Degree Thesis for Master of Science

A Study on Frame Prediction Method based on Operation Probability Map

Supervisor : Prof. Dong-Hoan Seo

February 2018

Department of Electrical and Electronics Engineering

The Graduate School of Korea Maritime and Ocean University

Soo-Hwan Lee

Thesis submitted by Soo-Hwan Lee in Fulfillment of Requirement for the Degree of Master of Science

Committee Chairman : D. Eng. Yoon-Sik Kim ㊞

Committee Member : D. Eng. Dong-Hoan Seo ㊞

Committee Member : D. Eng. Jong-Gun Lim ㊞

February 2018

Department of Electrical and Electronics Engineering

The Graduate School of Korea Maritime and Ocean University

Contents

Contents of Figure and Table

<Figure Contents>

<Table Contents>

동작 확률 지도 기반
프레임 예측 기법에 대한 연구

by Soo-Hwan Lee

Department of Electrical & Electronics Engineering
Graduate School of Korea Maritime and Ocean University
Busan, Republic of Korea

Abstract

동영상내에서 손상에 의해 소실된 프레임을 복원하거나 연속적인 새로운 프레임을 생성하는 기법인 프레임 예측은 객체들의 동작 예측이 필요한 자율주행, 보안 등의 미래 주요 기술로서 주목받고 있다. 최근 이 기술은 딥러닝 기술과 결합하여 예측 정확도가 많이 향상되고 있으나 많은 학습데이터와 연산량이 수반되기 때문에 실질적인 적용에는 어려움이 존재한다. 기존의 딥러닝 기반 예측 모델은 새로운 프레임 생성 과정에서 예측에 의해 생성된 프레임을 피드백하기 때문에 누적오차가 많이 발생하여 시간이 지남에 따라 예측 정확도가 감소한다. 따라서 본 논문에서는 convolution neural network (CNN)와 long short-term memory (LSTM)으로 구성된 네트워크를 통해 프레임들의 동작 특징들을 추출하고 패턴을 학습하여 동작 확률 지도를 생성하여 움직임이 발생한 영역에 대하여 deconvolution neural network(DNN)를 통해 이후 프레임을 생성하는 새로운 프레임 예측 모델을 제안한다. 제안한 모델은 CNN과 LSTM을 통해 프레임들의 동작 특징들을 추출하고 패턴을 학습하여 동작 확률 지도를 생성한다. 이를 통해 임의의 한 프레임에서 동작이 발생하는 영역를 판별하고 이 영역만 DNN을 통해 새로운 프레임을 획득한다. 이때 학습 난이도가 높은 DNN의 효율적인 학습을 위해 generative adversarial nets(GAN) 기법을 적용한다. 제안된 새로운 모델의 학습과 검증을 위하여 무작위로 일부 프레임이 제거된 로봇 움직임 영상을 기반으로 생성된 영상과 원본 영상을 PSNR로 비교 분석하였다. 그 결과, 제안한 프레임 예측 모델의 PSNR은 35.16으로 비교한 3개의 다른 모델에 비해 최대

14.06이 향상되었다. 또한 생성된 프레임에 따른 PSNR의 감소도 4번째 프레임 이전에는 2, 이후에는 7로 평균 5가 개선되었다.

A Study on Frame Prediction Method based on Operation Probability Map

by Soo-Hwan Lee

Department of Electrical & Electronics Engineering
Graduate School of Korea Maritime and Ocean University
Busan, Republic of Korea

Abstract

Frame prediction, which is a technique to reconstruct frames lost due to damage or to generate new consecutive frames in the video, is attracting attention as a main technology which is indispensable for the autonomous vehicle and the artificial intelligence based security system that require motion prediction of objects. Recently, this technology has improved prediction accuracy in combination with deep learning technology, but it is difficulties in practical application because it involves a lot of learning data and computation amount. The existing deep learning based prediction model, since the frame generated by the prediction is feedback in the new frame generation process, is decreased the prediction accuracy over time. Therefore, in this paper, we propose an operation probability map based new frame prediction model using convolution neural network (CNN), long short-term, memory (LSTM) and deconvolution neural network(DNN) to minimize unnecessary computation regions in the frame and prediction error. The proposed model extracts the operating characteristics of the frames through CNN and LSTM and learns the patterns to generate the operation probability map. Through this process, a region in which an operation occurs is determined in one frame, and a new frame is obtained through DNN only in this region. At this time, the generative adversarial nets(GAN) technique is applied for efficient learning of DNN with the high learning complexity. For the learning and verification of the proposed new model, we compared and analyzed the generated frame and the original frame based on robotic motion images with some frames removed randomly using PSNR. As a result, the PSNR of the proposed frame prediction model is 35.16, which is 14.06 higher than the other three models. Also, the decrease of the PSNR according to the generated frame is decreased to 2 before the 4th frame and then to 7 thereafter, and is improved by 5 on the average.

Chapter 1 Introduction

The problem of understanding the sequence of situations and reconstructing the situation by reasoning the intermediate process is relatively easy for humans. But, in the case of the machine, because it understands and inferences the situation by learning the video without auxiliary techniques such as image segmentation and abstraction, It is difficult to expect a human-like level of results for changes in object size, color, shape, and angle. Therefore, in order to apply a completely unmanned system such as autonomous navigation and unmanned surveillance in an actual environment where many variations of similar objects occur, an algorithm that solves the problem of lowering the accuracy of the image understanding is required. So several studies have been actively conducted in the field of computer vision both at domestic and overseas. Recently, Deep Learning, which is a learning model capable of deep abstraction, has been improved to solve this problem. However, Deep Learning model based on an artificial neural network requires a large amount of computation, so it is limited to be applied in real time. Therefore, we need an approach to reduce the computational complexity by learning the features of the image selectively, rather than the heuristic approach of the existing learning model.

The frame prediction is very similar to the frame interpolation method used in the conventional animation or image restoration, by extracting patterns in a video and estimating a subsequent frame of a target frame. However, unlike the frame interpolation method, which is based on the comparison between the previous frame and the post frame, the frame prediction is more difficult than the frame interpolation because it is estimated based on the features of the previous frames without the information of the after frame. The frame interpolation technique has been used in various image industries that require post correction to increase the number of frames per second of the original image by generating an intermediate frame. But, since real-time processing is important to be applied to unmanned systems such as autonomous navigation and unmanned surveillance, there is a need for a frame prediction method that estimates the posterior situation without posterior information.

The technique of predicting a frame of a video is a field in which the recent research in the field of computer vision has begun. Because of the similarity with the existing frame interpolation algorithms, the same technique applied to the existing algorithm is used. The fundamental principle of frame prediction is a method of estimating the position of pixels in a post frame by learning or modeling a movement pattern of pixels by accumulating the displacement difference of the same pixel in frames in a moving image. For this frame prediction, the movement of pixels of the whole moving picture is extracted by a descriptor called motion vector, and the tendency of the whole moving picture is estimated by posterior frame estimation based on various learning algorithms or probability models. In recent years, there have been many researches such as the convolution neural network (CNN), which extracts the spatial relation through the connection between neighboring pixels, and the recurrent neural network (RNN), which learns the temporal relation through the connection of the next node recursively. Algorithms for predicting posterior frames based on previous frames are being studied through a model that is constructed by properly combining neural network layers.

The existing research is to extract the optical flow, which is an optical change from the frames in the image, by the descriptor and generate the intermediate frame through the flow of the target frame before and after[1, 2]. However, since the movement of the object is represented by using the characteristics of light as the descriptor, it is difficult to extract the descriptor in the image with severe distortion due to the light, There is a limitation in that the accuracy of the prediction is inferior in case of a complicated operation. Another study uses a motion vector that expresses the variation pattern of similar pixels between frames, unlike optical flow, which uses light characteristics as a descriptor[3, 4, 5]. The technique using motion vector descriptor extracts the change of frames in the video as a descriptor and models the change tendency of the video to estimate the frame. In this way, it is possible to predict more precise and general features by abstracting the various and complex change characteristics in the image by expressing the change characteristics of the pixels as descriptors or constructing them as models. However, since the prediction method based on the motion vector uses the pixel variation of the moving picture as the descriptor, the predictable motion is limited and the prediction accuracy is weak in the unusual image where many similar objects occur.

