement in model performance. A possible reason for this finding is that the deep feature receptive field is large, which improves the accuracy of key point recognition. However, because of low resolution and less detailed information, when the fault tolerance threshold is small, the recognition accuracy is not significantly improved. To solve this problem, we used deep and shallow feature fusion based on the ASPP module to further improve the performance of the designed model.

The results show that when the ASPP module is located in block3 and the expansion rate is (1, 2, 4), the effect of adding dark and light feature fusion becomes worse under 80 cm height and black background. The reason for this finding may be the use of the ASPP module with the best performance. The module parameters render the receptive field larger and improve recognition accuracy. However, when the deep and shallow features are fused, the details of the shallow layers are more obvious. On a black background, the background color is similar to the body color of the mouse, which may be learned in the network. To reduce the sharpness of irrelevant information, we adjusted the parameters of the ASPP module and conducted the experiments. The experimental results showed that when the ASPP module is added in block2 and the void ratio is (1, 2, 4), the best model performance can be obtained with deep and shallow feature fusion. Figure 7 shows the change of recognition accuracy before and after adjustment, and Figure 8 shows some recognition errors, showing the limitations of the algorithm.

c. Experimental results and analysis between different models.

As shown in Table 4, the detection accuracy of our designed algorithm for the four key points achieved the highest level; the average *PCK* was 9%, 6%, and 2% higher than those of CPM, Stacked Hourglass, and DeepLabCut, respectively. A comparison of detection speed and model size revealed the detection speed and model size of our designed model are better than those of CPM and Stacked Hourglass, which can meet the requirement of real-time detection, but slightly inferior to that of DeepLabCut. Table 5 shows the performance comparison of the four models for identifying different key points and at different scales. From the detection accuracy of different key points, the detection accuracy of the left ear and the right ear is significantly higher than that of the nose tip and the tail root. This is because the left and right ears of the mouse remain relatively stable and do not change greatly when the posture changes. However, the nose tip and tail root are often occluded and deformed, which are relatively difficult to detect. The issue of low accuracy of the nose tip and tail root will be investigated in our future work.

