The Application of CNN and Hybrid Networks in Medical Images Processing and Cancer Classification

The Application of CNN and Hybrid Networks in Medical Images Processing and Cancer Classification

By

Yuriy Zaychenko, Galib Hamidov
and Bohdan Chapaliuk

Cambridge
Scholars
Publishing

The Application of CNN and Hybrid Networks in
Medical Images Processing and Cancer Classification

By Yuriy Zaychenko, Galib Hamidov and Bohdan Chapaliuk

This book first published 2023

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-5275-1539-7
ISBN (13): 978-1-5275-1539-0

TABLE OF CONTENTS

PREFACE

This book is devoted to the actual problem of information technology applications and artificial intelligence methods for medical image processing, tumour detection, and cancer classification. It is structured into six chapters.

In Chapter One, the book discusses the use of convolutional neural networks (CNN), the most efficient modern technology in the problem of medical image processing and analysis. The chapter describes and analyses the idea of convolution and the main operators of CNN, including convolution, pooling and their implementation. Additionally, the book describes and analyses the architecture of modern CNNs, such as VGG16, VGG19, ResNet50, Inception, and Xception, comparing and analysing their properties.

This chapter pays much attention to machine learning methods for CNN, including the stochastic gradient descent method, RMSProp, Adam, and others.

Chapter Two focuses on breast tumour classification and cancer detection. The chapter starts with an introduction that presents a review of known works on the problem of breast cancer classification. The next section describes the standard dataset BreakHis, which is specially developed for breast tumour analysis and cancer classification.

The chapter describes investigations of the most popular CNN architectures for binary and multiclass classification of breast cancer, analysing and comparing their efficiency. Consequently, the most efficient architecture of modern CNN is determined.

Chapter Three introduces the hybrid CNN FNN, which uses the fuzzy neural network NEFClass for feature extraction and the CNN VGG for breast tumour classification. The chapter presents an experimental investigation of the suggested hybrid CNN using the BreakHis dataset. The chapter analyses the classification performance of the model and compares it to that of SVM and random forest classifiers. Moreover, the chapter discusses the problem of dimension reduction in breast tumour images and proposes the use of principal components as a potential solution. The effectiveness of this method is also investigated.

Chapter Four is devoted to the problems of increasing the sensitivity of breast cancer detection compared to known works and reducing the training time of CNN. The chapter introduces a novel hybrid CNN architecture that

employs a convolutional encoder for feature extraction and image dimension adjustment while using CNN ResNet121 as a tumour classifier. The chapter further presents the experimental investigation of the suggested hybrid CNN and compares it with known CNN architectures confirming the higher sensitivity (recall) and reduced training time of the developed hybrid CNN. All results presented in this chapter are new and have never been published before.

Chapter Five is dedicated to the problem of lung cancer detection. The chapter consists of four sections providing comprehensive information on building lung cancer detection systems. The first section is divided into three subsections describing approaches to building a lung cancer detection system. This section also contains information about related works and trained convolutional neural networks (CNN), a comparison of the different approaches, and a discussion of which provides the best accuracy in this chapter's research. The second section explains the types of data typically used in lung cancer screening systems and introduces two core datasets – LUNA and Data Science Bowl 2017 – that were used to train neural networks for this study. The third section describes common pre-processing steps used to train CNN and provides a comparison of the obtained results. The final section presents an analysis of the different approaches that utilize 2D and 3D CNNs for processing lung images and detecting cancer, followed by a conclusion on their perspectives. The state-of-the-art lung cancer detection system uses 3D CNNs to build a detection system pipeline. This system works with a CT scan's 3D representation of the lungs to recognize spatial information about malignant masses in the lungs. Usually, the lung cancer detection task is split into smaller tasks: segmentation and classification. However, the best model also adds more components that generate features from a specific pulmonary nodule's global or time context, thereby providing more useful information in determining cancer presence. This flexibility makes the detection system more robust, reduces the required dataset size, and can lead to human-level performance.

Chapter Six is devoted to brain tumour analysis and cancer detection. The chapter focuses on the medical image analysis of brain tumours and the classification of detected tumours into two categories: benign and malignant. To achieve this, a hybrid CNN-FNN was developed wherein CNN VGG16 was used for feature extraction and FNN ANFIS for the classification of detected tumours. To train ANFIS, the authors implemented the adaptive stochastic gradient method and evaluated its efficiency. The experimental investigations of the suggested hybrid CNN-ANFIS network were conducted on the specialized dataset of Brain MRI images for brain tumour detection. The efficiency of the suggested CNN-

ANFIS network for medical image recognition was confirmed by comparing its classification accuracy with other methods, such as CNNs, SVMs, and NNs. The results showed that the application of hybrid networks is efficient for medical image recognition.

