Convolutional Neural Network based Age Estimation from Facial Image and Depth Prediction from Single Image

Jiayan Qiu

B. Eng. (Honours)
Australian National University

January 2016

A thesis submitted for the degree of Master of Philosophy
at The Australian National University

Computer Vision Group
Research School of Engineering
College of Engineering and Computer Science
The Australian National University

Declaration

The contents of this thesis are the results of original research and have not been submitted for a higher degree to any other university or institution.

Part of the work in this thesis has been published in a conference proceedings.

Conferences

1. J. Qiu, Y. Dai, Y. Zhang and J. M. Alveraz, "Hierarchical Aggregation based Deep Aging Feature for Age Prediction," The International Conference on Digital Image Computing: Techniques and Applications 2015 (DICTA), Adelaide, Australia, Nov. 2015.

The research work presented in this thesis has been performed jointly with Dr. Yuchao Dai, Mr. Bo Li and Dr. Yuhang Zhang. The substantial majority of this work was my own.

Jiayan Qiu
College of Engineering and Computer Science,
The Australian National University,
Canberra,
ACT 0200,
Australia.

Acknowledgements

The work presented in this thesis would not have been possible without the support of a number of individuals and organizations and they are gratefully acknowledged below.

I am grateful to Dr. Yuchao Dai for supervising me. I thank him for his patience, his encouragements and time. I learned a lot from him especially in the area of computer vision. Specially, Dr. Yuchao Dai taught me a lot about how to conduct research and how to write research papers.

Thanks to Professor Henry Gardner for helping me to improve the academic writing of my first research paper.

I would like to acknowledge the support from and useful discussions with my fellow research students in the computer vision group. Special thanks to Cong Dong and Jia Li for the constant help on the technical difficulties I faced.

I am indebted to the the administrative staff at the Research School of Engineering for all their help.

Finally, I would like to cherish the memory of my aunt Jun Guo, who treated me as her own son and passed away during the period I was preparing this thesis, hope she will be treated well in heaven.

Abstract

Convolutional neural network (CNN), one of the most commonly used deep learning methods, has been applied to various computer vision and pattern recognition tasks, and has achieved state-of-the-art performance. Most recent research work on CNN focuses on the innovations of the structure. This thesis explores both the innovation of structure and final label encoding of CNN. To evaluate the performance of our proposed network structure and label encoding method, two computer vision tasks are conducted, namely age estimation from facial image and depth estimation from a single image.

For age estimation from facial image, we propose a novel hierarchical aggregation based deep network to learn aging features from facial images and apply our encoding method to transfer the discrete aging labels into a possibility label, which enables the CNN to conduct a classification task rather than regression task. In contrast to traditional aging features, where identical filter is applied to the entire facial image, our deep aging feature can capture both local and global cues in aging. Under our formulation, convolutional neural network (CNN) is employed to extract region specific features at lower layers. Then, low layer features are hierarchically aggregated by using fully connected way to consecutive higher layers. The resultant aging feature is of dimensionality 110, which achieves both good discriminative ability and efficiency. Experimental results of age prediction on the MORPH-II and the FG-NET databases show that the proposed deep aging feature outperforms state-of-the-art aging features by a margin.

Depth estimation from a single image is an essential component toward understanding the 3D geometry of a scene. Compared with depth estimation from stereo images, depth map estimation from a single image is an extremely challenging task. This thesis addresses this task by regression with deep features, combined with surface normal constrained depth refinement. The proposed framework consists of two steps. First, we implement a convolutional neural network (CNN) to learn the mapping from multi-scale image patches to depth on the super-pixel level. In this step, we apply the proposed label encoding method to transfer the contin-

uous depth labels to be possibility vectors, which reformulates the regression task to a classification task. Second, we refine predicted depth at the super-pixel level to the pixel level by exploiting surface normal constraints on depth map. Experimental results of depth estimation on the NYU2 dataset show that the proposed method achieves a promising performance and has a better performance compared with methods without the proposed label encoding.

The above tasks show the proposed label encoding method has promising performance, which is another direction of CNN structure optimization.

List of Acronyms

DL	Deep Learning
HADF	Hierarchical Aggregation based Deep Aging Feature
DAE	Denoising Auto-encoder
CNN	Convolutional Neural Network
SVM	Support Vector Machine
RF	Random Forest
2D	Two-Dimensional
3D	Three-Dimensional
NN	Neural Network
LBP	Local Binary Patterns
BIF	Biological-Inspired Features
SVR	Support Vector Regressor
MAE	Mean Absolute Error
FIP	Face Identity-Preserving Features
REL	Mean Relative Error

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Convolutional neural network (CNN) has achieved promising performance on lots of computer vision and pattern recognition tasks due to its strong ability of self-learning and dealing with large scale data. Although most recent research work focuses mainly on the innovation of CNN structure [2–6], the research on label encoding is a development direction of CNN, because a reasonable implementation of label encoding method could help to optimize the CNN structure. Therefore, this thesis mainly explores the implementations of CNN with well-performed label encoding method. In order to evaluate the performance of CNN and label encoding method, two computer vision tasks, age estimation from facial image and depth estimation from a single image, are conducted.

In this chapter, we firstly introduce CNN and label encoding method. After that, we separately give descriptions of age estimation from facial image task and depth estimation from a single image task, which are the tasks conducted for e-valuating our proposed CNN structures and label encoding method. Finally, the main contributions and outline of this thesis are introduced.

1.1 Convolutional Neural Network

Recently, CNN, one of the most commonly used deep learning methods, becomes popular because of its outstanding performance on object recognition [2], face information extraction [4], image retrieval [3] and lots of other tasks in computer vision areas. CNN is an end-to-end learning architecture to leverage collected information [7]. It trains the description features directly from raw data, which means decreasing the human effort of designing features. Meanwhile, CNN is able to deal with large scale data because of its sufficient capacity and reasonable model structure. Therefore, CNN solves two of the most important problems of traditional algorithms in computer vision, the large efforts of feature designing [8] and the weak ability to deal with big data [9]. These advantages of CNN ensure its excellent performances.

1.2 Encoding of target output labels

Traditionally, CNN changes the type of its last layer to conduct tasks with different types, for example, uses softmax layer to conduct classification tasks and uses Euclidean layer to conduct regression tasks. In this thesis, we utilize the fact that softmax layer outputs a possibility vector for classification task to explore methods that can transfer a regression task to classification task.

Usually, the labels in classification tasks are discrete and independent to each other. Therefore, when the output is a possibility vector, we choose the label with the highest value in the vector as its classification results. However, for the classification tasks transferred from regression tasks, this winner-take-all strategy is not reasonable, since the adjacent labels are interdependent. For example, a 40-year old face could also be partly similar with face with 35 years old and face with 45 years old. Therefore, a label encoding method is proposed in this thesis, which enables an encoded label to contain the information of its adjacent labels. This method ensures dependent relationship between two labels.

1.3 Age estimation from facial image

A recent study [10] has shown that human faces are fundamental to human social interaction, which means that faces are essential for daily communications. For example, faces are important in identifying emotional tendencies, health qualities and origins. Human faces contain lots of visually nonverbal information, such as age, gender, ethnicity and emotion. An instance is Age Estimation Systems (AES), which have a wide range of applications. In security area, an AES can help prevent teenagers browsing adult web pages or purchasing age restricted material from the internet. Age information can also be used in law enforcement. It can be used to quickly locate the suspects in a specific age group in the videos to be processed. This can improve the efficiency in suspect matching.

Therefore, automatic age estimation is an important task in computer vision area. And the key of this task is the high quality aging features extracted from facial images. Fig. 1.1 shows the age estimation task and an example of our aging feature.

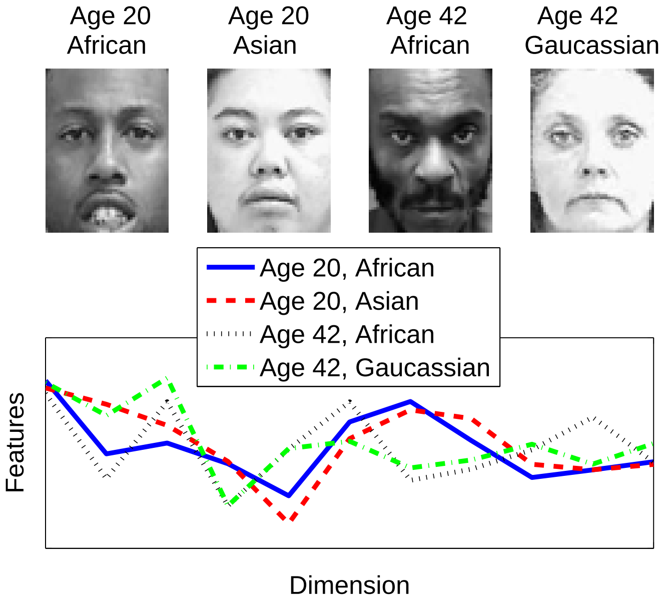


Figure 1.1: Four sample facial images and their corresponding deep aging features as extracted by our method. Despite of the remarkable intra-class diversity and inter-class similarities in facial appearance of these images, our method can extract broadly similar aging features for similar ages, regardless of race and gender. In the bottom figure, we illustrate the 110-dimension deep aging features for each facial image.

In this thesis, we introduce a new, hierarchical, aggregation-based deep network to learn aging features from facial images. To capture the aging cues in local regions, our framework trains independently for local facial regions in lower layers of the network. Facial features from lower layers are hierarchically aggregated in a fully connected way to reach higher layer representations. In this way, we extract not only local but also global cues for age prediction. The resultant aging feature vector is of significantly lower dimension – 110 dimensions compared with hand-crafted aging feature vectors with thousands of dimensions, such as BIF and LBP. It also has more discriminative power.

1.4 Depth map prediction from a single image

Recovering 3D depth from images is a basic problem in computer vision, which helps provide richer representations of objects and the surrounding environment, then enables lots of further applications in robotics [11], 3D modeling [12] and physics and support model [13]. Fig. 1.2 shows an illustration of depth estimation.

Most previous works on depth estimation focus on binocular vision [14] that require multiple images, such as structure from stereo images or motion [15] and

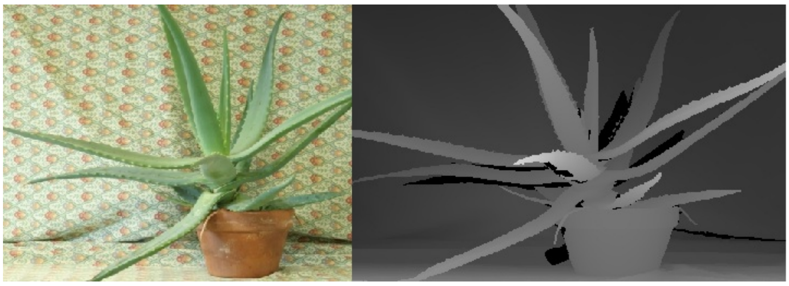


Figure 1.2: An illustration of depth estimation. Depth estimation is to estimate the distance between image pixels and camera focal point. Source from www.jayeshkawli.com.

depth from defocus [16]. Compared with the above algorithms, single image depth estimation becomes popular until recently, since this is a difficult task, which requires the usage of monocular depth cues, such as line angles, and the global structure of the image.

To explore a new efficient and well-performed method to conduct this task, this thesis presents a new framework consisting of depth regression via deep features and depth refining via surface normal constraints. Firstly, we use a deep network and formulate the problem of depth estimation as a classification problem, rather than a regression one as in [6], to exploit the relation between a color image and its corresponding depth. A multi-scale deep feature is extracted by a deep network. Secondly, to further refine the depth maps and achieve effective estimation, we introduce a surface normal constraint model to take various potentials into consideration to estimate depth for each pixel, thus upgrading the depth estimation from super-pixel level to pixel level.

1.5 Main Contributions

The main contributions of this thesis are summarized as below:

1. We propose a new CNN structure to extract aging features from facial image, which is able to extract both the local and global aging cues. Moreover, we propose a new label encoding method to transfer the discrete aging labels into a continuous possibility vector, which improves the performance of our CNN structure. Our proposed framework achieves state-of-the-art performance on age estimation task.

2. We propose a new framework to conduct depth estimation from a single image task. In this framework, we successfully transfer the regression task into a classification task by applying label encoding method, which improves the performance of our CNN structure by a large margin. Moreover, we implement the

surface normal constraints on depth refinement stage, which increases the accuracy of depth estimation. Our proposed framework achieves promising performance on depth estimation from a single image task.

3. Our proposed label encoding method shows a strong capability to improve the performance of CNN without significantly change of the structure. This could be an interesting research direction of CNN.

1.6 Outline

In this thesis, Chapter 1 gives an overall introduction on motivations, scopes and contributions. Chapter2 provides some background knowledge on convolutional neural network (CNN), and latest research progress in this area is also included. Chapter3 introduces the related research work on age estimation from facial image task and demonstrates our CNN-based architecture on this task, which achieves the-state-of-art performance. Chapter4 introduces the related research work on depth estimation from a single image task and demonstrates our CNN-based methods on this task, which achieves a promising performance. Chapter5 concludes this thesis and provides a discussion on future work related to the research work in this thesis.

Chapter 2

Background on Convolutional Neural Network

2.1 Introduction

A Convolutional Neural Network (CNN) is a type of feed-forward artificial neural network where the individual neurons are tiled in such a way that they respond to overlapping regions in the visual field. CNN is a biologically-inspired variant. From the early work of Hubel and Wiesel [17] on the cats visual cortex in 1968, it is known that the visual cortex contains a complex arrangement of neurons. These neurons are sensitive to small sub-regions of the visual field, called a receptive field. The sub-regions are tiled to cover the entire visual field. These neurons work as local filters over the input space and are well-suited to exploit spatially local correlation presented in natural images.

Fukushima's work [18] first computes models based on these local connectivities between neurons and on hierarchically organized transformations of the image. In this work, he found that when neurons with the same parameters are applied on patches of the previous layer at different locations, a form of translational invariance is achieved. According to this idea, LeCun designed and trained CNNs using the error gradient with back-propagation algorithm, obtaining state-of-the-art performance on lots of pattern recognition tasks [19, 20]. One of the main contributions of CNN on neural network area is the implementation of weights sharing, appling same weights on same feature map, which increases the learning efficiency by significantly reducing the number of trainable parameters.

