Morphological Landmark Detection on Lobsters using Attention Networks

Parmeet Singh
Dalhousie University
pr819318@dal.ca

Mae Seto
Dalhousie University
mz715250@cs.dal.ca

Abstract

The aim of this paper is map landmarks on lobsters using convolution neural networks. An attention mechanism for improving the performance of the VGG network to that effect is used. Regular CNN architectures do not consciously extract detailed features from images. The attention mechanism learns to focus on regions around the lobster landmarks amongst the whole image. Existing approaches use a cascade of regression models for landmark prediction. Proposed, is a single iteration model augmenting an attention mechanism to produce similar results. The adaptability of the attention mechanism over any network, such as VGG16 or Resnet, avoids the need to learn the network from scratch.

1 Introduction

1.1 Motivation

Biologists use morphometry to perform quantitive analysis on the size and shape of organisms. Traditional morphometry consists of physical measurements of an organism like length, width, weight and area. Landmark-based morphometry uses a set of anatomical positions to analyse the shape of a population. These anatomical positions are biologically meaningful points that are consistent across similar species of a population. Morphometry is thus used to distinguish between species of similar shape.[13] Truss network measurements[14] are a set of distance measurements between landmarks. The truss network system uses geometric morphometry for stock identification. Stock identification can be used to cluster a population into groups with different growth rates. There are ecological and economic sustainability reasons to assess and monitor stocks and their sizes.

This paper reports on approaches to learn mapped landmarks on lobsters from training images and to ultimately extract these landmarks from test imagery. The approaches considered different deep learning architectures that include vanilla convolution neural networks (CNN), cascade of CNN and CNN augmented with attention mechanisms.

A regular neural network consists of a series of hidden layers. The input to a network is a single vector which is sequentially modified by the hidden layers. Each hidden layer consists of neurons that provide an output value from applying a function to the input values from the receptive field in the prior layer. This function is in the form of a vector of weights and a bias. The hidden layers are also fully-connected layers because each neuron in the one layer is connected to every other neuron in the next layer. Learning is achieved through incremental changes to these weights and bias via back propagation. Convolutional neural networks(CNNs) are a category of neural networks that modify input volumes with layers like convolution layers, pooling layers and fully connected layers. Convolution layers contain a set of learnable filters that slide across the width and height of the input volume during the forward pass. Pooling layers downsample the spatial dimension of the input volume through averaging or max pooling.

Attention mechanisms in deep learning are inspired by the ability of humans to focus their brain onto a subset of the environment [20]. The visual attention mechanism in humans is able to selectively focus onto a target of an image in high resolution while ignoring the rest of the image. Similarly, an algorithm that maps landmarks on lobsters should focus more on the lobster body compared to the background image. A CNN augmented with an attention mechanism could potentially map lobster landmarks more accurately by learning to focus in higher resolution on the lobster's body.

2 Related Work

Current research in deep learning using facial landmark detection algorithms can be applied to feature recognition on marine animals. [21]. Facial landmark detection techniques map the coordinates of key facial features like eyes, nose and mouth from images and videos of faces. Early work towards this includes Active Appearance Models (AAM) [1] that create a statistical model using principal component analysis (PCA). PCA learns the orthogonal bases that capture the variations of the face. Then, the AAM finds the coefficients of the statistical model that best fits the test image.

Dollar et al.[5] calculates object (landmark) pose in images with a cascade of random fern regressors. Each regressor takes as input the image patch around landmarks predicted by the previous regression model. Xiong et al. [6] uses scale-invariant fea-

ture transform[12] features of image patches as input to multiple cascade regressors to calculate facial landmarks at each stage. Sun et al. [7] propose a three-level cascaded regression method with a CNN classifier at each level. The first CNN provides a robust initial estimation of the facial landmarks. The following two CNN refine the initial prediction towards higher accuracies. The work of Zhang et al. [9] use a cascade of coarse to fine autoencoders. The first autoencoder calculates preliminary landmarks from a low resolution version of the face. Then, subsequent following autoencoders refine the landmarks by taking the local features extracted around the current landmarks. However, cascaded regression networks have shortcomings. For example, the learning process is independant of other stages. To address that, Trigeorgis et al.[10] proposed mnemonic descent which uses long short-term memory networks to model the dependancies between iterations in the cascade learning.