Overall, this book focuses on the issue of processing medical images for detecting human cancer and developing diagnostic systems as the ultimate goal. The book presents innovative approaches for improving the efficiency of cancer detection in comparison to existing methods and CNN structures. The main idea proposed in the book is to develop a new class of CNN structures known as hybrid CNNs, which incorporate various classes of neural networks within a general CNN architecture. Through extensive experimental investigations, it was found that the hybrid CNN approach outperformed known CNN architectures in terms of various classification criteria.

Notably, most presented results are original and have never been published before in monographs.

INTRODUCTION

Now cancer constitutes the greatest problem for health defence all over the world. Based on the data of the International Agency of Cancer Research (IARC), 8.2 million death cases were registered in 2012, and 27 million new cases of illness are expected by 2030 [1]. Among the various types of cancer, breast cancer ranks second in terms of its frequency of occurrence in women. Furthermore, the mortality rate for breast cancer is significantly higher when compared to other types of cancer [1].

Annually, the American Cancer Society (ACS) estimates the number of new cancer cases and deaths in the United States. They also compile the most recent data on population-based cancer occurrence and outcomes. As per the most recent ACS projection for 2022, the United States is expected to experience 1,918,030 new cancer cases and 609,360 cancer-related deaths. This is the equivalent of about 5250 new cases daily, including approximately 350 deaths per day from lung cancer, which is the leading cause of cancer death [2]. Furthermore, it is anticipated that approximately 51,400 women will be diagnosed with ductal carcinoma in situ of the breast, 97,920 with melanoma in situ of the skin, 236,740 with lung cancer, and 117,910 will experience fatal outcomes.

Nowadays, information technologies are widely utilized at all stages of medical diagnostics. The main goals of automated medical systems are to expand the scope of practical tasks that can be performed with the aid of computers and to enhance the level of intelligent decision support provided to doctors, particularly during rapid diagnostics based on analysis of medical images of human tissue acquired through various sources such as MRI, CT scans, X-rays, among others.

In medical diagnostic problems, a significant portion of the challenge lies in extracting relevant features for further processing and selecting appropriate features for the classification method. The demand for training algorithms is rising with the advancement and widespread dissemination of decision-support systems. The reliability and simplicity of applications influence the speed and quality of decision-making, which is crucial in the context of rapid medical diagnostics. The advantages of medical diagnostics systems are speed, automation, and reliability, rendering them very comfortable tools for express medical diagnostics. Despite medical informatics being a relatively young field of no more than 30 years,

information technologies quickly penetrated various spheres of medicine and healthcare (family medicine, insurance medicine, building unified information space, integration in European medical space, and so on).

Despite the progress achieved by diagnostic technologies, pathologist-anatomists continue to be responsible for conducting the final diagnosis of breast cancer, including tumour classification and diagnosis, by visually analysing histological patterns through a microscope. However, recent advancements in image processing technologies and machine learning have facilitated the development of automatic detection and diagnostic systems, which can assist pathologist-anatomists in making accurate diagnoses and accelerate their work. Image analysis systems for automatic cancer diagnostics often prioritise the classification of histopathology images into different patterns corresponding to cancerous and non-cancerous states of tissue.

To date, several models and methods have been developed for breast cancer detection using various machine learning algorithms. Utilizing AI technologies such as neural networks and SVMs [2],[3], these methods have achieved diagnostic accuracies ranging from 76% to 94% on datasets containing 92 images.

Zhang and others [4] proposed a cascade classifiers approach for breast cancer detection. In this approach, the classifiers first reject easy cases that are evidently non-cancerous, and the remaining cases are passed to the second level, which employs a more sophisticated classification system for further analysis. This method was applied to a database of 361 images from the Israel Technological Institute and achieved an accuracy of 97%. Most recent papers in the breast cancer classification field focus on classifying digital images [3–6]. However, the widespread implementation of breast image classification (BIC) and other forms of digital pathology faces challenges such as the high cost of implementation, insufficient productivity for a huge number of clinical procedures, internal technological issues, and opposition from the pathologist-anatomist. Until now, most studies based on the histological analysis of breast cancer have been performed on small datasets. Some improvement presents a dataset with 7909 breast images obtained from 82 patients [7]. In this research, the authors estimated various texture descriptors and various classifiers and conducted the experiments with 82% to 85% accuracy.

One of the widely-used approaches to image processing and classification includes texture descriptors. In [7], authors used various texture descriptors and classifiers to detect breast cancer, achieving an accuracy range of 82% to 85%. Based on the results presented in [7] and some other investigations, one can conclude that texture descriptors may

propose a good solution for image processing. However, one drawback of texture descriptors is that the traditional approach to feature detection based on descriptors requires significant effort and high-level expertise, which is often task-specific, making it challenging to apply the same method in other cases.