Recently, deep learning techniques have shown good performance in many areas of computer vision based on high abstraction ability. Deep learning is a neural network, which enables higher abstraction than existing neural network algorithms, and can extract various features of information compared to other models[6, 7]. Long et al[8]. applied deep learning to input the previous and subsequent frames into the deep learning model for image matching and interpolate the frames. However, since this study uses the existing deep learning model, there is a problem that the interpolated frame is blurred due to the limit of the generator. Niklaus et al[9]. combine frames created by passing previous and subsequent frames and a convolution neural network model to generate intermediate frames. Also, a technique for predicting images through various deep-running models has been studied recently [10, 11].

We apply the method used in existing interpolation algorithms and try to solve the limitations of existing deep learning based algorithms. In this paper, we propose a model that learns the features of motion through the change of motion in frames and predicts the area with the high probability of operation based on the previous frame of the target frame to be generated and generates the motion in part afterward. The model also proposes a frame prediction algorithm that combines the previous frame and the generated region to produce a posterior frame. Convolution neural network (CNN) is constructed to extract the characteristics of motion according to the frame progression of video and learns the tendency of changing extracted features into a recurrent neural network (RNN). This model predicts the probability of occurrence of an operation, generates an operation probability map and finds a portion likely to be operated. Since the area with a low probability of operation will not change, we designed an algorithm that suppresses generation and combines with the previous frame to suppress noise and reduce the amount of computation. To learn the proposed model, ten frames are generated by predicting the frame by feeding back the generated frame with a natural image of 30 fps or more, and comparing with the original frame, the peak signal to noise ratio.

Chapter 2　Related Works

2.1 Convolutional neural network

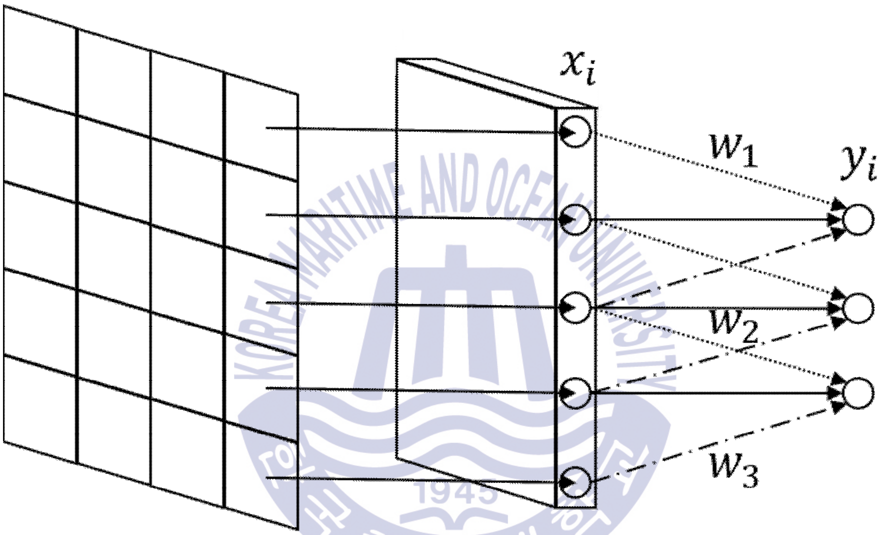


Fig. 2.1　The concept of basic convolution neural network

In general, deep learning, which is an artificial neural network stacking technique, consists of a fully connected network in which both input nodes and output nodes are connected to each other. This layer can perform various operations through weighted learning, but it is very vulnerable to a recognition of local features because it has high computational complexity and learns global features. Therefore, in the field of computer vision aiming at the image, most of them use convolution neural network (CNN). But in real image problems, it is necessary to understand objects based on local characteristics. Therefore, CNN improved the general artificial neural network. Fig. 2.1 shows the connection structure of the single-layer CNN based on the input image. In Fig. 2.1, x_i is input information, y_i is output information, and w_1, w_2, and w_3 are weights of each line. Fig. 2.1 summarizes the results as shown in:

$$y_i = \sum_{m=i-1}^{i+1} \sum_{k=1}^{n} w_k * x_m. \tag{2.1}$$

Equation (2.1) is assumed to be one-dimensional information, unlike general video situation. In the case of an image, the result of y axis is added to equation (2.1). As shown in Fig. 2.1 and equation (2.1), CNN learns only based on surrounding pixel information. Also, CNN can be applied to other areas of w_1, w_2, and w_3 with the same weight, thereby reducing computational complexity. However, since a single weight map can extract only one feature, a weight group that plays a role of various filters is required for classification of various features. A weight map corresponding to one filter is called a kernel, and the number of kernels is equal to the number of filters. Therefore, CNN constructs various kernels according to each extracted feature to extract various features[12].

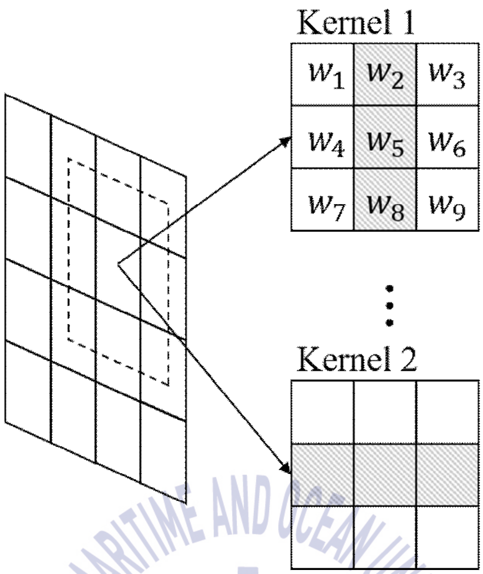


Fig. 2.2 The kernel of basic convolution neural network

Fig. 2.2 shows the configuration in the CNN of these filters. w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, and w_9 in Fig. 2.2 represent the weights of kernel 1. In Fig. 2.2, the output is determined through the convolution of the weights of the left input image and the normal CNN mask size. If kernel 1 and 2 are assumed to weight 1 and the rest of 0, kernel 1 is a feature of the vertical orientation of the image, and kernel 2 is a filter that extracts features of the horizontal orientation of the image. By increasing these various kernels, it is possible to extract small features from all over the world through various filters that extract the same feature in all regions.

