Journal of the Korean Society of Marine Engineering, Vol. 42, No. 2 pp. 121~126, 2018
J. Korean Soc. of Marine Engineering (JKOSME)
https://doi.org/10.5916/jkosme.2018.42.2.121

ISSN 2234-7925 (Print)
ISSN 2234-8352 (Online)
Original Paper

Image reconstruction technique using deep learning architecture

Soo-Hwan Lee[1] · Jong-Chan Kim[2] · Dong-Hoan Seo†

(Received January 26, 2018 ; Revised February 5, 2018 ; Accepted February 18, 2018)

Abstract: Existing frame interpolation techniques require a large amount of training data because they use models to extract general motions from a large number of videos. Using these models, they estimate a motion vector for the next frame using a motion vector from the previous frame of the same video; this causes poor accuracy, characterized by blurred or distorted images, because the cumulative error increases with the learning time and the number of generated frames. Therefore, it is necessary to construct a robust model, by simplifying the learning and generation models, which reduces computation and generates only the changed regions of the intermediate frames. In order to realize a simplified model that generates pixels in changed regions, we constructed a motion feature extraction and learning model based on convolution neural network (CNN) and recurrent neural network (RNN). In addition, we propose an algorithm which used a motion probability map to generate intermediate frames via deconvolution layers for regions where motion occurs. To train and verify the proposed model, we performed experiments based on actual videos from a dataset created by capturing the motions of a robot.

Keywords: Deep learning, Frame interpolation, Image reconstruction

1. Introduction

It is relatively easy for humans to infer the intermediate conditions by recognizing the situation before and after. However, it is difficult for machines to produce the same results even for variations of the same object with different size, color, or shape. This is because machines learn an entire image with no divisions, but the accuracy of image understanding is low since similar object transformations occur frequently in real situations, such as in autonomous navigation, and unattended surveillance. Several studies in the field of computer vision have attempted to address this problem. Since the emergence of deep learning, the reasoning capabilities of artificial intelligence have strengthened, thereby improving its problem-solving ability. However, the application of this deep learning model in real time is limited due to the large amount of computation required; as such, there is a need for an approach which reduces the amount of computation by selectively learning space or time, rather than the heuristic method of the learning model **[1]-[3]**.

Frame interpolation is a technique which can be used to generate intermediate images between frames based on two consecutive frames. It has been studied and used in various applications such as animation, and image restoration. Previous studies have calculated the pixel values of an intermediate image by using the difference in pixel values between the previous and subsequent frames, while others have used the overall tendency by extracting motion vectors from the entire video. In addition, algorithms for interpolating frames have been investigated using convolution neural networks (CNNs), which extract image features based on deep learning, and recurrent neural networks (RNNs), which learn the sequential nature of image features recursively **[4]**.

Therefore, in this study, we propose a frame interpolation algorithm that predicts regions of motion between two consecutive frames, generates intermediate frames, and combines these with the original frames. We present an algorithm that can simultaneously reduce the amount of noise and computation by using a model consisting of CNNs and RNNs to learn the changes in between frames. The algorithm generates a motion probability map in order to remove regions where motion does not occur, and to combine regions that have a low probability of motion with previous frames. For the learning

† Corresponding Author (ORCID: http://orcid.org/0000-0003-3610-0356): Division of Electronics and Electrical Information Engineering, Korea Maritime and Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: dhseo@kmou.ac.kr, Tel: 051-410-4412
1 Department of Electrical and Electronics Engineering, Korea Maritime and Ocean University, E-mail: config5246@naver.com, Tel: 051-410-4822
2 Department of Electronics, Kyungbuk College, E-mail: kjc@kbc.ac.kr, Tel: 054-630-5067

and verification of the proposed model, experiments were carried out using natural images (30 frames per second or greater). We compared the 10 frames that were generated with actual frames and represented the differences by various indices.