Another approach to medical image processing and classification is the application of machine learning. Some researchers believe that the main weakness of modern machine learning methods lies in their inability to extract and organise discriminative information from the input data. It means that machine learning algorithms should be less dependent on feature engineering and capable of extracting and organising discriminative information directly from medical images, thus learning representations [8].

The idea of learning representations is not new, but it can now be implemented with the advent of GPUs (Graphics Processing Units), which are capable of providing high-speed performance at a relatively low cost due to their parallel architecture.

The alternative to these approaches is the application of CNN for medical image processing and diagnostics. It has been demonstrated that CNNs can overcome conventional texture descriptors [9],[10].

Nowadays, CNN is a powerful tool for medical image processing, tumour detection, and classification. Different types of CNNs have been developed, differing in the number of layers, the main principles of feature extraction and training algorithms, and other factors. Their structure typically consists of two main parts: convolutional and pooling layers for extracting informative features and fully-connected layers for image classification.

The main advantages of CNNs are:

- Universality, capability to process and analyse different types of images of various modalities, including optical, MRT, CT, X-rays, etc.;
- Capability to process images in both 2D and 3D;
- A set of efficient training algorithms.

However, there are some drawbacks to using CNNs. As images usually have high dimensions, large datasets of labelled images are required to train CNNs effectively. Therefore, for many types of tumour classification problems, there may be a lack of available information for training. Additionally, due to the large number of layers in a CNN, the problem of overfitting can occur, and special regularisation methods need to be applied to prevent it.

Despite these drawbacks, CNNs have been successfully applied for medical image processing, express diagnostics, and cancer classification in the breast [4, 5, 6, 8], lungs [10-14], and brain [15, 16].

The main challenges in applying CNNs for tumour classification and cancer detection are improving classification accuracy, increasing sensitivity for detecting malignant tumours, and reducing training time for CNNs. The authors of this book address these challenges by developing and researching new CNN structures, new methods of image dimension reduction, and more efficient training algorithms.

The main goals of this book are to present new approaches for improving the accuracy and sensitivity of cancer classification in the breast, lungs, and brain using new CNN structures, including hybrid CNNs, and to compare the results with those of well-known works.

References

[1] P. Boyle and B. Levin, Eds., World Cancer Report 2012, Lyon: IARC, 2012. [Online]. Available: http://www.iarc.fr/en/publications/pdfsonline/wcr/2008/wcr_2012.pdf .

[2] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, “Cancer statistics, 2022,” CA: A Cancer Journal for Clinicians, vol. 72, no. 1, pp. 7–33, 2022. https://doi.org/10.3322/caac.21708.

[3] S. R. Lakhani and S. Schnitt, WHO Classification of Tumours of the Breast, 4th ed., Lyon: WHO Press, 2012.

[4] Y. Zhang, B. Zhang, F. Coenen, and W. Lu, "Breast Cancer Diagnosis from Biopsy Images with Highly Reliable Random Subspace Classifier Ensembles," Machine Vision and Applications, vol. 24, pp. 1405-1420, 2012. https://doi.org/10.1007/s00138-012-0459-8.

[5] Y. Zhang, B. Zhang, F. Coenen, J. Xiao, and W. Lu, "One-class kernel subspace ensemble for medical image classification," EURASIP Journal on Advances in Signal Processing, vol. 2014, pp. 1-13, 2014. https://doi.org/10.1186/1687-6180-2014-17.

[6] S. Doyle, S. C. Agner, A. Madabhushi, M. D. Feldman, and J. E. Tomaszeweski, "Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features," in Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008, pp. 496-499. https://doi.org/10.1109/ISBI.2008.4541041.

[7] Y. Bengio, A. C. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pp. 1798-1828, 2012.

https://doi.org/10.1109/TPAMI.2013.50.

[8] A. Singh, H. Mansourifar, H. Bilgrami, et al., "Classifying Biological Images Using Pre-trained CNNs," Available: https://docs.google.com/document/d/1H7xVK7nwXcv11CYh7hl5F6pM0m218FQloAXQODP-Hsg/edit?usp=sharing.

[9] F. A. Spanhol, L. Oliveira, C. Petitjean, and L. Heutte, "A Dataset for Breast Cancer Histopathological Image Classification," IEEE Transactions on Biomedical Engineering, vol. 63, pp. 1455-1462, 2016. https://doi.org/10.1109/TBME.2015.2496264.