2.2 Long Short-Term Memory

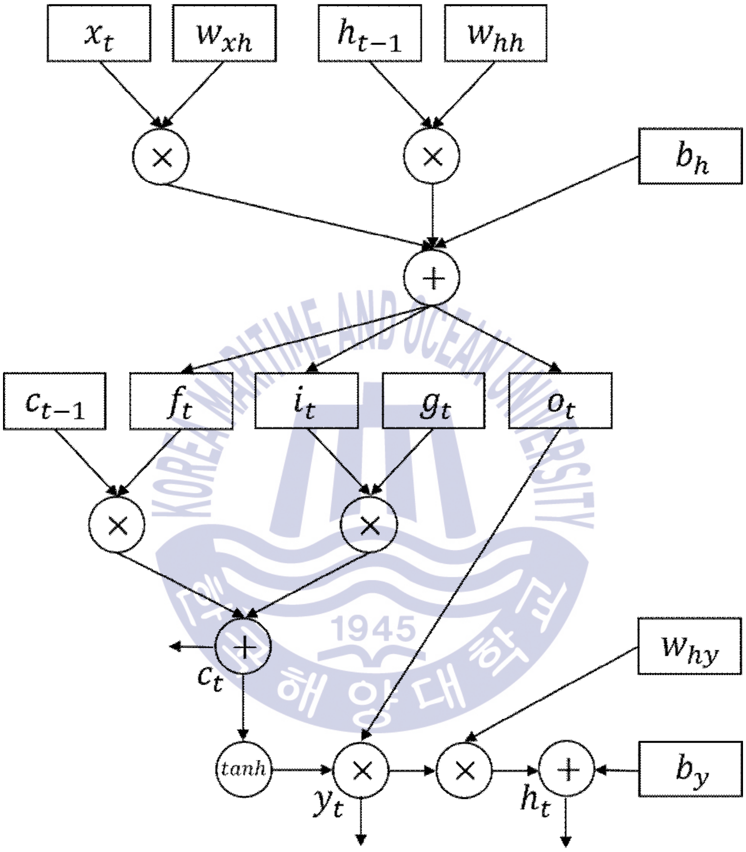


Fig. 2.3 The architecture of Long Short-Term Memory

The videos are arranged in chronological order in each frame. Therefore, a special type of neural network is needed to characterize sequential signals. It is a recurrent neural network that is designed to extract the characteristics of sequential signals by recursively repeating the input of the node's output to the next node. As the RNN progresses in time, the gradient vanishes due to backpropagation for learning, which makes learning difficult. RNN has been designed to add LSTM by adding five gates to solve this problem of increasing learning difficulty. Fig. 2.3 represents the monolayer structure of LSTM and represents the weight and gate of t moment. Where x is input information, h is intermediate output information, y is final output information, w is weight, f, i, o and c are the four gates of LSTM. f is a forget gate, i is an input gate, o is an output gate, and c is a cell gate. The parameters of each gate are given by:

$$\begin{aligned} f_t &= \sigma(w_{xh.f}x_t + w_{hh.f}h_{t-1} + b_{h.f}) \\ i_t &= \sigma(w_{xh.i}x_t + w_{hh.i}h_{t-1} + b_{h.i}) \\ o_t &= \sigma(w_{xh.o}x_t + w_{hh.o}h_{t-1} + b_{h.o}) \\ g_t &= \tanh(w_{xh.g}x_t + w_{hh.g}h_{t-1} + b_{h.g}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t), \end{aligned} \tag{2.2}$$

where $\odot$ is the Hadamard product, which means multiplication by element. b is the coefficient that adds the bias of the function. These gates, which are added to the basic RNN, are designed to manage the memory, such as the maintenance and deletion of stored information. Typically, forget gate f_t is a gate for deleting previous information, which receives h_{t-1} and x_t and is derived from an active function. In general, since the sigmoid function is used as an activation function, it has a value between 0 and 1, so 0 is deleted, and 1 is memorized. The input gate i_t is a gate that determines what information is stored in the cell. These gateways determine and handle the deletion and maintenance of information in the long term[13].

The primary LSTM input node and output node are designed to receive only one input of the preceding node. However, it is possible to design the LSTM layer with various configurations by modifying the arrangement of the LSTM input node and the output node. The LSTM-CNN layer can be constructed by transforming the input node of the LSTM into the input form of the CNN layer so that the LSTM can extract the characteristics of the image. In this paper, we use the LSTM-CNN layer that combines both the CNN spatial feature extraction function and the LSTM temporal feature extracting function[14,15].

2.3 Generative adversarial network

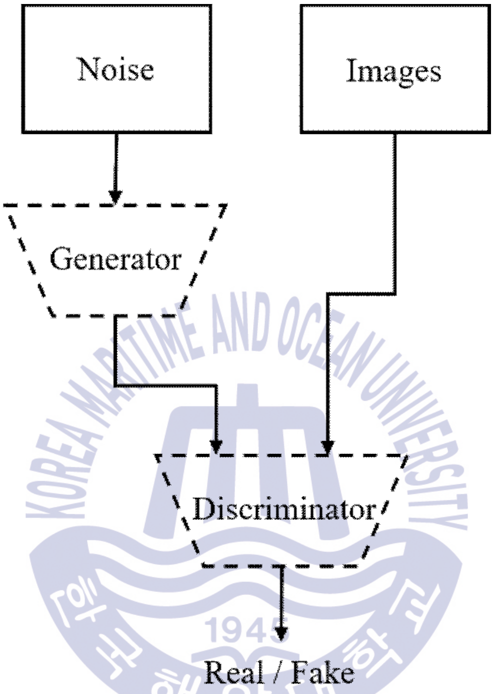


Fig. 2.4 The architecture of Generative adversarial network

In the fields of frame prediction and image restoration, a new image is generated in an initial image mixed with noise, unlike a general image field which modulates an image. Therefore, unlike the existing computer vision algorithms, this generation model is tough due to limitations of the map learning model. Recently, the generative adversarial network has been proposed, and many improvements have been made to the learning method of this generation model. Fig. 2.4 shows the architecture of the GAN. GAN is a method of learning by constructing two networks of mutually opposing pairs, and these two opposing pairs are composed of a generator and a discriminator. The generator reproduces the original target image based on the image including the noise, and the discriminator determines whether the original image is compared with the image reproduced by the generator and the original image. Generators that are difficult to learn are relatively easy to complete by learning to generate images that can not be distinguished by the discriminator through competition between the discriminator and itself. The $Loss$ function on GAN is expressed as:

$$Loss = \min\max Loss(G, D) = X[\log D(x)] + Z[\log(1 - D(G(z))], \tag{2.3}$$

where X is the original image to be input, Z is the noise input to the generator, G is the $Loss$ function of the generator, and D is the $Loss$ function of the discriminator. Through the equation (2.3), the generator G tries to reduce the probability of the success of the discrimination, and the discriminator D tries to increase the probability of success of the discrimination so that the H function to be learned can be explained by the min-max game of the generator and the discrimination period. However, since the simultaneous learning of the generator and the discriminator is impossible, it is possible to express as:

$$\begin{aligned} L_0 &= \max_D L(G,D) = X[\log D(x)] + Z[\log(1 - D(G(z))] \\ L_1 &= \max_G L(G) = Z[\log D(G(z))] \\ L_2 &= \min_G L(G) = Z[\log(1 - D(G(z)))], \end{aligned} \tag{2.4}$$

where L_0 is a $Loss$ function for learning discriminator D, and L_1 and L_2 are $Loss$ functions for learning generator G. In order to learn GAN effectively, we first learn L_0, which is relatively easy to learn, several times, then learn L_1 of the generator to learn the generator that is difficult to learn at first slowly to the discriminator. Then, if the learning is stabilized, it learns through L_2 and converges quickly. Through this process, it is possible to efficiently learn the generator which is difficult to learn as compared with other models[16,17].

Chapter 3 The proposed frame prediction model

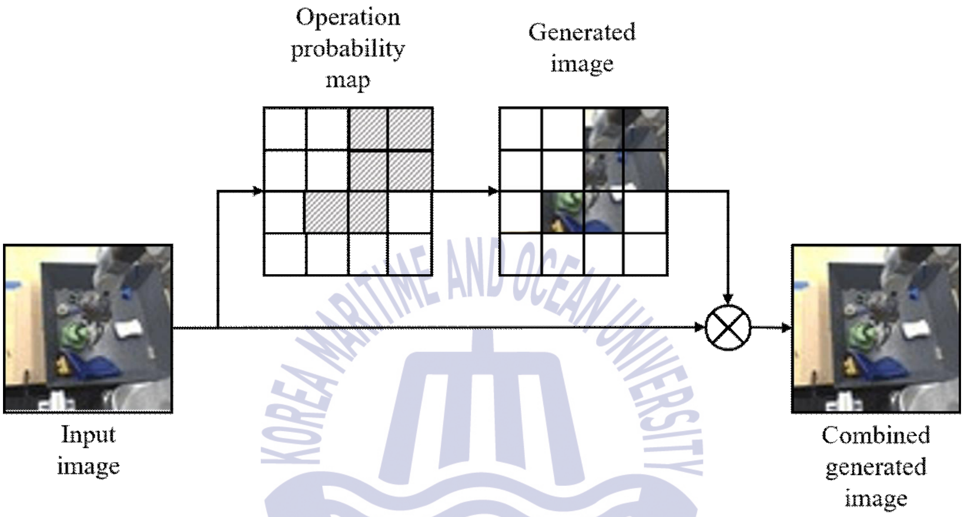


Fig. 3.1 The concept of proposed model

In this paper, we propose an operation probability map based frame prediction model that generates only highly probable regions by estimating the operation probability using a deep neural network. Fig. 3.1 is a graphical representation of the idea of the proposed model. The operation probability map based frame prediction model is a model that generates the operation probability map at the top by estimating the operation possibility of the object in the input image shown in Fig. 3.1 and generates only areas with a high probability of being hatched based on the operation probability map. In the general video, it can be seen that the background area except for the moving object hardly changes. Therefore, only the region of the moving object is generated, and the image of the background region which is not generated is taken from the image of the input image and combined to complete the predicted image. Reducing redundant operations on unchanging backgrounds further reduces learning and creation time. In the following section, we describe the structure of the proposed model and the composition and arrangement of the individual layers.