2. Related work

Image frame interpolation has been extensively studied as a technique for image processing and computer vision. Most existing frame interpolation algorithms have been designed to generate an intermediate frame by associating the frames prior to, and subsequent to the target frame, through the optical flow of two or more input frames containing numerous motions **[5][6]**. These methods work without complex configurations such as learning because they use only the video subjected to frame interpolation. However, this method is limited because it uses only light features, and does not consider motion features. Thus, videos with significant changes in light or complex motion result in inappropriate interpolation.

Other studies have expressed motions in video by analyzing the differences between consecutive images and changing the magnitude of the difference to the descriptor (also called a motion vector). The A technique which uses this descriptor to interpolate frames by compensating for the motion of the next image has also been studied **[7]-[9]**. This method used the descriptor to expresses the characteristics of changes, thereby enabling more accurate interpolation as various complex changes can be accounted for. However, such methods are vulnerable to unusual or poor-quality images since they are based on the general features of the progress of images.

In this study, we constructed an image frame interpolation model by applying a deep learning algorithm. This algorithm is able to accurately extract the features of various signals because it stacks up neural networks step by step, thus achieving a higher abstraction performance compared to existing learning algorithms. G. Long *et al.* **[10]** performed frame interpolation for image matching by applying deep learning, but the interpolated frames were blurred. S. Niklaus *et al.* **[11]** generated intermediate frames by combining frames generated from a model comprising of the previous and subsequent frames, and a CNN.

3. The proposed frame interpolation model

Existing frame interpolation methods generate an intermediate frame, with the same size as the input images, by learning a motion model, and creating a background image from the neighboring input images. The intermediate frame is completed by combining the model with the background image. These methods can reduce errors in the generation step but lead to additional unnecessary computation, and errors. These errors occur where motion occurs locally, as the generation unit time is reduced, resulting in a large difference between the background regions of the entire image, and the regions without motion. Therefore, this study proposes a model, based on an actual video, which generates an intermediate frame by predicting motion in order to reduce unnecessary computation and errors. **Figure 1** represents an overall framework of the proposed model. Intermediate frames are generated by predicting the actual regions of motion, and partially estimating the next motion. The image in **Figure 1** is spatially