[10] D. Ardila, A. P. Kiraly, S. Bharadwaj, B. Choi, J. J. Reicher, L. Peng, D. Tse, M. Etemadi, W. Ye, G. Corrado, D. P. Naidich, and S. Shetty, "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography," Nature Medicine, vol. 25, pp. 954-961, 2019. https://doi.org/10.1038/s41591-019-0447-x.

[11] T. D. Bui, J. Shin, and T. Moon, "3D Densely Convolutional Networks for Volumetric Segmentation," ArXiv, abs/1709.03199, 2017. [Online]. Available: https://arxiv.org/abs/1709.03199.

[12] F. Liao, M. Liang, Z. Li, X. Hu, and S. Song, "Evaluate the Malignancy of Pulmonary Nodules Using the 3-D Deep Leaky Noisy-OR Network," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, pp. 3484-3495, 2019. https://doi.org/10.1109/TNNLS.2019.2892409.

[13] W. Zhu, C. Liu, W. Fan, and X. Xie, "DeepLung: Deep 3D Dual Path Nets for Automated Pulmonary Nodule Detection and Classification," in Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 673-681. https://doi.org/10.1109/WACV.2018.00079.

[14] A. A. Setio et al., "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge," Medical Image Analysis, vol. 42, pp. 1-13, 2017. https://doi.org/10.1016/j.media.2017.06.015.

[15] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks," arXiv preprint arXiv:1705.03820, 2017. [Online]. Available: https://arxiv.org/abs/1705.03820.

[16] P. Arya and A. K. Malviya, "A Survey on Brain Tumor Detection and Segmentation from Magnetic Resonance Image," Tissue Engineering (Topic), 2019. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3350289.

CHAPTER ONE

CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks are one of the types of neural networks of forward propagation. Convolutional neural networks are used to process data with a network topology. Examples of such data include time series, which can be considered a one-dimensional grid that takes samples at regular intervals, and images, which can be considered a two-dimensional grid of pixels. Convolutional neural networks have been incredibly successful in practice. They owe their name to the use of the mathematical *convolution operation*.

Convolution is a special kind of linear operation. Convolutional neural networks are neural networks that use convolution instead of traditional matrix multiplication in at least one of their layers [1].

1.1. The Idea of Convolution

In general, a convolution is an operation on two argument functions with real values. Convolution is a mathematical operation that transmits a signal through a linear, time-invariant system or filter [1]. Suppose we have a signal $x(t)$ passing through a system with a pulse response $w(t)$, resulting in the output signal $s(t)$ that is obtained by convolving $x(t)$ with $w(t)$. Convolution is simply an integral of the product of two functions, where one function is reversed in time and then shifted across the other function:

$$s(t) = \int_0^\infty x(a)w(t-a)da$$

This variant of the record is called the convolution operator and is denoted as follows:

$$s(t) = (x * w)(t)$$

In convolutional neural network terminology, the function x is called the input, and the function w is called the *kernel*. The result of the operation is called a map of features.

In a discrete form, the convolution operation can be written as follows:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{+\infty} x(a)w(t-a)$$

In machine learning programs, the input typically consists of a multidimensional dataset, and the central component is a multidimensional array of parameters that are iteratively adjusted by a learning algorithm during the training process. These arrays are called tensors. Because each input element and kernel must be explicitly stored separately, it is generally assumed that these functions are zero everywhere except for a finite set of points, for which we store values. In practice, this means that the sum of a finite number of elements of the array can replace the sum of minus to plus infinity.

Convolutions can be applied simultaneously along multiple axes. For two-dimensional images, the convolution must also be two-dimensional. Given an image I and the kernel K we have:

$$\mathrm{G}(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

$$\mathrm{G}(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n)$$

Since the convolution operation is commutative, the formula can be written as follows:

$$G(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$

1.2. General Features of CNN

Convolution has three important ideas that help to improve the machine learning system: sparse connections, shared access to parameters, and equivariant representation [2]. We will describe these ideas below.

Figure 1-1 shows an example of a convolution for an image.

