Scale-Invariant Cascaded Convolutional Neural Network Face Detection Algorithm Postprint

Authors: Zheng Chenghao, Liu Bing, Zhou Yong

Date: 2018-05-18T00:00:00+00:00

Abstract

Convolutional Neural Networks require fixed-size images as input for image processing, a constraint that leads to substantial information loss in the original image during scaling. Furthermore, current face detection algorithms predominantly utilize single-structure networks for feature extraction, resulting in weak generalization capability. To address these two limitations, we propose a face detection algorithm that combines cascaded convolutional neural networks with spatial pyramid pooling. This approach concatenates three-level convolutional neural network models that are distinct from one another, with structures ranging from simple to complex, to extract different facial features and filter images at various network levels, thereby accomplishing detection of face regions in images. Additionally, a spatial pyramid pooling layer is incorporated at each network level; this pooling strategy eliminates the need for fixed-size input, thus increasing the flexibility of model input dimensions. On standard face datasets, the proposed method achieves multi-scale model input compared to traditional approaches, improves detection performance, and reduces face detection time.

Full Text

Face Detection Algorithm Based on Scale-Independent Cascade Convolutional Neural Network

Zheng Chenghao[1], Liu Bing[1,2], Zhou Yong[1] [1]. School of Computer Science & Technology, China University of Mining & Technology, Xuzhou, Jiangsu 221116, China [2]. Institute of Electronics, Chinese Academy of Sciences, Xuzhou, Jiangsu 221116, China

Abstract

Convolutional neural networks require fixed-size images as input, which causes significant information loss during image scaling. Additionally, most existing

face detection algorithms employ single-structure networks for feature extraction, resulting in weak generalization capability. To address these two issues, this paper proposes a face detection algorithm that combines cascade convolutional neural networks with spatial pyramid pooling. The method connects three distinct convolutional neural network models in series, with structures ranging from simple to complex, extracting different facial features at different network levels to detect face regions in images. Simultaneously, a spatial pyramid pooling layer is incorporated at each network level. This pooling strategy eliminates the need for fixed-size inputs, increasing flexibility in model input dimensions. Experiments on standard face datasets demonstrate that, compared with traditional methods, the proposed approach achieves multi-scale input capability, improves detection performance, and reduces face detection time.

Keywords: cascade convolutional neural network; spatial pyramid pooling; face detection

0 Introduction

Face detection is a hot research topic in object detection and recognition, involving the identification of face regions in arbitrary images to return face positions or other information. Early image recognition systems primarily relied on methods such as Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) for feature extraction, followed by classification. These features were essentially hand-crafted, and their quality significantly impacted system performance. This required researchers to have deep understanding of the problem to design effective features, thereby improving system performance. Image detection and recognition systems from this era were mostly tailored to specific problems, resulting in poor overall generalization capability. Additionally, the data volumes processed by these systems were relatively small, making it difficult to achieve accurate recognition in practical applications [1].

Deep learning, a branch of machine learning, has achieved major breakthroughs and become a research hotspot in recent years [2~6]. Since 2011, researchers first applied deep learning to speech recognition, improving accuracy by 20%–30% and achieving the most significant progress in a decade. Just one year later, deep learning models based on convolutional neural networks achieved substantial performance improvements on large-scale image classification tasks, sparking a deep learning boom [7,8].

However, when convolutional neural networks process images, they require fixed-size inputs. This is because after convolution and pooling operations, the resulting data is fed into fully connected layers with a fixed number of neurons, which means the number of weights connected to these layers must remain constant. If the weight count changes, weight computation and updating become impossible. Consequently, almost all convolutional networks require input images to be scaled or cropped to uniform dimensions before training or testing.

To address this issue, He et al. [9] proposed spatial pyramid pooling (SPP) to enable multi-scale inputs for convolutional neural networks. SPP adds a pyramid pooling layer before the fully connected layers, transforming feature maps of different sizes from convolution and pooling operations into uniformly dimensioned data for the subsequent fully connected layers. This eliminates the need for uniform input sizes, allowing images to retain more information during preprocessing and increasing the likelihood of extracting key features later.

Moreover, many convolutional neural network-based image processing tasks employ single network models, resulting in relatively homogeneous feature extraction and weak generalization performance. Such models may work well for certain problems but perform poorly on others. The three main challenges in face detection are: (a) excessive variability of faces in cluttered scenes; (b) the vast number of possible face positions and sizes in images; and (c) insufficient robustness of single-structure models for highly variable problems.

To tackle these challenges, Li et al. [10] proposed a cascade convolutional neural network model that effectively addresses these difficulties. The theoretical foundation of cascade convolutional neural networks stems from Viola et al.' s [15] 2001 algorithm using a boosted cascade of simple features, which established an approach to combine simple features into classifiers. While many improvements to the V-J algorithm were subsequently proposed, the cascade number and simple features used affected detection accuracy. Moreover, the selected features often required manual design—when feature selection was erroneous, overall model performance degraded significantly, and generalization capability for complex scenarios remained weak. For these reasons, Li et al. chose convolutional neural networks for feature extraction. Unlike hand-crafted features, convolutional networks can automatically learn features and capture various complex and variable conditions in face regions during training on large datasets, which is crucial for building accurate face detection algorithms.

This paper proposes a scale-independent deep convolutional neural network face detection algorithm by combining pyramid pooling with cascade convolutional neural networks. First, a three-level cascade convolutional neural network model is designed based on cascade network principles. Then, pyramid pooling is embedded into each level, ensuring no level is affected by varying input image scales. The overall face detection process consists of three steps: (a) scanning the test image with sliding windows to generate candidate regions; (b) feeding candidates into the trained cascade network for face classification; and (c) applying non-maximum suppression (NMS) to classified face images for final integration and marking face regions in the original image. Experiments on standard face datasets demonstrate significant improvements in detection performance and time efficiency compared to traditional methods.

1 Convolutional Neural Networks

Convolutional neural networks originated in the early 1960s when Hubel and Wiesel, through research on cats' visual cortex systems, proposed the concept of receptive fields and discovered hierarchical information processing mechanisms in visual pathways, earning them the Nobel Prize in Physiology or Medicine. In the mid-1980s, Fukushima et al. built upon receptive field concepts to propose the neocognitron, considered the first implementation of convolutional neural networks and the first artificial neural network based on local connectivity and hierarchical organization among neurons. In 1990, LeCun et al. first proposed a gradient backpropagation-trained convolutional neural network for handwritten digit recognition, demonstrating superior performance on the MNIST dataset compared to contemporary methods. Today, convolutional neural networks have become a research hotspot in image recognition, representing the first truly successful algorithm for training multi-layer neural networks with clear advantages for multi-dimensional input signals [11~13].

1.1 Cascade Convolutional Neural Networks

Recent face detection research has focused on uncontrollable aspects of recognized face regions, such as exaggerated expressions, pose variations, and occlusions [14], all of which affect final detection performance. Given these challenges, single-structure models struggle to achieve good generalization, resulting in low robustness for practical applications.

Li et al. [10] proposed a cascade convolutional neural network algorithm at CVPR2015. The simplified model structure is shown in [Figure 1: see original paper]. This model connects different convolutional network structures for progressive feature extraction, ranging from simple to complex. The initial simple network performs coarse feature extraction, roughly classifying input images. Images classified as faces are fed to the next, more complex network for finer classification. This process repeats until the final, most complex network produces the ultimate classification result. By leveraging different models for feature extraction with progressively increasing precision, the approach reduces detection time while improving accuracy.

The main structure of Li et al.' s cascade convolutional neural network is shown in [Figure 2: see original paper]. The model consists of three cascade networks, each comprising a binary classification network (12-net, 24-net, 48-net) and a calibration network (12-calibration, 24-calibration, 48-calibration). The three networks differ primarily in input image resolution, which gradually increases to improve recognition accuracy while reducing runtime and enhancing efficiency. Additionally, the three networks have different structures, clearly progressing from simple to complex. The simpler early networks perform coarse feature extraction, while later complex networks provide more precise classification of filtered images. The workflow processes test images through the first binary classification network. If classified as a face, the image enters the first calibra-

tion network for position adjustment; otherwise, it is discarded. Face-classified images then proceed to the second binary classification network for similar processing. After the final network, face region candidates undergo non-maximum suppression (NMS) to mark face positions in the original image.

This model achieves good accuracy on standard face datasets. Moreover, thanks to the simpler structures of the first two networks, overall detection speed improves significantly, with substantially reduced time compared to other traditional networks.

1.2.1 Introduction to Spatial Pyramid Pooling

Most current convolutional neural networks require fixed-size inputs, necessitating data scaling to uniform dimensions before training or testing. For example, the famous AlexNet model requires 227×227 input images. This preprocessing step (scaling or cropping to uniform size, as shown in [Figure 3: see original paper]) prevents traditional CNNs from handling multi-scale inputs when image sizes vary, causing greater data loss compared to multi-scale preprocessing and impacting subsequent training and testing.

He et al. [9] proposed spatial pyramid pooling (SPP) to address this scale variation problem. Since convolutional and pooling layers don' t require fixed-size inputs—only the fully connected layers after them do—we can observe that these layers can process arbitrary image sizes without preprocessing. For instance, a 100×100 input produces 5×98×98 feature maps after five 3×3 convolutions, while a 102×102 input yields 5×100×100 feature maps. After 2×2 pooling, these become 25×25 and 26×26 feature maps, respectively. Thus, convolutional and pooling layers can handle any input size, but fully connected layers require fixed dimensions. If the final convolutional layer has 50 outputs and the next fully connected layer has 1,000 neurons, the connection matrix dimension is 50×1000. If input image sizes vary, the matrix dimensions change, preventing network training or testing.

SPP solves this by adding a spatial pyramid pooling layer before fully connected layers, ensuring any input image size is processed into uniform-dimensional data. As shown in [Figure 4: see original paper], input images no longer require preprocessing, enabling multi-scale inputs for convolutional neural networks.

1.2.2 Spatial Pyramid Pooling Algorithm

As shown in [Figure 5: see original paper], this is a traditional network architecture with convolutional layers followed by fully connected layers. The solution involves adding a pyramid pooling layer before the fully connected layers to handle varying input image sizes.

The pyramid pooling layer performs three pooling operations on the feature map from the previous convolutional layer: the top operation pools the entire feature map, the middle divides it into four parts for pooling, and the bottom

divides it into sixteen parts. This produces a 16+4+1=21 dimensional feature vector for the fully connected layer, solving the inconsistent input size problem. Each pooling operation uses basic methods (e.g., max pooling) but with different window sizes and strides.

Algorithm 1: SPP Method Analysis Process *Input:* Feature matrix after convolution and pooling
Output: Feature vector after spatial pyramid pooling

Step 1: Calculate w = ceil(a/n), h = ceil(b/n), stride1 = floor(a/n), stride2 = floor(b/n), where n = 1, 2, 3···
Step 2: Perform pooling on feature matrix X using the calculated parameters to obtain features f1, f2, f3···
Step 3: Concatenate the obtained features to produce new feature F.

2 Model and Algorithm Based on Pyramid Pooling Cascade Convolutional Neural Network

2.1 Model Design

While cascade convolutional neural networks demonstrate excellent performance in face detection, they don' t inherently support multi-scale inputs, causing significant information loss during preprocessing. Spatial pyramid pooling solves the multi-scale input problem for convolutional neural networks. Therefore, this paper proposes a scale-independent cascade convolutional neural network face detection algorithm by combining cascade networks with SPP advantages.

The model structure is shown in [Figure 6: see original paper]. A pyramid pooling layer with five channels is added before the fully connected layers, enabling each cascade level to support multi-scale image inputs while maintaining the overall simple-to-complex structural progression. Compared with [10], this algorithm also employs a three-level cascade structure but omits calibration networks for faster processing. The resulting model contains only three convolutional networks, significantly reducing both training and detection time.

2.2 Algorithm Description

1) Training Phase
The training phase uses the AFLW face dataset as positive samples. AFLW contains approximately 21,000 images (mostly high-resolution) with about 24,000 annotated face rectangles, 3D rotation angles, occlusion information, and glasses status. Positive training samples are generated by cropping face regions according to AFLW annotations, then applying random translation, rotation, and flipping. Negative samples are randomly cropped from COCO dataset images containing no faces. The final dataset comprises approximately 35,000 positive and 30,000 negative samples.

Due to SPP' s ability to handle multi-scale inputs, training employs two resolu-

tions. Samples are first scaled to 24×24 for initial training. After convergence, the trained model is further trained with 12×12 images until convergence again, completing the training process.

2) Testing Phase

During testing, input images undergo preprocessing through pyramid image processing to generate a set of images at different scales. All images in this set are processed using sliding windows. The sliding window first operates at 24×24 size, generating all 24×24 candidate regions and recording their positions in the original image. All candidates are fed into the trained cascade model, where the first level eliminates non-face candidates while preserving face candidates. Subsequent networks repeat this process until the final level produces ultimate candidates. These undergo non-maximum suppression (NMS) to remove highly overlapping boxes, yielding final face region candidates that are marked in the original image. Additional testing with 12×12 sliding windows is performed to complete multi-scale evaluation. The overall algorithm flow is described in Algorithm 2.

Algorithm 2: Cascade Convolutional Neural Network Detection Algorithm Based on Pyramid Pooling

Input: Image to be detected

Output: Image with detected face regions marked

a) Train the model using training samples

b) Apply pyramid image processing to the test image to obtain multi-scale image sets

c) Apply 24×24 (or 12×12) sliding windows to all images in the set and record position information

d) Feed all candidate windows into the trained first-level CNN (containing: 1 convolutional layer with 32 3×3 filters, 1 pooling layer with 3×3 max pooling, 1 pyramid pooling layer with 5 channels, 1 fully connected layer with 64 neurons)

e) Feed first-level filtered candidates into the second-level CNN (containing: 1 convolutional layer with 64 5×5 filters, 1 pooling layer with 3×3 max pooling, 1 pyramid pooling layer with 5 channels, 1 fully connected layer with 128 neurons)

f) Feed second-level filtered candidates into the third-level CNN (containing: 2 convolutional layers with 64 5×5 filters, 2 pooling layers with 3×3 max pooling, 1 pyramid pooling layer with 5 channels, 1 fully connected layer with 256 neurons)

g) Apply NMS to final candidate windows, removing windows with similarity

> 0.5

h) Mark detected face regions in the original image based on candidate window position information

3 Experimental Results Analysis

3.1 Training Phase Analysis

During training, we first compared model convergence and training speed with spatial pyramid pooling layers when training images were 12×12 versus 24×24. Then we compared convergence and speed with and without pyramid pooling layers for 24×24 inputs. [Figure 7: see original paper] shows training results for the second-level CNN.

When input sizes were 12×12 and 24×24, convergence behavior was nearly identical, as shown in [Figure 7: see original paper]. However, training time differed significantly: the 12×12 model trained approximately 3,000 samples per second, while the 24×24 model trained only about 750 samples per second. This occurs because, with identical model structures and parameters, smaller images contain less data, enabling faster computation.

This demonstrates that with spatial pyramid pooling layers, different input image sizes yield similar convergence behavior (both converging around 8,000 iterations), meaning input size doesn' t affect model convergence.

Subsequently, we compared training effects with and without pyramid pooling layers for 24×24 inputs, as shown in [Figure 8: see original paper]. The results indicate that with pyramid pooling, models converge earlier. This is because pyramid pooling implements coarse-to-fine feature extraction, enabling faster identification of key information from global to local perspectives and providing better data generalization, thus accelerating convergence.

3.2 Testing Phase Analysis

Testing employed the FDDB face dataset, a globally authoritative face detection platform containing 2,845 images with 5,171 faces. The test set includes images with various poses, resolutions, rotations, and occlusions, in both grayscale and color. Standard face annotations are elliptical; for this test, we converted these to rectangular bounding boxes using the standard elliptical annotation information.

Comparison with and without SPP
[Figure 9: see original paper] compares detection performance with and without spatial pyramid pooling layers (all using 24×24 inputs). The ROC curve (receiver operating characteristic curve) is used as the performance metric, where larger area under the curve indicates better performance. FDDB evaluation uses both continuous and discontinuous scores; we show continuous score results here, with both scores presented in final comparisons.

The three-level CNN with SPP at each level demonstrates superior detection performance. Additionally, the SPP-equipped model' s curve stabilizes more quickly, indicating more stable detection performance and high accuracy even with smaller sample sizes. As sample numbers increase, performance remains consistent, demonstrating that pyramid pooling improves both detection performance and model robustness.

Different Resolution Inputs
We tested model performance with 12×12, 18×18, and 24×24 input sizes, as shown in [Figure 10: see original paper]. The three input sizes produce similar results, with 24×24 performing best, followed by 12×12, then 18×18.

Although the model was only cross-trained on 12×12 and 24×24 sizes, the 24×24 results are superior for two reasons: (a) scaling to 12×12 loses more information, reducing extracted features compared to 24×12; (b) pyramid image operations are required before input, and when 12×12 windows scan complete face regions, the already-scaled images have high distortion, while 24×24 windows capture more complete information. Despite performance differences among the three sizes, detection results remain very close. Notably, the model wasn' t trained on 18×18 images, yet achieves similar results, demonstrating that pyramid pooling-enabled cascade CNNs can handle multiple input scales with comparable accuracy.

Comparison with State-of-the-Art Methods
[Figure 11: see original paper] compares our model with well-known methods: the classic V-J face detection model, SURF cascade-based face detection [16], and Li et al.' s cascade CNN model [10]. Our algorithm shows superior detection capability compared to traditional V-J and SURF-based methods, achieving ideal results for various face conditions and demonstrating good generalization on large datasets.

Compared with Li et al.' s cascade CNN [10], our algorithm shows slightly lower performance but significantly reduces model complexity and training time. Detection time is also substantially reduced: Li et al.' s model achieved approximately 14 FPS on 640×480 images using an E5-2620 CPU, while ours reaches about 19 FPS on the same size images using a lower-performance i7-4712MQ CPU (with other hardware differences also contributing). Furthermore, the pyramid pooling layer enables multi-scale image detection while improving performance.

[Figure 12: see original paper] and [Figure 13: see original paper] show final detection results. Since sliding windows are square, detection boxes are also square. Results for 24×24 and 12×12 inputs show minimal differences. Overall, the proposed model achieves multi-scale input capability for convolutional neural networks while improving performance over traditional algorithms.

4 Conclusion

This paper addresses the problems of convolutional neural networks' inability to support multi-scale data input and the weak generalization capability of single-structure models for complex scenarios. We propose a face detection algorithm based on pyramid pooling cascade convolutional neural networks. By introducing pyramid pooling, we enable multi-scale inputs for convolutional neural networks and improve performance to some extent. Combining this with cascade convolutional networks moves beyond single network structures, enhancing generalization capability for complex conditions.

Future improvements include: (1) modifying square detection boxes to rectangular or even elliptical shapes for more precise detection in face datasets; (2) further reducing candidate window numbers to improve computational efficiency.

References

[1] Lu Hongtao, Zhang Qinchuan. Survey on deep convolutional neural networks in computer vision [J]. Data Acquisition and Processing, 2016, 31(1): 1-17.
[2] Lecun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521(7553): 436-444.
[3] Schmidhuber J. Deep learning in neural networks: an overview [J]. Neural Networks the Official Journal of the International Neural Networks Society, 2014, 61: 85-117.
[4] Guo Y, Oerlemans A, Lao S, et al. Deep learning for visual understanding [J]. Neurocomputing, 2016, 187(C): 27-48.
[5] Sun Zhiyuan, Lu Chengxiang, Shi Zhongzhi, et al. Deep learning research and development [J]. Computer Science, 2016, 43(2): 1-8.
[6] Sun Zhijun, Xue Lei, Xu Yangming, et al. Survey on deep learning [J]. Computer Application Research, 2012, 29(8): 2806-2810.
[7] Duc H H, Jung K. Applying tensorflow with convolutional neural networks to train data and recognize national flags [C]// Advanced Multimedia and Ubiquitous Engineering. 2017.
[8] Bianco S, Buzzelli M, Mazzini D, et al. Deep learning for logo recognition [J]. Neurocomputing, 2017, 245(C): 23-30.
[9] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2014, 37(9): 1904-1916.
[10] Li Haoxiang, Lin Zhe, Shen Xiaohui, et al. A convolutional neural network cascade for face detection [C]// Proc of Computer Vision and Pattern Recognition. 2015: 5325-5334.
[11] Bouvrie J. Notes on convolutional neural networks [J]. Neural Nets, 2006, 31(1): 1-17.
[12] LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series [M]// The Handbook of Brain Theory and Neural Networks. 1995.
[13] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep

convolutional neural networks [C]// Proc of International Conference on Neural Information Processing Systems. 2012: 1097-1105.
[14] Hao Biao, Kang D S. The research of face expression recognition based on CNN using tensorflow [J]. Journal of Advanced Information Technology and Convergence, 2017, 7.
[15] Viola P A, Jones M J. Rapid object detection using a boosted cascade of simple features [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition, 2003: 511-518.
[16] Li J, Wang T, Zhang Y. Face detection using SURF cascade [C]// Proc of IEEE International Conference on Computer Vision Workshops. 2012: 2703-2710.

Note: Figure translations are in progress. See original paper for figures.