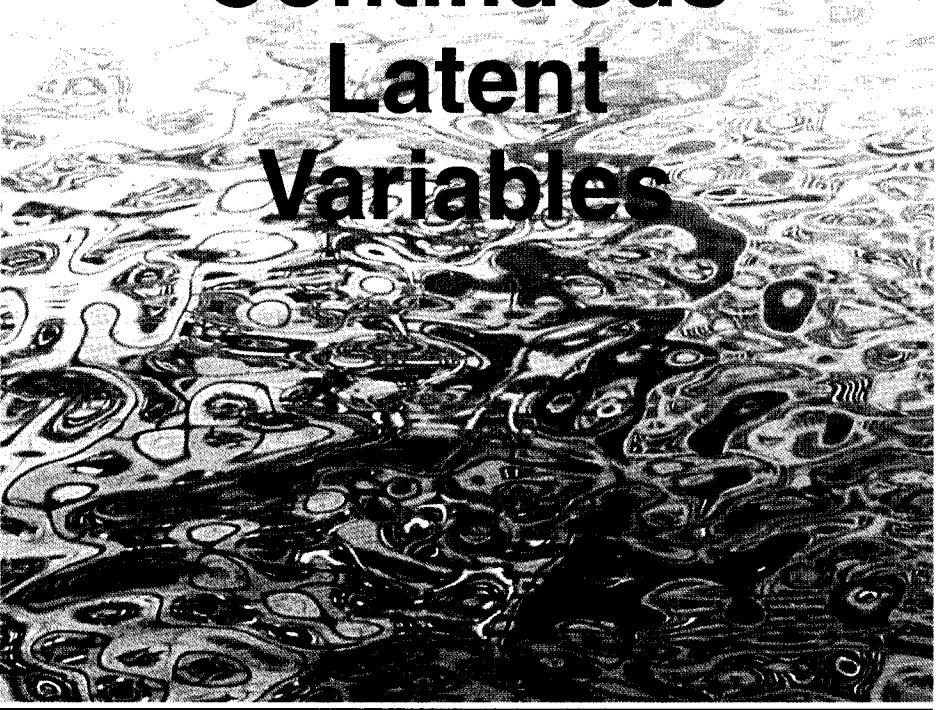

12

Continuous Latent Variables



Appendix A

In Chapter 9, we discussed probabilistic models having discrete latent variables, such as the mixture of Gaussians. We now explore models in which some, or all, of the latent variables are continuous. An important motivation for such models is that many data sets have the property that the data points all lie close to a manifold of much lower dimensionality than that of the original data space. To see why this might arise, consider an artificial data set constructed by taking one of the off-line digits, represented by a 64×64 pixel grey-level image, and embedding it in a larger image of size 100×100 by padding with pixels having the value zero (corresponding to white pixels) in which the location and orientation of the digit is varied at random, as illustrated in Figure 12.1. Each of the resulting images is represented by a point in the $100 \times 100 = 10,000$ -dimensional data space. However, across a data set of such images, there are only three *degrees of freedom* of variability, corresponding to the vertical and horizontal translations and the rotations. The data points will therefore live on a subspace of the data space whose *intrinsic dimensionality* is three. Note

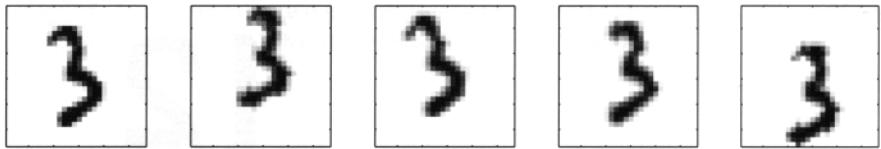


Figure 12.1 A synthetic data set obtained by taking one of the off-line digit images and creating multiple copies in each of which the digit has undergone a random displacement and rotation within some larger image field. The resulting images each have $100 \times 100 = 10,000$ pixels.

that the manifold will be nonlinear because, for instance, if we translate the digit past a particular pixel, that pixel value will go from zero (white) to one (black) and back to zero again, which is clearly a nonlinear function of the digit position. In this example, the translation and rotation parameters are latent variables because we observe only the image vectors and are not told which values of the translation or rotation variables were used to create them.

For real digit image data, there will be a further degree of freedom arising from scaling. Moreover there will be multiple additional degrees of freedom associated with more complex deformations due to the variability in an individual's writing as well as the differences in writing styles between individuals. Nevertheless, the number of such degrees of freedom will be small compared to the dimensionality of the data set.

Appendix A

Another example is provided by the oil flow data set, in which (for a given geometrical configuration of the gas, water, and oil phases) there are only two degrees of freedom of variability corresponding to the fraction of oil in the pipe and the fraction of water (the fraction of gas then being determined). Although the data space comprises 12 measurements, a data set of points will lie close to a two-dimensional manifold embedded within this space. In this case, the manifold comprises several distinct segments corresponding to different flow regimes, each such segment being a (noisy) continuous two-dimensional manifold. If our goal is data compression, or density modelling, then there can be benefits in exploiting this manifold structure.

In practice, the data points will not be confined precisely to a smooth low-dimensional manifold, and we can interpret the departures of data points from the manifold as ‘noise’. This leads naturally to a generative view of such models in which we first select a point within the manifold according to some latent variable distribution and then generate an observed data point by adding noise, drawn from some conditional distribution of the data variables given the latent variables.

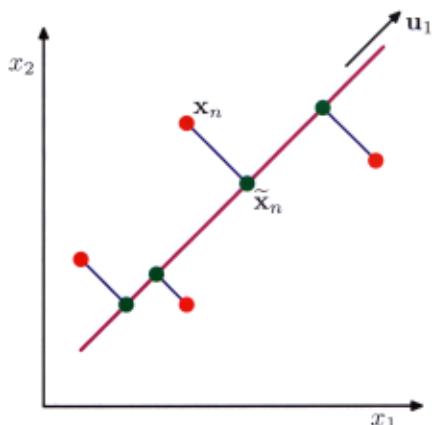
Section 8.1.4

The simplest continuous latent variable model assumes Gaussian distributions for both the latent and observed variables and makes use of a linear-Gaussian dependence of the observed variables on the state of the latent variables. This leads to a probabilistic formulation of the well-known technique of principal component analysis (PCA), as well as to a related model called factor analysis.

Section 12.1

In this chapter we will begin with a standard, nonprobabilistic treatment of PCA, and then we show how PCA arises naturally as the maximum likelihood solution to

Figure 12.2 Principal component analysis seeks a space of lower dimensionality, known as the principal subspace and denoted by the magenta line, such that the orthogonal projection of the data points (red dots) onto this subspace maximizes the variance of the projected points (green dots). An alternative definition of PCA is based on minimizing the sum-of-squares of the projection errors, indicated by the blue lines.



Section 12.2

a particular form of linear-Gaussian latent variable model. This probabilistic reformulation brings many advantages, such as the use of EM for parameter estimation, principled extensions to mixtures of PCA models, and Bayesian formulations that allow the number of principal components to be determined automatically from the data. Finally, we discuss briefly several generalizations of the latent variable concept that go beyond the linear-Gaussian assumption including non-Gaussian latent variables, which leads to the framework of *independent component analysis*, as well as models having a nonlinear relationship between latent and observed variables.

Section 12.4

12.1. Principal Component Analysis

Principal component analysis, or PCA, is a technique that is widely used for applications such as dimensionality reduction, lossy data compression, feature extraction, and data visualization (Jolliffe, 2002). It is also known as the *Karhunen-Loëve* transform.

There are two commonly used definitions of PCA that give rise to the same algorithm. PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the *principal subspace*, such that the variance of the projected data is maximized (Hotelling, 1933). Equivalently, it can be defined as the linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections (Pearson, 1901). The process of orthogonal projection is illustrated in Figure 12.2. We consider each of these definitions in turn.

12.1.1 Maximum variance formulation

Consider a data set of observations $\{\mathbf{x}_n\}$ where $n = 1, \dots, N$, and \mathbf{x}_n is a Euclidean variable with dimensionality D . Our goal is to project the data onto a space having dimensionality $M < D$ while maximizing the variance of the projected data. For the moment, we shall assume that the value of M is given. Later in this

chapter, we shall consider techniques to determine an appropriate value of M from the data.

To begin with, consider the projection onto a one-dimensional space ($M = 1$). We can define the direction of this space using a D -dimensional vector \mathbf{u}_1 , which for convenience (and without loss of generality) we shall choose to be a unit vector so that $\mathbf{u}_1^T \mathbf{u}_1 = 1$ (note that we are only interested in the direction defined by \mathbf{u}_1 , not in the magnitude of \mathbf{u}_1 itself). Each data point \mathbf{x}_n is then projected onto a scalar value $\mathbf{u}_1^T \mathbf{x}_n$. The mean of the projected data is $\mathbf{u}_1^T \bar{\mathbf{x}}$ where $\bar{\mathbf{x}}$ is the sample set mean given by

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \quad (12.1)$$

and the variance of the projected data is given by

$$\frac{1}{N} \sum_{n=1}^N \{ \mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \bar{\mathbf{x}} \}^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \quad (12.2)$$

where \mathbf{S} is the data covariance matrix defined by

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T. \quad (12.3)$$

We now maximize the projected variance $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ with respect to \mathbf{u}_1 . Clearly, this has to be a constrained maximization to prevent $\|\mathbf{u}_1\| \rightarrow \infty$. The appropriate constraint comes from the normalization condition $\mathbf{u}_1^T \mathbf{u}_1 = 1$. To enforce this constraint, we introduce a Lagrange multiplier that we shall denote by λ_1 , and then make an unconstrained maximization of

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1). \quad (12.4)$$

By setting the derivative with respect to \mathbf{u}_1 equal to zero, we see that this quantity will have a stationary point when

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \quad (12.5)$$

which says that \mathbf{u}_1 must be an eigenvector of \mathbf{S} . If we left-multiply by \mathbf{u}_1^T and make use of $\mathbf{u}_1^T \mathbf{u}_1 = 1$, we see that the variance is given by

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1 \quad (12.6)$$

and so the variance will be a maximum when we set \mathbf{u}_1 equal to the eigenvector having the largest eigenvalue λ_1 . This eigenvector is known as the first principal component.

We can define additional principal components in an incremental fashion by choosing each new direction to be that which maximizes the projected variance

amongst all possible directions orthogonal to those already considered. If we consider the general case of an M -dimensional projection space, the optimal linear projection for which the variance of the projected data is maximized is now defined by the M eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_M$ of the data covariance matrix \mathbf{S} corresponding to the M largest eigenvalues $\lambda_1, \dots, \lambda_M$. This is easily shown using proof by induction.

Exercise 12.1

To summarize, principal component analysis involves evaluating the mean $\bar{\mathbf{x}}$ and the covariance matrix \mathbf{S} of the data set and then finding the M eigenvectors of \mathbf{S} corresponding to the M largest eigenvalues. Algorithms for finding eigenvectors and eigenvalues, as well as additional theorems related to eigenvector decomposition, can be found in Golub and Van Loan (1996). Note that the computational cost of computing the full eigenvector decomposition for a matrix of size $D \times D$ is $O(D^3)$. If we plan to project our data onto the first M principal components, then we only need to find the first M eigenvalues and eigenvectors. This can be done with more efficient techniques, such as the *power method* (Golub and Van Loan, 1996), that scale like $O(MD^2)$, or alternatively we can make use of the EM algorithm.