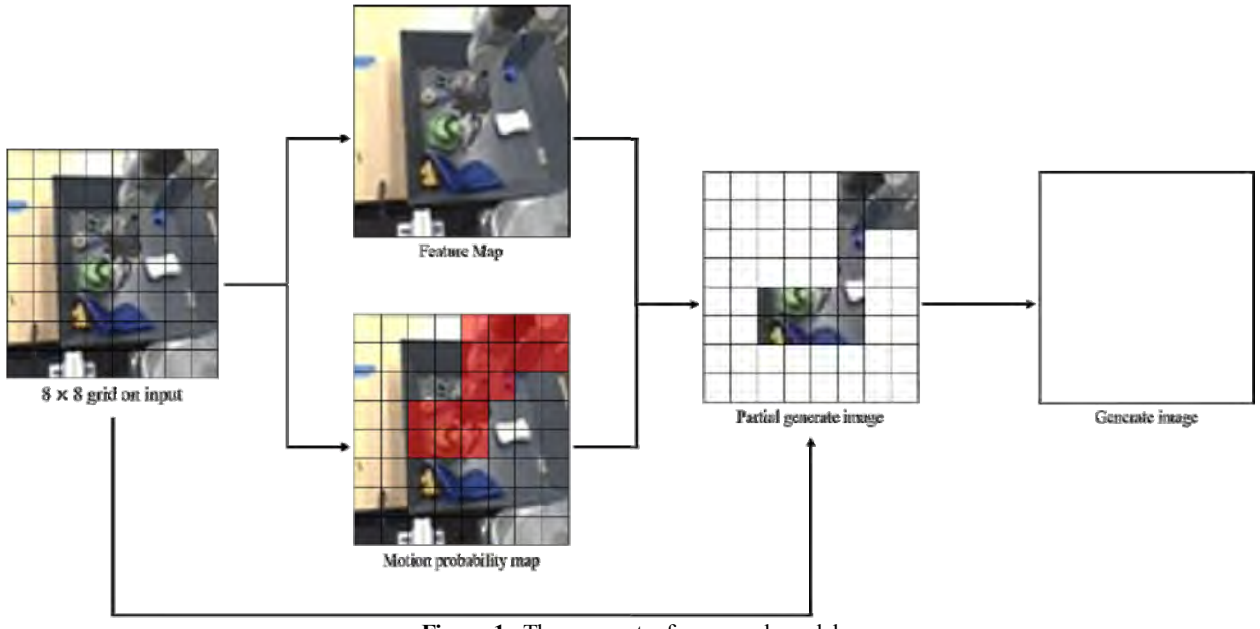


Figure 1: The concept of proposed model

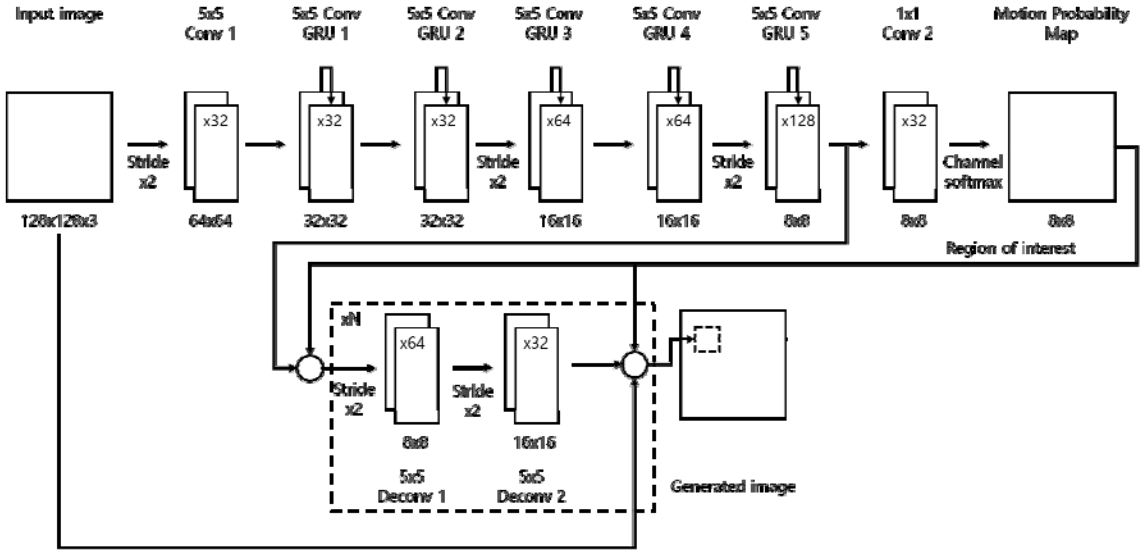


Figure 2: The architectures of proposed model

divided by applying a deep neural network to the two neighboring frames of an image, and learning the motion features of the image. In addition, we propose an algorithm that generates an intermediate frame by generating only the regions that have a possibility of motion according to the motion probability map. The algorithm then combines these regions with the no-motion regions of the previous frame. The proposed model is divided into two parts: creating a matrix by predicting the motion probability of each region, and predicting the intermediate frame of the regions having a high probability which is combined with the regions that have a low probability. **Figure 2** shows the overall configuration of the proposed model; the top section shows the network that estimates the motion probability in the previous frame, and the section below, inside the rectangle with the dotted line, indicates the network that generates each region of interest part by part.

