Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy , Thrissur,
Kerala 679531

Jyothi Engineering College
NAAC Accredited College with NBA Accredited Programmes*
Approved by AICTE & affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777
NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SEMINAR REPORT

Image Recognition Method Based on Deep Learning

Submitted by

SAURAV MUNDANATT
SATHEESH KUMAR
JEC17CS091

Supervised by

Mrs. NINU FRANCIS
Assist. Prof., Dept. of CSE

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING
of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

CREATING TECHNOLOGY LEADERS OF TOMORROW

DECEMBER 2020

Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy , Thrissur,
Kerala 679531

Jyothi Engineering College
NAAC Accredited College with NBA Accredited Programmes*
Approved by AICTE & affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777
NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SEMINAR REPORT

Image Recognition Method Based on Deep Learning

Submitted by

SAURAV MUNDANATT
SATHEESH KUMAR
JEC17CS091

Supervised by

Mrs. NINU FRANCIS
Assist. Prof., Dept. of CSE

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING
of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

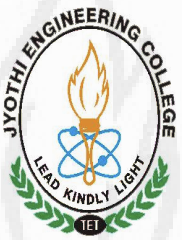
CREATING TECHNOLOGY LEADERS OF TOMORROW

DECEMBER 2020

Department of Computer Science and Engineering

JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY

THRISSUR 679 531



DECEMBER 2019

BONAFIDE CERTIFICATE

This is to certify that the seminar report entitled **Image Recognition Method Based on Deep Learning** submitted by **Saurav Mundanatt Satheesh Kumar (JEC17CS091)** in partial fulfillment of the requirements for the award of **Bachelor of Technology** degree in **Computer Science and Engineering** of **A P J Abdul Kalam Technological University** is the bonafide work carried out by him under our supervision and guidance.

Mrs. Ninu Francis	**Dr. Swapna B Sasi**	**Dr. Vinith R**
Seminar Guide	Seminar Coordinator	Head of The Dept
Assistant Professor	Associate Professor	Professor
Dept. of CSE	Dept. of CSE	Dept. of CSE

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

COLLEGE VISION

Creating eminent and ethical leaders through quality professional education with emphasis on holistic excellence.

COLLEGE MISSION

- To emerge as an institution par excellence of global standards by imparting quality engineering and other professional programmes with state-of- the-art facilities.
- To equip the students with appropriate skills for a meaningful career in the global scenario.
- To inculcate ethical values among students and ignite their passion for holistic excellence through social initiatives.
- To participate in the development of society through technology incubation, entrepreneurship and industry interaction.

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

DEPARTMENT VISION

Creating eminent and ethical leaders in the domain of computational sciences through quality professional education with a focus on holistic learning and excellence.

DEPARTMENT MISSION

- To create technically competent and ethically conscious graduates in the field of Computer Science & Engineering by encouraging holistic learning and excellence.
- To prepare students for careers in Industry, Academia and the Government.
- To instill Entrepreneurial Orientation and research motivation among the students of the department.
- To emerge as a leader in education in the region by encouraging teaching, learning, industry and societal connect

PROGRAMME OUTCOMES (POs)

1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-Long Learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. The graduates shall have sound knowledge of Mathematics, Science, Engineering and Management to be able to offer practical software and hardware solutions for the problems of industry and society at large.
2. The grad uates shall be able to establish themselves as practising professionals, researchers or Entrepreneurs in computer science or allied areas and shall also be able to pursue higher education in reputed institutes.
3. The graduates shall be able to communicate effectively and work in multidisciplinary teams with team spirit demonstrating value driven and ethical leadership.

Programme Specific Outcomes (PSOs)

1. An ability to apply knowledge of data structures and algorithms appropriate to computational problems.
2. An ability to apply knowledge of operating systems, programming languages, data management, or networking principles to computational assignments.
3. An ability to apply design, development, maintenance or evaluation of software engineering principles in the construction of computer and software systems of varying complexity and quality.
4. An ability to understand concepts involved in modeling and design of computer science applications in a way that demonstrates comprehension of the fundamentals and tradeoffs involved in design choices.

Course Outcomes (COs)

C418.1 **Presentation Skills in terms of Content** : Students will be able to show competence in identifying relevant information, defining and explaining topics under discussion. They will demonstrate depth of understanding, use primary and secondary sources; they will demonstrate the working, complexity, insight, cogency, independent thought, relevance, and persuasiveness. They will be able to evaluate information and use and apply relevant theories.

C418.2 **Presentation Skills in terms of Organization** : Students will be able to show competence in working with a methodology, structuring their oral work, and synthesizing information. They will make a detailed study on the previous works related to their topic and will present the observations.

C418.3 **Presentation Skills in terms of Delivery** : Students will use appropriate registers and vocabulary, and will demonstrate command of voice modulation, voice projection, and pacing. They will be able to make use of visual, audio and audio-visual material to support their presentation, and will be able to speak cogently with or without notes.

C418.4 **Discussion Skills** : Students will be able to judge when to speak and how much to say, speak clearly and audibly in a manner appropriate to the subject, ask appropriate questions, use evidence to support claims, respond to a range of questions, take part in meaningful discussion to reach a shared understanding, speak with or without notes, show depth of understanding.

C418.5 **Listening Skills** : Students will demonstrate that they have paid close attention to what others say and can respond constructively. Through listening attentively, they will be able to build on discussion fruitfully, supporting and connecting with other discussants.

C418.6 **Argumentative Skills and Critical Thinking** : Students will develop persuasive speech, present information in a compelling, well-structured, and logical sequence, respond respectfully to opposing ideas, show depth of knowledge of complex subjects, and develop their ability to synthesize, evaluate and reflect on information.

Course Outcome							
Programme Outcomes		C418.1	C418.2	C418.3	C418.4	C418.5	C418.6
	1	3	3	3	3	3	3
	2	3	3	3	3	3	3
	3	3	3	3	3	3	3
	4	3	3	3	3	3	3
	5	3	3	3	3	3	3
	6	3	3	3	3	3	3
	7	3	3	3	3	3	3
	8	3	3	3	3	3	3
	9	3	3	3	3	3	3
	10	3	3	3	3	3	3
	11	3	3	3	3	3	3
	12	3	3	3	3	3	3

PO - CO Mapping

PEO - CO Mapping

Course Outcome							
Programme Educational Objective		C418.1	C418.2	C418.3	C418.4	C418.5	C418.6
	1	3	3	1	1	-	2
	2	3	3	3	3	1	3
	3	1	2	3	3	1	3

PSO - CO Mapping

Course Outcome							
Programme Specific Outcomes		C418.1	C418.2	C418.3	C418.4	C418.5	C418.6
	1	3	3	3	3	3	3
	2	3	3	3	3	3	3
	3	3	3	3	3	3	3
	4	3	3	3	3	3	3

Seminar Outcome

1. Studied about the concept of Deep Learning.
2. Studied about different neural networks.
3. Analyzed and compared the general architecture of CNN, RBM, Autoencoder and Sparse Coding.
4. Studied about different deep learning frameworks.
5. Analyzed the methods used in image recognition.
6. Analyzed the trends and challenges in deep learning.

Seminar Outcome - CO Mapping

	Course Outcome						
		C418.1	**C418.2**	**C418.3**	**C418.4**	**C418.5**	**C418.6**
Seminar Outcome	**1**	3	3	3	1	3	3
	2	3	3	2	1	3	3
	3	3	3	2	2	2	3
	4	3	2	2	2	3	3
	5	3	1	3	3	3	3
	6	3	3	3	2	3	3

ACKNOWLEDGEMENT

I take this opportunity to express my heartfelt gratitude to all respected personalities who had guided, inspired and helped me in the successful completion of this seminar. First and foremost, I express my thanks to **The Lord Almighty** for guiding me in this endeavour and making it a success.

I take immense pleasure in thanking the **Management** of Jyothi Engineering College and **Dr. Sunny Joseph Kalayathankal**, Principal, Jyothi Engineering College for having permitted me to carry out this seminar. My sincere thanks to **Dr. Vinith R**, Head of the Department of Computer Science and Engineering for permitting me to make use of the facilities available in the department to carry out the seminar successfully.

I express my sincere gratitude to **Mr. Shaiju Paul & Dr. Swapna B Sasi**, Seminar Coordinators for their invaluable supervision and timely suggestions. I am very happy to express my deepest gratitude to my mentor **Mrs. Ninu Francis**, Assistant Professor,Department of Computer Science and Engineering, Jyothi Engineering College for her able guidance and continuous encouragement.

Last but not least, I extend my gratefulness to all teaching and non-teaching staff who directly or indirectly involved in the successful completion of this seminar work and to all friends who have patiently extended all sorts of help for accomplishing this undertaking.

ABSTRACT

Deep learning algorithms are a subset of the machine learning algorithms, which aim at discovering multiple levels of distributed representations. Recently, numerous deep learning algorithms have been proposed to solve traditional artificial intelligence problems. The computer vision field is greatly influenced by deep learning. This is because of high-speed computing hardware and the availability of labelled image datasets. Deep learning has a strong learning ability and can make better use of datasets for feature extraction. Because of its practicability, deep learning becomes more and more popular for many researchers to do research works. The proposed paper aims to evaluate and compare the performance of different neural networks, thus concluding the effectiveness of using the most precise neural network in image classification. The neural networks which will be compared in this paper are CNN (Convolutional Neural Networks), RBM (Restricted Boltzmann Machine), Autoencoder and Sparse Coding. . It first gives an overview of various deep learning approaches and their recent developments, and then briefly describes their applications in diverse vision tasks. Finally, the paper summarizes the future trends and challenges in designing and training deep neural networks

CONTENTS

List of Figures

List of Abbreviations

CNN	: *Convolutional Neural Network*
RBM	: *Restricted Boltzmann Machine*
DBN	: *Deep Belief Network*
DBM	: *Deep Boltzmann Machines*
SPP	: *Spatial Pyramid Pooling*
NIN	: *Network – in – Network*
DEMs	: *Deep Energy Models*
SVM	: *Support Vector Machines*
BoW	: *Bags of Visual Words*
TDNN	: *Time Delay Neural Network*
CRBM	: *Continuous Restricted Boltzmann Machine*

CHAPTER 1
INTRODUCTION

1.1 Overview

Deep learning is a subfield of machine learning which attempts to learn high-level abstractions in data by utilizing hierarchical architectures. It is an emerging approach and has been widely applied in traditional artificial intelligence domains, such as semantic parsing, transfer learning, natural language processing, computer vision and many more. There are mainly three important reasons for the booming of deep learning today: the dramatically increased chip processing abilities (e.g. GPU units), the significantly lowered cost of computing hardware, and the considerable advances in the machine learning algorithms. Various deep learning algorithms has been proposed to solve traditional AI problems. By adopting hierarchical architectures, the Deep learning algorithms learn high-level abstractions. Compared to traditional machine learning methods, deep learning has a strong learning ability and can make better use of datasets for feature extraction. Because of its practicability, deep learning becomes more and more popular for many researchers to do research works. One of the major applications of deep learning is image recognition. It is the processing of the image seen by the machine (computer) in such a way that by analyzing the digital data recording, it is possible to classify the observed objects in order to make further decisions. It can be used in a huge number of fields — from cartography and geology, through medicine, archaeology, astronomy to physics, bio-identification, security, industry, and robotics.