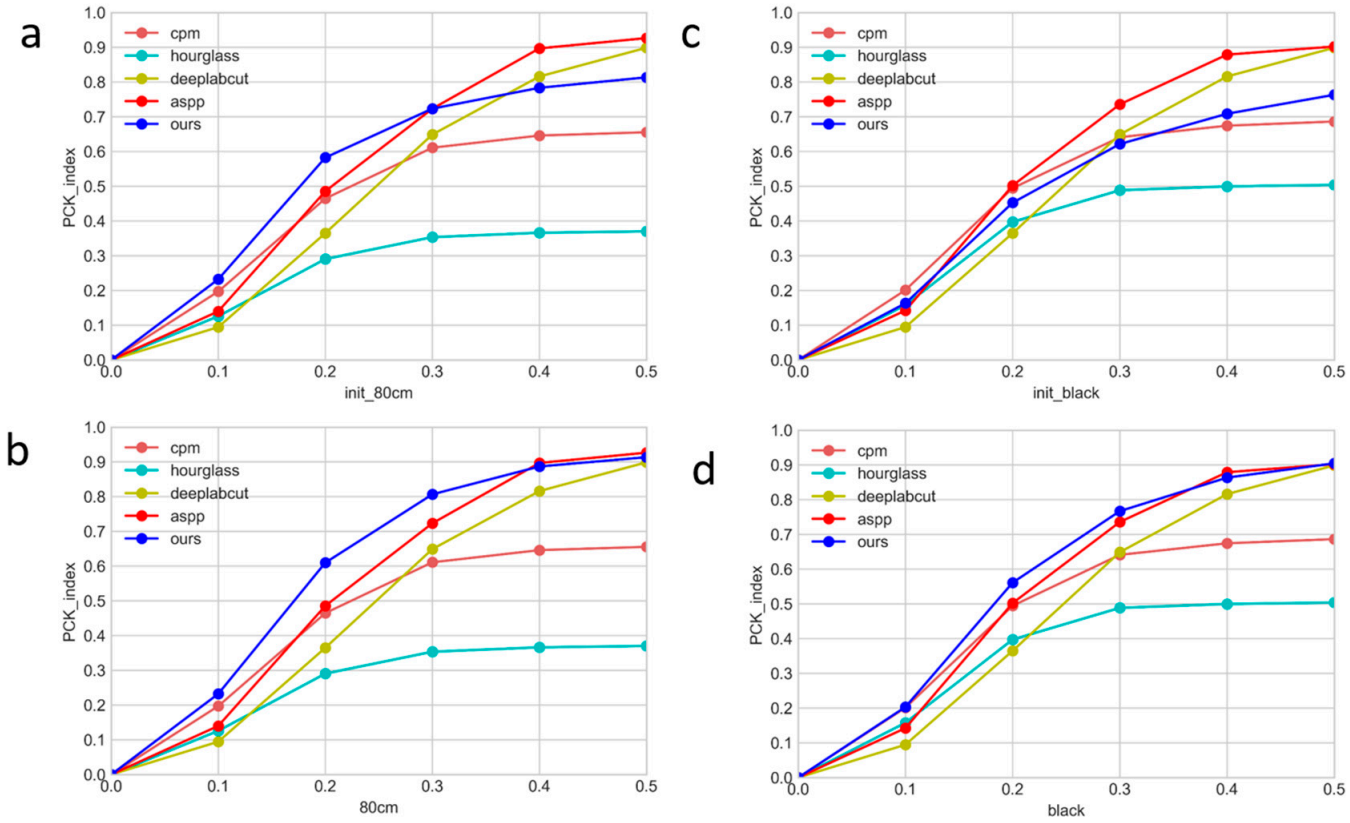

Figure 7. (**a**). The ASPP module is located in block3 and the expansion rate is (1, 2, 4); the performance is evaluated at the height of 80 cm after deep and shallow feature fusion. (**b**). The ASPP module is located in block3 and the expansion rate is (1, 2, 4); the background color is black after deep and shallow feature fusion. (**c**). The ASPP module is located in block2 and the expansion rate is (1, 2, 4); the performance is evaluated at the height of 80 cm after deep and shallow feature fusion. (**d**). The ASPP module is located in block2 and the expansion rate is (1, 2, 4); the background color is black after deep and shallow feature fusion.

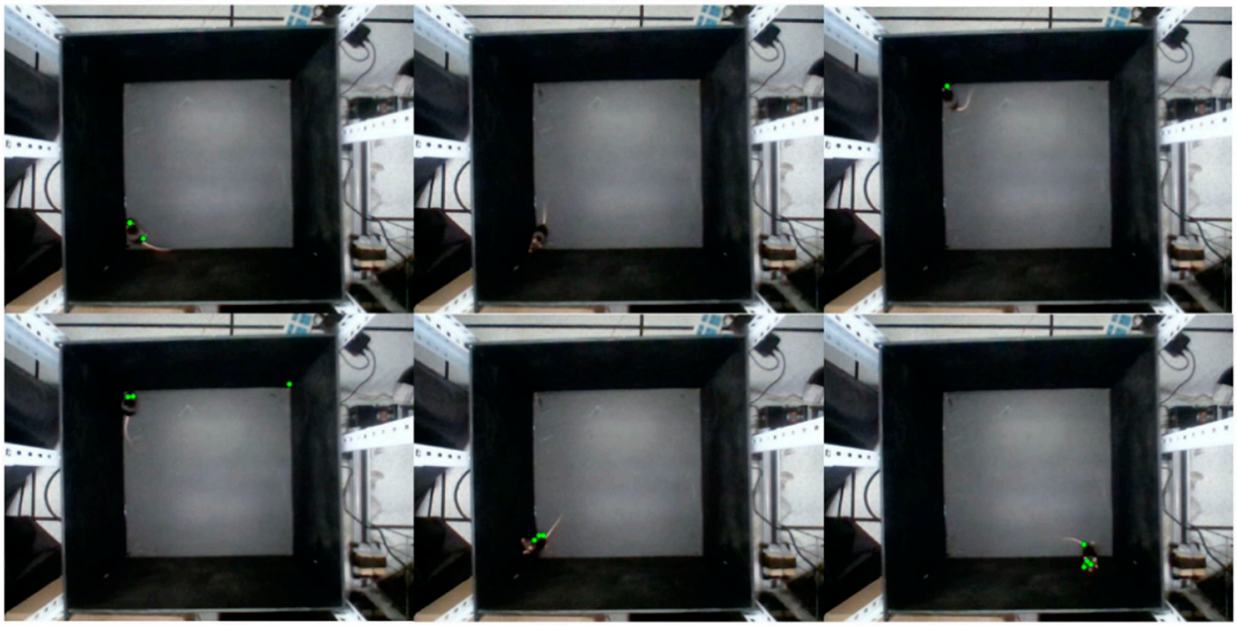

Figure 8. False detections and missed detections in the test results.