Then, in recent years, active functions and dropout algorithm [1] were implemented on CNN, which increases the non-linearity and independence of feature maps, then leads to higher-understanding and more stable learnt features. In ad-

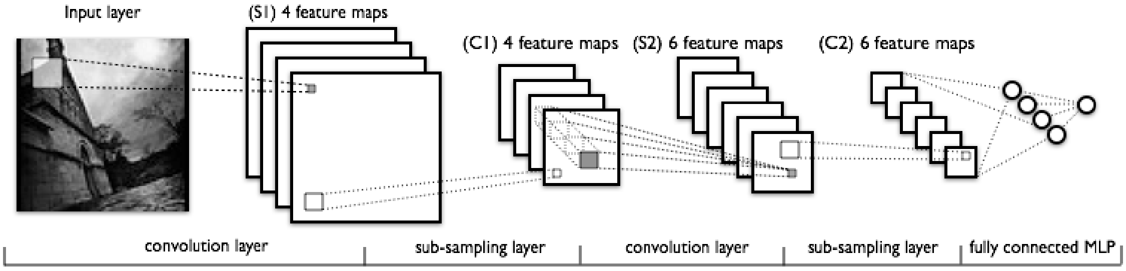


Figure 2.1: A simple CNN flow diagram. Source from www.deeplearning.net

diction to performing as an end-to-end structure, some research [21] show that the deep-learning features within the structure could also be used as features to feed into classifiers to conduct computer vision tasks, which is same with traditional hand-craft features.

Compared with traditional pattern recognition and computer vision algorithms, convolutional neural networks require less pre-processing. In other word, the network is responsible for learning filters that is hand-engineered in traditional algorithms. Compared with existing difficulty to design hand-engineered features, the independence on prior-knowledge is a major advantage of CNN.

2.2 Structure of CNN

There are two fundamental layer types in a CNN: convolutional layers and pooling layers. And with the development of CNN, active functions, dropout are added in order to increase the performance of CNN. And in the last layer of CNN, different loss layers are chose according to the type of tasks. I will briefly explain all elements we implemented in this thesis. Fig. 2.1 shows a sample of CNN overall structure.

2.2.1 Convolutional Layer

Convolutional layer is a core building block of CNN, which differs CNN with traditional artificial neural networks. To avoid the situation of learning billions of parameters (if all layers are fully connected), the idea of using convolutional operations on small regions has been introduced. One major advantage of convolutional networks is the weights sharing in convolutional layers, which means implementing same filters on same feature map. Weights sharing helps to reduce the required computing memory and to improve CNN performance on computer vision tasks [22]. Fig . 2.2 shows the weights sharing's effect on parameter reduction. Then, by reducing the number of trainable parameters, the over-fitting problem of traditional neural network was alleviated [23].

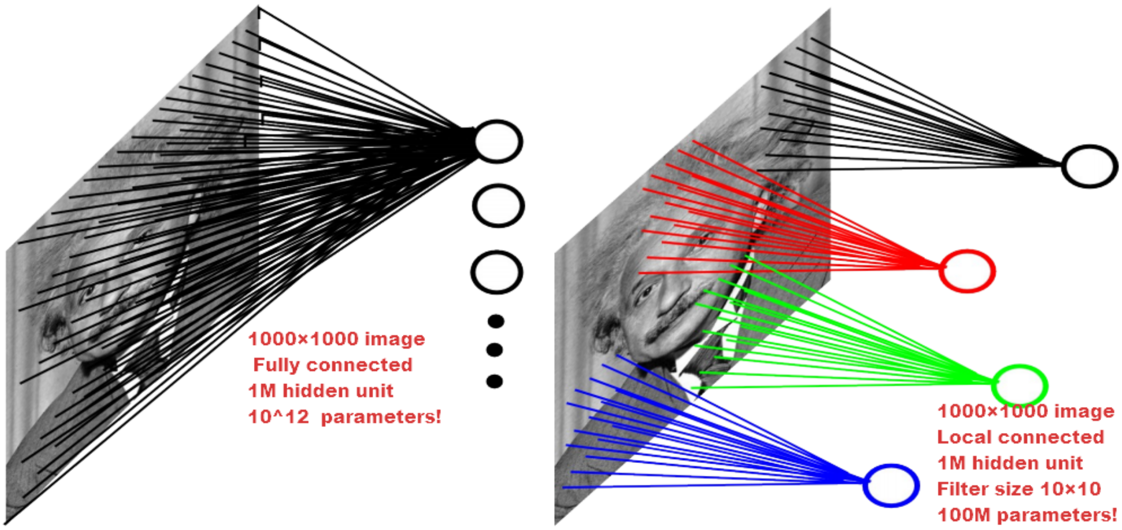


Figure 2.2: Comparison between convolutional layers with weights sharing and fully connected. It can be seen that with same hidden unit (neurons at the adjacent next layer), the number of trainable parameters with implementation of weights sharing is just 0.01% of the number of trainable parameters without weights sharing.Source from www.deeplearning.net.

The parameters of convolutional layer consist of a set of learnable filters, which is small spatially. During the forward pass, each filter is convolved across the width and height of the input volume, producing a 2-dimensional activation map of that filter. The network learns filters that will be activated by specific types of features from the input at certain positions, which is same with the convolutional operation in the traditional feature designed algorithms - extracting basic features from inputs. Then, stacking these activation maps for all filters along the depth dimension forms the full output volume. With the help of weights sharing, the number of learnt filters in convolutional increased, which enables the extraction of more information from input data.

In a convolutional layer, a feature map is obtained by repeated application of a function across sub-regions of the entire image, in other words, by convolution of the input image with a filter, adding a bias term. If we denote the k-th input feature map of a given convolutional layer as h^k, whose filter is set as W^k and bias is b^k, then the output feature map of this convolutional layer h^{k+1} is obtained as:

$$h^{k+1} = W^k \otimes h^k + b^k, \tag{2.1}$$

where the $\otimes$ denotes the convolutional operation.

For fully connected layer, it is a special case of convolutional layer. A fully connected layer is a convolutional layer that takes all neurons in the previous layer and connects them to every single neuron it has, which means a convolutional layer without weights sharing and each filter of this layer is with size 1×1. The main function of fully connected layer is to reduce the spatially located of neurons and form high-understanding features.

2.2.2 Pooling Layer

Pooling layer is commonly inserted between convolutional layers periodically in a CNN architecture. The function of pooling layer is to reduce the resolution of feature maps, thus achieving spatial invariance as well as alleviating the overfitting problem [24]. In a pooling layer, each pooled feature map corresponds to a feature map of the previous layer. A small $n \times n$ patch, as shown in Fig. 2.1, is used to combine units of the feature map, thus creating position invariance over larger local areas. Meanwhile, it down-samples the input by a factor of $n \times n$ along each direction [25]. Algorithm. 1 shows the general pooling operation in CNN.

```
Require: Parameters of spatial extent (filter size) F and stride S
  1). Accepts an input batch with size $X \times Y \times Z$
  2). Produce an output feature map with size $\hat{X} \times \hat{Y} \times \hat{Z}$, where
  $\hat{X} = \frac{X-F+S}{S}$
  $\hat{Y} = \frac{Y-F+S}{S}$
  $\hat{Z} = Z$
```

Algorithm 1: The pooling operation.

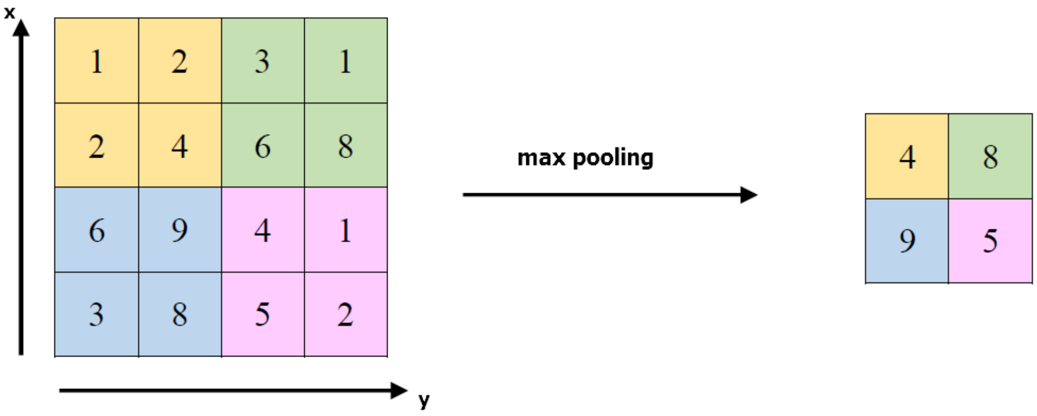
There are two main pooling layers used in CNN, the sub-sampling pooling and max pooling.

Sub-sampling Pooling

For the sub-sampling pooling, the function shows below:

$$a_j = \beta \sum_{N \times N} a_i^{n \times n} + b, \tag{2.2}$$

where a_j denotes the output of pooling layer and a_i denotes the input of pooling layer. The sub-sampling pooling operation takes the sum of the inputs, multiply it with a trainable scalar β, then adds a trainable bias b. It can be seen that, average pooling that commonly used in CNN is a special situation of sub-sampling pooling, which set $\beta = \frac{1}{N \times N}$ and $b = 0$.



1	2	3	1
2	4	6	8
6	9	4	1
3	8	5	2

4	8
9	5

Figure 2.3: Max Pooling with a filter of size 2 × 2 and a stride of 2. Illustration of how a 4 × 4 patch is down-sampled to a 2 × 2 patch by putting through max pooling layer. The maximum value is taken over four values of a 2 × 2 patch (since the filter size is 2 × 2). The filter then shifts by 2 pixels (the stride size) each time and takes the maximum value over the next 2 × 2 patch. In this way, the filter shifts and takes maximum value along the way from the top left to the bottom right over the input pitch, discarding 75% of the activations, while the dimension of width remained unchanged. The left pitch then is down sampled to the right pitch.

Max Pooling

For the max pooling, the function shows below:

$$a_j = max(a_i^{n \times n} u(n, n)), \tag{2.3}$$

which applied a window function $u(x, y)$ on the input data and extract the maximum in the neighbourhood. Figure. 2.3 shows an example of the max pooling operation.

Although average pooling was commonly used in traditional CNN, max pooling has shown a better performance in experiments and is widely used in recent CNN architecture designs. Scherer et al. [24] conducted an experiment to compare the performance of max pooling and sub-sampling on object recognition task. The result shows that max pooling is superior to sub-sampling for invariance capture in image-like data. Moreover, max pooling enables faster convergence rate by choosing superior invariant features which improves performance in generalization and reduces the number of trainable parameters, thus minimizing calculations and computing time, resulting in a better efficiency during training [26].

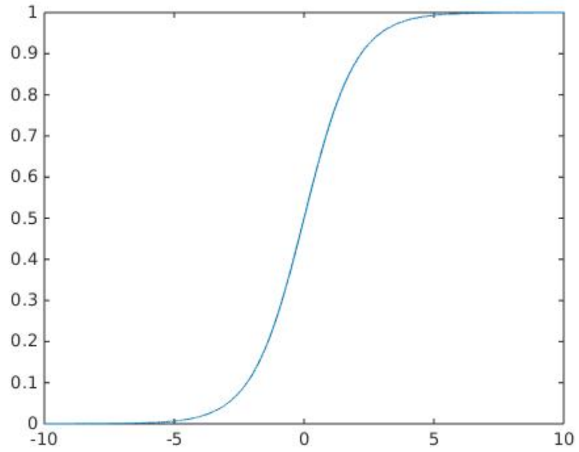


Figure 2.4: $sigm(x)$ function plotting $(-10 < x < 10)$.

2.2.3 Active Function

In CNN, one of the most significant factors is the implementation of active function. Active functions increase the non-linearity in networks, which leads to high-understanding of the input batch. In addition to non-linearity, active function also gives out a feature map without extreme data values, which increases the independency of neurons in next layer, then results increase the stability of the whole network.

Three most commonly used active functions in CNN are introduced here. One important common acknowledgement is that all active functions should be differentiable, which ensures the usage of back-propagation algorithm in the training process.

Sigmoid Function

The sigmoid function is defined as:

$$sigm(x) = \frac{1}{1+e^{-x}}, \tag{2.4}$$

and the figure of sigmoid functions shows in Fig. 2.4.

As can be seen from Fig. 2.4, sigmoid function takes a real-valued number and squashes it into range between 0 and 1. In particular, large negative numbers tend to be 0 and large positive numbers tend to be 1. The sigmoid function has been frequently used since it has a nice interpretation as the firing rate of a neuron: from not firing at all, i.e. 0, to fully-saturated firing at an assumed maximum frequency, i.e. 1.

Although sigmoid function has been widely used, recent research [27] shows that the non-linearity of sigmoid function performs not good in some practice situations, since it has two major disadvantages.

First, Sigmoid function is easily saturated and kills gradients in the training process. A very unsatisfactory property of the sigmoid neuron is that when the neuron's activation saturates at either tail of 0 or 1, the gradients at these regions are almost zero. During back-propagation, the gradient will be multiplied to the gradient of this gate's output for the whole objective. Therefore, if the local gradient is very small, it will effectively kills the gradient and almost has no signal flow through the neuron to its weights and then to its data, recursively. Moreover, one must pay extra attention when initializing the weights of sigmoid neurons to prevent saturation. For example, if the initial weights are too large then most neurons would be saturated and the network almost have no learning ability.

