# IIEEJ Transactions on

# Image Electronics and Visual Computing

## Special Issue on
## Internet of Things and its Related Technologies in Image Electronics

Vol. 7, No. 1 2019

The Institute of Image Electronics Engineers of Japan

# Editorial Committee of IIEEJ



## Contact Information

The Institute of Image Electronics Engineers of Japan (IIEEJ)
3-35-4-101, Arakawa, Arakawa-ku, Tokyo 116-0002, Japan
Tel : +81-3-5615-2893 Fax : +81-3-5615-2894
E-mail : hensyu@iieej.org
http://www.iieej.org/ (in Japanese)
http://www.iieej.org/en/ (in English)
http://www.facebook.com/IIEEJ (in Japanese)
http://www.facebook.com/IIEEJ.E (in English)

IIEEJ Transactions on
Image Electronics and Visual Computing
Vol.7 No.1 June 2019

CONTENTS



Published two times a year by the Institute of Image Electronics Engineers of Japan（IIEEJ）
3-35-4-101, Arakawa, Arakawa-ku, Tokyo 116-0002, Japan
Tel：+81-5615-2893 Fax：+81-5615-2894 E-mail：hensyu@iieej.org http://www.iieej.org/

# Upon the Special Issue on Internet of Things and its Related Technologies in Image Electronics

Editor
Shunichi KIMURA
(Fuji Xerox Co., Ltd.)

In recent years, a wide variety of devices designed to operate over the Internet are increasingly coming into the market. These devices and related technologies, commonly called Internet of Things (IoT), are expected to create new social values. Moreover, from a viewpoint of Image Electronics Technology, widespread of small optical devices in mobile phones, automobiles, or personal computers are also regarded as IoT devices because these devices can be used as connected sensors to get image information. Image Electronics Technology treating such imaging information will be one of the main fields to enhance studies of the IoT.

At the call for submission, we received several interesting papers in this special issue. Based on the extensive peer review processes, six papers are accepted for publication; "Diminished Reality for Privacy Protection by Hiding Pedestrians in Motion Image Sequences Using Structure from Motion" shows a novel object hiding method, "A Development of Wearable System for Information Providing Applications Using Scene Text Recognition" and "Designs and Evaluation of Raccoon Detector for Anti-Vermin Surveillance Cameras" report new systems for practical applications, "Green-Noise Diffusion Watermarking Method Enabling to Embed Multiple Information" proposes a sophisticated digital watermarking method embedding multiple information, and "A Flicker Noise Reduction Method of Holographic Projected Images with the Use of the Fog Screen" and "Fast Image Quality Enhancement for HEVC by Postfiltering via Shallow Neural Networks" devise effective methods to improve image qualities. First five papers are written in Japanese and already published in IIEEJ Journal in April, and the last one written in English is included in this Transaction.

Finally, I would like to thank all the reviewers for their enormous effort in producing careful and effective reviews for submitted papers. I am also deeply grateful to the members of the editorial committee of IIEEJ and the staff at IIEEJ office for various kinds of support.

IIEEJ **Paper** *-- Special Issue on Internet of Things and its Related Technologies in Image Electronics --*

# Fast Image Quality Enhancement for HEVC by Postfiltering via Shallow Neural Networks

Antoine CHAUVET†, Tomo MIYAZAKI†, Yoshihiro SUGAYA†, Shinichiro OMACHI†(*Member*)

† Tohoku University

<**Summary**> The study proposes a lightweight adaptive postfilter based on neural networks for use in H.265 High Efficiency Video Coding (HEVC). The proposed filter is adaptive because it uses a different set of parameters based on encoding settings and most significantly on the quantization parameter. With the aforementioned information, the filter most efficiently improves each block. We trained the filter for 4 different QP values and we demonstrate that the use of the filter leads to a decrease in bitrate of over 4% in a few cases and a decrease in bitrate of 1.5% on average for both All Intra and Random Access modes. In contrast to a few filters that use several passes and require specific ordering, the proposed filter changes each pixel at most once and the input uses only initial values, thereby allowing perfect parallelization. Furthermore, the use of only one convolutional layer and eight feature layers maintains the computing cost and memory footprint to the minimum possible extent, and this makes real-time processing possible even on embedded hardware.