Section 12.2.2**Appendix C****12.1.2 Minimum-error formulation**

We now discuss an alternative formulation of PCA based on projection error minimization. To do this, we introduce a complete orthonormal set of D -dimensional basis vectors $\{\mathbf{u}_i\}$ where $i = 1, \dots, D$ that satisfy

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}. \quad (12.7)$$

Because this basis is complete, each data point can be represented exactly by a linear combination of the basis vectors

$$\mathbf{x}_n = \sum_{i=1}^D \alpha_{ni} \mathbf{u}_i \quad (12.8)$$

where the coefficients α_{ni} will be different for different data points. This simply corresponds to a rotation of the coordinate system to a new system defined by the $\{\mathbf{u}_i\}$, and the original D components $\{x_{n1}, \dots, x_{nD}\}$ are replaced by an equivalent set $\{\alpha_{n1}, \dots, \alpha_{nD}\}$. Taking the inner product with \mathbf{u}_j , and making use of the orthonormality property, we obtain $\alpha_{nj} = \mathbf{x}_n^T \mathbf{u}_j$, and so without loss of generality we can write

$$\mathbf{x}_n = \sum_{i=1}^D (\mathbf{x}_n^T \mathbf{u}_i) \mathbf{u}_i. \quad (12.9)$$

Our goal, however, is to approximate this data point using a representation involving a restricted number $M < D$ of variables corresponding to a projection onto a lower-dimensional subspace. The M -dimensional linear subspace can be represented, without loss of generality, by the first M of the basis vectors, and so we approximate each data point \mathbf{x}_n by

$$\tilde{\mathbf{x}}_n = \sum_{i=1}^M z_{ni} \mathbf{u}_i + \sum_{i=M+1}^D b_i \mathbf{u}_i \quad (12.10)$$

where the $\{z_{ni}\}$ depend on the particular data point, whereas the $\{b_i\}$ are constants that are the same for all data points. We are free to choose the $\{\mathbf{u}_i\}$, the $\{z_{ni}\}$, and the $\{b_i\}$ so as to minimize the distortion introduced by the reduction in dimensionality. As our distortion measure, we shall use the squared distance between the original data point \mathbf{x}_n and its approximation $\tilde{\mathbf{x}}_n$, averaged over the data set, so that our goal is to minimize

$$J = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2. \quad (12.11)$$

Consider first of all the minimization with respect to the quantities $\{z_{ni}\}$. Substituting for $\tilde{\mathbf{x}}_n$, setting the derivative with respect to z_{nj} to zero, and making use of the orthonormality conditions, we obtain

$$z_{nj} = \mathbf{x}_n^T \mathbf{u}_j \quad (12.12)$$

where $j = 1, \dots, M$. Similarly, setting the derivative of J with respect to b_i to zero, and again making use of the orthonormality relations, gives

$$b_j = \bar{\mathbf{x}}^T \mathbf{u}_j \quad (12.13)$$

where $j = M+1, \dots, D$. If we substitute for z_{ni} and b_i , and make use of the general expansion (12.9), we obtain

$$\mathbf{x}_n - \tilde{\mathbf{x}}_n = \sum_{i=M+1}^D \{(\mathbf{x}_n - \bar{\mathbf{x}})^T \mathbf{u}_i\} \mathbf{u}_i \quad (12.14)$$

from which we see that the displacement vector from \mathbf{x}_n to $\tilde{\mathbf{x}}_n$ lies in the space orthogonal to the principal subspace, because it is a linear combination of $\{\mathbf{u}_i\}$ for $i = M+1, \dots, D$, as illustrated in Figure 12.2. This is to be expected because the projected points $\tilde{\mathbf{x}}_n$ must lie within the principal subspace, but we can move them freely within that subspace, and so the minimum error is given by the orthogonal projection.

We therefore obtain an expression for the distortion measure J as a function purely of the $\{\mathbf{u}_i\}$ in the form

$$J = \frac{1}{N} \sum_{n=1}^N \sum_{i=M+1}^D (\mathbf{x}_n^T \mathbf{u}_i - \bar{\mathbf{x}}^T \mathbf{u}_i)^2 = \sum_{i=M+1}^D \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i. \quad (12.15)$$

There remains the task of minimizing J with respect to the $\{\mathbf{u}_i\}$, which must be a constrained minimization otherwise we will obtain the vacuous result $\mathbf{u}_i = 0$. The constraints arise from the orthonormality conditions and, as we shall see, the solution will be expressed in terms of the eigenvector expansion of the covariance matrix. Before considering a formal solution, let us try to obtain some intuition about the result by considering the case of a two-dimensional data space $D = 2$ and a one-dimensional principal subspace $M = 1$. We have to choose a direction \mathbf{u}_2 so as to

minimize $J = \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2$, subject to the normalization constraint $\mathbf{u}_2^T \mathbf{u}_2 = 1$. Using a Lagrange multiplier λ_2 to enforce the constraint, we consider the minimization of

$$\tilde{J} = \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2 + \lambda_2 (1 - \mathbf{u}_2^T \mathbf{u}_2). \quad (12.16)$$

Setting the derivative with respect to \mathbf{u}_2 to zero, we obtain $\mathbf{S} \mathbf{u}_2 = \lambda_2 \mathbf{u}_2$ so that \mathbf{u}_2 is an eigenvector of \mathbf{S} with eigenvalue λ_2 . Thus any eigenvector will define a stationary point of the distortion measure. To find the value of J at the minimum, we back-substitute the solution for \mathbf{u}_2 into the distortion measure to give $J = \lambda_2$. We therefore obtain the minimum value of J by choosing \mathbf{u}_2 to be the eigenvector corresponding to the smaller of the two eigenvalues. Thus we should choose the principal subspace to be aligned with the eigenvector having the *larger* eigenvalue. This result accords with our intuition that, in order to minimize the average squared projection distance, we should choose the principal component subspace to pass through the mean of the data points and to be aligned with the directions of maximum variance. For the case when the eigenvalues are equal, any choice of principal direction will give rise to the same value of J .

Exercise 12.2

The general solution to the minimization of J for arbitrary D and arbitrary $M < D$ is obtained by choosing the $\{\mathbf{u}_i\}$ to be eigenvectors of the covariance matrix given by

$$\mathbf{S} \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (12.17)$$

where $i = 1, \dots, D$, and as usual the eigenvectors $\{\mathbf{u}_i\}$ are chosen to be orthonormal. The corresponding value of the distortion measure is then given by

$$J = \sum_{i=M+1}^D \lambda_i \quad (12.18)$$

which is simply the sum of the eigenvalues of those eigenvectors that are orthogonal to the principal subspace. We therefore obtain the minimum value of J by selecting these eigenvectors to be those having the $D - M$ smallest eigenvalues, and hence the eigenvectors defining the principal subspace are those corresponding to the M largest eigenvalues.

Although we have considered $M < D$, the PCA analysis still holds if $M = D$, in which case there is no dimensionality reduction but simply a rotation of the coordinate axes to align with principal components.

Finally, it is worth noting that there exists a closely related linear dimensionality reduction technique called *canonical correlation analysis*, or CCA (Hotelling, 1936; Bach and Jordan, 2002). Whereas PCA works with a single random variable, CCA considers two (or more) variables and tries to find a corresponding pair of linear subspaces that have high cross-correlation, so that each component within one of the subspaces is correlated with a single component from the other subspace. Its solution can be expressed in terms of a generalized eigenvector problem.

12.1.3 Applications of PCA

We can illustrate the use of PCA for data compression by considering the off-line digits data set. Because each eigenvector of the covariance matrix is a vector

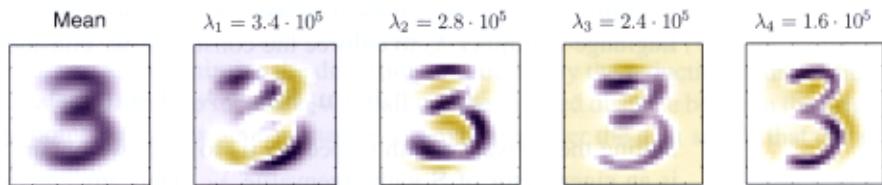


Figure 12.3 The mean vector $\bar{\mathbf{x}}$ along with the first four PCA eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_4$ for the off-line digits data set, together with the corresponding eigenvalues.

in the original D -dimensional space, we can represent the eigenvectors as images of the same size as the data points. The first five eigenvectors, along with the corresponding eigenvalues, are shown in Figure 12.3. A plot of the complete spectrum of eigenvalues, sorted into decreasing order, is shown in Figure 12.4(a). The distortion measure J associated with choosing a particular value of M is given by the sum of the eigenvalues from $M + 1$ up to D and is plotted for different values of M in Figure 12.4(b).

If we substitute (12.12) and (12.13) into (12.10), we can write the PCA approximation to a data vector \mathbf{x}_n in the form

$$\tilde{\mathbf{x}}_n = \sum_{i=1}^M (\mathbf{x}_n^T \mathbf{u}_i) \mathbf{u}_i + \sum_{i=M+1}^D (\bar{\mathbf{x}}^T \mathbf{u}_i) \mathbf{u}_i \quad (12.19)$$

$$= \bar{\mathbf{x}} + \sum_{i=1}^M (\mathbf{x}_n^T \mathbf{u}_i - \bar{\mathbf{x}}^T \mathbf{u}_i) \mathbf{u}_i \quad (12.20)$$

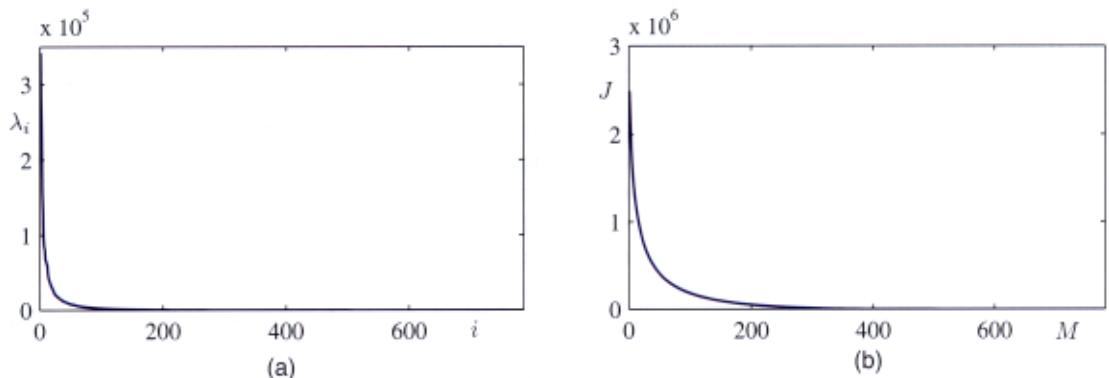


Figure 12.4 (a) Plot of the eigenvalue spectrum for the off-line digits data set. (b) Plot of the sum of the discarded eigenvalues, which represents the sum-of-squares distortion J introduced by projecting the data onto a principal component subspace of dimensionality M .

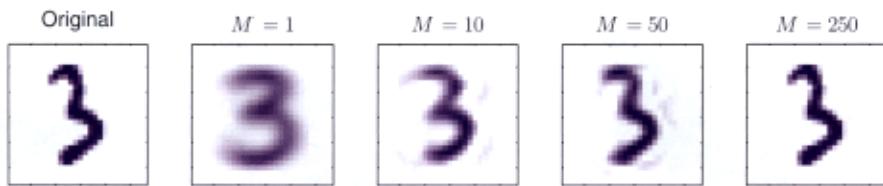


Figure 12.5 An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining M principal components for various values of M . As M increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.

where we have made use of the relation

$$\bar{\mathbf{x}} = \sum_{i=1}^D (\bar{\mathbf{x}}^T \mathbf{u}_i) \mathbf{u}_i \quad (12.21)$$

which follows from the completeness of the $\{\mathbf{u}_i\}$. This represents a compression of the data set, because for each data point we have replaced the D -dimensional vector \mathbf{x}_n with an M -dimensional vector having components $(\mathbf{x}_n^T \mathbf{u}_i - \bar{\mathbf{x}}^T \mathbf{u}_i)$. The smaller the value of M , the greater the degree of compression. Examples of PCA reconstructions of data points for the digits data set are shown in Figure 12.5.

Another application of principal component analysis is to data pre-processing. In this case, the goal is not dimensionality reduction but rather the transformation of a data set in order to standardize certain of its properties. This can be important in allowing subsequent pattern recognition algorithms to be applied successfully to the data set. Typically, it is done when the original variables are measured in various different units or have significantly different variability. For instance in the Old Faithful data set, the time between eruptions is typically an order of magnitude greater than the duration of an eruption. When we applied the K -means algorithm to this data set, we first made a separate linear re-scaling of the individual variables such that each variable had zero mean and unit variance. This is known as *standardizing* the data, and the covariance matrix for the standardized data has components

$$\rho_{ij} = \frac{1}{N} \sum_{n=1}^N \frac{(x_{ni} - \bar{x}_i)}{\sigma_i} \frac{(x_{nj} - \bar{x}_j)}{\sigma_j} \quad (12.22)$$

where σ_i is the variance of x_i . This is known as the *correlation* matrix of the original data and has the property that if two components x_i and x_j of the data are perfectly correlated, then $\rho_{ij} = 1$, and if they are uncorrelated, then $\rho_{ij} = 0$.