Secondly, the outputs of sigmoid function are not zero-centred. This is undesirable since neurons in the next layer will receive non-zero-centred data. This has impacts on the dynamics during gradient descent, because if the data coming into a neuron is always positive (e.g. $x > 0$), then the gradient on the weights during back-propagation will become either all positive, or all negative (depending on the gradient of the whole expression). This could introduce undesirable zig-zagging dynamics in the weights updating process. However, once these gradients are added up across a batch of data, the final updating for the weights could have variable signs, which minimizes the caused error [27]. Therefore, this is an inconvenience but it is less significant compared to the saturated activation problem.

tanh function

The *tanh* function is defined as:

$$tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}, \tag{2.5}$$

and the figure of *tanh* functions shows in Fig. 2.5.

As can be concluded from Fig. 2.5, similar with sigmoid function, *tanh* function squashes a real-valued number to a range of $[-1, 1]$ in an non-linearity way. Although its outputs is zero-centered, which avoid the zig-zagging dynamics in the weights updating process, it still has saturated activation problem. Therefore, although the *tanh* function has improvement over sigmoid function, it still performs not well in practice.

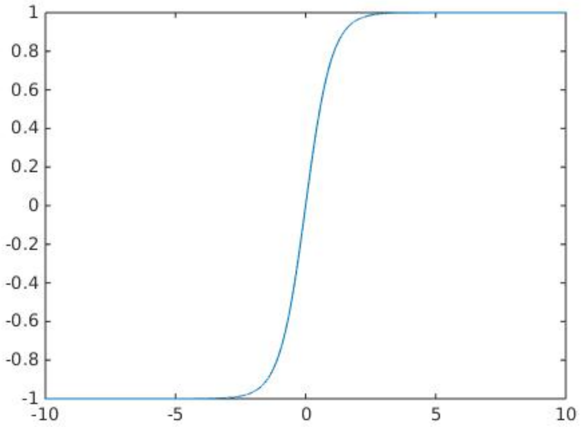


Figure 2.5: $tanh(x)$ function plotting $(-10 < x < 10)$.

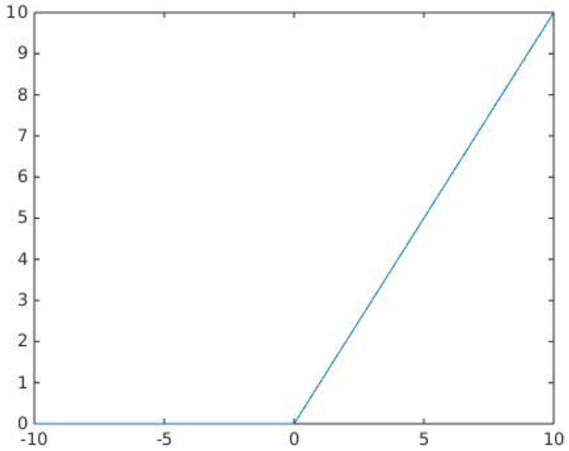


Figure 2.6: Rectified Linear Unit (ReLU) plotting $(-10 < x < 10)$.

Rectified Linear Unit (ReLU)

ReLU is the most widely used active function in CNN now. Its function is defined as:

$$f(x) = max(0, x), \tag{2.6}$$

and the figure of ReLU shows in Fig. 2.6.

There are two main advantages of ReLU function. First, compared with sigmoid function and tanh function that involve complicated operations, i.e. exponentials, ReLU can be implemented by simply set a threshold at zero. Therefore, CNN with ReLU trains several times faster than their equivalents with tanh function and sigmoid function [2]. Second, ReLU does not suffer from saturating which enhances the CNN's advantage that does not need lots of pre-processing.

However, there is also a drawback for ReLU function, i.e. ReLU units can die during training. For example, ReLU can irreversibly die and do not activate any data point during training since it will get knocked off the data manifold if the learning rate is set too high. But this is less frequently an issue with a proper setting of the learning rate.

To address this drawback, recently, a large class of ReLU functions (e.g. Leaky ReLU, Parametric ReLU, etc.), aims to fix the dying problem, called the Rectified Linear Unit Family [28]. Instead of the function being zero when $x < 0$, a leaky ReLU will instead have a small negative slope (i.e. 0.01), hence the expression could be updated as:

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha x & x < 0, \end{cases} \tag{2.7}$$

the key point of using this leaky ReLU is to find suitable α.

2.2.4 Dropout

A CNN architecture contains multiple non-linear hidden layers,which makes CNN as expressive model that is able to learn the complicated relationships between the inputs and outputs. However, under the situation where there is limited training data, many of these complicated relationships will be the result of sampling noise, which only exists in the training set but not in test data, even if the testing data is drawn from the same distribution [29]. Then, overfitting would occur during training process. Many methods have been developed to reduce such issue, such as stopping the training as soon as performance on a validation set starts to get worse, introducing various kinds of weights penalties and soft weight sharing [30].

Dropout [1] is a powerful algorithm introduced to solve the overfitting problem, which reduces the generalization error of large neural networks. It reduces complex co-adaptations of neurons, since in dropout algorithm a single neuron cannot rely on the presence of other neurons. Dropout, therefore, enhances CNN to be able to learn more robust features and stable structure [31]. The term dropout refers to dropping out units (hidden and visible) in a neural network. By dropping a unit out, we mean temporarily removing it from the network, along with its all connections [1].

As can be seen from Fig. 2.7, compared with standard neural network, the network with dropout method largely reduces the number of trainable parameters,

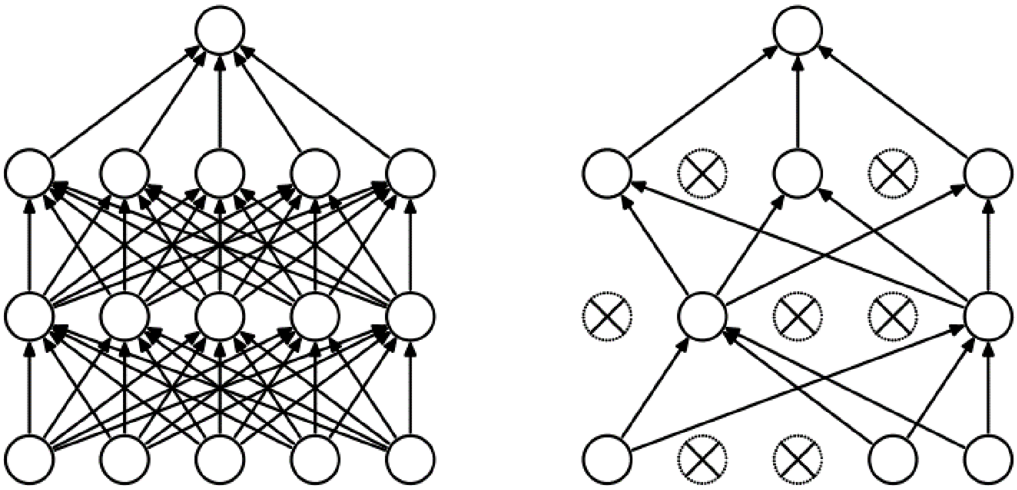


Figure 2.7: A sample of dropout. Left: A standard neural net with 2 hidden layers. Right: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped. Source from [1].

thus reduces the number of calculations and increase the computational efficiency. In practical situation, the CNN is much larger and deeper than this sample model, then the effect of dropout method is more obvious.

Dropout can be interpreted as a method to regularize a CNN by adding noise to its hidden units. The idea of adding noise to the states of units has previously been used in the context of Denoising Auto-encoders (DAEs) [32], where noise is added to the input units of an auto-encoder and the network is trained to reconstruct the noise-free input. However, different from DAEs in training process, dropout can be used in all layers except loss layer and occurs during supervised training with end-to-end back-propagation. The choice of which units to drop is random. The forward process of dropout with a convolutional layer is showed below:

$$h^{k+1} = M * (W^k \otimes h^k + b^k), \tag{2.8}$$

where M denotes the dropout mask and $*$ denotes the element-wise multiplication. Set the dropout rate to be $p, (0 < p < 1)$, then each elements of M has probability p equals to zero and probability $1 - p$ equals to one, therefore, M is a binary mask.

At testing time, it is not practical to take average of the predictions from models with all dropout situations occurred in training process. A simple approximate averaging method is implemented to solve this problem and works well. The idea is to use the trained network without dropout at testing time. The weights of this network are scaled-down versions of the trained weights. If a unit is trained with a probability p in training process, the outgoing weights of that unit are multiplied

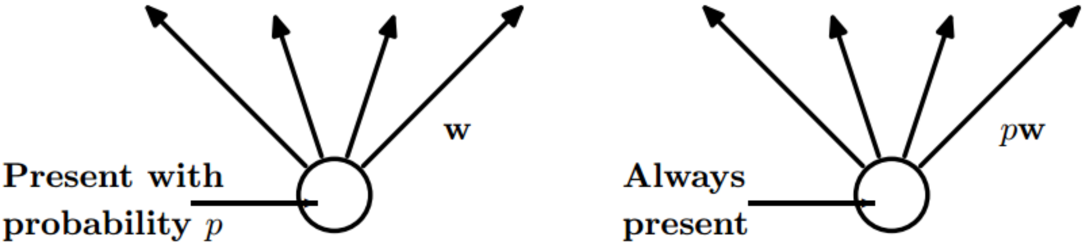


Figure 2.8: Left: a unit in training process; Right, a unit in testing process. The unit in training process is presented with probability p and is connected to units in the next layer with trained weights w. In testing process, the unit is always presented and the weights are multiplied by p. The output of testing process is same with the expected output of training process. Source from [1].

by p at test time, as shown in Fig. 2.8. By doing this scaling, it is observed that for any hidden unit the expected output (under the distribution used to drop units at training time) is the same as the actual output at test time. [33] shows that training a network with dropout and using this approximate averaging method at test time leads to significantly lower generalization error on a wide variety of classification problems compared to training with other regularization methods.

2.2.5 Loss Layer

Different loss functions are chosen for different tasks in CNN. In this subsection, we mainly introduce two commonly used loss functions - Softmax loss and Euclidean loss.

Euclidean loss

Euclidean loss is used for real-value regression tasks. Since it is for a single real-value, the last layer of CNN with Euclidean loss is 1×1 size. The mathematical function of Euclidean loss shows below:

$$L = \frac{1}{2N} \sum_{i=1}^{N} \|\hat{d}_i - d_i\|_2^2, \tag{2.9}$$

where $\hat{d}_i$ denotes the regressed outputs, d_i denotes the target outputs, N denotes the number of outputs.

Softmax loss

Softmax loss is used for predicting a single class of K mutually exclusive classes and outputs a possibility vector with size $1 \times k$, where all elements in the vector sums to be one. The mathematical function of Softmax loss shows below:

$$L = -\sum_{j} y_j \log p_j, \tag{2.10}$$

where y_j is the ground truth class, when the target belongs to j-th class, $y_j = 1$, otherwise, $y_j = 0$. p_j denotes the predicted possibility of input belongs to j-th class. When output the predicted possibility vector, the mathematical function of Softmax function shows below:

$$p_j = \frac{e^{o_j}}{\sum_k e^{o_k}}, \tag{2.11}$$

where o_j denotes the output at j-th position of the last layer of CNN. Although Softmax loss is designed for classification tasks, it can also be implemented on regression tasks, in this thesis, we applied softmax loss to two regression tasks, age estimation and single image depth prediction.

2.3 Summary

This chapter introduced the background of CNN, which is a structure with well self-learning ability. This ability increase with the non-linearity within the network, which can be achieved by increasing the number of layers in the network. Therefore, one research direction of CNN is increasing the number of network layers while keeping the size of feature maps with reasonable scale. Moreover, a reasonable shape of CNN structure is also important and a structure with pyramid shape always gives out better results compared with structures with other shapes [34]. Therefore, optimizing the shape of CNN structure via encoding output labels is also a possible research direction.

In the following two chapters, we detailed demonstrates two our proposed frameworks on age estimation from facial image task and depth estimation from a single image task, respectively. Both these two proposed frameworks based on CNN.

Chapter 3

Hierarchical Aggregation based Deep Aging Feature for Age Prediction

3.1 Introduction

Despite recent progress in the area of automatic age prediction from facial images, this area remains a very challenging one for computer vision and pattern recognition (see [35] for a recent overview). Important areas of concern include: 1) aging processes are affected by external factors as well as human genetics; 2) males and females may have very different aging characteristics; 3) people of different races have different aging cues. Images of people of the same age may have different facial appearances and images of people of different ages may have similar facial features. Such intra-class diversity and inter-class similarities pose big challenges for automatic age prediction as shown in Fig.1.1.

In past decades, there has been considerable research work on extracting robust and discriminative facial features. For age prediction, the most informative features are usually located in the regions where wrinkles typically appear, such as the eye and mouth corners, nasolabial folds, and cheeks [35]. Various appearance-based features, such as local binary patterns (LBP) [36, 37] and encoding-based sampling [38] have been proposed to capture information for facial skin wrinkles. Besides appearance-based features, gradient-based features, such as Sobel [39], Gabor [40] filters and Biological-Inspired Features (BIF) [41] have also been applied to capture facial wrinkles. Based on the anthropometric model, recent work [42, 43] has represented a sequence of individual aging face images into a sequence by learning a subspace representation. Most recently, Han et al. have proposed a hierarchical age estimator that combines both intrinsic factors and extrinsic factors to perform age estimation from facial images [35].

The use of hand-crafted features for age-estimation usually enforces one identical filter over the entire facial image [37] or specific filters on specific facial regions [44], which may be heavily affected by region mis-localization. Furthermore, in comparison with learnt features, hand-crafted features are generally not as discriminative in many computer vision tasks such as classification [3], depth estimation [45] and semantic labelling [46]. By contrast, features learnt using Convolutional Neural Networks (CNN) have been shown to significantly improve the image classification accuracy on the ImageNet database [2, 3].

Recently, lots of CNN-based frameworks achieve the state-of-art results on age estimation. Wang et al. [47] extract deep learned aging features directly from the entire face images. Moreover, Yi et al. [4] proposed an end-to-end CNN-based structure to directly estimate age from face images. However, in [47], although CNN has strong ability to learn translation invariant features, this results may be affected by the large mis-location of the human face on the image. And in [4], the classification ability of the last layer of CNN is relatively weak than linear classifiers, such as SVM and Random Forest. Then this will decrease the performance the structure.