**Keywords**: block-based coding, video coding, video filtering, convolutional neural networks

## 1. Introduction

High Efficiency Video Coding (HEVC)[1]is a standard of video coding that is used extensively for High Definition content. However, defects appear on videos coded via HEVC. For example, blocking artifacts are visible defects caused by the division in coding units. Furthermore, fine details are typically blurred because the high frequency components are more aggressively quantized. This can make text unreadable at lower bitrates.

In order to reduce the artifacts caused by encoding, recent standards, such as Advanced Video Coding (AVC), included a deblocking filter[2]. Specifically, HEVC is a successor of AVC that also includes a deblocking filter[3] and a new Sample Adaptive Offset (SAO) filter[4]. The former exhibits a significant improvement by removing block artifacts with the only disadvantage that it can filter excessively and blur a few details. The latter is most useful to reduce ringing artifacts that occur around edges because of the high frequency and quantization. Both filters are in-loop, and this implies that the filtered frame is used for reference in contrast to filters that process the decoded stream and are also termed as postfilters.

Recently, the Internet of Things is an extremely popular subject of research and its popularity leads to the need for efficient processing of data because costs and electrical consumption should remain low. Several small devices, such as phones and tablets, stream video from the Internet, and it is not surprising that a refrigerator or an oven that performs the same will be developed in a few years. Naturally, algorithms that can efficiently process video are desired. The aim of the proposed filter is that it should be operable on the aforementioned devices, and thus it should be sufficiently simple to respect the processing power limitations of these devices.

As shown in the AVC in-loop deblocking filter, an understanding of the quantization parameter (QP) aids in determining whether edges are more likely to be artifacts or true edges and increase filtering efficiency. Thus, the proposed filter uses the QP value as a parameter in conjunction with the frame input.

The proposed filter runs after the decoding, and thus the performance concerns from using an in-loop deblocking filter can be alleviated. The limitation of in-loop filters is that they should be processed in encoding order because the frame is then used as a reference. This creates two issues when the filter becomes too complicated. First, there are inherent limits on parallelization because a frame must be filtered before commencing to process the next frame. Second, skipping the filter when the frame is

delayed to avoid dropping frames that affect the quality of all subsequent frames in the current Group of Pictures because they rely on the correctly filtered frame for reference. Based on the key frame interval, reduced quality can last a few seconds with errors compounding over time until the next intra-frame. This implies that in-loop filters should be processed in a considerably short time to realize the correct real-time decoding.

With a postfilter, in the former case, all the filtering can be performed as a completely independent process and does not exhibit any temporal dependencies, and thus it possible to use separate processing units for each frame in parallel. In the latter case, if processing cannot be completed in time, then the current picture is left partially unimproved although other pictures do not suffer any degradation and can be accurately filtered provided that sufficient processing capacity is available for them. In practice, a few frames that are more expensive to decode can use excessive processing power to allow filtering although other frames can run the filter as intended.

The proposed filter operates independently on each channel. Hence, only luma can be processed if the computing power is limited such as in the case of embedded or mobile devices. This offers most of the gains for a lower cost. The neural network-based architecture allows changes in the number of filters and allows even the less capable devices to improve to a certain extent. Scalability is important because it allows a larger adoption rate and allows implementers to determine the efficiency/cost ratio that optimally works for them. For example, the proposed filter uses eight feature layers although four layers still exhibit improvement for half of the processing cost. Over eight feature layers exhibit little improvement and prevent real-time processing on most hardware, and thus eight feature layers correspond to a good compromise. As opposed to aiming to obtain a higher efficiency than that of image reconstruction state of the art approaches, the aim of the proposed filter is to be practical and usable in several devices.