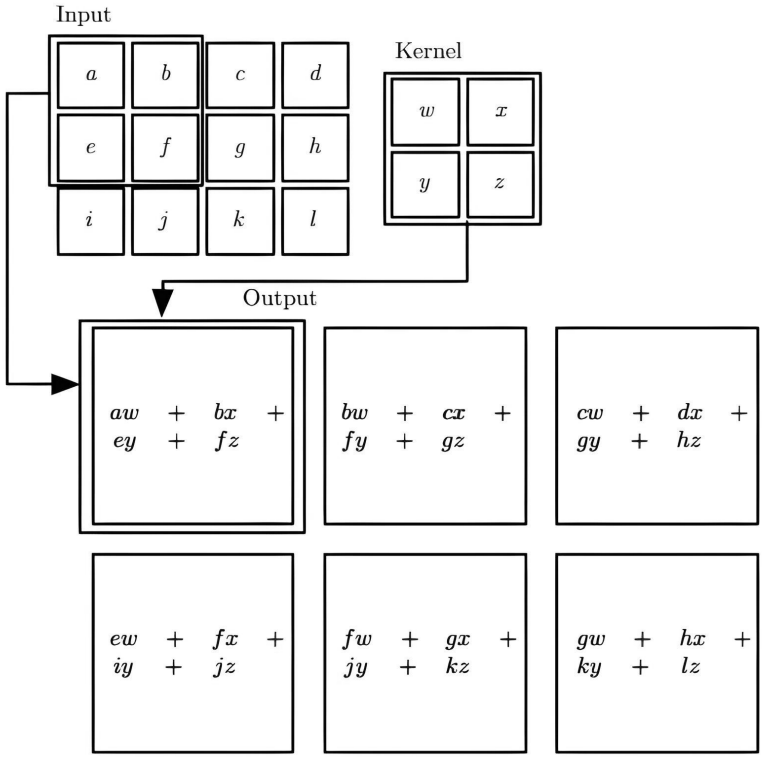

Figure 1-1. Example of convolution operation for an image

Sparse connections. Sparse connections are achieved due to the fact that the size of the kernel is smaller than the size of the input. For example, an image may contain thousands or millions of pixels, but small significant features, such as borders or angles, can be detected using the tens of pixel kernels. Therefore, fewer parameters need to be saved, which reduces the required memory capacity of the model and increases its statistical efficiency.

Parameter sharing. Sharing parameters means that the same model parameter is used in more than one function. In a traditional neural network, each parameter is used exactly once when multiplied by the corresponding input. In a convolutional neural network, each kernel element is used for each input image pixel, except for the extreme pixels. Because of that, instead of learning a separate set of parameters for each input pixel, only one set needs to be learned for the kernel, which can then be applied to each

input pixel. This dramatically reduces the amount of memory required for the model.

Equivariant representation. Convolutional neural networks have the property of equivariance with respect to parallel translation. Functions are considered equivariant if their outputs change equally to the input changes. The function f(x) is equivariant to the function g if f(g(x)) = g(f(x)). In the case of convolution, if g is the parallel translation or shift of the input, then the convolution function is equivariant with respect to g. In practice, this means that once a feature is detected in the lower right corner of the image, the network can recognize the same feature anywhere in the image. A network built on fully connected layers must re-examine the pattern so that it appears in a new position.

1.3. Architecture of Convolutional Neural Networks

Convolutional neural networks are subspecies of neural networks that use the forward propagation process. They share many similarities with other neural networks, including the presence of artificial neurons, weights, biases, and the ability to be trained through learning algorithms. The whole network works as a single evaluation function – it receives the image at the input and determines the class of the image at the output. Convolutional neural networks typically use fully connected layers with a loss function at the top level, where image classification is performed.

Conventional neural networks receive a vector at the input and transform it through a sequence of hidden layers. Each hidden layer consists of a number of neurons that are completely connected to the neurons of the previous layer. The final output layer presents the result of the network in the form of a class prediction.

Convolutional neural networks consist of the following types of layers: convolutional layers, pooling layers, normalization operation, dropout operation, and fully connected layers. Convolutional neural networks are built by superimposing the above layers on top of each other. Typical components of convolutional neural networks are presented in Figure 1-2.