3.1 Structure of the proposed model

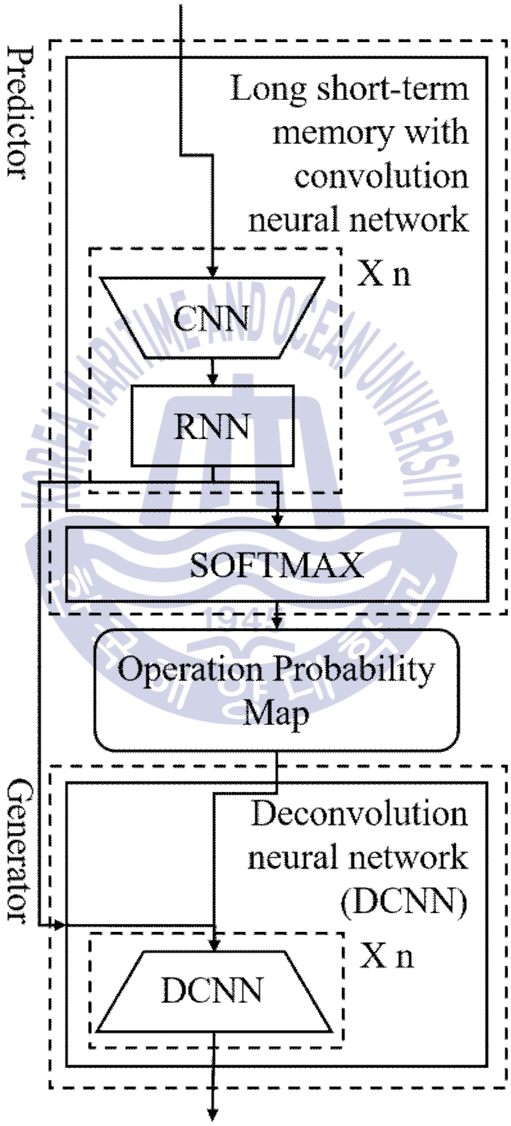

Predictor
Long short-term memory with convolution neural network
X n
CNN
RNN
SOFTMAX
Operation Probability Map
Generator
Deconvolution neural network (DCNN)
X n
DCNN

Fig. 3.2 The architectures of proposed model at the testing phase

Since the constructed model is based on map learning, it is divided into the training phase and testing phase. Fig. 3.2 shows the architecture during the testing phase of the proposed model. The layer configuration at the testing phase consists of a predictor and a generator. In order to learn several frames of video sequentially, time and spatial features are extracted through LSTM-CNN layer which combines LSTM node and CNN node. Based on the extracted features, we estimate the operation probability of the frame generated by the softmax layer. The operation probability map generated through this process is input to the generator together with the last feature map of the LSTM-CNN layers. The generator consists of deconvolution layers which are modified to fit the image generation by reversing the CNN node. An operation probability map and a feature map are input together so that only an area having a high operation probability is generated. The n shown in Fig. 3.2 is conFig.d by adjusting the input/output size of the predictor and generator.

Since GAN technique is introduced, training method is slightly different from general network. Thus, the model consists of adding a discriminator at the end of the generator during the training phase.

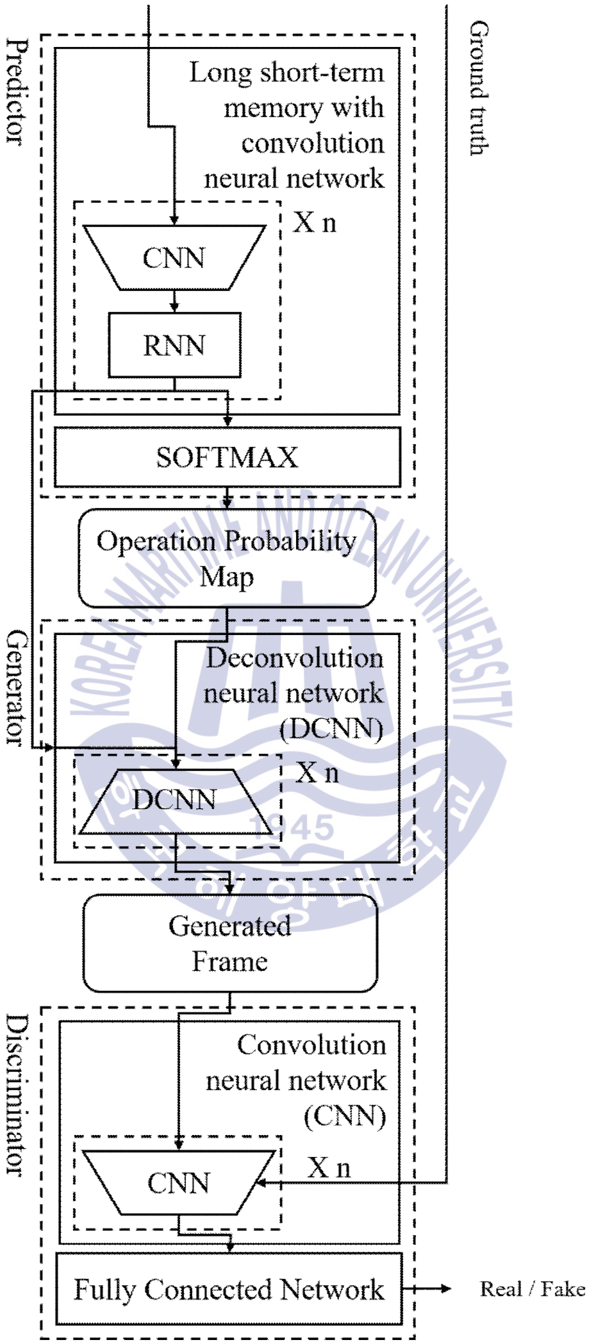

Predictor
Long short-term memory with convolution neural network
X n
CNN
RNN
SOFTMAX
Ground truth
Operation Probability Map
Generator
Deconvolution neural network (DCNN)
X n
DCNN
Generated Frame
Discriminator
Convolution neural network (CNN)
X n
CNN
Fully Connected Network
Real / Fake

Fig. 3.3 The architectures of proposed model at the training phase

Fig. 3.3 shows the architecture of the proposed model during the training phase, which shows the structure in which the discriminator is added by applying the GAN technique. It helps to learn the generator by adding a discriminator to the model during the testing phase. Since the performance of the discriminator is not an important issue in this model, a simple and superior model is selected and used as a discriminator. Therefore, the proposed model at the bottom of Fig. 3.3 learns whether the generated frame and ground truths are the original image or the fake image through CNN and the fully connected layer.

Through the training phase and the testing phase, we can improve the learning efficiency of the proposed model and reduce errors in the generation phase. In the following sections, the configuration and functions of predictors, generators, and discriminators are described.