The improvement of the in-loop filters is significant because they use additional information for increased efficiency, and thus the aim of the proposed filter is to supplement them and not to replace them. Previous studies[5] on a filter that used the same information as the in-loop filter indicated that even in the All Intra case (wherein the improvement from a frame helping the following frames does not exist), the use of both filters exhibited optimal results. The improvement for filtered blocks was significant. However, in other modes, the number of filtered blocks is significantly lower leading to less overall improvement. Filtering the whole picture was more effective than only specific blocks. Furthermore, it removes the need for preserving the encoding information because a single parameter (QP) is used for the whole frame.

## 2. Convolutional Neural Networks in Image Filtering

The proposed filter uses Convolutional Neural Networks (CNN) to perform filtering. Specifically, CNN is the leading method in image recognition competitions. CNN were also used in image filtering before to remove Gaussian noise from images[6),7)], pulse noise[8)]or for upscaling[9)–12)].

In Jain et al.[6)], the CNN-based method achieved extremely impressive restoration of the original picture. However, the deep structure with up to six layers makes it considerably slow. This type of a deep architecture is still too complex for the speed required by real-time video filtering that our filter aims to achieve. Zhang et al.[7)]improved the aforementioned method although it still required 60 ms to process an image with a GPU. It is possible to use this for video processing although it requires expensive hardware.

Du et al.[8)]offered impressive reconstruction of images degraded with pulse noise. The Median Filter already offers good reconstruction because the degradation is extreme although it does not affect every pixel: most of the information is still easily extracted. The proposed filter extracts the information more efficiently than the median filter and avoids blurring the image.

Jampani et al.[9)]use bilateral filters inside CNN. The filter used sparse and high dimensional convolutions. It exhibited significant results for both color upsampling and depth upsampling.

Shi et al.[10)]used a learned filter to perform upscaling and achieves real-time processing of 1080p videos on a GPU. It used deconvolution to increase the size of the input.

Lai et al.[11)]obtained better results than Shi et al.[10)] by using a more complex architecture and maintaining upscaling with deconvolutions while also adding a feature extraction branch wherein results were then merged into the deconvolution result.

Dong et al.[12)]discussed an interesting network that can perform real-time processing (above 40 fps) on a CPU although this was for images that were 240×240 pixels,

and it was not suitable for high definition content because the cost increased with the number of pixels.

However, none of the aforementioned approaches are used for image degradation specifically resulting from encoding. Artifacts from encoding are considerably specific, and thus a filter trained specifically for removing encoding noise is necessary. Super resolution and other upscaling methods use low-resolution pictures that are simply downsampled occasionally with Gaussian noise although nothing similar to the loss results from encoding. The computing cost of these methods is also excessively high for the real-time processing of HD content without using expensive hardware.

However, a current few studies focused on alternative in-loop filters or post-processing methods for HEVC [13)–15)]. The issue with these methods is there are limitations despite a few improvements.

Li et al.[13)]detailed an overall reduction in bit rate of 1.6% although it used an extremely complex network of 20 layers. It used different models for different values of QP, which required exporting some metadata while decoding or integrating directly with the decoder. Although the need decreased while using constant QP, it filtered only the luma component and appeared to use the same sequences in training and testing. While they were encoded differently, it is likely that the results improved because of the large similarities between the training and testing sets.

The study by Park et al.[14)]was based on super resolution CNN (SRCNN) trained on degraded images that were reconstructed via the decoder. It used different models for variable values of QP to provide adequate filtering for the images. It exhibited an impressive reduction of 4.8% for All Intra mode and between 1.6 and 2.8% with Low Delay P and All Intra. The computing time exceeded[13)]with 1.2 s per frame on 832×480 sequences, which corresponded to one frame per 5.2 s for FullHD. Furthermore, it exhibited the same limitations as[13)]because it used the same training and testing sequences.

Dai et al.[15)]revealed a 4.6% reduction in bitrate over many sequences. The results for All Intra are interesting, although of little use in real applications that use inter-picture prediction and where in-loop filtering offers large gains. In-loop filters are disabled, and thus running the filter only during post-processing is likely to lead to extremely low to negative improvements on the more realistic Random Access configuration. It is also considerably slow and takes almost one second per frame on a high-end CPU or half that on a GPU for a resolution of only 176×144. With respect to FullHD (1080p) video, which is now considerably common, it takes 80 times as much time, making it considerably unpractical. Even additional processing power makes real-time processing unlikely. Hence, although it offers improvement over Park et al.[14)]in terms of its general nature that allows it to work on all three components instead of only luma, it is impractical for actual use due to its speed.