This paper provides a detailed aspects of various deep learning algorithms implemented in image recognition and introduce some advanced neural networks of deep learning and their applications It provides an overview of various deep learning algorithms and their applications, especially those that can be applied in the computer vision domain. deep learning algorithms are divided into four categories: Convolutional Neural Networks, Restricted Boltzmann Machines, Autoencoder and Sparse Coding. Some well-known models in these categories as well as their developments are listed. This paper also intends to describe the contributions and limitations for these models and the achievements of deep learning schemes in various computer vision applications. At last, it discusses some major challenges for deep learning, together with the existing trends that might be developed in the future.

1.2 Objective

The main objective of this seminar is to introduce the various deep learning models which can be implemented in image recognition and computer vision applications. It intends to discuss the contributions, advantages, and challenges for these models.

1.3 Organization Of The Report

The report is organised as follow:

- **Chapter 1:Introduction** Gives an introduction to deep learning and their applications in image classification.
- **Chapter 2:Literature Survey** Summarizes the research on different neural networks and its identifiable features.
- **Chapter 3: A deep learning approach for image recognition** Discusses in depth about the four different deep learning methods and their frameworks.
- **Chapter 4:Implementation & Results** Contains the implementation and results of the research.Describes and compares the performances of different neural networks for image classification.
- **Chapter 5:Advantages & Applications** List out the advantages and applications of Deep Learning particularly in the field of image recognition.
- **Chapter 6:Conclusion** The overall development and its inferred results are concluded with probable best practice.
- **References** Includes the references for deep learning methods for future purpose.

CHAPTER 2
LITERATURE SURVEY

2.1 Deep Learning based Computer Vision

Computer vision[11] tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information. It may be in the forms of decisions. The feature extraction is strongly carried out by the deep learning with promising benefits, it has been broadly utilized as a part of the field of computer vision and among others, and step by step supplanted conventional machine learning algorithms.

The computer vision basically enables the vision properties on a computer. The examples of a computer may be in the form of CCTV, drones, smartphones etc. The output of sensor is in the form of a digital form and computers interpret this form. The deep learning methods solves the different problems occurred in computer vision. Deep learning entered in in computer vision research field after development of AlexNet model. Convolutional Neural Networks (CNNs), Restricted Boltzmann Machines (RBMs), Deep Belief Nets (DBNs), and Autoencoders (AEs) outperformed in computer vision applications like surveillance, remote sensing etc.

Among the most prominent factors that contributed to the huge boost of deep learning are the appearance of large, high-quality, publicly available labelled datasets, along with the empowerment of parallel GPU computing, which enabled the transition from CPU-based to GPU-based training thus allowing for significant acceleration in deep models' training. Additional factors may have played a lesser role as well, such as the alleviation of the vanishing gradient problem owing to the disengagement from saturating activation functions (such as hyperbolic tangent and the logistic function), the proposal of new regularization techniques (e.g., dropout, batch normalization, and data augmentation), and the appearance of powerful frameworks like TensorFlow, theano, and mxnet which allow for faster prototyping.

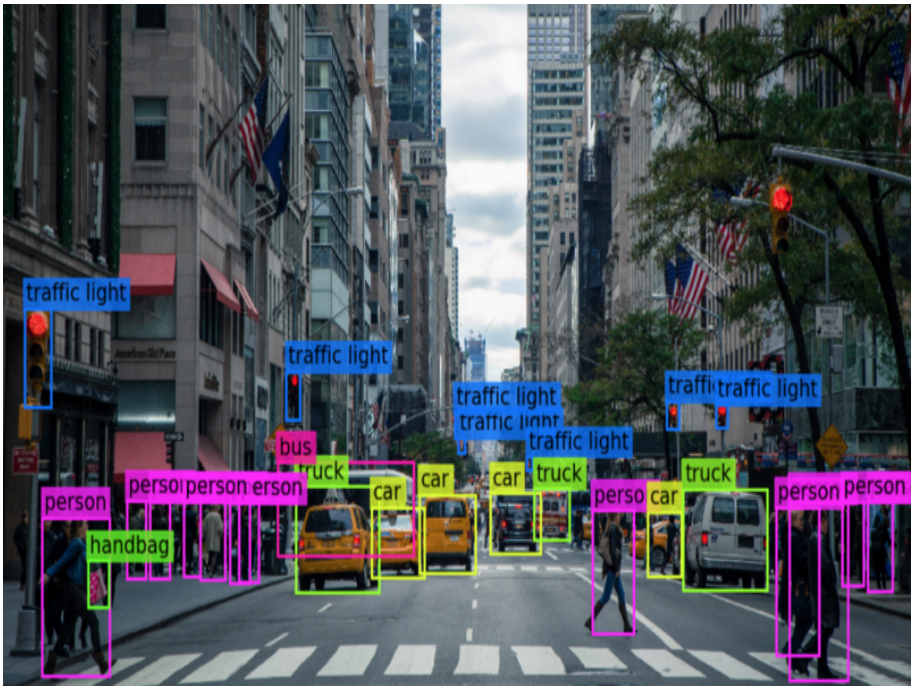


Figure 2.1: Computer Vision

2.1.1 Computer Vision Applications

- **Face recognition**

  For extracting high-level visual features, ConvNets has claimed with good results. The learned feature of CNN can directly conduct face verification. The DeepID project uses a 4-layers convolutional neural network. This CNN is without the input, output and max pooling layers. The DeepFace uses a 5-layers convolutional neural network structure. This CNN is not including the input layer and the output layer, where the last three layers do not use weight sharing. The DeepID2 consisted of the four convolutional layers. There is local weight sharing in the third and fourth convolutional layers, and the output layer. The third and fourth layers are fully connected. This technique outperformed against all deep learning algorithms and shows significant improvement in the recognition rate.

- **Object Detection**

  Detection of object is the process of detecting instances of semantic objects of a certain class in digital images and Video. A class may be humans, birds, airplanes, or plants etc. Object detection includes the creation of a large set of candidate windows. Such windows are in the sequel classified using CNN features. The paper employed selective search to derive object proposals. For each it extracts CNN features for each proposal.

For deciding whether the windows include the object or not, it feeds the features to an SVM classifier.

- **Activity Recognition**

  The human action includes the labeling of people's actions in the video. In the security applications, the monitoring system marked the criminal actions of terrorists and thieves. The camera can be used to identify the elderly or children with dangerous behaviors. Activity recognition comprises of two stages namely: feature extraction and behavioral understanding.

Figure 2.2: Block diagram of human activity recognition

2.2 Extraction and Classification of Visual Motion Patterns for Hand Gesture Recognition

This paper presents a new method for extracting and classifying motion patterns to recognize hand gestures.[14] First, motion segmentation of the image sequence is generated based on a multi-scale transform and attributed graph matching of regions across frames. This produces region correspondences and their relative transformations. Second, color information of motion regions is used to determine skin regions. Third, human head and palm regions are identified based on the shape and size of skin areas in motion. Finally, these transformations defining a region's motion between successive frames are concatenated to construct the region's motion trajectory. Gestural motion trajectories are then classified by a time-delay neural network trained with back propagation learning algorithm.

This paper is concerned with the problem of detecting two-dimensional motion across image frames and classifying motion patterns associated with certain hand actions. Classification is aimed at the recognition of the action represented by the motion pattern. Such a capability is quite central to human vision, and useful in many application domains. For concreteness, both extraction of motion patterns and their interpretations are carried out for the domain of hand gesture recognition in this work. However, the results can be easily extended to other scenarios.

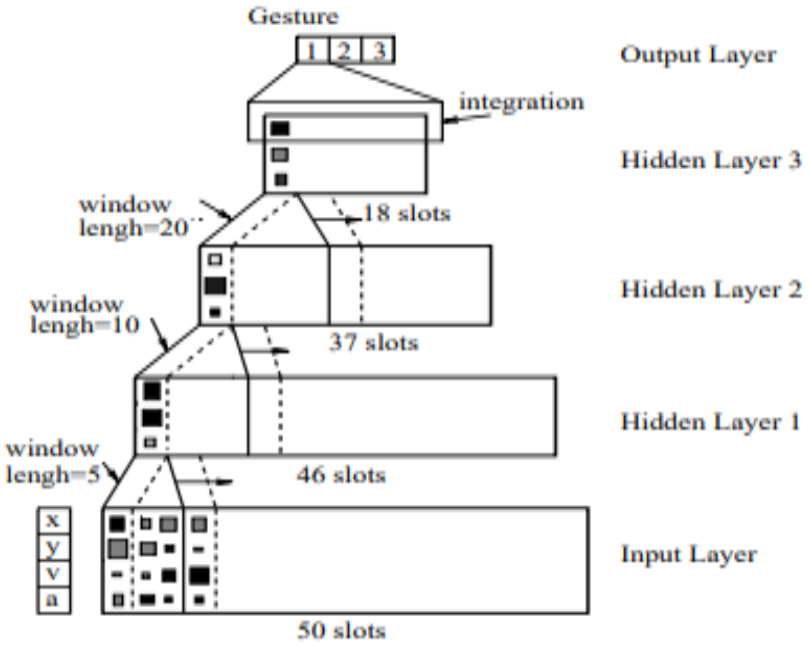


Figure 2.3: Time-Delay Neural Network

Most of the past work on gesture recognition focuses on static hand gestures, with much less attention given to the dynamic characteristics of gestures.

This paper performs motion segmentation to group pixels of similar motion into regions and find region correspondences across frames. Although there are many motion regions in each frame, the movements of palm regions contain significant information about the gesture meaning, and therefore palm motion is extracted.To distinguish among the different moving regions, it uses color and geometric characteristics. Both head and palm regions have skin color and have similar elliptic shapes, but differ in size. Motion regions with skin color are first identified, and a connected component analysis is then performed. Recognition of the motion patterns is performed using a time-delay neural network (TDNN) with an error back propagation learning algorithm. Several studies have shown that such networks are capable of classifying spatio-temporal signals. TDNNs are appropriate for recognizing motion patterns because the input data are organized as a temporal sequence, where the data sampled during a time window are input to the network simultaneously. To get a time sequence of output data, this window is moved stepwise in time.

2.3 Face Recognition Based on Convolutional Neural Network

Face recognition is of great importance to real world applications such as video surveillance, human machine interaction and security systems. As compared to traditional machine learning approaches, deep learning based methods have shown better performances in terms of accuracy and speed of processing in image recognition. This paper proposes a modified Convolutional Neural Network (CNN) architecture by adding two normalization operations to two of the layers. The normalization operation which is batch normalization provided acceleration of the network. CNN architecture was employed to extract distinctive face features and Softmax classifier was used to classify faces in the fully connected layer of CNN.[3]

In recent years, CNN has got lots of attentions from many researchers. It is an excellent model that can accomplish tasks efficiently. There are many types of CNN structures, such as LeNet, AlexNet, ZFNet, VGGNet and GoogleNe. LeCun proposed a convolutional neural network namely LeNet, and applied to handwriting recognition. AlexNet is mainly used to object detections. After that, ZFNet, VGGNet and GoogleNet were put forward based on AlexNet. At present, CNN is still an active topic with many directions to explore. Some researchers want to increase the complexity of CNN structures. Others want to combine CNN with other traditional machine learning.

Traditional methods based on shallow learning based methods only utilize from some basic features of images and depend on artificial experience to extract sample features. Deep learning based methods can extract more complicated face features. Deep learning is making crucial advances in solving problems that have restricted the best attempts of the artificial intelligence community for many years.

2.3.1 CNN

CNNs are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. CNNs are a type of feed-forward neural networks made up of many layers. CNNs consist of filters or kernels or neurons that have learn-able weights or parameters and biases. Each filter takes some inputs, performs convolution and optionally follows it with a non-linearity. The structure of CNN contains Convolutional, Pooling, Rectified Linear Unit (ReLU), and Fully Connected layers.