In this chapter, we introduce a new, hierarchical, aggregation-based deep network to learn aging features from facial images. To capture the aging cues in local regions, our framework trains independently for local facial regions in lower layers of the network. Facial features from lower layers are hierarchically aggregated in a fully connected way to reach higher layer representations. In this way, we extract not only local but also global cues for age prediction. The resultant aging feature vector is of significantly lower dimension – 110 dimensions compared with many thousands of dimensions of hand-crafted aging feature vectors. It also has more discriminative power. As illustrated in Fig. 1.1, our method yields very similar aging features for people from different races with identical ages despite of the differences in the facial appearances of the images shown.

3.2 Method

Our target is to design an aging feature extractor that can extract discriminative features for age prediction. As different facial regions provide different aging cues [41], our method learns features for each facial region individually at the lowest level. Because aging cues for age prediction can also include high-level features, simply concatenating all the low-level, region-specific features together may not be sufficient to capture all the needed cues at a global level. Therefore, our method applies a hierarchical, aggregation-based network structure, where three levels (region-specific local level, row-based middle level and global level) of hierarchical nonlinear aggregation (using fully connected layers) are enforced. The structure of the whole feature extraction network is illustrated in Fig. 3.1. The specific parameters shown in this figure have been found to provide the best trade off between training speed and accuracy. Once we have learned the aging features, we use linear Support Vector Regressor (SVR) to conduct age prediction.

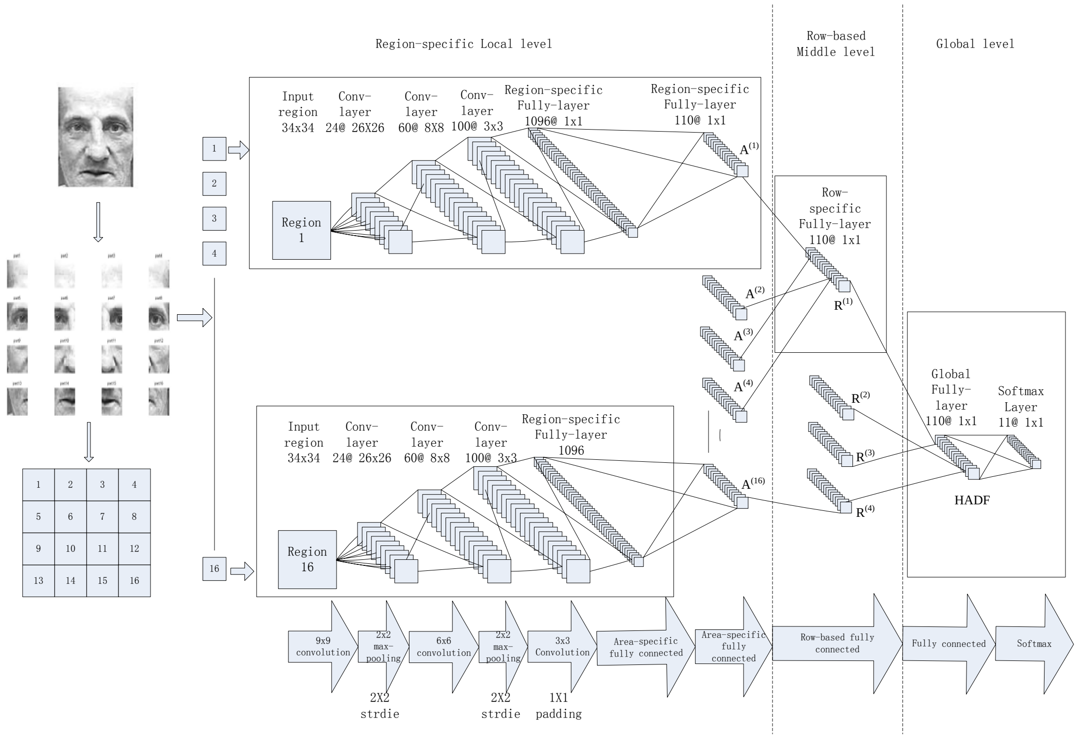


Figure 3.1: Our hierarchical aggregation based deep aging feature extraction network. In our implementation, facial images are first resized to 128 × 96 and then are divided into 4×4 regions with the sub-image size 34×34. There are 1 pixel and 5 pixels' overlap between adjacent regions in the same column and row respectively. Each region is fed into a region-specific network. Region specific features from lower level are hierarchically aggregated in higher levels by using a fully connected layer. In the last layer (softmax layer), we propose to use a new label method to transform the discrete age labels to a 11-dimension probability label.

3.2.1 Region-specific Local Level

At the lowest level, our method learns local-level features for each region of overlapping 4×4 divisions of the facial images. Traditionally, most CNNs share weights of all neurons on the same map. However, this sharing does not work well on images with fixed spatial layout, such as human faces [48]. For example, the eyes and the nose may share the same low-level features, but their contributions to understanding high-level features could be quite different. Therefore, we first divide the facial image into 4×4 divisions and then learn region-specific layers for each division. There are 1 and 5 pixels overlap on column and row between adjacent local regions, which makes sure the local region input contains the meaningful information, such as nose and eyes. The overlaps are set with experiments results on the validation set, which will explain at Section.3.3.

Our region-specific level contains 5 layers: three convolution (+ pooling) layers and two fully connected layers. The pooling layers in the network alleviate the effects of registration error and make the aging feature translation invariant. In order to reduce over-fitting in local feature extraction, a 0.5 dropout rate has been enforced on the two fully connected layers. Rectified linear units ($max(out, 0)$, Relu) are used as active functions. The combination of Relu and dropout can significantly improve the efficiency of CNN. This modification over the traditional convolutional networks was presented in [1]. Our experiments, described below, show its effectiveness in our task.

3.2.2 Row-based Middle Level and Global Level

As age prediction is a high level task, simply concatenating all the low level region specific features may not lead to success in capturing aging cues. Therefore, we propose two hierarchical aggregation levels (middle level and global level) to aggregate the low level region specific aging features in nonlinear way and fine tune our net on the entire facial image.

In the first aggregation level, region-specific features from 4×4 division of the original facial image are aggregated row-wise to a feature representation for each row in the row-specific fully-layer. As human faces are highly symmetric, there should be lots of information redundancy between two regions that correspond symmetrically with each other on the same row. The row-based level aims at aggregating row-wise feature from low level feature extraction while reducing the redundancy in them. It takes 4 region specific features from the same row as inputs and outputs a 110 dimensional row-specific feature vector.

In the second aggregation level, the row-wise 4×1 aging features are further aggregated into a final single feature vector in the global level. The global layer aims at extracting high level aging cues from the entire image and its output is our deep aging feature vector for age prediction. The softmax layer outputs a 11-dimension probability vector corresponding to the age label vector.

In our implementation, aggregation is realized by using a fully connected layer with Relu as active function. In this way, our deep aging feature extracts both local region-specific cues and high level or even global cues in human aging.

3.2.3 Age Labelling and Loss Function

Our method uses a probability distribution representation to encode age, thus transforming the age from a discrete value to a continuous probability vector. The transformed age is fit into the last softmax layer of our network. In our experiments, we partition the age axis into 11 age segments. Each age segment cover an age range of $\delta = 8$ years according to experiments. The resultant age axis partition is $V = [3.5, 11.5, 19.5, 27.5, 35.5, 43.5, 51.5, 59.5, 67.5, 75.5, 83.5]$. The elements in V are the breakpoints θs. Thus an age y is encoded as:

$$f = [0 \ldots \frac{\delta - |y - \theta_i|}{\delta}, \frac{\delta - |y - \theta_{i+1}|}{\delta} \ldots 0], \tag{3.1}$$

where θ_i and θ_{i+1} are the two nearest breakpoints to y, $\theta_i \leq y \leq \theta_{i+1}$. Therefore, a new age label f only contains 2 non-zero elements on the positions i and $i + 1$ while all the other positions are 0. The 2 non-zero elements actually encodes the similarity between the labeled age and its two closest breakpoint values. Given a transformed age label f_i, the corresponding age value is $y_i = V f_i^T$. Based on this age labelling, the loss function of our network is defined as:

$$\mathcal{L} = \sum_{i=1}^{N} \|f_i - \hat{f}_i\|, \tag{3.2}$$

where N is the number of training samples, f_i and $\hat{f}_i$ correspond to the ground truth age and predicted age respectively. Stochastic gradient descent is used with gradients calculated by back-propagation. Note that, since the aging features are extracted from the output of the fully connected layer in the global level, there is no constraint on the output of the softmax layer that makes it contains only two non-zero elements.

By using the above age labelling strategy and loss function, we formulate the age prediction problem as a regression problem. Meanwhile, the proposed age labelling method reduces the number of age labels from a high dimensions (62 in the MORPH-II database) to 11-dimensions, thus effectively reducing the number of parameters and the dimensions of aging features in the net architecture. Since the performance of deep network usually achieves the best when its layer dimensions perform pyramid, that is the dimensions decrease steadily from lower layers to high layers, with a similar ratio [49].

3.3 Experiments

In this section, we report experimental results on various configurations to evaluate the features learnt by our method for age prediction. The performance of age prediction is evaluated by the mean absolute error (MAE) $\sum_{k=1}^{N} |\hat{y_k} - y_k|/N$, where y_k is the ground truth age for the k-th test sample, $\hat{y_k}$ is the predicted age, and N is the number of testing samples.

The aging features are tested with two learning methods - Support Vector Machine (SVM) [50] and Random Forest (RF) [51]. In addition to being widely used as the state-of-art methods, they are of different types. SVM uses a maximum margin approach while Random Forest is an ensemble-based learner. This is to illustrate the effectiveness of our features regardless of the leaner used.

3.3.1 Database

In our experiments, we used the MORPH-II [52] database ,the FG-NET database and the FACES database [53]. The MORPH-II database consists of 55,132 facial images with age, gender and race label. In our experiment setting, 5670 and 1880 face images were randomly chosen as testing set S_1 and validation set S_2 respectively, while the remaining 47K images were used as training set S_3. S_1, S_2 and S_3 share similar distributions of age, gender and ethnic with the entire database, the age distributions of the three sets are shown in Fig.3.2 in the form of accumulated probability distribution. The FG-NET, is used in the cross-database evaluation experiment, for it only contains 1002 face images, which is not enough for the training of CNN. Moreover, the FACES facial image dataset was used in the generalization evaluation experiment only, in order to test the HADF's ability to deal with facial images with expressions. And it only contains 1046 face images, thus is not enough to train a deep network.

In our experiments, all the faces in MORPH-II [52] database and the FG-NET database were cropped and resized to 128×96. To make fair comparison and get rid of the influence of illumination, all the facial image were transformed to grey scale. In the experiments, we observed that our Hierarchical Aggregation based Deep Feature (HADF) shows robustness to small rotations, translations, and scaling in facial image alignment.

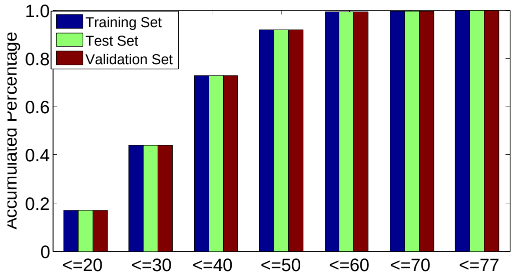


Figure 3.2: The age distribution of training, testing and validation sets. To better illustrate the similarity in age distributions, we used the form of accumulated probability distribution.

3.3.2 Quantitative Comparison

We compared our deep feature HADF with the state-of-the-art features for age prediction. Specifically, we extracted the LBP feature and the BIF feature from S_3 and S_1 and applied the best parameters tested on S_2. The LBP feature aims at capturing primarily the skin aging changes and the BIF feature is designed to capture the deep and apparent wrinkles on the face. The dimension of the BIF feature was reduced using PCA from 11080- dimensions to 1000-dimensions to reduce the noise. Recent work has shown that pre-trained CNN features trained on the ImageNet [2] can also be transferred to new classification or recognition problems and boost remarkable performance. Here we used the CNN features learnt from ImageNet as aging features for age prediction. Zhu et al. [54] trained a CNN for face identification, although it is not for age estimation, their architecture is able to capture high understanding of facial images. Here we used the face identity-preserving features (FIP) [54] for age estimation.

Experimental results are illustrated in Table 4.5. Additionally, we illustrated the accumulated accuracy with respect to absolute age error for each feature in Fig. 3.3. From the table, our HADF feature achieves the lowest age prediction errors, which proves the effectiveness of our proposed features.

Table 3.1: Performance of aging features for age prediction with different classifiers.(In the table, C means classification and R means regression. Same representation will be used in the following parts in this section.)

Method	SVM-C	SVM-R	RF-C	RF-R
LBP [37]	5.21	5.13	5.40	5.24
BIF [41]	4.23	4.17	4.26	4.21
ImageNet [2]	7.35	7.14	7.36	7.16
FIP	5.09	4.92	5.31	5.14
HADF	**3.51**	**3.41**	**3.52**	**3.43**

3.3.3 Influence of Division Size

In the above sections, we fixed the division size as 4 × 4. Here we evaluate the performance of our features with respect to different divisions of sub-images (from 1 × 1, ie. the whole image to 5 × 5). Experimental results are reported in Fig. 3.4 and Table 3.2. The figure shows that the performance of age prediction peaks at the division 4×4. This can be explained as 4×4 division captures the most suitable detailed aging information in the facial images, while 3 × 3 is too rough and 5 × 5 is over-detailed. Additionally, as shown in Table 3.3, the combination of HADF with different division size results in a better result. In this table, the optimum result was achieved by a concatenation of 3 HADF features together with division sizes of $3 \times 3, 4 \times 4$ and 5×5 (with overlaps between adjacent local parts) and then use principal component analysis to reduce the dimensionality to be 110. In this experiment, all aging features are feed into an SVM-R and then predict the age.