3.1 Pixel transformation to motion probability

The proposed model estimates the motion probability based on the neural network in order to reduce the computation of unnecessary regions; the motion probability of each object in the image determines whether a region can be classified as unnecessary. This estimation is carried out by the structure shown in the lower part of **Figure 1**. A RNN is used, as with existing models that use several sequential frames, to extract the features of successive motions from the previous and subsequent frames of an image. Unlike existing feedforward models, or feedforward encoder-based models that are used for learning images, we applied deep convolutional gated recurrent units (GRUs) to sequentially learn the results of the previous and next frames. A convolutional recurrent network connects the recurrent network that processes the separated sequential information with the pixels in close proximity; because of this, it is suitable for videos with multiple frames that represent an important spatial image. Moreover, the amount of computation can be reduced by using GRU, which is a simplified model of the long short-term memory (LSTM) network that was used in many existing studies.

To generate a motion probability map, the motion features are learned by the model in which the convolutional GRUs are stacked, and the motion probability of each region is outputted through the softmax channel of the last layer. The model for outputting this motion probability is the part between the input and the output, which is the motion probability map shown in the top of **Figure 2**. It is composed of five layers of convolutional GRUs, three convolution layers, and a softmax layer.

$$N_x = \frac{A}{16} \qquad (1)$$
$$N_y = \frac{B}{16}$$

The motion probability map expresses the probability by dividing the input image, shown in **Figure 1**, into 8×8 regions. The generated grid is variable according to the size of the input image and can be calculated using **Equation (1)**. $A \times B$ represents the size of the input frame. N_x and N_y represent the grid numbers of the x-axis and y-axis, respectively, on the motion probability map. The size of the minimum divided region is 16×16. In this study, the size of the divided region is set to 8×8 since the size of the target image is 128×128.

3.2 Generating and compositing image

References To effectively generate an intermediate frame by suppressing the generation of unnecessary regions, this model

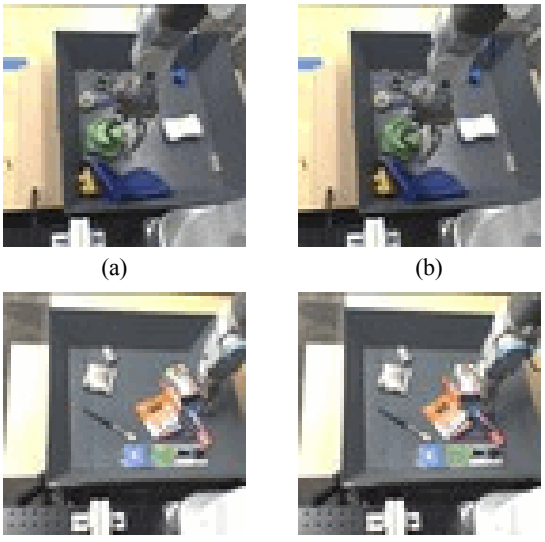
only generates an image for regions where motion is predicted according to the motion probability map. The generated image is combined with the input image, creating the intermediate frame. In general, an autoencoder, or deconvolution model is applied for image generation. The autoencoder model is based on non-supervised learning and can be used with small datasets; however, it has a relatively low accuracy and therefore, we chose to use the deconvolution model. For the intermediate frame generation model, we use typical deconvolution layers with no recurrent techniques. This model generates an image according to the expression type of the implied motion, unlike motion probability estimation, which varies according to the characteristics of the motion. The lower part of **Figure 2** shows the details of its structure. Duplicate computations can be reduced by sharing the output of the network that is used to estimate the motion probability, rather than extracting new motion features. The intermediate images are generated through the deconvolution layers, which are based on the probability map, by connecting the tensor of the motion probability map with the layer in the generation process. By pooling the generated images and the input images, duplication of the unchanged regions can be avoided.

4. Datasets

In general, there are no abrupt or extensive pixel changes for natural images except for the boundary region. Thus, we verified the proposed image frame interpolation model using a natural image dataset with smooth motions. This study used 2 million frames for model learning selected from video datasets used in other studies. Two datasets were used for the model learning, the Caltech pedestrian detection dataset, and the robotic pushing dataset. The Caltech pedestrian detection dataset is a collection of pedestrian walking scenes captured by multiple fixed cameras; it is suitable for learning and verifying motions that occur in the entire the image. The robotic pushing dataset is suitable for learning because it contains various types of motion generated by robots with a dataset of 1.5 million frames consisting of 57 000 situations, wherein robots move objects in a fixed area.