- **Convolutional Layer:**

  This layer performs the core building block of a Convolutional Network that does most of the computational heavy lifting. The primary purpose of Convolution layer is to

extract features from the input data which is an image. It produces a feature maps in the output image and after that the feature maps are fed as input data to the next convolutional layer.

- **Pooling Layer**

  Pooling layer reduces the dimensionality of each activation map but continues to have the most important information. The input images are divided into a set of non-overlapping rectangles. Each region is down-sampled by a non-linear operation such as average or maximum. This layer achieves better generalization, faster convergence, robust to translation and distortion and is usually placed between convolutional layers.

- **Fully Connected Layer**

  Fully Connected Layer refers to that every filter in the previous layer is connected to every filter in the next layer. The output from the convolutional, pooling, and ReLU layers are embodiment of high-level features of the input image. The goal of employing the FCL is to employ these features for classifying the input image into various classes based on the training dataset. FCL is regarded as final pooling layer feeding the features to a classifier.

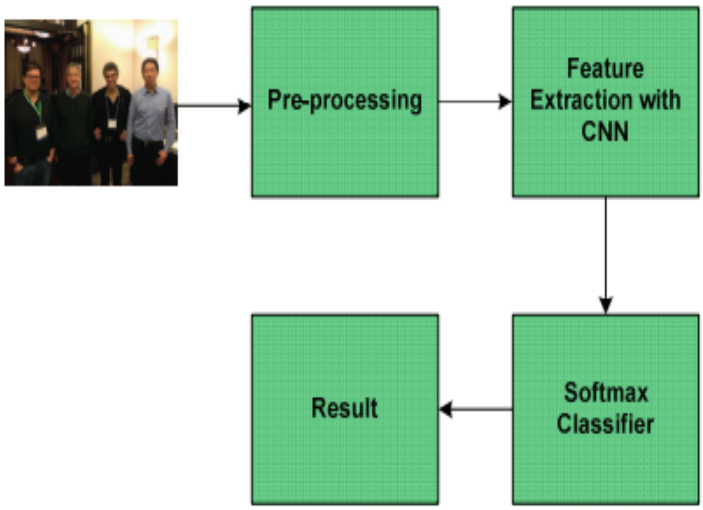


Figure 2.4: Block diagram of the proposed CNN algorithm

The prominent features of the proposed algorithm is that it employs the batch normalization for the outputs of the first and final convolutional layers and that makes the network reach higher accuracy rates. In fully connected layer step, Softmax Classifier was used to classify the faces.The results showed satisfying recognition rates. Liu proposed an SSD model which could detect objects efficiently with high accuracy. The model consists of a truncated base network structure and auxiliary structure. The truncated base network in adopts VGG-16,

and the auxiliary structure adopts some feature layers to the end of VGG-16. The network can produce a set of fixed-size bounding boxes from many feature maps. It can also give category scores if there is an object in the bounding boxes and corresponding offsets. When training SSD, the loss function which is the weighted sum of the localization loss and confidence loss will be produced on forward propagation. Lastly, the loss function can be used to fine-tune the model on back propagation

2.3.2 Experimental Results

The CNN is designed with Beta23 version of MatConvNet software tool. After pre-processing stage, size of each image was changed as 16x16x1, 16x16x3, 32x32x1, 32x32x3, 64x64x1, and 64x64x3. 66% of images were assigned as training set, 34% as test set. Different tests were implemented by making changes in image size, learning rate, batch size, and etc. CNN was trained for 35 epochs. Performance of the proposed CNN was evaluated according to top-1 and top-5 errors. Top-1 error rate checks if the top class is the same as the target label and top-5 error rate checks if the target label is one of your top five predictions. The results are better than those in the literature that use shallow learning techniques.

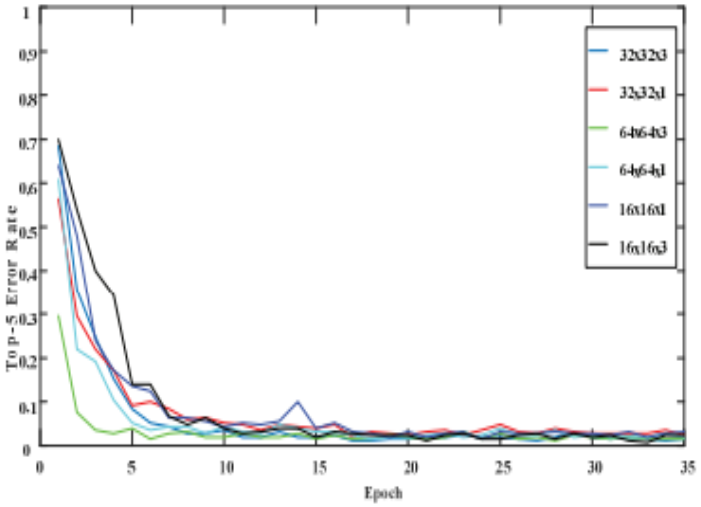


Figure 2.5: Top-5 error rate of the proposed CNN

2.4 Phone Recognition using Restricted Boltzmann Machines

For decades, Hidden Markov Models (HMMs) have been the state-of-the-art technique for acoustic modeling despite their unrealistic independence assumptions and the very limited representational capacity of their hidden states. Restricted Boltzmann Machines (RBMs)

have recently proved to be very effective for modeling motion capture sequences and this paper investigates the application of this more powerful type of generative model to acoustic modeling.[10]

In this work, it proposes using variants of Restricted Boltzmann Machines to model the spectral variability in each phone. RBM's use a distributed hidden state that allows many different features to cooperatively determine each output frame, and the observations interact with the hidden features using an undirected model. An RBM is a bipartite graph in which visible units that represent observations are connected to hidden units using undirected weighted connections. The hidden units learn non-linear features that allow the RBM to model the statistical structure in the vectors of visible states. RBMs have been used successfully for hand-written character recognition, object recognition, collaborative filtering and document retrieval. It can also be used to model high-dimensional, sequential data and they have proved to be very successful for modeling motion capture data.

This paper also introduces The Conditional RBM (CRBM), which is a variant of the standard RBM that models vectors of sequential data by considering the visible variables in previous time steps as additional, conditioning inputs. Here, two types of directed connections are added, one is autoregressive connections from the past n frames of the visible vector to the current visible vector, and connections from the past n frames of the visible vector to the hidden units. One drawback of the CRBM is that it ignores future frames when inferring the hidden states, so it does not do backward smoothing. So performing backward smoothing correctly in a CRBM would be intractable.

2.5 Performance Comparison of Three Types of Autoencoder Neural Networks

This paper presents a comparison performance on three types of autoencoders, namely, the traditional autoencoder with Restricted Boltzmann Machine (RBM), the stacked autoencoder without RBM and the stacked autoencoder with RBM. The performances are compared based on the reconstruction error for face images and using the same values for the parameters such as the number of neurons in the hidden layers, the training method, and the learning rate. Principal component analysis is a widely used method for reducing the dimensionality of data. However it can only do a linear mapping from a high-dimensional space to a low-dimensional code space.[13]

Autoencoder networks are feedforward neural networks which can have more than one

hidden layer. These networks attempt to reconstruct the input data at the output layer. The targets at the output layer are the same as the input data, thus the size of input and the output layer is almost the same. They are trained using gradient descent method, such as back-propagation. Size of the hiddern layer is smaller than the input data, and thus the dimensionality of input data is reduced to a smaller-dimensional code space at the hidden layer. The autoencoders can give mappings in both directions between the data and the code space.

Xiong proposed a modified autoencoder network to recognize and separate anomalous ones from a set of geochemical samples. Continuous Restricted Boltzmann Machine (CRBM) is used as the part of the autoencoder network. The authors adopt three steps to train the model, which are pre-training CRBMs, unrolling CRBNs to construct the network and fine-tuning parameters via back-propagation. Finally, this approach achieves good results in recognizing multivariate geochemical anomalies.

The main difference of the autoencoder and the traditional neural network is the size of the output layer. The autoencoder network comprises two components namely "encoder" and "decoder". "Encoder" network transforms the high-dimensional input data into a low-dimensional code and the "decoder" network (which is similar to the "encoder" network) reconstructs the original high-dimensional data from the low-dimensional code. The networks can be trained by minimizing the mean square error between the original and the reconstructed data. The required gradient is easily obtained by using the chain rule to back propagate the error derivatives first through the decoder network and then through the encoder network.

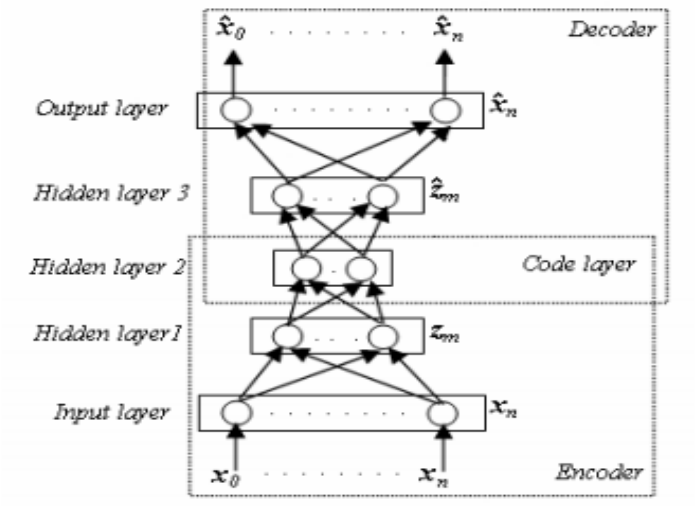


Figure 2.6: Multilayer autoencoder neural network architecture

- **Stacked Autoencoder**

it is difficult to optimize the weights in autoencoders that have multiple hidden

layers (>=2). With large initial weights, autoencoders typically find poor local minima, with small initial weights, the gradients in the early layers are tiny, making the training impossible. During the first phase, the autoencoder is assumed to have three layers, namely, input layer x, output layer y and the hidden layer h1. The size and value of input and output layers are the same. The weights can be trained using backpropagation. After the training of the separate onehidden-layer network is completed, all the weights of the autoencoder are trained together to converge to a global minimum. This kind of autoencoders is called stacked autoencoders.

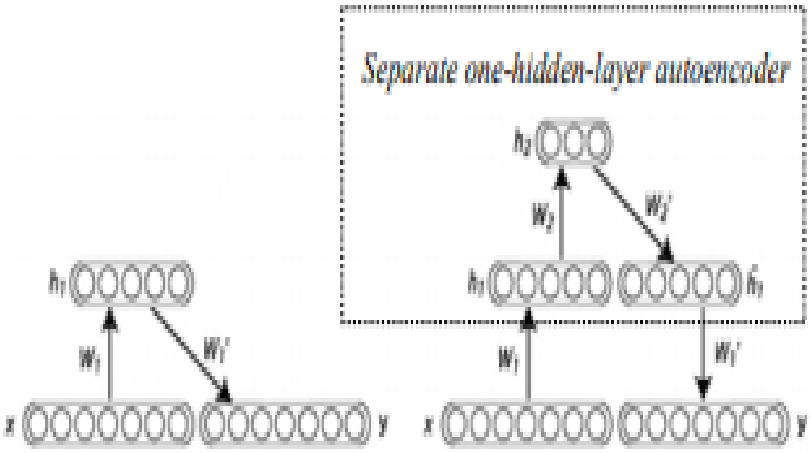


Figure 2.7: Stacked Autoencoder

- **Autoencoder with RBM**

An alternative method for training the multilayer autoencoders was proposed by Hinton. All the weights of the multilayer autoencoder can be trained in a single phase provided the weights are pretrained with RBM. Since the weights are close to good solution after RBM pre-training, backpropagation works well for fine-tuning the weights in the multi-layer autoencoders. The RBM can be modeled as a two-layer network in which the energy function of pixels connected to feature detectors. After pre-training with RBM, backpropagation can be used for fine-tuning the weights in multilayer autoencoders provided the initial weights and biases are close to optimum solution.