Table 3.2: Comparison between different division size

Division	1 × 1	2 × 2	3 × 3	4 × 4	5 × 5
MAE	4.52	3.90	3.64	**3.41**	3.89

Table 3.3: Combination of different division size

Division	3 × 3	4 × 4	5 × 5	3 × 3 + 4 × 4 + 5 × 5
MAE	3.64	3.41	3.89	**3.27**

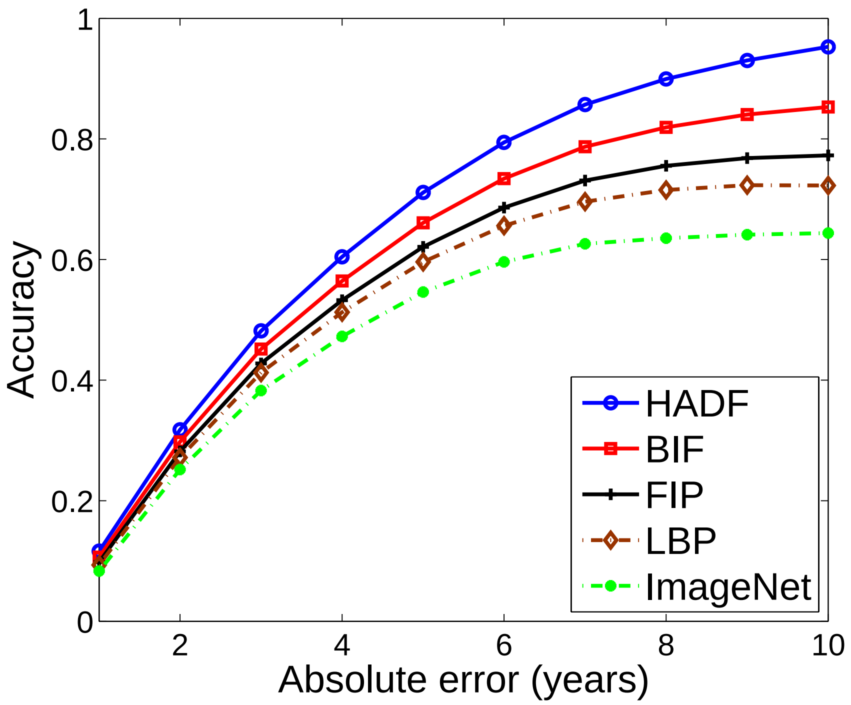


Figure 3.3: Changes of accuracy with respect to absolute age error. The accumulated accuracy is calculated with the change of absolute error of estimated ages. For example, when x-axis is 4, the accuracy on y-axis means if the absolutely estimated error within 4, the estimation is accepted as correct.

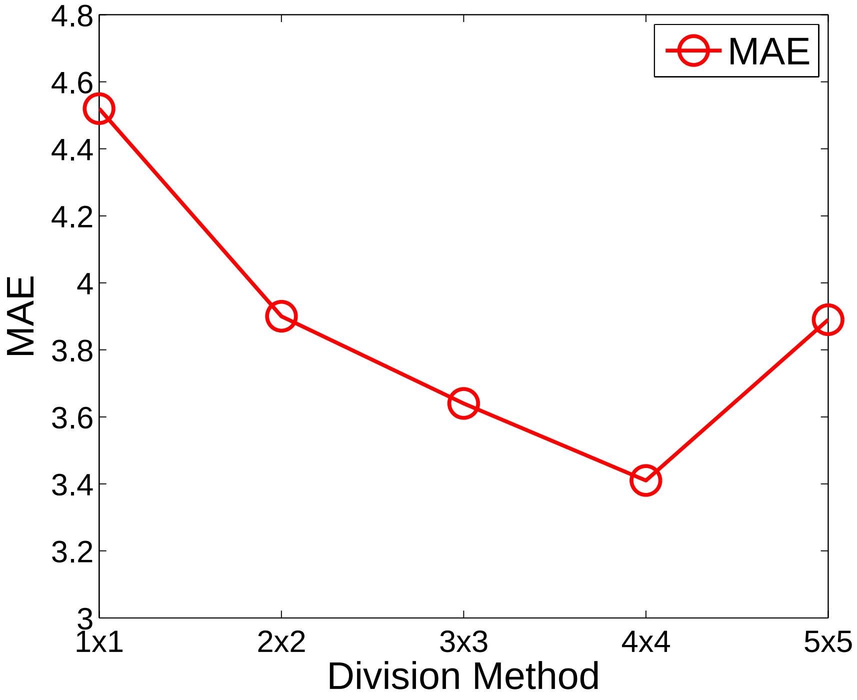


Figure 3.4: Age prediction (MAE) with respect to division size.

3.3.4 Cross-expressions evaluation

In this section, we test the generalization ability of our features on facial images with expressions by carrying out an experiment. This experiment used the FACES database to conduct the evaluation, it investigates if our feature extraction architecture can perform well when applied to extract features on facial images with expressions. Note that, the same architecture learnt from MORPH-II training set is used, the FACES samples are used only for testing and there is no retraining in this process. Moreover, the age distribution of FACES is quite different from MORPH-II, therefore, we only evaluated samples with ages$\leq$77 (936 samples). Table 3.4 demonstrates the results, which shows that our HADF feature achieves the best generalization for the cross expressions age prediction task. Then, Table 3.5 shows the results after we get rid of the neutral expression in FACES dataset (434 samples), in order to test the performance of our HADF on unknown expressions. In this experiment, all aging features are feed into an SVR and then predict the age.

Table 3.4: Generalization ability: Cross-expression experiment.(Including neutral expression)

Method	LBP [37]	BIF [41]	ImageNet [2]	FIP	HADF
MAE	9.87	8.64	9.02	8.89	**7.75**

Table 3.5: Generalization ability: Cross-expression experiment.(Without neutral expression)

Method	LBP [37]	BIF [41]	ImageNet [2]	FIP	HADF
MAE	10.70	9.81	11.34	10.91	**8.69**

3.3.5 Cross-ethnic and Cross-gender evaluation

In this section, we test the generalization ability of our features by carrying out two experiments. In the first experiment, all the African people in the MORPH-II database (about 42K face images) were used as a training set and other races (about 13K face images) were used as the testing set. Experimental results are reported in Table 3.6, where our HADF feature achieves the best performance.

In the second experiment, all the male people in the MORPH-II database (about 47K face images) were used as a training set and female people (about 8K face

Table 3.6: Generalization ability: Cross-ethnic experiment on the MORPH-II database.

Method(MAE)	SVM-C	SVM-R	RF-C	RF-R
LBP [37]	6.23	6.11	6.29	6.16
BIF [41]	5.77	5.60	5.85	5.72
ImageNet [2]	7.61	7.14	7.36	7.24
FIP	5.92	5.79	6.07	5.93
HADF	**4.79**	**4.72**	**4.88**	**4.80**

images) were used as the testing set. Experimental results are reported in Table 3.7, where our HADF feature achieves the best performance. All the three experiments further prove the generalization ability and robustness of our HADF deep feature.

Table 3.7: Generalization ability: Cross-gender experiment on the MORPH-II database.

Method(MAE)	SVM-C	SVM-R	RF-C	RF-R
LBP [37]	7.13	7.05	7.20	7.11
BIF [41]	6.19	5.98	6.24	6.07
ImageNet [2]	7.56	7.38	7.69	7.42
FIP	6.25	6.11	6.31	6.19
HADF	**4.51**	**4.43**	**4.68**	**4.53**

These two experiments shows the strong stability of our HADF feature, which comes from the combination of local details and global information of face images. Different with LBP and BIF, our HADF extracts global information at the Global Level. And compared with FIP and Imagenet, HADF extracts more detail information at the Region-specific Local Level. Therefore, the combination of global information and local details in HADF results in its stability in the cross-ethnic and cross-gender experiments.

3.3.6 Efficiency

The experiments in the previous section show the effectiveness of the proposed learnt features. In this section, we discuss the efficiency of the features in two aspects: feature extraction and age estimation. While the other features have relatively high dimensions (more than 1000), (1180,11080,4096,512 for LBP, BIF, ImageNet and FIP respectively), the dimension of HADF is just 110-dimension.

Moreover, the features are extracted in one pass in by our architecture which is much faster than the other features (pixel sampling(LBP), or bank of Gaussians with different orientations and scales(BIF)). Efficiency is extremely important to implementations with limited computational ability, such as wearable devices (Google Glass) where realtime processing is a must. To compare with other features in low-dimension scale, PCA is used to reduce the dimension of LBP and BIF into 110. Table. 3.8 shows the results using SVM-R (as it gives the best results for all features). For LBP and BIF, the MAE increased by about 10% and 20% when reducing the dimension to be 110 and 11 respectively. This further shows the compactness advantage of the proposed features over the other features.

Table 3.8: Comparison between four types of features and HADF;Local Binary Pattern(LBP) and Bio-Inspired Features(BIF).(The dimensions of LBP and BIF were reduced to 110.)

Method(MAE)	110-D	All-D
LBP [37]	5.73	5.13
BIF [41]	5.01	4.17
ImageNet [2]	7.62	7.14
FIP	5.53	4.92
HADF	**3.41**	**3.41**

3.3.7 Generalizability evaluation

We test the generalizability of our features by carrying an experiment by using FGNET database to do the cross-database evaluation. This experiment investigates if our feature extraction architecture can perform well when used to extract features from different databases. Note that, the same architecture learnt from MORPH-II train set is used, the FGNET samples are used only for testing and there is no retraining in this process. Moreover, the age distribution of FGNET is quiet different with MORPH-II, therefore, we only evaluate samples with ages$\geq$16 (427 samples). The best performance was achieved with our features which further suggests the robustness of the proposed features.

3.3.8 The effect of age label encoding

Table. 3.10 shows the effects of our proposed age labelling method. The framework without the proposed age label encoding method use the original age label as the

Table 3.9: Comparison of generalizability between four types of features,Local Binary Pattern(LBP), Bio-Inspired Features(BIF), ImageNet Features, Face Identity-preserving Features (FIP) and the proposed feature HADF using different classifiers.

Method(MAE)	SVM-C	SVM-R	RF-C	RF-R
LBP [37]	7.87	7.79	7.94	7.86
BIF [41]	7.76	7.67	7.89	7.70
ImageNet [2]	8.91	8.78	9.10	8.93
FIP	7.69	7.60	7.81	7.71
HADF	**7.49**	**7.42**	**7.61**	**7.45**

target output. Parameters with the best performance are used and the extracted features are fed into different kinds of classifiers to produce an overall evaluation of the proposed age label encoding method. It is obviously that our proposed age label encoding method significantly improve the performance of the framework.

Table 3.10: Comparison of the results between age label encoding and age label without encoding (the same deep structure was used).

Method(MAE)	SVM-C	SVM-R	RF-C	RF-R
Age Label Encoding	3.51	3.41	3.52	3.43
No Age Label Encoding	4.62	4.57	4.77	4.61

3.4 Summary

This chapter has presented a new, hierarchical, aggregation-based deep network to extract aging features from facial images. We employ region specific convolutional neural network (CNN) at lower layers. These low layer features are hierarchically aggregated into consecutive higher layers. Our aging feature is of dimensionality 110 and achieves both good discriminative ability and efficiency. Experimental results of age prediction on the MORPH-II and the FG-NET databases show that our method outperforms other state-of-the-art systems for age estimation.

Chapter 4

Surface Normal Constrained Single Image Depth Estimation

4.1 Introduction

Depth estimation is an important task in computer vision area and is the basic of many applications, such as 3D reconstruction [55]. Traditionally, depth estimation is solved by algorithms that require multiple images, such as structure from stereo images and motion [56]. Compared with these algorithms, estimating depth of a scene from a single image is a highly ambiguous problem due to the lack of depth cues and inadequate geometry constraints, and thus this challenging task attracts lots of attention.

Recently, several approaches that estimate depth from a single image are proposed. Saxena et al. [57] predict depth from a set of image features using linear regression and a Markov Random Field (MRF). However, this framework highly relies on horizontal alignment of images, which limit its generability. While Ladicky et al. [58] improve the performance by combining using of semantic label and monocular depth features, this system relies on handcrafted features. Then, Karsh er al. [59] implement a SIFT Flow-based kNN transfer mechanism to estimate the depth from single image, however, it performs expensive alignment procedures at run time.

More recently, data-driven depth estimation on single image, which directly learns to predict scene geometry from data, gains popularity. An instance is CNN-based frameworks. Nowadays, lots of CNN-based frameworks on depth estimation from single image achieve the state-of-art performance both on computational efficiency and accuracy. Eigen et al. [60] designed a end-to-end structure to directly estimate the depth map of single image and achieved state-of-the-art performance.

Moreover, Li et al. [6] proposed a two stages framework, which predicts depth map on super-pixel level first and then refined it to pixel level by using a CRF model. Although the accuracy of [60] is high, the computing complexity of this structure is too large, which limits its implementation on real-time situation. While [6] solved the problem of high computational complexity, its last layer at the first stage is Euclidean loss layer for regression tasks and the deep features it used are classification features, which may affect the final performance.

In this chapter, we present a new framework consisting of depth regression via deep features and depth refining via surface normal constraints. Firstly, we use a deep network and formulate the problem of depth estimation as a deep feature classification problem to exploit the relation between a color image and its corresponding depth. A multi-scale deep feature is extracted by a deep network. Secondly, to further refine the depth maps and achieve efficient estimation, we present a surface normal constraint model to take various potentials into consideration to estimate depth for each pixel, thus upgrading the depth estimation from super-pixel level to pixel level by the usage of more strict mathematical constraint given by surface normal.

4.2 Method

In this subsection, we introduce a pixel-level single image depth estimation method, which consists two stages: depth regression and depth refining by using surface normal constraints. Firstly, we formulate super-pixel level depth estimation as a classification problem by depth encoding, which is similar with the work by Li et al. [6]. Secondly, we refine the depth estimation from super-pixel level to pixel level by using surface normal constraints. The relationship between depth and surface normal within a small plane leads to an approximate constraint, which refines the depth more accurately. The Fig. 4.1 shows the overall flowchart of our proposed framework.