## 3. Proposed Method

### 3.1 Input and output blocks

The proposed filter receives 12×12 pixels blocks as input. As significant part of the noise introduced during encoding is situated around block boundaries (and thus the need and improvement is offered by the deblocking filter), each block is centered over an 8×8 grid. For example, the first filtered block is centered around (8, 8), the second is centered around (16, 8), and the series continues. **Figure 1** shows how the blocks are aligned. The position is selected to place the HEVC 8×8 block boundaries immediately in the middle on the input block and offer sufficient context information for filtering.

The input blocks overlap, and this is extremely helpful to avoid potential blocking artifacts at the edges of the 8×8 output block because smaller 4×4 blocks exhibit edges overlapping the 8×8 block edges. The figure shows only two input blocks to avoid confusion because the overlapping makes it difficult to specify where each block ends. It is important to use 12 pixels for the blocks to provide information on neighboring pixels from both sides. We use the same block size as the output to significantly reduce the efficiency, and thus 12 pixels is a good compromise between paucity of information and excessive information that results in a significantly higher process-

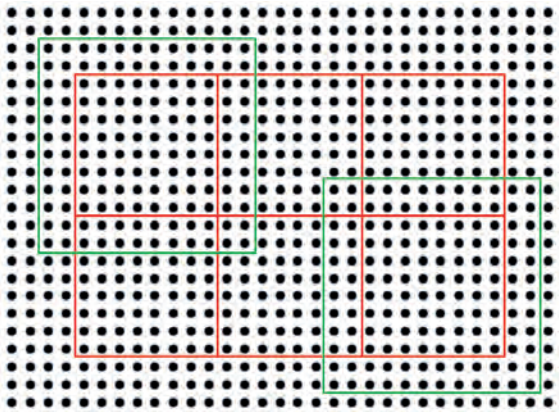


**Fig. 1** Representation of filtered blocks (in red) and input blocks (in green)

ing time.

With respect to the output, the filter affects only the 8×8 closest to the center. The output blocks do not overlap, which is extremely important because it allows parallel processing. It also covers the complete picture and potentially affects every pixel while the existing in-loop filters mainly affect the two pixels closest to the block boundary while other pixels are significantly less affected. The output targets the residual, namely the difference between the original image and reconstructed image because it makes training easier. The input is also normalized by removing the average value of the block for the same reasons.

The filter also uses the same architecture for chroma even while using chroma subsampling. Smaller blocks, such as the ones used in the HEVC deblocking filter, did not show good results.

The filter requires a larger block for input than output, and thus the four pixels closest to the edge are excluded from the filter. Although this can limit the efficiency of the filter and especially for chroma while using 4:2:0 subsampling, the edges are typically less important for visual quality because attention is focused on the center of the image. Furthermore, the four pixels that are left unfiltered represent a small part of the image when dealing with high definition content, thereby making this a minor issue.

### 3.2 Filtering layers

With the number of blocks in a regular image, it is impossible to perform real-time or even acceptable fast filtering with methods via twenty layers of convolution as reported by Park et al.[14]. Thus, the proposed filter only uses two filtering layers and a normalization layer. The normalization layer computes the average value of the input blocks and subtracts it from the original values.

The proposed filter uses a 7×7 convolution for the convolutional layer. The convolution also exploits the correlation along the axis of the block boundary to obtain improved results when compared to that of the in-loop filter. The in-loop filter uses 4 pixels from each side in the computation to offer efficiency exceeding it, and thus the use of several pixels is important in the convolution. Different designs are tested with deeper layers such as 5×5 followed by 3×3. However, optimal results are obtained with a 7×7 convolution.