3.2 Model for feature extraction and operation probability estimation

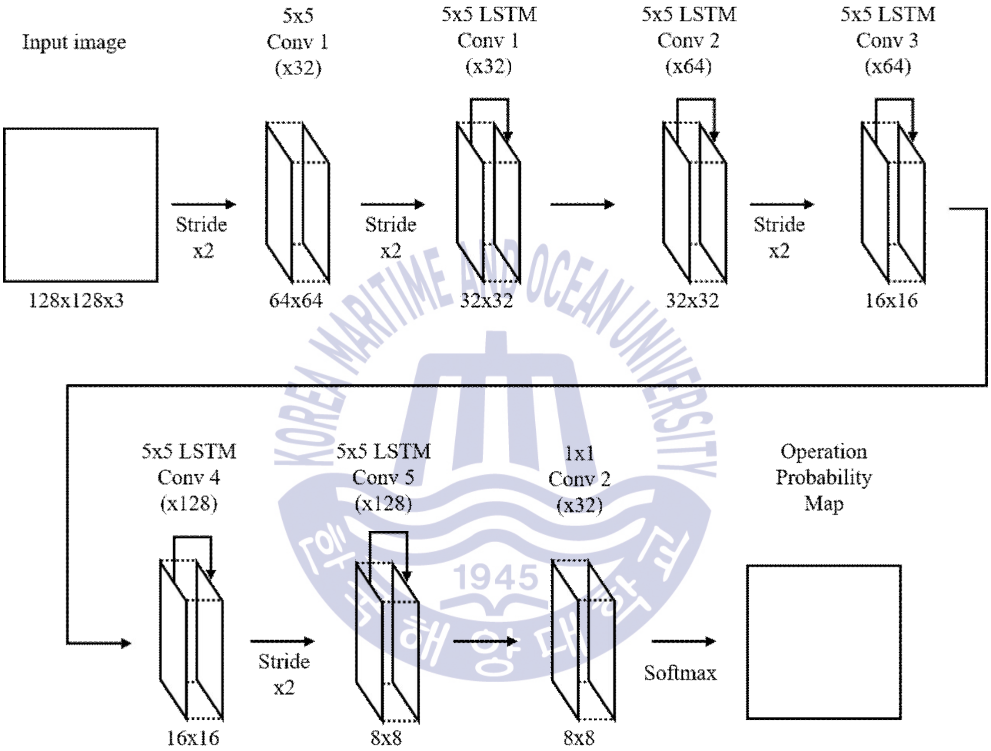


Fig. 3.4 The architectures of feature extraction and operation probability estimation

In this model, it is necessary to generate the map by estimating the operation probability to limit the operation of the unnecessary area. In order to generate this operation probability map, the motion possibility is estimated by extracting the spatial and temporal features in the video based on the artificial neural network. Fig.

3.4 shows the predictor in Fig. 3.2 and Fig. 3.3 in detail. The predictor consists of four different layers. A general CNN that extracts spatial features, rather than an operation that directly extracts motion features directly from the input image, is placed. Because the features extracted from the CNN in the first stage are boundaries, colors, etc., basic characteristics can be reliably extracted without affecting temporal influences[18]. Then, we use the RNN-based layer as several models that use existing sequential features to extract the features of the continuous operations in the frame progress of moving images. A feedforward model or a feedforward encoder based model[19] used to learn existing videos has considerable difficulty in learning. Also, when the RNN model is deepened, vanishing gradient problem is not learned and it is not suitable. In order to solve this problem, LSTM has been proposed which improved the problem of RNN by learning the result through several gates. Therefore, this study uses LSTM-CNN, which learns spatial features extracted from CNN through LSTM in time. The LSTM-CNN layer is constructed so that pixels at close distances can learn critical spatial images temporally.

In order to generate the operation probability map, the operation characteristics of the moving picture are learned, and the operation probability of the frame is estimated by the five stacked layers of the LSTM-CNN layers described above. After that, the CNN layer of 1×1 is placed, which replaces the existing pooling layer, compressing the kernel and helping to produce stable results on the back side. Finally, the feature map output from the CNN layer is transformed into a probability map through softmax, and the operation probability map is output. The input/output size for each side of the predictor can be derived from:

$$N_{x.i} = \frac{N_{x.i-1} - f + 2p}{s+1}$$
$$N_{y.i} = \frac{N_{y.i-1} - f + 2p}{s+1}, \tag{3.1}$$

where N is the size of the input/output channel, and the image is composed of the x axis and the y axis. i is the layer of the model, and $i-1$ is the entire layer of i. f is the size of the kernel, p is the size of the padding, and s is the size of the stride. This formula allows us to set a layer size and iteratively sets the layers between the given input image and the target output channel. Therefore, the size of the network is variable according to the size of the input image, and the form can be constructed through equation (3.1). In this paper, the size of the input image is 128×128, the size of the kernel is 5×5, the size of padding is 2, and the size of stride is 2.

3.3 Model for generating and combining images

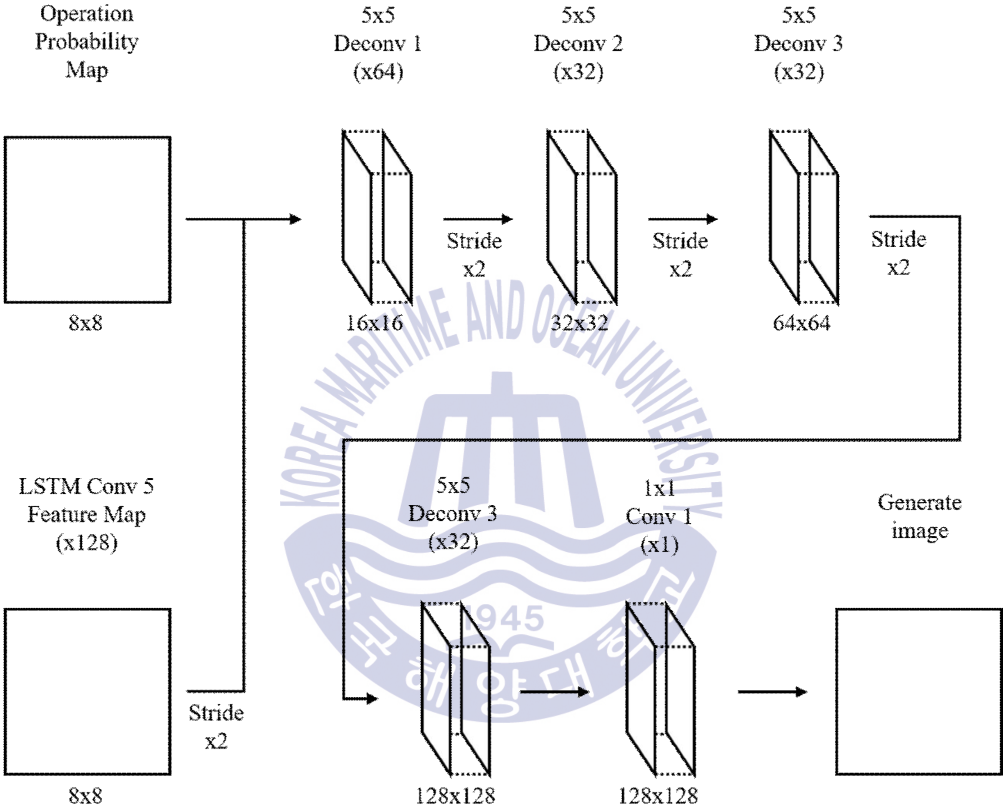


Fig. 3.5 The architectures of generating and combining images

The role of the generator based on the generation model is to generate the target image according to the feature map extracted by the predictor, and the autoencoder model[20] and the deconvolution model[21] are applied to the existing image generation model. Because autoencoder is an unsupervised learning base, learning data sets can be learned at least, but training speed is slow, and accuracy is low.

Therefore, we apply deconvolution model, which is a supervised learning model, to solve learning speed problem based on unsupervised learning. Also, to efficiently generate frames by suppressing the generation of unnecessary regions in the generation process, this model is generated only in the area where the future operation is predicted based on the estimated operation probability map. The image is then combined with the input original image to complete the frame. The frame generation process uses a deconvolution neural network that does not use the recurrent technique because it generates an image according to the expression type that is different from the operation probability estimation that recognizes the change of operation. Fig. 3.5 shows the model details of the generator in Fig. 3.3 and Fig. 3.4. The generator consists of two layers. It consists of a deconvolution layer for generating images based on input features and a CNN layer for pooling. The layer design of the deconvolution model is following as:

$$\begin{aligned} N_{x.i} &= (s+1)N_{x.i-1} + f - 2p \\ N_{y.i} &= (s+1)N_{y.i-1} + f - 2p, \end{aligned} \tag{3.2}$$

where N is the size of the input/output channel, and the image is composed of the x axis and the y axis. i is the layer of the model, and $i-1$ is the entire layer of i. f is the size of the kernel, p is the size of the padding, and s is the size of the stride. This formula allows us to determine the layer configuration to match the target image size. In this paper, the size of the output image is 128×128, the size of the kernel is 5×5, the size of padding is 2, and the size of stride is 2.