- **Stacked Autoencoder with RBM pre-training**

This architecture uses RBM pre-training for each hidden layer. Starting from a traditional one-hidden layer autoencoder, the weights are initialized and pretrained with RBM. Then the weights are fine-tuned with PR backpropagation. The outputs from the

hidden layer are used as the inputs for the next autoencoder in the stack. The same training process is carried out for the new separate one-hidden layer autoencoder to be stacked onto the existing trained autoencoder. The resulting autoencoder has a smaller number of hidden layer neurons, which leads to dimensionality reduction. After stacking the new autoencoder, PR backpropagation is applied to fine tune the overall weights. This process is repeated for stacking more hidden layers and for constructing a 'slimmer-waist' autoencoder.

2.5.1 Experiments and Results

The test images used are ORL face dataset. This dataset contains 400 images of size 112 X 92 for forty different people with 10 images for each person. The images are rescaled to size 37 X 30 using nearest neighbor interpolation. The intensities of the images are scaled to make the mean and the pixel variance values to be 0 and 1 respectively. The dataset is then divided into 200 training images, which contain 5 images of each person, and 200 testing images, the remaining 5 images of each person. The three architectures are compared using a layer size: (37X30)-500-300-100-30-100-300-500-1110. For training the networks, PR back-propagation method is used. The total number of training epochs employed is 230. For the two architectures which use RBM, the networks are pre-trained with RBM for 200 epochs for each hidden layer. For stacked autoencoder without RBM, the network is initialized with small random weights. In the case of stacked autoencoders, starting from a conventional one-hidden-layer network, the networks are trained for 50 epochs. In the next step, the outputs from the hidden neurons are assumed to be the inputs for the new separate one-hidden-layer autoencoder stacked on the top of the trained autoencoder. The new autoencoder is then trained by using the same method of PR backpropagation for 10 epochs. In the next phase, the whole network with three-hidden layers is again trained for 50 epochs. This process is iterated to stack more hidden layers onto the autoencoders. A total number of 230 epochs are used in the training process.

The RBM has proven to be a good pre-training method as it leads to a faster convergence compared to the stacked autoencoder without RBM. For the same number of training epochs, it is found that the autoencoder with RBM has the minimum testing MSE among all the architectures. One important phenomenon observed from these experiments is that the RBM pre-training tends to make the network more focused on the training dataset, resulting in a low generalization. The plot of MSE VS epochs reveals that the RBM pre-trains the autoencoder effectively as seen by the convergence of MSE during the training phase.

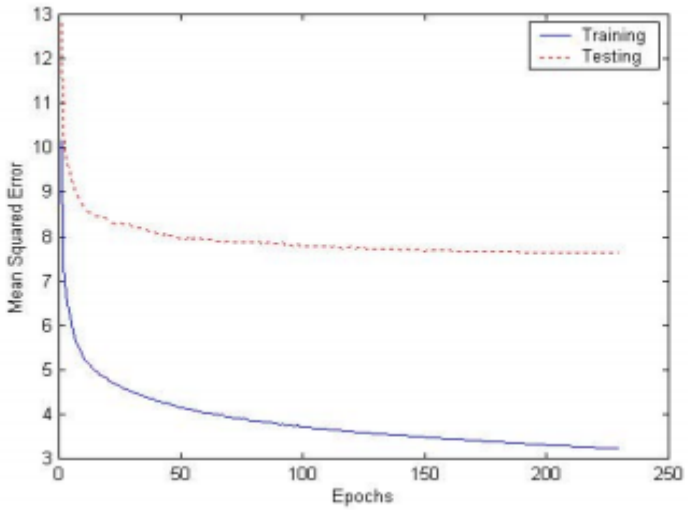


Figure 2.8: MSE of traditional autoencoder with RBM

In the test results obtained for stacked autoencoder without RBM, the MSE for both training and testing phases are extremely high at the beginning and lead to slow convergence at the end. This model has very low difference between the training and testing MSEs.

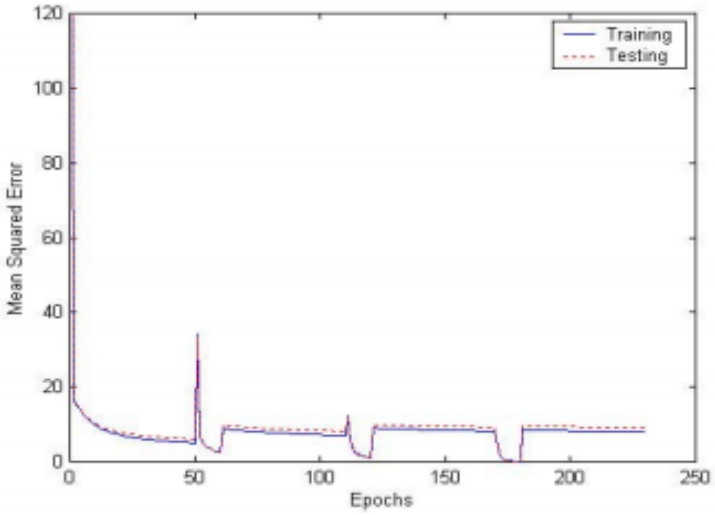


Figure 2.9: MSE of stacked autoencoder without RBM

In the performance of the stacked autoencoder with RBM pre-training, the RBM has proven to be a good pretraining method as it leads to a faster convergence compared to the stacked autoencoder without RBM. For the same number of training epochs, it is found that the autoencoder with RBM has the minimum testing MSE among all the architectures. RBM pre-training tends to make the network more focused on the training dataset, resulting in a low generalization.

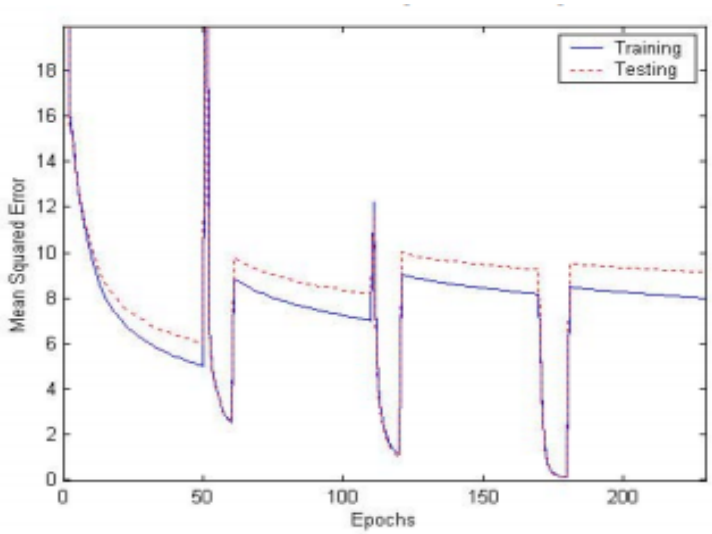


Figure 2.10: MSE of stacked autoencoder with RBM

The MSEs become worse when a new hidden layer is stacked on to the existing autoencoder. This means that the dimensionality reduction of autoencoders occurs at the expense of reconstruction efficiency as shown by the increase in the value of MSEs with each stacking. From these experiments, the stacked autoencoder with RBM outperforms the other two architectures with respect to the reconstruction error.

2.6 Convolutional Sparse Coding Classification Model for Image Classification

In this paper[2], it presents a novel classification model which combines the convolutional sparse coding framework with the classification strategy. In the training phase, the proposed model trained a convolutional filter bank by all images of each class. In the test phase, the label of test image is determined by all convolutional filter banks. Compared with canonical sparse representation and dictionary learning classification algorithm, more representative information of the corresponding images could be captured by the trained filters, thus better classification performance can be obtained. Inspired by the sparse coding mechanism of modeling receptive fields in human vision system, sparse coding approach has been widely used in many computer vision tasks. Recently sparse representation methods have been successfully used for image classification. The conventional sparse coding method assumes the ensemble of input samples are independent of one another. This independence assumption when applied to image signal leads to a problem that many basis elements are translated version of each other.

To remedy this shortcoming, convolutional sparse coding (CSC) which could learn shift invariance filters is proposed. Instead of sparsely representing a vector by the linear combination of dictionary atoms, CSC decomposes the input image into feature maps by convolution filters and gets more typical and succinct features of images. CSC has been utilized in several computer vision and pattern recognition tasks such as object recognition, image indenting, pedestrian detection, image super resolution, tissue classification, trajectory reconstruction and achieved superior performance than traditional sparse coding method. Although CSC model has achieved the state of art performances in many computer vision tasks, most of existing CSC frameworks are unsupervised that the labels of images are neglected in the stage of convolutional filter training.

In the previous works of image classification, CSC model is only used as the feature extractor, thus other classifier such as support vector machine (SVM) need to be introduced to decide the labels of test images. Because there is no class information in the trained filters, those filters are not optimal for classification. In this work, convolutional sparse coding classification (CSCC) approach is introduced which combines the traditional CSC framework with the traditional sparse representation classification methods. In the stage of filter training, the convolutional filters of each class are trained by the corresponding images, then the labels of test images could be decided by the reconstruction residual of all class filters. By introducing the label information into the trained convolutional filters, the proposed model has stronger ability to represent the corresponding image than previous SRC methods, thus better recognition results will also be obtained.

During the experiment, the proposed model and other sparse representation and dictionary learning classification algorithms on two image datasets are taken. For all methods, the best parameters are obtained by cross validation for a fair comparison. All images of two datasets are square. For proposed model, the training and test samples are the original images; while for other sparse representation and dictionary learning classification algorithms, all images need to be converted into column vectors. In order to analyze the influence of different filter dimensions (s) on the recognition rate, a experiment of different filter dimensions in a balanced set of 300 training images and 1000 test images is given. The average accuracies of this experiment are shown in Figure 1, and some visualizations of F6 which trained from number ”5” are shown in Figure 2. From Figure 1 and Figure 2, we could find that s should be neither too big nor too small. If s is too small (s = 5), filters can only capture the edge and local feature of images which are not useful for classification. If s is too big (s = 45), filters are the representation of overlapped images which will introduce the deviation in classification, and the training of bigger filters is time consuming. So the filters with appropriate dimension could extract the key, complete and global representation of the corresponding images, and the

optimal classification results could be achieved by these appropriate filters.