Table 4. Experimental results and analysis under different dilation ratios ($T_k = 0.4$).

Algorithm	*PCK*at Different Camera Heights		
	60 cm	**70 cm**	**80 cm**
Block3_(1,2,4)	95.32%	96.57%	87.16%
Cpm	84.63%	80.57%	64.59%
Hourglass	64.71%	60.89%	36.62%
deeplabcut	93.74%	91.37%	65.12%

Table 5. Comparison of four models ($T_k = 0.4$) T_k refer to threshold value.

Algorithm	*PCK*							
	Nose	**Leftear**	**Rightear**	**Tailbase**	**60 cm**	**70 cm**	**80 cm**	**Average**
CPM	75.77%	75.75%	81.42%	73.46%	84.63%	80.57%	64.59%	76.61%
Hourglass	48.54%	58.58%	64.83%	44.07%	64.71%	60.89%	36.62%	64.71%
DeeplabCut	86.25%	90.13%	89.96%	85.94%	93.74%	91.37%	65.12%	91.66%
Ours	92.74%	96.25%	96.00%	90.23%	96.71%	96.06%	88.67%	93.82%

We then verified the *PCK* indicators of each algorithm under different thresholds. As shown in Figure 9, different thresholds represent different tolerances for errors. The smaller the threshold, the more refined are the requirements for the prediction results. Figure 9 shows that our designed algorithm achieved the highest accuracy under different thresholds, indicating that the prediction results of the designed algorithm have a high accuracy.