As shown in the left part of Fig. 3.5, the generator model inputs the feature map output from the LSTM-CNN 5 layer at the end of the predictor into the deconvolution layer, so that the output of the network used to estimate the operation probability share. Through this process, the generator model suppresses the generation of the low probability region and strengthens the high probability region, thereby eliminating the error in the generation process and unnecessary duplication.

3.4 Model for learning of generative model

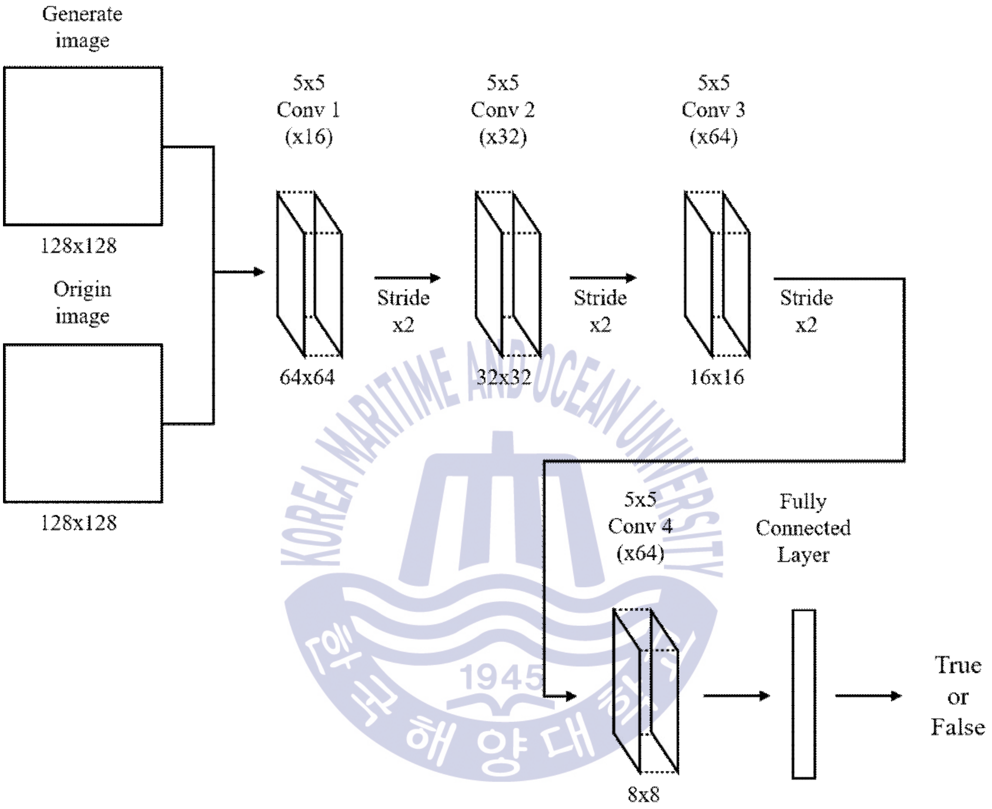


Fig. 3.6 The architectures of learning of generative model

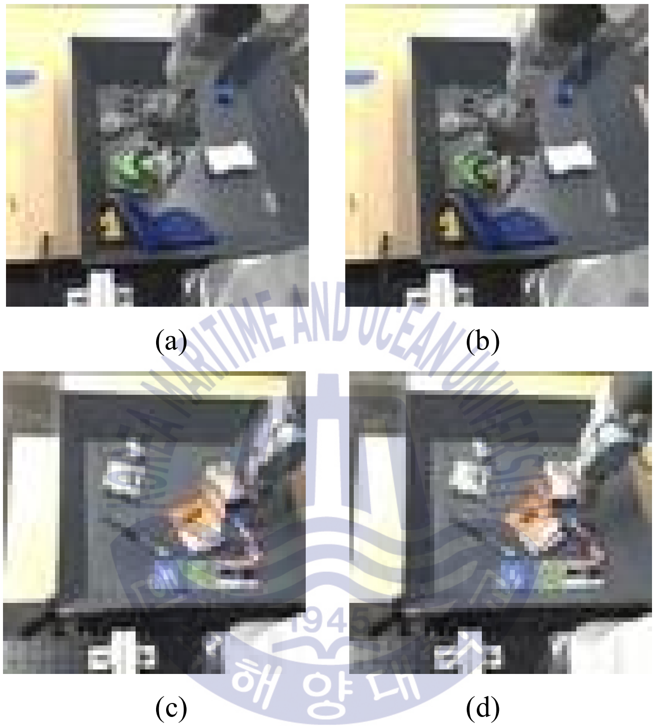
The general generative model has features that are very difficult to learn compared to other learning algorithms. In this paper, a deconvolution model composed of generative models also requires an assistive technique for learning. Therefore, this paper applies the GAN technique. In order to apply the GAN technique, it is necessary to select the generator and the discriminator appropriately. Fig. 3.6 shows the details of this discriminator. Since the generator is a deconvolution model to learn, and the discriminator is not a part that affects the performance of the model, we use a relatively simple classification model, CNN of 5 layers and fully connected layer. In this network, equation (3.1) is applied in the same way as the predictor, but the target output result is a slightly different design from 1×1.

Chapter 4 Experiments and result

4.1 Dataset for learning and testing

This section describes the dataset for learning the proposed model. Unlike artificially edited images, general natural images have a sudden and wide range of pixel changes except for the boundary area. Therefore, we use natural motion data sets with smooth motion for the prediction of moving images, which is the goal of this paper, We used the image frame prediction model. We selected about 2 million frames of data from the video data set used in other papers and used it for model learning. The verification was also conducted using some of the data sets used. The datasets used are two, Caltech Pedestrian Detection Dataset and Robotic Pushing Dataset. The Caltech Pedestrian Detection Dataset is a set of gait scenes taken by some fixed cameras and is suitable for learning and verifying motion that occurs throughout the screen. The Robotic Pushing Dataset is suitable for learning because it contains various types of motion generated by robots with a data set of 1.5 million frames consisting of 57,000 situations in which the robot moves objects in a fixed area. The Caltech Pedestrian Detection Dataset is used in the learning process and is not used for validation.

4.2 Analysis of experimental results



(a) (b)

(c) (d)

Fig. 4.1 The origin image of robotic pushing dataset

Experimental simulation implementation of this study is based on TensorFlow. For the learning and verification of the proposed model, we performed the Caltech Pedestrian Detection Dataset and Robotic Pushing Dataset described in Section 4.1. Learning was conducted by constructing mini-batches by randomly mixing two sets of data in video units for efficient learning and preventing overfitting. Fig. 4.1 shows part of the original image of Robotic Pushing Dataset. Fig. 4.1 (a), (b), (c) and (b) show the difference of 4 frames, which are different from (a), (b), (c) and (d). Experiments were carried out to generate the first frame by inputting randomly selected frames into the proposed model and then input the generated frames again to generate a total of 10 frames repeatedly. The results are compared with the original frame and the peak signal to noise ratio (PSNR).