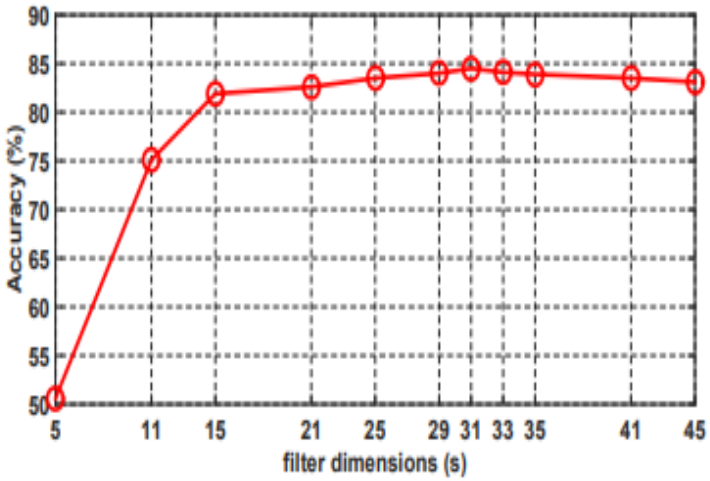


Figure 2.11: Recognition rates of different filter dimensions

The experimental result shows that the proposed CSCC method achieved the highest accuracy in different training sizes. Especially at bigger training size, the advantage of the proposed method is more apparent. Although CSCC method overcomes other classification algorithms in this experiment, the recognition rates of all methods are very low. Because the recognition of CIFAR-10 is a very challenging task, each class has some low resolution object images, the key information of object is not easy to get. To obtain higher accuracy in CIFAR-10, deeper supervised feature extraction approach should be introduced. In the future, a deep CSCC model like deep convolutional neural network should be researched.

CHAPTER 3

A DEEP LEARNING APPROACH FOR IMAGE RECOGNITION

Deep learning is a sub-field of machine learning which attempts to learn high-level abstractions in data by utilizing hierarchical architectures. It is an emerging approach and has been widely applied in traditional artificial intelligence domains, such as semantic parsing, transfer learning , natural language processing, computer vision and many more.Deep networks have been shown to be successful for computer vision tasks because they can extract appropriate features while jointly performing discrimination. [4].

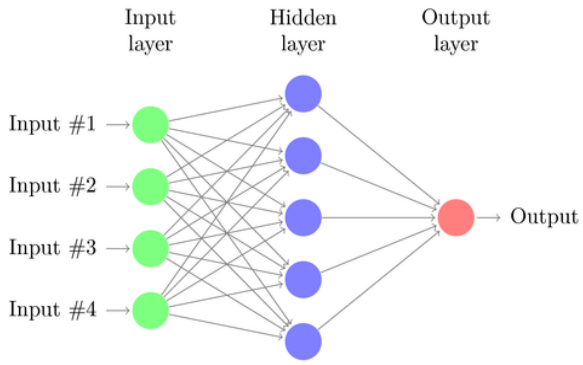


Figure 3.1: A Neural Network

Deep learning models usually adopt hierarchical structures to connect their layers. The output of a lower layer can be regarded as the input of a higher layer via simple linear or nonlinear calculations. These models can transform low-level features of the data into high-level abstract features. Owning to this characteristic, deep learning models can be stronger than shallow machine learning models in feature representation.The performance of traditional machine learning methods usually rely on user's experiences, while deep learning approaches rely on the data. Therefore, deep learning approaches have reduced the demands for users. With the progress of computer technology, computer's performance is rapidly improved. Meanwhile, information on the Internet is also spreading out. These factors provide a strong impetus for deep learning to develop and make deep learning become the prevalent method in machine learning.

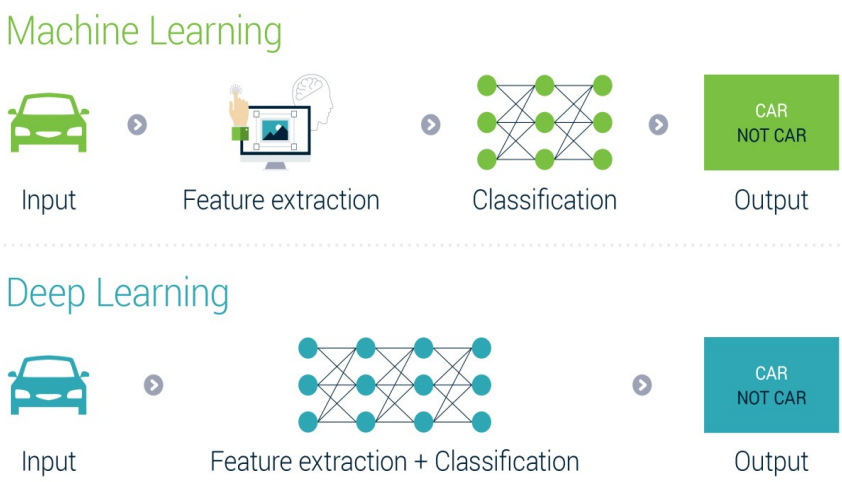


Figure 3.2: Machine Learning v/s Deep Learning

The concept of deep learning was put forward in 2006 at first. There are many outstanding figures who leads the research direction of deep learning, such as Geoffrey Hinton, Yoshua Bengio, Yann LeCun and Andrew Ng. Some companies, like Google and Facebook, have made lots of research achievements in deep learning and applied them to various fields. Google's DeepDream is an excellent software which can not only classify images but generate strange and artificial paintings based on its own knowledge, while Facebook's Deep Text is a deep learning-based text understanding engine which can classify massive amounts of data, provide corresponding services after identifying user's chatting messages and clean up spam messages.

- **BASICS OF DEEP LEARNING:**

  1. **NEURONS**:

     Just like a neuron forms the basic element of our brain, a neuron forms the basic structure of a neural network. A neuron receives an input, processes it and generates an output which is either sent to other neurons for further processing or it is the final output.

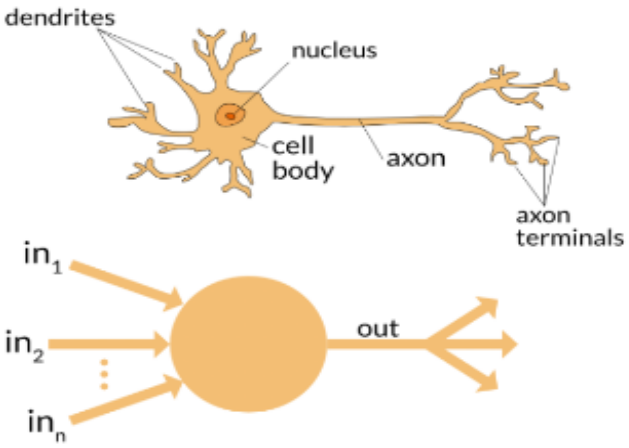


Figure 3.3: Human Neuron and Artifical Neuron

2. **WEIGHT AND BIAS**:

When input enters the neuron, it is multiplied by a weight. For example, if a neuron has two inputs, then each input will have has an associated weight assigned to it. We initialize the weights randomly and these weights are updated during the model training process. The neural network after training assigns a higher weight to the input it considers more important as compared to the ones which are considered less important.

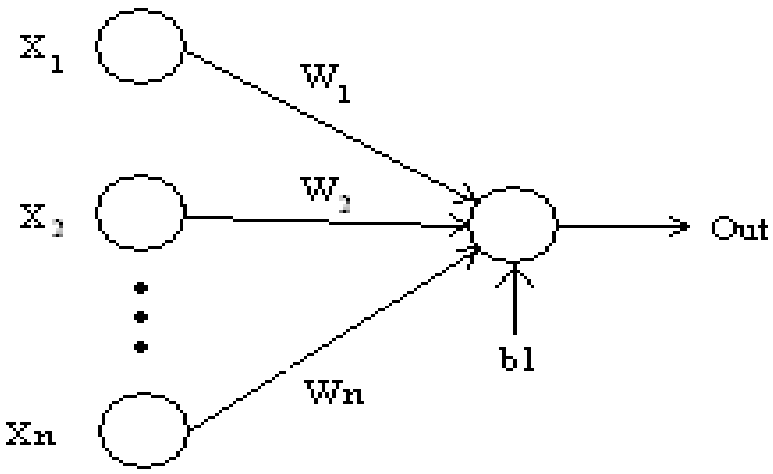


Figure 3.4: Weight and Bias

In addition to the weights, another linear component is applied to the input, called as the bias. It is added to the result of weight multiplication to the input. The bias is basically added to change the range of the weight multiplied input. After adding the bias, the result would look like a*W1+bias. This is the final linear component of the input transformation.

3. **ACTIVATION FUNCTION:**

Activation functions are functions that decide, given the inputs into the node, what should be the node's output? Because it's the activation function that decides the actual output, we often refer to the outputs of a layer as its "activations".One of the simplest activation functions is the Heaviside step function. This function returns a 0 if the linear combination is less than 0. It returns a 1 if the linear combination is positive or equal to zero.

$$f(h) = \begin{cases} 0 & \text{if } h < 0 \\ 1 & \text{if } h \geq 0 \end{cases}$$

Figure 3.5: A Sample Activation Function

4. **LEARNING RATE**

The learning rate is defined as the amount of minimization in the cost function in each iteration. In simple terms, the rate at which we descend towards the minima of the cost function is the learning rate. We should choose the learning rate very carefully since it should neither be very large that the optimal solution is missed and nor should be very low that it takes forever for the network to converge.

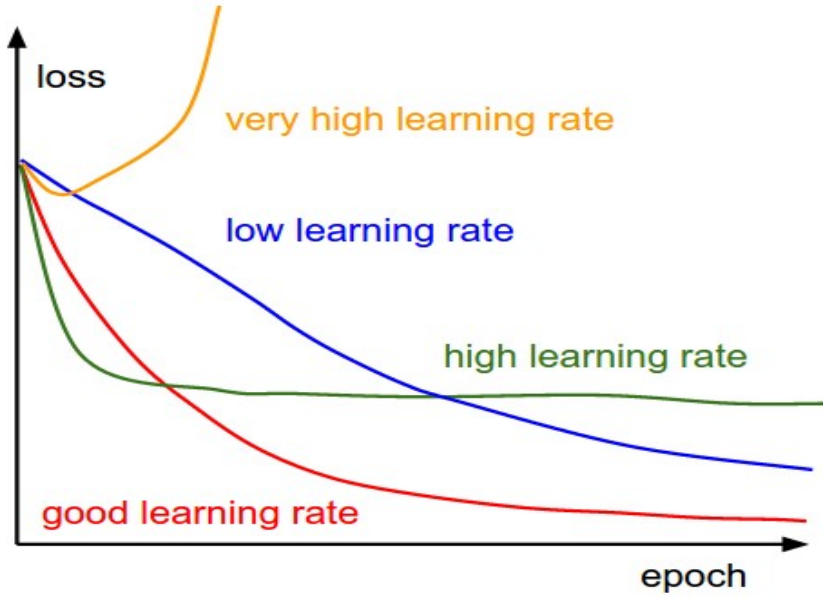


Figure 3.6: Different Learning Rates

5. **TRAINING:**

Weights start out as random values, and as the neural network learns more about what kind of input data leads to a student being accepted into a university(above example), the network adjusts the weights based on any errors in categorization that the previous weights resulted in. This is called training the neural network. Once we have the trained network, we can use it for predicting the output for the similar input.

6. **ERROR:**

In the training phase of the network, the neural networks make use of error value to adjust the weights so that it can get reduced error at each step. The goal of the training phase to minimize the error. Mean Squared Error is one of the popular error function. it is a modified version Sum Squared Error.

$$SSE = \sum_i (\text{target}^{(i)} - \text{output}^{(i)})^2$$

$$MSE = \frac{1}{n} \times SSE$$

Figure 3.7: Mean Squared Error

3.1 Major Contribution

In this paper, we emphasize on 4 Deep Learning models for image recognition and briefly review each of these deep learning methods and their most recent developments.