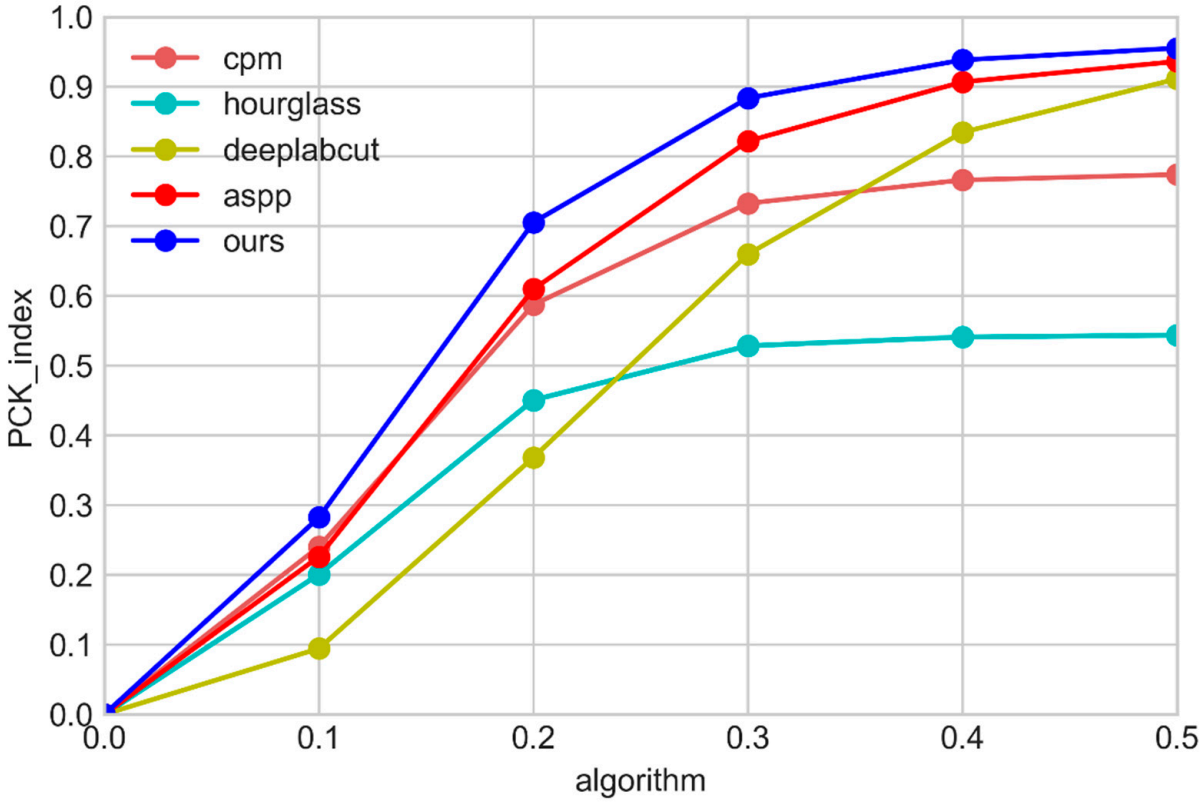

Figure 9. *PCK* index of each algorithm under different thresholds.

4.5. Discussion Section

According to the Figure 9, it can be intuitively seen that the overall performance of the algorithm in this paper has achieved superior performance at various thresholds, because it is easier to obtain global information after adding the ASPP module to feel the field increase, and after adding the jump connection, the semantic information and location information are integrated. The algorithm mainly focuses on the effect of solving the scale change on the recognition accuracy of mice, so some other factors that may affect the recognition accuracy are discussed. In order to cope with the impact of complex background on the performance of the algorithm, we used four different colors of background to verify the performance of the algorithm, and the experimental results showed that the recognition accuracy on the black background was low, and the analysis was due to the confusion caused by the black background being similar to the color of the mice themselves. Therefore, when the color contrast between the background and the experimental object is obvious, the recognition accuracy of the scale change is higher.

5. Conclusions

By combining with the existing key point detection algorithms, we designed a mouse key point detection model with dilated convolution and deep and shallow feature fusion. A mouse key point detection dataset was established, and by using this dataset, the best performance of the model was obtained by adjusting the dilated convolution parameters and fusion of deep and shallow features. The average *PCK* index of our designed model for mouse key point detection reached 93.82% under the threshold of 0.4, which is higher than the average *PCK* index of CPM (76.61%), Stacked Hourglass (64.71%), and DeepLabCut (83.44%). Our designed model achieved excellent performance in mouse key point detection for scale changes and laid a good foundation for our future studies on mouse behavior detection based on key point detection.

Author Contributions: Conceptualization, Z.X.; Data curation, R.L.; Formal analysis, S.W.; Funding acquisition, Z.W.; Investigation, S.W.; Methodology, Z.X.; Project administration, Z.W.; Software, J.Z.; Validation, R.L.; Visualization, J.Z.; Writing—original draft, Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation for Young Scholars of China, grant number 61803344.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The murine keypoint detection dataset used to support the findings of this study have been deposited in the Github repository (https://github.com/Martin-xu-ma/Dataset, accessed on 5 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krakauer, J.W.; Ghazanfar, A.A.; Gomez-Marin, A.; Maciver, M.A.; Poeppel, D. Neuroscience Needs Behavior: Correcting a Reductionist Bias. *Neuron* **2017**, *93*, 480–490. [CrossRef]
2. Sridhar, V.H.; Roche, D.G.; Gingins, S. Tracktor: Image-based automated tracking of animal movement and behaviour. *Methods Ecol. Evol.* **2018**, *10*, 815–820. [CrossRef]
3. Anderson, D.J.; Perona, P. Toward a Science of Computational Ethology. *Neuron* **2014**, *84*, 18–31. [CrossRef]
4. Mathis, A.; Mamidanna, P.; Cury, K.M.; Abe, T.; Murthy, V.N.; Mathis, M.W.; Bethge, M. DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **2018**, *21*, 1281–1289. [CrossRef]
5. Zhao, R.; Li, Q.; Wu, J.; You, J. A nested U-shape network with multi-scale upsample attention for robust retinal vascular segmentation. *Pattern Recognit.* **2021**, *120*, 107998. [CrossRef]
6. Xia, C.; Peng, J.; Ma, Z.; Li, X. A Multi-Scale Network with the Encoder-Decoder Structure for CMR Segmentation. *J. Inf. Hiding Priv. Prot.* **2019**, *1*, 9. [CrossRef]
7. Zhu, G.; Zeng, X.; Jin, X.; Zhang, J. Metro passengers counting and density estimation via dilated-transposed fully convolutional neural network. *Knowl. Inf. Syst.* **2021**, *63*, 1557–1575. [CrossRef]