4.2.1 Depth prediction via deep network

As shown in Fig. 4.1, in the first stage, we initialize the lower layers of our deep network with pre-trained CNN and keep the parameters of pre-trained layers unchanged in the training process. Then, the last three fully connected layers are used as a classifier to estimate depth. An important contribution of this chapter is that we transfer the real value depth label to a possibility vector by applying

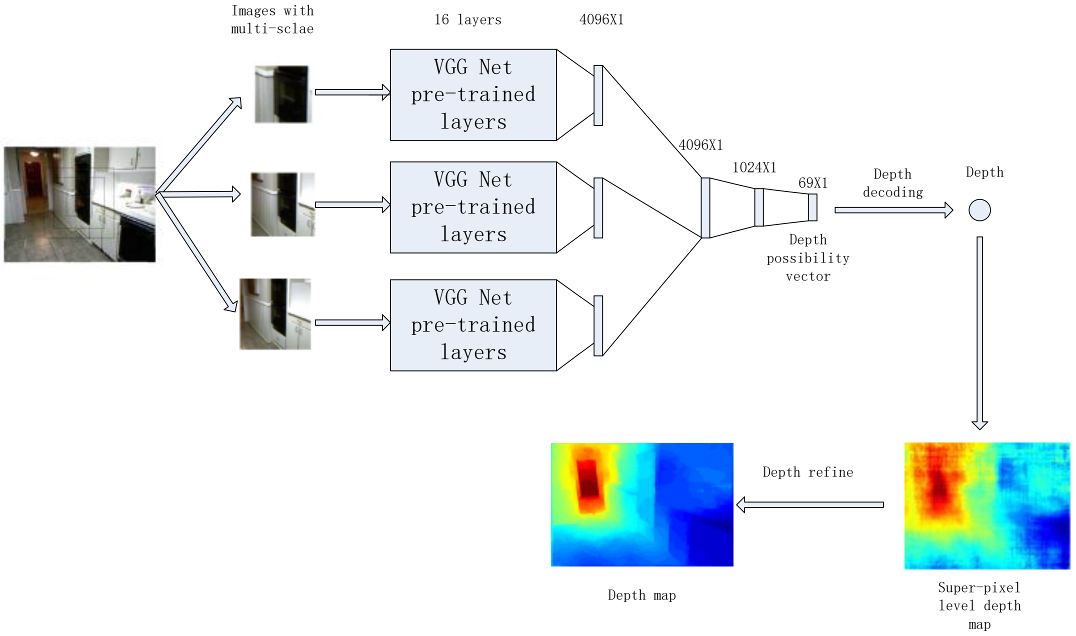


Figure 4.1: Overall flowchart of our proposed framework.

proposed depth coding method, which enabling the transition of depth estimation from a regression problem to a classification problem.

It can also be seen from Fig. 4.1 that our deep network use multi-scales images which centered at the chose point as inputs. The proposed network utilizes the first 16 layers of pre-trained VggNet [5] as lower layers. These pre-trained layers keep unchanged and independent from each other in the training process. At the 17th layer, we obtain 3 4096 × 1 sized features from three scales inputs, respectively. Then the last 3 fully connected layers conduct feature fusion and depth estimation task.

Depth Coding

Our method uses a probability distribution representation to encode depth, thus transforming depth value to a continuous probability vector. The transformed depth is fit into the last softmax layer of our network. In our experiments, the depth values are first transfered into log space and shift 0.5 to right direction of axis, which ensuring all encoded depth labels are positive. Then, we partition the depth axis into 69 segments, which is decided by experiments. Each depth segment cover an depth range of $\delta = 0.04$. The resultant depth axis partition is $V = [0, 0.04, 0.08, ... , 1.40, 1.44, 1.48, ... , 2.68, 2.72, 2.76]$. The elements in V are the breakpoints θs. Thus an depth y is encoded as:

$$f = [0 \dots \frac{\delta - |y - \theta_i|}{\delta}, \frac{\delta - |y - \theta_{i+1}|}{\delta} \dots 0], \tag{4.1}$$

where θ_i and θ_{i+1} are the two nearest breakpoints to y, i.e. $\theta_i \leq y \leq \theta_{i+1}$. Therefore, a new depth label f only contains 2 non-zero elements on the positions i and $i+1$ while all the other positions are 0. The 2 non-zero elements actually encodes the similarity between the labeled depth and its two closest breakpoint values. Given a transformed depth label f_i, the corresponding depth value is $y_i = V f_i^T$. Based on this depth labelling, the loss function of our network is defined as:

$$\mathcal{L} = \sum_{i=1}^{N} \|f_i - \hat{f}_i\|, \tag{4.2}$$

where N is the number of training samples, f_i and $\hat{f}_i$ correspond to the ground truth depth and predicted depth respectively. Stochastic gradient descent is used with gradients calculated by back-propagation.

The reason for implementation of depth encoding in our network is the usage of pre-trained layers of VggNet, which is a network for classification task. Thus its pre-trained layers perform better on conducting classification task compared with regression task. Therefore, by transferring the regression task to classification task, the last three fully connected layers of our network focus on estimating depth.

Effect of multi-scale feature

Our deep network use a multi-scale blocks to extract depth cues, we used three sizes blocks, 121×121, 271×271 and 407×407. Table. 4.1 shows that the accuracy of depth estimation improved with the increase of block size, because more graphic information and depth cues are extracted with the increasing of block size. And the concatenation of multi-scale blocks features result in more improvement, because it contains both global information and detailed information of depth cues, which improve the final prediction. This conclusion is also shows in [6]. However, Table. 4.1 also shows that the 55×55 sized blocks effects slightly on the concatenation results, which means that over-detailed information is not necessary in depth prediction. Therefore, in the following experiments in this section, we only use 3 sized blocks, which achieves a relatively good result and efficiency at the same time.

Table 4.1: Depth prediction results on the NYU V2 data set under different size of single scale block setting and multi-scale setting.

block size	Rel	log10	Rms
55 × 55	0.2819	0.1142	0.9791
121 × 121	0.2371	0.0957	0.8339
271 × 271	0.2026	0.0851	0.7714
407 × 407	0.1994	0.0832	0.7652
4 blocks	**0.1908**	**0.0789**	**0.7245**
3 blocks(without 55 × 55)	**0.1909**	**0.0791**	**0.7244**

4.2.2 Surface normals from depth map

In the above section, we shows how we predict depth of super-pixels by our deep network. In this section, our goal is to refine the depth estimation from super-pixel level to pixel level by enforcing surface normal constraints.

Virtual disparity

Disparity is generally defined in stereo vision problem, which characterizes the displacement in image pixel between the left image and the right image. For monocular depth estimation, we aim to estimate depth d_i for each pixel. To facility the formulation and computation, we denote $r_i = \frac{1}{d_i}$ as a virtual disparity for that pixel. Note that in stereo vision, disparity and depth are related as:

$$r_i = \frac{fB}{d_i}, \tag{4.3}$$

where f is the focal length of the camera and B is the distance between the camera centers. In our experiments, these parameters are given by NYU2 Datasets.

From depth map to point cloud data

Under the perspective camera model, a 3D point $\mathbf{p}_i = (x_i, y_i, z_i)^T$ is projected to an image point (2D point) (u_i, v_i) as:

$$\lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}\ \mathbf{t}] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \tag{4.4}$$

where $\mathbf{K}$ is the intrinsic matrix while $(\mathbf{R}, \mathbf{t})$ is the extrinsic matrix. For the NYU2 datasets, we have $\mathbf{R} = \mathbf{I}$ and $\mathbf{t} = \mathbf{0}$. Therefore, the intrinsic matrix $\mathbf{K} = \mathrm{diag}(f, f, 1)$ where f is the focal length. The imaging process can be simplified as:

$$\lambda_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}. \tag{4.5}$$

where u_0 and v_0 are the position of centre point of the image.

Given the estimated depth d_i for the 3D point $\mathbf{p}_i$ and intrinsic camera matrix $\mathbf{K}$, the 3D point is derived as:

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \frac{d_i(u_i - u_0)}{f} \\ \frac{d_i(v_i - v_0)}{f} \\ d_i \end{bmatrix}. \tag{4.6}$$

Under the spherical coordinate, $r_i = \sqrt{x_i^2 + y_i^2 + z_i^2} = d_i\sqrt{(\frac{(u_i-u_0)}{f})^2 + (\frac{(v_i-v_0)}{f})^2 + 1}$, which is proportional to d_i.

Total least squares

In 3D space, a plane is defined by the equation:

$$n_x x + n_y y + n_z z - d = 0, \tag{4.7}$$

where $(x, y, z)^T$ is the cloud of points lie on the plane, (n_x, n_y, n_z) is the normal of the plane and d is the scalar element. Given a subset of k 3D points $\mathbf{p}_i$ of the surface, least squares finds the optimal surface normal vector $\mathbf{n} = (n_x, n_y, n_z)^T$ and scalar d that minimizes:

$$e = \sum_{i=1}^{k} (\mathbf{p}_i^T \mathbf{n} - d)^2, \quad \text{subject to} \quad \|\mathbf{n}\| = 1. \tag{4.8}$$

The closed-form solution for $\mathbf{n}$ is given by finding the eigenvector corresponding to the smallest eigenvalue of the sample covariance matrix $\mathbf{M} = \frac{1}{k}\sum_{i=1}^{k}(\mathbf{p}_i - \overline{\mathbf{p}})(\mathbf{p}_i - \overline{\mathbf{p}})^T$ with $\overline{\mathbf{p}} = \frac{1}{k}\sum_{i=1}^{k}\mathbf{p}_i$. A problem is the closed-form solution for $\mathbf{n}$ involves eigen value decomposition, which makes the optimization complex. We will solve this problem in the following sections.

Fast Approximate Least Squares

According to [61], by using the spherical coordinate, the cost function in Equation. 4.8 is evaluated as:

$$e = \sum_{i=1}^{k} (r_i \mathbf{v}_i^T \mathbf{n} - d)^2, \tag{4.9}$$

where r_i is the spacial distance of image points and v_i is the surface normal vector of image points.

Dividing both sides of the equation with d^2, we have

$$\tilde{e} = \sum_{i=1}^{k} (r_i \mathbf{v}_i^T \tilde{\mathbf{n}} - 1)^2 = \sum_{i=1}^{k} r_i^2 (\mathbf{v}_i^T \tilde{\mathbf{n}} - r_i^{-1})^2, \tag{4.10}$$

where $\tilde{\mathbf{n}}$ is the sought normal vector defined up to a scale. d is the signed distance from the origin to the plane. Note that $\mathbf{v}_i$ depends on the image coordinate only.

Since all points $\mathbf{p}_i$ are in a small neighborhood, all $r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ are similar. Dropping the r_i^2 from the above equation leads us to the following approximate formulation of the loss function:

$$\hat{e} = \sum_{i=1}^{k} (\mathbf{v}_i^T \hat{\mathbf{n}} - r_i^{-1})^2, \tag{4.11}$$

whose solution for $\hat{\mathbf{n}}$ is given by:

$$\hat{\mathbf{n}} = \hat{\mathbf{M}}^{-1} \hat{\mathbf{b}}, \tag{4.12}$$

with $\hat{\mathbf{M}} = \sum_{i=1}^{k} \mathbf{v}_i \mathbf{v}_i^T$ and $\hat{\mathbf{b}} = \sum_{i=1}^{k} \frac{\mathbf{v}_i}{r_i}$. Set $s_i = \frac{1}{d_i}$, then

$$\hat{\mathbf{b}} = \sum_{i=1}^{k} \mathbf{v}_i s_i \frac{1}{\sqrt{(\frac{(u_i - u_0)}{f})^2 + (\frac{(v_i - v_0)}{f})^2 + 1}}, \tag{4.13}$$

In this formulation, the matrix $\hat{\mathbf{M}}^{-1}$ is independent of the depths, and depends only on the image coordinate, thus can be precomputed.

The surface normal is found by normalization, i.e. $\mathbf{n} = \frac{\hat{\mathbf{n}}}{\|\hat{\mathbf{n}}\|}$.

We further denote $g_i = \sqrt{(\frac{(u_i - u_0)}{f})^2 + (\frac{(v_i - v_0)}{f})^2 + 1}$, which is independent of the depth map.

One way to deal with the inverse depth is to optimize over $\frac{1}{r_i} = \frac{1}{d_i} \frac{1}{g_i}$. Due to the relationship between depth and disparity, $\frac{1}{r_i}$ and $\frac{1}{d_i}$ play similar role as disparity. During the computation, $\hat{\mathbf{n}} = \frac{\mathbf{n}}{d}$ and $\mathbf{n} = d\hat{\mathbf{n}}$, where d is the point-plane distance between the camera center and the local 3D plane. In the iteration, we used $d^{(it-1)}$

, which means the value of d in the $it - th$ iteration, instead.

Efficient computation

Surface normal is determined from a small region, where a 10×10 sized square neighbour area of the current pixel is used in this thesis.

$$r_i = d_i g_i, \tag{4.14}$$

where g_i can be pre-computed while d_i is the estimated depth from our deep network.

4.2.3 Proposed surface normal constrained depth refinement

Given a depth map, we could compute the surface normal using the method introduced in above sections. Our energy function is similar with [6], which consists of three terms, namely the data term, the smoothness term and the auto-regression term at the pixel level. The smoothness term defined at the super-pixel level improves the smoothness between adjacent super-pixels. For auto-regression potential term, the key insight behind this term is that the depth channel and RGB channel are locally correlated, thus we can characterize the local structure of the depth map with the guidance of the corresponding colour image. The coefficients of the auto-regression term are extracted from the corresponding colour image. However, in our energy function we further refine the depth by using a data-driven surface normal constraint. Here is our energy function:

$$\mathbf{E}(\mathbf{d}) = \sum_{i \in \mathcal{S}} \phi_i(d_i) + \sum_{(i,j) \in \mathcal{E}_\mathcal{S}} \phi_{ij}(d_i, d_j) + \sum_{\mathcal{C} \in \mathcal{P}} \phi_\mathcal{C}(\mathbf{d}_\mathcal{C}) + \mathcal{R}(\mathbf{n}(\mathbf{d})), \tag{4.15}$$

where $\mathcal{E}_\mathcal{S}$ denotes the set of pairs of super-pixels that share same boundary, $\mathcal{S} = s_1, ..., s_m$ is the set of super-pixels and $\mathcal{P}$ is the set of patches designed on the pixel level. Now, we explain the potentials used in Eq. 4.15.

Potential 1: Data term

$$\sum_{i \in \mathcal{S}} \phi_i(d_i) = (d_i - \overline{d_i})^2, \tag{4.16}$$

where $\overline{d_i}$ is the depth prediction results from our deep network. This term is defined to measure the quadratic distance between the predicted depth d_i and the regressed depth at super-pixel level.

Potential 2: Smoothness term

$$\sum_{(i,j)\in\mathcal{E}_{\mathcal{S}}} \phi_{ij}(d_i, d_j) = w_1(\frac{d_i - d_j}{\lambda_{ij}})^2, \tag{4.17}$$

this super-pixel level smoothness term enhances coherence between adjacent super-pixels.

Potential 3: Auto regression term

$$\sum_{\mathcal{C}\in\mathcal{P}} \phi_{\mathcal{C}}(\mathbf{d}_{\mathcal{C}}) = w_2(d_u - \sum_{r\in\mathcal{C}_u} \alpha_{ur} d_r), \tag{4.18}$$

where d_u is the predicted depth by the regression model, C_u is the neighbour area of pixel u and λ_{ur} is the model self-expressive coefficient for pixel r in the neighbour area C_u. Similar with [6], we set $\lambda_{ur} \propto e^{-\frac{1}{2}(\frac{g_u - g_r}{\sigma_u})^2}$ and $\sum \alpha_{ur} = 1$, where g represents the intensities value of corresponding pixels and σ_u is the variance of the intensities in the local patch around u.

Potential 4: Surface normal constraint term

$$\mathcal{R}(\mathbf{n}(\mathbf{d})) = \omega_3 \sum_{i\in\mathcal{S}} \|\mathbf{n}_i(\mathbf{d})\|_{\text{TV}}, \tag{4.19}$$

where the TV denotes the total variation. The first 3 potentials solve the problem of smooth, however, they focus on local areas, which means a lack of global refine of the depth map. Therefore, this surface normal constraint term is introduced to refine the depth map from a global level.

We decouple the problem into two sub-problems by introducing an auxiliary variable $d_i = 1/s_i$: enforcing $d_i s_i = 1$. Then, our energy function is:

$$\mathbf{E}(\mathbf{d}, \mathbf{s}) = \sum_{i\in\mathcal{S}} \phi_i(d_i) + \sum_{(i,j)\in\mathcal{E}_{\mathcal{S}}} \phi_{ij}(d_i, d_j) + \sum_{\mathcal{C}\in\mathcal{P}} \phi_{\mathcal{C}}(\mathbf{d}_{\mathcal{C}}) + \frac{1}{2\theta}\sum_{i=1}^{n}(d_i s_i - 1)^2 + \sum_{i\in\mathcal{S}} \phi_{\mathbf{n}}(\mathbf{n}_i(\mathbf{s})), \tag{4.20}$$

where θ is large at the beginning of the optimization process, which gives some unconstraint space on the strict mathematical constraint $d_i s_i = 1$. Then, *theta* decrease with the increase of the number of iterations in optimization process, which makes the mathematical constraint $d_i s_i = 1$ more strict. Therefore, this potential add a global constraint on depth estimation process, which results in a global optimization solution.

Update of d:

$$\mathbf{d} = \arg\min_{d} \mathbf{E}_1(\mathbf{d}) = \sum_{i \in \mathcal{S}} \phi_i(d_i) + \sum_{(i,j) \in \mathcal{E}_\mathcal{S}} \phi_{ij}(d_i, d_j) + \sum_{\mathcal{C} \in \mathcal{P}} \phi_\mathcal{C}(\mathbf{d}_\mathcal{C}) + \frac{1}{2\theta} \sum_{i=1}^{n} (d_i s_i - 1)^2, \tag{4.21}$$

Then, the update of **d** owns a closed-form solution. Because the **S** is constant in this step of updating **d**, therefore, it is not appear in the variable position.

$$\mathbf{E}(\mathbf{d}) = \|\mathbf{Hd} - \overline{\mathbf{d}}\|_2^2 + w_1 \|\mathbf{QHd}\|_2^2 + w_2 \|\mathbf{Ad}\|_2^2 + \frac{1}{2\theta} \|\mathbf{Sd} - \mathbf{1}\|_2^2, \tag{4.22}$$

where $\mathbf{S} = \mathrm{diag}(s_1, \cdots, s_i, \cdots, s_n)$, $\overline{d}$ is the output of the regression model, **H** is the indication matrix to select corresponding super-pixel, **Q** represents the neighbouring relationship of super-pixels and **A** is the neighbouring matrix corresponding to the regressive model in local patch.

As the energy function is quadratic with respect to **d**, a closed-form solution can be derived algebraically:

$$\mathbf{d}_{\mathrm{MAP}} = (\mathbf{H}^\top \mathbf{H} + w_1 \mathbf{H}^\top \mathbf{Q}^\top \mathbf{QH} + w_2 \mathbf{A}^\top \mathbf{A} + \frac{1}{2\theta} \mathbf{S}^T \mathbf{S})^{-1} (\mathbf{H}^\top \overline{\mathbf{d}} + \frac{1}{2\theta} \mathbf{S}^T \mathbf{1}), \tag{4.23}$$

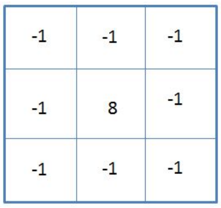
Update of s:

$$\mathbf{s} = \arg\min_{s} \mathbf{E}_2(\mathbf{d}, \mathbf{s}) = \frac{1}{2\theta} \sum_{i=1}^{n} (d_i s_i - 1)^2 + \sum_{i \in \mathcal{S}} \phi_\mathbf{n}(\mathbf{n}_i(\mathbf{s})), \tag{4.24}$$

$$\mathbf{s} = \arg\min_{\mathbf{s}} \mathbf{E}_2(\mathbf{d}, \mathbf{s}) = \frac{1}{2\theta} \sum_{i=1}^{n} (d_i s_i - 1)^2 + \sum_{i \in \mathcal{S}} \|w_i d_i \mathbf{n}(S)\|_{TV}, \tag{4.25}$$

therefore,

$$\mathbf{E}(\mathbf{s}) = \frac{1}{2\theta} \|\mathbf{Sd} - \mathbf{1}\|_2^2 + \sum_{i \in \mathcal{S}} \| \sum_{j \in \mathcal{R}} w_{ij} \hat{n}_{ij} \|_{TV}, \tag{4.26}$$



-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 4.2: An example of 3 × 3 sized w_{ij}.

where $\mathcal{R}$ denotes the neighbour area of pixel i and since the smoothness term $\sum_{i\in\mathcal{S}} \| \sum_{j\in\mathcal{R}} w_{ij}\hat{n}_{ij}\|^2$ is dynamic, which means do not exit a closed-form, we calculate the numerical gradient of this energy function. For the weights w_{ij}, we used a filter same with edge detector, which enforce the smoothness of surface normal within a small area. An example of implemented w_{ij} is shown in Fig. 4.2. Table. 4.3 and Fig. 4.3 show the relationship between the depth prediction results and the size of w_{ij}. In this experiment, the step size for numerical gradient calculation is 0.001 and the step size for gradient change is 0.01. And the super-pixel level depth gave out by framework in Sec. 4.2.1. It can be seen that the size 5 × 5 gives the best result among all tested sizes, in the following experiments, we use 5 × 5 sized w_{ij}.

Table 4.2: Depth prediction results with different size of w_{ij}.

filter size	3 × 3	5 × 5	7 × 7	9 × 9	11 × 11
Rel	0.1913	**0.1909**	0.1921	0.2007	0.2261

Then, the algorithm of our surface normal constrained depth refinement shows below:

```
Require: Depth prediction $d_i, i \in \mathcal{S}$, Surface normal prediction $\mathbf{n}_i, i \in \mathcal{S}$,
  super-pixel over-segmentation of the input image, camera intrinsic matrix $\mathbf{K}$.
  Initialize: $\theta_0$ and $\eta$.
  while Not converged do
    1). Update $\mathbf{s}$ iteratively;
    2). Update $\mathbf{d}$ by using the closed-form solution;
    3). Check the convergence conditions: $\|d_{k+1} - d_k\|_\infty \leq \epsilon_1$,
    $\sum_{i=1} \|d_i s_i - 1\| \leq \epsilon_2$;
    4). Update $\theta$.
  end while
Ensure: Depth map $\mathbf{d}$ and surface normal map $\mathbf{r}$.
```

Algorithm 2: Surface normal constrained single image depth estimation.

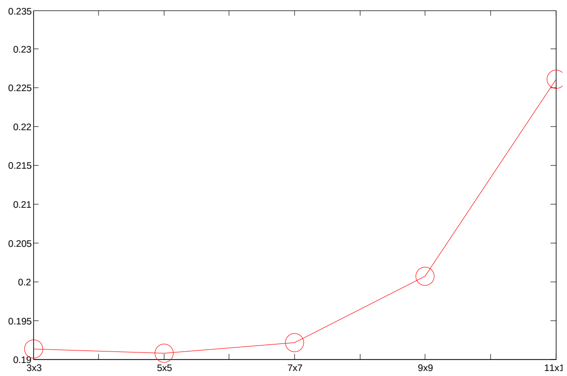


Figure 4.3: Depth prediction results with different size of w_{ij}.

4.3 Experiments

In this section, we report our experimental results on NYU V2 Kinect dataset, since it is one of the most used dataset for indoor depth prediction. We compared our method with all the state-of-the-art methods published recently. Some examples of our result depth map shows in Fig. 4.4.

Error metrics Following error metrics are used in our report of errors for quantitative evaluation. These error metrics are widely used [12, 60].

1. Mean relative error (Rel): $\frac{1}{|T|}\sum_{d\in T}|\hat{d}-d|/d$
2. Mean log_{10} error (log_{10}): $\frac{1}{|T|}\sum_{d\in T}|log_{10}\hat{d}-log_{10}d|$
3. Root mean squared error (Rms): $\sqrt{\frac{1}{|T|}\sum_{d\in T}\|\hat{d}-d\|^2}$

where d is the ground truth depth, $\hat{d}$ is the predicted depth and T is the set of all points in the images.

4.3.1 Database and experiment results

In our experiments, we used the NYU2 Kinect Dataset, which contains 1449 images, where 795 images are used for training and 654 images are used for testing. In

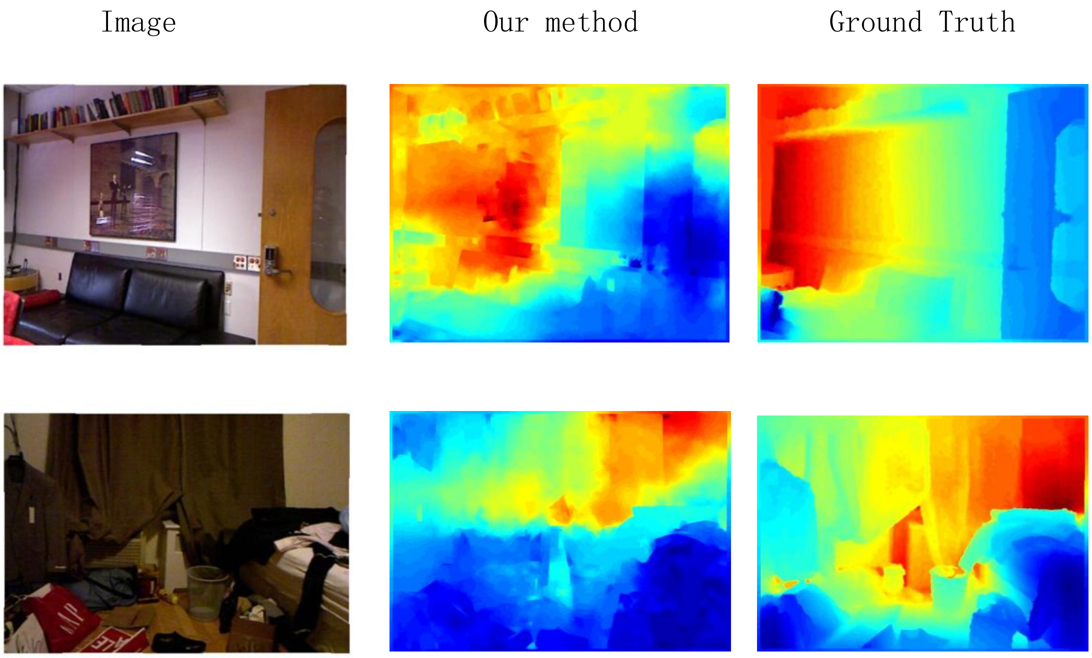


Figure 4.4: Examples of depth estimation from our framework.

Table. 4.3, we compared our method with state-of-art methods: depth transfer [59], discrete-continuous depth estimation [62], pulling things out of perspective [58], multi-scale deep network [60] and [6].

Table 4.3: Comparison of depth prediction errors with different methods.

Method	Rel	log10	Rms
Liu [62]	0.335	0.127	1.06
Eigen [63]	**0.158**	-	-
Depth transfer [59]	0.374	0.127	1.12
Li [6]	0.223	0.0907	0.759
Eigen [60]	0.215	0.094	0.871
regression only	0.2133	0.0886	0.7725
Our method	**0.1909**	**0.0791**	**0.7244**

Analysis It can be seen from Table. 4.3 that the results obtained from our proposed framework outperform most state-of-the-art algorithms except the framework proposed by Eigen et al. [63].This framework achieves the best performance in this experiment, however, it estimates depth by an end-to-end CNN structure, which means its computational complexity is high. As for our proposed framework, the part with the highest computational complexity is the three fully connected layers,

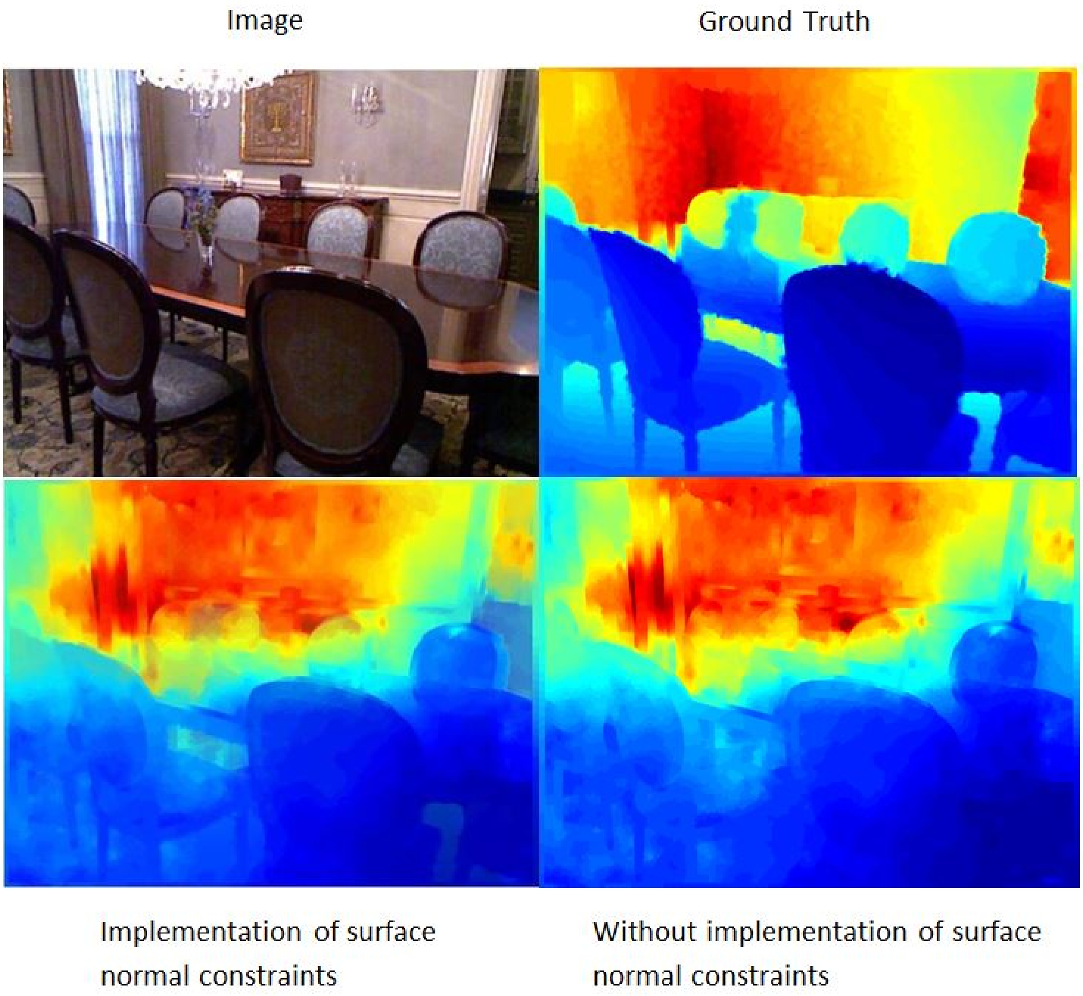


Figure 4.5: Comparison between depth refine results with and without surface normal constraint.

which contains much less trainable parameters compared with Eigen's framework, thus is more efficient.

4.3.2 Effect of surface normal constraints

Fig. 4.5 and Table. 4.4 shows the effects of surface normal constraint in the depth refine process. As shown in Table. 4.4, the surface normal constraint almost has no effect on the final depth estimation performance. However, as shown in Fig. 4.5, the depth map we obtained by applying surface normal constraint in depth refine process is more smooth than depth map without applying surface normal (obvious at the region with larger depth), which means the surface normal constraint improves the visual quality of depth map.

Table 4.4: Comparison between depth refine results with and without surface normal constraint.

Method	Rel	log10	Rms
Depth refine without surface normal constraint	0.1908	0.0791	0.7244
Depth refine with surface normal constraint	0.1909	0.0790	0.7243

Analysis Although the experiment results show that the surface normal term almost has no improvement on depth estimation performance, the potential of surface normal constraint is large, since the w_{ij} we implement have only smooth effect on depth map. If w_{ij} is designed dynamically with the change of image patch, which means the surface normal term conducts smoothness task on plane area and conducts sharpness task when it encounters the area with two different plane, the performance will be definitely improved.

4.3.3 Effect of depth label encoding

Table. 4.5 shows our depth coding method improved the results. Even using the same pre-trained lower layers with [6], the improvement with using our depth coding method is obviously.

Table 4.5: The improvement by using the depth coding to transfer the depth prediction into a classification method.

Method	Rel	log10	Rms
VggNet with Depth Coding	**0.1909**	**0.0791**	**0.7244**
VggNet without Depth Coding	0.211	0.086	0.761
AlexNet with Depth Coding	0.2123	0.089	0.7754
AlexNet without Depth Coding [6]	0.232	0.094	0.821

4.4 Summary

In this chapter, we present a new framework for depth estimation from a single image, which consists of depth prediction via a deep network and depth refining via surface normal constraints. The result is very promising. However, in the surface constraints part, we focus only on the smoothness of surface normals on the plane without considering color and edge information. In the future, we plan to add the color and edge information in the surface constraints parts.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we implement our proposed method of label encoding on both age estimation from facial image and depth estimation from a single image tasks, after combining with some innovations of deep structure, the results show that our proposed label encoding method successfully transfers the regression tasks to be classification tasks and achieves a promising performance.

In Chapter 3, we have presented a new, hierarchical, aggregation-based deep network to extract aging features from facial images. We employ region specific convolutional neural network (CNN) at lower layers. These low layer features are hierarchically aggregated into consecutive higher layers. Our aging feature is of dimensionality 110 and achieves both good discriminative ability and efficiency. Experimental results of age prediction on the MORPH-II and the FG-NET databases show that our method outperforms other state-of-the-art systems for age detection.

In Chapter 4, we has presented a new framework for depth estimation from a single image, which consists of depth prediction via a deep network and depth refining via surface normal constraints. The result is very promising.

5.2 Future Work

In this section, we outline a number of future research directions that arise from the work presented in this thesis.

Age estimation task: In this dissertation, the proposed framework worked well on age estimation task and achieved state-of-art results. In the future, we want to extend it to be a multi-task framework, which is able to recognize the race, gender and age at the same time. Moreover, these three pieces of information could enhance each other during the training and testing process, which may lead to better results.

Depth estimation task: In this dissertation, the proposed framework works well on depth estimation from single image task and has promising performance. However, even we transferred the regression task to a classification task, the VGG features still could have unsatisfactory problem in the depth estimation task, since it is trained for object recognition task on other databases. In the future, we have an idea to refine the VGG net with NYU2 data and our encoding labels, which could improve the performance of our framework. Another problem is that in the surface constraints part, we focus only on the smoothness of surface normals on the plane without considering color and edge information. In the future, we plan to add the color and edge information in the surface constraints parts.

Label encoding method: In this dissertation, the proposed label encoding method works well on both age estimation task and depth estimation task. However, the encoding labels usually distribute formally on breakpoints, which may not suitable for uniformly distributed data. In the future, we plan try to encoding labels according to the distribution of labels, this may achieve better performance.

References

[1] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *arXiv preprint arXiv:1409.4842*, 2014.

[4] D. Yi, Z. Lei, and S. Z. Li, “Age estimation by multi-scale convolutional network,” in *Computer Vision–ACCV 2014*. Springer, 2015, pp. 144–158.

[5] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learning Representations*, 2015.

[6] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He, “Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015, pp. 1119–1127.

[7] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun, “Learning convolutional feature hierarchies for visual recognition,” in *Advances in neural information processing systems*, 2010, pp. 1090–1098.

[8] R. Rautkorpi and J. Iivarinen, “A novel shape feature for image classification and retrieval,” in *Image Analysis and Recognition*. Springer, 2004, pp. 753–760.

[9] S.-C. Zhu, R. Zhang, and Z. Tu, “Integrating bottom-up/top-down for object recognition by data driven markov chain monte carlo,” in *Computer Vision and*

Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 1. IEEE, 2000, pp. 738–745.

[10] A. C. Little, B. C. Jones, and L. M. DeBruine, "The many faces of research on face perception," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 366, no. 1571, pp. 1634–1637, 2011.

[11] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *Journal of Field Robotics*, vol. 26, no. 2, pp. 120–144, 2009.

[12] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, 2009.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2012, pp. 746–760.

[14] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[15] D. A. Forsyth and J. Ponce, "A modern approach," *Computer Vision: A Modern Approach*, 2003.

[16] S. Das and N. Ahuja, "Performance analysis of stereo, vergence, and focus as depth cues for active vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 12, pp. 1213–1219, 1995.

[17] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[18] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," *arXiv preprint arXiv:1404.1777*, 2014.

[22] K. Korekado, T. Morie, O. Nomura, H. Ando, T. Nakano, M. Matsugu, and A. Iwata, "A convolutional neural network vlsi for image recognition using merged/mixed analog-digital architecture," in *Knowledge-Based Intelligent Information and Engineering Systems.* Springer, 2003, pp. 169–176.

[23] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition." in *INTERSPEECH*, 2013, pp. 3366–3370.

[24] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks–ICANN 2010.* Springer, 2010, pp. 92–101.

[25] J. Nagi, F. Ducatelle, G. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, L. M. Gambardella *et al.*, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on.* IEEE, 2011, pp. 342–347.

[26] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 111–118.

[27] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv preprint arXiv:1502.01852*, 2015.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] S. J. Nowlan and G. E. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural computation*, vol. 4, no. 4, pp. 473–493, 1992.

[31] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber, “Flexible, high performance convolutional neural networks for image classification,” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, p. 1237.

[32] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[33] J. Ba and B. Frey, “Adaptive dropout for training deep neural networks,” in *Advances in Neural Information Processing Systems*, 2013, pp. 3084–3092.

[34] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631. Citeseer, 2013, p. 1642.

[35] H. Han, C. Otto, X. Liu, and A. Jain, “Demographic estimation from face images: Human vs. machine performance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1148–1161, June 2015.

[36] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[37] Z. Yang and H. Ai, “Demographic classification with local binary patterns,” in *Advances in Biometrics*. Springer, 2007, pp. 464–473.

[38] F. Alnajar, C. Shan, T. Gevers, and J.-M. Geusebroek, “Learning-based encoding with soft assignment for age estimation under unconstrained imaging conditions,” *Image and Vision Computing*, vol. 30, no. 12, pp. 946–953, 2012.

[39] J.-D. Txia and C.-L. Huang, “Age estimation using aam and local facial features,” in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2009, pp. 885–888.

[40] F. Gao and H. Ai, “Face age classification on consumer images with gabor feature and fuzzy lda method,” in *Advances in biometrics*. Springer, 2009, pp. 132–141.

[41] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, “Image-based human age estimation by manifold learning and locally adjusted robust regression,” *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1178–1188, 2008.

[42] X. Geng, Z.-H. Zhou, Y. Zhang, G. Li, and H. Dai, “Learning from facial aging patterns for automatic age estimation,” in *ACM international conference on Multimedia*. ACM, 2006, pp. 307–316.

[43] C. Zhang and G. Guo, “Age estimation with expression changes using multiple aging subspaces,” in *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 2013, pp. 1–6.

[44] S. E. Choi, Y. J. Lee, S. J. Lee, K. R. Park, and J. Kim, “Age estimation using a hierarchical classifier based on global and local facial features,” *Pattern Recognition*, vol. 44, no. 6, pp. 1262–1281, 2011.

[45] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in *IEEE International Conference on Computer Vision*. IEEE, 2009, pp. 2146–2153.

[46] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[47] X. Wang, R. Guo, and C. Kambhamettu, “Deeply-learned feature for age estimation,” in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 534–541.

[48] Y. Sun, X. Wang, and X. Tang, “Deep convolutional network cascade for facial point detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 3476–3483.

[49] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le *et al.*, “Large scale distributed deep networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[50] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, “Support vector machine classification and validation of cancer tissue samples using microarray expression data,” *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[51] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[52] K. Ricanek and T. Tesafaye, “Morph: A longitudinal image database of normal adult age-progression,” in *International Conference on Automatic Face and Gesture Recognition.* IEEE, 2006, pp. 341–345.

[53] N. C. Ebner, M. Riediger, and U. Lindenberger, “Facesa database of facial expressions in young, middle-aged, and older women and men: Development and validation,” *Behavior research methods*, vol. 42, no. 1, pp. 351–362, 2010.

[54] Z. Zhu, P. Luo, X. Wang, and X. Tang, “Deep learning identity-preserving face space,” in *IEEE International Conference on Computer Vision.* IEEE, 2013, pp. 113–120.

[55] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments,” in *In the 12th International Symposium on Experimental Robotics (ISER.* Citeseer, 2010.

[56] D. F. Fouhey, A. Gupta, and M. Hebert, “Data-driven 3D primitives for single image understanding,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2013.

[57] A. Saxena, J. Schulte, and A. Y. Ng, “Depth estimation using monocular and stereo cues,” in *Proc. IEEE Int. Joint Conf. Artificial Intell.*, vol. 7, 2007.

[58] L. Ladicky, J. Shi, and M. Pollefeys, “Pulling things out of perspective,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2014, pp. 89–96.

[59] K. Karsch, C. Liu, and S. B. Kang, “Depth extraction from video using non-parametric sampling,” in *Proc. Eur. Conf. Comp. Vis.* Springer, 2012, pp. 775–788.

[60] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[61] C. Hane, L. Ladicky, and M. Pollefeys, “Direction matters: Depth estimation with a surface normal classifier,” in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2015, pp. 381–389.

[62] F. Liu, C. Shen, and G. Lin, “Deep convolutional neural fields for depth estimation from a single image,” *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[63] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.