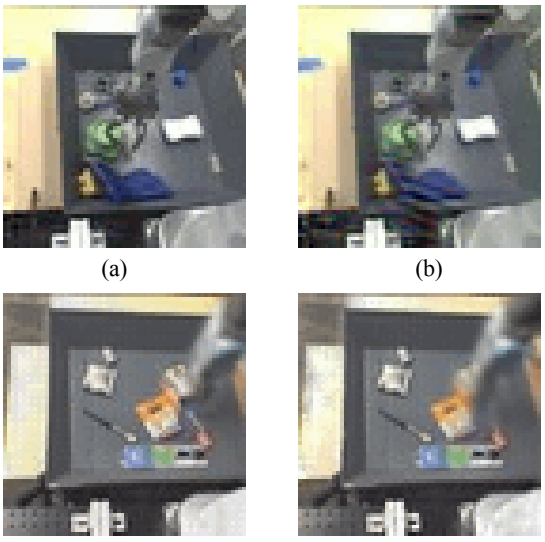
5. Experiments

We used the Caltech pedestrian detection dataset, and the robotic pushing dataset described in Section 4 to verify the proposed model. To generate intermediate frames in the two datasets, we generated the initial images using the proposed model and then generated up to 10 frames by re-inputting the generated images. All experiments were conducted using the TensorFlow library and were optimized using the Adam algorithm. **Figure 3** shows the original images from the robotic pushing dataset. There are four frame differences between (a) and (b), and between (c) and (d). The situation of (a) and (b) is different from that of (c) and (d). **Figure 4** shows the results of the proposed model with the same elements as **Figure 3.** Therefore, we can confirm the performance by comparing **Figure 3** with **Figure 4**. When the generated intermediate frames are seen with the naked eye, (a) and (c) are almost identical, but the position of the green object is predicted differently for (b). In addition, it can be seen that the prediction for a region with frequent motion is blurred in (d). **Figure 5** shows the results of the peak signal-to-noise ratio (PSNR) expressed in each frame, where the horizontal axis represents the number of frames, and the vertical axis represents the average PSNR. **Figure 5** confirms that the PSNR drops as the frame



(a) (b) (c) (d)

Figure 3: The original image



(a) (b) (c) (d)