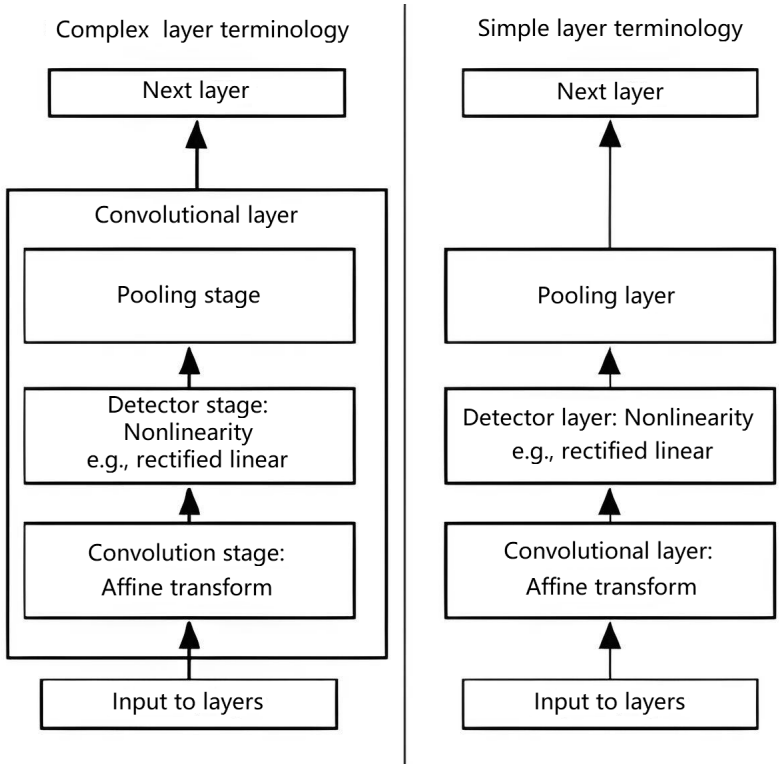

Figure 1-2. Typical components of convolutional networks

Various simple architectures of convolutional neural networks with different levels of complexity are shown in Figure 1-3.

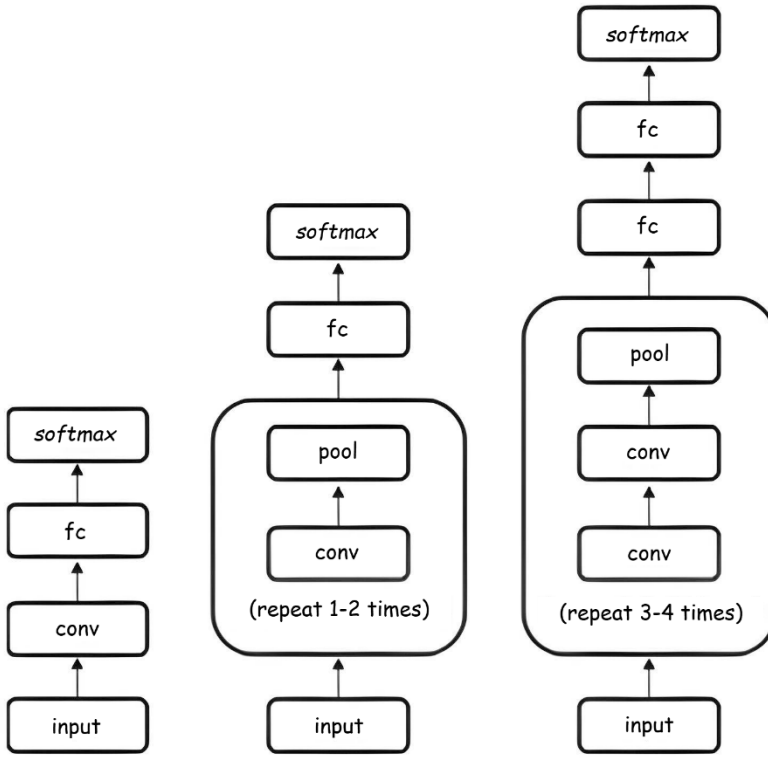

Figure 1-3. Different architectures of convolutional neural networks of different complexity

Among the layers mentioned above, only convolutional and fully connected layers can be learned during network training. Activation and dropout layers are not considered trainable. The pooling layer is as important as the convolutional and fully connected layers because it directly affects the spatial dimension of the image.

1.3.1. Convolutional Layer

In a convolutional neural network, each layer is organised as an array of feature maps rather than connecting all the neurons in one layer to the neurons of the previous layer, as is the case in a traditional neural network. Feature map arrays are parallel filter planes. In the convolutional layer, the

weighted amounts are calculated in a sliding window, and the weights are the same for all image pixels, just as in a regular image convolution [2].

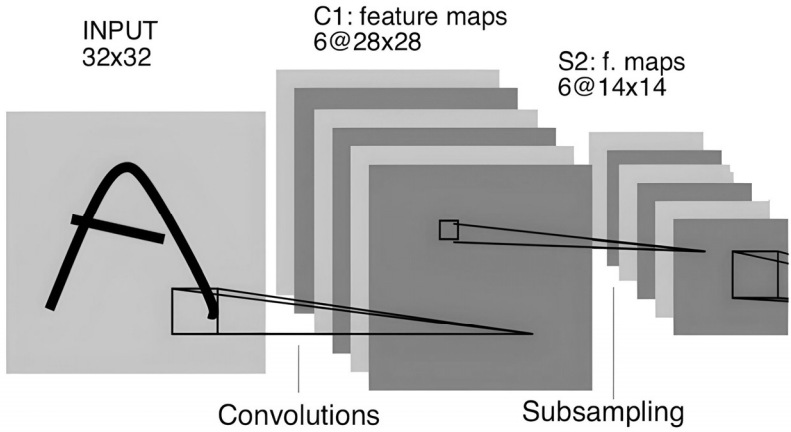

Figure 1-4. Part of the convolutional neural network LeNet-5

Unlike conventional convolution, where each filter is applied to each colour channel, a neural network convolution is usually a linear combination of the activation result of each of the C_1 input channels of the previous layer. It uses different convolutional kernels for each of the C_2 output channels. The essence of this approach is that the main task of convolutional layers of the network is to obtain local features and combine them in different sequences to obtain more accurate features.