However, using PCA we can make a more substantial normalization of the data to give it zero mean and unit covariance, so that different variables become decorrelated. To do this, we first write the eigenvector equation (12.17) in the form

$$\mathbf{S}\mathbf{U} = \mathbf{U}\mathbf{L} \quad (12.23)$$

Appendix A

Section 9.1

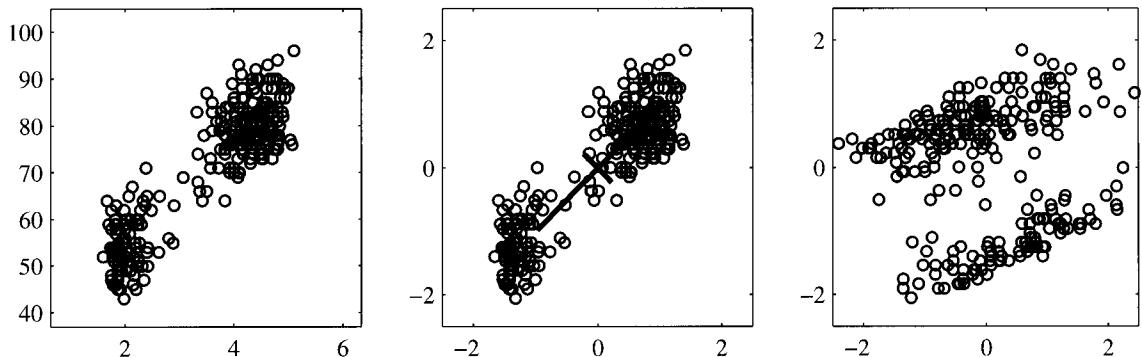


Figure 12.6 Illustration of the effects of linear pre-processing applied to the Old Faithful data set. The plot on the left shows the original data. The centre plot shows the result of standardizing the individual variables to zero mean and unit variance. Also shown are the principal axes of this normalized data set, plotted over the range $\pm \lambda_i^{1/2}$. The plot on the right shows the result of whitening of the data to give it zero mean and unit covariance.

where \mathbf{L} is a $D \times D$ diagonal matrix with elements λ_i , and \mathbf{U} is a $D \times D$ orthogonal matrix with columns given by \mathbf{u}_i . Then we define, for each data point \mathbf{x}_n , a transformed value given by

$$\mathbf{y}_n = \mathbf{L}^{-1/2} \mathbf{U}^T (\mathbf{x}_n - \bar{\mathbf{x}}) \quad (12.24)$$

where $\bar{\mathbf{x}}$ is the sample mean defined by (12.1). Clearly, the set $\{\mathbf{y}_n\}$ has zero mean, and its covariance is given by the identity matrix because

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \mathbf{y}_n \mathbf{y}_n^T &= \frac{1}{N} \sum_{n=1}^N \mathbf{L}^{-1/2} \mathbf{U}^T (\mathbf{x}_n - \bar{\mathbf{x}}) (\mathbf{x}_n - \bar{\mathbf{x}})^T \mathbf{U} \mathbf{L}^{-1/2} \\ &= \mathbf{L}^{-1/2} \mathbf{U}^T \mathbf{S} \mathbf{U} \mathbf{L}^{-1/2} = \mathbf{L}^{-1/2} \mathbf{L} \mathbf{L}^{-1/2} = \mathbf{I}. \end{aligned} \quad (12.25)$$

This operation is known as *whitening* or *sphereing* the data and is illustrated for the Old Faithful data set in Figure 12.6.

Appendix A

It is interesting to compare PCA with the Fisher linear discriminant which was discussed in Section 4.1.4. Both methods can be viewed as techniques for linear dimensionality reduction. However, PCA is unsupervised and depends only on the values \mathbf{x}_n whereas Fisher linear discriminant also uses class-label information. This difference is highlighted by the example in Figure 12.7.

Appendix A

Another common application of principal component analysis is to data visualization. Here each data point is projected onto a two-dimensional ($M = 2$) principal subspace, so that a data point \mathbf{x}_n is plotted at Cartesian coordinates given by $\mathbf{x}_n^T \mathbf{u}_1$ and $\mathbf{x}_n^T \mathbf{u}_2$, where \mathbf{u}_1 and \mathbf{u}_2 are the eigenvectors corresponding to the largest and second largest eigenvalues. An example of such a plot, for the oil flow data set, is shown in Figure 12.8.

Figure 12.7 A comparison of principal component analysis with Fisher's linear discriminant for linear dimensionality reduction. Here the data in two dimensions, belonging to two classes shown in red and blue, is to be projected onto a single dimension. PCA chooses the direction of maximum variance, shown by the magenta curve, which leads to strong class overlap, whereas the Fisher linear discriminant takes account of the class labels and leads to a projection onto the green curve giving much better class separation.

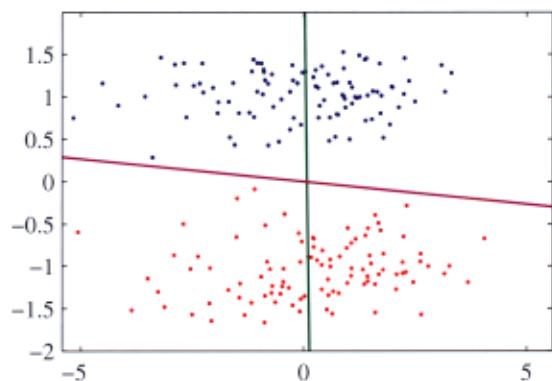
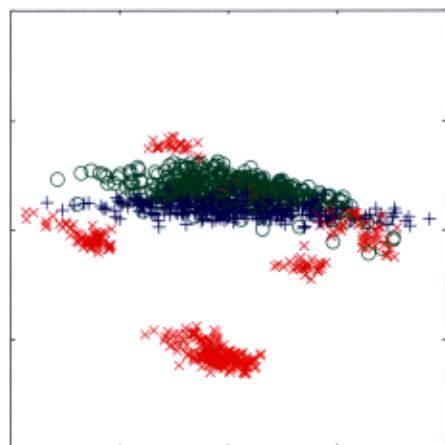


Figure 12.8 Visualization of the oil flow data set obtained by projecting the data onto the first two principal components. The red, blue, and green points correspond to the 'laminar', 'homogeneous', and 'annular' flow configurations respectively.





Article

Go-Game Image Recognition Based on Improved Pix2pix

Yanxia Zheng and Xiyuan Qian *

School of Mathematics, East China University of Science and Technology, Shanghai 200237, China;
y30211280@mail.ecust.edu.cn

* Correspondence: xyqian@ecust.edu.cn; Tel.: +86-021-64253147

Abstract: Go is a game that can be won or lost based on the number of intersections surrounded by black or white pieces. The traditional method is a manual counting method, which is time-consuming and error-prone. In addition, the generalization of the current Go-image-recognition methods is poor, and accuracy needs to be further improved. To solve these problems, a Go-game image recognition based on an improved pix2pix was proposed. Firstly, a channel-coordinate mixed-attention (CCMA) mechanism was designed by combining channel attention and coordinate attention effectively; therefore, the model could learn the target feature information. Secondly, in order to obtain the long-distance contextual information, a deep dilated-convolution (DDC) module was proposed, which densely linked the dilated convolution with different dilated rates. The experimental results showed that compared with other existing Go-image-recognition methods, such as DenseNet, VGG-16, and Yolo v5, the proposed method could effectively improve the generalization ability and accuracy of a Go-image-recognition model, and the average accuracy rate was over 99.99%.

Keywords: pix2pix; image recognition; CCMA; DDC



Citation: Zheng, Y.; Qian, X. Go-Game Image Recognition Based on Improved Pix2pix. *J. Imaging* **2023**, *9*, 273. <https://doi.org/10.3390/jimaging9120273>

Academic Editors: Antonio Fernández-Caballero and Byung-Gyu Kim

Received: 9 October 2023

Revised: 30 November 2023

Accepted: 3 December 2023

Published: 7 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Originating in China, Go is an important competitive game that is widely known. Go consists of a square board with black and white round pieces, and 19 vertical and horizontal lines on the board divide it into 361 intersections on which the black and white pieces land. During the game, two players take turns to place pieces and, at the end, the number of black and white pieces are added to the number of intersections surrounded by them, and the side with higher number is the winner. The most commonly used calculation method is the manual counting method, but the manual method is inefficient and prone to miscounting. At present, a machine-assisted method has been applied in many competitions, which has greatly improved the efficiency and accuracy, as compared to the manual method. Therefore, computer-assisted counting can be used, but the problem remains regarding how a computer can accurately identify images of a Go game, that is, correctly identify the black and white pieces and their positions.

The current Go-image-recognition methods are divided into two categories. One is the traditional image-processing methods based on OpenCV. The advantages of these methods are their simple model structures and small calculation amounts. However, the disadvantages are that most only consider the current information in the image, which is usually not robust to noise and thus unable to obtain satisfactory results in complex scenes, resulting in insufficient generalization. The other is deep-learning processing methods based on convolutional neural networks. The advantages of these methods are that the accuracy of the model increases. However, due to the need to use multiple models for checkerboard recognition and chess-piece recognition, the number of parameters in the model increases, and then the inference speed decreases.

To this end, we constructed a new Go-game-image dataset. Furthermore, we proposed a chess-image-recognition algorithm based on an improved pix2pix. In order to pay more attention to the local details of the input image, a channel-coordinate mixed-attention

(CCMA) structure was proposed. At the same time, in order to better capture the semantic information based on a long-distance context, a new deep, extended convolution module (DDC) was proposed. With these two new modules added to the pix2pix, the experiment showed that the improved model could handle a scenario with a complex background while improving the accuracy and enhancing the robustness. In addition, the model recognized the whole image, which greatly sped up the inference speed.

2. Related Work

A lot of work has been complete in order to recognize Go game images by using traditional image-processing methods. Huang [1] proposed a chess recognition algorithm based on chain coding, which used vertex chain-coding technology to parse images into coding trees and then used computer-aided notation to judge the winners and losers. However, this method required manual input. Seewald [2] proposed an optimized SVM to identify chess spectrum images from video records of Go games under constrained conditions, but the recognition rate of this method was only 72.7%. Chang [3] proposed a Go-image-segmentation algorithm based on OpenCV, but it did not perform well on a board with a complex background. Gui [4] used image-processing technology to detect, locate, and segment chess pieces in a chess game involving board information extraction of the Go robot with an accuracy rate of 93.3%, but the problem of illumination influence was not solved. On this basis, Zhao [5] used an MLP model to overcome the influence of uneven illumination, and the recognition accuracy of the Go robot was more than 90%. The above methods involved simple structures and small amounts of calculation but had the problem of insufficient generalization.

With the development of deep learning, convolutional neural networks (CNN) have been widely used for image recognition. At present, there is much research on Go-game image recognition based on CNN. Czyzewski [6] proposed a digital chessboard-configuration algorithm by combining traditional algorithms with neural networks. Based on this algorithm, he designed a chessboard-recognition and chess-piece-detection method, and the accuracy rate of the chess-piece recognition was close to 95%. Quintana [7] proposed a functional framework called LiveChess2FEN that could digitize chess-game images in real time. After positioning the board, the system used a network model, such as DenseNet, to classify all the pieces, eventually achieving 92% accuracy. Neto [8] proposed a method for identifying synthetically generated chess images in Blender using a Python API, achieving 97% accuracy in chess-piece classification by fine-tuning the VGG-16 convolutional network. Zhuo [9] proposed a Go-image-recognition method based on Yolo v5 that could resist the influence of light reflection, but the model was more complex and required higher computational performance. The above methods had the advantages of better model accuracy and robustness. However, these methods required multiple models to identify the chessboard and the chess pieces successively, which thus increased the workload of dataset annotation and the number of parameters of the model.

Since Goodfellow [10] proposed generative adversarial nets (GANs), GANs have become a hotspot in the research. Since GAN do not require much labeling of training datasets, the pre-processing work is greatly simplified. However, since GANs are too free to be controllable, Mirza [11] proposed conditional generative adversarial nets (CGAN), which constrained the generation of the models by adding a condition y . Based on this, Isola [12] proposed pix2pix, which greatly improved the accuracy of generated images by introducing paired data labels. Lundine [13] compared various models, including pix2pix, when identifying pits in various seabed environments and obtained high accuracy and recall rates. Therefore, it was reasonable to consider using pix2pix as the basic model, which has achieved high accuracy in image recognition, and pix2pix directly recognizes an entire image, reducing the annotation work on the dataset.

3. Basic Model

The generative adversarial network (GAN) was composed of two parts: the generative model (G) and the discriminative model (D). The generator used random noise as input, and its training goal was to learn the distribution of real data in order to generate pseudo-data similar to real data. The goal of the discriminator training was to accurately distinguish between real data and fake data. The original GAN had no constraints on the generator, which increased the randomness of the data generation. CGAN adds a constraint condition y on the basis of the original GAN in order to make the network generate samples in a given direction. However, both GAN and CGAN use random noise as input, and although the generated data could learn the distribution of real data, it was not sufficiently related to the input data. Pix2pix was an improvement based on CGAN where the input was no longer random noise but, instead, sample data x . The implementation principle is shown in Figure 1.

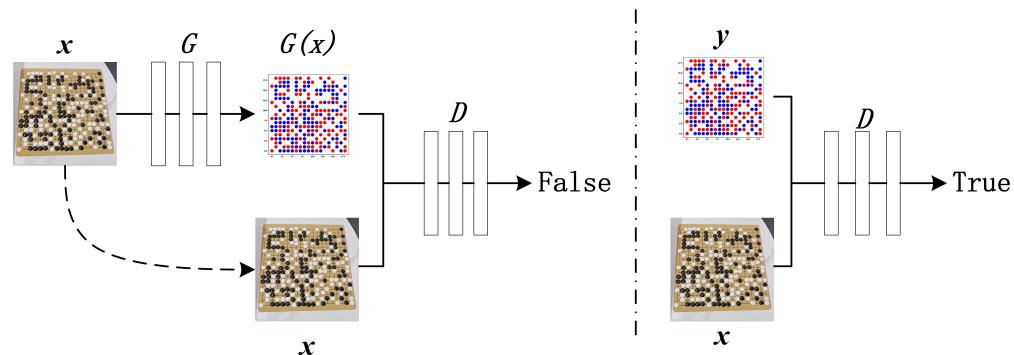


Figure 1. Pix2pix training principle.