8. Zhang, S.; Jiang, D.; Yu, C. A mixed depthwise separation residual network for image feature extraction. *Wirel. Netw.* **2021**, 1–12. [CrossRef]
9. Zhang, Y.; Kers, J.; Cassol, C.A.; Roelofs, J.J.; Idrees, N.; Farber, A.; Haroon, S.; Daly, K.P.; Ganguli, S.; Chitalia, V.C. U-Net-and-a-half: Convolutional network for biomedical image segmentation using multiple expert-driven annotations. *arXiv* **2021**, arXiv:2108.04658.
10. Iglovikov, V.; Shvets, A. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. *arXiv* **2018**, arXiv:1801.05746.
11. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. *arXiv* **2021**, arXiv:2105.05537.
12. Ferrari, V.; Marin-Jimenez, M.; Zisserman, A. Progressive search space reduction for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
13. Toshev, A.; Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1653–1660.
14. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25 NIPS, Harrah's and Harveys, Lake Tahoe, CA, USA, 3–8 December 2012.
15. Yang, Y.; Ren, Z.; Li, H.; Zhou, C.; Wang, X.; Hua, G. Learning Dynamics via Graph Neural Networks for Human Pose Estimation and Tracking. *arXiv* **2021**, arXiv:2106.03772.
16. Wei, S.E.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional Pose Machines. *arXiv* **2016**, arXiv:1602.00134.
17. Hua, G.; Li, L.; Liu, S. Multipath affinage stacked—hourglass networks for human pose estimation. *Front. Comput. Sci.* **2020**, *14*, 144701. [CrossRef]
18. Newell, A.; Yang, K.; Jia, D. Stacked Hourglass Networks for Human Pose Estimation. *arXiv* **2016**, arXiv:1603.06937.
19. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *Int. J. Comput. Vis.* **2020**, *128*, 642–656. [CrossRef]
20. Li, M.; Wang, Y.; Wang, C. Recursive residual atrous spatial pyramid pooling network for single image deraining. *Signal Process. Image Commun.* **2021**, *99*, 116430. [CrossRef]
21. Li, Y.; Li, K.; Chen, C.; Zhou, X.; Li, K. Modeling Temporal Patterns with Dilated Convolutions for Time-Series Forecasting. *ACM Trans. Knowl. Discov. Data* **2021**, *16*, 1–22. [CrossRef]
22. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv* **2018**, arXiv:1802.02611.
23. Wang, W.; Ning, Y. Multi-Scale Context Enhanced Network for Monocular Depth Estimation. *J. Phys. Conf. Ser.* **2021**, *1848*, 012023. [CrossRef]
24. Kwon, H.J.; Koo, H.I.; Soh, J.W.; Cho, N.I. Inverse-Based Approach to Explaining and Visualizing Convolutional Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–12. [CrossRef] [PubMed]
25. China Association for Science and Technology. *2016–2017 Development Report of Computer Science and Technology discipline: Development Report of Computer Science and Technology Discipline*; Science and Technology of China Press: Beijing, China, 2017; pp. 325–343.
26. Zhang, Z.; Fang, W.; Du, L.; Qiao, Y.; Zhang, D.; Ding, G. Semantic Segmentation of Remote Sensing Image Based on Encoder-Decoder Convolutional Neural Network. *Acta Opt. Sin.* **2020**, *40*, 0310001. [CrossRef]
27. Yin, Y. A Remote Sensing Image Road Extraction Method Based on Improved Encoder-Decoder Network. *Adv. Appl. Math.* **2021**, *10*, 274–281. [CrossRef]
28. Ying, W.; Li, J.; Wu, Y.; Zheng, K.; Li, J. U-Net with Dense Encoder, Residual Decoder and Depth-wise Skip Connections. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020.
29. Gong, X.; Li, Z. An Image Denoising Method Using Deep Asymmetrical Skip Connection. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/J. Comput. -Aided Des. Comput. Graph.* **2019**, *31*, 295–302.