Figure 4: The result

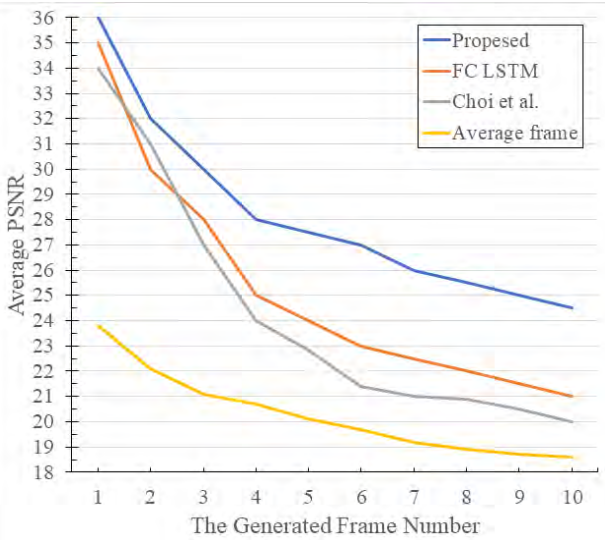


Figure 5: The result

Figure 6: The motion probability map

progresses. This is because the intermediate frame is not generated based on the original image but by inputting the predicted frame again; therefore, the accuracy decreases due to the accumulation of errors. The average PSNR of the proposed model is 37 for the first frame and 25 for the tenth frame. Given that the existing algorithm has average PSNRs of 34 and 24, the accuracy is shown to have improved. As the number of frames increases, the average PSNR decreases. This is because the generation of the first frame is based on the original frame, but further frames are based on the previously generated frame, resulting in an accumulation of errors. Therefore, if we focus on generating a single frame, we can obtain an image similar to the original frame, which is confirmed by **Figures 4 (b)**, and **4 (d)**. **Table 1** shows a comparison of the predicted results for the first frame with those of the existing algorithms and confirms that the accuracy of the proposed model is higher than that of the existing algorithms. **Figure 6** shows the evidence of the softmax layer used to output the motion probability map based on the feature map computed through the model. The four images are binary images created by inputting robot motion videos into the proposed model, and giving thresholds for the generated motion probability maps. Comparing the four elements of **Figure 6** with **Figure 3**, it can be seen that the threshold value is achieved in the area around the robot arm and the area in contact with the object.

Table 1: The average PSNR of the previous 5 frames output from the experiment

Model	PSNR
Average frame	21.6
D. S. Choi *et al.* **[7]**	27.8
FC LSTM **[12]**	28.4
Proposed	**30.7**

6. Conclusion

In this study, we proposed a model for the frame interpolation of images where intermediate frames were generated by estimating the motion probability based on the motion features of the previous frame. To demonstrate the validity of the model, we trained and verified the model by using robot motion images and videos with 2 million frames. We generated 10 frames in the experiment to investigate the sequential changes of the video and compared this model with existing algorithms

It should be noted that the proposed model is based on supervised learning; therefore, there is a possibility that it may not be robust against new types of motion. Therefore, further research is needed for an unsupervised learning model. Further, a typical frame interpolation technique generates an intermediate image by comparing the previous and subsequent frames, and it is similar to a video prediction technique that predicts the next frame using only the previous frame. Therefore, the same model could be applied to both techniques.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NO.2016R1 D1A1B03934812).

References

[1] J. H. Seong and D. H. Seo, “Environment adaptive localization method using Wi-Fi and bluetooth low energy,” Wireless Personal Communications, pp. 1-14, 2017.

[2] J. H. Seong, D. H. Seo, E. C. Choi, and J. S. Lee,

"High-speed positioning and automatic updating technique using Wi-Fi and UWB in a ship," Wireless Personal Communications, vol. 94, no. 3, pp. 1105-1121, 2017.

[3] W. Y. Kim, D. H. Seo, and J. C. Kim, "Welding and weaving speed estimation using acceleration sensor based on SVM," Journal of the Korean Society of Marine Engineering, vol. 41, no. 6, pp. 1018-1023, 2017 (in Korean).

[4] J. H. Lee, D. H. Seo, and J. C. Kim, "A study on image caption algorithm based on object detection," Journal of the Korean Society of Marine Engineering, vol. 41, no. 7, pp. 683-689, 2017 (in Korean).

[5] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1164-1172, 2015.

[6] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. S. Hornung, "Phase-based frame interpolation for video," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1410-1418, 2015.

[7] D. S. Choi, W. S. Song, H. Choi, and T. J. Kim, "MAP-based motion refinement algorithm for block-based motion-compensated frame interpolation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 10, pp. 1789-1804, 2016.

[8] D. Rüfenacht and D. Taubman, "Temporally consistent high frame-rate upsampling with motion sparsification," Multimedia Signal Processing (MMSP), 2016 IEEE 18th International Workshop on. IEEE, pp. 1-6, 2016.

[9] D. Guo, and Z. Lu, "Motion-compensated frame interpolation with weighted motion estimation and hierarchical vector refinement," Neurocomputing, vol. 181, no. C, pp. 76-85, 2016.

[10] G. Long, L. Kneip, J. M. Alvarez, and H. Li, "Learning image matching by simply watching video," European Conference on Computer Vision, Springer International Publishing, 2016.

[11] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," arXiv preprint arXiv:1703.07514, 2017.

[12] M. Mathieu, C. Couprie, and Y. LeCun. "Deep multi-scale video prediction beyond mean square error," arXiv preprint arXiv:1511.05440, 2015.