Figure 2 shows an example of the pix2pix recognition. It was observed that for the original pix2pix, although the image $G(x)$ generated by pix2pix was close to the label y , most chess pieces could be accurately identified, but a careful observer will find that there were many cases of poor chess-piece recognition, missing detection, and incorrect color identification of the chess pieces. This shows that the original pix2pix could not meet the demand, and it needed to be optimized to improve the accuracy of the model.

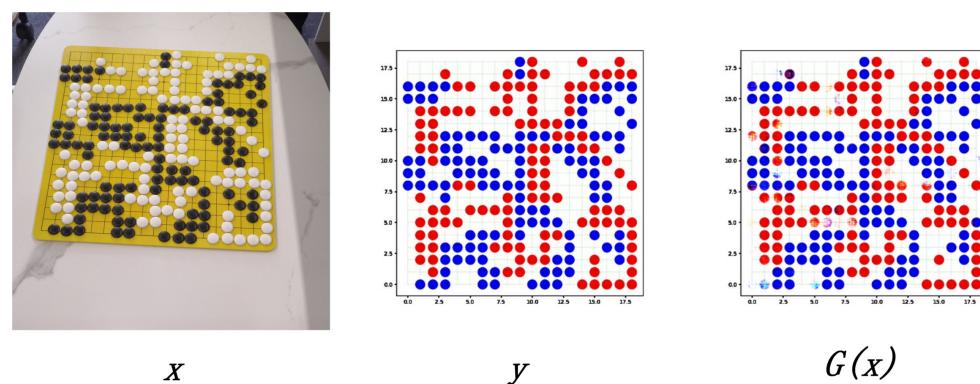


Figure 2. An example of pix2pix training.

4. Improved Pix2pix

4.1. Generator

Pix2pix used the U-net [14] as a generator. For the generator, this study retained the U-Net structure in the original pix2pix, including down-sampling, up-sampling, and skip connection. The specific structure is shown in Figure 3. Although skip joins were used to fuse semantic information at each level, a large amount of information was still lost

during down-sampling. During the task of image recognition of Go in this study, the model needed to learn more accurate pixel-level prediction. Therefore, CCMA and DDC modules were added into the down-sampling process of the U-net network. The former module enabled the model to learn the target feature information with increased attention, and the latter module enabled the model to learn the long-distance contextual information at different scales.

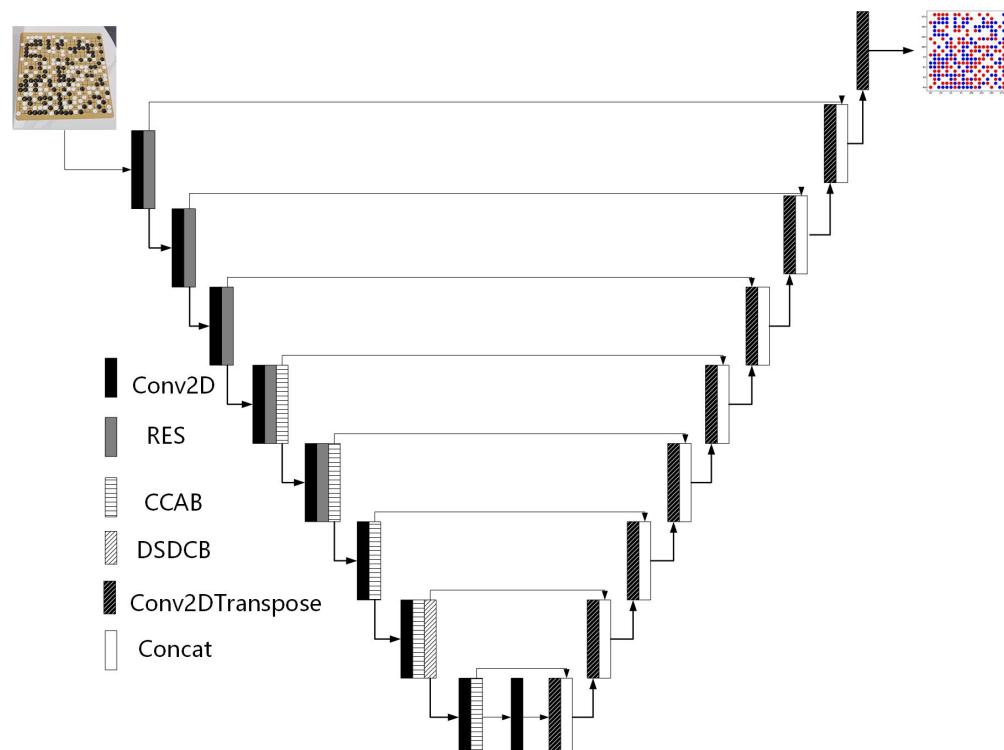


Figure 3. Generator network.

4.1.1. Channel-Coordinate Mixed Attention (CCMA)

By changing the weights of different features in the image, the attention mechanism could make the model quickly focus on the key information in the image and improve the feature-expression ability of the model. Hu [15] proposed that the SENet network adapted the characteristic response of the calibration channel by explicitly modeling the interdependence between the channels. However, the SENet network only considered the channel information inside the image, ignoring the position information. Woo [16] proposed that the convolutional block attention module (CBAM) introduced positional information through global pooling in the channels, combining channel attention with spatial attention. However, this method could only obtain local position information, and the texture details of the generated image were missing. Hou [17] put forward the coordinate attention (CA) module, which extracted the attention-feature information in the X direction and the Y direction, so as to obtain the global position information of the input feature map. However, when using this method, the position information was embedded in each channel, so the relationship between the features of each channel was weakened.

Based on the above attention methods, this study proposed the channel-coordinate mixed-attention (CCMA) mechanism. As shown in Figure 4, CCMA contained two independent sub-modules. In the first sub-module, the input feature map was firstly subjected to global maximum pooling (MAX) and global average pooling (AVG), and then the two output feature maps were added. Finally, the channel feature map (Re-feature) was obtained by weighting the previous features, one by one, through multiplication in order to complete the re-calibration of the original features in the channel dimension. In the second sub-module, the Re-feature underwent global average pooling (X-MAX) in the

X direction and Y-AVG in the Y direction, and the two output feature maps were separated after the add-operation; then, the two coordinate-attention features were generated after a 1×1 convolution-and-activation operation. Finally, the feature re-calibration in the coordinate dimension was completed by multiplication, and the channel-coordinate feature map (Re-Re-feature) was obtained. The module combined channel attention with coordinate attention effectively by extracting channel- and coordinate-attention features successively. Not only was the cross-channel feature information captured, but also the position coordinate information of the features in each channel was considered, so that the model could obtain a better feature representation.

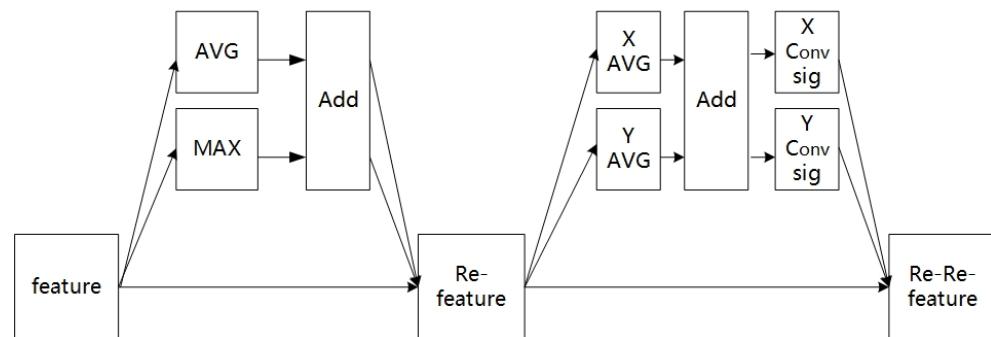


Figure 4. Channel-coordinate mixed-attention mechanism.

4.1.2. Depthwise Dilated Convolution (DDC)

The dilated convolution was originally proposed to solve the problem of image segmentation, as it could increase the receptive field while keeping the size of the feature map unchanged, so as to retain most of the information in an image. Chen [18] proposed the atrous spatial pyramid pooling in DeepLab v2 to capture the contexts of images by parallel sampling the void-convolution at different sampling rates for given inputs. Dai [19] combined the densely connected convolutional network with a dilated convolution and proposed a densely dilated convolution block to capture a wide range of scale changes in the densest possible way. The dense connection performance was good, but the parameters exponentially increased.

Based on the above methods, the depthwise dilated convolution (DDC) was proposed in this study. As shown in Figure 5, the dilated convolution was introduced to increase the model's receptive field. Although the dilated convolution increased the receptive field without losing the size of the feature graph, it also introduced new problems. Since the dilated convolution kernel was spaced when scanning the image, this meant that not all pixels in the image were involved in the calculation. Therefore, four convolution layers with different dilation rates, 1, 2, 4, and 8, were used to ensure that all inputs participated in the calculation. In order to reduce the parameters of the model, Chollet [20] proposed depthwise separable convolutions, which greatly reduced the parameters of the convolution by replacing 2D convolution with channel-by-channel and point-by-point convolutions. Inspired by this, each dilated convolution layer in this study was changed to consist of a combination of a 3×3 channel-by-channel expanded convolution and a 1×1 point-by-point convolution. This setup preserved information from denser scales with fewer parameters. Considering that point-by-point convolution lacked the spatial correlation information, the four layers adopted a dense connection method to detect convolutional features on multiple scales and realized continuous information transfer, thus learning long-distance contextual information at different scales.

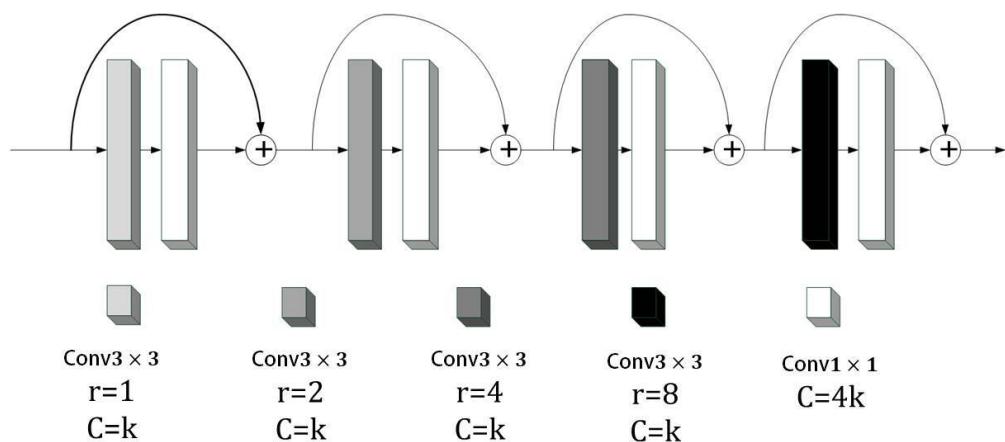


Figure 5. Depthwise dilated convolution module.

4.2. Discriminator

For the discriminator, this study made some targeted improvements while retaining the original discriminator structure, as shown in Figure 6. Firstly, spectral normalization was added. The training of GANs has been prone to pattern collapse or non-convergence, whereby the discriminator would then enter the ideal state early because it had a simpler structure than the generator and could not, therefore, provide effective gradient information to the generator. Based on this, Miyato [21] proposed the application of spectral normalization to the discriminator. Spectral normalization made the function conform to Lipschitz continuity, which could limit the drastic change in the loss function. The Lipschitz continuum was defined as follows:

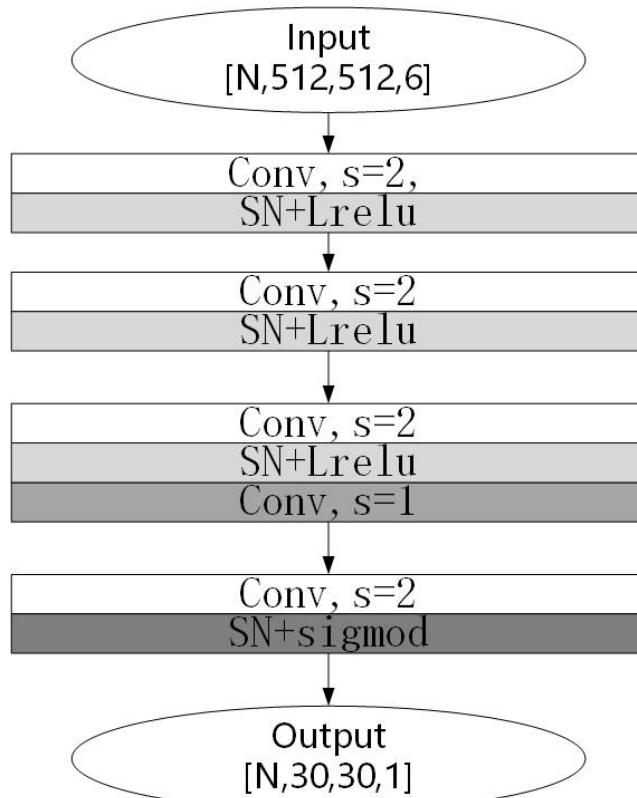


Figure 6. Discriminator network.