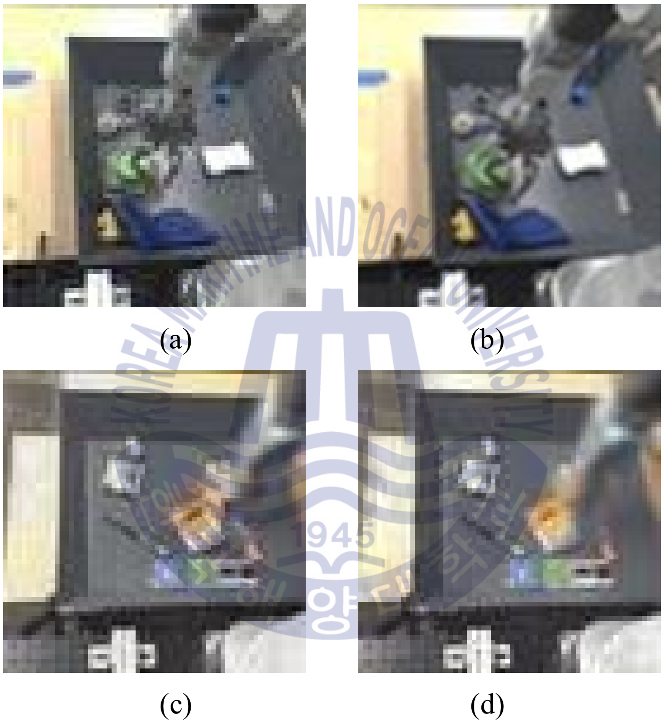


Fig. 4.2 The result image of proposed model

Fig. 4.2 shows the predicted frame results through the proposed model. Fig. 4.1 (a), (b), (c), and (d) are the same frames in Fig. 4.2. So we can compare the two to see the results. When the generated frame is visually confirmed, (a) and (c) are generated almost the same as the original image, but in (b), the position of the green object is predicted differently. In (d), it can be seen that the prediction of the

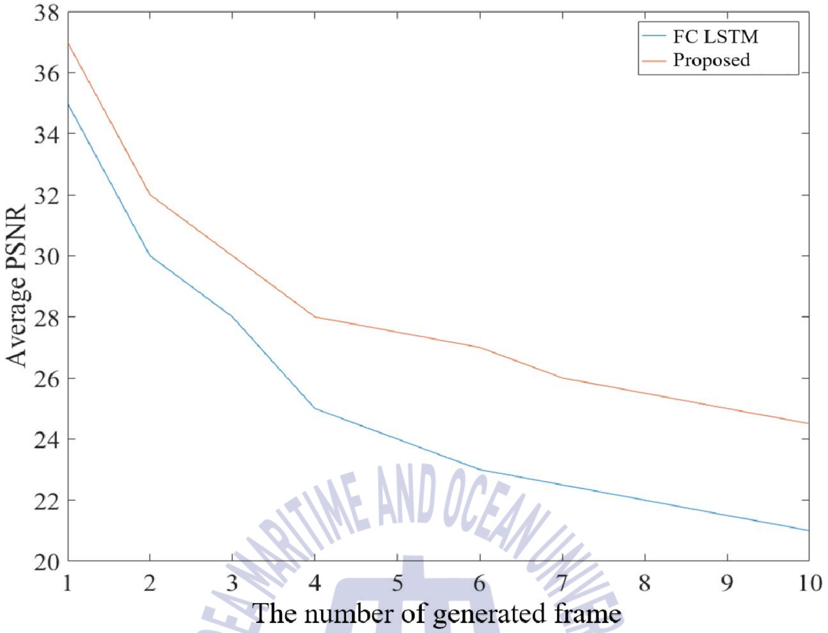


Fig. 4.3 Average PSNR according to the number of generated frames

region where the operation frequently occurs is blurred. It can be seen that the error that is generated as the frame progresses is accumulated and the prediction accuracy is lowered.

Fig. 4.3 shows the average PSNR for each frame generated. The horizontal axis in Fig. 4.3 represents the number of frames generated, and the vertical axis represents the average PSNR. The red line in Fig. 4.3 is the proposed model, and the blue line is the FC LSTM model. In both models, the PSNR decreases as the frame is generated. The decrease in PSNR is due to the accumulation of errors in the process of re-inputting the predicted frame, and the results of Fig. 4.1 and Fig. 4.2 can be numerically confirmed. We show that the proposed model is relatively

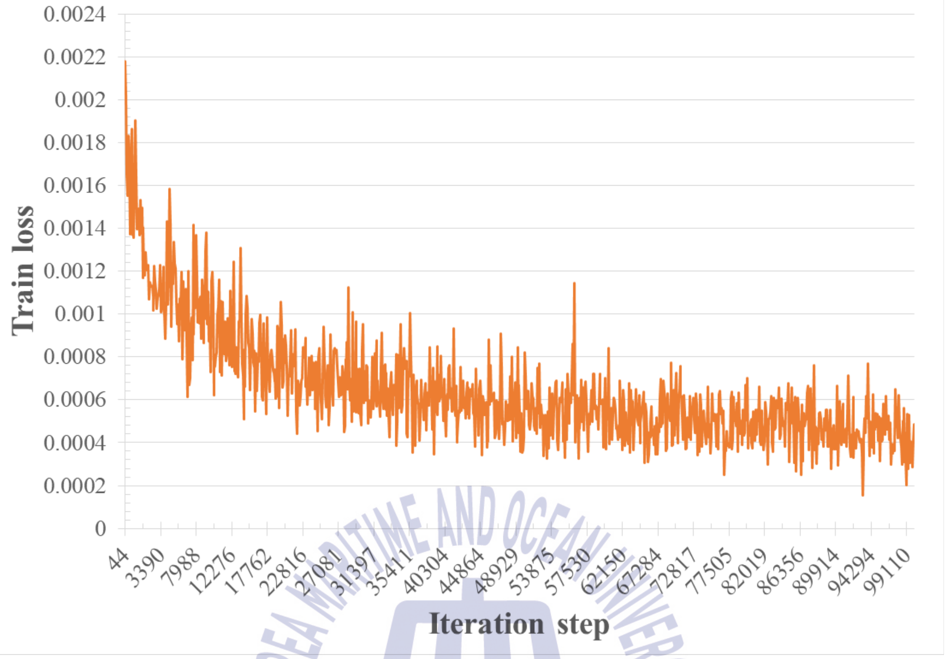


Fig. 4.4 Training loss according to iteration

improved because it is 37 in the first frame and 25 in the 10th frame and the existing algorithm is 34 to 24. Compared with the conventional algorithm, the proposed algorithm is suitable for multi-frame prediction by reducing the error in the accumulation process of the frame.

Fig. 4.4 shows the training loss during the training phase. The horizontal axis in Fig. 4.4 is the number of repetitions, and the vertical axis is the training loss. Fig. 4.4 shows that the iterations converge quickly at about 5000 times or less, and the convergence speed slows down after that. When the number of iterations exceeds about 50,000, it can be confirmed that convergence is almost completed.

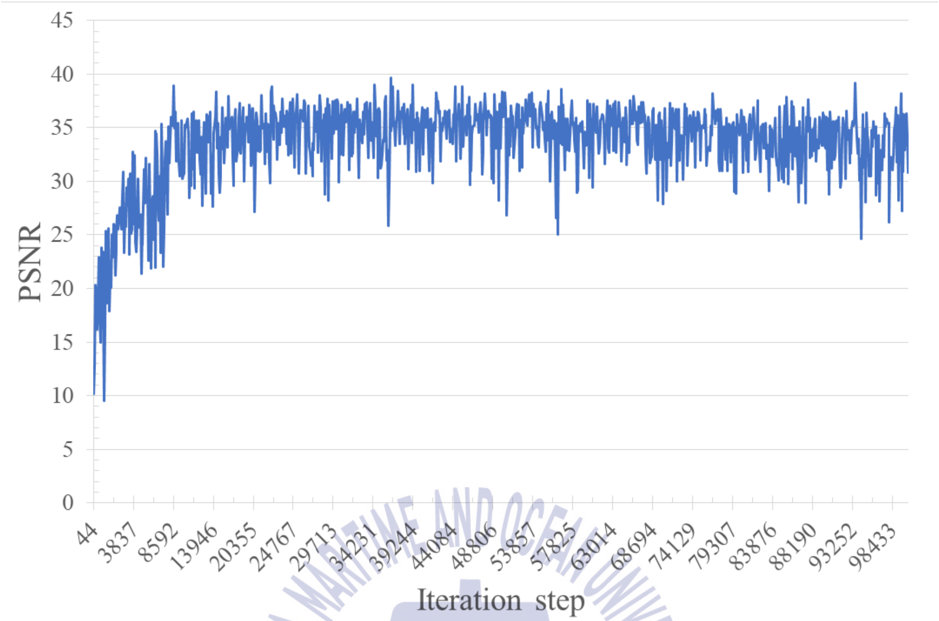


Fig. 4.5 PSNR according to iteration

Fig. 4.5 shows the PSNR during the raining phase. The horizontal axis in Fig. 4.5 is the number of repetitions, and the vertical axis is PSNR. As shown in Fig. 4.5, the PSNR value is over 30, which is a good accuracy, and the PSNR value converges to almost 50,000 when the repetition frequency is low.