3.2 Recent Developments

In recent years, deep learning has been extensively studied in the field of computer vision and as a consequence, a large number of related approaches have emerged.[7] Generally, these methods can be divided into four categories according to the basic method they are derived from:

1. Convolutional Neural Networks (CNNs)
2. Restricted Boltzmann Machines (RBMs)
3. Autoencoder
4. Sparse Coding

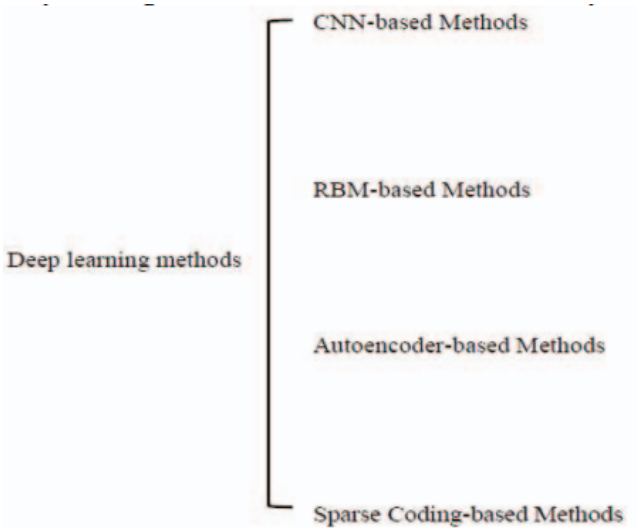


Figure 3.8: Deep Learning methods

3.3 Convolutional Neural Network

The Convolutional Neural Networks (CNN) is one of the most notable deep learning approaches where multiple layers are trained in a robust manner.[1] It has been found highly

effective and is also the most commonly used in diverse computer vision applications. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer — the output layer — that represent the predictions. Convolutional Neural Networks are a bit different. First of all, the layers are organised in 3 dimensions: width, height and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension. CNN is composed of two major parts:

1. Feature Extraction: In this part, the network will perform a series of convolutions and pooling operations during which the features are detected. If you had a picture of a zebra, this is the part where the network would recognize its stripes, two ears, and four legs.

2. Classification: Here, the fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.

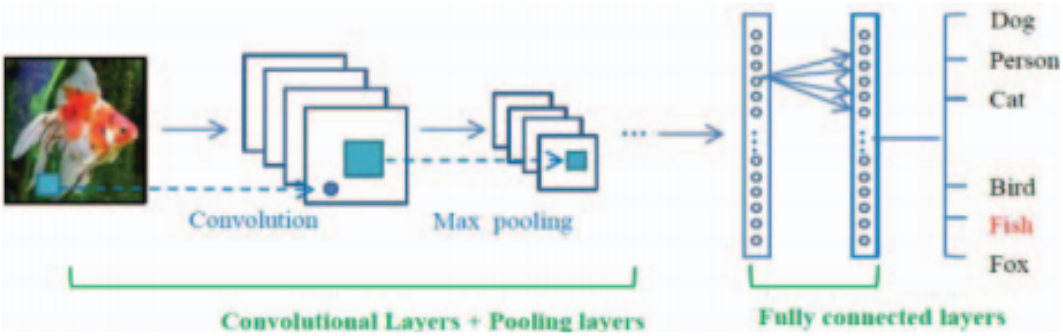


Figure 3.9: Convolutional Neural Network

Generally, a CNN consists of three main neural layers, which are convolutional layers, pooling layers, and fully connected layers. Different kinds of layers play different roles. There are two stages for training the network: a forward stage and a backward stage. First, the main goal of the forward stage is to represent the input image with the current parameters (weights and bias) in each layer. Then the prediction output is used to compute the loss cost with the ground truth labels. Second, based on the loss cost, the backward stage computes the gradients of each parameter with chain rules. All the parameters are updated based on the gradients, and are prepared for the next forward computation. After sufficient iterations of the forward and backward stages,the network learning can be stopped.

3.3.1 Convolutional Layer

In the convolutional layers, it utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps. Convolution in CNN is performed on an input image using a filter or a kernel. To understand filtering and convolution you will have to scan the screen starting from top left to right and moving down a bit after covering the width of the screen and repeating the same process until you are done scanning the whole screen. The filter slides over the input image one pixel at a time starting from the top left. The filter multiplies its own values with the overlapping values of the image while sliding over it and adds all of them up to output a single value for each overlap until the entire image is traversed.

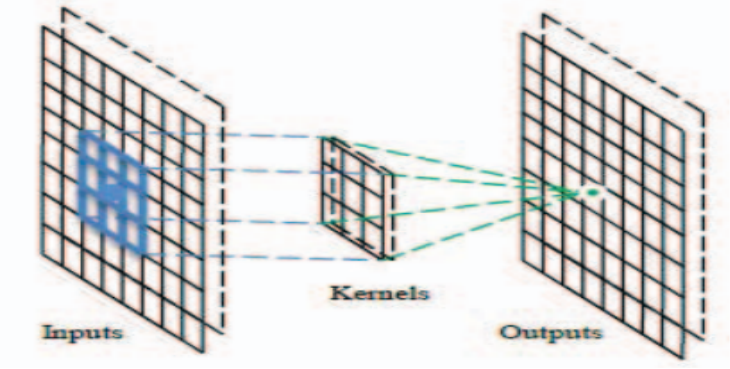


Figure 3.10: Operation of the convolutional layer

Similarly we compute the other values of the output matrix. This is the receptive field of this output value or neuron in our CNN. Each value in our output matrix is sensitive to only a particular region in our original image. Due to the benefits introduced by the convolution operation, some well-known research papers use it as a replacement for the fully connected layers to accelerate the learning process. One interesting approach of handling the convolutional layers is the Network in Network (NIN) method, where the main idea is to substitute the conventional convolutional layer with a small multi-layer perceptron consisting of multiple fully connected layers with nonlinear activation functions, thereby replacing the linear filters with nonlinear neural networks. This method achieves good results in image classification.

3.3.2 Pooling Layers

Generally, a pooling layer follows a convolutional layer, and can be used to reduce the dimensions of feature maps and network parameters. Similar to convolutional layers, pooling layers are also translation invariant, because their computations take neighboring pixels into

account. Average pooling and max pooling are the most commonly used strategies. Fig. 4 gives an example for a max pooling process. For 8x8 feature maps, the output maps reduce to 4x4 dimensions, with a max pooling operator which has size 2x2 and stride 2.

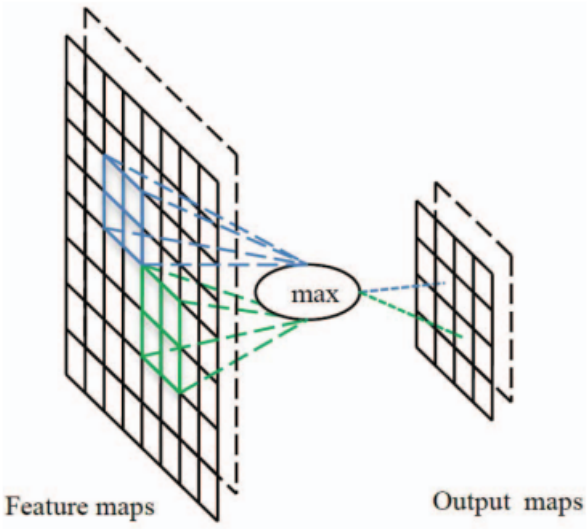


Figure 3.11: Operation of Max Pooling Layer

Compared to average pooling, max-pooling can lead to faster convergence, select superior invariant features and improve generalization.In recent years,various fast GPU implementations of CNN variants were presented, most of them utilize max-pooling strategy. A drawback of max pooling is that it is sensitive to overfit the training set, making it hard to generalize well to test samples. To overcome this, stochastic pooling approach is taken, by randomly picking the activation within each pooling region according to a polynomial distribution. It is equivalent to standard max pooling but with many copies of the input image, each having small local deformations. This probabilistic nature is helpful to prevent the overfitting problem.

Another challenge in pooling is that, for CNN-based methods, it requires a fixed-size input image. This restriction may reduce the recognition accuracy for images of an random size. To eliminate this limitation, the last pooling layer is replaced by spatial pyramid pooling layer. It can extract fixed-length representations from arbitrary images, generating a flexible solution for handling different scales, sizes, aspect ratios, and can be applied in any CNN structure to boost the performance of this structure. Handling deformation is a fundamental challenge in computer vision, especially for the object recognition task. To deal with this efficiently, deformation pooling layer is introduced to enrich the deep model by learning the deformation of visual patterns and can substitute the traditional max-pooling layer at any information abstraction level.

3.3.3 Fully Connected Layer

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space. There are several fully-connected layers converting the 2D feature maps into a 1D feature vector, for further feature representation as seen in below figure.

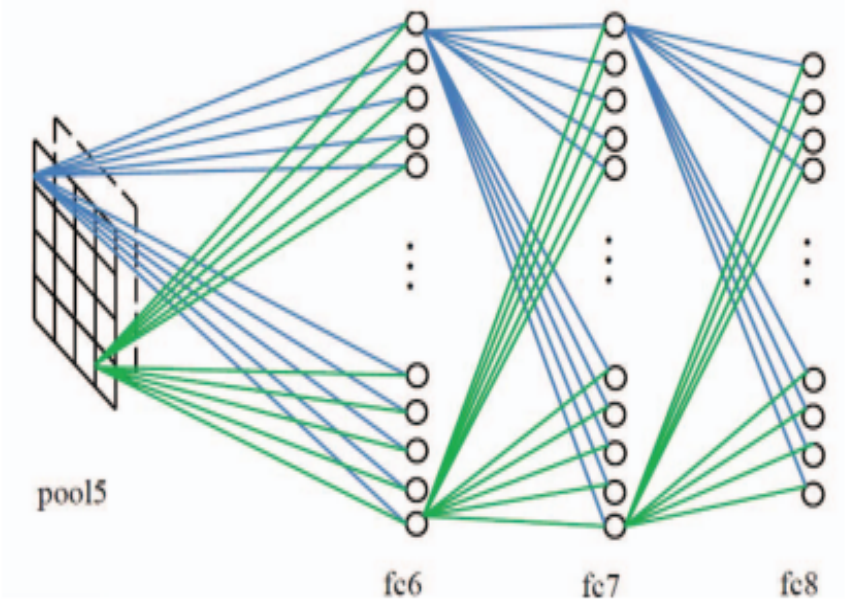


Figure 3.12: Fully Connected Model

Fully-connected layers perform like a traditional neural network and contain about 90% of the parameters in a CNN. It enables us to feed forward the neural network into a vector with a pre-defined length. We could either feed forward the vector into certain number categories for image classification or take it as a feature vector for follow-up processing. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them The drawback of these layers is that they contain many parameters, which results in a large computational effort for training them. To overcome this, GoogLeNet designed a deep and wide network while keeping the computational budget constant, by switching from fully connected to sparsely connected architectures.

3.3.4 Training Strategy

Compared to shallow learning, the advantage of deep learning is that it can build deep architectures to learn more abstract information.[9] However, the large amount of parameters

introduced may also lead to another problem: overfitting. Numerous regularization methods have emerged in defense of overfitting, including the stochastic pooling mentioned above. In this section, we will introduce several other regularization techniques that may influence the training performance.

1. Dropout and DropConnect: Dropout was proposed by Hinton and stated that during each training case, the algorithm will randomly omit half of the feature detectors in order to prevent complex co-adaptations on the training data and enhance the generalization ability. Several studies and researches have proved that dropout is an extremely effective ensemble learning method. One well-known generalization derived from Dropout is called DropConnect, which randomly drops weights rather than the activation. Although slightly slower, it can achieve competitive or even better results on a variety of standard benchmarks.