In GAN, suppose the discriminator $D: I \rightarrow R$, where I is the image space. If D is K -Lipchitz continuous, i.e., the maximum gradient of the function is K , then for any x and y in the image space, there is:

$$\|D(x) - D(y)\| \leq K\|x - y\| \quad (1)$$

where $\|\cdot\|$ is L2-norm, and if K is taken to the minimum value, then K is called the Lipschitz constant. In this study, spectral normalization was used to replace the normalized layer of the discriminator's convolutional kernel parameter matrix. The experiments showed that the proposed method could constrain the discriminator network in order to obtain stable training results.

Secondly, this paper used the Leaky Relu function as the activation function, which retained more image information than the Relu function.

5. Results and Analysis

In this study, the performance of the improved pix2pix was evaluated experimentally on a dataset of Go-game images. Firstly, the dataset was introduced, then the experimental environment and parameter settings of the model were provided, and finally, the evaluation index was used to measure the experimental results of the model.

5.1. Dataset

There were few publicly available datasets of Go images, so we considered constructing our own Go-game-image dataset. The steps of the dataset construction included data acquisition, data pre-processing, data annotation, and data-quality inspection. These four steps are described in detail below.

5.1.1. Data Acquisition

Data acquisition is the first step when building a dataset. We used the following devices: The camera type was a Canon EOS R5, the phone type was an Apple iPhone 13, and the tablet type was an iPad MK2N3CH/A. All the equipment was positioned directly and diagonally above the chessboard, and the linear distance from the board varied from 60 to 120 cm. The shooting conditions were in a bright indoor environment. The chessboard was made of wood, paper, and plastic, with a variety of background colors. Figure 7 shows some examples from the collected dataset, in which a total of 4500 Go images were collected.



Figure 7. This is the original image, including different shooting angles, board materials, and background colors of the board image.

5.1.2. Data Pre-Processing

The acquired images were filtered first, and the images with any occluded checkerboard area were removed. Then, in addition to the target board, there were extraneous details to remove in order to ensure that the board occupied the majority of the image. Finally, we carried out the data enhancement based on OpenCV, including a brightness adjustment and a noise addition. This was to simulate the recognition of the Go images under different lighting environments and noise levels in order to improve the robustness of the model to environmental factors. Figure 8 shows some examples of the data enhancement.

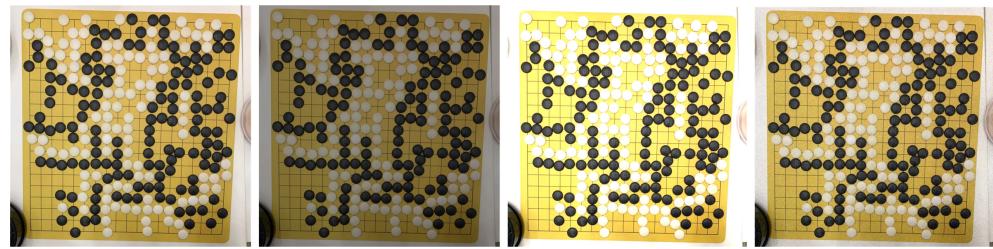


Figure 8. From left to right: original image, image with reduced brightness, image with increased brightness, and image with added noise.

5.1.3. Data Annotation

Data annotation processes the data and converts it into information that can be recognized by a computer. Here, it was divided into two steps. Step one, the pre-processed image O was annotated to obtain a 19×19 label matrix L . Step two, the matrix L was drawn into the label image P by the Matplotlib, and the specific steps including drawing the canvas and coordinate axis, drawing the background grid, drawing the corresponding circle according to the matrix L , and finally, generating the recognizable chess-labeled image P (This is shown by P in Figure 9). The corresponding relationship among the original image, the label matrix L , and the label image P is shown in Table 1. Finally, L and P were combined to form a matched Go-image and labeled figure.

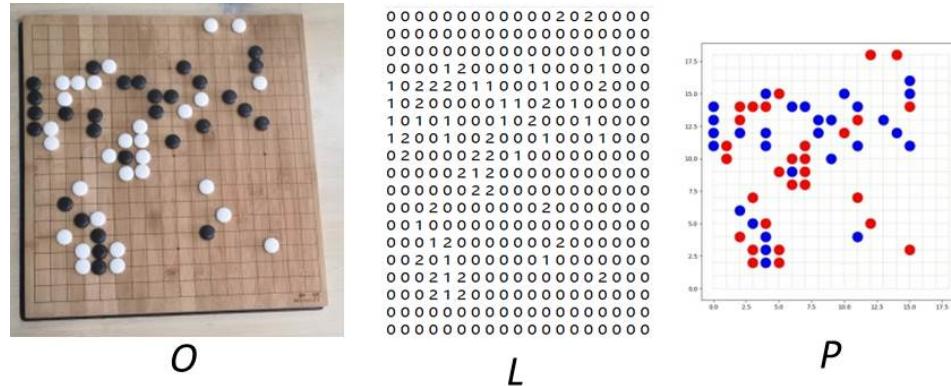


Figure 9. Image (O), label matrix (L), and label image (P).

Table 1. The mapping between the image (O), label matrix (L), and label image (P).

Image (O)	Label Matrix (L)	Label Image (P)
Black chess	1	Blue circle
White chess	2	Red circle

5.1.4. Data-Quality Inspection

After the final label graph P was obtained, its quality needed to be evaluated to ensure the accuracy of the dataset. The rule of the quality inspection was to evaluate whether O and P were paired one to one. The quality inspection process was divided into two rounds: 100% coverage in the first round and 10% coverage in the second round. The two rounds of quality inspection could reduce the workload and improve the accuracy of annotation.

After the above steps, the dataset had a good standard for both quantity and quality. Finally, the images after the quality inspection were divided as follows: 80% of the images were selected as the training set (in the Supplementary Materials), and 20% of the images were selected as the test set (in the Supplementary Materials).

5.2. Experimental Environment and Parameter Settings

The experimental environment was as follows. The operating system was a CentOS Linux release 8.5.2111, the graphics card was a Tesla T4, and the processor was a Intel(R) Xeon(R) Platinum 8163 CPU @2.50GHz. The experiment was performed in python3.8, tensorflow1.7.1-cuda11.0 environment.

The parameters were set as follows. The generator learning rate was set to 0.0002, the discriminator learning rate was set to 0.00005, the batch training amount was set to 8, the number of training times was set to 200, and the other parameter settings were left as default. The Adam optimizer was used to accelerate the training process, and the input images were randomly flipped to improve the generalization of the model.

5.3. Evaluation Metrics

5.3.1. Image Quality Evaluation Metrics

In this study, three indicators were used to evaluate the quality of the generated image, namely SSIM [22], PSNR [23], and FID [24]. SSIM is a structural similarity index measure, and it could quantify the structural similarity between the labels and the generated images; the score range was $[0, 1]$, and the larger the value, the smaller the image distortion. PSNR is the peak signal-to-noise ratio, and it was mainly used to quantify the reconstruction quality of images and videos affected by lossy compression, where the higher the PSNR value, the better the image quality. FID is the Frechet inception distance score, and it is a measure to calculate the distance between the label and the feature vector of a generated image; where the lower the score, the more similar the two sets of images are or the more similar the statistics of the two. In the best case, the score of FID would be 0.0, indicating that the two sets of images were the same.

5.3.2. Model Performance Evaluation Metrics

The ultimate goal of this study was to identify the chess pieces and their positions. These required high-quality generated images, as well as the color and the position of the red and blue circles in the images to be correct. We converted the generated images into a two-dimensional matrix I of 19×19 based on OpenCV. The specific steps were binary processing; contour retrieval; the drawing of the external rectangular frame and center point; and locating the red and blue circles. The corresponding relationship between the matrix I and the label matrix L in Table 1 were the same. For matrix I , its label L was evaluated from three aspects: accuracy, recall, and precision.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

where TP is true positive, FP is false positive, FN is false negative, and TN is true negative.

In addition, in order to pay more attention to the error rate of the chess-piece recognition, this study referred to [9] and added the following three evaluation indicators: *AERIP*, *AERI*, and *AERTD*. *AERIP* is the average error rate of erroneous intersection points, *AERI* is the average error rate of the images where errors appeared, and *AERID* is the average error rate of the images where errors could disrupt the judgment of the winner.

$$\text{AERIP} = \frac{\text{NEIP}}{N \times 361} \quad (5)$$

$$\text{AERI} = \frac{\text{NIEP}}{N} \quad (6)$$

$$AERID = \frac{NIED}{N} \quad (7)$$

where $NEIP$ is the number of erroneous intersection points, $NIEP$ is the number of images where errors appeared, $NIED$ is the number of images where errors could disrupt the judgment of the winner, and N is the total number of test images.

5.4. Ablation Experiment

In order to verify the validity of each module in the algorithm, an ablation experiment was performed, as described in this section. The visualized results are shown in Figure 10. It was observed that the Go images generated by the original pix2pix model had unclear contours and image-noise points, and it was easy to distinguish between true and false images. After adding DDC, the image outline was obviously clear, indicating that the quality of the generated image had been effectively improved after adding the DDC module. However, for the target chess pieces, there were many position errors. After adding CCMA, the position of chess pieces was basically correct, which indicated that after adding CCMA module, the model paid attention to small target information. However, the chess pieces were the wrong color. Through the combination of the DDC and CCMA modules, the enhanced algorithm in this study had significantly improved the image quality. The generated images and labels were basically identical, and it was difficult to distinguish between true and false images.

The quantization results are shown in Table 2. It was observed that compared with pix2pix, the image quality of the model after adding DDC and CCMA was significantly improved, despite the parameter increase of 1.24M (million). It was reduced to an FID score of 0.8384, increased to an SSIM score of 0.9914, and increased to a PSNR score of 44.2977. For the recognition of the target chess pieces, the Accuracy, recall and precision of the proposed algorithm reached 0.9991, 0.9997, and 0.9982, respectively.

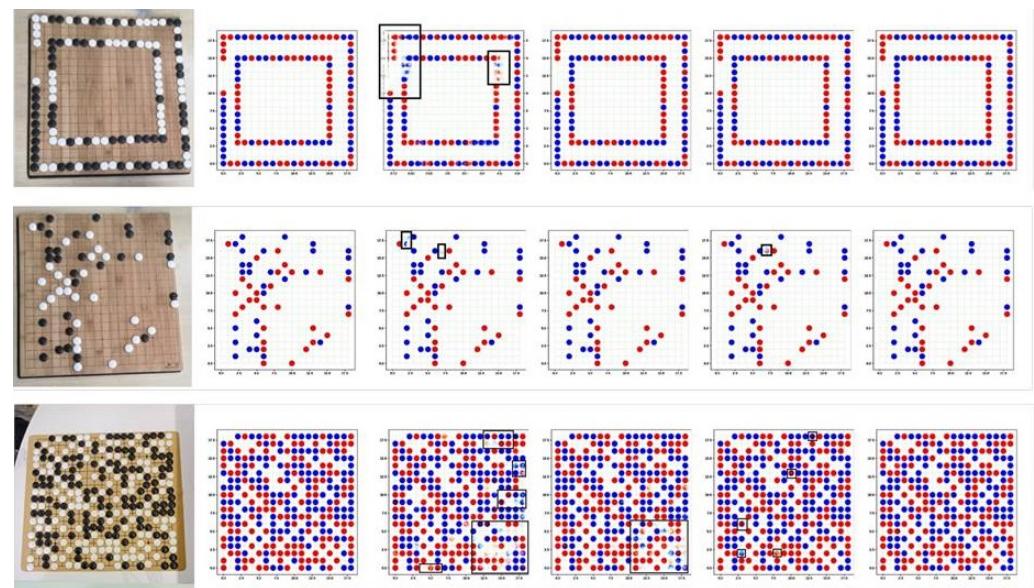


Figure 10. The results of ablation experiment on chess dataset. The columns, from left to right, include the original image, the label, Pix2pix, +DDC, +CCMA, and ours. The error results are marked by black boxes.

Table 2. The results of the ablation experiment.