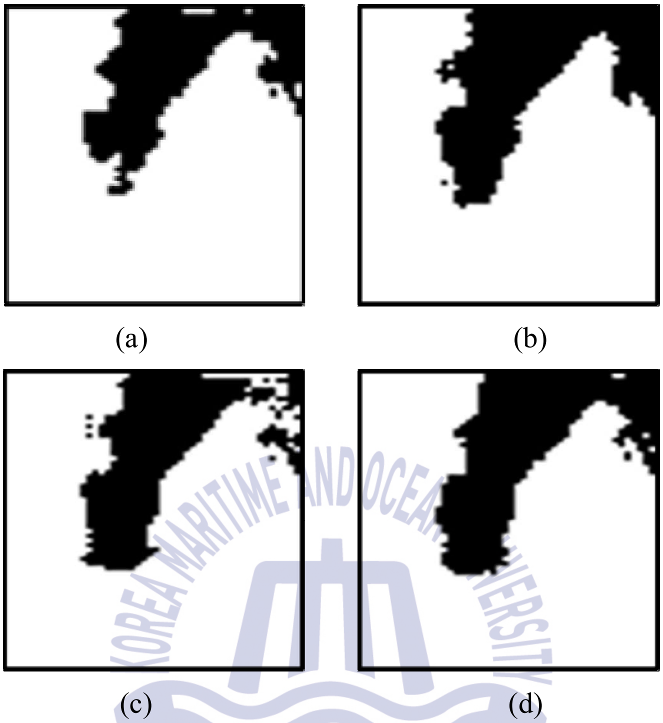


Fig. 4.6 Operational probability map extracted from the proposed model

Fig. 4.6 shows the transformation of the feature map from the predictor into a binary image by taking the threshold after converting it into probability through softmax. Fig. 4.6 (a), (b), (c), and (d) show four frames in order. Fig. 4.6 shows that the operation probability is concentrated in the robot arm part where an operation is most frequent. Therefore, the operation is appropriately predicted and generated.

Table 4.1 The average PSNR of simulation

Model	PSNR
Average frame	21.1
FC LSTM.[10]	24
CDNA[11]	35
Proposed	**35.16**

Finally, table 4.1 compares with the existing algorithms. For the basic comparison, we compared the algorithm used in frame interpolation and the frame prediction algorithms FC LSTM and CDNA with the average PSNR. The average frame is 21.1, the FC LSTM is 24, and the CDNA is 35. The proposed model is 35.16, which is slightly improved than CDNA.

Chapter 5 Conclusion

In this study, we applied the deep learning method to frame prediction of moving picture. It is a model that predicts the frame by estimating the occurrence probability of motion based on the learned motion characteristics in the previous frame. We have learned and verified by using video images of robot motion changes and various videos of 2 million frames to demonstrate model learning and validity. Model learning was randomly sampled to prevent model overfitting for the input video. Also, the experiment repeatedly feeds back the generated frame to analyze the error caused by the accumulation of the prediction frame, generates ten frames afterward, and compares the result with the original frame using the peak signal-to-noise ratio (PSNR). As a result, the performance was verified by training loss and PSNR. The result of this paper is 35.16 PSNR which is better than the existing algorithm and the PSNR reduction according to the generated frame is improved to 25 from the 10th frame.

Experimental results show that the proposed model reduces the error as more frames are produced than the conventional algorithm. However, since the estimation result of the probability of operation is small, the motion of the small object is not applied to the predicted frame. Therefore, further studies are needed to consider the probability of small motion in future studies.

Through this study, it is expected that it can be applied as an auxiliary algorithm of image analysis by increasing the number of frames per second of images collected by industrial or publicly installed image devices with limited performance.

Reference

[1] Revaud, J., Weinzaepfel, P., Harchaoui, Z., & Schmid, C. (2015). “Epicflow: Edge-preserving interpolation of correspondences for optical flow”. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1164-1172).

[2] Meyer, S., Wang, O., Zimmer, H., Grosse, M., & Sorkine-Hornung, A. (2015). “Phase-based frame interpolation for video”. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1410-1418).

[3] Choi, D., Song, W., Choi, H., & Kim, T. (2016). “MAP-Based Motion Refinement Algorithm for Block-Based Motion-Compensated Frame Interpolation”. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(10), (pp. 1789-1804).

[4] Rüfenacht, D., & Taubman, D. (2016, September). “Temporally consistent high frame-rate upsampling with motion sparsification”. *In Multimedia Signal Processing (MMSP), 2016 IEEE 18th International Workshop* on (pp. 1-6).

[5] Guo, D., & Lu, Z. (2016). “Motion-compensated frame interpolation with weighted motion estimation and hierarchical vector refinement”. *Neurocomputing*, 181, (pp. 76-85).

[6] Bengio, Y. (2009). “Learning deep architectures for AI”. *Foundations and trends® in Machine Learning*, 2(1), (pp. 1-127).

[7] Schmidhuber, J. (2015). “Deep learning in neural networks: An overview”. *Neural networks*, 61, (pp. 85-117).

[8] Long, G., Kneip, L., Alvarez, J. M., Li, H., Zhang, X., & Yu, Q. (2016, October). “Learning image matching by simply watching video”. *In European Conference on Computer Vision* (pp. 434-450).

[9] Niklaus, S., Mai, L., & Liu, F. (2017). “Video Frame Interpolation via Adaptive Convolution”. *arXiv preprint* arXiv:1703.07514.

[10] .Mathieu, M., Couprie, C., & LeCun, Y. (2015). "Deep multi-scale video prediction beyond mean square error". *arXiv preprint* arXiv:1511.05440.

[11] Oh, J., Guo, X., Lee, H., Lewis, R. L., & Singh, S. (2015). "Action-conditional video prediction using deep networks in atari games". *In Advances in Neural Information Processing Systems* (pp. 2863-2871).

[12] Finn, C., Goodfellow, I., & Levine, S. (2016). "Unsupervised learning for physical interaction through video prediction". *In Advances in Neural Information Processing Systems* (pp. 64-72).

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks". *In Advances in neural information processing systems* (pp. 1097-1105).

[14] Hochreiter, S., & Schmidhuber, J. (1997). "Long short-term memory". *Neural computation*, 9(8), (pp. 1735-1780).

[15] Pinheiro, P., & Collobert, R. (2014, January). "Recurrent convolutional neural networks for scene labeling". *In International Conference on Machine Learning* (pp. 82-90).

[16] Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016, August). "Dimensional sentiment analysis using a regional CNN-LSTM model". *In ACL 2016—Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany* (Vol. 2, pp. 225-230).

[17] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). "Generative adversarial nets". *In Advances in neural information processing systems* (pp. 2672-2680).

[18] Radford, A., Metz, L., & Chintala, S. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks". *arXiv preprint* arXiv:1511.06434.

[19] Zeiler, M. D., & Fergus, R. (2014, September). "Visualizing and understanding convolutional networks". *In European conference on computer vision* (pp. 818-833).

[20] Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). "DRAW: A recurrent neural network for image generation". *arXiv*

preprint arXiv:1502.04623.

[21] Hong, C., Yu, J., Wan, J., Tao, D., & Wang, M. (2015). “Multimodal deep autoencoder for human pose recovery“. *IEEE Transactions on Image Processing*, 24(12), (pp. 5659-5670).

[22] Xu, L., Ren, J. S., Liu, C., & Jia, J. (2014). “Deep convolutional neural network for image deconvolution”. *In Advances in Neural Information Processing Systems* (pp. 1790-1798).