2. Data Augmentation: When a CNN is applied to visual object recognition, data augmentation is often utilized to generate additional data without introducing extra labeling costs. The well-known AlexNet employed two distinct forms of data augmentation consisting of generating image translations, horizontal reflections, and altering the intensities of the RGB channels in training images. It improved the translation and color in-variance by extending image crops with extra pixels and adding additional color manipulations. Further researchers included color casting, vignetting and lens distortion techniques, which produced more training examples with broad coverage.

3. Pre-training and fine-tuning: It means to initialize the networks with pre-trained parameters rather than randomly set parameters. It is quite popular in models based on CNNs, due to the advantages that it can accelerate the learning process as well as improve the generalization ability. Studies have shown that Pre-trained networks work better than networks trained in a traditional way.

3.4 Restricted Boltzmann Machines

Restricted Boltzmann Machine (RBM)[5] is a generative stochastic neural network, proposed by Hinton in 1986. It is a variant of the Boltzmann Machine, with the restriction that the visible units and hidden units must form a bipartite graph. This restriction allows for more efficient training algorithms, in particular the gradient-based contrasting divergence algorithm. It is the most widely-used generative model for visual recognition. There are no visible-visible or hidden-hidden connections but all visible units typically have connections to all hidden units. The weights on the connections and the biases of the individual units define a probability dis-

tribution over the state vectors, v of the visible units via an energy function. The visible units correspond to input data (e.g., image pixels) and hidden units are the extracted abstract representations. Unlike HMMs, RBMs use a distributed hidden state that allows many different features to cooperatively determine each output frame, and the observations interact with the hidden features using an undirected model. The hidden units learn non-linear features that allow the RBM to model the statistical structure in the vectors of visible states.

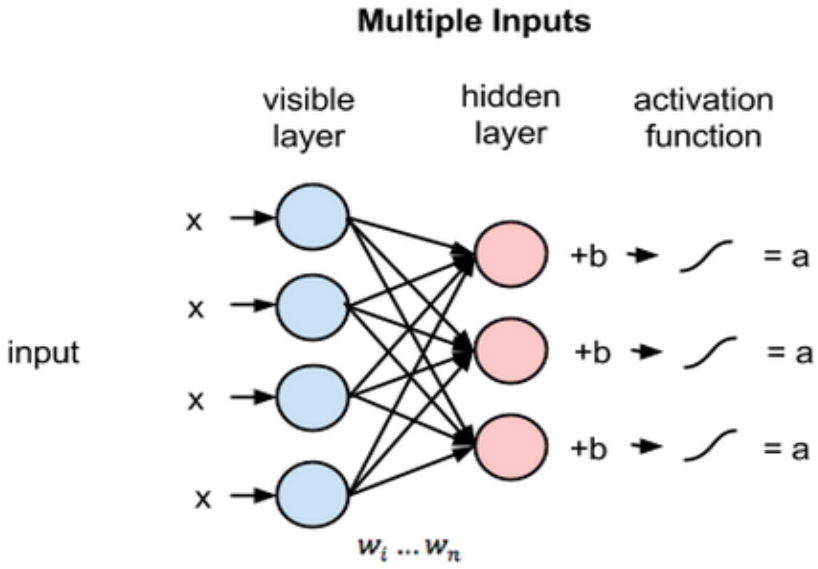


Figure 3.13: Restricted Boltzmann Machine

RBMs have been used successfully for hand-written character recognition, object recognition, collaborative filtering, and document retrieval. By conditioning on previous observations, RBMs can be used to model high dimensional, sequential data and they have proved to be very successful for modeling motion capture data. Utilizing RBMs as learning modules, we can compose the following deep models: Deep Belief Networks(DBNs), Deep Boltzmann Machines (DBMs) and Deep Energy Models (DEMs).

3.4.1 Deep Belief Networks

Deep Belief Network[6] is a kind of neural network which is stacked by several restricted Boltzmann machines (RBMs). Although RBM has inherited the two-layer neuron structure of the Boltzmann machine, there is no connection between neurons in the same layer with only the whole connection between the visual layer and the hidden layer.

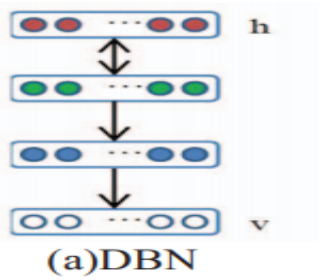


Figure 3.14: Deep Belief Network

After increasing the number of the hidden layers of RBM, we can get deep Boltzmann machine. Then, we adopt a top-down directed connection near the visual layer so that we can get DBN model. When training the network, the greedy unsupervised layer-wise pre-training method can be used to get the network weights. It only trains one layer at a time with the output of the lower layer being used as the input of the higher layer. Then, back-propagation algorithm is used to fine-tune the whole network. There are also many other approaches that aim to improve the effectiveness of DBMs. The improvements can either take place at the pre-training stage or at the training stage.

3.4.2 Deep Boltzmann Machines

The Deep Boltzmann Machine (DBM), proposed by Salakhutdinov[12], is another deep learning algorithm where the units are again arranged in layers. Compared to DBNs, whose top two layers form an undirected graphical model and whose lower layers form a directed generative model, the DBM has undirected connections across its structure. DBM's have the potential of learning internal representations that become increasingly complex, which is considered to be a promising way of solving object and speech recognition problem.

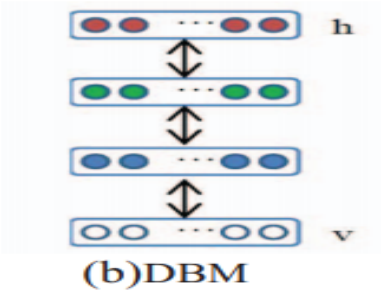


Figure 3.15: Deep Boltzmann Machine

In DBM, high-level representations can be built from a large supply of unlabeled inputs

and very limited labeled data can then be used to fine-tune the model for a specific task at hand. This allows deep Boltzmann machines to better propagate uncertainty about, and hence deal more robustly with, ambiguous inputs. DBM significantly outperforms many of the competing methods. It helps generalization because it ensures that most of the information in the model parameters comes from modeling the input data. The very limited information in the labels is used only to slightly adjust the layers of features already discovered by the deep Boltzmann machine.

3.4.3 Deep Energy Models

The Deep Energy Model (DEM), introduced by Ngiam, is a more recent approach to train deep architectures. Unlike DBNs and DBMs which share the property of having multiple random hidden layers, the DEM just has a single layer of random hidden units for efficient training and inference. These models transform the input into a new representation using a feedforward neural network, before modeling the output of this feedforward network with a single layer of stochastic hidden units.

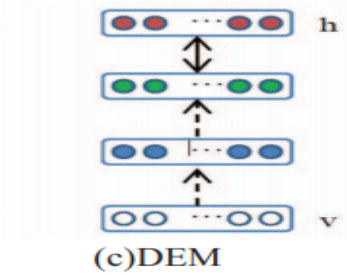


Figure 3.16: Deep Energy Models

Using deterministic hidden units instead of stochastic hidden units allows us to perform efficient learning and inference (conditioned on a visible state). This joint learning significantly improves generative performance and also changes the representations learned at each level.

3.5 Autoencoder

The autoencoder is a special type of artificial neural network used for learning efficient encodings.Instead of training the network to predict some target value Y given inputs X, an autoencoder is trained to reconstruct its own inputs X, therefore, the output vectors have the same dimensionality as the input vector[13]. Then we adopt the inverse weighting and mapping methods to make y transform to the output x' whose dimension is as the same as the input x.

Now, all we have to do is to make the error function L(x, x') be the smallest by training iterative the network weights.

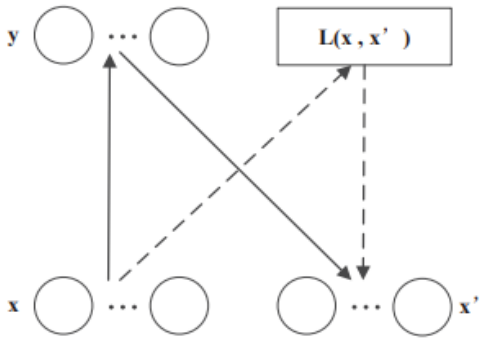


Figure 3.17: The basic principle of Autoencoder

AE also has many improved structures like Denoising Autoencoder, and Sparse Autoencoder. For Denoising Autoencoder, it uses the original data with random noise to train network weights, which makes extracted features become more robust. For Sparse Autoencoder, besides increasing the number of hidden layers and neurons, and limits the activation state of hidden nodes, which only a small number of hidden nodes are in the activated state and most of hidden nodes are in the unactivated state. The auto encoder can be optimized by reducing the reconstruction error and the corresponding code is the learned feature. We cannot get the discriminative and representative features of raw data using a single layer. Basically the deep autoencoder forwards the code learnt from the previous autoencoder to the next, to accomplish their task.

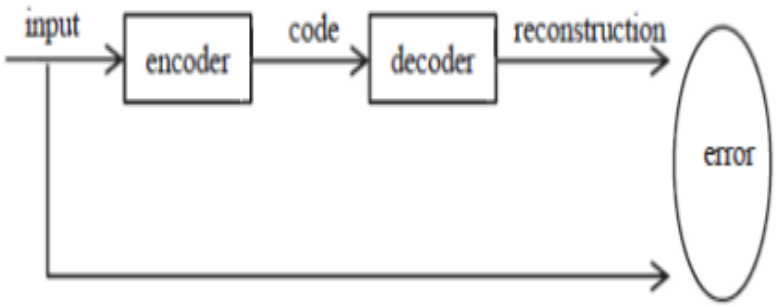


Figure 3.18: Autoencoder

Though often reasonably effective, this model could become quite ineffective if errors are present in the first few layers. This may cause the network to learn to reconstruct the average

of the training data. A proper approach to remove this problem is to pre-train the network with initial weights that approximate the final solution[23]. There are also variants of autoencoder proposed to make the representation as "constant" as possible with respect to the changes in input.

3.6 Sparse Coding

The purpose of sparse coding is to learn an over-complete set of basic functions to describe the input data.[8] Sparse coding approach has been widely used in many computer vision tasks. Recently sparse representation methods have been successfully used for image classification. The conventional sparse coding method assumes the ensemble of input samples are independent of one another. Image retrieval has been gaining wide spread importance off-late as there is an increasing need for efficient techniques and systems to retrieve information from a huge pool of data as early as possible.

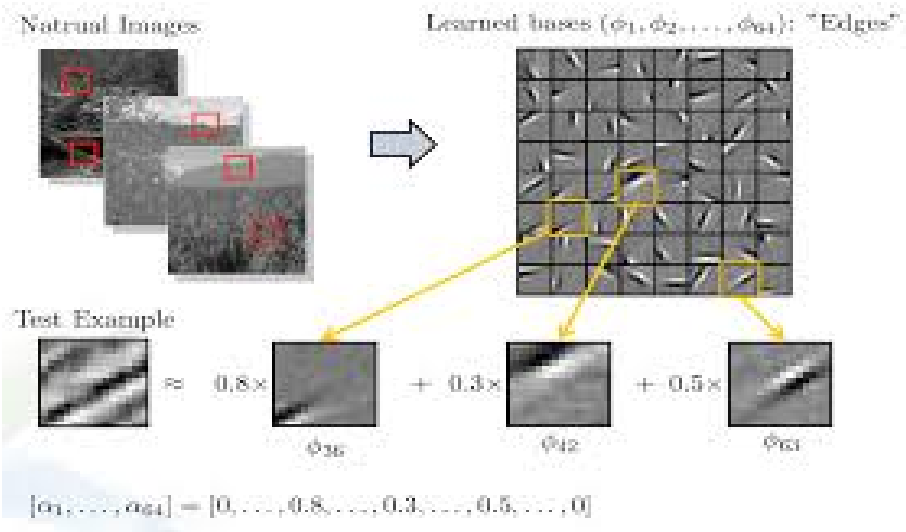


Figure 3.19: Sparse Coding

It finds a wide range of applications not limited to military, forensics for identification of criminal records, hospital management systems for patient information recovery etc. Sparse coding is to derive a compact yet discriminative image representation from multiple types of features for large-scale image retrieval. It first convert each feature descriptor into a sparse code, and aggregate each type of sparse coded features into a single vector by max-pooling. Multiple vectors from different types of features are then concatenated and compressed to obtain the final representation. This approach allows to add more types of features to improve discriminability without sacrificing scalability.