The weighted linear combination in the convolutional layer can be written as follows:

$$s(i,j,c_2) = \sum_{c_1 \epsilon C_1} \sum_{(k,\ l) \in N} [w(k,l,c_1,c_2)x(i+k, j+l,\ c_1) + b(c_2)]\,,$$

where $x(i,j,c_1)$ is an output of the previous layer; N is a two-dimensional shift in the kernel size for convolution operations; C_1 is a set of previous layer channels, and c_1 is a specific channel within that set.

It should be noted that since the offsets (k,l) are added to the pixel coordinates (i,j), this operation is correlated in principle.

Because the weights in the convolution kernels are the same for all the pixels in each layer, these weights are distributed as if we were drawing connections between pixels in different layers. It means convolutional

neural networks need to learn much fewer parameters than networks with fully connected layers.

1.3.2. Pooling

A pooling layer is used to reduce the size of the input data. Generally, the pooling function is placed after the convolutional layer. The main purpose of this function is to reduce the spatial size of the input data gradually. That allows a reduction of the number of parameters and calculations in the network [1].

The pooling function replaces the network output at some point in the aggregate statistics of adjacent outputs. For example, the max pooling operation returns the maximum value in a rectangular neighbourhood. There is also an averaging function in a rectangular neighbourhood. Max pooling layers are typically used within the convolutional network, while averaging is closer to the last layers, where it is desirable to avoid using fully connected layers in the final stage [1].

The pooling operation makes the representation approximately invariant with respect to small parallel displacements of the input.

In the most common case, the pooling layer operates with a 2x2 filter, which is applied in step 2 and reduces the input signal by half the width and height, reducing the image by 75%.

An example of the use of max pooling is shown in Figure 1-5.

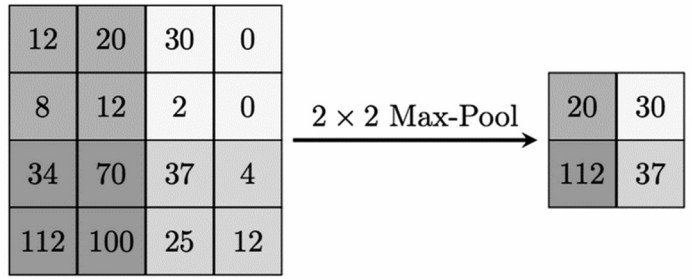

Figure 1-5. An example of the application of the max pooling layer

1.3.3. Batch Normalization

Batch normalization was first used in [3], where a batch normalization layer was used to normalize data before passing it to the next layer of the networks.

Assuming that x is a mini-batch, we can calculate the normalized $\hat{x}$ using the following equation:

$$\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}}$$

where ε is a small positive value, such as 10^{-7}, to prevent taking the square root from zero.

During training, the values μ_β and σ_β^2 are calculated at each mini-batch β, where:

$$\mu_\beta = \frac{1}{M}\sum_{i=1}^{m} x_i$$

$$\sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2$$

The application of this equation assumes that the activations after the batch normalization operation will have an average value of approximately zero and a variance of approximately one.

During testing, the values μ_β and σ_β are replaced by the moving averages μ_β and σ_β that were calculated during training.

It turned out that the normalization of batches is a very effective way to reduce the number of training epochs for the neural network. Also, the normalization of mini-batches stabilizes network training and enables a wider range of learning rates.

Typically, batch normalization occurs right before the application of nonlinearity, such as $x = Wu + b$.

1.3.4. Method Dropout

Dropout techniques are a form of regularization used to prevent overfitting by increasing testing accuracy, possibly at the cost of training accuracy. The dropout layer randomly disables, with probability p, the inputs from the previous layer to the next layer for each mini-batch in the training set [4].