Yue et al. [3] propose a fully end-to-end convolutional regression network, as shown in Fig: 1, that yields facial landmarks in a single iteration. They introduce an attention mechanism where the network learns to pay "attention" to regions around landmarks without using image patches. The network generates spatial attention maps from the outputs of its multiple convolutional layers. The network explores image features at different scales with an attention mechanism that uses intermediate supervision to learn features that are relevant to facial landmarks. The total network loss is the sum of the regression loss at each level of the network. However, this network needs to be trained from scratch. This attention mechanism cannot augment pre-trained architectures like VGG or Resnet.

2.1 Contribution

The contributions of this paper are as follows: (1) to consider, and prove viable, the use of deep learning towards automating the mapping of landmarks on lobster images and (2) design and implement a convolution network that uses attention mechanisms for geometric morphometry of lobsters.

The rest of this paper is organized as follows. The proposed methodology is described in detail with subsections on the training data, model architectures, training and then performance evaluation with the proposed attention mechanism.

3 Proposed Methodology

First, a modified VGG16 network to predict landmarks on the lobster was used. Secondly, a cascade of CNNs where every CNN refines the prediction of the previous one was used. Thirdly, the modified VGG16 network with an attention mechanism (initially proposed by Rodriguez et al.[2] for fine-grained classification) was augmented. Lastly, a lobster shape embedding from the landmark data was learned. The modified VGG16 network with the attention mechanism on the embedded data, instead of direct regression on the landmarks, was used.

3.1 Training Data

The training data was 350 images of lobsters. Figure 3 shows a representative image from which 11 landmarks were mapped. Two landmarks were mapped on the claws, one on the eyes, one at the end of the lobster carapace, six between the segments of the lobster abdomen and two on the tail of the lobster. Bounding boxes were also marked around each of the lobster images.

3.2 Image Augmentation

This culled dataset was augmented using the *imgaug* [11] software image augmentation library. The following image augmentations were performed: randomly rotate horizontally and vertically; affine transformations like image translation from -10% to 10%; rotations from -45° to +45 °; images sharpened with pixel intensity multiplicative ratios from 0.75 to 1.5; image brightness changed for each RGB channel by adding pixel intensity from -10 to 10 and contrast normalization ratios ranging from 0.9 to 1.10. As part of the data preparation for the training, the images were re-sized to 448×448 pixels for input into the CNN.

3.3 Model Architectures

VGG16, proposed by Simonyan and Zisserman [4], was trained by the Visual Geometry Group. Their network uses 3×3 convolutional layers stacked on top of each other. The first step is a convolution of the image. Then, the image size is reduced through down sampling (max pooling). This alternates until the two layers become fully connected. The '16' in VGG16 refer to the number of convolutional layers in the networks.

3.3.1 Vanilla CNN

For feature extraction, a pre-trained network VGG16 CNN was used. The weights used for the VGG16 are those from the ImageNet dataset. The deeper layers of the pre-trained networks were made trainable to fine tune the accuracy.

As shown in Fig 2, the three fully-connected layers at the end of the network were removed. Next, a convolution layer with 1024 3×3 filters followed by a 4×4 max pooling layer was added. This was followed by another convolution layer with 1024 3×3 filters followed by a 4×4 max pooling layer. The final layer is a 1×1 convolution layer. The number of output channels for the 1×1 convolution layer is twice the number of

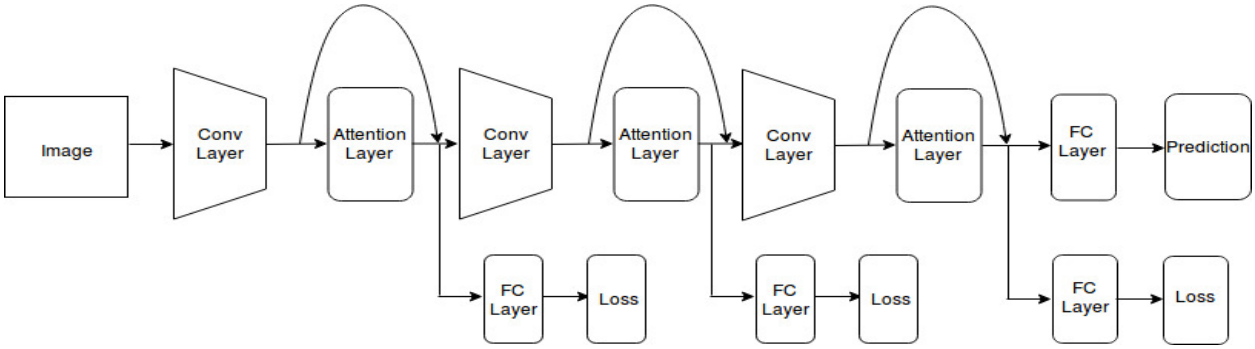

Figure 1. Alignment Attention Mechanism as mentioned in Yue et al.[3].

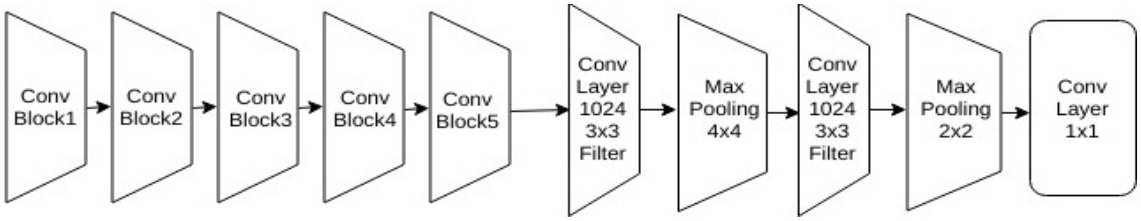

Figure 2. Modified VGG16 architecture. VGG16 contains five convolution blocks. Each convolution block contains two convolutional layers and one pooling layer. The fully-connected layers at the end were replaced with two convolutional layers containing 1024 3×3 filters and two max pooling layers.

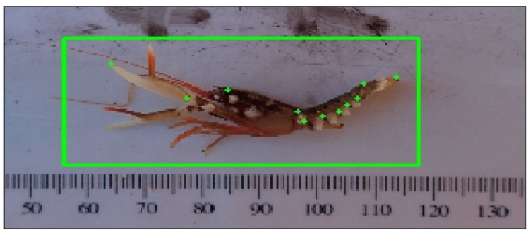

Figure 3. A representative training image that shows 11 landmarks on the lobster body. The manually drawn green bounding box shows the lobster extracted from the background. This was needed for the training of the cascaded CNN.

landmarks on the lobster (corresponding to an x and y coordinate) for each landmark.

3.3.2 Coarse to Fine CNN Cascade

Inspired by [7] and [15], a cascade of CNN regressors that progressively refines the output at each stage was created. First, an axis-aligned bounding box is detected around the target. A modified VGG16 network, as shown in Fig 2, initalized with ImageNet weights[16] was used. The final layer is a 1×1 convolution layer containing output channels equal to the four corners of an axis-aligned bounding box around the target as defined by the lower left and upper right coordinates. The input image is then cropped using the predicted bounding box coordinates of the network.

Secondly, another modified VGG16 network, as shown in Fig 2, was used to detect an initial eastimate of the landmark points on the cropped image. The final layer is a 1×1 convolution layer containing output channels equal to the twice the number of landmarks on the lobster(x and y for each landmark). The cropped images were resized to 448 $\times$ 448 keeping the aspect ratio the same i.e. the borders were padded to preserve the aspect ratio.

In the final step, separate modified VGG16 networks, as in Fig 2, were trained for each landmark point. The final layer of each network is a 1×1 convolution layer with two channels(x and y coordinate for each landmark). The input to each of these networks is a 90 $\times$ 90 patch around the predicted landmark from the croppped image. All patches are resized to 224 $\times$ 224 again to preserve the aspect ratio. These networks are trained to refine the initial predictions with the input to the networks being small regions around the landmarks.

3.3.3 Attention-Based Landmark Detection

In this approach, an attention mechanism proposed by Rodriguez et al.[2] for fine-grained classification was applied. However, the same mechanism for landmark regression was also used. This attention mechanism is independant since it can adapt to pre-trained architectures like VGG[4] or ResNet. The approach con-

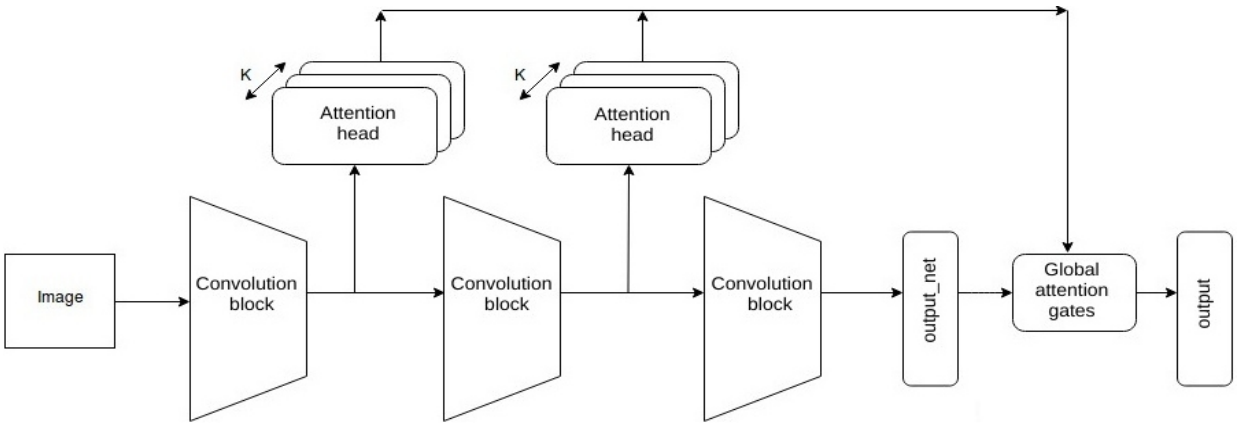

Figure 4. Attention Mechanism as shown in Rodrigues et al [2]

sists of an attention module that can be added after each convolutional layer without changing the underlying information pathways of the architecture. This is helpful since it augments architectures like VGG and ResNet with no additional supervision unlike the attention mechanism proposed in [3].The attention mechanism can be plugged into any existing trained network to quickly perform transfer learning. This is especially helpful if the training dataset is small.

As shown in Figure 4, the original CNN can be augmented with attention modules at arbitrary depths. Each attention module contains K attention heads that tap the feature activations at an arbitrary depth. The K attention heads make a prediction based on these feature activations. The original network $output_{net}$ is then corrected based on the output from the attention modules by means of the global attention gates to yield the final output.

The following equations describe the attention mechanism by Rodriguez et al. [2].

The features activations from layer l Z_l of dimension $\mathbb{R}^{c\times h\times w}$ are fed to an attention module containing K attention heads. Here, c is the number of channels of the feature activations and h, w are the spatial dimensions of the feature activation. ${W_H}^l$ in Eq 1 is the convolution kernel with K filters. The softmax function changes an n-dimensional vector z of arbitrary real values to an n-dimensional vector $\sigma(z)$ where $\sum_{i=1}^{K}\sigma_i = 1$.

Z_l is convolved with ${W_H}^l$ and then followed by a spatial softmax. Spatial softmax is a channel-wise softmax operation performed to normalize attention scores across the K attention heads. H_l^k represents the attention score given by the kth attention head to the feature activations at layer l.

$$H_l = spatial\_softmax({W_H}^l \bullet Z_l) \tag{1}$$

Here, $\bullet$ refers to the inner product operator. $W_{O_k{}^l}$ in Eq 2 is a convolution kernel for an attention head K which is applied on the feature activation Z^l. $W_{O_k{}^l}$ has filters equal to the number of labels i.e the number of classes it needs to classify. Rodriguez et al.[2] uses $W_{O_k{}^l}$ to calculate class probability scores for the classification problem. However, since we have a regression problem at hand, $W_{O_k{}^l}$ in our model has number of filters equal to twice the number of landmarks(x and y coordinate of each landmark). The output dimension of ${O_k}^l$ after convoluting Z^l by $W_{O_k{}^l}$ is $\mathbb{R}^{2\times landmarks\times h\times w}$.

The convolution operation by $W_{O_k{}^l}$ can be performed for all attention heads K in a single pass by setting the number of output filters to be $K \times 2 \times number\ of\ landmarks$.

${O_k}^l$ represents the output vector from the feature activations extracted at layer l and the kth attention head.

$${O_k}^l = W_{O_k{}^l} * Z^l \tag{2}$$

The o_k^l Eq. 3 is obtained by an element-wise product between H_k^l and O_k^l. H_k^l has dimension $h \times w$ and O_k^l has dimension $\mathbb{R}^{2\times landmarks\times h\times w}$. Therefore, H_k^l is repeated $2 \times number\ of\ landmarks$. The output o_k^l from Eq. 3 is the predicted landmarks from the kth attention head weighted by the attention score H_k^l and spatially averaged over the x, y.

$$o_k^l = \sum H_k^l \circledast O_k^l \tag{3}$$

Here, $\circledast$ is the element-wise multiplication operator.

The output of the lth attention module o^l Eq. 4 is the sum of outputs from the individual attention heads o_k^l Eq. 3 weighted by $g_{H_k^l}$.

$$o^l = \sum_k g_{H_k^l} o_k^l \tag{4}$$

g_H in Eq. 5 is obtained by first convolving Z^l with W_g^l. The resulting dimension after convolution with W_g^l becomes $\mathbb{R}^{|H|\times h\times w}$ where $|H|$ is the number of attention

modules. The resulting output is multiplied elementwise with H_l followed by a softmax operation.

$g_{H_k^l}$ represents the weight given to the output of each attention head K i.e o_k^l.The weight $g_{H_k^l}$ is a function of the feature activation Z^l and the attention scores H_l.

$$g_H^l = softmax(tanh(\sum_{x,y}(W_g^l \star Z^l) \circledast H_l)) \quad (5)$$

The output o^l from each attention module is weighted using candidate values c_l Eq. 6.c_l are a function of the feature activation Z^l.

$$c_l = tanh(W_G Z^l) \quad (6)$$

where W_G is the weight given to feature activation Z^l.

The global attention gates g Eq.7 are obtained by normalizing the set of candidate scores for all attention modules by means of a softmax function. g_{o^l} represents the weight given to the output predictions from the attention module l. i.e o^l

$$g_{o^l} = \frac{e^{c^l}}{\sum_{i=1}^{|G|} e^{c^i}} \quad (7)$$

The $final\_output$ Eq.8 of the network is the weighted sum of original output $output_{net}$ and output from the L attention modules o^l.

$$final\_output = g_{net} * output_{net} + \sum_l g_l \cdot o^l \quad (8)$$

The modified VGG16, as shown in Fig:2, is augmented with the above-mentioned attention mechanism with two attention modules. The feature activations from ConvBlock3 and ConvBlock4 (Fig:2) are fed into two separate attention modules with attention width $K = 10$. The last layer of the modified VGG16 network is a 1×1 convolution layer with the number of output filters equal to twice the number of landmarks(x and y coordinate) which is 22 in our case(since we have regressed for 11 landmarks). The VGG16 network prediction is combined with the predictions from the two attention modules with attention gates g. (Eq. 7)

3.4 Wing Loss

The mean square error function (also called the L2 loss) penalizes large errors more than the smaller ones (square of larger numbers is greater). The mean absolute error function (also called the L1 loss) is more sensitive to outliers since it does not square the errors but takes their absolute value. Feng et al.[19] propose a new loss function called wing loss as defined in Eq. 9 which magnifies errors in the range of $(-\omega, \omega)$ compared to mean squared error and mean absolute error. The value of ω can be adjusted to define the range where loss needs to be amplified. The wing loss function switches from L1 loss to logarithmic loss when the error is in the range of $(-\omega, \omega)$ and hence puts more focus on errors in the range of $(-\omega, \omega)$.

$$wing(x) = \begin{cases} \omega \ln(1 + \frac{|x|}{\epsilon}) & \text{if } |x| < w, \\ |x| - C & otherwise \end{cases} \quad (9)$$

where C = $\omega \ln(1 + \frac{|x|}{\epsilon})$. In this method, we optimize the parameters of modified VGG16 network augmented with the attention mechanism as described in Rodriguez et al.[2] using the wing loss [19] instead of the mean square error.

Table 1. Comparison of normalized mean squared error(NME) for various CNN model for landmark regression performance across five folds.

Model	Normalized mean squared error					Mean
VGG16	18.44	16.93	33.77	12.73	16.61	19.70
Cascade	12.50	11.11	11.75	15.96	11.88	12.64
Attention	14.72	10.47	12.96	11.78	10.43	12.07
Attention WingLoss	14.32	10.30	12.66	10.93	9.10	11.45

4 Experiments and Results

The training and evaluation methodology was consistent for all the convolution models. Five-fold cross validation [17] was used for model evaluation. At each iteration, four folds were used for training and the fifth for testing and evaluation. The data augmentation (Section 3.2) was performed on the training set but not on the test and evaluation set. In each iteration, the training was performed for 50 epochs. The mean squared error (NME) Eq. 10 was the loss function and the adam[18] optimizer was used for optimization. The NME is an evaluation metric to compare convolutional model performance.

$$NME = \frac{\sum_{i=1}^{N} \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}}{N} \quad (10)$$

such that N is the number of landmarks, $\hat{x}_i, \hat{y}_i$ are the predicted landmark coordinates and x_i, y_i are the ground truth landmark coordinates. This is summarized for the models tested in Table 1.

5 Discussion and Conclusion

The results in Table 1 indicate that the cascaded CNN approach and the attention augmented approaches have lower average normalized mean squared error compared to the vanilla VGG16 network. The

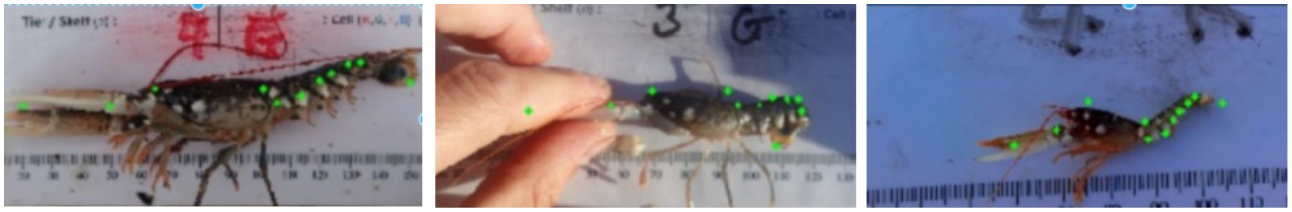

Figure 5. The above figure shows the predicted landmarks on test data trained using the modified VGG16 augmented with the attention mechanism(Rodrigeuz et al[2] and optimized using the wing loss.

cascaded CNN approach uses a series of CNNs where each CNN refines landmark predictions from the previous stage. The input image to a CNN is an image patch around the landmark predicted from the previous stage. The idea is to learn more detailed features around the landmark points for more accurate predictions. However, multiple cascaded regressors can increase compute and memory requirements. The regressors cannot be trained simultaneously since the input to subsequent CNNs depend upon the output of previous regressors. CNNs with attention mechanisms[2] [3] learn to pay attention to regions for fine grained inference. The modified VGG16 augmented with attention mechanism1 perform marginally better than cascaded CNNs. However, they predict landmarks in a single iteration and use a single model. The CNN augmented with attention mechanism trained using wing loss has lower NME than training with mean squared loss. The wing loss makes the CNN robust to outliers like occlusions since it emphasizes errors by giving more weight to smaller errors.

References

[1] Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. IEEE Transactions on Pattern Analysis & Machine Intelligence, (6), 681-685.

[2] Rodrguez, Pau, et al.. "Attend and Rectify: a Gated Attention Mechanism for Fine-Grained Recovery." (2018).

[3] Yue, L., Miao, X., Wang, P., Zhang, B., Zhen, X., & Cao, X. Attentional Alignment Network.

[4] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

[5] Dollar, P., Welinder, P., & Perona, P. (2010, June). Cascaded pose regression. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (pp. 1078-1085). IEEE.

[6] Xiong, X., & De la Torre, F. (2013). Supervised descent method and its applications to face alignment. In Proceedings of the IEEE conference on computer vision and pattern recognition.

[7] Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3476-3483).

[8] Miao, X., Zhen, X., Liu, X., Deng, C., Athitsos, V., & Huang, H. (2018). Direct Shape Regression Networks for End-to-End Face Alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5040-5049).

[9] Zhang, J., Shan, S., Kan, M., & Chen, X. (2014, September). Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In European Conference on Computer Vision (pp. 1-16). Springer, Cham.

[10] Trigeorgis, G., Snape, P., Nicolaou, M. A., Antonakos, E., & Zafeiriou, S. (2016). Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4177-4187).

[11] Aleju/imgaug. https://github.com/aleju/imgaug

[12] Lindeberg,T. (2012). Scale invariant feature transform.

[13] Costa, C., Loy, A., Cataudella, S., Davis, D., & Scardi, M. (2006). Extracting fish size using dual underwater cameras. Aquacultural Engineering, 35(3), 218-227.

[14] Strauss, R. E., & Bookstein, F. L. (1982). The truss: body form reconstructions in morphometrics. Systematic Biology, 31(2), 113-135.

[15] Mao, R., Lin, Q., & Allebach, J. P. (2018). Robust Convolutional Neural Network Cascade for Facial Landmark Localization Exploiting Training Data Augmentation. Electronic Imaging, 2018(10), 374-1.

[16] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). Ieee.

[17] Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).

[18] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[19] Feng, Z. H., Kittler, J., Awais, M., Huber, P., & Wu, X. J. (2017). Wing loss for robust facial landmark localisation with convolutional neural networks. arXiv preprint arXiv:1711.06753.

[20] Larochelle, H., & Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order Boltzmann machine. In Advances in neural information processing systems (pp. 1243-1251).

[21] Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S. H., Murthy, M., & Shaevitz, J. W. (2018). Fast animal pose estimation using deep neural networks.