Method	Params (M)	FID	SSIM	PSNR	Accuracy	Recall	Precision
pix2pix	80.92	2.1952	0.9833	41.8361	0.9942	0.9949	0.9931
+DDC	81.37	0.9738	0.9892	41.7730	0.9982	0.9995	0.9977
+CCMA	82.16	0.8484	0.9906	43.0144	0.9997	0.9982	0.9967
ours	82.61	0.8384	0.9914	44.2977	0.9991	0.9997	0.9982

5.5. Comparison Experiment

In order to evaluate the performance of the proposed algorithm, it was compared with existing chess-image-recognition methods, including DenseNet, VGG-16, and Yolo v5. See Table 3 for comparison results. It was observed that compared with other methods, the algorithm proposed in this study had a higher accuracy of 0.9982, which was 2.61%, 1.01%, and 3.10% higher than the DenseNet, VGG-16, and Yolo v5 models, respectively. In addition, the algorithm in this study significantly reduced the average error rate (AERI) of the images with errors to 0.03, which was 11%, 6.5%, and 1.5% lower than DenseNet, VGG-16, and Yolo v5 models, respectively.

In addition, we quantitatively compared the parameter sizes (parameters in millions) and the inference speed (frames per second, FPS) of these models on the test set. The recognition method based on DenseNet actually contained two models, one for identifying the chessboard and the other for identifying the chess pieces. In the process of running the model, the chessboard and then the chess pieces needed to be identified first, so the FPS was reduced by half, as compared to the single model. The same was true for VGG-16. The recognition method based on Yolo v5 actually contained three models, a chessboard-recognition model and two chess-piece-recognition models, so it had the largest number of parameters and the lowest inference speed. The model in this study directly identified the whole board and its pieces, and after obtaining the label image P , the two-dimensional matrix I could be obtained based on OpenCV, and the chess pieces could be successfully identified. Although the number of parameters was larger than DenseNet and VGG-16, the inference speed was not much different because there was only one model. For Go-game image-recognition tasks, the error rate was a very important evaluation index, which was directly related to the final victory judgment. Therefore, the lower cost (slightly lower FPS) in exchange for a lower error rate was worth it.

Table 3. The results of the comparison experiment.

Method	Accuracy	Recall	Precision	AERIP	AERI	AERID	Params (M)	FPS
DenseNet	0.9799	0.9699	0.9721	0.0037	0.1400	0.1050	67.63	68
VGG-16	0.9871	0.9852	0.9881	0.0022	0.0950	0.0700	60.05	73
Yolo v5	0.9985	0.9952	0.9672	0.0007	0.0450	0.0400	96.53	47
ours	0.9991	0.9997	0.9982	0.0004	0.0300	0.0150	82.61	66

In order to verify the effectiveness of the CCMA proposed in this study, SENet, CA, and CBAM were used in comparison experiments. The visualization results are shown in Figure 11. It was observed that the generation effect of SENet had the worst results, including missing pieces and unclear outlines. CBAM and CA were slightly better, but there were still situations where the outlines were not clear and there were noise points. Comparatively, the CCMA mechanism proposed in this study had the best generation effect, the outlines were clear, and the generated images were closest to the labeled sample.

The quantitative results are shown in Table 4. It was observed that the parameters of the CCMA model only increased by 0.08M, as compared to CA, but they decreased by 0.0542 in the FID index and increased by 0.6676 in the PSNR index. The effectiveness of the proposed CCMA mechanism was fully demonstrated.

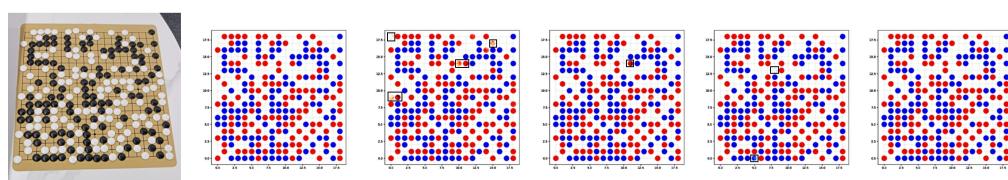


Figure 11. Results of the different attention mechanisms comparison experiment on the Go-image dataset. From left to right: original image, label, SENet, CBAM, CA, and CCMA. The error results are marked by black boxes.

Table 4. Results compared with other attention mechanisms.

Method	Params (M)	FID	SSIM	PSNR
SENet	81.42	1.4824	0.9894	39.6564
CBAM	83.54	0.9540	0.9899	41.7389
CA	82.08	0.9026	0.9902	42.3468
CCMA	82.16	0.8484	0.9906	43.0144

6. Conclusions

In this study, an improved conditional-generation, adversarial-network Go-image-recognition algorithm was proposed. Through the introduction of a CCMA module, channel attention and coordinate attention were effectively combined, so the model could learn the feature information of the target that needed attention, that is, the information of the small target chess pieces. By introducing a DDC module, the dilated convolutions with different dilated rates were densely linked, so the model could learn long-distance contextual information at different scales. The experimental results showed that the improved method was superior to the existing methods, and the average accuracy of the chess-piece recognition was more than 99.99%. The experimental results on real Go-game images showed the effectiveness and the practicability of the proposed method. The research in this study also provides a potential reference for the recognition of small objects with few features. The dataset will be further enriched to improve the algorithm model.

Supplementary Materials: The dataset of this study can be downloaded at: <https://github.com/Zhengyanxia0310/dateset.git>. The dataset contains two folders: the train set and the test set.

Author Contributions: Conceptualization, Y.Z.; Formal analysis, Y.Z. and X.Q.; Funding acquisition, X.Q.; Methodology, Y.Z. and X.Q.; Supervision, X.Q.; Writing—original draft, Y.Z.; Writing—review and editing, X.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shanghai Municipal Financial Funds for Promoting the Development of Cultural and Creative Industries, grant number 2020011278_V0.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the Supplementary Materials.

Acknowledgments: The authors would like to thank all who contributed to this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Huang, S. Research on Chess Record Recognition Algorithm Based on Chain Coding. Ph.D. Thesis, East China Normal University, Shanghai, China, 2007.
- Seewald, A.K. Automatic extraction of go game positions from images: A multi-strategical approach to constrained multi-object recognition. *Appl. Artif. Intell.* **2010**, *24*, 233–252. [CrossRef]
- Chang, S.; Song, P. Research on go image segmentation algorithm based on openCV. In Proceedings of the 10th Academic Conference on Dynamics and Control, Boston, MA, USA, 6–8 July 2016.

4. Gui, Y.; Wu, Y.; Wang, Y.; Yao, C. Visual Image Processing of Humanoid Go Game Robot Based on OPENCV. In Proceedings of the 2020 Chinese Control Furthermore, Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 3713–3716. [[CrossRef](#)]
5. Zhao, X.; Zhao, X. Go recognition method under uneven illumination based on neural network. *Softw. Eng.* **2022**, *25*, 1–4.
6. Czyzewski, M.A.; Laskowski, A.; Wasik, S. Chessboard and Chess Piece Recognition with the Support of Neural Networks. *Found. Comput. Decis. Sci.* **2020**, *45*, 257–280. [[CrossRef](#)]
7. Quintana, D.M.; del Barrio García, A.A.; Matías, M.P. LiveChess2FEN: A Framework for Classifying Chess Pieces based on CNNs. *arXiv* **2020**, arXiv:2012.06858.
8. de Sá Delgado Neto, A.; Mendes Campello, R. Chess Position Identification using Pieces Classification Based on Synthetic Images Generation and Deep Neural Network Fine-Tuning. In Proceedings of the 2019 21st Symposium on Virtual and Augmented Reality (SVR), Rio de Janeiro, Brazil, 28–31 October 2019; pp. 152–160. [[CrossRef](#)]
9. Zhuo, Y.; Fang, H.; Zhu, K.; Zhan, H.; Yuan, J. Reliable Go Game Images Recognition Under Strong Light Attack. *IEEE Access* **2021**, *9*, 160064–160071. [[CrossRef](#)]
10. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [[CrossRef](#)]
11. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
12. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976. [[CrossRef](#)]
13. Lundine, M.A.; Brothers, L.L.; Trembanis, A.C. Deep learning for pockmark detection: Implications for quantitative seafloor characterization. *Geomorphology* **2023**, *421*, 108524. [[CrossRef](#)]
14. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015.
15. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [[CrossRef](#)] [[PubMed](#)]
16. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
17. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13708–13717. [[CrossRef](#)]
18. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
19. Dai, F.; Liu, H.; Ma, Y.; Zhang, X.; Zhao, Q. Dense scale network for crowd counting. In Proceedings of the 2021 International Conference on Multimedia Retrieval, Taipei, Taiwan, 21–24 August 2021; pp. 64–72.
20. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [[CrossRef](#)]
21. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv* **2018**, arXiv:1802.05957.
22. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
23. Wang, T.; Yang, X.; Xu, K.; Chen, S.; Zhang, Q.; Lau, R.W. Spatial Attentive Single-Image Deraining With a High Quality Real Rain Dataset. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12262–12271. [[CrossRef](#)]
24. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

PAPER • OPEN ACCESS

Image deblurring: comparison and analysis

To cite this article: Zhiyi Yu 2023 *J. Phys.: Conf. Ser.* **2634** 012034

View the [article online](#) for updates and enhancements.

You may also like

- [Research on Image Deblurring Processing Technology Based on Genetic Algorithm](#)
Erhui Xi and Jiali Zhang

- [Multi-scale blind motion deblurring using local minimum](#)
Chao Wang, LiFeng Sun, ZhuoYuan Chen et al.

- [The maximum entropy on the mean method for image deblurring](#)
Gabriel Rioux, Rustum Choksi, Tim Hoheisel et al.

Image deblurring: comparison and analysis

Zhiyi Yu¹

¹ Boston University Department of Computer Science, Boston, MA, USA

steveyu@bu.edu

Abstract. Technological advancements and the advent of digital devices and media make images an important part of today's social life. Image blurring is a common challenge that results from multiple factors such as object movement, camera shake, and raindrops, among others. Image deblurring has progressively become an important field of image restoration as directed by research findings. After research for more than five decades, significant research efforts have yielded useful technologies of image deblurring. This article provides an overview of the current knowledge on image deblurring technology by focusing on the classical methods and modern trends in the field. The article reviews the conventional methods and achievements made in past studies using evidence from 34 scholarly articles. The article also examines the application of algorithms in specific deblurring methodologies adopted in recent works. It covers the recent trend of learning-based models used to restore images and their effectiveness. They include Convolutional Neural Networks, Recurrent Neural Networks and Graph Convolutional Networks. Novel deep-learning deblurring techniques are also explored. Based on the findings, issues of concerns, opportunities and direction for future research are provided to advance image deblurring technologies.

Keywords: image blurring, blind deblurring, non-blind deblurring, blur kernel, deep learning, point of spread.

1. Introduction

Deblurring image is a long-standing area of research interest [1]. It is one of the key parts of the image processing realm focused on restoring an image by recreating blurred images into decipherable ones. Digital images are likely to exhibit blur due in the form of motion blur, gaussian blur and average blur [1]. Motion blur results from relative motion between the image-capturing system and the image, and its restoration requires an estimation of the motion path. Gaussian blur involves the mathematical function applied to an image to cause a blur. Gaussian blur is mostly applied in reducing noise and details by unifying pixels [2]. Average blur results from the set of pixels equal to the value of the average pixel in a particular box neighborhood [3]. The average filter can reduce noise in an average blur. Because of the image degradations from different blur scenarios, the non-readability of blurred images often affects the algorithmic output. Image restoration is a field focused on reducing the degradation of input images. In this field, different techniques are applied to restore the degraded images into clear versions. Although there is not a sole generic model of deblurring images, this is a well-established field in literature. A plausible explanation for the absence of a single deblurring model is that deblurring can exist in different forms. Deriving an equation from the multiple forms is a challenge that is yet to manifest in the extant literature. Similar to the blur types, noise also contributes to image degradation

due to different contributing factors such as lighting, and poor calibration of image-capturing systems, among others.

To address the image-blurring challenge, several deblurring methods have emerged over the years. This paper provides an overview of the literature on image deblurring techniques as revealed by researchers. The paper summarizes 34 references on image deblurring and explores opportunities for development in the future of the image deblurring field.

2. Image deblurring methods

Two broad classifications of deblurring techniques are blind deblurring (BD) and non-blind deblurring (NBD). While BD mainly focuses on recovering the exact estimate of the blur kernel, NBD aims to restore the original image from a known blur image and give the blur estimate [4]. The methods are explored below.

2.1. Blind Image Deblurring

BD is a classical method of restoring an image where there is no information about the blurring and noise that cause image degradation. Deblurring images using this method is achieved without a known point of spread function (PSF) [5]. PSF is a “point input, represented as a single pixel in the “ideal” image, which will be reproduced as something other than a single pixel in the “real” image” [6]. The application of blind methods is broader than non-blind methods. In most cases, the PSF is known not to provide accuracy. Because NBD techniques are sensitive to incongruities between the point of spread function in BD and the actual blurring point of spread, lack of blurring PSF knowledge often results in poor deblurring outcomes.

Two techniques can be categorized in BD. The first involves the making of initial estimations in restoring the true blurred image and PSF [1, 2]. It follows an iterative process until predefined match criteria are achieved. The advantage of this approach is that it is not sensitive to noise. It follows a synchronized evaluation of the image output and PSF, leading to a more complex computational algorithm. The second approach is based on the maximum likelihood of restoration [2]. The approach involves an estimation of the indicators such as covariance and PSF matrices. Estimations of PSF use simple algorithms with minimal computational sophistication to obtain the blur and noise of the original image. A representative approach for BD is ‘deep unrolling for blind deblurring (DUBLID)’ a neural network architecture [7]. Blind deconvolution algorithm is also applied in deblurring images when there is no known information about distortion [8].

Shan et al. [8] suggested the deblurring method using a unified probabilistic equation to estimate the kernel and restoration image. The proposed algorithm included terms such as spatial noise randomness and smoothness prior that limited contrasts in the unblurred image when the blurred image exhibited low contrast. One of the assumptions made in this model is that the input kernel is largely inaccurate and thus the maximum a posteriori estimation (MAP) is applied.

Research in different deblurring methods is provided [8, 9]. Removing motion blur is a major problem of BD, and it is solved through linear and non-linear processing [10]. Non-linear processing is a classical technique that uses the Local Radon Transform to estimate bur kernel and image restoration for a clearer and sharp image [11]. BD takes into account the blur kernel in learning the deblurring of event-based images. Non-linear processing is used to recover images degraded by motion [11, 12]. The authors started with a formulation of sequential event-based motion deblurring, then demonstrated how end-to-end deep architecture can help in the optimization. The proposed nonlinear model utilized a recurrent neural network with knowledge of visual and temporal aspects of the image at the local and global scales. Experimental results showed that deep learning architecture is a novel approach to achieving a superior performance of image deblurring using real-world datasets.

BD involves weak assumptions about the filter, to address this problem, the approach focuses on image edges, a process called edge estimation [12]. The estimation of the kernel is initialized by utilizing the edge information [13]. The MAP and the Total Variation (TV) approaches provide techniques for kernel estimation. The MAP approach minimizes the problem to estimate the kernel, assuming that there

is a fixed combination of ununified segmentation masks. The TV framework is also applied thanks to its superior ability to preserve edges. The TV approach makes it possible to achieve a deblurring algorithm faster and with better quality. The two frameworks are associated with the Maximum Likelihood method of estimation. Above all, the BD methods lack the ability to estimate preliminary data about the image scene.

Literature also shows the use of loss functions such as multi-scale frequency reconstruction (MSFR) and multi-scale charbonnier (MSC) loss, and multi-scale edge (MSED) loss functions (Jiang et al. [12] and Ezumi et al. [14]. Ezumi et al. used the loss functions for blur removal due to raindrop. The authors incorporated a non-local operator (Global Context Network) to capture long-range dependencies. They also included loss functions applied in the training stage by adding models for recovering important components for high-frequency details [15]. They also included a multiscale loss to train smaller-scale processing blocks.

As this suggests the BD approaches are suitable where there is no information about SPF. The deconvolution function helps to deblur and restore PSF and image concurrently. The method requires that initial parameters are assumed to initiate deblurring through the iterative phases. An example of a motion image deblurred using the method is shown in Figures 1 below. Jiang et al. [12] proposed a deep learning model to deblur images in Figure 1 by learning to recover details from the degraded image and the motion event.



Figure 1. Comparison of Blurred Image and the generated sharp image. The results is from [11].

2.2. Non-Blind Deblurring

As opposed to blind deblurring, most non-blind methods (NBD) suggested in the literature are designed to perform better when the motion is presumed as accurately known [2]. Accordingly, NBD methods often do not perform in situations where blur estimates are noisy. The conventional NBD deblurring mode is represented by the model:

$$b = l * k + n. \quad (1)$$

where b is the blurred image and l is related to b through k (blur kernel). ‘ $*$ ’ denote the convolution operation and n is the noise (Additive White Gaussian Noise). Most NBD approaches use MAP formulation where the kernel estimates determine the latent image through optimization. As such, NBD requires prior knowledge about the PSF for deconvolution to take place. Several NBD approaches exist as provided in the literature.

3. Image deblurring methods

Wiener Filter is one of the classical NBD techniques used to clear noise problems that characterize blur image deconvolution [1]. Wiener Filter works attempt to minimize the impact of blur at frequencies with a low signal-to-noise ratio. The approach requires knowledge of PSF parameters. Image deblurring is possible where the frequency information and additive noise are certain. Without noise, the approach

becomes an inverse filter. It deconvolution noise by minimizing mean square error existing between the desired and estimated random processes [2].

Lucy-Richardson (LR) is another classical algorithm used in deblurring. Similar to Wiener Filter, the LC algorithm works in situations where PSF is identified but with no information for noise [4]. This is a cyclical process focused on restoring images using the known blurring operator. One main challenge with this approach is the many times of repeating processes. As the number of iterative cycles increases, the computational process slows down and this may increase the noise. Both Wiener Filter and LR algorithms are often ineffective, and research has proposed new formations focused on developing more accurate image priors [4, 5]. Yet, advanced models often exhibit optimization problems that are difficult to eliminate, and this limits their practical use. In recent years, research has exploited convolutional neural networks and deep learning as more promising methods of image deblurring, and they have shown great potential.

4. Recent Trends

4.1. Convolutional Neural Networks and Recurrent Neural Networks

Deblurring images using neural networks is a recent trend to address complexities associated with classical methods. Different Convolutional Neural Networks (CNNs) emerged since the advent of deep learning [16]. The CNN approaches exist in different forms, in terms of components and structure. In 2014, for example, Schuler et al. [17] suggested a deep-layered architecture for image deblurring. Cheng et al. [18] used deep CNN to formulate the image prior and interpreted an effective deblurring approach anchored on a discriminative image prior. Nah et al. [16] developed a fully CNN that proved effective in directly estimating latent clear images. In doing this, the model utilized multi-scale residual networks. Other works adopted Nah and others' model to develop multi-structures with a capacity to aggregate image features in a sharper manner [18, 19]. The CNN-based deblurring methods provide more benefits in deblurring images in scenes with low latency. In doing this, the methods circumvent the repetitive optimization process characterizing traditional methods.

Another good example is the deblurring model designed using CNNs and Recurrent Neural Networks (RNNs). The design focused on learning spatially changing depths. Using multiple hierarchical networks with varying depths, Zhang et al [20] proposed a more robust model of image deblurring. It is clear that deep learning models have a better capability of enhancing deblurring performance. In addition to single-image deblurring, multi-image or video deblurring is an emerging process in the image deblurring field. Several recent works show efforts to deblur multiple images to restore videos to better sharpness and contrast.

4.2. Generative Adversarial Networks

Another recent trend is the use of Generative Adversarial Networks (GANs), a type of machine learning where two neural networks compete, with one losing to the other [21]. GAN emerged in 2014 and has had massive success. What makes GANs interesting is that they can generate new content from a given dataset. A well-known illustration is the creation of results from training on two datasets (TFD and MNIST) (See Figure 2). The dataset in the rightmost column has true data that are closest to the adjacent generated samples.



Figure 2. Samples Results from GAN after training MNIST and TFD Data [21].

The result shows that the produced data sets show that GANs can produce outcomes based on learning as opposed to memorization.

With a given training dataset, GAN learns to produce new datasets resembling the training dataset. In image blurring, GAN can learn from images to generate new images that look almost authentic in the eyes of a human being. Although GANs started as regenerative models that would learn unsupervised, they have proved effective in learning when semi-supervised or under reinforced learning. The overall idea of GANs as conceived by Goodfellow and others is defining a context between the discriminator network and generator network [21]. The samples generated challenge the discriminator to learn the differences between them. The generator serves as a 'fool' to the discriminator by generating samples that are not practically distinguishable from the actual samples. An example of the GAN Model Architecture is provided in Figure 3.

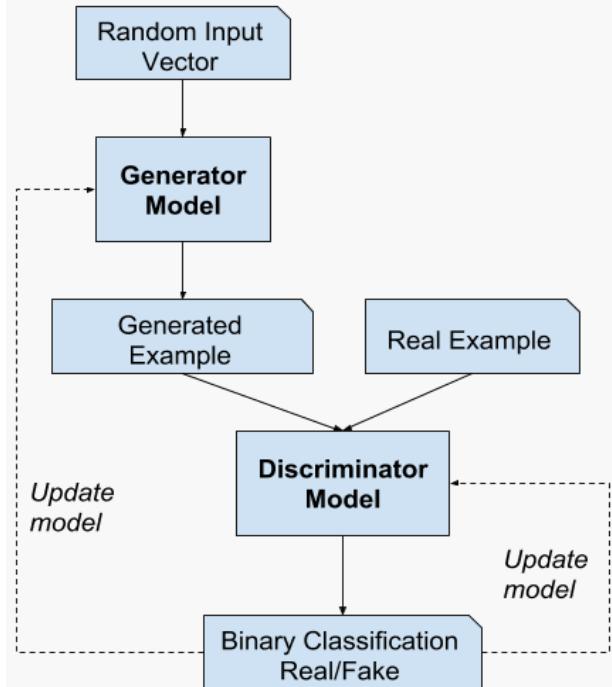


Figure 3. GAN Model Architecture [21].

4.3. Graph Convolutional Networks

Another recent development is the advent of Graph convolutional networks (GCNs) that have demonstrated the capacity to deal with complex data points such as graphs and clouds [22]. The GCNs

have attracted significant attention in machine learning and image deblurring. Evidence of GNC application in image classification exists [23]. Literature shows that GCN works better on encoded data due to the high-level semantics. Based on a comparison of network structures of data classification and image rebuilding, Xu, and Yin. [24] argued that there are semantic relationships in low-level properties such as intermediate feature maps in CNNs. An encoder-decoder network is proposed by Xu and Yin with additional graph convolutions [24]. To do this, they converted feature maps to tips of a pre-created graph to unnaturally create graph-structured data. The application of graph regularization makes the data more structured. The outcome of this experiment showed that GCNs can enhance the image deblurring process by providing even higher resolution (See Figure 4).

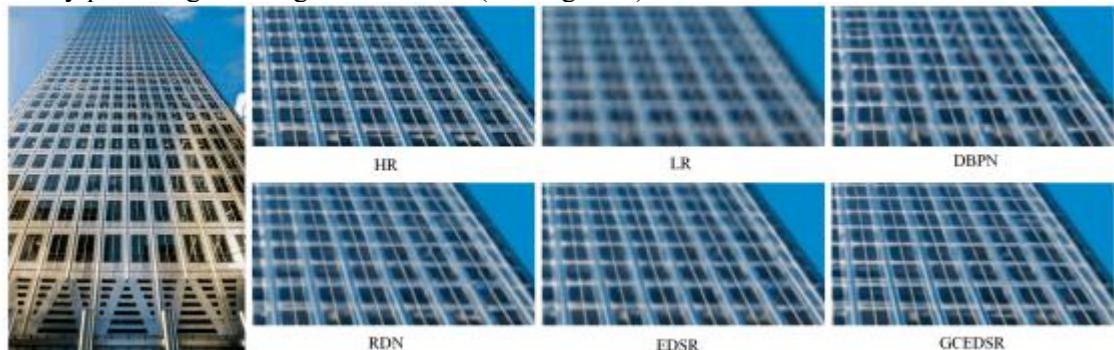


Figure 4. Visual Image Comparison [24].

A comparison of the results in Figure 4 above shows that the proposed image restoration model outperforms previous methods in terms of resolution. Yet, this is the initial work on using GCN and it is expected that future works will build on it.

4.4. Novel Deep-learning Image Deblurring

Due to the poor performance of image deblurring networks in images with larger kernels, new designs of deblurring networks have emerged. Nah's et al. [25] proposal of Deepdeblur revealed that it is not possible to achieve quality outputs as the network runs for a long time, though relatively faster than classical methods. A plausible solution suggested is the use of parameter-sharing to minimize the algorithmic size. Yet, the computational challenge still exists and [25] attempted to solve it by designing a 'coarse-to-fine' structure through GAN, a multiple-scale model. Yet, knowledge is insufficient about the working of network components in deblurring. Also, the most relevant network for image deblurring is a question of further research.

Achieving high performance and efficiency in learning-based algorithms requires a comprehensive understanding of network backbones as they influence image processing outcomes. Given the direct influence of the efficiency of image deblurring, different backbones are adopted as provided in the literature, and this serves as the basis of the different methods used in learning-based processing. Some authors suggested and adopted the use of special blocks to enhance efficiency in the encoder/decoder feature map [23, 24]. Others use parameter sharing to ensure a low number of parameters are used for simplified training and thus better performance [26]. Recent research shows that a simplified encoder/decoder structure can also generate better output minus parameter sharing [27].

Another basic determinant of the performance of learning-based networks is frameworks used in deblurring processes. The encoder-decoder structure proved to generate better performance [26]. The use of encoder and decoder in multiscale structures such as DeepDeblur will subject the input image to several encode/decode processes, achieving a finer image restoration. However, a multiscale network is likely to slow the learning process, especially where parameter sharing lacks [25]. To address this challenge, Shafiq and Gu [28] proposed a residual leaning structure (DnCNN) to improve the accuracy of avoiding degradation. Also proposed is a dense network connection (DenseNet) where each layer uses knowledge from the preceding layers (Figure 5) [29].

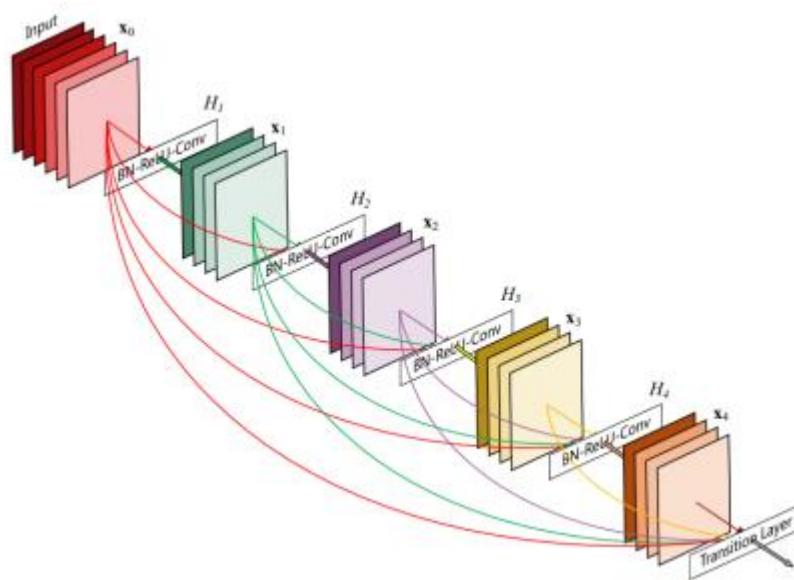


Figure 5. DenseNet Illustration [29].

The accurate segmentation of input images provides a significant impact on image accuracy, an important step in deblurring processes.

5. Summary, Challenges and Opportunities for Image Deblurring Methods

Machine learning and deep learning-based methods are the recent developments in the image restoration field. As opposed to classical methods, learning-based methods are associated with significant benefits in image deblurring. Evidence shows enhanced performance in the deblurring process. In most standard datasets, learning-based methods outpace traditional technologies by reducing the iterative cycles of deblurring processes [12]. Besides, learning provides opportunities for more realistic restoration of single and multiple images. For example, it is now more possible to restore images and videos from degradation by filling in missing points in consecutive frames. Algorithms for deep learning expectedly fit in typical computing hardware, resulting in higher computational efficiency. As opposed to the conventional deblurring models, the neural networks enhanced the efficiency of graphical processing, suggesting an advancement in the iterative cycle of image restoration.

However, some challenges exist in the current image deblurring technology. While the recent trends provide better image deblurring techniques, they come with increased computation costs that complicate processing in real-time [30]. Also, the high computational requirements come with high hardware requirements such as memory and graphical processing units. It becomes challenging to satisfy these requirements using embedded systems typically utilized in industry. This limitation suggests the need for simpler models to process image restoration. In terms of performance, existing techniques still have a downside in the processing of images with large blur kernels [16]. This suggests the need for improving algorithmic performance. The architecture of learning-based deblurring techniques is improved based on rules and experience in computer tasks such as PSF detection and classification [24]. This denotes the need for a profound understanding of the network elements and structure for image processing.

While CNNs have proved effective in image deblurring, adapting GCNs in CNNs provides new opportunities for convolutional neural networks. Although existing GCNs focus on image classification, the potential for graph structures in CNNs' feature maps is growing to enhance image deblurring. Experiments by Xu and Yin [25] showed that adapting GCN structures in CNNs can enhance the image deblurring performance, although this suggests an additional computational requirement. Indeed, examining topologic associations in the feature maps may enhance the image restoration processes.

6. Conclusion

This review shows that while image deblurring remains one of the most researched problems in the image processing realm, it is still a topic of further research. For the last two decades, some notable achievements include the application of deep neural network architectures and the adoption of graph structures in CNN feature maps to improve image output. Experiment shows that the adaptation of recent developments can improve image processing and restoration. However, several research opportunities exist to improve the quality of deblurring. First, Image deblurring for low-cost devices is a future direction. While deblurring motion images is achievable by today's deblurring technologies, algorithmic performance remains a major challenge for devices with low hardware and software capacity. Deep learning today is more effective when applied to large models, most of which cannot match the capacity of mobile devices [31]. Some researchers such as Chiang et al. [32] sought to address computational complexity by proposing a portable architecture focused on achieving better quality-latency with a deep-learning accelerator, but some limitations still emerged. They concluded that there is a need to search for a portable network architecture taking into account the limitations of mobile devices, including quantization and pruning as well as hardware preferences and limitations. The research also revealed the need for a systematic search approach to portable network architecture like Network Architecture Search for improved portability [32]. For high-quality deblurring of real-time motion images captured on mobile device cameras, model compression without affecting the performance of the algorithm is an important direction for future research. With the popular method of unsupervised learning, datasets play a critical role in learning for image deblurring. Developing methods of improving the quality of data would improve training models. Porav et al. [33] presented a system that deblurs images degraded by adherent raindrops. They used a dataset of images affected by actual raindrops. While the experiment generated quality image restoration, the role of quality dataset emerged as an important direction for future studies. Future research may design a method of creating computer-generated raindrop data that is not different from real raindrops in terms of their relevance in training models. This discovery would improve the quantitative performance of image-processing tasks. A research opportunity also exists in the development of a training model using a blurred image without depending on external training datasets. It is also possible to develop a function that can determine the extent of image blur to allow for reinforced learning. This discovery would allow deep-learning models to generate the sharpest image. Further, future research should focus on restoring images when low illumination or Gaussian blur causes degradation. It is possible to generate algorithms that can process multip

References

- [1] P. Satish, M. Srikantaswamy and N. Ramaswamy, "A Comprehensive Review of Blind Deconvolution Techniques for Image Deblurring", *Traitemen du Signal*, vol. 37, no. 3, pp. 527-539, 2020. Available: 10.18280/ts.370321.
- [2] F. Vankawala, A. Ganatra and A. Patel, "A Survey on Different Image Deblurring Techniques", *International Journal of Computer Applications*, vol. 116, no. 13, pp. 15-18, 2015. Available: 10.5120/20396-2697.
- [3] A. Husna and H. Binu, "Image Deblurring via Alternating Direction Method of Multipliers", *International Journal of Science and Research (IJSR)*, vol. 6, no. 7, pp. 522-527, 2017. Available: 10.21275/art20175307.
- [4] S. Vasu, V. Maligireddy and A. Rajagopalan, "Non-blind Deblurring: Handling Kernel Uncertainty with CNNs", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. Available: 10.1109/cvpr.2018.00345.
- [5] Z. Ye and H. Mohamadian, "Comparative and Quantitative Study of Fundamental Approaches on Digital Aerial Image Deblurring", *International Journal of Modeling and Optimization*, pp. 378-383, 2012. Available: 10.7763/ijmo.2012.v2.147.
- [6] M. Sada and M. Goyani, "Image Deblurring Techniques – A Detail Review", *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 4, no. 2, pp. 176-188, 2018.

- [7] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image Deblurring via algorithm unrolling," *IEEE Transactions on Computational Imaging*, vol. 6, pp. 666–681, 2020.
- [8] Q. Shan, J. Jia and A. Agarwala, "High-quality motion deblurring from a single image", *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1-10, 2008. Available: 10.1145/1360612.1360672.
- [9] M. Slutsky, "Noise-Adaptive Non-Blind Image Deblurring", *Sensors*, vol. 22, no. 18, p. 6923, 2022. Available: 10.3390/s22186923.
- [10] F. Xu et al., "Motion Deblurring with Real Events", *Computer Vision Foundation*, pp. 2583-2592, 2021.
- [11] T. Askari Javarani and H. Hassanpour, "Using a Blur Metric to Estimate Linear Motion Blur Parameters", *Computational and Mathematical Methods in Medicine*, vol. 2021, pp. 1-8, 2021. Available: 10.1155/2021/6048137.
- [12] Z. Jiang, Y. Zhang, D. Zou, J. Ren, J. Lv and Y. Liu, "Learning Event-Based Motion Deblurring", *Computer Vision Foundation*, pp. 3320-3329, 2020. Available: 10.1109/tpami.2004.1.
- [13] D. Perrone and P. Favaro, "A Clearer Picture of Total Variation Blind Deconvolution", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1041-1055, 2016. Available: 10.1109/tpami.2015.2477819.
- [14] S. Ezumi and M. Ikebara, "Single Image Raindrop Removal Using a Non-Local Operator and Feature Maps in the Frequency Domain", *IEEE Access*, vol. 10, pp. 91976-91983, 2022. Available: 10.1109/access.2022.3202888.
- [15] S. Cho, S. Ji, J. Hong, S. Jung and S. Ko, "Rethinking Coarse-to-Fine Approach in Single Image Deblurring", Department of Electrical Engineering, Korea University, 2021.
- [16] S. Nah et al., "NTIRE 2019 Challenge on Video Deblurring and Super-Resolution: Dataset and Study", *Computer Vision Foundation*, pp. 1-10, 2019.
- [17] C. Schuler, M. Hirsch, S. Harmeling and B. Scholkopf, "Learning to Deblur", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1439-1451, 2016. Available: 10.1109/tpami.2015.2481418.
- [18] S. Cheng, R. Liu, Y. He, X. Fan and Z. Luo, "Blind image deblurring via hybrid deep priors modeling", *Neurocomputing*, vol. 387, pp. 334-345, 2020. Available: 10.1016/j.neucom.2020.01.004.
- [19] S. Wan et al., "Deep Convolutional-Neural-Network-Based Channel Attention for Single Image Dynamic Scene Blind Deblurring", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 2994-3009, 2021. Available: 10.1109/tcsvt.2020.3035664.
- [20] H. Zhang, Y. Dai, H. Li and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5978-5986.
- [21] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning* (Adaptive Computation and Machine Learning series). Cambridge; Massachusetts; London: MIT Press, 2015.
- [22] X. Wang X, Y. Ye and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6857-6866.
- [23] A. Fabijanska, "Graph Convolutional Networks for Semi-Supervised Image Segmentation", *IEEE Access*, vol. 10, pp. 104144-104155, 2022. Available: 10.1109/access.2022.3210533.
- [24] B. Xu and H. Yin, "Graph Convolutional Networks in Feature Space for Image Deblurring and Super-resolution", *Computer Vision and Pattern Recognition*, pp. 1-8, 2021. Available: <https://arxiv.org/pdf/2105.10465.pdf>.
- [25] S. Nah, T. Kim and K. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring", *Computer Vision and Pattern Recognition*, pp. 1-21, 2016. Available: <https://arxiv.org/pdf/1612.02177.pdf>.
- [26] X. Tao, H. Gao, X. Shen, J. Wang and J. Jia, "Scale-recurrent network for deep image deblurring",

- The Chinese University of Hong Kong, 2018.
- [27] A. Gurita and I. Mocanu, "Image Segmentation Using Encoder-Decoder with Deformable Convolutions", Sensors, vol. 21, no. 5, p. 1570, 2021. Available: 10.3390/s21051570.
 - [28] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey", Applied Sciences, vol. 12, no. 18, p. 8972, 2022. Available: 10.3390/app12188972.
 - [29] G. Huang, Z. Liu, L. Van Der Maaten and K. Weinberger, "Densely Connected Convolutional Networks", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. Available: 10.1109/cvpr.2017.243.
 - [30] X. Yang, Y. Chen, R. Tao, Y. Zhang, Z. Liu and Y. Shi, "Endoscopic Image Deblurring and Super-Resolution Reconstruction Based on Deep Learning", 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), 2020. Available: 10.1109/icaice51518.2020.00039.
 - [31] Q. Wang, J. Tan, T. Xing, F. Chen and J. Niu, "SID: Sensor-Assisted Image Deblurring System for Mobile Devices", IEEE Access, vol. 7, pp. 146607-146619, 2019. Available: 10.1109/access.2019.2937137.
 - [32] C. Chiang, Y. Tseng, Y. Xu, H. Kuo and Y. Tsai, "Deploying Image Deblurring across Mobile Devices: A Perspective of Quality and Latency", IEEE Explore, pp. 1-11, 2020.
 - [33] H. Porav, T. Bruls and P. Newman, "I Can See Clearly Now: Image Restoration via De-Raining", 2019 International Conference on Robotics and Automation (ICRA), 2019. Available: 10.1109/icra.2019.8793486.