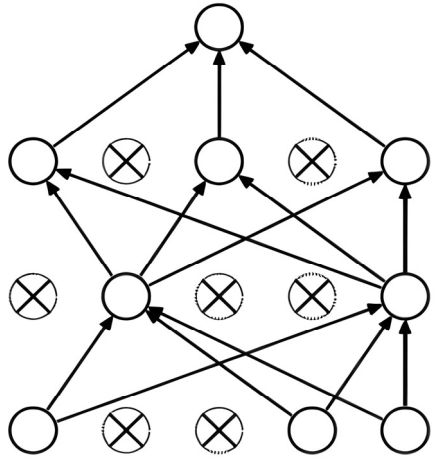

Figure 1-6. Excluding connections between two layers

The idea of using link exclusion is to reduce retraining by explicitly changing the network architecture during training. Accidental disconnection ensures that no node in the network will be responsible for activation if it is presented with a specified pattern. Instead, it is guaranteed that several neurons will be activated in the presence of such an input, which helps the model make generalizations.

1.3.5. Fully Connected Layer

There is a fundamental difference between fully connected and convolutional layers: fully connected layers study global patterns in the input feature space, while convolutional layers study local parameters and, in the case of images, find patterns in small, two-dimensional input windows.

In a fully connected layer, all neurons are fully connected to all neurons in the previous layer, as in a standard forward propagation network. Such layers are always applied at the end of the network before the image classifier. Usually, one or two fully connected layers are used before applying the classifier.

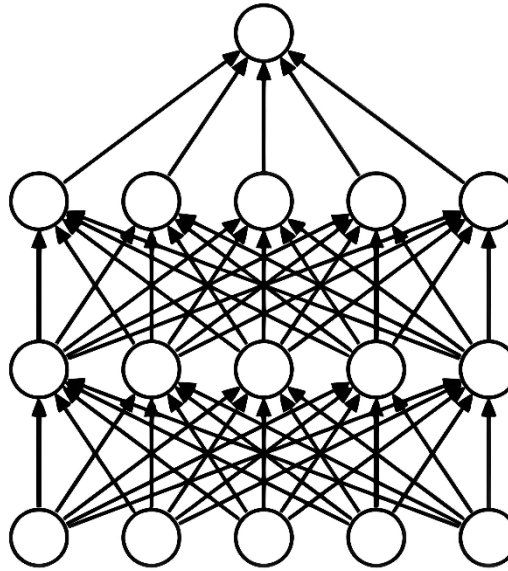

Figure 1-7. Fully connected layers

1.4. Algorithms for Optimising the Cost Function in Machine Learning for CNN

Currently, CNN learning packages posses a wide range of cost function optimisers. Most optimisers operate by calculating the gradient; however, there are specific optimisers developed to tackle the issue of local minima [7]. This section considers the most known algorithms for optimising the cost function.

1.4.1. Gradient Descent

Gradient descent is a basic method of finding the local minimum of a function [8]. The rule of updating weights θ_i is expressed by the following formula:

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \lambda \frac{\partial C^{(t)}}{\partial \theta_i}, \tag{1.1}$$

where λ is the learning rate; $\theta_i^{(t)}, \theta_i^{(t+1)}$ are the parameter values for i at iterations t and $t+1$, respectively; and $C^{(t)}$ represents cost function.

1.4.2. Adagrad

Adaptive gradient descent (Adagrad) normalises the learning rate for each dimension on which the cost function depends. During each iteration, the global learning rate is divided by the l_2 norm of the previous gradients for each dimension [5]. The following formula expresses the parameter update rule:

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \frac{\lambda}{\sqrt{\sum_{\tau=1}^{t} \theta_i^{(\tau)2} + \varepsilon}} \frac{\partial C^{(t)}}{\partial \theta_i}, \tag{1.2}$$

where λ is the learning rate; $\theta^{(t)}, \theta^{(t+1)}$ are parameter vectors at iterations t and $t+1$, respectively; $C^{(t)}$ is the cost function; and

$$\sqrt{\sum_{\tau=1}^{t} \theta_i^{(\tau)2} + \varepsilon}$$

represents l_2 norm of the parameters of all preceding iterations for feature i.

In the Adagrad algorithm, the parameters are updated at different learning rates. The more frequently a feature is encountered, the lower its learning rate becomes; conversely, the more infrequently a feature is encountered, the higher its learning rate becomes. This is because for the sparse features, the value of l_2 norm will be smaller, resulting in a higher overall learning rate. This algorithm performs well in natural language and image processing, which are areas characterised by sparse data [7].

1.4.3. RMSprop

The RMSprop algorithm [6],[7] is a mini-batch implementation of the resilient backpropagation optimisation technique (Rprop), which is best suited for full-batch learning. The Rprop algorithm addresses the problem of gradient vectors not being oriented towards the minimum in elliptical contours of the value function. One of the defining characteristics of the Rprop algorithm is that it only uses the signs of gradients for each weight during updates [7].

In the first step, the Rprop algorithm sets the same update rates for all weights: