Thesis for the Degree of Doctor of Philosophy

TOWARDS IMAGE SEMANTIC SEGMENTATION AND CLASSIFICATION USING BRACKET-STYLE CONVOLUTIONAL NEURAL NETWORK AND ITS VARIANTS

HUA CAM HAO

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Republic of Korea

February 2022

TOWARDS IMAGE SEMANTIC SEGMENTATION AND CLASSIFICATION USING BRACKET-STYLE CONVOLUTIONAL NEURAL NETWORK AND ITS VARIANTS

HUA CAM HAO

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Republic of Korea

February 2022

Towards Image Semantic Segmentation and Classification using Bracket-style Convolutional Neural Network and Its Variants

by

HUA CAM HAO

Supervised by

Prof. Sungyoung Lee
Prof. Sung-Ho Bae

Submitted to the Department of Computer Science and Engineering
and the Faculty of the Graduate School of Kyung Hee University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Dissertation Committee:

Prof. Seungkyu Lee _____

Prof. TaeChoong Chung _____

Prof. Seong-Bae Park _____

Prof. Seok-Won Lee _____

Prof. Sung-Ho Bae _____

Prof. Sungyoung Lee _____

KYUNG HEE UNIVERSITY

Abstract

Department of Computer Science and Engineering

Doctor of Philosophy

Towards Image Semantic Segmentation and Classification using Bracket-style Convolutional Neural Network and Its Variants

by Hua Cam Hao

Nowadays, thanks to the exponential advancements of computational resources along with the massive surge of image quantity and quality, deep learning technique, a special branch of Artificial Intelligence, achieves extraordinary performance in various computer vision tasks comprising image classification and semantic segmentation. Besides that, in the current era of Industry 4.0, vision-oriented applications become vastly significant in everyday life, smart healthcare, and industrial manufacture, to name a few. Accordingly, in the literature, there emerges tremendous researches that introduce deep learning architecture in form of convolutional neural network (CNN) for tackling the problem of understanding image semantically for the above-mentioned software products. However, since there are still limitations in the related works of semantic image segmentation and image classification in several specialized domains, this thesis presents a Bracket-style CNN and its variants to tackle the existing issues, respectively.

Firstly, regarding the problem of semantic image segmentation, which is equivalent to image's pixel-level classification, the key mechanism in a predefined deep learning model is to be capable of coordinating globally contextual information with locally fine details in the input image for generating optimal segmentation map. But nonetheless, existing work did not exhaustively exploit middle-level features in the CNN, which carry reasonable balance between fine-grained and semantic information, to boost the effectiveness of the above-mentioned procedure. Hence, a Bracket-shaped CNN is proposed to leverage the exploitation of middle-level feature maps in a tournament by exhaustively pairing adjacent ones through attention embedded combination modules. Such routine repeats round-by-round until the prediction map of densely enriched

semantic contexts is finalized. It is worth noting that the approach of combining two neighboring feature maps having different resolutions is defined by adopting a cross-attentional fusion mechanism, namely CAF module. The major objective is to properly fusion semantically rich information (of the lower-resolution inputs) with finely patterned features (of the higher-resolution versions) for the outputs. As a consequence, the proposed semantic segmentation model is trained and evaluated on three well-known datasets, from which competitive performance in terms of mean Intersection of Union (compared to novel methods in the literature) is attained as follows: PASCAL VOC 2012 [20] (83.6%), CamVid [9] (76.4%) and Cityscapes [18] (78.3%) datasets. Furthermore, the proposed architecture is shown to be flexibly manipulated by round-wise features aggregation to perform the per-pixel labeling task efficiently on dataset with heavily class-imbalancing issue such as DRIVE [80], which aims at retinal blood vessel segmentation, in comparison with the state-of-the-arts. Particularly, Sensitivity, Specificity, Accuracy, and Area Under the Receiver Operating Characteristics achieve 79.32%, 97.41%, 95.11%, and 97.32%, respectively.

Secondly, the proposed Bracket-style concept in this thesis can be extended as variants for effectively classifying image in specialized domains such as Diabetic Retinopathy (DR) grading and facial expression recognition (FER). Concretely, in such kind of deep learning model, channel-wise attentional features of semantically-rich (high-level) information are integrated into finely-patterned (low-level) details in a feedback-like manner, a.k.a. single-mode Bracket-structured network (sCAB-Net). Accordingly, feature maps of different scales can be amalgamated for extensively involving spatially-rich representations to the final predictions. From the evaluation process, impressive benchmark results on the aforementioned areas, wherein spatially-rich factors play an important role to the decision of image label, are achieved. On the one hand, with respect to DR recognition, the proposed architecture reaches a remarkable quadratic weighted kappa of 85.6% on Kaggle DR Detection dataset [47]. On the other hand, about FER, it gains a mean class accuracy of 79.3% on RAF-DB dataset [58].

In overall, the above-mentioned operational characteristics and experimental achievements demonstrate a promising capability of the proposed Bracket-style network toward complete image understanding (by either semantic segmentation (pixel-level labeling) or classification (image-level labeling) performance) for further practical computer perception-based applications.

Acknowledgements

Doctoral thesis is the final benchmark to assess the capabilities of investigation, exploitation, research & development, and summarization from documentaries specialized in a narrowed topic for making valuably technical contributions to science and society. Moreover, it is obvious that the ability of utilizing knowledge accumulated during a decade (starting from an undergraduate student at the age of 18 till now), logical thinking, proficiency, and creativity are all amalgamated in this thesis - a pivotal apparatus for the post-graduate career in either academia or industry. During the progress of completing this thesis, I have realized that what I achieved is far much more than which I have ever thought or imagined before. Thus, first of all, I would like to express nothing but sincere gratitude to all of those whom I met, talked, discussed, or just simply greeted with a smile on the corridor during such a long and tough Ph.D. journey.

I would like to express my deep gratitude to Professor Sungyoung Lee and Professor Sung-Ho Bae as well as Professor Thuong Le-Tien (Ho Chi Minh City University of Technology, Vietnam), the professional and enthusiastic supervisors during my graduate and undergraduate studies journey, respectively. Thanks to their valuable advices, I have gradually improved research and development skills from how to perform a comprehensive literature survey, brainstorm and conceptualize new ideas, write a whole scientific paper in an proficient manner, upgrade technical problem-solving skills, and promote presentation strategies to wide-ranging audiences.

I would like to express deep gratitude to my parents and relatives, who always support, encourage, and motivate me to overcome difficulties when studying abroad. As for me, keeping up all the efforts for a successful doctoral career is the best way to acknowledge their loves for me.

I would like to thank all of my colleagues in the Ubiquitous Computing Laboratory from the day I join Kyung Hee University as a graduate student, especially Dr. Thien Huynh-The, who are cooperative, warm-hearted, and willing to support to each other not only for research comments and feedbacks but also other aspects in daily life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview of Deep Learning

Over the last decade, the world has witnessed extraordinary advancements of vision-related technology that human beings experience in daily life. For instance, image recognition technology is widely adopted in smart devices from big-tech corporations. End-users can easily search and sort their pictures based on specific objects without any efforts of tagging from a cat to a rainy scenery or even abstract actions like hugging, selfie, to name a few. Moreover, several high-tech products can verbally describe the image's content for the blind. When a user requests to organize an album of dog image, the application in use has to determine corresponding species ranging from Chihuahua to Shiba Inu on any background scenes, and simultaneously excluding similar images of cat or wolf. Besides that, other areas such as biomedicine, robotics, drone-based monitoring, and autonomous driving can significantly benefit from such kind of technological growth. In particular, some intelligent medical platforms are capable of interpreting X-ray, Magnetic Resonance Imaging, and Computed Tomography in an efficient manner. Lately, several large tech companies are re-defining the way we travel through the research and development of autonomous driving cars having the capability of fully scene understanding. The above-mentioned observations raise an apparent question as follows: How can those softwares and products do such incredible functions?

All those technologies fundamentally originate from the same root called deep learning, a special branch of Artificial Intelligence (AI). Many scientists still prefer calling it in the original name, which is deep neural network. Practically, it is impossible for developers to manually program the smart features mentioned above. Instead, they generate algorithm that helps the computer self-learn from gigabyte or even terabyte of relevant data (e.g. millions of natural images). This

continuous interaction gradually "train" the computer to be able to automatically recognize the images according to preset requirements. Similar to the way a baby learns to realize the surrounding environment after long-term observation and listen to what the adults communicate, the computer can "perceive" the location of a predefined object as well as "understand" the whole context in an image after numerous training iterations. For instance, how the deep neural network can recognize a dog in an image is outlined as follows. During the training procedure, the deep model is provided thousands of animal images to learn the discrimination behind. Initially, a dog image without any associated label is fed into the network. Then, shallower neural layers have various responses to different body parts of the dog. Next, deeper layers shall acquire more complicated details of the dog's appearance. Afterwards, final neurons of the network can explore the most discriminated features and abstract representations of the dog in comparison with those of other animals. At the output, this deep learning model can classify the objects based on representational features exploited from the training data of diverse perspectives.

The neural network is not a whole new technology since its debut was in 1950s. Accordingly, significant breakthroughs were made during 1980s and 1990s. However, the major reason why deep learning technology emerges again in the last 10 years is that the scientists have finally been capable of utilizing the power of computational resources along with big visual data, which are fundamental factors leveraging the effectiveness of neural network. The evolution of hardware has enabled the enormous rise of deep learning. The surge of computing resources not only takes place on Moore's Law-based device but also comes from the appearance of 1^{st}-gen Graphical Processing Unit (GPU) of NVIDIA, which brings in marvelous vision experience for computer users. Nowadays, besides providing impressive 3D gaming experience, GPU is also employed to boost the computation speed in biomedicine system, computer vision, financial modeling, etc. around 20-50% when applying deep learning compared to the conventional Central Processing Unit (CPU). The second factor, i.e., big visual data from present Internet-of-Thing (IoT) devices, has been initiated when Internet is born but only reaches peak during this new century. Those two catalysts have begun a revolution for deep learning. In 2011, the supercomputer Watson of IBM only applies AI to defeat the best player in Jeopardy game show, and to this end, deep learning has been additionally integrated for the deployment of more than 30 service groups that this system offers. In 2012, Google only launches two AI-related projects, and till now, that amount is up to 1000 for enormous frameworks, services and operating system like searching, Gmail, Android, Translation, Youtube, autonomous driving, etc.

But nonetheless, deep learning still has remarkable limitations. Firstly, it always requires a big amount of annotated data for a certain task. This process is time-consuming and demands powerful computing resources for fast training procedure. Otherwise, the final performance is not as impressive as expected. Secondly, deep learning still faces difficulties in recognizing very complex scenarios and vulnerable to adversarial attacks [26]. The primary reason is that the contemporary technology is not good enough to make decision in a transparent and logical manner like human beings. In brief, since deep learning still lies in early phase of its era, those drawbacks are unavoidable. It should take much more time so that AI system in common is fully equipped with "realistic senses" like human beings, but the development of AI era is now being leveraged to move on a super-highway in the next decades.

1.2 Image Classification and Semantic Segmentation using Deep Learning

According to previous section, extensive development of powerful computation and increment of big visual data has leveraged deep learning in numerous computer vision tasks for further industrial deployment. To this end, Convolutional Neural Network (CNN), one of the most well-known lines of deep learning technique, has attracted numerous researchers thanks to its significant performance boost in various problems ranging from categorizing overall content [17, 29, 34, 79] to labeling every single pixel [6, 67, 75] of images. Specifically, the former is basically referred to as classification issue at image level, which can be applied into human activity recognition [39, 40], disease progression identification [37], to name a few. Meanwhile, the latter is called semantic segmentation which takes a further step of doing the same job at pixel level for semantic scene understanding.

In fact, such per-pixel labeling problem remains an open research area due to the following reason: the recently rapid development of perception-related applications (e.g., medical image analysis, augmented reality, computational photography, autonomous driving) requires higher pixel-wise categorization performance for retrieving more comprehensive knowledge from the given scenes. For example, segmenting regions of interest semantically from a medical image can provide valuable information such as tumor density (by pixel level) to the physicians for better diagnosis and treatment [7]. As a result, a large amount of semantic segmentation models has

been proposed and benchmarked with large-scale datasets [9, 18, 20] for being efficiently applied into the aforementioned technologies.

1.3 Problem Statement

Generally, to tackle such pixel-wise grouping problem, most existing approaches have utilized CNN primarily designed for classifying images like VGGNet [79], ResNet [29], Xception [17], to name a few, as the backbone network to exhaustively exploit its powerful feature representation. In concrete, shallow layers learn finely patterned but weakly semantic features due to partial view on the original inputs. Oppositely, deep layers acquire feature maps which depict abstract appearance (a.k.a., coarse pattern) but carry semantically rich information due to multiple subsampling stages and larger field of view on the input images, respectively. In concrete, features learned at shallow layers are finely patterned but weakly semantic due to partial view on the original inputs. Oppositely, features acquired at deeper layers depict abstract appearance (a.k.a., coarse pattern) but carry semantically-rich information due to multiple subsampling stages and larger field of view on the input images, respectively. In other words, following the feedforward process of the CNNs, wherein spatial resolution of the learned feature maps gradually decreases while corresponding channel dimension increases significantly, local details and global contextual information are extracted successively. Besides that, since semantic segmentation framework aims to generate densely labeled output having spatial size same as that of the input, it emerges the following research question: *How to design an optimal decoding strategy being able to balancedly combines local information (finely patterned features) with global context (semantically rich features) extracted from shallow-to-deep layers of the backbone CNN?*

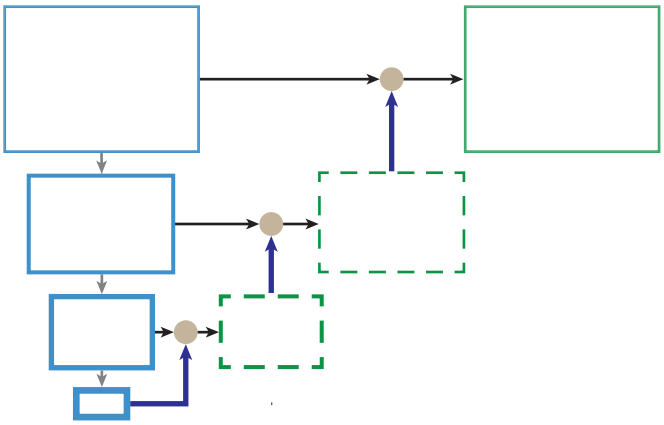
To address this, Fully Convolutional Network (FCN) [67] - the pioneering model of end-to-end trainable segmentation architecture - utilized skip-connection mechanism to fuse contextual information captured from middle- to high-level layers. Accordingly, to resolve those drawbacks as well as take the segmentation performance to new heights, numerous efforts have been made in the literature. In terms of network topology, there are two major groups, i.e., *symmetrically-* [6, 8, 42, 52, 59, 62, 63, 72, 75, 83] and *asymmetrically-structured* [12, 23, 54, 56, 96–98, 101–103, 105] frameworks as shown in Fig. 1.1a and 1.1b, respectively. In specific, the former reversely imitates information flow of the feedforward process to retrieve the final output of label values in stepwise coarse-to-fine manner. It is also worth noting that the newly decoded feature maps

are usually linked with the corresponding ones in the encoding stage, i.e., backbone CNN, by skip-fusion or concatenation strategies. Such kind of procedure was proven to enhance the capability of accurately embedding semantic information into proper instances. It is also worth noting that feature maps in the encoding stage, i.e., backbone CNNs, are often linked with the corresponding ones by skip-fusion or concatenation strategies during the decoding process in such kind of approach in order to enhance the capability of accurately embedding semantic information into proper instances. Meanwhile, instead of taking into account the encoding flow as well as stage-wisely extracted feature tensors of interest, the group of *asymmetric* architecture mainly incorporates spatial pyramid pooling schemes on the coarsest-resolution feature map of the base CNN. This strategy can exploit meaningful multi-scale contexts and/or involve simple yet effective aggregation schemes as previously mentioned to refine newly inferred features for finalizing the dense prediction map.

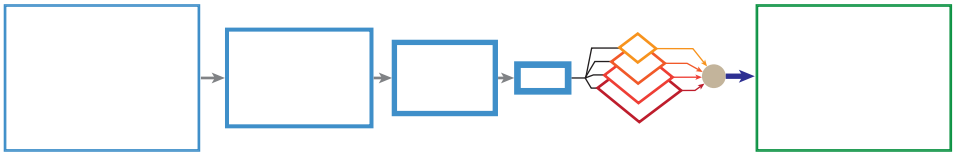
Regarding the upsampling strategy, Fig. 1.1a (i.e., *symmetrically-structured* network group) conceptually shows that only the lowest-resolution feature map inferred from the backbone CNN is upsampled step-wisely to form into the highest-resolution prediction map. In addition, during this progress, all the intermediate upsampled features are refined by counterparts learned at encoding stage via certain combination mechanisms. Note that the whole structure of this approach can be also referred to as a U-/Ladder-shaped architecture. Similarly, demonstration of the *asymmetrically-structured* network group in Fig. 1.1b delivers the same idea in which the decoding process initiates from the coarsest feature map for further spatial pyramid pooling and upsampling steps. It can be observed from these architectures that feature maps obtained at middle layers of the backbone CNN are not utilized significantly. Clearly, they just perform a single role of excluding contextual ambiguities from the corresponding upsampled versions in the *symmetrically-structured* group. Meanwhile, they even contribute nothing during the decoding stage in the *asymmetrically-structured* group.

1.4 Objectives

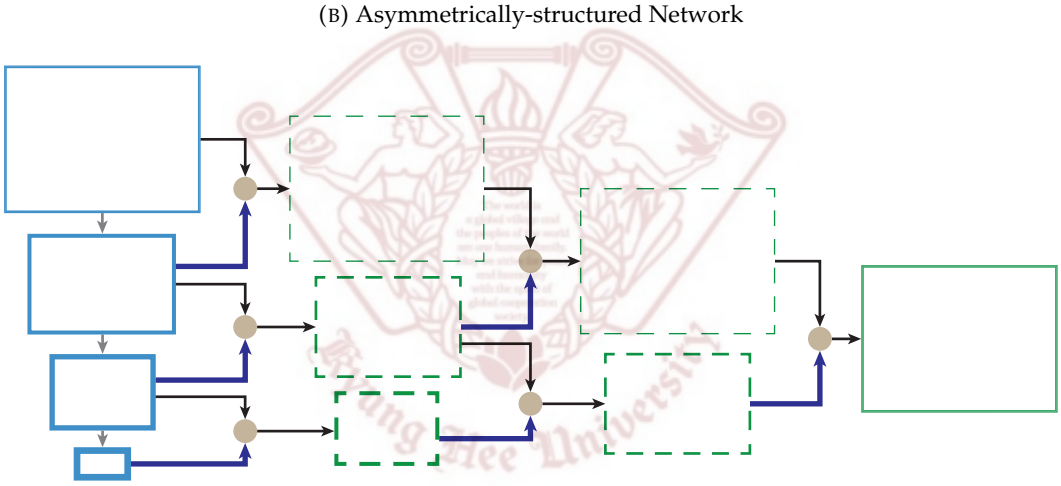
Accordingly, motivated by the fact that the middle-level features are not exploited thoroughly in the existing work, this thesis introduces a Cross-Attentional Bracket-shaped Convolutional Neural Network, namely CAB-Net, to leverage their contributions to the process of retrieving final pixel-wise prediction map. In concrete, it is hypothesized that each middle-level feature keeps



(A) Symmetrically-structured Network



(B) Asymmetrically-structured Network



(C) The proposed Bracket-structured Network

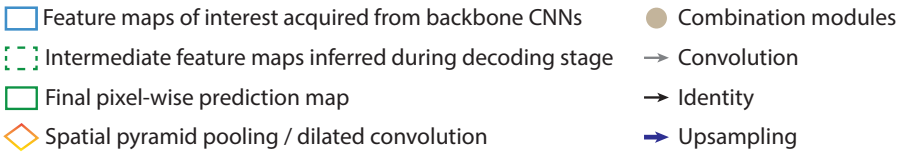


FIGURE 1.1: Conceptual diagrams of (A) symmetrically-structured network, (B) asymmetrically-structured network, and (C) the proposed Bracket-style network for semantic segmentation. Spatial and channel dimension of the feature maps are represented by corresponding perimeter and border thickness, respectively.

a reasonable balance between fine-grained details and semantic information, which is capable of simultaneously refining pixel-wise context of coarser-resolution feature maps and eliminating ambiguities existent at finer-resolution versions. Hence, as conceptually depicted in Fig. 1.1c, not only the coarsest one, every feature map of interest (except for the one with highest spatial dimension) is now upsampled and then combined with the adjacent higher-resolution version to produce

finer outputs. Continuously, these newly decoded features repeat the same procedure round-by-round until shaping the final feature map of finest resolution. Notably, to strengthen the semantic contexts in the results of the combination between two adjacent feature maps, a cross-attentional scheme inspired from SENet [34] and SCA-CNN [11] is embedded into the mergers.

Intuitively, given feature maps of different scales extracted along the feedforward process of a backbone network, i.e., $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$, conceptual operations in the *symmetrically-structured* network topology (Fig. 1.1a) can be formulated as follows.

$$\begin{aligned} \mathbf{F}_{4\to3} &= \mathcal{C}(U_{\times2}(\mathbf{F}_4), \mathbf{F}_3) \\ \mathbf{F}_{4\to3\to2} &= \mathcal{C}(U_{\times2}(\mathbf{F}_{4\to3}), \mathbf{F}_2) \\ \mathbf{F}_{4\to3\to2\to1} &= \mathcal{C}(U_{\times2}(\mathbf{F}_{4\to3\to2}), \mathbf{F}_1) \\ \mathbf{F}_{seg} &= Softmax(\mathbf{F}_{4\to3\to2\to1}) \end{aligned} \tag{1.1}$$

where $\mathcal{C}$ indicates predefined combination module between feature maps of different resolution; $U_{\times s}$ represents the operation of upsampling the considered feature map by s times; and $\mathbf{F}_{seg}$ denotes the output segmentation map. Meanwhile, the corresponding operations in *symmetrically-structured* network counterpart (Fig. 1.1b) are given as below.

$$\begin{aligned} \mathbf{F}_{4\to3} &= U_{\times2}(\mathbf{F}_4) \\ \mathbf{F}_{4\to2} &= U_{\times4}(\mathbf{F}_4) \\ \mathbf{F}_{4\to1} &= U_{\times8}(\mathbf{F}_4) \\ \mathbf{F}_{seg} &= Softmax(\mathcal{C}(\mathbf{F}_4, \mathbf{F}_{4\to3}, \mathbf{F}_{4\to2}, \mathbf{F}_{4\to1})) \end{aligned} \tag{1.2}$$

Obviously, these abstract formulations show that the contributions of the middle-level features, which possess well-defined balance between finely-patterned details and semantically contextual information, are exploited appropriately for the finalization of pixel-wise segmentation map as mentioned above. Meanwhile, exhaustive utilization of those feature maps can be attained in the

proposed Bracket-style decoding scheme (Fig. 1.1c) as demonstrated by the following equation.

$$\begin{aligned}
\mathbf{F}_{4\to3} &= \mathcal{C}(U_{\times2}(\mathbf{F}_4), \mathbf{F}_3) \\
\mathbf{F}_{3\to2} &= \mathcal{C}(U_{\times2}(\mathbf{F}_3), \mathbf{F}_2) \\
\mathbf{F}_{2\to1} &= \mathcal{C}(U_{\times2}(\mathbf{F}_2), \mathbf{F}_1) \\
\mathbf{F}_{4\to3\to2} &= \mathcal{C}(U_{\times2}(\mathbf{F}_{4\to3}), \mathbf{F}_{3\to2}) \\
\mathbf{F}_{3\to2\to1} &= \mathcal{C}(U_{\times2}(\mathbf{F}_{3\to2}), \mathbf{F}_{2\to1}) \\
\mathbf{F}_{4\to3\to2\to1} &= \mathcal{C}(U_{\times2}(\mathbf{F}_{4\to3\to2}), \mathbf{F}_{3\to2\to1}) \\
\mathbf{F}_{seg} &= Softmax(\mathbf{F}_{4\to3\to2\to1})
\end{aligned} \tag{1.3}$$

It can be realized that the middle-scale feature maps $\mathbf{F}_2$, $\mathbf{F}_3$, and $\mathbf{F}_{3\to2}$ are exploited intensively in the proposed idea, instead of only once in the *symmetrically-structured* design or even not being considered during the decoding process as in the *symmetrically-structured* counterpart. Specifically, they play both roles of semantically-richer representation (corresponding to lower-resolution input) and finer-grained representation (corresponding to higher-resolution input) in the involved combination modules. As a consequence, their contributions to the proposed hierarchical process of generating segmentation results can be leveraged more comprehensively, which is expected to further improve the segmentation accuracy compared with the existing network concepts.

Furthermore, the concept of Bracket-style CNN can be extended to tackle the image classification problem in several specialized domains such as Diabetic Retinopathy (DR) grading or Facial Expression Recognition (FER), wherein the result is inferred from the combination of various spatially rich details (e.g., structural biomarkers for the former topic or facial muscles regarding the latter) in the original image. This can be briefly explained as follows: since sequentially downsampling process along the feedforward path in the CNN vanishes various spatial structures of the image, only relying on the deepest (lowest-resolution) features for the final classifier as in Fig. 1.2a may yield misleading predictions. Based on these observations, a Single-mode Cross-Attentional Bracket-style CNN (sCAB-Net) is proposed to leverage the learnable integration of channel-wise attention at multi-level features in a pretrained CNN as conceptually illustrated in Fig. 1.2b, which allows reaching superior recognition performance in a cost-effective way. In particular, feature maps of predefined levels chosen from the backbone CNN are employed to firstly extract their channel-wisely attentional representations. Then, the obtained vectorized results are combined in a single-round Bracket-structured mechanism, of which the cross-scale outcomes are

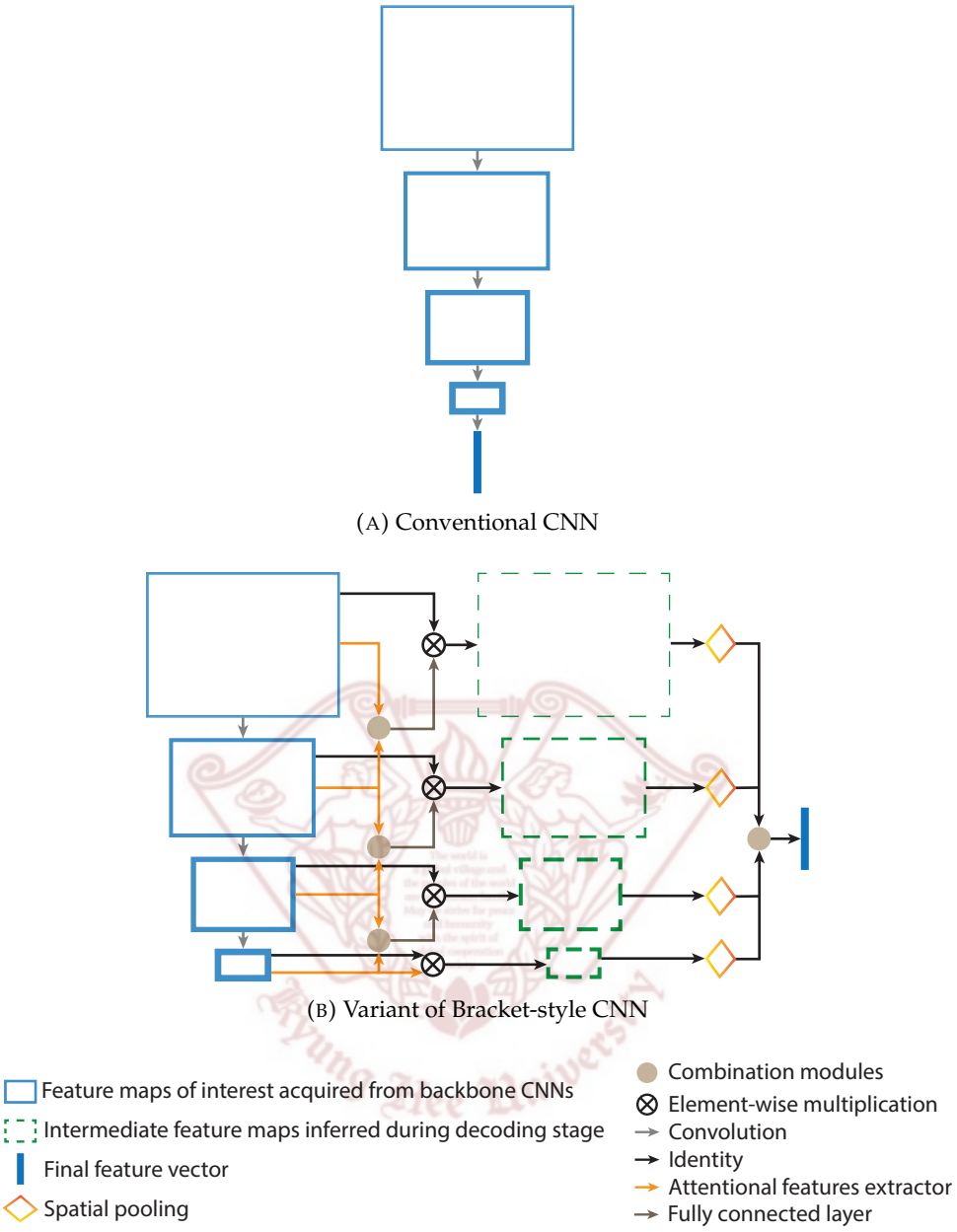


FIGURE 1.2: Conceptual diagrams of (A) conventional CNN and (B) the variant of Bracket-style CNN for image classification. Spatial and channel dimension of the feature maps are represented by corresponding perimeter and border thickness, respectively. Color view is recommended for the best visualization.

subsequently adopted for recalibrating the semantic context in those feature maps. Afterwards, the refined features of different resolutions are aggregated via globally spatial pooling layers followed by a concatenation module to construct the final feature vector, which is more robust than that of the conventional CNN.

Briefly, given that informative features are channel-wisely encoded from shallow to deep layers, smooth integration of such semantically-rich (high-level) details into the finer-grained (low-level) patterns by attentional extractors (which are motivated from [34]) in a Bracket-style reversed

manner is taken into account. In specific, the attachment of Channel-wisely Cross-Attentional (CCA) stream into the backbone CNN facilitates spatial representations of important DR-oriented factors (for the DR detection domain) as well as facial modalities (for the FER domain), which are comprehensively refined by semantic context of higher-level features ahead, to be comprehensively involved in the final prediction of given supervised classes. Obviously, such effective aggregation scheme of various semantic information from the multi-level feature maps in a CNN is the fundamental key for recognizing corresponding DR severity level or facial emotion label more accurately.

1.5 Major Contributions

Based on the aforementioned problem statement and objective, main contributions of this thesis are summarized as follows:

- A Bracket-shaped CNN is proposed to leverage the exploitation of middle-level feature maps by exhaustively pairing adjacent ones through attention embedded combination modules. Such routine repeats round-by-round until the final prediction map of densely enriched semantic contexts is retrieved.
- An effective approach of combining two neighboring feature maps having different resolutions is defined by adopting a cross-attentional fusion mechanism, namely CAF module. The major objective is to properly fusion semantically rich information (of the lower-resolution inputs) with finely patterned features (of the higher-resolution versions) for the outputs.
- The proposed semantic segmentation model is trained and evaluated on well-known semantic segmentation datasets including PASCAL VOC 2012 [20], CamVid [9], Cityscapes [18], and MS-COCO [64], on which the performance is competitive with well-known deep learning models in the literature.
- The proposed architecture can be flexibly manipulated by round-wise features aggregation to perform the per-pixel labeling task efficiently on dataset with heavily class-imbalancing issue such as DRIVE [80], which aims at retinal blood vessel segmentation, in comparison with the state-of-the-arts.
- The proposed concept can be extended as a variant tackling the image classification problem, wherein channel-wise attentional features of semantically-rich (high-level) information are

integrated into finely-patterned (low-level) details in a feedback-like manner, a.k.a. single-mode Bracket-structured network (sCAB-Net). Accordingly, feature maps of different scales can be amalgamated for extensively involving spatially-rich representations to the final predictions.

- The proposed Bracket-style network variant for image classification achieves impressive benchmark results on specialized domains, wherein spatially-rich factors play an important role to the decision of image label, like DR recognition (Kaggle DR Detection dataset [47]) as well as FER (RAF-DB dataset [58]).

- In overall, the above-mentioned points demonstrate a promising capability of the proposed Bracket-style network toward complete image understanding (by either semantic segmentation (pixel-level labeling) or classification (image-level labeling) performance) for further practical computer vision-oriented applications.

1.6 Thesis Organization

For convenience, chapters of this thesis are organized as follows:

- **Chapter 1 - Introduction:** this chapter firstly delivers the overview of deep learning, the core AI-based approach applied in this thesis. Then, brief description of image classification and semantic using that technique is given. Afterwards, problem statement based on the existing issues in those topics is expressed in details. Subsequently, corresponding objectives followed by major contributions of this thesis are elaborated.

- **Chapter 2 - Related Work:** since the main scope of this research is semantic image segmentation, existing works related to this problem are focused. Concretely, approaches in two different lines, i.e., *symmetrically-structured* and *asymmetrically-structured* networks, are respectively reviewed and discussed.

- **Chapter 3 - Proposed Methodology:** this chapter sequentially provides the relevant preliminaries such as overview of CNN and its constituents, corresponding modeling process, as well as configurations and hyperparameter settings for the training procedure. Next, in-depth description of the proposed Bracket-shaped CNN is given. Finally, Cross-Attentional Fusion module, the core component of the segmentation-based Bracket-shaped architecture, is characterized extensively.

- **Chapter 4 - Experiments on Natural Image Segmentation:** in this chapter, the proposed methodology is comprehensively evaluated on well-known semantic segmentation datasets to demonstrate its effectiveness for vision-related applications like object localization, autonomous driving, so on. At first, background of the benchmark datasets is introduced. Then, details of training configurations are mentioned. Afterwards, ablation study plus comparison with the state-of-the-art approaches followed by relevant analyses are intensively discussed.

- **Chapter 5 - Bracket-style Network Variant for Medical Image Segmentation:** This chapter aims at the first expandable capability of the proposed Bracket-structured deep learning model, i.e., for medical image segmentation. Particularly, domain overview, descriptions of the variant for retinal blood vessel segmentation subject to the heavily class-imbalancing issue, details of evaluated datasets, training configurations, and analyses of experimental results are in-turn delivered in this chapter.

- **Chapter 6 - Bracket-style Network Variant for Image Classification:** This chapter aims at the second expandable capability of the proposed Bracket-style concept, i.e., for effective image classification in specialized domains such as DR grading and FER. Similar to the outline of previous chapter, domain overview, descriptions of the variant, and details related to the conducted experiments in those research topics are covered.

- **Chapter 7 - Conclusions and Future Direction:** This chapter summarizes all the research outcomes comprising findings and contributions of the proposed methodology for various vision-oriented tasks like pixel-wise segmentation and image classification. Furthermore, limitations and future direction are included for the ultimate objective of complete image understanding with high effectiveness and efficiency.

Chapter 2

Related Work

2.1 Symmetrically-structured Networks

Models belonging to this group mainly follow the framework of symmetric encoder-decoder. Conceptually, backbone CNNs pre-trained on large-scale dataset for classification are often utilized as the encoder for gradually extracting from local to global features. Subsequently, the decoder is constructed in layer-wise reversed manner based on the encoder's inherent architecture to progressively integrate semantic contexts into local details in the final per-pixel segmentation map. It is obvious that involving extracted features at the encoding stage to the upsampling process at the decoder can significantly boost the pixel-wise labeling performance.

Typically, SegNet [6] made use of max pooling indices from pooling layers at the backbone VGG-Net [79] to directly locate pixels of lower-resolution feature maps in the corresponding upsampled versions. Then, the convolution layers with specific settings same as the counterparts at the encoder are subsequently applied. This strategy enables important features to be sustained throughout the network but clearly ruins the correlation between neighboring pixels.

Meanwhile, for the purpose of maintaining localization precision while being able to learn meaningful contextual information, various combination styles between corresponding feature maps at the decoder and encoder in the upsampling process were introduced. They can be either simple concatenation technique as in U-Net [75] or specialized modules as in G-FRNet [42] and GFF [59]. U-Net [75] concatenates the above-mentioned feature maps along channel dimension prior to other manipulations in the decoding process. Another impressive model named G-FRNet [42] introduced Gate Unit (based on element-wise multiplication) followed by Gated Refinement Unit (based on concatenation) to modulate encoded features for generating densely labeled output. Furthermore, a Gated Fully Fusion architecture [59] is proposed to enable the

learning of selectively important features in dense manner. These schemes are shown to yield promising performance in many benchmarks but require high footprint for training due to large depth-sized tensors. Accordingly, Tian *et al.* [83] defined an efficiently data-dependent upsampling scheme to reduce the necessity of exhaustively involving high-dimensional features in the backbone CNN during the decoding process.

On the other hand, instead of concatenation, Feature Pyramid Network [63], SwiftNetRN-18 [72] and LDN [52] introduced a Lateral Connection Module (LCM), wherein an upsampled feature map is element-wisely added to the corresponding version extracted from the encoder before being fed into learnable convolution filters. This module is executed step-by-step until forming the final prediction map. Also based on the core of pixel-wise summation, Bilinski *et al.* [8] proposed the scheme of Dense Decoder Shortcut Connections (containing Encoder Adaptation, Fusion, and Semantic Feature Generation modules) to enhance meaningful contexts captured from features at multiple scales. Similarly, RefineNet [62] further took into account additional refinement units (consisting of Residual Convolution Unit and Chained Residual Pooling) to comfort the training process and acquire global contextual information accurately.

2.2 Asymmetrically-structured Networks

Deep learning models categorized into this group contain a specialized upsampling strategy for aggregating contextual information from multiple strides without involving multi-level feature maps of the encoder.

In particular, a line of work such as ParseNet [66], HistNet [104], HolisticNet [33] incorporated an auxiliary network stream to capture global context more efficiently in addition to the main stream of semantic segmentation. Such kind of two-stream learning approach can generate the pixel-wise prediction map without local ambiguities and unexpected noises thanks to refinement from the additional stream. Besides that, attaching Recurrent Neural Network (RNN) to the pretrained CNN is an alternative way since the RNN can robustly represent the dependencies of pixel-level information with respect to global context through an evolutionary process of learning from hidden states. In concrete, RLS [54] presented the series of densely horizontal-vertical sweeping and level set method, respectively, for such evolutionary learning strategy. As a result, objects' appearance achieves better consistency while their distinction from one another becomes more explicit in the final segmentation map.

Concurrently, there is another suggestion that equipping each neuron with a larger field of view on lower-level feature maps enables the semantically rich information to be captured more effectively without sacrificing spatial resolution abundantly. Thus, recently proposed models like DeepLab [12], FSSNet [102], DenseASPP [96], PSPNet [103], and SSPP-ES [105] utilized dilated (atrous) convolution layers, which have larger receptive field but similar amount of trainable weights compared to those of the original versions. Subsequently, aggregating the extracted features learned from various dilation rates, so-called spatial pyramid mechanism, is the key factor earning impressive segmentation performance in these networks. In concrete, Chen *et al.* proposed DeepLab [12] with the utilization of Atrous Spatial Pyramid Pooling (ASPP), i.e., concurrently applied convolution having rates of *6, 12, 18, 24*, along with conditional random field to efficiently recognize objects from multi-scale viewpoints and precisely localize corresponding boundaries, respectively. Besides that, FSSNet [102] was introduced with the same mechanism but offering faster processing by the proposed blocks of factorized convolutional layers along with unpooling technique in [6]. Recently, DenseASPP [96] leveraged the scheme of ASPP by the idea of densely concatenating the coarsest feature map of the backbone network with outputs of earlier dilated convolution layers. Then the results are fed into the next layer having higher dilation rate. This iterative procedure facilitates the usage of layers having much higher dilated rate without detriment when it is required to capture enormous FOV in high-resolution images. However, given that the capability of dilated convolution remains several shortages in effectively capturing multi-scale contextual information, Meanwhile, PSPNet [103] additionally introduced Pyramid Pooling module. Particularly, average pooling layers with different stride and size settings are applied onto the final feature map learned from dilated convolution layers in the backbone CNN. Then, the retrieved outputs are concatenated before being fed into the convolution layer followed by bilinear upsampling operator for the inference of final segmentation map.

On the other hand, to avoid facing the complicated padding issue caused by the dilated convolution, a depth-wisely attentional mechanism in PAN [56], EncNet [101], BiSeNet [98], and DFN [97] is additionally exploited along with their hybrid architectures for emphasizing semantically richer information in higher-level feature maps onto responses of the lower-level counterparts. In specific, PAN [56] introduced a Global Attention Upsample module, wherein average spatial-based pooling is applied to the features acquired at high-level layers of the encoder. Subsequently, the obtained weight vectors are employed to guide semantically rich information to proper locations in the final prediction map. Furthermore, Zhang *et al.* [101] took advantages of

dilation approach as well as attention scheme to design an EncNet, which is composed of (i) a backbone CNN with dilated convolutions for extracting features and (ii) a Context Encoding for embedding semantic details back into the encoded features, to accurately classify every pixel. Besides that, a similar two-stream approach called BiSeNet [98] is introduced in the literature. It consists of cost-efficient Feature Fusion and Attention Refinement Modules in the main and auxiliary context paths, respectively, for the improvement of both accuracy and inference speed. Meanwhile, DFN [97] was proposed to enhance consistent appearance of segmented objects, for which Channel Attention Blocks were designed to re-weight feature responses of the finer-resolution maps by semantically richer context in the adjacent coarser ones. Furthermore, DANet [23] applied both spatial- and channel-based attention schemes onto the deepest-level feature map in parallel, of which the outputs are summed for subsequent learning layers followed by a final softmax classifier.

In this study, for jointly learning valuable information from the adjacent feature maps, not only the channel-wisely but also the spatially attentional blocks are adopted to seamlessly combine semantically rich context with finely patterned features while ensuring an effective training process. Notably, this work utilizes the two types of attention mechanism in crossing manner for the connections between all-level features along the decoding stage, which is different from the aforementioned DANet [23].

Chapter 3

Proposed Methodology

This chapter describes details of the proposed CAB-Net, with corresponding demonstration in Fig. 3.13, for semantic segmentation as follows. Firstly, preliminaries related to convolutional neural networks for computer vision tasks like image classification and semantic segmentation are delivered. Then, the decoding process of Bracket-shaped structure for generating the pixel-wise prediction map is elaborated. Afterwards, a thorough explanation of the proposed combination module for two adjacent feature maps of interest is given.

3.1 Preliminaries

3.1.1 Overview of Convolutional Neural Network

Convolutional Neural Networks (CNNs) have been firstly introduced for classifying image-based digits at the end of last century by LeCun *et al.* [55] and emerged tremendously in computer vision starting from this decade. The promotion of such cornerstone comes from the huge advancement in parallel computing ability of GPUs. Basically, given an input image, a CNN plays the role of manifold complex transformations that extract and select informative features for a predefined domain task like classification. Accordingly, the optimization procedure in terms of backpropagation can be executed more rapidly by GPUs in comparison with CPUs.

In general, a vanilla CNN is constructed by sequential stacks of layers which are either linearly learnable (e.g., *Convolution*, *Fully Connected* layers) or non-linear transformations (e.g., activation functions, max/average pooling). Specifically, basis component of the CNN is *Convolution* layer followed by a predefined *Non-linear Activation* function. The former component is defined as convolution filters/kernels having a pre-specified spatial size while the depth dimension identical to that of the input image or output of preceding layer. All the elements within a kernel refer to

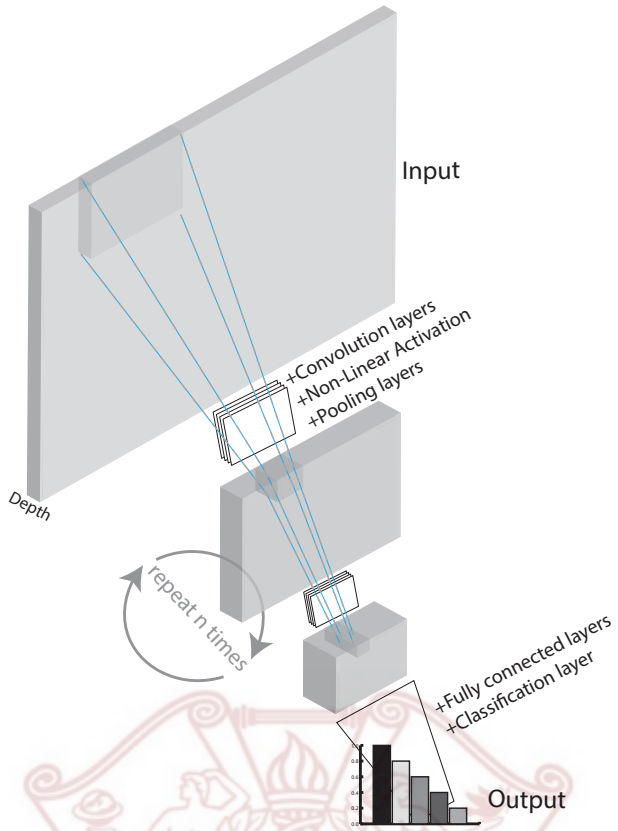


FIGURE 3.1: Basis architecture of Convolutional Neural Network.

as trainable parameters (weights). Such fixed-sized operator slides through the given input horizontally and vertically to observe its partial regions, so-called receptive field, in stepwise manner. Consequently, each entry of the output map is the weighted sum between a kernel and those in the receptive field of the input. Note that such weight-sharing characteristic explains the robustness of CNN against well-known manipulations like scaling, translations. Then, the latter component, i.e., *Non-linear Activation* function, is involved to rescale the inferred output's responses in a predefined range (for usage of *Sigmoid* or *Tanh*) or zero out the negative ones (*Rectified Linear Unit* (*ReLU*)). The key idea behind is to enhance the representational complexity of the learned features. During feedforward process of the CNN, *Max/Average Pooling* layers are usually embedded to decrease the feature resolution for easing computational burden while maintaining core representations. This layer family has similar operating principle to that of the convolution kernel but without learnable weights. Concretely, each output response is the maximum or average value of the input's entries within a pre-specified receptive field. Finally, *Fully Connected* layers are applied at the end of the CNN to exhaustively exploit high-level correlations between responses of the flatten feature maps. Afterwards, *Softmax* layer shall handle the finalized features corresponding to the given training categories in classification problem.

With the selected deep learning model such as CNN, one need to design the corresponding architecture by heuristics, which is a very important remark. In particular, number of layers in the CNN needs to be determined, it could be 50 or 100 layers. Besides that, number of convolution filters per layer need to be defined as well, it could be 64 or 128, etc. Briefly, the above-mentioned architectural designs are of problem-dependent heuristics. Clearly, each CNN model shall comprise parameters to extract input's features. Hence, the parameter initialization method needs to be decided as well. For instance, it could be arbitrarily random or follows Gaussian rule. Those model's parameters should be quantified properly by a so-called loss function. Thus, a proper loss function, which can be cross-entropy or mean square-error, should be determined. In addition, overfitting issue may occur during the training procedure. Therefore, a regularization scheme is another concern to avoid this issue by either increasing dataset size or involving regularization loss. Finally, in order to achieve the ultimate training objective, which is the optimization of model's parameters, a proper optimizer should be well-defined. For instance, existing optimizers popular in the literature are gradient descent or Adam [51].

More operational details of *Convolution*, *Non-linear Activation*, *Pooling*, *Fully Connected*, and *Softmax (Classification)* layers, which are fundamental constituent of a vanilla CNN, are respectively delivered in the following sub-sections:

Convolutional **layer**

Fundamentally, a *Convolutional* layer is used to analyze the structural details of input images through cross-correlation operations [100] as follows. In each of this layer type, an input tensor is connected to predefined convolution kernels to produce corresponding output tensor, so-called feature maps. As an example illustrated in Fig. 3.2, given an input with size of $3 \times 3 \times 1$ and the convolution kernel having size of 2×2 with associated weights (learnable parameters), the corresponding output can be attained by the following operational steps. Initially, the kernel is applied onto the 2×2 areas of the input's top-left corner. Then, its parameters are multiplied by corresponding values of the input in that receptive field, of which all the results are subsequently summed up for the inference of a scalar value at the relative position in the output tensor. In particular, at step 0 in Fig. 3.2, it can be observed that the computation of $1 \times 3 + 2 \times 2 + 6 \times 0 + 5 \times 1$ yields 12 in the output. Next, the convolution kernel slides horizontally to the right to 'see' the next receptive field in the input, from which another linear combination is performed as shown in

step 1 in Fig. 3.2 (wherein 16 is result of $2 \times 3 + 3 \times 2 + 5 \times 0 + 4 \times 1$). After reaching the right-most side of the input tensor, the kernel takes a downward stride (step size) and repeats another left-to-right sliding procedure with the aforementioned computation strategy as depicted in steps 2 and 3 in Fig. 3.2.

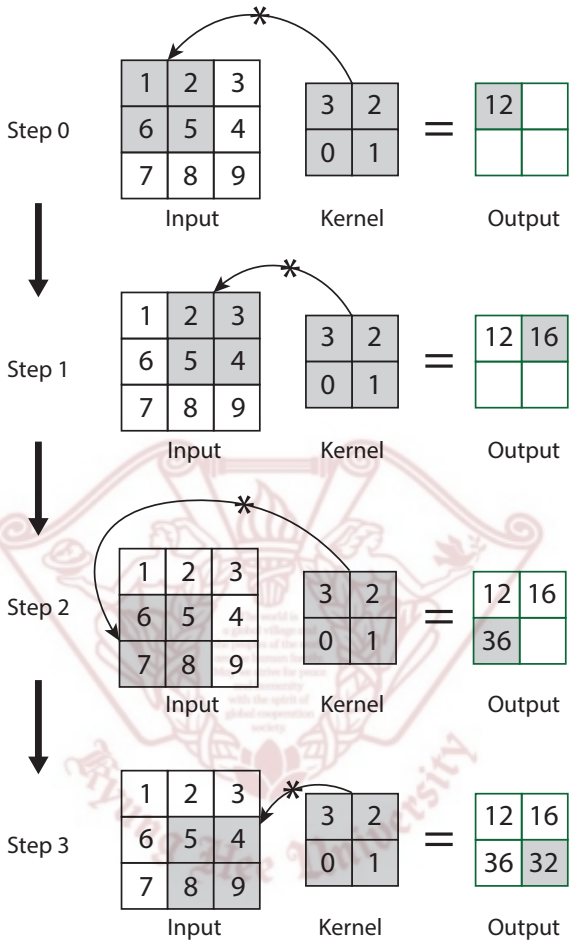


FIGURE 3.2: Example operations of convolution in a Convolutional Neural Network, wherein the output is obtained by the linear combinations between weights (parameters) of convolution kernel and corresponding values within the receptive field (area contemporarily covered by the kernel) in the input. Step 0: $1 \times 3 + 2 \times 2 + 6 \times 0 + 5 \times 1 = 12$. Step 1: $2 \times 3 + 3 \times 2 + 5 \times 0 + 4 \times 1 = 16$. Step 2: $6 \times 3 + 5 \times 2 + 7 \times 0 + 8 \times 1 = 36$. Step 3: $5 \times 3 + 4 \times 2 + 8 \times 0 + 9 \times 1 = 32$.

It can be realized that the resolution of the output is smaller than that of the input. As a CNN regularly consists of numerous *Convolution* layers, such kind of spatial loss can be accumulated and becomes significant at later part of the deep network, which may hamper the capability of image's feature extraction. Accordingly, in order to avoid the size difference issue between output and input, padding certain values around the input's boundary is the most appropriate solution to enlarge the effective spatial dimension of the input. The choice of padded values can be zero (most common), one, or replications of those locating at the input's boundary. As presented in Fig.

3.3, padding of 0's around the input's boundary is firstly executed as demonstrated by the dashed-line units. Then, similar to the process depicted in Fig. 3.2, the output is obtained by the linear combinations between weights (parameters) of the 3×3 convolution kernel and corresponding values within the receptive field (area contemporarily covered by the kernel) in the input without any losses of spatial size. Moreover, using padding allows the boundary features to be properly involved during such computation procedure.

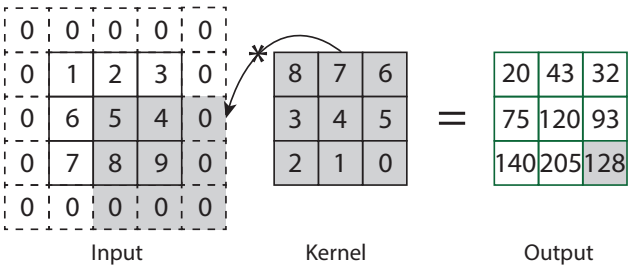


FIGURE 3.3: Example operations of convolution with zeros padding in a Convolutional Neural Network. Padding of 0's around the input's boundary is firstly executed as demonstrated by the dashed-line units. Then, similar to Fig. 3.2, the output is obtained by the linear combinations between weights (parameters) of convolution kernel and corresponding values within the receptive field (area contemporarily covered by the kernel) in the input.

Besides that, there are usual cases where output size needs to be reduced for reducing tensor footprint and improving computational efficiency while padding is in use, the convolution kernel should be shifted by larger stride (i.e., ignoring several intermediate points in the input tensor). In the above-mentioned examples, the stride is set at 1 as default. Meanwhile, as exhibited in Fig. 3.4, the stride is set at 2 in both horizontal and vertical dimensions. Consequently, the sequential steps of cross-correlation with 3×3 kernel and zero padding still can yield the output with spatial size of 2×2 in comparison with that of the input (3×3).

Formally, the spatial size of the *Convolutional* layer's output can be determined as follows

$$O = \frac{I - K + 2P}{S} + 1 \tag{3.1}$$

where O and I are spatial size (height and width) of the output and input tensors, respectively; K stands for the convolution kernel size; P indicates the number of padded rows or columns; and S is the value of stride. In general, given an input tensor of size $H_I \times W_I \times C_I$, employing N convolution kernels, each of which must be of size $K \times K \times C_I$, shall infer an output tensor of size $H_o \times W_o \times N$. Note that $C_I = 1$ and $N = 1$ in the above examples for the simplicity of demonstration. On the other hand, H_o and W_o depend on the configurations of kernel size

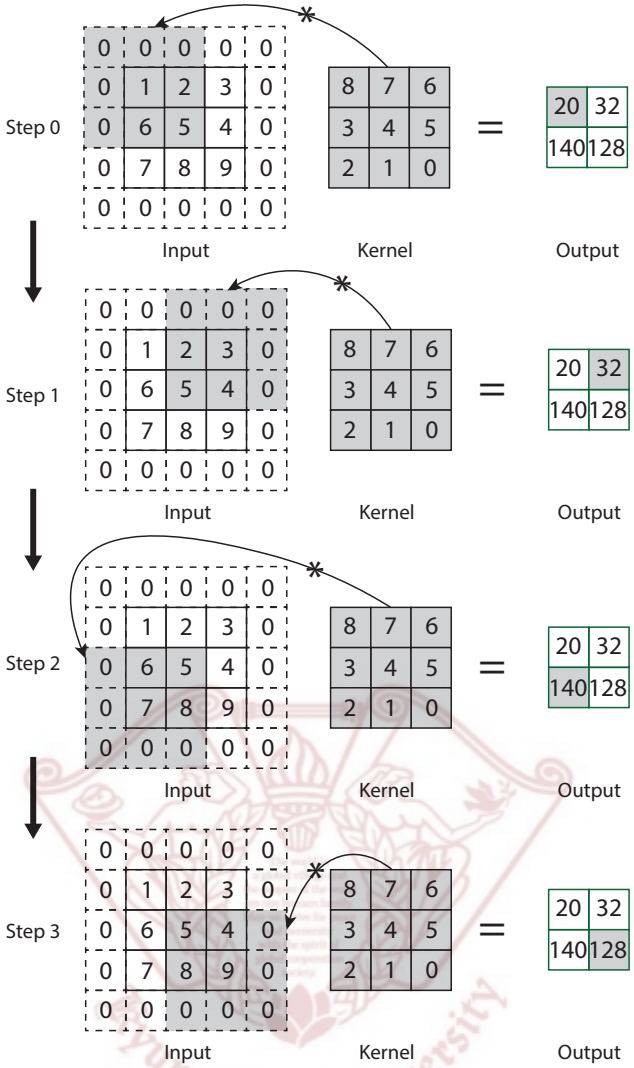


FIGURE 3.4: Example operations of convolution with zeros paddding and stride of 2 in a Convolutional Neural Network. Padding of 0's around the input's boundary is firstly executed as demonstrated by the dashed-line units. Then, similar to Fig. 3.2, the output is obtained by the linear combinations between weights (parameters) of convolution kernel and corresponding values within the receptive field (area contemporarily covered by the kernel) in the input. Remarkably, both horizontal and vertical sliding step size (stride) of the convolution kernel are 2 (instead of 1 as in Fig. 3.2 and Fig. 3.3).

K, padding P, and stride S as defined in (3.1). Furthermore, all elements in the output can be optionally added to another type of learnable parameters, so-called bias, to make the whole model generalize better.

In terms of operational principles, it can be realized such *Convolutional* layer possesses two major properties, i.e., local connectivity and translation invariance [100]. The former implies that the linear combination between the input tensor and the convolution kernel's weights occurs at patch-wise manner. The latter means that all those patches (receptive fields) inside the input tensor are processed by the same computation strategy with parameter-sharing scheme until yielding the

corresponding output feature map. Accordingly, such mechanism brings in the benefits of training cost reduction (compared with the conventional multi-layer perceptron technique) as well as robustness against variations of objects' viewpoint and scale in the considered image.

***Non-linear Activation* layer**

The operations in *Convolutional* layer are basically linear transformations. Thus, non-linear functions are highly necessary to strengthen the whole model's capability of representing very complex and diverse distributions in the image data. For that purpose, the layer of *Non-linear Activation* function is always coupled with the *Convolutional* layer to non-linearly transform a neuron (a.k.a. feature's response or element) in the feature maps of multiple extraction levels. To this end, *Rectified Linear Unit* (*ReLU*) and *Sigmoid* are the most popular type of activation function used in the CNN. Note that there are still other variants such as *Tanh*, *ELU*, to name a few, in the literature but they are beyond the scope of this thesis.

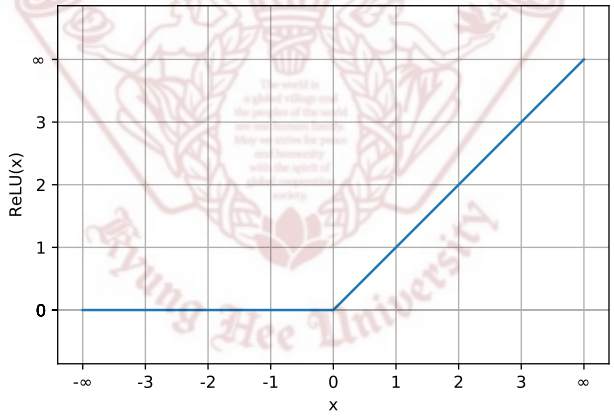


FIGURE 3.5: Graphical representation of *Rectified Linear Unit* activation function.

On the one hand, *ReLU* is widely adopted in the novel CNN thanks to its operational simplicity and effectiveness in numerous recognition tasks. Given a neuron x, the *ReLU* activation is defined as the function returning the maximum value between x and 0

$$ReLU(x) = max(x, 0) \tag{3.2}$$

In other words, *ReLU* only retains the positive elements and sets 0 for the negative counterparts. Remarkably, since the derivative of the *ReLU* function is quite straightforward, i.e., either it simply let the argument pass through or it vanishes [100], the optimization by backpropagation procedure becomes easier and can avoid the conventional issue of vanishing gradient. The graphical

representation of this activation function is shown in Fig. 3.5.

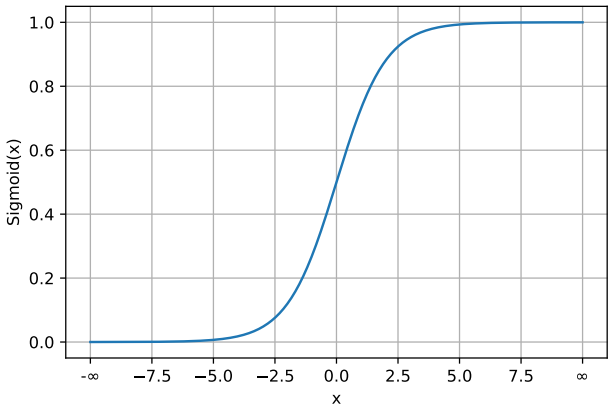


FIGURE 3.6: Graphical representation of *Sigmoid* activation function.

On the other hand, *Sigmoid*, an old-school activation function, is still an essential choice for specialized designs (e.g., attention mechanism [34]) in a CNN thanks to its expressiveness of feature's importance as well as smooth and differentiable properties for model optimization. Fundamentally, *Sigmoid* squashes the feature's entries belonging to $\mathcal{R}$ into outputs within the range of $(0, 1)$ by the following equation

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.3}$$

The graphical representation of this activation function is illustrated in Fig. 3.6, from which it can observed that the *Sigmoid* activates linearly when the input's values are around zero and saturates when the absolute of those are large. This facilitates the utilization of *Sigmoid* in vastly expressing the informative features while weakening the less meaningful ones during feedforward process of SENet [34], as well as yielding confidence scores for the final classifier in the conventional CNN.

Pooling **layer**

Normally, the neural elements in shallow layers have significant impacts on those at deeper counterparts through multiple *Convolutional* followed by *Non-linear Activation* layers of learning semantic information from various receptive fields. During such feedforward process in the CNN, it is desired that feature dimension is reduced for computational efficiency while the contextual information is effectively amalgamated at late layers. Therefore, *Pooling* layer is introduced in the literature to represent the feature maps compactly while retaining multi-level semantic details across the preceding layers. This layer family performs the similar operational principle to that

of the *Convolutional* layer, wherein a kernel (with predefined size and stride of 2) slides horizontally and then vertically on the input tensor to calculate values of corresponding output elements. However, the major difference is that the *Pooling* layer does not have any learnable parameters. Instead, its kernel continually specifies a local region in which the input's entries are employed to return either maximum or average values for the corresponding elements in the output tensor until the all of input's entries are covered. The former is defined as *Max Pooling* (see Fig. 3.7) while the latter is referred to as *Average Pooling* (see Fig. 3.8) layers. It is worth noting that the stride is always set at 2 such that the output's resolution is reduced by half (compared with that of the input tensor).

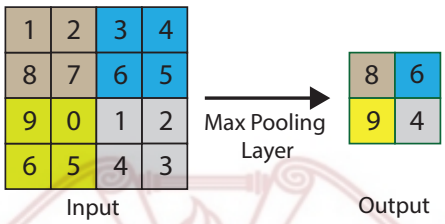


FIGURE 3.7: Graphical representation of *Max Pooling* layer with 2×2 kernel.

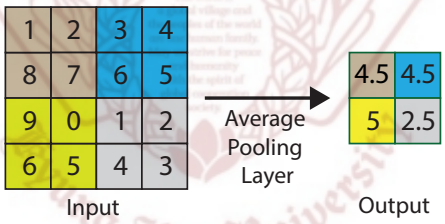


FIGURE 3.8: Graphical representation of *Average Pooling* layer with 2×2 kernel.

In general, given an input tensor of size $H_I \times W_I \times C_I$, employing C_I pooling kernels, each of which has size of $K \times K$ and stride of 2, shall infer an output tensor of size $\frac{H_I}{2} \times \frac{W_I}{2} \times C_I$. Note that $H_I = W_I = 4$, $C_I = 1$, and $K = 2$ in Fig. 3.7 and Fig. 3.8 for the simplicity of demonstration. As default, the *Pooling* layers mentioned in this section aim to manipulate the spatial dimension for mitigating the tensor footprint as well as the sensitivity of the CNN with respect to objects' location in the input image. Furthermore, the similar procedure can be applied to reduce the channel dimension of intermediate feature maps, which is called *Global Pooling*, for the advanced spatial attention mechanism in CNN [11] (more details are described in Section 3.3).