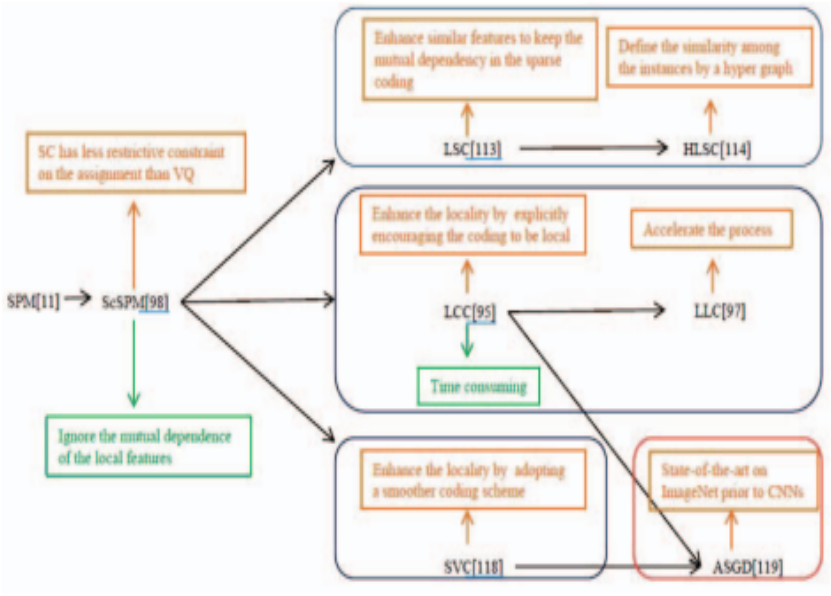


Figure 3.20: The well-known sparse coding algorithms, relations, contributions and drawbacks

3.7 Deep Learning Frameworks

A deep learning framework is an interface, library or a tool which allows us to build deep learning models more easily and quickly, without getting into the details of underlying algorithms. They provide a clear and concise way for defining models using a collection of pre-built and optimized components. Some of the key features of a good deep learning framework are:

1. Optimized for performance
2. Easy to understand and code
3. Parallelize the processes to reduce computations
4. Automatically compute gradients

The most commonly used deep learning frameworks include Caffe, TensorFlow, Torch, and Theano.

- **CAFFE**

  Caffe is a kind of deep learning framework thatis suitable for CNN models based on several computing libraries like MKL, OpenBLAS and cuBLAS. Caffe provides a set of tools to be used for training, predicting, fine-tuning and so force. It also has many

reference models and routines for learners to use. The configuration files of Caffe are simple to set up. And the Matlab and Python interfaces it provided are convenient to use. Compared to other frameworks, Caffe is easier to understand so that many beginners prefer to choose it.

- **TENSORFLOW**

TensorFlow is a large-scale machine learning framework which provides an interface for machine learning algorithms to execute. It has been used in many fields, including speech recognition, computer vision, robotics, information retrieval, and natural language processing. Tensorflow is developed from DistBelief. It takes computations described using a dataflow-like model and maps them onto different hardware platforms such as such as Android and iOS. And it supports single-device, multi-device and distributed execution.

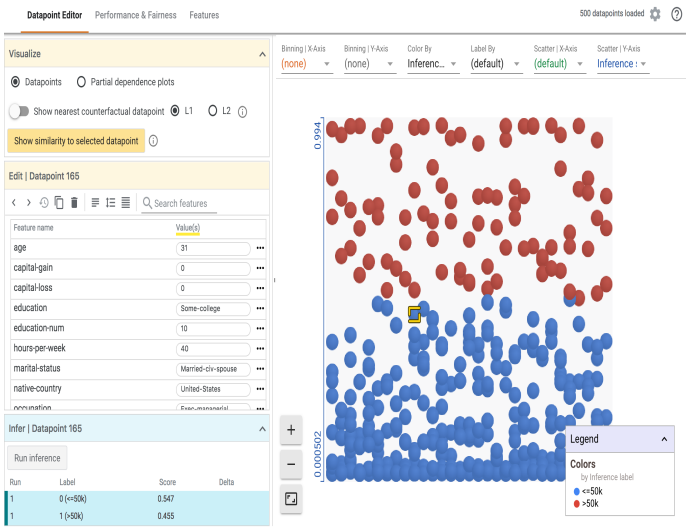


Figure 3.21: TensorFlow

- **TORCH**

Torch can support most of the machine learning algorithms. It includes most popular algorithms and models such as multi-layer perceptrons, support vector machines, Gaussian mixture models, hidden Markov models, spatial and temporal convolutional neural networks, AdaBoost, Bayes classifiers and so on. Besides supporting CPU and GPU, Torch also can be embedded into iOS, Android, and FPGA.

- **THEANO**

Theano is a framework based on Python. It can support some unsupervised and

semi-supervised learning approaches as well as supervised learning approaches, such as logistic regression, multi-layer perceptron, deep CNN, AE, RBM, and DBN. Thanks to these functions, Theano is usually be used for teaching at aboard. However, Theano has a weakness that its speed is too slow.

3.8 Results

Deep learning has been widely adopted in various directions of computer vision, such as image classification, object detection, image retrieval and semantic segmentation, and human pose estimation, which are key tasks for image understanding. In this part, we will briefly summarize the developments of deep learning.

3.8.1 Image Classification

The image classification task consists of labeling input images with a probability of the presence of a particular visual object class. the most commonly used methods in image classification were methods based on bags of visual words (BoW), which first describes the image as a histogram of approximate visual words, and then feeds the histogram into a classifier, typically a Support Vector Machine. This pipeline was based on the order-less statistics, to incorporate spatial geometry into the BoW descriptors. A spatial pyramid approach is integrated into the pipeline, which counts the number of visual words inside a set of image sub-regions instead of the whole region. This is further improved by importing sparse coding optimization problems. It recieves best performance on the ImageNet 1000-class classification in 2010. Sparse coding is one of the basic algorithms in deep learning, and it is more discriminative than the original hand-designed ones, i.e. Histogram of Oriented Gradients and Local Binary Pattern. [11]

Figure 3.22: Image classification examples from AlexNet

Despite the potential capacity possessed by larger models, they also suffered from over-

fitting and underfitting problems when there is little training data or little training time. To avoid this shortcoming, DeepImage strategy is introduced for data augmentation and usage of multi-scale images. It contains highly optimized parallel algorithm, and the classification result achieved a relative 20% improvement over the previous one with a top-5 error rate of 5.33%. The Parametric Rectified Linear Unit generated the traditional rectified activation units and derived a robust initialization method. This scheme led to 4.94% top-5 test error and surpassed human-level performance (5.1%).

3.9 Trends and Challenges

Despite the progress achieved in the theory of deep learning, there is significant room for better understanding in evolving and optimizing the CNN architectures toward improving desirable properties such as invariance and class discrimination. Larger models demonstrate more potential capacity and have become the tendency of recent developments. However, the shortage of training data may limit the size and learning ability of such models, especially when it is expensive to obtain fully labeled data. How to overcome the need for enormous amounts of training data and how to train large networks effectively remains to be addressed.

As deep learning related algorithms have moved forward the-state-of-the-art results of various computer vision tasks by a large margin, it becomes more challenging to make progress on top of that. There might be several directions for more powerful models:

1. To increase the generalization ability by increasing the size of the networks.
2. To combine the information from multiple sources.
3. To design more specific deep networks.

CHAPTER 4
ADVANTAGES AND APPLICATIONS

4.1 Advantages

1. Deep learning is an efficient method in classification as compared to other conventional techniques such as shallow learning.
2. A deep learning algorithm will scan the data to search for features that correlate and combine them to enable faster learning without being explicitly told to do so.
3. Ability to deliver high-quality results.
4. Elimination of unnecessary cost and data labeling.

4.2 Applications

1. Object detection and tracking
2. Image Reconstruction
3. Automatic image caption generation

CHAPTER 5
CONCLUSION

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Image Recognition Method Based on Deep Learning presents a comprehensive review of deep learning and develops a categorization scheme to analyze the existing deep learning literature. It divides the deep learning algorithms into four categories according to the basic model they derived from: Convolutional Neural Networks, Restricted Boltzmann Machines, Autoencoder and Sparse Coding. The state-of-the-art approaches of the four classes are discussed and analyzed in detail. For the applications in the computer vision domain, the paper mainly reports the advancements of CNN based schemes, as it is the most extensively utilized and most suitable for images. Some CNN-based algorithms have already exceeded the accuracy of human raters.Despite the promising results reported so far, there is significant room for further advances. In future, it is foreseeable that deep learning could establish perfect theories to explain its performances. It is also predicted that neural network structures will become more complex so that they can extract more semantically meaningful features, and we can use these advantages to accomplish more tasks. This paper describes these challenges and summarizes the new trends in designing and training deep neural networks, along with several directions that may be further explored in the future.

REFERENCES

[1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. IEEE, 2017.

[2] Boheng Chen, Jie Li, Biyun Ma, and Gang Wei. Convolutional sparse coding classification model for image classification. In *2016 IEEE international conference on image processing (ICIP)*, pages 1918–1922. IEEE, 2016.

[3] Musab Coşkun, Ayşegül Uçar, Özal Yildirim, and Yakup Demir. Face recognition based on convolutional neural network. In *2017 International Conference on Modern Electrical and Energy Systems (MEES)*, pages 376–379. IEEE, 2017.

[4] Xuedan Du, Yinghao Cai, Shuo Wang, and Leijie Zhang. Overview of deep learning. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pages 159–164. IEEE, 2016.

[5] Asja Fischer and Christian Igel. An introduction to restricted boltzmann machines. In *Iberoamerican congress on pattern recognition*, pages 14–36. Springer, 2012.

[6] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[7] Xin Jia. Image recognition method based on deep learning. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 4730–4735. IEEE, 2017.

[8] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801–808, 2006.

[9] Guangxin Lou and Hongzhen Shi. Face image recognition based on convolutional neural network. *China Communications*, 17(2):117–124, 2020.

[10] Abdel-rahman Mohamed and Geoffrey Hinton. Phone recognition using restricted boltzmann machines. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4354–4357. IEEE, 2010.

[11] Ashwini Patil and Amit Zore. Deep learning based computer vision: A review. 2018.

[12] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

[13] Chun Chet Tan and Chikkannan Eswaran. Performance comparison of three types of autoencoder neural networks. In *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, pages 213–218. IEEE, 2008.

[14] Ming-Hsuan Yang and Narendra Ahuja. Extraction and classification of visual motion patterns for hand gesture recognition. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 892–897. IEEE, 1998.