***Fully Connected* layer**

In the conventional CNN, the *Fully Connected (FC)* layer is often deployed at the end of its architecture to finalize class-oriented scores of the high-level context encoded along the depth dimension of the coarse feature maps. Fundamentally, the working principle of this layer type is similar to the *Convolutional* layer with kernel of size 1×1, wherein linear combination between the input vector (or flattened tensor) and learnable parameters (weights and bias) for inferencing the encoded output. As demonstrated in Fig. 3.9, given the input tensor of size $2 \times 2 \times 1$, of which the flattened version has size of 4×1, using the *FC* layer with hidden size of 3 shall produce the output representation of that size. In specific, the computation processes are performed as follows

$$\begin{aligned} 8 \times -1 + 6 \times -1 + 9 \times -1 + 4 \times -1 &= -27 + 20 = -7 \\ 8 \times 0 + 6 \times 0 + 9 \times 0 + 4 \times 0 &= 0 + 2 = 2 \\ 8 \times 2 + 6 \times 2 + 9 \times 2 + 4 \times 2 &= 54 + (-48) = 6 \end{aligned} \tag{3.4}$$

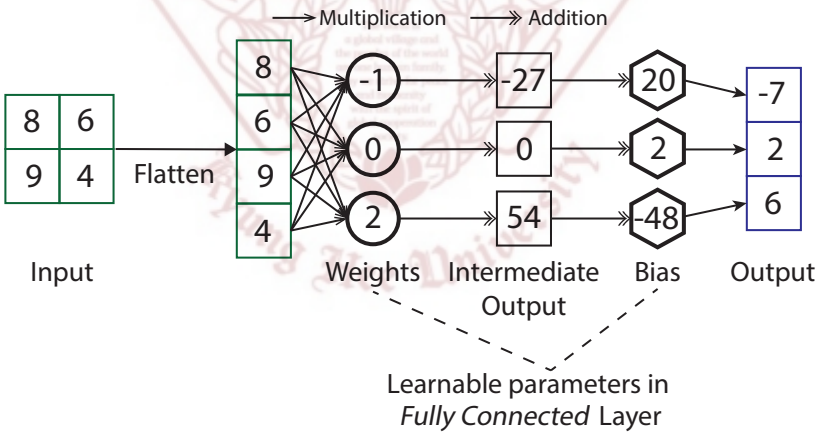


FIGURE 3.9: Example operations of *Fully Connected* layer in a Convolutional Neural Network, wherein the output is obtained by the linear combinations between trainable parameters (weights with bias) and all of the input's elements.

Generally, feeding an input of size $\mathcal{R}^n$ into the *FC* layer with trainable parameters of size m shall yield the output of size $\mathcal{R}^m$. It is worth noting that the final *FC* layer in a CNN always has size equivalent to the number of supervised labels (a.k.a. classes or categories) in certain training datasets.

***Softmax (Classification)* layer**

As the final stage of a CNN, the classifier aims to interpret the logit produced by previous learning layers in terms of probabilities, from which the label index having highest value is the ultimate classification output. For that purpose, it needs to ensure that all the prediction scores are non-negative and their sum equals to 1 (which should also be applicable to new data obtained in the future). In addition, the classifier is required to be adopted as a differentiable objective function during the training procedure so that the model can accurately estimate the computed probabilities. Notably, for all circumstances, when the recognition probability is 0.5 should indicate that half of the considered samples are classified correctly. Such kind of procedure is also called as calibration mechanism.

The *Softmax* function, which is invented by the social scientist R. Duncan Luce for solving the problem of Choice Models in 1959 [100], satisfies all the above-mentioned requirements of an appropriate *Classification* layer in the CNN. In order that the logit are interpreted as non-negative values with sum of 1 while the differentiability is still valid, each element in the logit (of which its size equivalent to the number of supervised labels) is firstly exponentiated (for abiding by the non-negativity condition) and then divided by the sum of all elements (according to the sum-of-1 condition) as follows

$$\hat{\boldsymbol{y}} = \left\{ \hat{\boldsymbol{y}}_i = Softmax(\boldsymbol{o}_i) = \frac{e^{\boldsymbol{o}_i}}{\sum_{j=1}^{C} e^{\boldsymbol{o}_j}} \mid i = 1, \ldots, C \right\} \tag{3.5}$$

where C is the number of training classes; $\hat{\boldsymbol{y}} \in \mathcal{R}^C$ stands for the output of *Softmax* layer; and $\boldsymbol{o} \in \mathcal{R}^C$ denotes the output of the last *FC* layer (a.k.a. logit). On the one hand, during the training phase, $\hat{\boldsymbol{y}}$ is utilized in a predefined objective (loss) function, which is detailedly described in the next sub-section, to quantify the compatibility of the CNN's parameters with respect to the ground-truth labels. On the other hand, during the validation/testing/real-time execution phase, the element having highest probability score in $\hat{\boldsymbol{y}}$ is used to indicate the label classified by the model.

3.1.2 Modeling of Convolutional Neural Network

Once the CNN is completely constructed, the next stage is to train it using a labeled dataset for a predefined computer vision task. The term "train" here means the procedure of exploring the

optimal CNN's parameters to perform the target task as much effectively as possible.

In order to achieve this goal, there are four essential components in a training flow of a classification CNN as manifested in Fig. 3.10: (i) Training dataset (comprising images associated with ground-truth labels); (ii) pre-built CNN; (iii) Loss function; and Regularizer, and (iv) Optimizer. Given an input image fed into the pre-constructed CNN (consisting of sequential stacks of the aforementioned *Convolutional*, *Non-linear Activation*, and *Pooling* layers), the obtained output is prediction scores (which is resulted from the last *Fully Connected* layers followed by the *Softmax* function). Then, the ground-truth label corresponding to the image and the prediction scores are employed as the input of the Loss function with regularization scheme for the calculation of loss value, which is used for assessing the quality of current CNN's parameters in terms of recognition accuracy. Note that the larger the loss value is, the worse the parameters are. Hence, an Optimizer is defined for minimizing that loss value with respect to those learnable parameters to automatically calculate more optimal values through backpropagation mechanism and update them into the CNN. Such kind of procedure repeats until a predefined stop condition (e.g., maximum training iterations/epochs, convergence threshold of loss value or classification accuracy on validation dataset) is met.

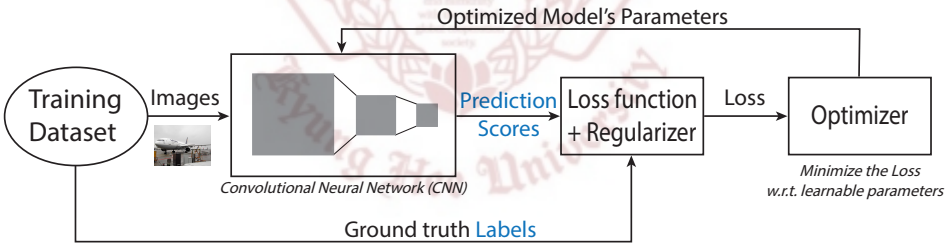


FIGURE 3.10: Abstract training flow of a CNN for image classification.

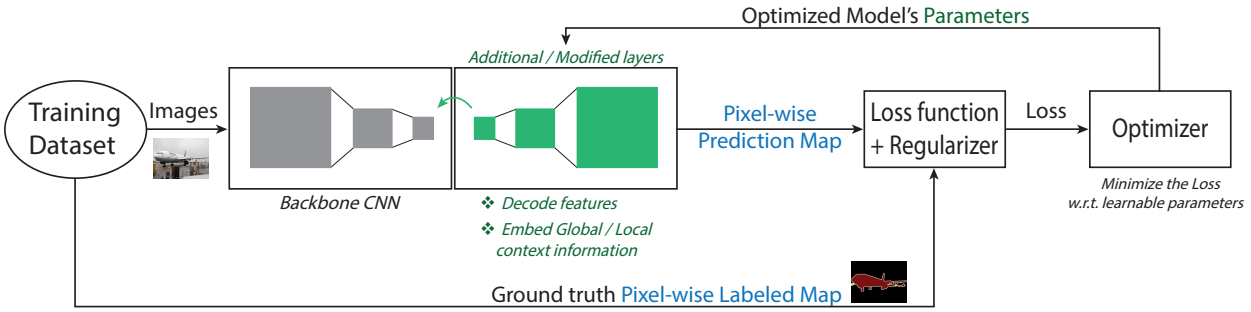


FIGURE 3.11: Abstract training flow of a CNN for semantic image segmentation.

The constituents and process of training a semantic segmentation CNN are similar to that of the classification version as illustrated in Fig. 3.11. The basis difference stays in the CNN itself, wherein additional or modified layers are attached to a backbone CNN with excluded *FC* and

Softmax layers (which is primarily designed for the classification problem) to decode features by embedding global into local contextual information. As a consequence, the ultimate output is the pixel-wise prediction map instead of the image-level recognition scores. Correspondingly, the target for the model to learn from is pixel-wise labeled map.

3.1.3 Configurations and Hyperparameter Settings for Training Process

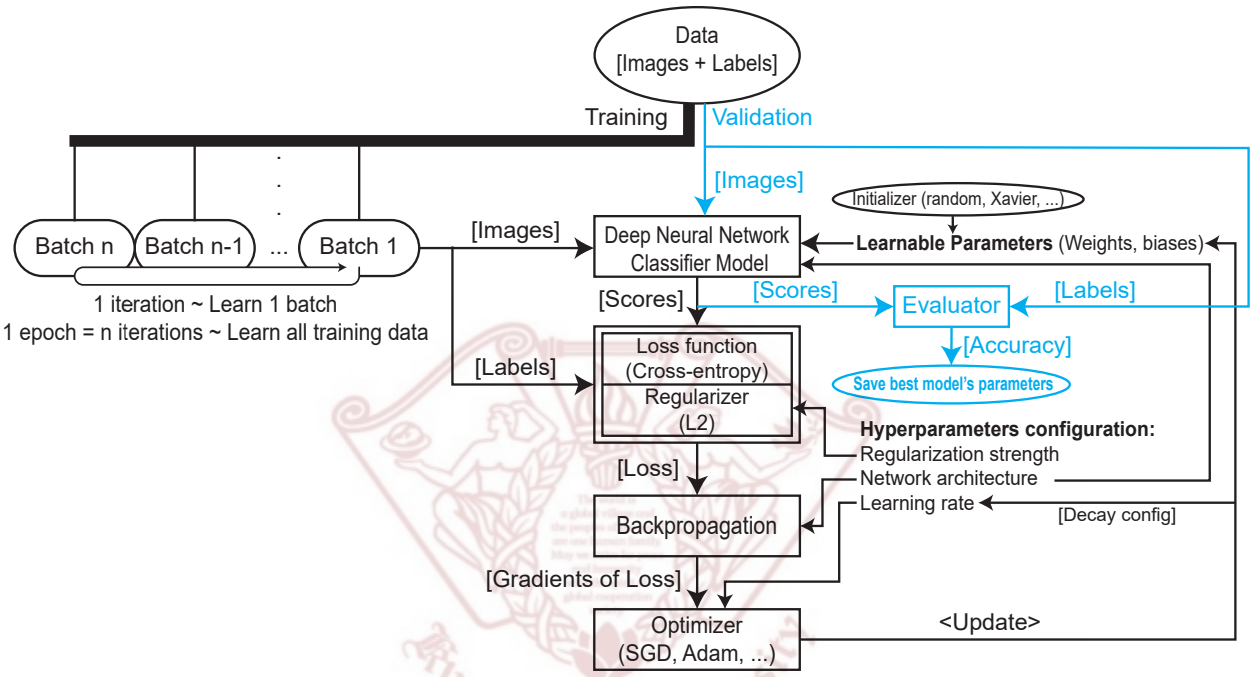


FIGURE 3.12: Configurations and hyperparameter settings for training procedure. Black arrows represent the training phase while the blue ones indicate the signify the validation phase.

The previous sub-section abstractly introduces the training flow and operational components for a CNN regarding the classification or semantic segmentation task. To this end, more details of related configurations as well as hyperparameter settings during the training agenda as demonstrated in Fig. 3.12 are delivered.

Initially, the training set is split into n batches, each of which contains a predefined number of images and corresponding ground-truth labels. It is worth noting the batch size (i.e., the number of images with labels per batch) should be configured based on the available capacity of the GPU in use. Then, each time feeding one batch into the deep model is counted as one training iteration. After n iterations, the model can learn from all training data, which marks the completion of one epoch. Therefore, the number of training epochs should be defined appropriately so that the

trained CNN can reach optimal convergence, i.e., its performance on unseen data (validation and test set) is maximized.

About the deep model, assume that its architecture was end-to-end constructed in prior, the initialization of the its learnable parameters is considerable due to the noticeable impact on network learning efficiency. There are different strategies to be adopted such as random assignment, Xavier's method [25], He's approach [30], to name a few.

After the intensively computational process in the deep learning model finishes, a Loss function is designed to receives the output scores of *Softmax* layer and ground-truth labels for evaluating the quality of contemporary predictions. According to the previous section, the *Softmax* function returns a vector $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_C]$ which indicates the condition probabilities of each predicted class. In order to verify whether the estimation is compatible with the expected outcome, the probability at which the network predicts the ground-truth class is expressed by

$$P(Y|X) = \prod_{i=1}^{C} P(\boldsymbol{y}_i|x_i) \tag{3.6}$$

which is equivalent to

$$-\log P(Y|X) = \sum_{i=1}^{C} -\log P(\boldsymbol{y}_i|x_i) = -\sum_{i=1}^{C} \boldsymbol{y}_i \log \hat{\boldsymbol{y}}_i \tag{3.7}$$

hence, maximizing $P(Y|X)$, which means minimizing $-\log P(Y|X)$, can benefit the classification performance. This leads to the definition of log-likelihood Loss function as below

$$\mathcal{L} = \sum_{m=1}^{n} l_m = \sum_{m=1}^{n} -\log P(Y_m|X_m) = \sum_{m=1}^{n} \left(-\sum_{i=1}^{C} \boldsymbol{y}_i \log \hat{\boldsymbol{y}}_i \right) \tag{3.8}$$

where n is the batch size as mentioned earlier. The Loss function is literally called Cross-entropy or Softmax Loss. Here, it is employed by considering $\hat{\boldsymbol{y}}$ as a discrete probability distribution and $\boldsymbol{y}$ as the one-hot vector. Thus, the sum of indices i will vanishes to form into one unique value. Since all $\hat{\boldsymbol{y}}_i$ are probabilities, their log is always smaller than 0. Accordingly, the Loss function is not decreased more in the case that $\boldsymbol{y}$ is correctly predicted with absolute certainty, i.e., $P(\boldsymbol{y}|x) = 1$ corresponding to the actual label. However, such kind of circumstance is usually impossible because of the following two aspects. Firstly, wrong labeling might exist in the training dataset. Secondly, the learned features in the CNN are not robust enough for the perfect discrimination

between the supervised classes. The coordination of the Loss $\mathcal{L}$ with the aforementioned *Softmax* function can be represented as follows. Replacing $\hat{\boldsymbol{y}}_i$ in (3.8) by its equivalence in (3.5) and provided that $\sum_{i=1}^{C} \boldsymbol{y}_i = 1$ (based on Softmax's definition) give

$$\begin{aligned} l_m &= -\sum_{i=1}^{C} \boldsymbol{y}_i \log \hat{\boldsymbol{y}}_i \\ &= -\sum_{i=1}^{C} \left(\boldsymbol{y}_i \log \frac{e^{\boldsymbol{o}_i}}{\sum_{j=1}^{C} e^{\boldsymbol{o}_j}} \right) \\ &= -\sum_{i=1}^{C} \left(\boldsymbol{y}_i (\boldsymbol{o}_i - \log \sum_{j=1}^{C} e^{\boldsymbol{o}_j}) \right) \\ &= -\sum_{i=1}^{C} \boldsymbol{y}_i \boldsymbol{o}_i + \sum_{i=1}^{C} \left(\boldsymbol{y}_i \log \sum_{j=1}^{C} e^{\boldsymbol{o}_j} \right) \\ &= -\sum_{i=1}^{C} \boldsymbol{y}_i \boldsymbol{o}_i + \log \sum_{j=1}^{C} e^{\boldsymbol{o}_j} \\ &= \log \sum_{j=1}^{C} e^{\boldsymbol{o}_j} - \sum_{i=1}^{C} \boldsymbol{y}_i \boldsymbol{o}_i \end{aligned} \tag{3.9}$$

Subsequently, the partial derivative of l_m with respect to the logit o_i is represented by

$$\frac{\partial l_m}{\partial \boldsymbol{o}_i} = \frac{e^{\boldsymbol{o}_i}}{\sum_{j=1}^{C} e^{\boldsymbol{o}_j}} - \boldsymbol{y}_i = Softmax(\boldsymbol{o}_i) - \boldsymbol{y}_i = P(\boldsymbol{y} = i|x) - \boldsymbol{y}_i \tag{3.10}$$

It can be realized that the derivative is equivalent to the difference between the probability of the actual label predicted by the model (which is signified by the softmax function) and that actual (ground-truth) label itself (which is encoded as one-hot vector format). Notably, this Cross-entropy (or Softmax) Loss can be considered as the most popular objective function in the classification problem using deep learning.

Furthermore, in the conventional training system, the total Loss function $\mathcal{L}$ also includes an L2-norm regularization term for combating the overfitting issue. Note that this challenge is referred to as the poor generalization capability of the CNN, which is basically caused by the fact the CNN is too complex while the number of training data is quite small. Besides that, since the complexity of a model is can be assessed by its squared norm of the trainable weights $||\mathbf{W}||^2$, which can be

minimized together with the Loss $\mathcal{L}$ in (3.8). Consequently, we have

$$\mathcal{L}_{total} = \mathcal{L} + \lambda \mathcal{L}_{Reg} = \mathcal{L} + \lambda ||\mathbf{W}||^2 \tag{3.11}$$

where the hyperparameter λ denotes the regularization strength and should be non-negative. The higher value the λ is set, the stronger constraint is applied onto the amplitude of the squared norm. Remarkably, the fact that L2-norm can significantly penalize the large components in the weights set is the most prominent reason for its utilization against the overfitting problem. This leads to the circumstance where the network is directed to learn uniformly distributed weights for extracting deep features throughout its architecture. In addition, according to the objective of minimizing the weights' values to approach 0 as mentioned before, L2-norm is sometimes referred to as weight decay strategy, in which the Optimizer tries to not only figure out optimal parameters' values but also continuously decay the weights to simplify the CNN's complexity during the training progress.

Intuitively, gradient of a function with respect to certain parameters (variables) signifies the orientation along which the increasing rate of those function's parameters is largest. Accordingly, negative gradient of the considered function indicates the opposite direction, i.e. maximum decreasing rate of the parameters. Hence, gradient can be used as an indicator of minimizing the $\mathcal{L}_{total}$. In neural network, backpropagation is the unique technique to compute the gradient with respect to the learnable parameters. Briefly, this methodology performs gradient computation through the network in reverse route, i.e., from the last to the first layer of the architecture, using the chain rule in calculus. The intermediate variables (i.e., partial derivatives) required in the procedure of calculating the gradient with respect to the parameters of interest is exhaustively employed. For instance, based on the feedforward equations (3.8) and (3.11), we have

$$\frac{\partial \mathcal{L}_{total}}{\partial \mathbf{W}_p} = \frac{\partial \mathcal{L}_{total}}{\partial \mathcal{L}} \frac{\partial \mathcal{L}}{\partial \mathbf{W}_p} + \frac{\partial \mathcal{L}_{total}}{\partial \mathcal{L}_{Reg}} \frac{\partial \mathcal{L}_{Reg}}{\partial \mathbf{W}_p} = \frac{\partial \mathcal{L}}{\partial \mathbf{W}_p} + 2\lambda \mathbf{W}_p \tag{3.12}$$

Likewise, along with the formulation computed in (3.10), the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_p}$ can be then be expanded based on the network architecture as follows

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_p} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{o}} \frac{\partial \boldsymbol{o}}{\partial \mathbf{W}_p} = (Softmax(\boldsymbol{o}) - \boldsymbol{y}) \frac{\partial \boldsymbol{o}}{\partial \mathbf{W}_{p+1}} \frac{\partial \mathbf{W}_{p+1}}{\partial \mathbf{W}_p} = \dots \tag{3.13}$$

where $\mathbf{W}_p$ is referred to as the learnable weights at layer p. Note that the bias parameters $\mathbf{B}$ are not involved in the above equations for simplicity of demonstration.

Algorithm 1: Modeling process for a deep learning architecture

Input : training and validation Images with corresponding Ground-truth labels
Output: optimal Model's parameters
Hyperparameters:
Number of training epochs E;
Batch size;
Regularization strength λ;
Learning rate α;
Learning rate decay schedule;
Miscellaneous:
Number of training iterations per epoch: n = Number of training Images / Batch size;
Number of validation iterations v = Number of validation Images / Batch size;
Model architecture ;
Loss function $\mathcal{L}_{total}$;
Optimizer;
begin
    best accuracy = 0 ;
    Initialize Model's parameters;
    for $e = 0 : E$ **do**
        %*Training phase*
        Perform Learning rate decay schedule at e^{th} epoch;
        for $i = 0 : n$ **do**
            logits = Model(i^{th} training Image batch);
            loss = $\mathcal{L}_{total}$(logits, Groudn-truth labels, λ, current Model's parameters);
            gradient of loss = Backpropagation(loss);
            updated parameters = Optimizer(gradient of loss, α);
            Model $\leftarrow$ updated parameters;
        end
        %*Validation phase*
        validation accuracy = 0;
        for $i = 0 : v$ **do**
            logits = Model(validation Image batch);
            predicted labels = argmax(logits);
            validation accuracy += Evaluator(predicted labels, Groundt-truth labels);
        end
        average validation accuracy = *validation accuracy* / v;
        if *average validation accuracy* > *best accuracy* **then**
            best accuracy = *average validation accuracy*;
            Save current optimal Model's parameters;
        end
    end
end

After the backpropagation activity completes, the predefined Optimizer is activated to find the optimal parameters (based on which the Loss function $\mathcal{L}_{total}$ reaches global minimum) and

update them in the deep learning model. However, it is worth noting that global minimum is nearly impossible to attain for neural network family because its complex transformations formed into a non-convex function. Instead, the deep learning architecture is usually optimized at local minimum, at which the final performance is evaluated by the recognition accuracy on validation phase (as expressed by blue arrows in Fig. 3.12). Generally, te core concept of exploring the highly qualified parameters follows the mechanism of Stochastic Gradient Descent (SGD) as below

$$\mathbf{W}_p^{new} = \mathbf{W}_p^{current} - \alpha \frac{\partial \mathcal{L}_{total}}{\partial \mathbf{W}_p^{current}} \tag{3.14}$$

where the minus sign is used to turn the gradient into negative for the purpose of minimization as mentioned earlier; α means the learning rate, which defines how magnitude of step size that the Optimizer should take to search for the newer parameter set that brings in better recognition performance. Notably, it is preferred that the learning rate be gradually reduced from the beginning to the end of the training process. The higher values help the network quickly scrutinize the most reasonable parameters while avoiding unexpected local minimum. Meanwhile, the lower counterparts allows the network to carefully inspect the best parameter state when the learning progress becomes much more stable and saturated at later epochs. Otherwise, the training progress shall last much longer to reach convergence condition and the trend of loss minimization tends to be error-prone and unstable. Thus, in practical use case, a heuristic configuration of hyperparameter called learning rate decay is additionally involved for the above-mentioned purpose of gaining high training efficiency. Recently, there are novel optimization algorithms such as Adam [51] which can automatically decide various learning rates for parameters at different layers in the deep model during the course of training. In summary, the generic process of modeling a deep learning architecture is presented in Algorithm 1 as another comprehensive perspective.

3.2 Bracket-shaped Convolutional Neural Network

It is worth noting that the proposed Bracket-shaped decoder can be easily fitted to any classification-based CNNs. In this work, ResNet-101 [29] pretrained with ImageNet dataset [76] is employed as the default backbone CNN (encoder) of the proposed architecture in Fig. 3.13 to extract meaningful features from the inputs. Accordingly, four encoded feature maps of specialized convolution blocks are taken into account for the Bracket-shaped decoder. Note that spatial resolution of these

features is reduced by half (i.e., they have strides of 4, 8, 16, and 32, respectively) while their channel dimension gets significantly deeper after each convolution block along the feedforward process. To be convenient, the selected features are respectively named *convmap-1* (with spatial size having stride of 4 compared to that of the original input and depth of d1), *convmap-2* (8 and d2), *convmap-3* (16 and d3), and *convmap-4* (32 and d4) as manifested in Fig. 3.13.

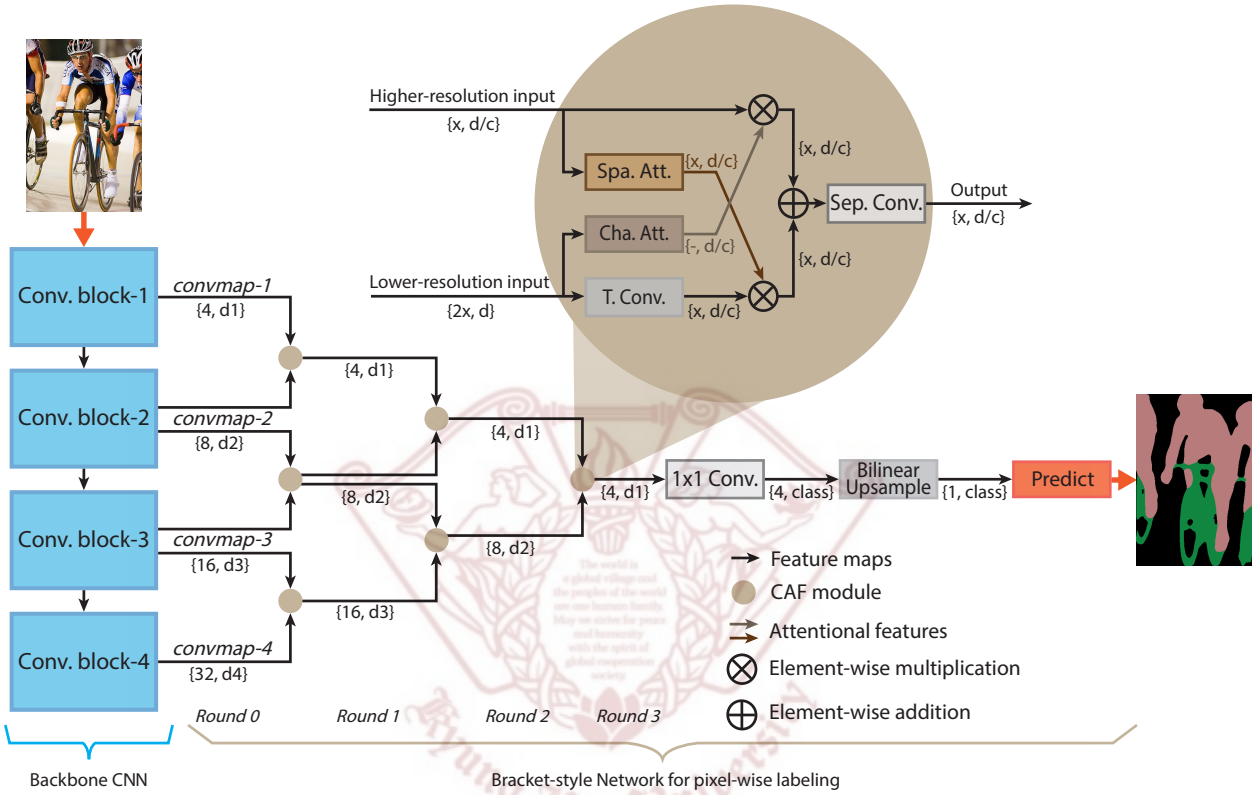


FIGURE 3.13: Architecture of the proposed CAB-Net. Given an input image fed into the backbone CNN containing series of predefined convolution blocks, final outputs of these blocks have strides of 4, 8, 16, and 32, respectively. Subsequently, these chosen feature maps (namely *convmap-1*, *convmap-2*, *convmap-3*, *convmap-4*) are utilized in the decoding process for pixel-wise labeling. In brief, these fine-to-coarse feature maps (represented by black arrows) are densely combined via the Cross-Attentional Fusion modules to produce outputs, which continuously pass through the same procedure until one final prediction map is retrieved. As for the obtained segmentation map, every pixel is assigned an object class within the predefined number of training classes. Since every inferred feature map fuses with its adjacent finer-resolution map at each round and the total number of feature maps decreases by one round-by-round, such process is named Bracket-shaped network. Note that the symbol $\{x, d\}$ attached to each arrow indicates the corresponding feature map having stride of x (i.e., its spatial dimension is $1/x$ as large as that of the input image) and d channels. Meanwhile, x = - (dash) means that the spatial size equals to 1×1. Besides that, 'T. Conv.' and 'Sep. Conv.' stand for *Transpose* and *Separable Convolution* layer while 'Spa. Att.' and 'Cha. Att.' represent Spatially and Channel-wisely Attentional blocks, respectively. Color view is recommended for the best visualization.

Next, every of those feature maps, except for the finest-resolution one (i.e., *convmap-1*), combines with the adjacent higher-resolution version through the CAF module to generate an output

having same dimension as that of the latter. In other words, the utilization of all the middle-level feature maps (e.g., *convmap-2* and *convmap-3*) is leveraged since each one simultaneously plays two roles, i.e., (i) integrating global context at a certain level to the final prediction map by upsampling itself, and (ii) refining semantically richer information of upsampled version of the adjacent coarser-resolution map by embedding its finer patterned features. Hence, it is clear that given n encoded feature maps chosen from the backbone CNN, such connection style infers $n-1$ outputs at the first round of the proposed Bracket-shaped decoder. Subsequently, as such routine iterates, total number of semantic feature maps decreases by one while average spatial dimension increases round-by-round until the final pixel-wise prediction map is retrieved.

In specific, let $\mathbf{F}_i^r$ be the i^{th} feature map at r^{th} round, where $i = 1, \ldots, n-r$ and $r = 0, \ldots, n-1$. Note that $i = 1$ indicates the feature map having highest resolution and $i = n-r$ corresponds to the lowest. Accordingly, $\mathbf{F}_1^0$ refers to as *convmap-1* and $\mathbf{F}_4^0$ corresponds to *convmap-4* at the initial 0^{th} round as presented in Fig. 3.13. Then, the feature maps of next rounds are continuously determined by

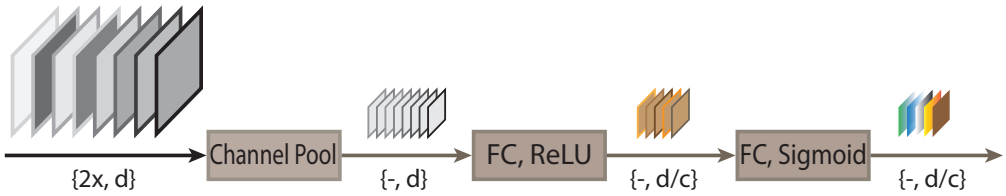
$$\mathbf{F}_i^r = \mathcal{C}(\mathbf{F}_i^{r-1}, \mathbf{F}_{i+1}^{r-1}), \qquad r \geq 1 \tag{3.15}$$

where $\mathcal{C}(.)$ is the CAF module, which is fully depicted at section 3.3. It is obvious that until the $(n-1)^{\text{th}}$ round, the final prediction map containing finely patterned features fulfilled by semantically rich context is acquired. Since every decoded feature map fuses with its adjacent finer-resolution map at each round and the total number of feature maps decreases by one round-by-round, such process is named Bracket-shaped network.

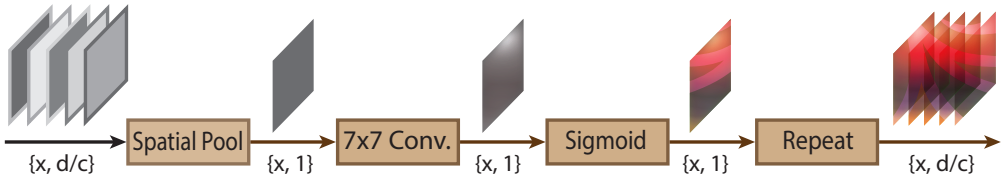
Fundamentally, there are two apparent advantages of using the Bracket structure: (i) missing or ambiguous details are suppressed significantly since every upsampled feature map is always refined by the equivalent-sized version of finer-grained information; and (ii) semantically rich information is densely enhanced in the final per-pixel segmentation map because such upsampling plus dense mixture strategy is applied for all fine-to-coarse feature maps at all rounds during the decoding stage.

3.3 Cross-Attentional Fusion Module

Obviously, the ultimate purpose of the upsampling process in a semantic segmentation architecture is to ensure that visual details in the upsampled version of certain coarse-resolution feature



(A) Channel-wisely Attentional (Cha. Att.) Block



(B) Spatially Attentional (Spa. Att.) Block

FIGURE 3.14: Details of operators in attentional schemes, i.e., 'Spa. Att.' and 'Cha. Att.'. Note that 'FC' stands for *Fully Connected* layer and '7x7 Conv.' indicates one *Convolution* layer having kernel size of 7x7.

map are capable of bearing the semantic information reasonably. To achieve this, refining local ambiguities appearing in the upsampled ones by effectively involving well-representational knowledge in the corresponding encoder's feature maps plays a critical role in many model designs.

Existing work introduced various refinement styles for the upsampled high-level features, which range from simple channel-wise concatenation [75, 83] to more complicated lateral connection components [8, 42, 52, 59, 62] or attention-based blocks [56, 97, 101]. However, to efficiently coordinate with the capability of the proposed Bracket-structured decoder, the CAF module built upon the attentional mechanism (inspired from [11, 34]) followed by *Separable Convolution (Sep. Conv.)* layers [17] is proposed as illustrated in the copper circle in Fig. 3.13. Concretely, each CAF unit comprehensively carries out contextual information from the two inputs of different resolution in twofold: (i) Channel-wisely Attentional (Cha. Att.) block, which depth-wisely re-weights the lower-level features of the higher-resolution input by using semantically richer features of the lower-resolution counterpart; and (ii) Spatially Attentional (Spa. Att.) block, which spatially re-calibrates features of the upsampled lower-resolution input by utilizing finer patterns of the higher-resolution one. As a consequence, fusioning the acquired cross-attentional information can infer fruitful feature maps for the dense prediction.

The first block (a.k.a., Cha. Att.) is executed according to the fact that the coarser and deeper feature map possesses much more informative context along the depth dimension than the finer and shallower one does. Therefore, it is beneficial to the final performance when conducting the

impact of that channel-wise semantic information on the fine-grained features in feedback-like manner. To address this, a depth-wise calibration strategy inspired from the attention mechanism in [34] is employed as demonstrated in Fig. 3.14a. It is worth noting that the DFN [97] also adopts such scheme. Specifically, all feature responses are re-weighted through a step of cross-channel learning on the global pooling information, which is acquired from the considered feature map itself [34] or that concatenated with the adjacent scale [97], a.k.a. self-attention. Differently, the proposed approach collects informative attributes across channels of the lower-resolution input only in order to depth-wisely enhance corresponding responses of the higher-resolution one, a.k.a. cross-attention. As shown in Fig. 3.14a, each channel of the coarser-resolution input, of which the spatial and depth size are $\frac{1}{2^x}$ as large as that of the original image and d respectively, is averaged spatially to form a vector having length of d. Accordingly, this vector, namely $\boldsymbol{g} \in \mathrm{R}^{\mathrm{d}}$, compactly carries reasonable information in channel-wise manner as follows

$$\boldsymbol{g} = \left[g_1(\mathbf{F}_{i+1}^{r-1}), \ldots, g_d(\mathbf{F}_{i+1}^{r-1})\right]^T \tag{3.16}$$

where $g_d(.)$ is the *Channel Pool* operation taking place on d^{th} channel of a considered feature map f, of which the corresponding formulation is

$$g_d(\mathbf{F}_{i+1}^{r-1}) = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{F}_{i+1\,h,w,d}^{r-1} \tag{3.17}$$

where (h, w) indicates pixel coordinates of the considered feature map $\mathbf{F}_{i+1}^{r-1}$ having spatial resolution of $H \times W$. Consequently, every channel of the lower-resolution input has its own representative response in the d-length vector $\boldsymbol{g}$. Next, to correspondingly express the relative importance degree of each channel onto that of the higher-resolution input, the vector $\boldsymbol{g}$ is firstly filtered by two *Fully Connected (FC)* layers with *ReLU* activation in the middle so as to learn cross-channel relationships. Notably, the size of applied hidden layers is set to be identical to the number of the higher-resolution input's channels. These learning operations are equivalent to the following equation

$$\boldsymbol{g}_{ca} = \mathbf{W}_{fc_2}\left(ReLU(\mathbf{W}_{fc_1}\boldsymbol{g} + \mathbf{B}_{fc_1})\right) + \mathbf{B}_{fc_2} \tag{3.18}$$

where $\{\mathbf{W}_{fc_1} \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}} \times \mathrm{d}}, \mathbf{B}_{\mathrm{fc}_1} \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}}}\}$ and $\{\mathbf{W}_{fc_2} \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}} \times \frac{\mathrm{d}}{\mathrm{c}}}, \mathbf{B}_{\mathrm{fc}_2} \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}}}\}$ are learnable parameters of the first and second *FC* layers, respectively, and $\boldsymbol{g}_{ca}$ is the yielded channel-wise attention feature vector of $\frac{d}{c}$ length. Then, the *Sigmoid* activation $\sigma(.)$ is utilized to rescale values of elements in the vector

g_{ca} within the range from 0 to 1. Subsequently, the resulting channel-wisely attentional features are used to modify the responses of the higher-resolution input $\mathbf{F}_i^{r-1} \in \mathrm{R}^{2\mathrm{H}\times 2\mathrm{W}\times \frac{\mathrm{d}}{\mathrm{c}}}$ in depth-wise manner as below

$$\mathbf{F}_{i_{ca}}^{r-1} = \left\{ \mathbf{F}_i^{r-1}{}_{:,:,\delta} \otimes \sigma(g_{ca})_\delta \mid \delta = 1, \ldots, \frac{d}{c} \right\} \tag{3.19}$$

where $\otimes$ symbolizes the element-wise multiplication and $\mathbf{F}_{i_{ca}}^{r-1} \in \mathrm{R}^{2\mathrm{H}\times 2\mathrm{W}\times \frac{\mathrm{d}}{\mathrm{c}}}$ is the channel-wisely attentional version of $\mathbf{F}_i^{r-1}$.

For the second block, it can be realized that higher-resolution feature maps possess finer patterns spatially is apparently profitable for the refinement of local details in the upsampled version of the lower-resolution ones. Therefore, important spatial features of the finer-resolution input are integrated into the upsampled partner in the CAF module through a spatially-attentional block exhibited in Fig. 3.14b. Different from [11], an early layer of *FC* and *ReLU* is additionally involved for acquiring the underlying attentional features more smoothly in spatial manner. In particular, the finer-resolution input $\mathbf{F}_i^{r-1} \in \mathrm{R}^{2\mathrm{H}\times 2\mathrm{W}\times \frac{\mathrm{d}}{\mathrm{c}}}$ is fed into a *Spatial Pool* operation, in which responses at every pixel (h, w) are averaged across channel dimension as follows

$$\mathbf{F}_{isp}^{r-1}{}_{h,w} = \frac{c}{d} \sum_{z=1}^{\frac{d}{c}} \mathbf{F}_i^{r-1}{}_{h,w,z} \tag{3.20}$$

wherein $\mathbf{F}_{isp}^{r-1} \in \mathrm{R}^{2\mathrm{H}\times 2\mathrm{W}\times 1}$ is the corresponding output of this operation. Subsequently, one trainable *Convolution* layer having kernel size of 7×7 with padding of 3, namely $\mathbf{W}_{7\times 7}$, followed by the *Sigmoid* activation $\sigma(.)$ is adopted to quantify the locally spatial dependencies as below formulation

$$\mathbf{F}_{i_{s77}}^{r-1} = \sigma(\mathbf{W}_{7\times 7} * \mathbf{F}_{isp}^{r-1}) \tag{3.21}$$

where $*$ and $\mathbf{W}_{7\times 7} \in \mathrm{R}^{1\times 7\times 7\times 1}$ represent the convolution operator and learnable parameters of the above-mentioned kernel, respectively, and $\mathbf{F}_{i_{s77}}^{r-1} \in \mathrm{R}^{2\mathrm{H}\times 2\mathrm{W}\times 1}$ is defined as the spatially attentional features. It is worth noting that this one-channel map is then repeated by $\frac{d}{c}$ times to be same depth size as that of the higher-resolution input of the CAF module. Simultaneously, the lower-resolution input $\mathbf{F}_{i+1}^{r-1}$ is upsampled into $\mathbf{F}_{i_u}^{r-1}$ using the *Transpose Convolution (T. Conv.)* layer having stride of 2 and the number of filters identical to channel dimension of the higher-resolution input. Notably, as shown in Fig. 3.13, given d as the channel size of the lower-resolution input,

then that of the higher-resolution one is c times smaller. The upsampling operation can be expressed as below

$$\mathbf{F}_{i_u}^{r-1} = upsample(\mathbf{F}_{i+1}^{r-1}) = \mathbf{W}_u *^u \mathbf{F}_{i+1}^{r-1} + \mathbf{B}_u \tag{3.22}$$

where $*^u$ is fractionally-strided convolution operation, $\mathbf{W}_u \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}} \times 3 \times 3 \times \mathrm{d}}$ corresponds to trainable weights in $\frac{d}{c}$ transposed convolution filters having size of $3 \times 3 \times d$, and $\mathbf{B}_u \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}}}$ stands for trainable biases. Finally, from (3.21) and (3.22), the spatially attentional version, denoted as $\mathbf{F}_{i_{sa}}^{r-1}$, of the originally upsampled map $\mathbf{F}_{i_u}^{r-1} \in \mathrm{R}^{2\mathrm{H} \times 2\mathrm{W} \times \frac{\mathrm{d}}{\mathrm{c}}}$ is obtained by multiple operations of Hadamard product as follows

$$\mathbf{F}_{i_{sa}}^{r-1} = \left\{ \mathbf{F}_{i_u}^{r-1}{}_{:,:,\delta} \otimes \mathbf{F}_{i_{s77}}^{r-1} \mid \delta = 1, \ldots, \frac{d}{c} \right\} \tag{3.23}$$

To this end, both semantically richer information and finely patterned features are exhaustively exploited in cross-attentional manner, i.e., $\mathbf{F}_{i_{ca}}^{r-1}$ and $\mathbf{F}_{i_{sa}}^{r-1}$, respectively. The next step is to integrate them by a simple pixel-wise addition scheme, of which the total result is continuously fed into the *Sep. Conv.* as follows

$$\mathbf{F}_i^r = \mathbf{W}_{sc} * ReLU(\mathbf{F}_{i_{ca}}^{r-1} \oplus \mathbf{F}_{i_{sa}}^{r-1}) \tag{3.24}$$

where $\oplus$ signifies the element-wise addition, $\mathbf{W}_{sc} = \{\mathbf{W}_{df} \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}} \times 3 \times 3}, \mathbf{W}_{\mathrm{pf}} \in \mathrm{R}^{\frac{\mathrm{d}}{\mathrm{c}} \times 1 \times 1 \times \frac{\mathrm{d}}{\mathrm{c}}}\}$ denotes the sequential execution of $\frac{d}{c}$ depth-wise convolution filters with 3×3 size and $\frac{d}{c}$ point-wise convolution filters with $1 \times 1 \times \frac{d}{c}$ size. It is also worth noting that the *Sep. Conv.* layer defined in this CAF module includes three consecutive operations, i.e., *ReLU* activation, *Sep. Conv.*, and *Batch Normalization* layer [41] (which was not shown in (3.24) for simplicity). Obviously, compared to using normal 3×3 convolution, such kind of filter can reduce the number of trainable parameters per layer from $\frac{d}{c} \times 3 \times 3 \times \frac{d}{c}$ to $\frac{d}{c}(3 \times 3 + \frac{d}{c})$ while effectively maintaining the capability of shrinking unexpected artifacts appearing caused from previous upsampling steps in such decoding process. Remarkably, it is also investigated that additionally taking the fusion in (3.24) with $\mathbf{F}_i^{r-1}$ and $\mathbf{F}_{i_u}^{r-1}$ is not necessary due to trivial performance improvement while being subject to more computation.

In a nutshell, instead of simply adding upsampled version of the coarser input to the naive finer-resolution one (which may hinder the precise integration of semantically rich features into spatial dimension), taking into account the proposed CAF module can improve the efficiency of context acquirement and corresponding pixel-wise localization.

Chapter 4

Experiments on Natural Image Segmentation

In this chapter, the proposed CAB-Net is intensively experimented on PASCAL VOC 2012 [20], CamVid [9], Cityscapes [18], and MS-COCO [64] datasets to show its effectiveness for applications of vision-based object localization and autonomous driving, to name a few, in the industry. Particularly, the benchmark datasets are introduced at first, then provide details of training configurations, and finally present ablation study on the proposed architecture as well as subsequent analyses of on-hand experimental results.

4.1 Benchmark Datasets

4.1.1 PASCAL VOC 2012 [20]

The dataset name is the abbreviation of "Pattern Analysis, Statistical Modelling and Computational Learning". This dataset aims to represent 20 semantic object categories common in real world (i.e., groups of person, animal, vehicle and indoor context). Originally, there are 1,464 training, 1,449 validation, and 1,456 testing images of various sizes in this challenge. It is noted that 513×513 is set as spatial size of the CAB-Net's inputs. Moreover, the training process follows the procedure of [6, 67] wherein additional annotations from Semantic Boundaries Dataset [28] are included for increasing the total number of training images to 10,582. Afterwards, the proposed CAB-Net is further fine-tuned with the original training plus validation set before being benchmarked by the test set on a designated testing server.

4.1.2 CamVid [9]

The name of this dataset stands for "Cambridge-driving Labeled Video Database". It is the collection of various road scenes recorded in 10 minutes by a dashboard camera, which acts as the eyes of an autonomous car. Accordingly, all 701 obtained 720×960 video frames are pixel-wisely labeled given 32 semantic categories. However, to be comparable with previous work, the conducted experiment uses the split of 367 training, 101 validation and 233 testing images with 12 finalized ground-truth labels (consisting of building, tree, sky, car, sign-symbol, road, pedestrian, fence, column-pole, side-walk, bicyclist and the background) to evaluate the proposed model. Besides that, all of those images are downsampled to 360×480 at first.

4.1.3 Cityscapes [18]

This dataset also represents things that an autonomous car should 'see' for understanding urban street scenes semantically. It offers a large pool of 5,000 and 20,000 1024×2048 images with fine and coarse annotations, respectively, corresponding to 19 semantic classes through a 50-city itinerary. In this work, only the set of fine annotations with 2,975 training, 500 validating, and 1,525 testing images is utilized for the evaluation of the proposed CAB-Net. Note that crop size of 768×768 is used to train the proposed deep network for ensuring reasonable mini-batch size. Correspondingly, there are 19 labels applied for such semantic segmentation problem, which include road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, and bicycle.

4.1.4 MS-COCO [64]

The name of this benchmark dataset stands for "Microsoft Common Object in Context". It is a very large-scale dataset with approximately 100,000 images containing reasonable pixel-wise representations of the 20 semantic categories defined in the PASCAL VOC 2012 dataset. Specifically, only the image having at least 1000 pixels annotated as one of the predefined 20 classes is valid for the experiments using this dataset. As a consequence, there are totally 95,737 training and 4,043 validation images chosen for evaluating the 20-class semantic segmentation performance of the proposed deep architecture. Note that the default input image resolution is set at 513×513. Similar to the strategy adopted for learning PASCAL VOC 2012 dataset, the model initially trained

with this MS-COCO dataset shall be continually fine-tuned with the original training plus validation set of PASCAL VOC 2012 prior the final evaluation with the test set on the designated server.

4.2 Training Configurations

In this work, the proposed CAB-Net is trained using PyTorch framework [73] with two NVIDIA GTX 1080Ti GPUs. The training images are augmented by following strategies: scaling with random factor in {0.5, 0.75, 1.0, 1.25, 1.5, 1.75}; random cropping to pre-specified size (513×513 for PASCAL VOC 2012, 360×480 for CamVid, and 768×768 for Cityscapes); randomly horizontal flipping; and channel-wise normalization with zero mean and standard deviation of one. Besides that, random Gaussian noise with random standard deviation in the range from zero to eight and rotation in [-10°, 10°] are further involved in experiments with PASCAL VOC 2012. Moreover, weight decay coefficient is set to $1e-5$ for promoting the proposed model's generalization capability. Note that the batch size of 12, 16, 6, and 16 are used for PASCAL VOC 2012, CamVid, Cityscapes, and MS-COCO, respectively.

About the backbone CNN, the powerful ResNet-101 [29] is applied as mentioned before. Concretely, final outputs of the 1^{st}, 2^{nd}, 3^{rd} and 4^{th} residual blocks (with d1 = 256, d2 = 512, d3 = 1024, d4 = 2048, respectively) are taken into account for the decoder.

Afterwards, each batch of augmented images is sequentially fed into the proposed network to produce pixel-wise segmentation maps which, along with corresponding ground-truth label maps, get through a softmax cross-entropy loss calculation step. The corresponding formulation is defined as follows

$$\begin{aligned} \mathcal{L}(Y,G) &= \sum_{Y_p}\left(\sum_{j=0}^{Cl}\alpha_j i_{p,j} log(s_{p,j})\right) \\ i_{p,j} &= \begin{cases} 1, & Y_{p,j} = G_{p,j} \\ 0, & otherwise \end{cases} \end{aligned} \tag{4.1}$$

where Y_p represents considered pixels of prediction map Y; α_j stands for balancing coefficient of class $j \in \{1, \ldots, Cl\}$ in which Cl corresponds to the total number of training classes; $i_{p,j}$ indicates the predicted class j of Y_p with respect to its actual class in ground-truth label map G; $s_{p,j}$ denotes softmax score of Y_p corresponding to class j. It is worth noting that in the case of evaluating with

the CamVid dataset, each category j is associated with a loss coefficient α_j determined from the median frequency balancing approach [19]. Meanwhile, as for the remaining datasets, $\alpha_j = 1 \; \forall j$.

Then, the optimization strategy of Chen *et al.* [12] is adopted to minimize the total softmax loss $\mathcal{L}_{\sum}$ in (4.1) with respect to the CAB-Net's parameters (which are initialized following [30]). In short, stochastic gradient descent with momentum of 0.9 is applied together with the 'poly' learning rate decay schedule, wherein learning rate at the i^{th} iteration equals to the initial learning rate (which is set at 0.01 in this work) multiplied by $(1 - \frac{i}{\max_i})^{0.9}$. Correspondingly, pretrained weights of the backbone network are fine-tuned with the contemporary learning rate multiplied by 0.01.

Finally, the CAB-Net is trained with PASCAL VOC 2012, CamVid, Cityscapes, and MS-COCO in 50, 500, 250, and 50 epochs, respectively. The mean Intersection of Union (mIoU) metric is used for performance evaluation. In particular, let us denote p_{xy} as the pixel belonging to ground truth label x is predicted to be of label y, and L as the total number of labels, the mIoU is determined by

$$mIoU = \frac{1}{L} \sum_{x=1}^{L} \frac{p_{xx}}{\sum_{y=1}^{L} p_{xy} + \sum_{y=1}^{L} p_{yx} - p_{xx}}$$

Moreover, a multi-scale test strategy is conducted for the final comparison with the state-of-the-arts besides simply feeding original test images into the finalized model to retrieve corresponding performance, which also report their experimental results applying the same procedure. In concrete, every original test image and its variously scaling (i.e., with factors of {0.5, 0.75, 1.25, 1.5, 1.75} compared to the original size) and horizontal-flipping versions are fed into the built network. The final prediction scores are then averaged from those of all the obtained outputs. Compared to the single-scale test approach, the multi-scale one is capable of boosting mIoU by $1.0 - 3.5\%$ approximately depending on the dataset as reported in Tables 4.4, 4.5, and 4.6, but trading-off a much more expensive computation.

4.3 Ablation Study

For the ablation study, the training plus augmentation (10,582 images) and validation (1,449 images) sets of PASCAL VOC 2012 are used for the evaluation of different setting strategies. In this section, the impact of backbone CNNs with various capacities on the segmentation performance is firstly examined. Next, the comparison between the proposed Bracket-style decoding network

and the Ladder/U-shaped counterpart for leveraging middle-level feature combinations is investigated. Afterwards, how the channel-wisely and spatially attentional mechanisms coordinates with the Bracket-shaped architecture is taken into account. Finally, how the proposed CAB-Net represents semantic details along the decoding process through the visualization of manifold feature maps is demonstrated.

4.3.1 The contribution of backbone CNN to final performance

TABLE 4.1: mIoU (%) on Pascal VOC 2012 [20] validation set and number of parameters with different backbone CNNs.

Backbone CNN	Depth sizes {d1, d2, d3, d4}	mIoU (%)	No. parameters		
			Backbone	Bracket	Total
VGG-16 [79]	{128, 256, 512, 512}	75.24	14.72M	7.13M	21.85M
Xception-65 [17]	{128, 256, 728, 2048}	77.96	20.81M	21.06M	41.87M
ResNet-50 [29]	{256, 512, 1024, 2048}	78.27	23.51M	38.97M	62.48M
ResNet-101 [29]		80.37	42.50M	38.97M	81.47M

{d1, d2, d3, d4}: Depth sizes of backbone CNN's feature maps involved to the Bracket-shaped decoding process (abbreviated as 'Bracket' in the fifth column)

To this end, The contribution of backbone CNN to the final performance in terms of mIoU is further exploited. Specifically, VGG-16 [79] and Xception-65 [17] are utilized as an alternative to the main backbone ResNet-101 [29]. Besides that, the shallower version, i.e., ResNet-50, is included in this experiment for providing further insights into the impact of varying-deep features on the pixel-wise segmentation performance.

In general, the more capacities an architecture has, which means superior representations of deep features are achieved, the better the segmentation performance in terms of mIoU gets (up to around 5.13% for ResNet-101 vs. VGG-16) as reported in Table 4.1. Correspondingly, the model complexity is enlarged as the total number of trainable parameters increases, which is determined by two major factors, i.e., backbone CNN's capacity and depth sizes of the feature maps involved to the Bracket-shaped decoding stage. The former is enumerated in the fourth column of Table 4.1. It is worth noting that despite possessing more layers than those in ResNet-50, Xception-65 has fewer trainable parameters thanks to the full usage of *Sep. Conv.*, which is more cost-efficient that the conventional version as described in Section 3.3. Regarding the latter, information of considered channel dimension is given in the second column of Table 4.1, from which those retrieved from ResNet have largest sizes compared to the counterparts. This leads to the increment of more

hidden nodes and convolution kernels for FC and Sep. Conv. layers, respectively, in the CAF modules. Therefore, applying ResNet as the backbone CNN results in a much larger number of learnable parameters in the Bracket-structured decoding process (more than 1.85 times compared to the others) as well as the whole architecture (more than 1.5 times) accordingly.

However, as aforementioned that semantically-rich details are essentially encoded in channel-wise manner, the deeper features acquired from ResNet are capable of contributing more generalized and informative context to the decoding step than those of VGG or Xception. In addition, since the proposed Bracket-shaped decoding procedure exhaustively involves such varying-scale feature maps through multiple rounds, depth-wisely representational abilities of those features are marked as strongly influential attributes benefiting the final segmentation performance. Consequently, it can be observed from Table 4.1 that employing ResNet-50 as the backbone network introduces a slightly better mIoU (despite fewer layers) while the 101-layer version improves by 2.41% (which is significant in this domain) in comparison with the usage of Xception-65.

4.3.2 The effectiveness of Bracket-style decoding network over the Ladder/U-shaped counterpart for leveraging middle-level features

TABLE 4.2: mIoU (%) on Pascal VOC 2012 [20] validation set and number of parameters with Bracket-style vs. Ladder/U-shaped decoding network.

Feature Combination Strategy	mIoU(%)	No. parameters
Ladder/U-shaped (Fig. 1.1a)	77.30	72.81M
Bracket-style (Fig. 1.1c)	80.37	81.47M

It can be observed that in comparison with the Ladder/U-shaped (Asymmetrically-structured network illustrated in Fig. 1.1a), the proposed Bracket-style decoding scheme (manifested in Fig. 1.1c) engages additional feature combination modules along the tournament of inferencing the final segmentation map. Particularly, middle-level features like $\mathbf{F}_2^0, \mathbf{F}_3^0, \mathbf{F}_2^1$ are extensively utilized to simultaneously combine with adjacent higher- and lower-resolution maps as follows: $\mathcal{C}(\mathbf{F}_1^0, \mathbf{F}_2^0)$, $\mathcal{C}(\mathbf{F}_2^0, \mathbf{F}_3^0)$, and $\mathcal{C}(\mathbf{F}_1^1, \mathbf{F}_2^1)$. Accordingly, an ablation study assessing the effectiveness of such Bracket-style feature combination over the Ladder/U-shaped version is conducted in this sub-section.

From the results reported in Table 4.2, the aforementioned extra combinations for leveraging middle-scale feature maps, which form into the proposed Bracket-style decoding network, make a

considerable gap of mIoU-based performance (↑3.07%) compared to the conventional Ladder/U-shaped counterpart. Notably, the involvement of more combination modules clearly leads to an increment of the parameters' amount by 11.89%, which is however worth trading off for a vast elevation of quantitative segmentation performance. In brief, this experiment further expresses the advantages of continually and extensively operating middle-level features in Bracket-structured manner for the semantic image segmentation objective.

4.3.3 The coordination between Bracket-shaped Network and CAF-based Connections for leveraging middle-level features

TABLE 4.3: mIoU (%) on Pascal VOC 2012 [20] validation set and number of parameters with various settings of attentional mechanism.

Settings		mIoU (%)	No. parameters
Cha. Att.	Spa. Att.		
		76.73	33.66M
	✓	77.86	33.66M
✓		79.45	38.97M
✓	✓	80.37	38.97M

Let's consider a middle-level feature map $\mathbf{F}_i^r$ (with $1 < i < n - r, \forall r$) which plays different roles in two adjacent CAF modules because of the Bracket-shaped connection manner, i.e., the lower-resolution input of $\mathcal{C}_1(\mathbf{F}_{i-1}^r, \mathbf{F}_i^r)$ and the higher-resolution input of $\mathcal{C}_2(\mathbf{F}_i^r, \mathbf{F}_{i+1}^r)$. In the CAF block $\mathcal{C}_1$, $\mathbf{F}_i^r$ contributes its finer representation via the learnable *T. Conv.* layer and the depth-based semantic information via the channel-wisely attentional mechanism to be adjusted by and re-calibrate the remaining input $\mathbf{F}_{i-1}^r$, respectively. Meanwhile, in the CAF module $\mathcal{C}_2$, $\mathbf{F}_i^r$ takes a reversed role in which its finely patterned features and neural units are employed to spatially refine and be re-calibrated in depth-wise manner by the partner $\mathbf{F}_{i-1}^r$, respectively. Consequently, each middle-level feature map is exhaustively exploited as both roles of coarser- and finer-resolution features for comprehensively embedding semantic into fine-grained details on the tournament of inferring the final pixel-wise prediction map.

The advantage of cross-attentional mechanism in the Bracket-style decoding procedure is quantitatively examined by validating various settings in Table 4.3. Compared to the baseline combination, performance improvement introduced by the embedded attentional mechanisms is considerable with 1.13% for spatial-based and especially 2.72% for channel-based attentions. Furthermore, by combining these two attention types in crossing manner, the mIoU is further elevated by 1.0%

approximately. This implies the powerful coordination between Bracket-structured network and the CAF-based connections for leveraging the capability of embedding semantically contextual information into finely patterned features.

Moreover, from the reported number of parameters in Table 4.3, it can be realized that the utilization of one simple 7×7 convolution kernel in each Spa. Att. block has nearly no impact on the model complexity. Meanwhile, employing fully connected layers in Cha. Att. modules increases the number of trainable parameters by approximately 15.8% due to large channel size of processed tensors. Obviously, it is worth trading off such minor complexity increment for an overall mIoU improvement of 3.64%, which is significant in the semantic segmentation problem.

4.3.4 Representation of feature maps with respect to different attentional schemes

In this part, the visual representations of key feature maps for semantic segmentation, comprising $\mathbf{F}_1^1$, $\mathbf{F}_1^2$, and $\mathbf{F}_1^3$, with respect to different attentional schemes are introduced. Given an image fed into the proposed CAB-Net, responses in the chosen features are averaged over corresponding channel dimension. Then, those pixel intensities are scaled to the range of [0, 255] as illustrated by two example cases in Fig. 4.1.

Clearly, since $\mathbf{F}_1^1$ is only decoded by low-level semantic information (from $\mathbf{F}_1^0$ and $\mathbf{F}_2^0$) in the CAB-Net, using naive upsampling followed by element-wise fusion still results in ambiguous features for next rounds. In contrast, applying any attentional mechanisms initializes more meaningful focuses (with high pixel intensities) on object details as shown in the last three rows in the first and fourth column of Fig. 4.1. Then, in the second round, features $\mathbf{F}_2^2$ inferred by the non-attention strategy (first row in Fig. 4.1) continue to hardly manifest the regions of interest. Although the Bracket-shaped network structure is able to smoothly embed semantically rich features to spatial context round-by-round, the representation of predefined object categories is still not optimal.

Accordingly, the utilization of spatially and channel-wisely attentional modules has strengthened the capability of expressing vital features and diminishing trivial ones. On the one hand, using Spa. Att. is able to precisely orientate the expressiveness while effectively maintaining spatial context as shown in $\mathbf{F}_1^1$ and $\mathbf{F}_1^2$ (see third row compared to those in the second and fourth rows). On the other hand, involving Cha. Att. blocks can leverage the contribution of semantic details encoded along depth dimension, which plays an important role for class discrimination. However, as can be observed from $\mathbf{F}_1^3$ in the third and fourth rows of Fig. 4.1, Spa. Att. blocks face

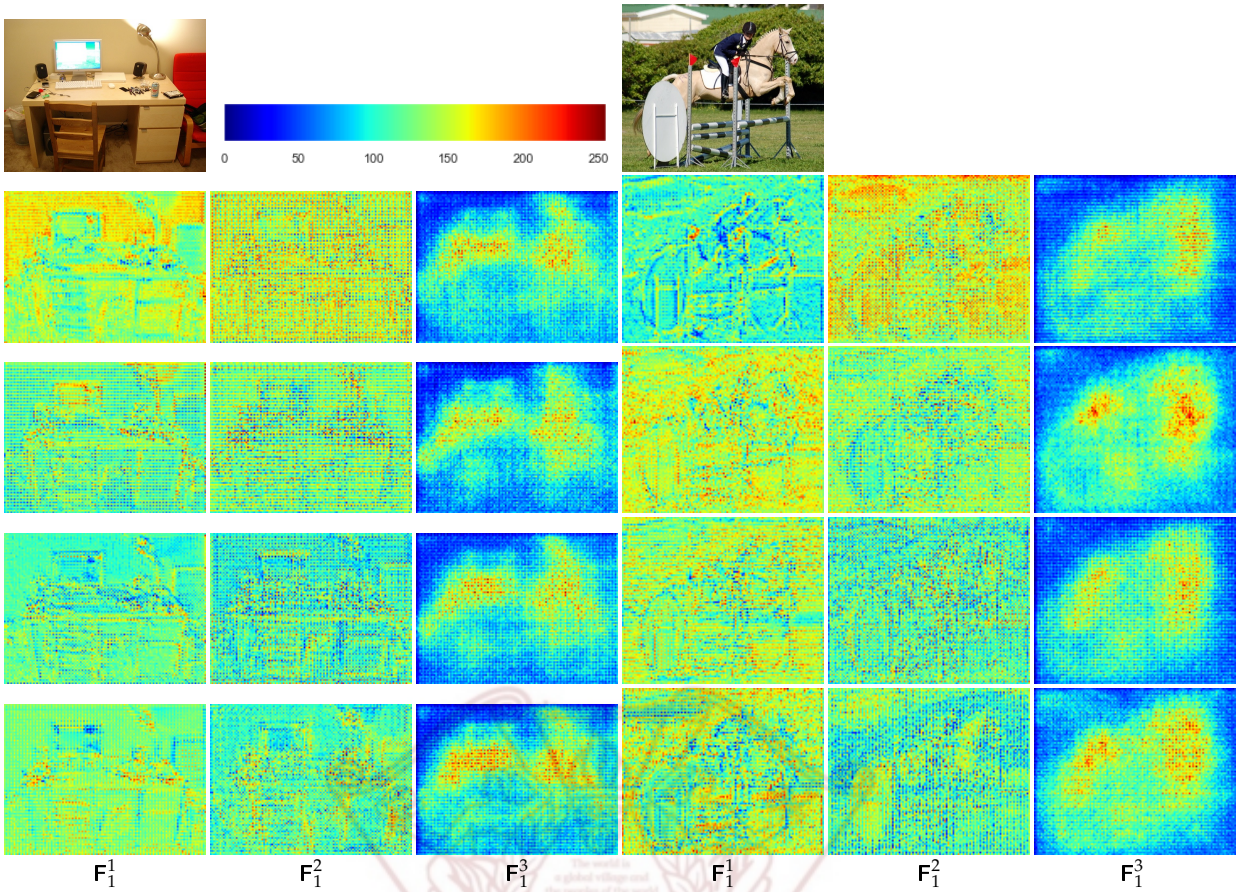


FIGURE 4.1: Representation of key feature maps (i.e., $\mathbf{F}_1^1$, $\mathbf{F}_1^2$, and $\mathbf{F}_1^3$ from left to right) extracted by the proposed CAB-Net with respect to different attentional schemes. Note that the responses presented in the feature maps are averaged over the depth dimension. Top row: example raw images in PASCAL VOC 2012 [20] validation set and a color-intensity indicator; 2^{nd} row: no Cha. Att. and Spa. Att.; 3^{rd} row: only using Spa. Att.; 4^{th} row: only using Cha. Att.; and last row: applying both Cha. Att. and Spa. Att. in the connection blocks. Color view is recommended for the best visualization.

difficulty of distributing semantically rich features (of which the pixels should have high intensities) over space. Meanwhile, Cha. Att. blocks show its weakness in highlighting extracted spatial features.

Finally, with the proposed cross-attention scheme described in Section 3.3, the advantages of both Spa. and Cha. Att. modules are comprehensively combined for better differentiation and localization of objects' features. In specific, feature maps in the last row of Fig. 4.1 perform the best coordination between semantically rich features and corresponding spatial context. For instance, compared to the counterparts, semantic features of the horse's body in the right $\mathbf{F}_1^2$ are expressed and localized more impressively, which leads to better representation of attentional responses in the subsequent $\mathbf{F}_1^3$.

4.4 Comparison with State-of-the-art Methods

4.4.1 PASCAL VOC 2012

The experimental performance on test set is quantitatively reported in Table 4.4. It can be observed that the proposed approach achieves competitive mIoU of 83.6% compared with that of the state-of-the-arts. Regarding the class-wise results, the CAB-Net attains the top performance with significant margin (up to 3.7%) for 10/20 semantic objects ranging from small to large scale. Meanwhile, state-of-the-art results of the remaining labels are shared between deep models applying dilated convolution operators such as EncNet [101], PSPNet [103], and WideResNet [95]. Another noteworthy methodology called Tree-structured Kronecker CNN (TCKN) [94] adopted Kronecker product as the custom convolutional layers, which nails the second-best overall performance in Table 4.4. Differently, by employing the *Sep. Conv.* layers under the unique Bracket-shaped structure, the proposed model is able to give superior achievements over those networks. In specific, the exhaustive employment of middle-level features during the inference process is able to continuously refine the integration of semantic context to high-resolution representation. As a consequence, the details of various scales are managed more efficiently. Besides that, thanks to the cross-manner operation of multiple CAF blocks along the Bracket-style decoder, the proposed architecture outperforms the DANet [23] (which uses dual attention applied only to the highest-level feature in parallel fashion) by 1.0%, which is significant in such a competitive semantic segmentation topic.

Moreover, typically visual results exhibited in Fig. 4.2 have shown the effectiveness of the CAB-Net in partitioning multiple categories of different scales. Additionally, compared to the outputs introduced by B-Net-VGG-LCM, the proposed network can reason better pixel-wise labeling performance, especially the *bird* and *chair* classes. However, the proposed model still fails in precisely segmenting objects which contain interior gaps (void labeled regions in ground-truth map) such as the light-brown chair and the horse's body parts overlapped by fence in the fourth and fifth rows of Fig. 4.2, respectively. In addition, a very small-sized airplane located at right side of the input image in the first row is not segmented in the prediction map. It is argued that the largest-resolution feature maps involved in the proposed decoding procedure have stride of 4 is the major reason for several improper representations of those small spaces for complete scene learning. Accordingly, taking into account features with stride of 2 may bring in finer details for better local context learning. Nevertheless, such approach requires an unworthy trade-off

TABLE 4.4: Comparison of per-class IoU and mIoU (%) on Pascal VOC 2012 [20] test set. **Boldface** numbers indicate the best performance at each class.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU (%)
FCN [67] (A)	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
B-Net-VGG-LCM [35] (A)	92.0	42.9	92.3	73.3	77.5	91.4	86.4	91.5	42.7	81.9	61.6	84.4	85.8	88.4	90.1	65.5	86.4	60.0	86.1	72.5	78.5
G-FRNet [42] (S)	91.4	44.6	91.4	69.2	78.2	95.4	88.9	93.3	37.0	89.7	61.4	90.0	91.4	87.9	87.2	63.8	89.4	59.9	87.0	74.1	79.3
DDSC [8] (ss) (S)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	81.2
WideResNet [95] (S)	94.4	72.9	94.9	68.8	78.4	90.6	90.0	92.1	40.1	90.4	71.7	89.9	93.7	**91.0**	89.1	71.3	90.7	61.3	87.7	78.1	82.5
PSPNet [103] (A)	91.8	71.9	94.7	71.2	75.8	95.2	89.9	**95.9**	39.3	90.7	71.7	**90.5**	94.5	88.8	89.6	**72.8**	89.6	64.0	85.1	76.3	82.6
DANet [23] (A)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	82.6
DFN [97] (A)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	82.7
EncNet [101] (A)	94.1	69.2	**96.3**	**76.7**	**86.2**	96.3	**90.7**	94.2	38.8	90.7	73.3	90.0	92.5	88.8	87.9	68.7	**92.6**	59.0	86.4	73.4	82.9
TKCN [94] (A)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	83.2
CAB-Net (ss) (A)	94.9	69.2	92.9	73.1	79.5	95.8	87.4	94.1	39.7	87.5	72.4	90.6	92.4	86.4	89.1	68.3	90.8	64.3	88.7	**78.5**	82.5
CAB-Net (A)	**96.0**	**75.6**	94.3	69.1	79.9	**97.1**	89.8	94.8	**40.4**	**91.2**	**74.6**	89.4	**94.7**	87.2	**91.7**	69.6	92.1	**65.5**	**88.8**	76.9	**83.6**

(ss): single-scale testing strategy; -: no data recorded in the original work; (A): Asymmetrically-structured Network; (S): Symmetrically-structured Network

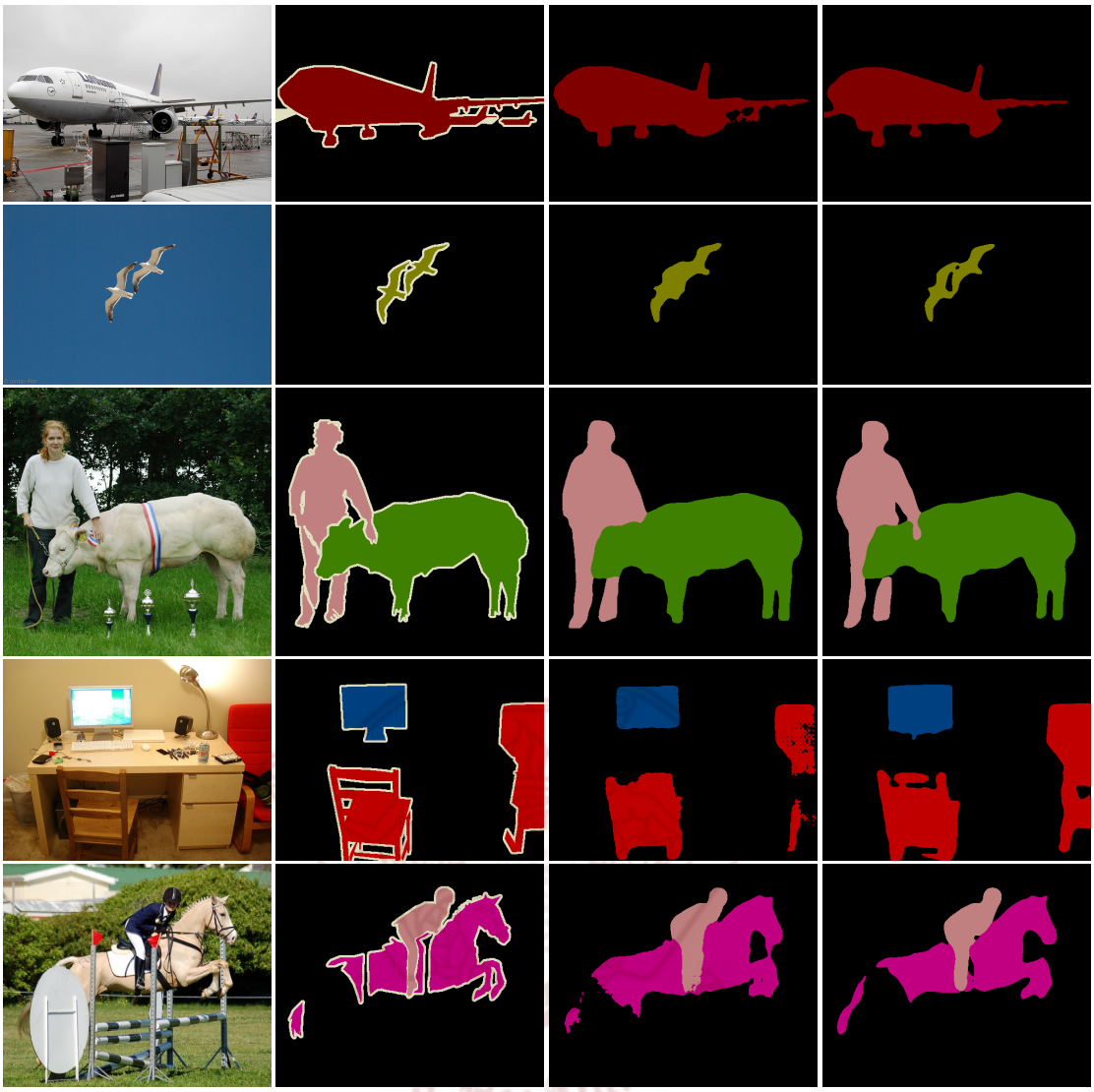


FIGURE 4.2: Several qualitative results on Pascal VOC 2012 [20] validation set. Left to right: original images, ground-truth labels, results of B-Net-VGG-LCM [35], and the proposed CAB-Net.

for much lower allowed size of training mini-batches and higher number of operations as well as model complexity during training, which should even make the overall performance worse accordingly.

4.4.2 CamVid

It can be observed from the Table 4.5 that the proposed CAB-Net obtains state-of-the-art mIoU of 76.4%. Regarding per-class performance, the proposed network reaches state-of-the-art class-wise IoU in 10 (*building*, *tree*, *sky*, *car*, *sign-symbol*, *road*, *pedestrian*, *fence*, *sidewalk*, and *bicyclist*) out of totally 11 semantic labels. Remarkably, significant margins ranging diversely from 0.1%

TABLE 4.5: Experimental per-class IoU and mIoU (%) on CamVid [9] dataset. **Boldface numbers** indicate the best performance at each class.

Approach	Building	Tree	Sky	Car	Sign-symbol	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	mIoU (%)
SegNet (3.5K dataset) [6] (S)	-	-	-	-	-	-	-	-	-	-	-	60.1
DeepLab-LFOV [14] (A)	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6
Dilation8 [99] (A)	82.6	76.2	89.9	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3
Dilation+FSO-DF [53] (A)	84.0	77.2	91.3	85.6	49.9	92.5	59.1	37.6	16.9	76.0	57.2	66.1
B-Net-VGG-LCM [35] (A)	81.4	75.3	92.8	82.5	42.8	89.2	60.8	47.8	**36.3**	66.4	54.8	66.4
G-FRNet [42] (S)	82.5	76.8	92.1	81.8	43.0	94.5	54.6	47.1	33.4	82.3	59.4	68.0
BiSeNet [98] (A)	83.0	75.8	92.0	83.7	46.5	94.6	58.8	53.6	31.9	81.4	54.0	68.7
DDSC [8] (ss) (S)	-	-	-	-	-	-	-	-	-	-	-	70.9
LDN121 16→2 [52] (S)	-	-	-	-	-	-	-	-	-	-	-	75.8
CAB-Net (ss) (A)	88.7	87.2	94.9	91.0	60.5	**94.9**	57.4	60.4	26.8	**85.4**	55.4	73.0
CAB-Net (A)	**91.1**	**88.9**	**95.7**	**93.0**	**64.8**	94.7	**66.5**	**70.5**	29.8	85.3	**60.3**	**76.4**

(ss): single-scale testing strategy; -: no data recorded in the original work; (A): Asymmetrically-structured Network; (S): Symmetrically-structured Network

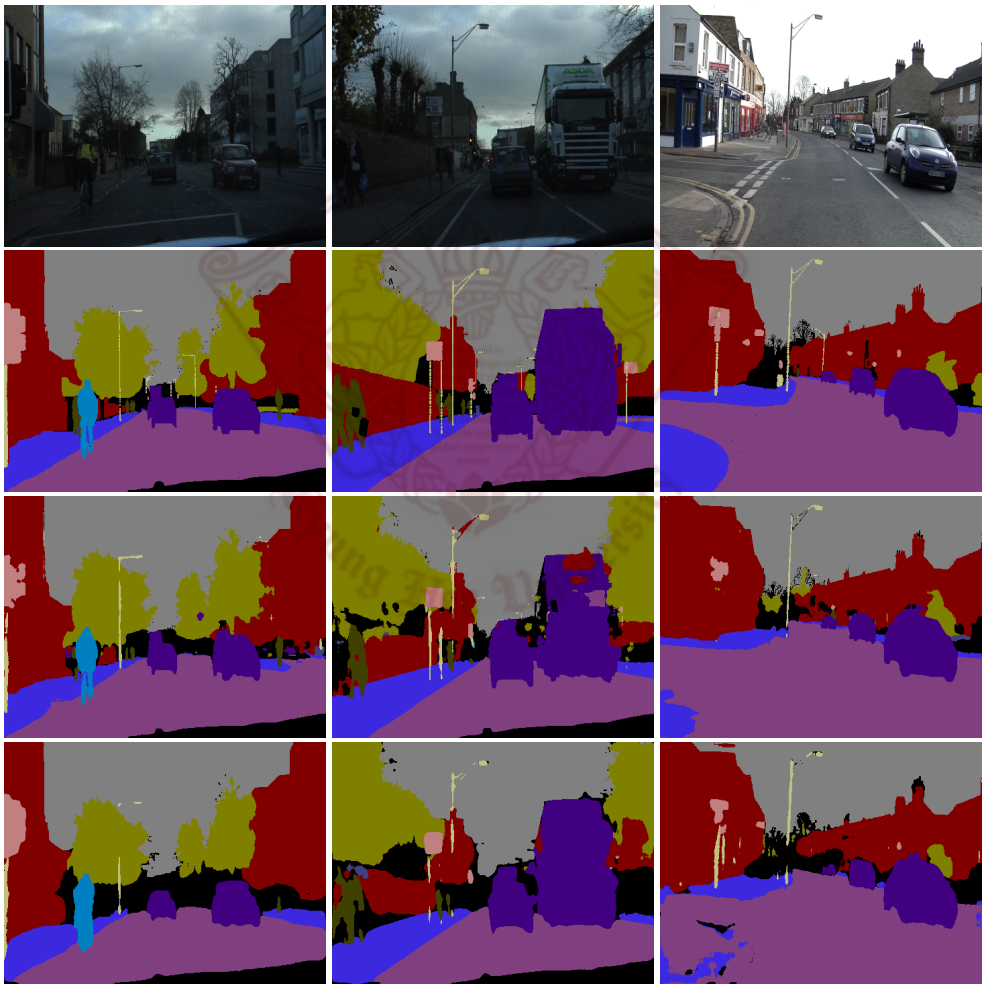


FIGURE 4.3: Several qualitative results on the CamVid [9] dataset. First row: original images; 2^{nd} row: corresponding ground truth labeled maps; 3^{rd} row: results inferred by B-Net-VGG-LCM [35]; and last row: CAB-Net.

to 16.9% are gained at these categories in comparison with the corresponding second places. These achievements show that the exhaustive utilization of middle-scale feature maps in terms

of Bracket-structured manner can effectively embed semantic information to the representation of medium- to small-sized objects (e.g., *tree*, *pedestrian*, *sign-symbol*) while providing precise annotations for large-sized ones (e.g., *building*, *car*). However, the segmentation performance for the *pole* label is still lower than that of B-Net-VGG-LCM [35] by 6.5%. Apparently, due to the heavily imbalanced class issue, it is challenging to produce the performance higher than 50% for this category, even in the existing work. In addition, despite the fact that feature maps with stride of 2 are also taken into account in [35] allows fine details like *pole* to be acquired more effectively, the corresponding mIoU is significantly lower than that of the proposed CAB-Net by 10%. This arguably implies that, as discussed at previous sub-section, the involvement of too large-sized tensors during the decoding process should encounter the issue of training convergence and following non-optimal test performance.

Besides that, several visual results compared with those of B-Net-VGG-LCM [35] and the corresponding ground-truth maps are illustrated in Fig. 4.3. Obviously, the proposed architecture is able to reduce the wrong labeling between *truck* (in purple) and *building* (in red) as displayed in the 2^{nd} row; *sidewalk* (in blue) and *road* (in magnetta) as shown in the 1^{st} and 3^{rd} row, respectively. This infers that the discrimination between similar-sized objects is performed better thanks to the usage of a more robust backbone CNN, the newly defined connection module called CAF and its powerful coordination with the Bracket-style decoding structure.

4.4.3 Cityscapes

The quantitative and qualitative benchmark results of this dataset from the evaluation server are presented in Table 4.6 and Fig. 4.4, respectively. The proposed CAB-Net achieves a competitive mIoU of 78.3%, where the performance of semantically recognizing motion objects like *fence*, *vegetation*, *rider*, *car*, *truck*, *bus*, and *motorbike* is superior over that of the state-of-the-art methods by a large margin (up to 2.7%). The performance of remaining categories, except for small-scale *traffic light* and *sign symbol*, has average lower IoU of 0.6% approximately in comparison with that of the state-of-the-art PSPNet [103]. Clearly, while varying-rate dilated convolution-oriented approach can capture contextual information well for segmenting static things impressively, the proposed technique is more robust at tackling motion instances of different scales thanks to the comprehensive utilization of middle-level features. In particular, despite the issue of handling very small-sized objects in high-resolution images, the proposed hybrid attentional mechanism coordinating in crossing manner still enables the proposed model to 'mark' diverse representation of medium-

TABLE 4.6: Comparison of per-class IoU and mIoU (%) on Cityscapes [18] test set. **Boldface** numbers indicate the best performance at each class.

Approach	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU (%)
SegNet [6] (S)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	56.1
FSSNet [102] (A)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	58.8
FCN [67] (A)	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
DeepLab-CRF [12] (A)	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
RefineNet [62] (S)	98.2	83.3	91.3	47.8	50.4	56.1	66.9	71.3	92.3	70.3	94.8	80.9	63.3	94.5	64.6	76.1	64.3	62.2	70.0	73.6
BiSeNet [98] (A)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	74.7
SwiftNetRN-18 [72] (S)	98.3	83.9	92.2	46.3	52.8	63.2	70.6	75.8	93.1	70.3	**95.4**	84.0	64.5	95.3	63.9	78.0	71.9	61.6	73.6	75.5
B-Net-VGG-LCM [35] (A)	98.4	84.8	92.4	55.1	55.5	62.1	71.7	76.3	93.3	71.4	95.0	85.1	67.9	95.6	60.5	72.0	62.4	67.3	74.9	75.9
DUC-HDC [89] (A)	98.5	85.5	92.8	**58.6**	55.5	**65.0**	73.5	77.9	93.3	72.0	95.2	84.8	68.5	95.4	70.9	78.8	68.7	65.9	73.8	77.6
PSPNet [103] (A)	**98.6**	**86.2**	**92.9**	50.8	58.8	64.0	**75.6**	**79.0**	**93.4**	**72.3**	**95.4**	**86.5**	**71.3**	**95.9**	68.2	79.5	73.8	69.5	**77.2**	**78.4**
CAB-Net (ss) (A)	98.3	83.4	92.3	52.2	54.8	62.0	70.8	75.4	93.3	70.9	95.2	85.1	68.4	95.4	65.4	80.3	**80.7**	67.7	73.7	77.1
CAB-Net (A)	98.5	85.4	92.8	55.6	**59.1**	63.3	70.9	75.6	**93.4**	71.1	95.2	86.4	**71.3**	**95.9**	**72.3**	**82.2**	72.3	**70.4**	76.5	78.3

(ss): single-scale testing strategy; -: no data recorded in the original work; (A): Asymmetrically-structured Network; (S): Symmetrically-structured Network

to large-scale targets reasonably like the moving instances. Simultaneously, with the dense combination scheme between the decoded feature maps by the Bracket-structured network, the localization of those categories is continuously refined for the optimal pixel-wise labeling as depicted in Fig. 4.4. Furthermore, compared to the sibling B-Net-VGG-LCM, the proposed CAB-Net, with the remarkable improvements in terms of backbone network as well as attentional connection scheme, can label the objects more accurately (e.g., the representations of *sidewalk* category in the second row of Fig. 4.4).

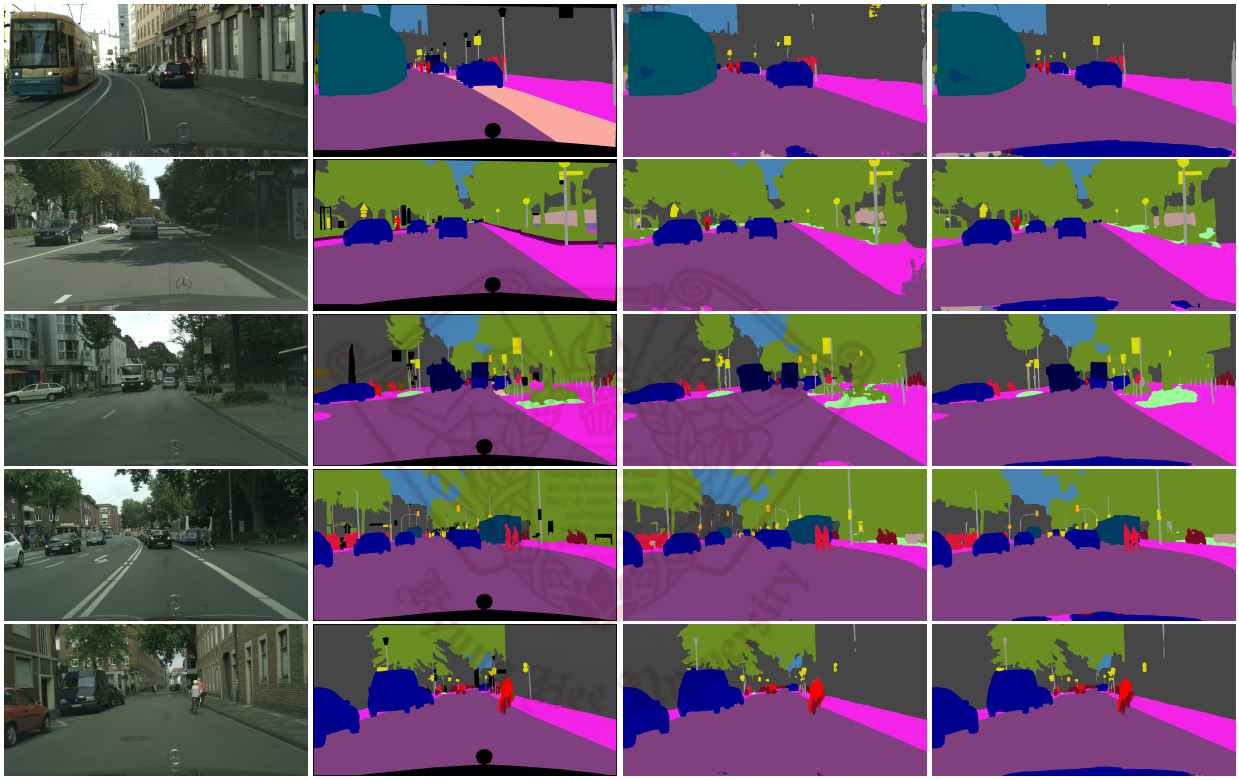


FIGURE 4.4: Several qualitative results on Cityscapes [18] validation set. Left to right: original images, ground-truth labels, B-Net-VGG-LCM [35], and CAB-Net.

4.4.4 MS-COCO

In this experiment, the compared methods, i.e., U-Net [75] (representative of existing *symmetrically-structured* network topology) and DeepLabv3+ [13] (*asymmetrically-structured* counterpart), are reproduced with several customization so that they all have similar amount of learnable parameters with that of the proposed CAB-Net (i.e., ≈81.5M) for a fairer comparison given the same environmental settings. In particular, the backbone network of the reproduced U-Net [75] is ResNet-101 [29] and the depth-wise concatenation operators in the combination modules are replaced by

TABLE 4.7: Comparison of per-class IoU and mIoU (%) on PASCAL VOC 2012 [20] test set for models additionally trained with MS-COCO [64] dataset. **Boldface** numbers indicate the best performance at each class.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU (%)
U-Net(ResNet101) [75] (S)	90.7	46.0	94.4	67.0	**83.7**	89.1	82.0	96.4	40.8	88.6	53.6	**93.1**	91.7	**93.3**	90.5	**75.7**	91.3	60.9	**90.8**	80.9	80.7
DeepLabv3+ [13] (A)	95.4	74.4	92.8	70.0	78.5	95.0	**90.1**	**96.7**	37.5	90.3	73.3	91.6	94.7	88.9	89.9	70.3	**92.6**	62.2	87.0	77.7	83.1
CAB-Net (A)	**95.6**	**78.9**	**95.1**	**74.7**	83.3	**97.0**	89.8	96.1	**44.4**	**92.3**	**78.8**	92.4	**97.0**	91.5	**91.9**	70.2	92.3	**64.2**	88.2	**81.1**	**85.3**

(A): Asymmetrically-structured Network; (S): Symmetrically-structured Network

the element-wise addition counterpart. Meanwhile, modifications in the DeepLabv3+ [13] with backbone ResNet-101 [29] are enumerated as follows: (i) output channel dimension of the Atrous Spatial Pyramid Pooling (ASPP) module is changed from 256 to 576; (ii) number of convolution filters applied to the considered low-level feature map in the decoder is changed from 48 to 256; and (iii) channel size of the features concatenated by the processed low-level feature map (with new channel size of 256) and the output feature of ASPP scheme (with new channel size of 576) is set to be compressed from 832 (=256+576) to 304 (instead of from 256+48=304 to 256 as in the original version) prior subsequent operations.

From the experimental results reported in Table 4.7, it can be realized that the proposed architecture attains the competitive mIoU of 85.2%, which significantly outperforms those of the compared networks by $2.2 - 4.6\%$. With respect to the class-wise segmentation results, the proposed CAB-Net reaches top performance on 12/20 semantic object categories (i.e., *airplane*, *bike*, *bird*, *boat*, *bus*, *chair*, *cow*, *table*, *horse*, *person*, *sofa*, and *television*). These outcomes demonstrate the effectiveness of the proposed deep learning architecture in labeling objects of various scales and deformations. Remarkably, it is obvious that the quality of segmenting the corresponding object classes in PASCAL VOC 2012 dataset (as shown in Table 4.4) is further leveraged thanks to the significant supplement of well-annotated training images' amount in this benchmark dataset. In particular, the gap between CAB-Net only trained with PASCAL VOC 2012 [20] dataset and that additionally learned from MS-COCO [64] in advance is 1.7%, which is considerable in such a competitive research area.

4.4.5 Computational Complexity

Finally, the computational complexity in terms of inference speed and total number of parameters is compared with several existing methods belonging to both *symmetrically* and *asymmetrically-structured* network families. In each group, approaches with different ultimate objectives, i.e., focusing more on either labeling accuracy or inference rapidity, are involved in the discussion. Accordingly, high-resolution images (1024×2048) in Cityscapes dataset [18] are taken into account for this experiment, from which the comparison details are given in Table 4.8. The proposed CAB-Net is run on a Linux OS-based desktop computer equipped with Intel® Core™ i7-7700 CPU at 3.6GHz $\times$ 8, NVIDIA GeForce GTX 1080Ti GPU, and 32GB RAM, which yields the inference speed of 20 frames per second (fps). Meanwhile, the B-Net-VGG-LCM [35] reaches 27 fps because channel sizes of the feature maps utilized from the backbone VGG-16 are considerably shallower.

TABLE 4.8: Comparison of mIoU, inference speed, and number of model parameters for input image with resolution of 1024×2048 in Cityscapes [18] dataset. **Boldface** numbers indicate the best performance at each criterion.

Network structure	Approach	NVIDIA GPU	mIoU (%)	Inference speed (fps)	No. parameters
Symmetric	SegNet [6]	Titan X	56.1	24	29.46M
	SwiftnetRN-18 [72]	GTX 1080Ti	75.5	39	11.80M
Asymmetric	PSPNet [103]	GTX 1080Ti	**78.4**	11	65.60M
	BiSeNet [98]	Titan Xp	74.7	**65.5**	49.00M
Bracket	B-Net-VGG-LCM [35]	GTX 1080Ti	75.9	27	25.92M
	CAB-Net	GTX 1080Ti	78.3	20	81.47M

Regarding the *symmetrically-structured* topology, both SegNet [6] and SwiftnetRN-18 [72] have faster segmentation speeds of 4 and 19 fps than the proposed model due to the employment of much lower-capacity CNNs, i.e., VGG-16 [79] and ResNet-18 [29], respectively. In concrete, for such kind of symmetric encoder-decoder, the process of inferring pixel-wise labeled map mainly relies on the inherent structure of backbone CNN. Therefore, applying shallower network that extracts features having smaller depth size to be involved in the decoding stage requires fewer parameters and operations throughout the whole architecture. However, there is a huge trade-off with the mIoU-based performance, wherein the proposed CAB-Net greatly outperforms the SegNet [6] and SwiftnetRN-18 [72] by 22.2% and 2.8%, respectively. On the other hand, it is noteworthy that the B-Net-VGG-LCM [35], another representative of Bracket-style structure, attains higher mIoU (of 19.8%) and processing rate (of 3 fps) while having same backbone CNN but fewer parameters (of 12%) in comparison with those of the SegNet [6]. The major reason is that its decoder is the reverse replication of the original VGG-16 [79], which is obviously more expensive than the connections between several selective feature maps only in the Bracket-shaped structure or the SwiftnetRN-18 [72].

Compared to PSPNet [103] and BiSeNet [98] in *asymmetrically-structured* group, the proposed CAB-Net contains more number of parameters but still accomplishes noticeable results in the remaining criteria. Particularly, the proposed method reaches a comparable mIoU (with trivially lower rate of 0.1%) but nearly double inference speed (20 vs. 11 fps) in comparison with the PSPNet [103]. It is argued that the primary cause is the manifold utilization of the deepest feature maps in ResNet-101 [29] for various pooling rates followed by conventional convolutional layers in that approach. Such strategy heavily elaborates the volume of operations (comprising multiply, add, max-value calculations, etc), which subsequently reduces the segmentation speed.

Meanwhile, the involvement of efficient attentional and *Sep. Conv.* layers to lower-depth features round-by-round in the proposed technique conducts cheaper operation burden despite carrying more learnable parameters. On the contrary, since the BiSeNet [98] targets at processing rapidity more favorably, it is built upon the lightweight backbone ResNet-18 [29] with an attached dual network stream for amalgamating global context and local details in a cost-efficient way. Hence, the inference speed is impressive with approximately 65 fps but compromising poorer mIoU with a gap of 3.6% compared to that of the proposed CAB-Net.

TABLE 4.9: Comparison of mIoU, inference speed, total multiply-adds operations and memory occupancy given similar number of model parameters for input image with resolution of 513×513 in PASCAL VOC 2012 [20] dataset. **Boldface** numbers indicate the best performance at each criterion.

Approach	NVIDIA GPU	No. parameters	mIoU (%)	Inference speed (fps)	Total mult-adds	Memory occupancy	
						Forward/Backward pass size	Parameters size
U-Net [75] (S)	Titan RTX	≈81.5M	80.7	**48**	163.1G	**1,701M**	326M
DeepLabv3+ [13] (A)			83.1	37	**145.4G**	1,836MB	326MB
CAB-NET (A)			**85.3**	41	169.4G	2,002MB	326MB

(A): Asymmetrically-structured Network; (S): Symmetrically-structured Network

Furthermore, in order to conduct a more comprehensive and fairer comparison of computational complexity, the two reproduced approaches, i.e., U-Net [75] and DeepLabv3+ [13], evaluated with MS-COCO [64] and PASCAL VOC 2012 [20] datasets as described in Section 4.4.4 are taken into account. Given the mIoU performance for input images of size 513×513 and similar number of trainable parameters (≈81.5M) with same backbone ResNet-101 [29] in these two networks and the proposed model as discussed before, statistics of inference speed, total multiply-adds operations, and memory occupancy in terms of forward/backward pass size and parameters size using one NVIDIA Titan RTX GPU are summarized in Table 4.9.

Particularly, since the proposed architecture targets at segmentation quality with the dense involvement of cross-attention fusion modules organized in a hierarchical manner, the mIoU performance achieves highest rate of 85.3%. However, this leads to unavoidable trade-offs that the total multiply-adds operations and the forward/backward pass size occupied in memory are largest with 169.4G and 2,002M, respectively. Meanwhile, the customized U-Net [75] reaches the fastest inference speed (with 7 more fps compared to that of the proposed CAB-Net) and least memory access (with around 300MB less than that of the proposed model). Clearly, as channel-wise concatenation is replaced by element-wise summation in combination modules of the considered U-Net variant as described previously, computational burden and memory occupancy are further

reduced but trading off the worst quantitative segmentation performance. On the other hand, despite the consequence of slowest throughput with 37 fps, the customized DeepLabv3+ [13] attains the fewest number of multiply-adds operations with 10.9-14.2% less than those of the compared techniques. It is argued that the significant increment of tensor's depth size in the ASPP mechanism followed by a specialized decoder in the modified DeepLabv3+ [13] leads to more memory usage and simultaneously slows down the operational momentum of the convolutional layers along the feedforward process. In a nutshell, it can be realized that the mIoU, inference speed, and model complexity in terms of parameters' amount and memory occupancy are strongly correlated criteria, to which the preference certainly depends on predefined purposes of each deep learning model.

Chapter 5

Bracket-style Network Variant for Medical Image Segmentation

In this chapter, overview of retinal blood vessel segmentation domain, a popular medical-related topic, is firstly delivered. Then, descriptions of Bracket-style network variant are given. Afterwards, the DRIVE dataset [80] with corresponding evaluation metrics for benchmark is presented. Next, training configurations are taken into account. Finally, ablation study along with performance analyses are conducted to show the effectiveness of the proposed methodology. Remarkably, the experimental procedures involving human subjects described in this chapter were approved by the Institutional Review Board.

5.1 Domain Overview

According to World Health Organization (WHO), Diabetic Retinopathy (DR) is the top-five and -four causes of vision impairment and blindness on earth, respectively. It is originated by the long-term impact of diabetes which results in adverse changes in nerves and blood vessels of the patient's retina. Specifically, the earliest sign of DR is the phenomenon of capillary wall dilatation [16] at retinal vessels, namely microaneurysm (MA). These swellings may cause blood and fluid outflow to the retina, which leads to a more severe level of vision loss. Therefore, early and accurate detection of the MA can help ophthalmologist easily deploy optimal treatment plan for DR progression prevention and management. As a consequence, developing a method be able to precisely localize the MA on color fundus photographs is currently an open research area. In color fundus photography, abnormal changes in representation of retinal blood vessel may tell initial

sign of common eye diseases comprising diabetic retinopathy (DR), glaucoma, ocular hypertension, cataracts, to name a few. For example, the phenomenon of capillary wall dilatation [16] at retinal vessels, namely microaneurysm, is the earliest indicator of suffering from DR. Therefore, efficiently extracting the vessel-based information can help ophthalmologists precisely diagnose and effectively deploy an optimal treatment plan for prevention and regulation of blindness and vision impairment for patients.

As a consequence, there has been a lot of efforts dealing with the problem of vessel segmentation in the last decade. Recently, resulting from the expeditious growth of computational resources like Graphical Processing Units (GPU) as well as the quantity of image datasets, Convolutional Neural Networks (CNNs) has been widely employed in various domains of medical image processing with impressive performance thanks to the powerful feature representation. Specifically, in the field of retinal blood vessel segmentation in color fundus image, there are many attempts handling such kind of binary classification problem at pixel level, a.k.a. semantic segmentation given 2 classes (background and vessel), based on fully convolutional neural networks (FCN) architectures. For instance, CNN-RFs [91] utilized CNNs and Random Forest as feature extractor and corresponding classifier, respectively, for the vessel segmentation. Besides that, the authors of [69] introduced a model including a base CNN for extracting meaningful features along with additional layer blocks specialized for simultaneously segmenting optic disc and retinal blood vessel. On the other hand, since the ratio between the number of vessel and background pixels is massively imbalanced, many works split a given fundus image into multiple overlapping patches [22, 31, 44, 49, 65], which are considered as newly augmented images, in order to address the class-imbalancing issue as well as increase dataset size for combating overfitting matter. In specific, methods proposed in [44, 65] took into account RGB patches as inputs of CNNs formed by stacks of convolution, max-pooling, and fully connected layers. Meanwhile, Feng *et al.* [22] proposed a technique called local entropy sampling to generate grayscale patches from original fundus photography as inputs of a predefined FCN having skip-connection scheme. He *et al.* [31] implemented a similar approach but additionally took into consideration of differences between small and large vessel regions by a local de-regression along with regression based deep architecture. Furthermore, instead of converting RGB to grayscale as in [22, 31], Kassim *et al.* [49] only involved green channel of the raw fundus image to constitute patches for training a predefined 14-layer CNN. The readers may refer to [71] for an intensive review of existing literature of retinal vessel segmentation area.

As aforementioned, small patches generation can reduce the imbalance between the amount of retinal blood vessel and background pixels, which facilitates the semantic segmentation model to encode features more effectively. However, it is obvious that such kind of patch-based approaches brings in expensive computations in both data preprocessing and execution stage for trading-off better performance. Therefore, in this section, an architecture called RFA-BNet is introduced to efficiently partition the blood vessels in color fundus photography without the necessity of costly processing small patches of raw images for training the deep learning network. Concretely, it can be realized that because of being pretrained with large-scale dataset, classification-based CNNs like VGG-Net [79], ResNet [29] can delineate the objects of interest at different levels of feature representation, i.e., from finely patterned to semantically rich features. This leads to the hypothesis that leveraging the utilization of those finely patterned features, which should be continuously enhanced the semantically rich information during the pixel-wise prediction map construction process, is capable of labeling small objects more precisely in case of heavily class-imbalancing issue. Hence, Round-wise Features Aggregation (RFA), as the step of exhaustively utilizing finely patterned features, is proposed to be embedded into the B-Net architecture [35], a light-weight version of CAB-Net in Chapter 3, with sorts of specialized manipulations. As a consequence, it is able to comprehensively exploit semantic context of middle-scale features onto the final per-pixel prediction map for the ultimate purpose of segmenting retinal blood vessels, which appear diversely and irregularly in terms of middle- to small-sized objects.

5.2 Descriptions of Bracket-style Network Variant for Medical Image Segmentation

This section firstly delivers a brief description of the initial version of the Bracket-shaped convolutional neural network (B-Net) [35] for semantic segmentation. Then, the approach of round-wise features aggregation embedded on top of the B-Net is proposed in order to effectively address the problem of partitioning retinal blood vessel in color fundus photography, in which the pixel ratio between regions of interest, i.e., the vessels, and the background is heavily imbalanced.

5.2.1 Bracket-shaped Convolutional Neural Networks

In the initial work [35], a novel deep learning based semantic segmentation model, namely B-Net, wherein a Bracket-style decoding process is introduced to construct the final pixel-wise labeled

map from typical feature maps of various scales learned at backbone VGG16-Net [79]. The idea is motivated from the observation that middle-scale features along a classification-based CNN's feed-forwarding path are not exploited intensively for the segmentation problem although they possess valuable balances between fine details and semantically contextual information, which is clearly profitable for the decoding (i.e., per-pixel prediction map inference) process. Accordingly, to leverage those features' usage, every pair of scale-adjacent feature maps chosen from the backbone network passes through predefined lateral connection modules to infer newly decoded outputs, which continuously repeat the same procedure round-by-round until one final prediction map of finest-resolution is obtained. Note that each round of such decoding approach is defined by the process in which n feature maps combining with neighboring versions to yield $n-1$ outputs possessing enhanced semantic information. In other words, the major contribution of their work is that feature maps at middle levels of spatial resolution are comprehensively utilized to simultaneously (i) contribute semantically richer contexts to the adjacent higher-resolution map and (ii) refine ambiguously coarse details in upsampled version of the adjacent lower-resolution one. Consequently, middle- to small-sized object representation is handled effectively in the final labeled map by the B-Net. Since the appearance of the retinal blood vessel is somewhat suitable to target function of the method proposed in [35], the Bracket-style CNN concept with several variations compared to the original work is adopted to maximize the retinal vessel segmentation performance.

In particular, as illustrated in Fig.5.1, the pretrained ResNet-101 [29] is utilized as backbone network of the proposed approach. Subsequently, four feature maps of different scales utilized for the decoding procedure are outputs of the initial convolution layer and three first residual blocks, with strides of 2, 4, 8, and 16 with respect to the input images' spatial dimension, respectively. Let these feature maps of *Round 0* (blue-line rectangles in Fig.5.1) densely combine with their adjacency as described previously, three newly decoded outputs (green-line rectangles of *Round 1* in Fig.5.1) are inferred. Next, the same procedure takes place two more rounds until one finest-resolution feature map (having stride of 2) is remained before the RFA module. The continuous combination between two certain scale-adjacent feature maps during the Bracket-structured decoding process is defined as follows

$$\mathbf{F}_i^r = Conv\left[\mathbf{F}_i^{r-1} \oplus \mathcal{U}(\mathbf{F}_{i+1}^{r-1})\right] \tag{5.1}$$

where $\mathbf{F}_i^r$ is i^{th} feature map at r^{th} round, wherein $r = 1, 2, 3$ and $i = 1, \ldots, 4 - r$ (the larger value of i, the lower spatial resolution (i.e., larger stride) the corresponding feature map has); $\oplus$ stands for element-wise sum; $\mathcal{U}(.)$ represents transposed convolution operator for 2x upsampling; $Conv[.]$ consists of following operations on the sum feature map: Rectified Linear Unit (ReLU) activation, separable convolution [17], and batch normalization [41] for diminishing adverse effects during the upsampling progress. It is worth noting that the number of both the transposed and separable convolution layers is specified to be identical to channel size of the corresponding higher-resolution input at each combination step.

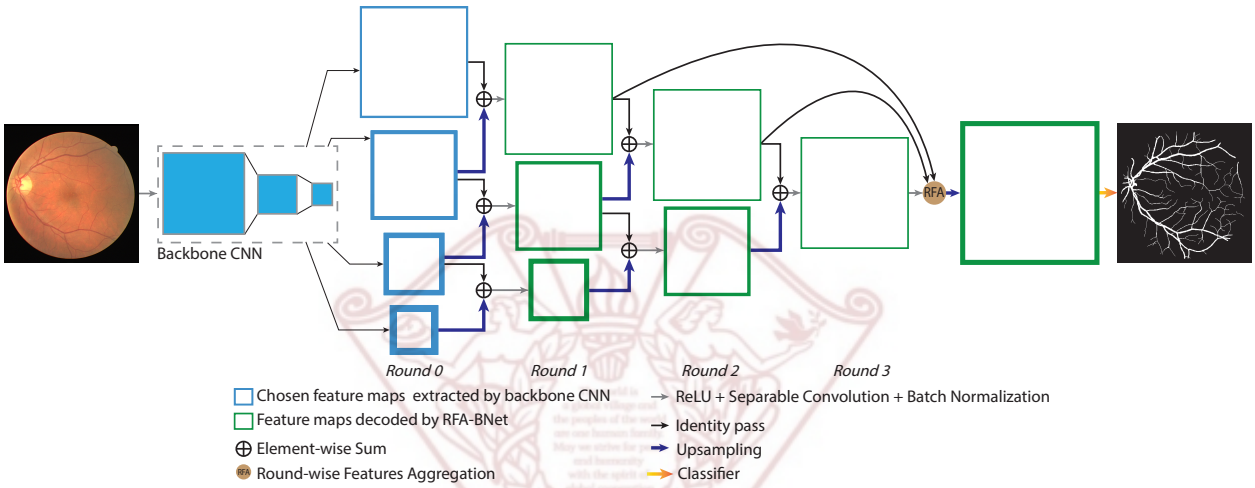


FIGURE 5.1: Architecture of the proposed RFA-BNet, a variant of CAB-Net. Let an input color fundus image be fed into the backbone ResNet-101 [29], final outputs of the initial convolution layer and three first residual blocks, i.e., with strides of 2, 4, 8, 16, respectively, are involved in the Bracket-manner decoding process (in three rounds) for retinal blood vessel labeling. Briefly, these fine-to-coarse feature maps are densely combined via the element-wise summation along with non-linear learning (ReLU, separable convolution [17], and batch normalization [41]) to infer outputs which repeat the same operations round-by-round until only one decoded feature map is left. Then, the highest-resolution decoded feature maps at each round are aggregated via depth-wise concatenation procedure before its upsampled version goes through the predefined classifier for pixel-wise segmentation. Since the finest-resolution feature maps decoded at each round of the Bracket-style CNN are aggregated to produce remarkable representation of retinal blood vessels, such process is called Round-wise Features Aggregation on Bracket-shaped Network (RFA-BNet). Note that area and thickness of rectangles demonstrate spatial and depth size of the corresponding feature maps, respectively.

5.2.2 Round-wise Features Aggregation

Apparently, the exhaustive utilization of middle-scale features by the Bracket-shaped decoder can effectively represent medium- to small-sized objects at pixel level, which is suitable for segmenting

blood vessel in fundus photography. However, naively applying the original structure of the B-Net is obviously not an optimal strategy since the representation of retinal blood vessels is diverse and irregular (if compared with usual contents in natural images), e.g. more and more sudden branches of thin vessels emerge when being away from the optic disc. Also, another noticeable factor is that the ratio between vessel and background pixels are heavily imbalanced (e.g. around 1.3:8.7 in training set of the DRIVE dataset [80]). Therefore, in this work, an approach of RFA is additionally proposed on top of the B-Net manipulated by another backbone network with lower output stride as specified in previous sub-section. Since the finest-resolution feature map at each round possesses different degrees of semantically rich features which may get rid of representation of thin and ambiguous vessels, the RFA module aims to aggregate finest-resolution feature maps of all rounds to make the built model flexibly learn weakly-to-strongly embedded semantic contexts while retaining proper annotations of fine details like thin vessels' edges. In concrete, the finest-resolution feature maps of each round are concatenated along the depth dimension and then the transposed convolution followed is applied by a final classifier as demonstrated in Fig.5.1. Accordingly, the final per-pixel prediction map $\mathbf{Y}$ is produced as below

$$\mathbf{Y} = \mathcal{U}(\mathcal{A}[\mathbf{F}_1^1, \mathbf{F}_1^2, \mathbf{F}_1^3]) \tag{5.2}$$

where $\mathcal{A}[.]$ means depth-wise aggregation procedure.

5.3 Experiments

5.3.1 Benchmark Dataset: DRIVE [80]

DRIVE stands for Digital Retinal Images for Vessel Extraction [80], which is used to validate studies on retinal blood vessel segmentation in fundus photography. The dataset pool consists of totally 40 images, half of which is designated for training and the remaining for testing. It is worth noting that image crop size around field of view (FOV) is fixed at 584×565. Also, mask of the FOV inside each fundus image is provided to specify regions of interest for fair performance evaluation. Hence, ground-truth labels of the retinal background, vessel, and non-FOV pixels are defined as 0, 1, and 255, respectively, during the training stage. For evaluation, well-known metrics such as sensitivity, specificity, accuracy, precision, and AUROC are involved to validate the effectiveness of the proposed method.

5.3.2 Training Configurations

In this work, Tensorflow [4] and Scikit-learn [3] are employed to train and evaluate the proposed deep network on one NVIDIA 1080TI GPU, respectively. Since the dataset pool is small, the training images are massively augmented by following manipulations: random scale of {0.5, 0.75, 1.0, 1.25, 1.5, 2.0}, random crop with centered FOV subject to predefined spatial dimension (i.e., 585×565), depth-wise mean intensity normalization, random horizontal and/or vertical flip. Then, each batch of five augmented images is continuously fed into the proposed architecture. Subsequently, weighted cross-entropy loss function is utilized to assess the compatibility between the resulting pixel-wise prediction maps Y and corresponding ground-truth label maps G in the scenario of class imbalance as follows

$$\mathcal{L}(Y,G) = \sum_{Y_p} \left(\sum_{j=0}^{1} \alpha_j i_{p,j} log(s_{p,j}) \right)$$
$$i_{p,j} = \begin{cases} 1, & Y_{p,j} = G_{p,j} \\ 0, & otherwise \end{cases} \tag{5.3}$$

where Y_p represents considered pixels of prediction map Y, α_j stands for balancing coefficient of class $j \in \{0,1\}$, $i_{p,j}$ indicates the predicted class j of Y_p with respect to its actual class in ground-truth label map G, $s_{p,j}$ denotes softmax score of Y_p corresponding to class j. In this work, α_0 and α_1 are set to be 1.0 and 6.975, respectively, which exhibit the ratio between total number of background (label value of 0) and vessel (label value of 1) pixels in the training dataset. As can be seen from equation 5.3, non-FOV pixels (label value of 255) are ignored during the loss computation procedure. From the measured loss, to optimize parameters initialized by He's approach [30] in the RFA-BNet, Adam optimizer [51] is adopted with learning rate, decay rate of moving average of gradient's first and second moment are assigned at 0.001, 0.9, and 0.999, respectively. In addition, weight decay of 0.0001 is included to boost the generalization capability of the proposed architecture. Finally, the training process runs 500 epochs to finalize the built model for multi-scale plus horizontal and vertical flipping testing.

The retinal vessel segmentation is equivalent to binary classification problem at each pixel of a given fundus image. Therefore, well-known performance metrics such as sensitivity (recall), specificity, accuracy, precision, AUROC (Area Under the Receiver Operating Characteristics) are used to evaluate the effectiveness of the proposed methodology.

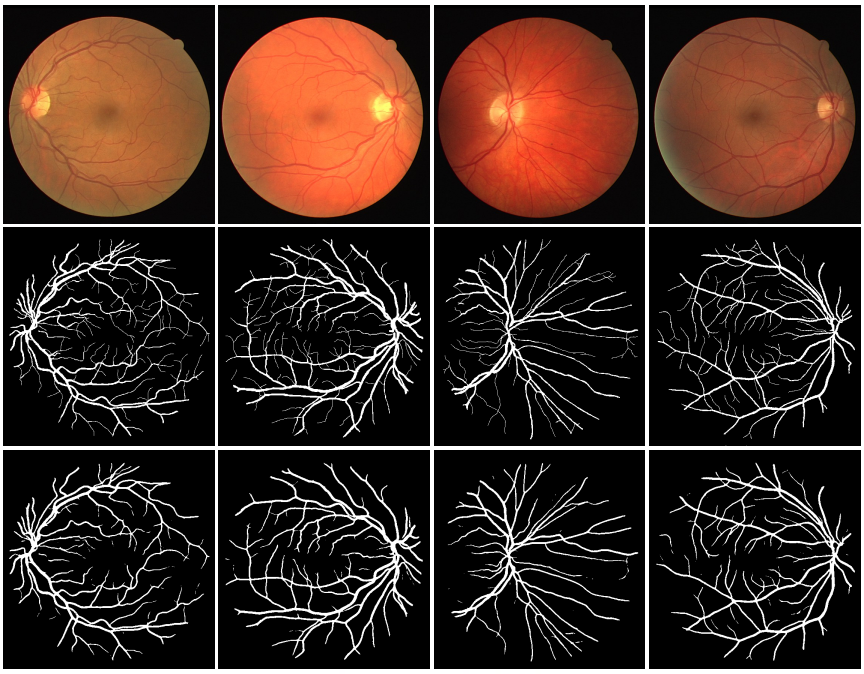


FIGURE 5.2: Typically qualitative results of the proposed RFA-BNet on several testing fundus images of DRIVE [80] dataset. Top row: Raw fundus images; Middle row: Ground truth; Bottom row: Results of the proposed RFA-BNet.

TABLE 5.1: Quantitative Results on DRIVE [80] dataset. **Boldface numbers** indicate the best performance of each measure. Note that (A) indicates Asymmetrically-structured Network and (S) stands for Symmetrically-structured Network.

Approach	Sensitivity	Specificity	Accuracy	AUROC
Liskowski *et al.* [65] (A)	0.7763	0.9768	0.9495	0.9720
Jiang *et al.* [44] (A)	0.7540	0.9825	**0.9624**	**0.9810**
Feng *et al.* [22] (S)	0.7811	**0.9839**	0.9560	0.9792
He *et al.* [31] (S)	0.7761	0.9792	0.9519	N/a
Baseline (w/o RFA) (A)	0.7807	0.9667	0.9484	0.9659
RFA-BNet (A)	**0.7932**	0.9741	0.9511	0.9732

5.3.3 Experimental Results and Analyses

As quantitatively shown in Table 5.1, compared to the baseline concept, the involvement of RFA scheme outperforms $0.0027 - 0.0125$ for all the measures. Moreover, the proposed RFA-BNet achieves state-of-the-art sensitivity (0.7932) among the compared methods. Meanwhile, the performance in terms of specificity, accuracy, and AUROC is still comparable to that of the patch-based methods with 0.9741 (< 0.0098 compared to the best performance reported in [22]), 0.9511

(< 0.0113 [44]), and 0.9732 (< 0.0078 [44]), respectively. Additionally, it can be observed from several typically qualitative results displayed in Fig.5.2 that irregular and diverse appearance of retinal blood vessel is carried out remarkably under various illumination conditions of input images compared to corresponding ground truth. These outcomes imply that the proposed architecture is able to effectively label challenging retinal vessel at pixel level without expensively utilizing patches augmented from the raw fundus photography.

In summary, this section introduced an approach using Round-wise Features Aggregation on Bracket-shaped convolutional neural networks for dealing with retinal blood vessel segmentation problem in color fundus image. The proposed method targets to efficiently infer pixel-wise labeled map without involving costly computation of generating patches from original color fundus image. For this objective, the Bracket-style decoding manner combining with comprehensive aggregation between decoded feature maps of highest-resolution enables the proposed RFA-BNet to identify vessels' location flexibly and precisely at pixel level as shown by the experimental results.

Chapter 6

Bracket-style Network Variant for Image Classification

In this chapter, at first, two domains of Diabetic Retinopathy detection and Facial Expression recognition with respectively existing issues in the literature are introduced. Then, descriptions of the adopted Bracket-style network variants are given to explain how they are applicable to tackle the above-mentioned natural and medical image classification domains. Afterwards, corresponding benchmark datasets, ablation studies as well as analyses of experimental results are in-turn presented to show the effectiveness of the proposed approaches.

6.1 Domain Overview

6.1.1 Diabetic Retinopathy Detection

Diabetic Retinopathy (DR) is the complication developed from long-term being affected by diabetes mellitus during a lengthy period, is among the leading causes of visual impairment and blindness [37]. Generally, the seriousness of DR is manually assessed through the aggravation of vasculature and occurrences of abnormal protrusions within retinal-related photograph. In concrete, the DR grades are decided by the combined evaluation of different structural features presented in the color fundus images, for instance, existence of microaneurysms, exudates, hemorrhages, neurodegeneration, retinal vascular complications [48, 50, 87, 90]. Traditionally, the severity of DR is determined based on the combined evaluation of different structural features presented in the color fundus images, for instance, existence of microaneurysms, exudates, hemorrhages, and neovascularization [32, 50, 87]. Accordingly, five severity grades comprising no apparent retinopathy (no DR), mild nonproliferative DR (NPDR), moderate NPDR, severe NPDR,

and proliferative DR (PDR) have been defined as international clinical classification system based on the Early Treatment Diabetic Retinopathy Study [74] (which are subject to aforementioned clinical criteria). However, such grading process remains time-consuming and challenging due to the heavy dependence on technical expertise and quality of the screened retinal fundus images. Moreover, it can be observed that effective diagnosis of DR severity level allows the ophthalmologists to deploy proper treatment procedure for the prevention of vision deterioration. These lead to the fact that the research topic of automatic DR detection from retinal-based images shows great interest in both ophthalmology and modern computer vision domains nowadays.

To this end, thanks to recently marvelous advancement of computational resources and image data regarding both quantity and quality, deep learning technique has been widely researched diverse fields, especially computer vision. Particularly, Convolutional Neural Network (CNN), a powerful DL architecture, has been applied in diverse vision-oriented areas ranging from natural image classification [29, 34, 79, 81], human action recognition [39, 40] to biomedicine [10, 88], medical image segmentation [36], and DR risk prognosis [37], to name a few. As a result, utilizing CNN to recognize DR severity scales from fundus images has also attracted numerous researches [2, 5, 15, 43, 45, 77, 84–86, 92, 106] thanks to the representational power of the above-mentioned biomarkers for automatic diagnosis. Particularly, the existing CNN-based methods are divided into two primary groups: (i) the employment of the CNN built by common trainable and non-linear layers for classification [15, 43, 77, 84, 85] and (ii) deep architecture of multiple network streams learning through ensemble scheme [45, 86, 92, 106].

Regarding the first line of work, the authors of [85], [84], [43], and [15] employed 11-, 17-, 18-, and 20-layer CNNs (which are commonly designed by sequential pipeline of convolution, rectified linear unit (*ReLU*) activation, max/average pooling, Fully Connected (*FC*) layers), respectively, for classifying corresponding DR grades. Similarly, in [77], the light-weight Inception-v3 architecture [82] is utilized using transfer learning tehnique. As for the second group, Vo et al. [86] introduced two modified versions of VGG [79] and GoogleNet [81], namely VNXK- and CKML-Net respectively, to recognize the DR grades. In addition, the authors considered L-, green-, and I_1-channel counterparts of the original fundus image as inputs of those two networks for performance boost. Meanwhile, a triplet of sub-CNNs [92], i.e., Main, Attention, and Crop Networks, was introduced for exhaustively examining multiple clinical details existing in the fundus photography. On the other hand, the methodology in [106] introduced two stages of patch-level learning

to comprehensively acquire fine-to-coarse details at multiple scales from the raw fundus photography for DR grade prediction. Besides that, the authors in [45] slightly modified the original ResNet-18 [29] by involving an additional attention stream to enhance inter-class discrimination. It is worth noting that large-scale dataset [47] was utilized in those studies for attaining an excellent performance of detecting the DR grades. In addition, the model training cost becomes a significant concern due to dealing with a large number of high-resolution images.

6.1.2 Facial Expression Recognition

Recently, extraordinary advancement of computing resources and visual data regarding both quantity and quality has facilitated deep learning technique to be widely applied into numerous areas, especially computer vision. To this end, CNN [29, 34, 38, 79], a well-known deep learning architecture, has attracted a great number of researchers thanks to its impressive performance enhancement in different recognition-based issues, such as human activity recognition [39], semantic scene understanding [35, 36], disease progression identification [37], and especially facial expression recognition (FER) [93].

In fact, FER has long been an active research field due to its diverse applications related to human-computer interactions [57]. Nowadays, with an increasing number of images collected from laboratory [68] and the wild [58], the power of CNN is exhaustively exploited in this image classification-related domain, of which further achievements are significantly accomplished. Particularly, there are three major CNN-based approaches proposed in the literature of FER: (i) ensemble of multiple deep networks [21, 24, 46]; (ii) algorithms of specialized objective function or statistical modules [1, 58, 61] attached to a conventional CNN; and (iii) attention mechanism embedded to pretrained CNNs [60, 70].

Since emotions via a person's face are represented by the combination of various muscular modalities (e.g., shape of eyes, eyebrow, nose, mouth, facial wrinkle, to name a few), several researches as shown in the first group aggregated multiple deep networks to express potentially facial features as well as contextual information for high recognition performance. In concrete, the authors in [24] took into account capsule, facial-attribute, and holistic-feature networks for coordinating spatial details with deep context smoothly throughout the whole architecture. Meanwhile, MRE-CNN [21] firstly divided the original input into multiple regions of interest based

on predefined facial landmarks, then fed those patches into different VGG-16 [79] models for ensemble learning. Another noticeable architecture, called ResiDen [46], is the mixture of two well-known concepts in deep learning-based computer vision, i.e., residual connections [29] and dense blocks [38] in a single network. Such combination style is capable of exhaustively maintaining important facial expression features and gradient signals in both feedforward and backpropagation stages. Obviously, expensive computation is the major limitation of these approaches. Hence, instead of involving additional sub-networks, methods in the second group mainly introduce locality preserving loss [58] or designated cluster loss [61] to minimize intra-category variation while maximize inter-category discrimination. Recently, SPDNet [1] offered specialized modules of covariance matrices for spatiotemporal pooling to combat distortion of facial landmarks during the learning process. However, the utilization of these objective functions or statistical modules sometimes results in trivial performance since certain discriminative features might not be focused properly. Accordingly, in order to express essential features extracted by trainable layers, attention scheme is of great interest in the third group. For instance, ACNN [60] introduced patch- and global-based attention networks to re-calibrate acquired feature responses at local regions and image level, respectively. On the other hand, FERAtt [70] involved an attention module with encoder-decoder structure to effectively reconstruct facial information from the output of a CNN-based feature extractor for further classification step. It can be realized that the attention mechanism is only applied to high-level feature maps in these techniques.

6.1.3 Common Problem Statement and Proposed Solution

Regarding the DR detection domain, it can be realized that the existing multi-stream networks are subject to costly computation while the remaining models do not involve multiple levels of semantic context in the constructed CNN. Particularly, the fact that multiple downsampling stages during feedforward process of the CNN leads to the loss of certain spatial correlations between the aforementioned DR-related signs, which are hardly encoded along depth dimension. Thus, it is hypothesized that only taking into account highest-level features for the classifier is insufficient in terms of predicting DR severity level.

With respect to the FER domain, existing multi-stream networks are subject to costly computation while attention-embedded models do not involve multiple levels of semantic context in a predefined CNN for FER. As aforementioned, the output emotion is represented by the fusion of different muscular modalities, which are exhaustively acquired at multiple levels by a CNN.

Therefore, manifold sub-sampling stages along feedforward pass of the CNN leads to the loss of certain spatial correlations between several facial tissues, which are hardly encoded in channel dimension. Consequently, it is hypothesized that only relying upon the outputs and corresponding attentional features of the deepest layer for the classifier is insufficient.

It can be realized that these two domains share the similar problem statement that the target label (DR severity grade or facial emotion class) heavily relies on the amalgamation of multi-scale structural details existing inside the considered image. Motivated by the aforementioned observations and hypotheses, a Single-mode Cross-Attentional Bracket-style CNN (sCAB-Net) is proposed to leverage the learnable integration of channel-wise attention at multi-level features in a pretrained CNN, which allows accomplishing superior recognition performance in a cost-effective way. Specifically, given that informative features are channel-wisely encoded from shallow to deep layers, densely embedding such semantically-rich details into the finer-grained patterns by the attention extractors (which are inspired from [34]) in a Bracket-style reversed manner is taken into account. Moreover, in order to smoothly coordinate the finely-patterned (low-level) and semantically-rich (high-level) features, a dense re-calibration procedure is taken into account. Consequently, the attachment of Channel-wisely Cross-Attentional (CCA) stream into the backbone CNN facilitates spatial representations of important DR-oriented factors (for the DR detection domain) as well as facial modalities (for the FER domain), which are comprehensively refined by semantic context of higher-level features ahead, to be comprehensively involved in the final prediction of given supervised labels. It is obvious that such effective aggregation scheme of various semantic information from the multi-level feature maps in a CNN is the principal key for recognizing corresponding DR severity level or facial emotion label more accurately.

Finally, as for the DR detection domain, the proposed sCAB-Net is evaluated using Kaggle DR detection dataset [47], of which the experimental results in terms of quadratic weighted kappa (QWK) are compared with the state-of-the-arts. Meanwhile, regarding the FER domain, RAF-DB dataset [58] is employed for the evaluation, of which the experimental results in terms of mean class accuracy (sum of diagonal elements in a confusion matrix) are also compared with novel approaches in the literature.

6.2 Descriptions of Bracket-style Network Variant for Image Classification

This section describes details of the proposed sCAB-Net, a variant of the original CAB-Net applied for image classification, with demonstrations in Fig. 6.1 and Fig. 6.2 corresponding to DR detection and FER, respectively. Generally, the proposed architecture is constructed by a backbone CNN associated with the stream of CCA. As demonstrated in Fig. 6.1 and Fig. 6.2, convolution blocks in the dashed box represent the fundamental components of the backbone CNN while the remaining sketch in green region illustrates the attention-oriented stream of amalgamating multi-level features for the classification of DR severity grade or facial emotion label.

6.2.1 Backbone CNN

Different backbone networks, i.e., VGG [79], ResNet [29], and DenseNet [38] are applied to prove the flexibility of the proposed attention-embedded stream with respect to diverse capacities of feature representation. In these classification networks, layers in each convolution block learn and perform acquired features at a specific scale corresponding to a relatively semantic level. For instance, both ResNet and DenseNet consist of four basis convolution blocks (as shown in Fig. 6.1 and Fig. 6.2), of which the final outputs have strides of 4, 8, 16, and 32 in comparison with the input's spatial size, respectively. Note that the total number of convolution and non-linear activation layers is varied in each block. Accordingly, in order to ensure the reasonable increment of computational cost, only four feature maps, which are ultimate outputs of the aforementioned learnable blocks, are taken into account for the stage of attentional features extraction. Meanwhile, since there is no explicit definition of convolution blocks in the VGG architecture, outputs of *ReLU* activation layers preceding the last four max-pooling layers, which also have same strides as specified previously, are chosen for further processes.

Obviously, spatial resolution of the extracted feature maps is reduced by half along the feed-forward flow between the convolution blocks while the corresponding depth size grows rapidly. Moreover, since the outcomes at later layers contain semantically-richer context in channel dimension compared to those obtained earlier, they can be utilized to recalibrate (i.e., strengthen the informative and weaken the less-productive) feature responses extracted at shallower layers in reversed fashion. Consequently, available ambiguities in spatial details of the considered low-level feature maps are eliminated thanks to the embedding of semantic information. Briefly, it

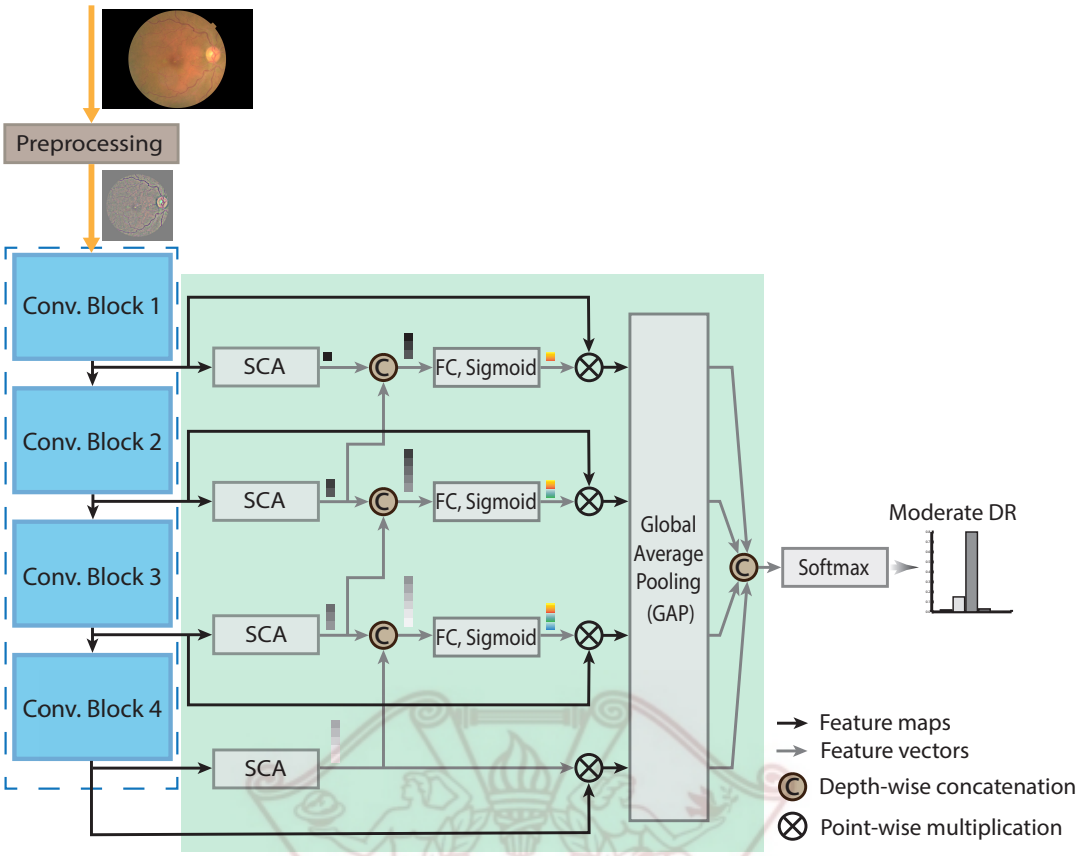


FIGURE 6.1: Architecture of the proposed sCAB-Net for DR severity classification. Given an input image fed into the backbone CNN containing series of predefined convolution blocks, final outputs of these blocks have strides of 4, 8, 16, and 32, respectively. Subsequently, these feature maps (namely *convmap-1*, *convmap-2*, *convmap-3*, *convmap-4*) are involved in the process of attention feature extraction. In brief, these fine-to-coarse feature maps (represented by black arrows) are densely combined via the Cross-Attentional Fusion modules to produce outputs, which continuously pass through the same procedure until one final prediction map is retrieved. As for the obtained segmentation map, every pixel is assigned an object class within the predefined number of training *classes*. Since every inferred feature map fuses with its adjacent finer-resolution map at each round and the total number of feature maps decreases by one round-by-round, such process is named Bracket-shaped network. Note that 'Conv. Block' and 'SCA' stand for the blocks of predefined *Convolutional* layers and Self-Context Aggregation, respectively. 'FC, Sigmoid' means *Fully Connected* layers followed by the *Sigmoid* activation function. View in color is recommended for the best visualization.

is advantageous to involve finely-patterned (high-resolution) feature maps, which possess well-organized representation of DR-oriented factors, with the semantically-rich (low-resolution) versions for higher recognition performance.

In other words, the CCA stream is coupled with the backbone CNN for leveraging the impact of finely-patterned features at earlier layers on the final prediction. Concretely, channel-wise semantic details of the higher-level features are utilized to enhance the informative responses while

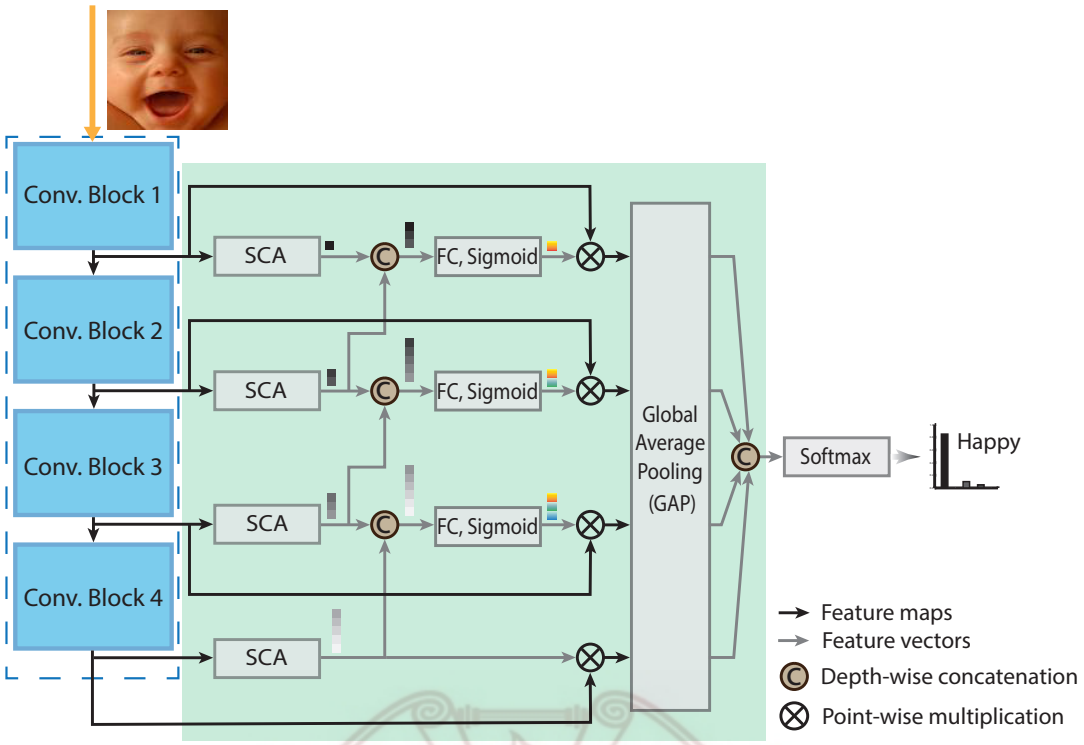


FIGURE 6.2: Architecture of the proposed sCAB-Net for facial expression recognition. Note that 'Conv. Block' and 'SCA' stand for the blocks of predefined *Convolutional* layers and Self-Context Aggregation, respectively. 'FC, Sigmoid' means *Fully Connected* layers followed by the *Sigmoid* activation function. View in color is recommended for the best visualization.

mitigating the less effective ones in feedback-like manner. As a consequence, such reverse refinement brings two noticeable benefits as follows. Firstly, it allows encoded features that rely on spatial representations to be extensively involved in the final classifier. This activity is achievable since finer-resolution (i.e., low-level) feature maps, of which the semantic information is much enhanced by higher-level context reversely, can be early engaged to the Softmax classifier without significant obscurity. Secondly, it acts as an extensive augmentation procedure at multiple feature levels because the sCAB-Net has an additional learning stream of backward and parallel styles besides the main feedforward path.

6.2.2 Channel-wisely Cross-Attentional (CCA) Stream

In general, the proposed CCA stream consists of three major components, i.e., (i) Self-Context Aggregation (SCA) inspired from Hu *et al.* [34], (ii) Bracket-style Attention (BsA), and (iii) Multi-level Fusion (MLF). Details are delivered as follows.

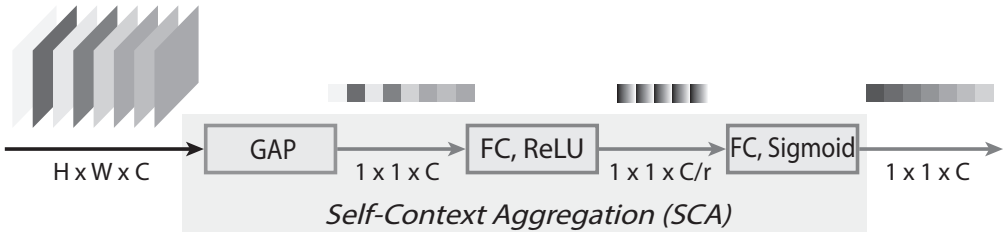


FIGURE 6.3: Functional layers in the Self-Context Aggregation module. 'GAP' signifies the *Global Average Pooling* layer.

Self-Context Aggregation At first, the four chosen feature maps (i.e., final output of the fundamental blocks of convolutional layers in ResNet) are fed into this SCA module for individually exploiting semantic context encoded along depth dimension. Let $\mathbf{F}_n$ denote those feature maps of interest, where $n = 1, \ldots, 4$ such that larger n indicates the higher-level features, which have semantically-richer context but smaller spatial size. Subsequently, the process of aggregating self-context shown in Fig. 6.3 is initially performed by a *Global Average Pooling (GAP)* layer, which is $\mathcal{G} : \mathbf{F}_n \in \mathrm{R}^{\mathrm{H_n x W_n x C_n}} \rightarrow \boldsymbol{g}_n \in \mathrm{R}^{\mathrm{C_n}}$. The corresponding formulation of $\mathcal{G}$ is defined as

$$\boldsymbol{g}_{n_c} = \mathcal{G}(\mathbf{F}_n) = \frac{1}{H_n \times W_n} \sum_{h=1}^{H_n} \sum_{w=1}^{W_n} \mathbf{F}_{n(h,w,c)} \tag{6.1}$$

where $h = 1, \ldots, H_n$; $w = 1, \ldots, W_n$; and $c = 1, \ldots, C_n$ are height, width, and channel coordinates of pixels in the considered feature maps $\mathbf{F}_n$, respectively.

Then, *FC* layers followed by ReLU activation are applied to exploit underlying cross-channel interactions of the retrieved vectors $\boldsymbol{g}_n$. Formally,

$$\begin{aligned} \boldsymbol{i}_n &= ReLU(\mathbf{W}_{fc1_n}^T \boldsymbol{g}_n + \mathbf{B}_{fc1_n}) \\ \boldsymbol{s}_n &= \sigma(\mathbf{W}_{fc2_n}^T \boldsymbol{i}_n + \mathbf{B}_{fc2_n}) \end{aligned} \tag{6.2}$$

where $\{\mathbf{W}_{fc1_n} \in \mathrm{R}^{\mathrm{C_n} \times \frac{\mathrm{C_n}}{\mathrm{r}}}, \mathbf{B}_{fc1_n} \in \mathrm{R}^{\frac{\mathrm{C_n}}{\mathrm{r}}}\}$ and $\{\mathbf{W}_{fc2_n} \in \mathrm{R}^{\frac{\mathrm{C_n}}{\mathrm{r}} \times \mathrm{C_n}}, \mathbf{B}_{fc2_n} \in \mathrm{R}^{\mathrm{C_n}}\}$ are respectively trainable parameters of two *FC* layers in use; $\boldsymbol{i}_n \in \mathrm{R}^{\mathrm{C_n}/\mathrm{r}}$ and $\boldsymbol{s}_n \in \mathrm{R}^{\mathrm{C_n}}$ are intermediate and final outputs of the SCA module, respectively; and $\sigma(.)$ symbolizes the *Sigmoid* activation function that weights vectors' entries from 0 to 1 based on corresponding utilities. It is noted that value of r, the compression rate for saving computational cost, is set to 16 following Hu *et al.* [34]. Besides that, the lengths C_n of $\boldsymbol{s}_n$, where $n = 1, 2, 3, 4$, are determined based on the ultimate output's channel size of the four fundamental convolution blocks. For instance, using VGG-16 as the backbone introduces $C_n = \{128, 256, 512, 512\}$ while the ResNet-101 counterpart give $C_n = \{256, 512, 1024, 248\}$.

Remarkably, in the original work [34], the output representational vector of this SCA module is subsequently used to re-calibrate its input feature map only at every layer, which can be referred to as intra-feature attention. Meanwhile, the counterpart in the proposed sCAB-Net is employed to further incorporate with the corresponding version at lower scale for performing both intra- and inter-feature (in a reversely cross manner as described at next sub-section) attention tasks. Another noteworthy difference is that the SCA blocks in the proposed model are only involved at the end of the four predefined convolution blocks in the backbone network.

TABLE 6.1: Lengths C of extracted attentional feature vectors g_i ($i = 1, 2, 3, 4$) with respect to different backbone CNNs.

Backbone CNN	g_1	g_2	g_3	g_4
VGG-16 [79]	128	256	512	512
ResNet-101 [29]	256	512	1024	2048
DenseNet-161 [38]	384	768	2112	2208

Note: These values of C are also identical to the depth size of corresponding feature maps $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$ extracted from the backbone networks.

Bracket-style Attention Previous step only introduces the utilization of intra-relationships across channels within each individual feature map taken into account. Accordingly, four attentional feature vectors g_1, g_2, g_3, and g_4 (demonstrated by outbound gray arrows of Att. Ext. modules in Fig. 6.1 and Fig. 6.2) corresponding to the four chosen feature maps $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$, and $\mathbf{F}_4$, respectively, are inferred by the above-mentioned attentional features extractor. With respect to various backbone models, the vectors g_i have different lengths of C as reported in Table 6.1. To this end, semantic inter-dependencies between the considered features by uniquely learning all pairwise concatenation of the self-context vectors, i.e., s_n and s_{n+1}, where $n = 1, 2, 3$, are additionally exploited. This allows deeper feature maps involved from the backbone CNN to enrich semantic representations onto the shallower counterparts reversely, which then suggests two advantages. Firstly, the refined low-level features, which possess high resolution, have stronger contributions since they can be alternatively applied as a shortcut to the final classifier. As a result, characterizations of small-sized factors related to early DR (e.g., microaneurysms, hemorrhages, or capillary abnormalities) or facial emotion (e.g., eyebrows, wrinkles at eyes and cheeks), which may certainly get loss at later layers due to spatial pooling operations, can be apprehended extensively to improve the recognition performance. Secondly, it is argued that incorporating a stream of manipulating multi-level features in reverse fashion can be considered as another intensive procedure

of feature-level augmentation for avoiding overfitting issue.

According to center part of the green region in Fig. 6.1 and Fig. 6.2, the workflow of this BsA module is formulated as follows.

$$\begin{aligned} \mathbf{F}_{bsa_4} &= \mathbf{F}_4 \otimes \boldsymbol{s}_4 \\ \mathbf{F}_{bsa_n} &= \mathbf{F}_n \otimes \sigma(\mathbf{W}_{fc3_n}^T (\mathcal{C}[\boldsymbol{s}_n, \boldsymbol{s}_{n+1}]) + \mathbf{B}_{fc3_n}) \end{aligned} \tag{6.3}$$

where $n = 1, 2, 3$ in this step; $\otimes$ refers to as the point-wise multiplication at each channel; and $\{\mathbf{W}_{fc3_n} \in \mathrm{R}^{(\mathrm{C_n}+\mathrm{C_{n+1}})\times \mathrm{C_n}}, \mathbf{B}_{fc3_n} \in \mathrm{R}^{\mathrm{C_n}}\}$ denote the parameters of the *FC* layers followed by another *Sigmoid* activation function. These learning layers manage the integration of features having semantically-richer information into those with finer representation of spatial-based details. Notably, such reverse combinations only take place in pairwise fashion to ensure reasonable increment of computational burden and refrain low-level self-context vectors from overwhelming acquisition of heterogeneous higher-level information. Then, the cross-context output vectors are utilized for re-calibrating the corresponding feature maps $\mathbf{F}_n$ via point-wise multiplication along channel dimension. Afterwards, the retrieved results, denoted as $\mathbf{F}_{bsa_n}$, are the finalized representatives of typical semantic and spatial scales adopted for multi-level learning by Softmax classifier in the proposed architecture.

Multi-level Fusion To this end, each $\mathbf{F}_{bsa_n}$ is fed into the GAP layer followed by channel-wise concatenation for gaining the mixture of multi-level context, which smoothly carries finely-patterned and semantically-rich features of DR-related factors. Such procedure is given as follows

$$\mathbf{F}_{final} = \mathcal{C}\left[\mathcal{G}(\mathbf{F}_{bsa_1}), \mathcal{G}(\mathbf{F}_{bsa_2}), \mathcal{G}(\mathbf{F}_{bsa_3}), \mathcal{G}(\mathbf{F}_{bsa_4})\right] \tag{6.4}$$

where $\mathbf{F}_{final}$ stands for the final features handled by the subsequent Softmax classifier. Eventually, the severity grading performance can be improved since DR-oriented clinical signs or FER-related appearances in various spatial scales are involved exhaustively and unambiguously thanks to the CCA stream.

6.3 Experiments on Diabetic Retinopathy Detection

6.3.1 Benchmark Dataset: Kaggle DR Detection [47]

Kaggle DR detection dataset [47] is used to evaluate the proposed methodology. It contains approximately 35,000 training, 11,000 validation (public test), and 43,000 private test images, which are categorized into five severity scales as aforementioned. Notably, all the color fundus images are supplied by EyePACS, a retinopathy screening platform.

6.3.2 Training Configurations

The proposed model and corresponding evaluations are implemented using Pytorch [73]. Initially, the procedure in [27] is adopted to preprocess the raw fundus images by re-scaling to a predefined radius and then subtracting local average color for suppressing the diverse difference of illumination and resolution in the dataset. Same as existing work, augmentation techniques such as randomly cropping to size of 448 $\times$ 448, horizontal and vertical flipping, and arbitrary rotation are also applied to the training batches, of which each includes 32 images. In addition, weight decay of 5e-4 is employed to generalize the proposed model intensively. In the proposed sCAB-Net, VGG-16, ResNet-101, and DenseNet-161 pretrained with ImageNet [76] are adopted as representatives of backbone CNN as aforementioned. Accordingly, the attached CCA scheme is shown to be capable of flexibly coordinating with various depth capacities of encoding features at multiple levels. As for the optimization phase, stochastic gradient descent with initial learning rate of 0.005 and momentum of 0.9 is applied. During training, the learning rate decreases by half after every 20 epochs. In total, 80 training epochs on one NVIDIA GTX 1080TI GPU is executed.

6.3.3 Ablation Study

To this end, three strategies, i.e., Baseline, AN, and sCAB-Net with respect to each backbone CNN, are experimented to benchmark the effectiveness of the proposed architecture. Notably, the Baseline means that the pretrained backbone CNN is finetuned end-to-end. Meanwhile, the AN refers to as the additional engagement of attentions at the end of basis convolution blocks but without the densely reversed stream. Finally, QWK measures of these strategies on the validation set are presented in Table 6.2.

TABLE 6.2: QWK on DR Kaggle [47] validation set with different types of backbone CNN and attention-embedded scheme.

Backbone CNN	Strategy			QWK (%)	Number of Parameters
	Baseline	AN	sCAB-Net		
VGG-16 [79]	✓			84.9	134.30M
		✓		85.4	14.81M
			✓	86.3	15.59M
ResNet-101 [29]	✓			85.4	42.51M
		✓		86.1	43.23M
			✓	86.7	47.36M
DenseNet-161 [38]	✓			85.5	26.49M
		✓		86.5	27.78M
			✓	86.9	39.56M

Apparently, AN and sCAB-Net with respect to different backbone CNNs show superior performance over the corresponding Baselines with higher QWK of 0.5-1.0% and 1.3-1.4%, respectively. This implies that the engagement of attention scheme at multi-scale features and subsequent depth-wise aggregation of corresponding outcomes are plausible in the scenario of classifying DR severity level. Also, it is argued that major reason is the aforementioned observation wherein different DR-related factors are captured at multiple levels in a CNN, from which the attention strategy can maintain those beneficial details intensively and efficiently for higher prediction performance.

Moreover, compared to AN, sCAB-Net is capable of boosting the QWK more 0.4% (in the case that VGG-16 is the backbone CNN), 0.6% (ResNet-101), and 0.9% (DenseNet-161). Accordingly, these improvements imply the advantage of exhaustively embedding deeper attentional feature vectors to recalibrate shallower features. In concrete, as previously mentioned in Section 6.2.2, the operator of dense concatenation in reversed manner further enables low-level features (which contain spatially informative details of DR-related factors) to be extensively involved in the final classifier.

Also, it can be observed that higher capacity of backbone CNN is able to produce better prediction performance. Particularly, sCAB-Net equipped with backbones of ResNet-101 and DenseNet-161 increases the QWK by 0.4% and 0.6%, respectively, compared to that employing VGG-16.

Regarding the computational costs enumerated in Table 6.2, the sCAB-Net with VGG as backbone can reduce the number of parameters by approximately 88.4% because of not involving expensive *FC* layers at backend of the baseline. Meanwhile, compared to original architectures of

TABLE 6.3: Comparison of QWK on Kaggle DR [47] test set.

Approach	QWK (%)
11-layer CNN [85]	76.7
SI2DRNet-v1 [15]	80.4
18-layer CNN [43]	85.1
Zoom-in-Net [92]	85.7
sCAB-Net (VGG-16)	**84.9**
sCAB-Net (ResNet-101)	**85.4**
sCAB-Net (DenseNet-161)	**85.6**

the ResNet, using the proposed scheme only increases the complexity by around 11.4%. Clearly, although the increment of parameters' amount is mainly caused by the stage of pair-wisely backward concatenation, it is worth gaining a QWK improvement of 0.4-0.9% as mentioned previously.

6.3.4 Comparisons with State-of-the-arts

For the comparison with other methods, the proposed sCAB-Net with three different backbone CNNs is evaluated by the Kaggle DR test set. The benchmark results reported in Table 6.3 show that of the proposed approach is competitive with the state-of-the-arts. Specifically, although Zoom-in-Net [92] achieves highest QWK, the superiority over the proposed sCAB-Net (which utilizes DenseNet-161 as backbone network) is insignificant (0.1%). It should be noted that their results are obtained from the expensive triplet of sub-CNNs and ensemble learning. In a nutshell, thanks to the dense attention of higher-level depth-wise features to spatially-rich details at lower levels in reversed scheme, which allows early involvement of finely-patterned features, the proposed architecture can achieve an impressive performance on such challenging dataset.

In summary, a CNN with densely reversed attention, i.e., sCAB-Net, has been introduced to effectively address the DR detection problem. Concretely, the proposed architecture enables finely-patterned (high-resolution) feature maps, which possess well-organized representation of DR-oriented factors, to be smoothly combined with the semantically-rich (low-resolution) counterparts for a better recognition performance. The key for such utilization is the dense embedding of a channel-wise attention mechanism into a pretrained CNN in reversed manner. As a consequence, experimental results have demonstrated consistent improvements of the proposed model, which is constructed from a baseline network to the involvement of multi-scale attentional features extractor and a further stream of densely reversed attention. In the future, sCAB-Net can be potentially extended for tackling other disease recognition problems besides detecting DR severity scale.

6.4 Experiments on Facial Expression Recognition

6.4.1 Benchmark Dataset: RAF-DB [58]

RAF-DB [58], standing for Real-world Affective Faces Database, is a large-scale dataset of in-the-wild facial expression. This database is challenging in the literature since its 30,000 images carries out a tremendous diversity of ages, genders and ethnicity, head poses, lighting conditions, occlusions, specialized manipulations, and so on. In this section, only the single-label set, i.e., each image exclusively indicates one of the seven basic classes of emotion (angry, disgust, fear, happy, neutral, sad, and surprise) is involved. Accordingly, 12,271 training and 3,068 testing images, which are prior cropped into the resolution of 100×100 around the regions of face, are involved for the designated experiments. Moreover, it should be noted that mean class accuracy (i.e., sum of diagonal elements in the resulting confusion matrix) is the golden metric to benchmark the classification performance due to the between-class imbalance issue stated in [58].

6.4.2 Training Configurations

The proposed model and corresponding evaluations are implemented using Pytorch [73] and Scikit-learn [3] frameworks, respectively. Same as existing work, the following augmentation schemes such as random change of hue and saturation, horizontal flipping, and rotation in range of (-20° , 20°) are applied to the training images with mini-batch size of 64. In addition, weight decay of 0.0005 is employed generalize the proposed model more robustly.

About the training stage, the initial learning rate is set at 0.005 and use Softmax loss to assess the quality of sCAB-Net's parameters given ground-truth labels. Then, in order to accordingly minimize the calculated loss with respect to those trainable parameters, the optimization procedure in [13], wherein stochastic gradient descent with momentum of 0.9 is utilized along with the 'poly'-style schedule of learning rate decay, is adopted. Notably, the training process takes place in 50 epochs on one NVIDIA 1080TI GPU.

6.4.3 Ablation Study

For the purpose of showing robustness of the proposed architecture regarding facial expression prediction, three different strategies, i.e., Baseline, AN, and sCAB-Net for each backbone network, are conducted. Note that the Baseline refers to as finetuning the pretrained model end-to-end. Meanwhile, the AN corresponds to the involvement of attentional features extractor at the end of

TABLE 6.4: Mean Class Accuracy on RAF-DB [58] test set with various settings of backbone CNN and attention strategy.

Backbone CNN	Strategy			Mean Class Accuracy (%)	Number of Parameters
	Baseline	AN	sCAB-Net		
VGG-16 [79]	✓			74.96	134.30M
		✓		77.35	14.81M
			✓	78.81	15.59M
ResNet-101 [29]	✓			77.10	42.51M
		✓		77.48	43.23M
			✓	79.33	47.36M
DenseNet-161 [38]	✓			77.21	26.49M
		✓		77.75	27.78M
			✓	79.37	39.56M

the basis convolution blocks but without densely backward concatenation scheme. Accordingly, quantitative performance of these strategies with different backbone CNNs on the testing images is reported in Table 6.4.

In general, AN and sCAB-Net outperform the baseline one 0.38-2.39% and 2.16-3.85%, respectively, for all backbone networks. This implies that the engagement of attention scheme at multi-scale features and subsequent depth-wise aggregation of corresponding outcomes are plausible in the scenario of facial expression identification. It is argued that major reason is the aforementioned observation wherein different muscular modalities are captured at multiple levels in a CNN, from which the attention strategy can maintain those beneficial details intensively and efficiently for higher prediction performance.

Moreover, regarding the effectiveness of the sCAB-Net compared to AN, the mean class accuracy is further improved 1.46% (in the case of using VGG-16 as backbone network), 1.85% (ResNet-101), and 1.62% (DenseNet-161). Such superior performance points out the importance of additionally integrating higher-level attentional feature vectors for recalibrating lower-level feature maps. As discussed in Section 6.2.2, the dense combination in backward manner helps the network flexibly express informative spatial features subject to multi-level semantic details along depth dimension.

It is also obvious that the greater capacity the core CNN has, the better performance is attained (but not significantly). Concretely, using ResNet-101 and DenseNet-161 as backbones yields the similar mean class accuracy of 79.33% and 79.37%, respectively, which are around 0.5% higher than that of employing VGG-16.

TABLE 6.5: Comparison of Mean Class Accuracy on RAF-DB [58] test set.

Approach	Mean Class Accuracy (%)
DLP-CNN [58]	74.20
3DMFA [61]	75.73
ResiDen [46]	76.54
MRE-CNN [21]	76.73
Capsule-based Net [24]	77.48
Double C*d*-LBP [78]	78.60
sCAB-Net (VGG-16)	**78.81**
sCAB-Net (ResNet-101)	**79.33**
sCAB-Net (DenseNet-161)	**79.37**

As for details of class-wise performance, confusion matrices of the proposed sCAB-Net corresponding to different backbone networks are further manifested in Fig. 6.4. All of these confusion matrices deliver common outcomes as follows. The prediction of happy feeling yields highest accuracy and that of neutral, sad, and surprise also gives remarkable performance. On the other hand, the expressions of disgust and fear are misclassified with neutral/sad and sad/surprise by an average rate of about 10%, respectively. It is argued that the learnable layers following the dense combination of attentional features (i.e., $\mathbf{W}_{fc3_1}$, $\mathbf{W}_{fc3_2}$, and $\mathbf{W}_{fc3_3}$ in (6.3)) have to trade-off unavoidable loss of concatenated semantic details, which leads to indistinguishable representations of facial modalities between the above-mentioned emotions.

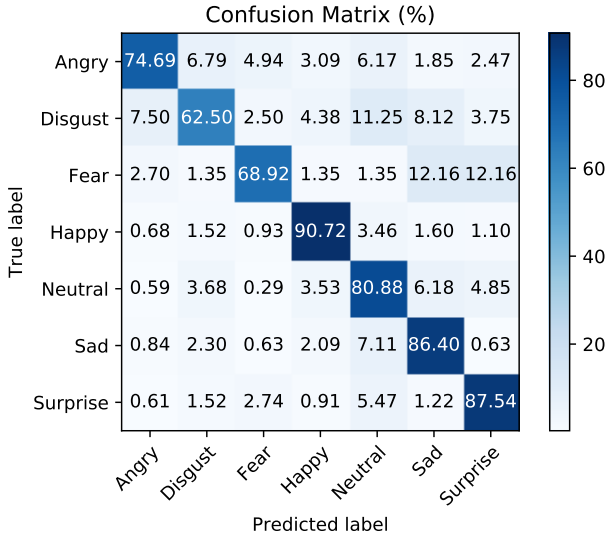
Regarding the computational costs enumerated in Table 6.4, the sCAB-Net with VGG as backbone can reduce the number of parameters by approximately 88.4% because of not involving expensive *FC* layers at backend of the baseline. Meanwhile, compared to original architectures of the ResNet, using the proposed scheme only increases the complexity by around 11.4%. Clearly, although the increment of parameters' amount is mainly caused by the stage of pair-wisely backward concatenation, it is worth gaining an improvement of 1.46-1.85% for mean class accuracy as aforementioned.

6.4.4 Comparison with State-of-the-art Methods

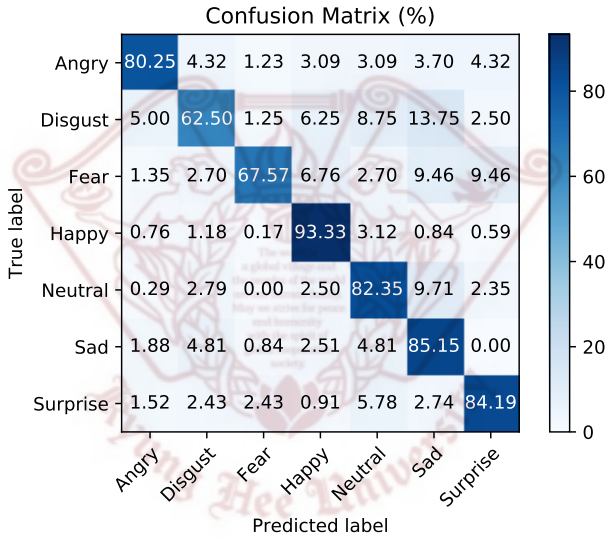
Through the quantitative comparison shown in Table 6.5, the proposed sCAB-Net achieves mean class accuracy competitive with that of the state-of-the-arts. In details, by only applying VGG-16 as the core network in the architecture, higher rates of 0.21-4.61% than those of the compared methods are attained. Furthermore, with deeper backbone networks like ResNet-101 or DenseNet-161, performance of the proposed approach is continuously improved, and reaching

state-of-the-art recognition rate. Clearly, such impressive performance on a challenging dataset expresses the advantage of aggregating low- and high-level features by the utilization of channel-wise attention mechanism in densely backward structure.

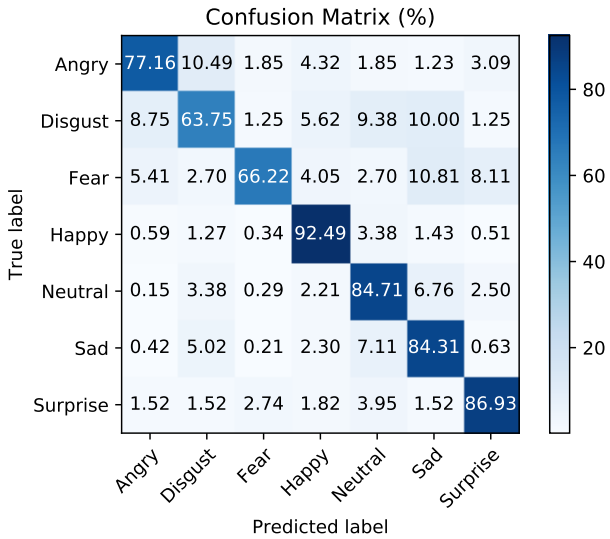
In summary, this section has introduced a cost-effective convolutional network with densely backward attention, namely sCAB-Net, for FER. The proposed approach aims to aggregate low- and high-level features in an efficient way according to the hypothesis that facial emotion is represented by the fusion of different muscular modalities extracted at multiple levels.For such purpose, attention mechanism is densely embedded in backward manner to a pretrained classification-based CNN for leveraging the performance of FER. The achievement of impressive experimental results enables the sCAB-Net to be widely applied in practical scenarios.



(A)



(B)



(C)

FIGURE 6.4: Confusion Matrices of the proposed sCAB-Net on RAF-DB dataset [58] with different backbone CNN: (a) VGG-16 [79], (b) ResNet-101 [29], (c) DenseNet-161 [38].

Chapter 7

Conclusions and Future Direction

7.1 Conclusions

This thesis introduced a novel approach towards semantic segmentation and classification of given images using Bracket-style CNN and its variants. Fundamentally, the proposed architecture is different from the existing works in the way it takes into account multi-level features acquired from the backbone CNN, wherein middle-scale representations are exploited exhaustively using predefined specialized attentional schemes. As such, it is flexible to customize the base concept as distinct variants for numerous problems comprising semantic segmentation in natural images (PASCAL VOC 2012 [20], CamVid [9], Cityscapes [18], and MS-COCO [64] datasets) and medical images (DRIVE [80]); as well as classification of facial emotion (RAF-DB [58]) and diabetic retinopathy severity (Kaggle DR Detection [47]). Notably, remarkable performance on those datasets suggested that the proposed methodology is capable of the two basis tasks of computer vision, i.e., image semantic segmentation and image classification, for further operations in practical perception-related applications.

Regarding the task of semantic segmentation, the proposed network in form of Bracket structure is able to coordinate with specialized modules called CAF to comprehensively incorporate the high-level (semantically-rich) features with the low-level (finely-patterned) counterparts. Such kind of cooperative procedure facilitates multi-level features extracted in the deep learning architecture to be continuously and thoroughly refined by various representations through a tournament of multiple decoding rounds. In particular, due to inherent characteristic of the bracket structure, the considered features in middle levels are incessantly utilized not only for enhancing semantic context in finer resolution but also for smoothing the appearances in coarser patterns. As a consequence, highly-qualified per-pixel segmentation map is achieved thanks to the exhaustive

exploitation of middle-level features for smoothly consolidating local details in the globally semantic context. Consequently, based on the impressively experimental results on public datasets like PASCAL VOC 2012 [20], CamVid [9], Cityscapes [18], and MS-COCO [64], the proposed architecture is potential to effectively interpret semantic classes from given images of daily life and/or street scenes for further operations in the practical perception-related applications.

Meanwhile, for the purpose of image classification, corresponding Bracket-shaped variants are proposed to facilitate the scenarios where spatial details have significant impacts on the predefined category of the concerned images (e.g., facial expression recognition, DR grading). In particular, as for the facial expression recognition problem, the emotions are decided by the amalgamation of different muscular modalities. Regarding the DR grading procedure, corresponding severity scales are concluded by the all-inclusive assessment of various structural biomarkers inside the fundus photograph. The operational principle for such advantage of the model in those specialized domains is briefly described as follows. Considering features of multiple scales extracted from a backbone deep learning network, outcomes at later layers contain semantically richer context in channel dimension compared to those obtained earlier. Thus, they can be utilized to recalibrate (i.e., strengthen the informative and weaken the less-productive) feature responses extracted at shallower layers in reversed fashion via the mechanism of channel-wise attention across feature maps of adjacent scales. Therefore, existent ambiguities in spatial details of the considered low-level feature maps are eliminated thanks to the embedding of semantic information. Subsequently, high-resolution spatial appearances with attended higher-level representations can be informatively involved in the multi-scale aggregation module followed by the final classifier.

7.2 Future Direction

To this end, the proposed methodology is shown to effectively tackle the two fundamental computer vision-related tasks, i.e., image semantic segmentation and image classification, in both generic and medical domains. For instance, common object (comprising groups of person, animal, vehicle, and indoor context) segmentation, street scene understanding, retinal blood vessel segmentation, facial expression recognition, and DR severity grading. It can be realized that inputs of the aforementioned problems are single and independent images per session of model inference.

But nonetheless, regarding the problems requiring inputs with temporal and/or sequentially-spatial information, the proposed idea in this dissertation is hardly applicable. Several examples of those problems are collective activity understanding, video captioning, video Question Answering, video classification, atomic actions, image and/or video colorization, to name a few. Clearly, in these research topics, the efficient processing of historical patterns in a continuous manner within a predefined temporal window length plays an essential role for ensuring a highly-qualified performance of the considered deep learning model. Meanwhile, the proposed deep architecture is primarily built from units of convolutional layers, which only share the learnable parameters across different locally receptive field and does not target at processing time-series information. Accordingly, the lack of operational manipulations for effectively learning temporal characteristics in the proposed technique yields unexpected recognition performance in those research areas.

On the other hand, there are still vacancies to further improve proficiency of the proposed Bracket-style CNN and its variants for more robust utilities in real-world practice. Firstly, as more operations are required to adapt the extensive utilization of middle-scale features in the proposed architecture and corresponding variants, it is quite challenging to meet the requirements of inference with very high frame rate or usage on mobile platforms. In such kind of context, constructing *fast* and *compact* versions of the proposed deep learning architecture is the next research objective to adapt various purposes based on trade-off prerequisites of accuracy, latency, and resource capacity. Accordingly, relevant research topics such as knowledge distillation, network pruning and/or quantization, neural architectural search, to name a few, shall be taken into account for that target. Secondly, since the final performance proportionally relies on the size of training dataset for any deep learning models in common, the strategy of unsupervised domain adaptation can be applied with the proposed Bracket-structured network to address the lack of well-labeled and big visual data. Note that annotations of image labels and especially pixel-wise categories are labor-intensive and time-consuming. As such, large-scale labeled data available from computer games or computer graphics programs can be used to train the deep learning model for pixel- and/or image-level classification of real-world images with same contents (but not largely annotated). Thirdly, besides the basis functions of image semantic segmentation and classification, the Bracket-style network concept is potential to additional manage more complicated tasks like object detection, panoptic segmentation (which performs instance and semantic segmentation simultaneously), image super-resolution, etc.

Bibliography

[1] D. Acharya et al. "Covariance Pooling for Facial Expression Recognition". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 480–4807. DOI: 10.1109/CVPRW.2018.00077.

[2] D. S. Ting, C. Y-L. Cheung, G. Lim et al. "Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images From Multiethnic Populations With Diabetes". In: *JAMA* 318.22 (Dec. 2017), pp. 2211–2223.

[3] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[4] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[5] V. Gulshan et al. "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus PhotographsAccuracy of a Deep Learning Algorithm for Detection of Diabetic RetinopathyAccuracy of a Deep Learning Algorithm for Detection of Diabetic Retinopathy". In: *JAMA* 316.22 (Dec. 2016), pp. 2402–2410. ISSN: 0098-7484.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495.

[7] Subhashis Banerjee, Sushmita Mitra, and B. Uma Shankar. "Automated 3D segmentation of brain tumor using visual saliency". In: *Information Sciences* 424 (2018), pp. 337 –353.

[8] Piotr Bilinski and Victor Prisacariu. "Dense Decoder Shortcut Connections for Single-Pass Semantic Segmentation". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2018, pp. 6596–6605.

[9] Gabriel J. Brostow et al. "Segmentation and Recognition Using Structure from Motion Point Clouds". In: *Computer Vision – ECCV 2008*. Springer Berlin Heidelberg, 2008, pp. 44–57.

[10] Chensi Cao et al. "Deep Learning and Its Applications in Biomedicine". In: *Genomics, Proteomics & Bioinformatics* 16.1 (2018), pp. 17 –32. ISSN: 1672-0229.

[11] L. Chen et al. "SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6298–6306.

[12] L. C. Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848.

[13] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018, pp. 833–851. ISBN: 978-3-030-01234-2.

[14] Liang-Chieh Chen et al. "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs". In: *3rd International Conference on Learning Representations, ICLR*. 2015.

[15] Y. Chen et al. "Diabetic Retinopathy Detection Based on Deep Convolutional Neural Networks". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 1030–1034. DOI: 10.1109/ICASSP.2018.8461427.

[16] Ning Cheung, Paul Mitchell, and Tien Yin Wong. "Diabetic retinopathy". In: *The Lancet* 376.9735 (2010), pp. 124 –136. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(09)62124-3.

[17] F. Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1800–1807.

[18] M. Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223.

[19] David Eigen and Rob Fergus. "Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture". In: *Proceedings of the 2015 IEEE*

International Conference on Computer Vision (ICCV). ICCV '15. IEEE Computer Society, 2015, pp. 2650–2658. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.304.

[20] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. 2012.

[21] Yingruo Fan, Jacqueline CK Lam, and Victor OK Li. "Multi-region ensemble convolutional neural network for facial expression recognition". In: *International Conference on Artificial Neural Networks*. Springer. 2018, pp. 84–94.

[22] Z. Feng, J. Yang, and L. Yao. "Patch-based fully convolutional neural network with skip connections for retinal blood vessel segmentation". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 1742–1746. DOI: 10.1109/ICIP.2017.8296580.

[23] Jun Fu et al. "Dual Attention Network for Scene Segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2019, pp. 3146–3154.

[24] S. Ghosh, A. Dhall, and N. Sebe. "Automatic Group Affect Analysis in Images via Visual Attribute and Feature Networks". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 1967–1971. DOI: 10.1109/ICIP.2018.8451242.

[25] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Vol. 9. Proceedings of Machine Learning Research. PMLR, 2010, pp. 249–256.

[26] Ian J. Goodfellow et al. "Generative Adversarial Nets". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada, 2014, 2672–2680.

[27] B. Graham. "Kaggle Diabetic Retinopathy Detection Competition Report". In: (2015).

[28] B. Hariharan et al. "Semantic contours from inverse detectors". In: *2011 International Conference on Computer Vision*. 2011, pp. 991–998.

[29] K. He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[30] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1026–1034. ISBN: 978-1-4673-8391-2.

[31] Q. He et al. "Multi-Label Classification Scheme Based on Local Regression for Retinal Vessel Segmentation". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 2765–2769. DOI: 10.1109/ICIP.2018.8451415.

[32] Y. He et al. "Segmenting Diabetic Retinopathy Lesions in Multispectral Images Using Low-Dimensional Spatial-Spectral Matrix Representation". In: *IEEE Journal of Biomedical and Health Informatics* 24.2 (2020), pp. 493–502.

[33] Hexiang Hu et al. "Recalling Holistic Information for Semantic Segmentation". In: *CoRR* abs/1611.08061 (2016). arXiv: 1611.08061.

[34] J. Hu, L. Shen, and G. Sun. "Squeeze-and-Excitation Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141.

[35] C.-H. Hua, T. Huynh-The, and S. Lee. "Convolutional Networks with Bracket-Style Decoder for Semantic Scene Segmentation". In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2018, pp. 2980–2985.

[36] C.-H. Hua, T. Huynh-The, and S. Lee. "Retinal Vessel Segmentation using Round-wise Features Aggregation on Bracket-shaped Convolutional Neural Networks". In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019, pp. 36–39.

[37] Cam-Hao Hua et al. "Bimodal Learning via Trilogy of Skip-connection Deep Networks for Diabetic Retinopathy Risk Progression Identification". In: *International Journal of Medical Informatics* (2019). ISSN: 1386-5056. DOI: https://doi.org/10.1016/j.ijmedinf.2019.07.005.

[38] G. Huang et al. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

[39] T. Huynh-The, C. Hua, and D. Kim. "Encoding Pose Features to Images With Data Augmentation for 3-D Action Recognition". In: *IEEE Transactions on Industrial Informatics* 16.5 (2020), pp. 3100–3111. ISSN: 1941-0050.

[40] Thien Huynh-The et al. "Image representation of pose-transition feature for 3D skeleton-based action recognition". In: *Information Sciences* 513 (2020), pp. 112 –126. ISSN: 0020-0255.

[41] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. 2015, pp. 448–456.

[42] M. A. Islam et al. "Gated Feedback Refinement Network for Dense Image Labeling". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4877–4885.

[43] S. M. S. Islam, Md M. Hasan, and S. Abdullah. "Deep Learning based Early Detection and Grading of Diabetic Retinopathy Using Retinal Fundus Images". In: *CoRR* abs/1812.10595 (2018). URL: http://arxiv.org/abs/1812.10595.

[44] Zhexin Jiang et al. "Retinal blood vessel segmentation using fully convolutional network with transfer learning". In: *Computerized Medical Imaging and Graphics* 68 (2018), pp. 1 –15. ISSN: 0895-6111. DOI: https://doi.org/10.1016/j.compmedimag.2018.04.005.

[45] P. Junjun et al. "Diabetic Retinopathy Detection Based on Deep Convolutional Neural Networks for Localization of Discriminative Regions". In: *2018 International Conference on Virtual Reality and Visualization (ICVRV)*. 2018, pp. 46–52. DOI: 10.1109/ICVRV.2018.00016.

[46] S. Jyoti, G. Sharma, and A. Dhall. "Expression Empowered ResiDen Network for Facial Action Unit Detection". In: *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*. 2019, pp. 1–8. DOI: 10.1109/FG.2019.8756580.

[47] "Kaggle: Diabetic Retinopathy Detection". In: (https://www. kaggle.com/c/diabetic-retinopathy-detection).

[48] S. S. Kar and S. P. Maity. "Automatic Detection of Retinal Lesions for Screening of Diabetic Retinopathy". In: *IEEE Transactions on Biomedical Engineering* 65.3 (2018), pp. 608–618. ISSN: 1558-2531.

[49] Y. M. Kassim, R. J. Maude, and K. Palaniappan. "Sensitivity of Cross-Trained Deep CNNs for Retinal Vessel Extraction". In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2018, pp. 2736–2739. DOI: 10.1109/EMBC.2018.8512764.

[50] Kiyoung Kim, Eung Suk Kim, and Seung-Young Yu. "Longitudinal Relationship Between Retinal Diabetic Neurodegeneration and Progression of Diabetic Retinopathy in Patients

With Type 2 Diabetes". In: *American Journal of Ophthalmology* 196 (2018), pp. 165 –172. ISSN: 0002-9394.

[51] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980.

[52] Ivan Kreso, Josip Krapac, and Sinisa Segvic. "Efficient Ladder-style DenseNets for Semantic Segmentation of Large Images". In: *CoRR* abs/1905.05661 (2019). arXiv: 1905.05661. URL: http://arxiv.org/abs/1905.05661.

[53] A. Kundu, V. Vineet, and V. Koltun. "Feature Space Optimization for Semantic Video Segmentation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3168–3175.

[54] T. H. N. Le et al. "Reformulating Level Sets as Deep Recurrent Neural Network Approach to Semantic Segmentation". In: *IEEE Transactions on Image Processing* 27.5 (2018), pp. 2393–2407.

[55] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[56] Hanchao Li et al. "Pyramid Attention Network for Semantic Segmentation". In: *British Machine Vision Conference 2018, BMVC*. 2018, p. 285.

[57] Shan Li and Weihong Deng. "Deep Facial Expression Recognition: A Survey". In: *CoRR* abs/1804.08348 (2018). arXiv: 1804.08348.

[58] Shan Li and Weihong Deng. "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Unconstrained Facial Expression Recognition". In: *IEEE Transactions on Image Processing* 28.1 (2019), pp. 356–370.

[59] Xiangtai Li et al. "GFF: Gated Fully Fusion for Semantic Segmentation". In: *CoRR* abs/1904.01803 (2019). arXiv: 1904.01803. URL: http://arxiv.org/abs/1904.01803.

[60] Y. Li et al. "Occlusion Aware Facial Expression Recognition Using CNN With Attention Mechanism". In: *IEEE Transactions on Image Processing* 28.5 (2019), pp. 2439–2450. ISSN: 1057-7149. DOI: 10.1109/TIP.2018.2886767.

[61] F. Lin et al. "Facial Expression Recognition with Data Augmentation and Compact Feature Learning". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 1957–1961. DOI: 10.1109/ICIP.2018.8451039.

[62] G. Lin et al. "RefineNet: Multi-Path Refinement Networks for Dense Prediction". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), pp. 1–1.

[63] T. Y. Lin et al. "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944.

[64] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *CoRR* abs/1405.0312 (2014).

[65] P. Liskowski and K. Krawiec. "Segmenting Retinal Blood Vessels With Deep Neural Networks". In: *IEEE Transactions on Medical Imaging* 35.11 (2016), pp. 2369–2380. ISSN: 0278-0062. DOI: 10.1109/TMI.2016.2546227.

[66] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. "ParseNet: Looking Wider to See Better". In: *CoRR* abs/1506.04579 (2015). arXiv: 1506.04579.

[67] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440.

[68] P. Lucey et al. "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. 2010, pp. 94–101. DOI: 10.1109/CVPRW.2010.5543262.

[69] K.K. Maninis et al. "Deep Retinal Image Understanding". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2016.

[70] Pedro D. Marrero Fernandez et al. "FERAtt: Facial Expression Recognition With Attention Net". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2019.

[71] Sara Moccia et al. "Blood vessel segmentation algorithms — Review of methods, datasets and evaluation metrics". In: *Computer Methods and Programs in Biomedicine* 158 (2018), pp. 71–91. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2018.02.001.

[72] Marin Orsic et al. "In Defense of Pre-Trained ImageNet Architectures for Real-Time Semantic Segmentation of Road-Driving Images". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2019, pp. 12607–12616.

[73] Adam Paszke et al. "Automatic Differentiation in PyTorch". In: *NIPS Autodiff Workshop*. 2017.

[74] "Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales". In: *Ophthalmology* 110.9 (2003), pp. 1677 –1682. ISSN: 0161-6420.

[75] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[76] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[77] J. Sahlsten et al. "Deep Learning Fundus Image Analysis for Diabetic Retinopathy and Macular Edema Grading". In: *Scientific Reports* 9.1 (2019), p. 10750. ISSN: 2045-2322. DOI: 10.1038/s41598-019-47181-w. URL: https://doi.org/10.1038/s41598-019-47181-w.

[78] F. Shen, J. Liu, and P. Wu. "Double Complete D-LBP with Extreme Learning Machine Auto-Encoder and Cascade Forest for Facial Expression Analysis". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 1947–1951. DOI: 10.1109/ICIP.2018.8451358.

[79] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR*. 2015.

[80] J. Staal et al. "Ridge-based vessel segmentation in color images of the retina". In: *IEEE Transactions on Medical Imaging* 23.4 (2004), pp. 501–509. ISSN: 0278-0062. DOI: 10.1109/TMI.2004.825627.

[81] C. Szegedy et al. "Going deeper with convolutions". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[82] C. Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826.

[83] Zhi Tian et al. "Decoders Matter for Semantic Segmentation: Data-Dependent Decoding Enables Flexible Feature Aggregation". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2019, pp. 3126–3135.

[84] J. Torre, A. Valls, and D. Puig. "A deep learning interpretable classifier for diabetic retinopathy disease grading". In: *Neurocomputing* (2019). ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2018.07.102. URL: http://www.sciencedirect.com/science/article/pii/S0925231219304539.

[85] M. C. A. Trivino et al. "Deep Learning on Retina Images as Screening Tool for Diagnostic Decision Support". In: *CoRR* abs/1807.09232 (2018). arXiv: 1807.09232.

[86] H. H. Vo and A. Verma. "New Deep Neural Nets for Fine-Grained Diabetic Retinopathy Recognition on Hybrid Color Space". In: *2016 IEEE International Symposium on Multimedia (ISM)*. 2016, pp. 209–215. DOI: 10.1109/ISM.2016.0049.

[87] Stela Vujosevic et al. "EARLY MICROVASCULAR AND NEURAL CHANGES IN PATIENTS WITH TYPE 1 AND TYPE 2 DIABETES MELLITUS WITHOUT CLINICAL SIGNS OF DIABETIC RETINOPATHY". In: *RETINA* 39.3 (2019), pp. 435–445.

[88] Michael Wainberg et al. "Deep learning in biomedicine". In: *Nature Biotechnology* 36 (Sept. 2018), 829 EP –.

[89] P. Wang et al. "Understanding Convolution for Semantic Segmentation". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2018, pp. 1451–1460.

[90] S. Wang et al. "Localizing Microaneurysms in Fundus Images Through Singular Spectrum Analysis". In: *IEEE Transactions on Biomedical Engineering* 64.5 (2017), pp. 990–1002. ISSN: 1558-2531.

[91] Shuangling Wang et al. "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning". In: *Neurocomputing* 149 (2015), pp. 708 –717. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2014.07.059.

[92] Z. Wang et al. "Zoom-in-Net: Deep Mining Lesions for Diabetic Retinopathy Detection". In: *Medical Image Computing and Computer Assisted Intervention, MICCAI 2017*. 2017, pp. 267–275. ISBN: 978-3-319-66179-7.

[93] M. Wu et al. "Weight-Adapted Convolution Neural Network for Facial Expression Recognition in Human-Robot Interaction". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2019), pp. 1–12. ISSN: 2168-2216. DOI: 10.1109/TSMC.2019.2897330.

[94] T. Wu et al. "Tree-Structured Kronecker Convolutional Network for Semantic Segmentation". In: *2019 IEEE International Conference on Multimedia and Expo (ICME)*. 2019, pp. 940–945.

[95] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition". In: *Pattern Recognition* 90 (2019), pp. 119 –133. ISSN: 0031-3203.

[96] M. Yang et al. "DenseASPP for Semantic Segmentation in Street Scenes". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3684–3692.

[97] C. Yu et al. "Learning a Discriminative Feature Network for Semantic Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1857–1866.

[98] Changqian Yu et al. "BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation". In: *Computer Vision – ECCV 2018*. 2018, pp. 334–349. ISBN: 978-3-030-01261-8.

[99] Fisher Yu and Vladlen Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *4th International Conference on Learning Representations, ICLR*. 2016.

[100] Aston Zhang et al. *Dive into Deep Learning*. https://d2l.ai. 2020.

[101] H. Zhang et al. "Context Encoding for Semantic Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7151–7160.

[102] X. Zhang et al. "Fast Semantic Segmentation for Scene Perception". In: *IEEE Transactions on Industrial Informatics* 15.2 (2019), pp. 1183–1192.

[103] H. Zhao et al. "Pyramid Scene Parsing Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6230–6239.

[104] Wang Zhe et al. "Learnable Histogram: Statistical Context Features for Deep Neural Networks". In: *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 246–262.

[105] Feng Zhou, Yong Hu, and Xukun Shen. "Scale-aware spatial pyramid pooling with both encoder-mask and scale-attention for semantic segmentation". In: *Neurocomputing* 383 (2020), pp. 174 –182. ISSN: 0925-2312.

[106] L. Zhou et al. "Deep multiple instance learning for automatic detection of diabetic retinopathy in retinal images". In: *IET Image Processing* 12.4 (2018), pp. 563–571. ISSN: 1751-9659.

Appendix A

List of Publications

SCI/SCIE Journal Papers:

[1] **Cam-Hao Hua**, Kiyoung Kim, Thien Huynh-The, Jong In You, Seung-Young Yu, Thuong Le-Tien, Sung-Ho Bae and Sungyoung Lee, "Convolutional Network with Twofold Feature Augmentation for Diabetic Retinopathy Recognition from Multi-modal Images", IEEE Journal of Biomedical and Health Informatics (SCI, IF:5.772), vol. 25, no. 7, pp. 2686-2697, July 2021.

[2] Mugahed A. Al-antari, **Cam-Hao Hua**, Jaehun Bang and Sungyoung Lee , "Fast deep learning computer-aided diagnosis of COVID-19 based on digital chest x-ray images", Applied Intelligence (SCI, IF:3.325), Vol. 51, pp. 2890-2907, 2021.

[3] **Cam-Hao Hua**, Thien Huynh-The, Sung-Ho Bae and Sungyoung Lee, "Cross-Attentional Bracket-shaped Convolutional Network for Semantic Image Segmentation", Information Sciences (SCI, IF:5.524), Vol.539, pp.277-294, 2020.

[4] **Cam-Hao Hua**, Thien Huynh-The, Kiyoung Kim, Seung-Young Yu, Thuong Le-Tien, Gwang Hoon Park, Jaehun Bang, Wajahat Ali Khan, Sung-Ho Bae, and Sungyoung Lee, "Bimodal Learning via Trilogy of Skip-connection Deep Networks for Diabetic Retinopathy Risk Progression Identification", International Journal of Medical Informatics (SCI, IF:2.731),Vol.132 , 2019.

[5] Thien Huynh-The, **Cam-Hao Hua**, Anh Tu Nguyen, Taeho Hur, Jaehun Bang, Dohyeong Kim, Muhammad B. Amin, Byeong Ho Kang, Hyonwoo Seung, Soo-Yong Shin, Eun-Soo Kim, Sungyoung Lee, "Hierarchical Topic Modeling With Pose-Transition Feature For Action Recognition Using 3D Skeleton Data", Information Sciences (SCI, IF:4.832), Vol.444, pp.20-35, 2018.

[6] Thien Huynh-The, **Cam-Hao Hua**, Anh Tu Nguyen, Taeho Hur, Jaehun Bang, Dohyeong Kim, Muhammad Bilal Amin, Byeong Ho Kang, Hyonwoo Seung and Sungyoung Lee, "Selective Bit Embedding Scheme For Robust Blind Color Image Watermarking", Information Science (SCI, IF:4.832), Vol.426, pp.1-18, 2018.

International Conference Papers:

[1] Mugahed A. Al-antari, **Cam-Hao Hua**, Jaehun Bang, Dong-ju Choi, Sun Moo Kang and Sungyoung Lee, "A Rapid Deep Learning Computer-Aided Diagnosis to Simultaneously Detect and Classify the Novel COVID-19 Pandemic", IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES 2020), Langkawi, Malaysia (Best Paper Awarded), March 1-3, 2021.

[2] **Cam-Hao Hua**, Thien Huynh-The and Sungyoung Lee, "DRAN: Densely Reversed Attention based Convolutional Network for Diabetic Retinopathy Detection", 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Montreal, QC, Canada, July 20-24, 2020.

[3] **Cam-Hao Hua**, Thien Huynh-The, Hyunseok Seo, and Sungyoung Lee, "Convolutional Network with Densely Backward Attention for Facial Expression Recognition", The 14th International Conference on Ubiquitous Information Management and Communication (IMCOM 2020), Taichung, Taiwan, Jan 3-5, 2020.

[4] **Cam-Hao Hua**, Thien Huynh-The and Sungyoung Lee, "Retinal Vessel Segmentation using Round-wise Features Aggregation on Bracket-shaped Convolutional Neural Networks", 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, July 23-27, 2019.

[5] **Cam-Hao Hua**, Thien Huynh-The and Sungyoung Lee, "Convolutional Networks with Bracket-style Decoder for Semantic Scene Segmentation", 2018 IEEE Conference on System, Man and Cybernetics (SMC), Oct 7-10, 2018.

[6] Thien Huynh-The, Sungyoung Lee, and **Cam-Hao Hua**, "ADM-HIPaR: An Efficient Background Subtraction Approach", 2017 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2017), Lecce, Italy, Aug 29-Sep 1, 2017.

[7] Thien Huynh-The, **Cam-Hao Hua** and Sungyoung Lee, "Improving NIC Algorithm Using Different Binary Structure Elements For Multi-modal Foreground Detection", In Proceedings of

the 10th International Conference on Ubiquitous Information Management and Communication (IMCOM '17), Beppu, Japan, Jan 5-7, 2017.

Patent Registration:

[1] Sungyoung Lee, **Cam-Hao Hua**, "IMAGE SEGMENTATION METHOD AND APPARATUS, AND COMPUTER PROGRAM THEREOF", Applicant: Kyung Hee University Industry-Academic Cooperation Foundation, Registration Number: (US)11,145,061, October 12, 2021.

[2] Sungyoung Lee, **Cam-Hao Hua**, "BRACKET-STYLE CONVOLUTIONAL NEURAL NETWORKS FOR SEMANTIC IMAGE SEGMENTATION", Applicant: Kyung Hee University Industry-Academic Cooperation Foundation, Registration Number: (JP)6890345, May 27, 2021.

[3] Sungyoung Lee, **Cam-Hao Hua**, "METHOD, APPARATUS AND COMPUTER PROGRAM FOR IMAGE SEGMENTATION", Applicant: Kyung Hee University Industry-Academic Cooperation Foundation, Registration Number: (KR)10-2215757, February 8, 2021.