The next layer corresponds to a rectified linear unit (ReLU) layer, which acts as a thresholding function that is defined as follows:

$$f(x) = \max(0, x). \tag{1}$$

The final layer is a fully connected layer that takes the output of the previous ReLU layer and outputs the 8×8 corrected values. In contrast to the in-loop filter, the proposed filter affects eight pixels on each line as opposed to only up to six. The use of a symmetric convolution allows only one pass of the filter as opposed to in-loop vertical and horizontal filtering that creates a temporal dependency because it is necessary to perform one prior to the other to obtain the expected results.

In order to determine the number of filtering layers, experiments are conducted with varying amounts. Increases in the amount improve the filtering in some cases although it also increases the number of operations and number of parameters, thereby making learning more difficult and longer. This typically leads to over learning, which can be solved with a larger training dataset. In the study, eight feature layers are used. Using a lower number of features significantly decreases the improvement. However, increasing to 16 leads to an extremely small improvement and even worse results for a few clips. Increasing the training dataset also increases the training time, which for the retained design corresponds to a week for all the different types of configurations.

A partial representation of the filter (normalization and ReLU layers not shown) is given in **Fig. 2**.

### 3.3 Training configuration

Training and testing is performed using Matlab and the MatConvNet library[16], which provides implementation of CNN for Matlab and training scripts. A common metric to evaluate filters objectively corresponds to Peak Signal to Noise Ratio (PSNR). Hence, it is logical to train the filter to maximize PSNR, and thus the loss function is set to Mean Square Error. This is a good loss function because it is easy to invert and requires an extremely less additional computations during back-propagation.

All the sequences for training and testing used the same encoding settings. Given the memory requirements and training time resulting from the use of an excessive number of frames, only thirty frames per video clip are used during training for luma and sixty for chroma. The training is performed several times over each encoding configuration.

The training and testing is obtained by using the original (before encoding) frame as an objective and the en-

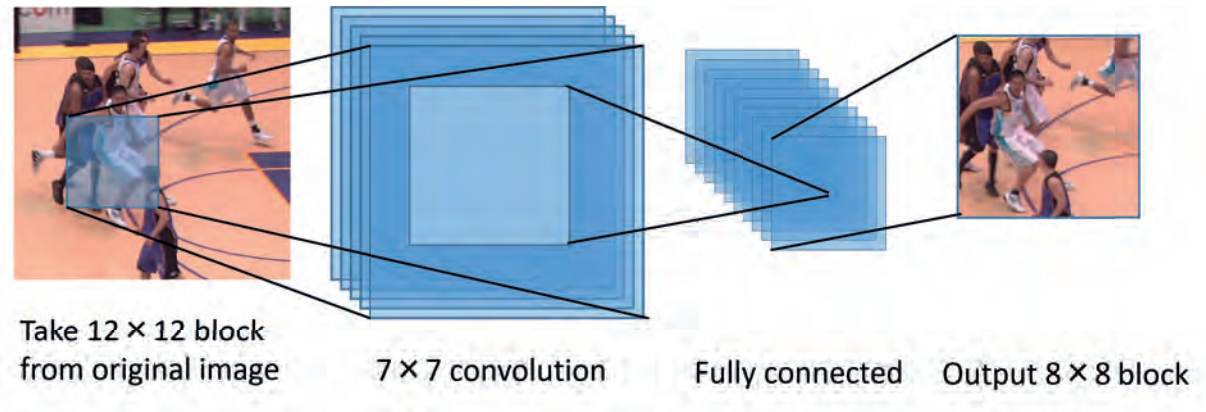



**Fig. 2** Architecture of the proposed filter

coded frame as an input.

## 4. Computational Complexity and Parallelism

While the filtering performance was tested in the experiments, the processing was conducted on a processor and without optimizations. This part discusses the parallelization possibilities, theoretical computing cost of the filter, hardware implementation requirements, and current implementation performance.

### 4.1 Parallelism in image filtering

Most image filters allow some parallelization and especially in recent years because processing on a single core is impossible. The deblocking filter in AVC was difficult to parallelize, and thus a parallelizable deblocking filter was introduced instead in HEVC. Thus, most filters consider parallelization to allow fast processing.

CNN are typically parallel in their processing. However, in contrast to the AVC deblocking filter, a few filters require the blocks to be processed in order, and this limits the parallelization on a per block basis. Large networks can use all resources for a single block although small networks with fewer operations cannot be significantly split, and thus the ability to compute several blocks is simultaneously valuable.

The use of a postfilter also exhibits an advantage for parallelization, as mentioned in the introduction. The ability to process frames completely independently significantly improves performance while processing on multiple devices with slow intercommunication such as multiple physical CPU or GPU.

### 4.2 Computational cost of the filter

In the current test case, all videos correspond to FullHD (1920×1080 pixels) with 4:2:0 chroma subsampling. The filter processes one block per eight pixels on both vertical and horizontal direction, and thus this leads to approximately 32,000 blocks.

The cost for each block is easy to compute by assuming that the ReLU layer is trivial to compute (because it performs a simple thresholding operation) and that most of the required processing lays in the two convolutional layers. Each multiply-add required in the convolution is counted as a single operation because this is a commonly implemented instruction. The first layer is a 7×7 convolution, and this leads to the following cost:

$$\begin{aligned} N_{op_1} &= 7 \times 7 \times (12 + 1 - 7) \times (12 + 1 - 7) \times F \\ &= 1764F. \end{aligned} \tag{2}$$

Where $F$ corresponds to the number of feature layers.

With respect to the fully connected layer, each value of the 6×6×$F$ output of the previous layer is connected to the 8×8 output. This leads to the following expression:

$$N_{op_2} = 6 \times 6 \times 8 \times 8 \times F = 2304F. \tag{3}$$

The summation of all the terms leads to the following expression:

$$N_{op_t} = 4068F. \tag{4}$$

Given the current design of eight feature layers, this leads to 32,544 operations. This indicates that a frame requires approximately one billion operations just for luma. While adding chroma, this corresponds to 1.5 billion operations. With respect to real-time processing and assuming 30 frames per second (fps), this requires 45 GFLOPS (floating point operations per second). Although the aforementioned figure still looked daunting a few years ago, consumer GPUs currently surpass 8,000 GFLOPS. Real-time processing is possible even with consumer-grade hardware. However, with respect to a CPU, this approximately corresponds to the limit of a recent multi core

CPU, and thus further increases in the cost increase the difficulty of real-time processing. A good GPU can process even higher definition video without significant issues. Further testing also investigates half-precision or fixed point computing for even faster computing and reduced memory footprint, thereby allowing the proposed filter to run on smaller specialized hardware such as FPGAs where the reduction in memory requirements for the parameters is encouraged. Although the network is already small, any reduction in the memory footprint results in significant savings while considering mass production of integrated circuits. For example, hardware vendors also pushed for limitations on the number of reference frames in AVC and HEVC due to memory requirements.

### 4.3 Hardware implementation

When porting an algorithm to a GPU, a common problem that appears corresponds to the limits on parallelization. However, the proposed filter does not exhibit inherent limitation for parallelization. Each block used as an input corresponds to a block before any filtering is applied and each pixel is only filtered once. There are no possible race conditions provided a buffer is maintained for storing the original picture. It is possible to divide the image into the maximum number of parts given the number of cores available to process each part independently without any issues. There is no issue if it is necessary to separate between two GPUs that cannot share memory easily, and each GPU can process half the image. Only a few redundant pixels are necessary because the input exceeds the output and overlaps slightly. If latency is not an issue, it is possible for different GPUs to process different images and completely avoid redundancies.

With respect to embedded FPGA or specialized circuits, the 7×7 convolution can be implemented more efficiently and with less waste than in CPU or GPU (the size and data are not aligned with SIMD instructions). There are smaller gains for the fully connected layer although it is not necessary to support anything other than arithmetic operations and this frees several transistors, thereby achieving the required performance at a significantly lower cost. Apart from adapting to the image size and QP value (once per frame), there are no branches, and thus an extremely deep and wide architecture is possible.

### 4.4 Current implementation

In practice, the current implementation using Matlab and the MatConvNet library[16]runs only at 2.5 fps per thread on an Intel Xeon E5-2630 (2.6 GHz) CPU and does not consider the I/O operations on the raw video files. However, it is possible to significantly improve the performance while using Intel Math Kernel Library (MKL)[17] as opposed to Matlab's implementation of the Basic Linear Algebra Subprograms (BLAS). The biggest part of the run time is spent on the convolutional layer. The fully connected layer requires more operations although the layer can be computed extremely efficiently with a single matrix–matrix multiply (gemm) simultaneously for all blocks. Conversely, it is significantly more difficult to optimize the 7×7 convolution in this manner. Even while using Intel MKL, a frame still takes approximately 0.1 s to process. In this case, the algorithm processes block per block. Thus, with some parallelization, the time per image can be reduced to an average of 16 ms on the 6 cores of the CPU. The time associated with the fully connected layer corresponds to 20 ms for Matlab's BLAS and 15 ms for MKL. Thus, the time to process an image is 31 ms, which corresponds to 32 fps. However, the full processing power of the CPU cannot be realistically used for filtering in practical uses, and thus a few further improvements or optimizations are useful to ensure reliable real-time processing.

MatConvNet GPU implementation of the convolution showed underwhelming results for the 7×7 convolution, making it slower than the CPU implementation. It is also slowed by the overhead of sending the data from CPU memory to GPU memory and back. However, a better implementation causes the decoded video to be already in the GPU memory in the first place and eliminates the need to return to the CPU because it is only displayed on the screen. Although this can optimize the GPU processing, this is not considered in the present study.

## 5. Testing Methodology

### 5.1 Encoding configuration

In order to obtain results that reflect practical use of the filter, each sequence was encoded with a x265 HEVC encoder[18], which is an encoder that is available free of charge under the GNU GPL license version 2.

Two different encoding modes were tested to evaluate performance, namely All Intra (AI) and Random Access (RA) modes. While they slightly differ from the reference encoder, the settings were kept as close as possible.

The encoding settings for both the modes are shown in **Table 1**. Every loop filter in HEVC was kept as-is, and thus the comparison baseline used the deblocking and SAO filters, and the proposed post-processing filter was further applied.

### 5.2 Dataset

We used video sequences from the ITE/ARIB Hi-Vision Test Sequence 2nd Edition[19]. Ten clips from ten different video sequences were used. The full list with extracted frame numbers is shown in **Table 2**. Most videos still use an 8-bit depth and 4:2:0 chroma subsampling. Thus, the source videos were cropped to 8 bits and half of the chroma values were discarded from the 10-bit 4:2:2 source.

Training requires enough different sequences for usability on clips outside the training clips. It is not extremely useful to present results for only a couple of sequences, and thus we used cross-validation. Most clips were used for training. The remaining clips were then tested and the training set was switched. We tested videos in subsets of three and two by repeating the training twice on seven video sequences and eight video sequences such that every video sequence was tested.

## 6. Results

This section details the results of the proposed filter using PSNR measurements 2 Bit Rate (BD-BR)[20]where a negative value indicates the percentage of bitrate saved by the proposed method. In order to assess the overall performance and not only the performance on the Y, U, or V channels, a combined PSNR using luma and chroma was computed as follows:

$$\mathrm{PSNR_{YUV}} = (6\times\mathrm{PSNR_Y}+\mathrm{PSNR_U}+\mathrm{PSNR_V})/8. \quad (5)$$

This is the same expression as that used in a popular comparison of video coding standards[21]. With respect to each encoding mode and video, the BD-BR for each channel and overall improvement using $\mathrm{PSNR_{YUV}}$ were computed. The BD-BR using YUV was computed from $\mathrm{PSNR_{YUV}}$, and thus the values can slightly differ than that while directly applying the formula to Y, U and V BD-BR values.

The BD-BR values are shown in **Table 3** and **Table 4**, and PSNR values are shown in **Table 5** and **Table 6**. The detailed consideration of each mode is shown below. With respect to the aggregate results using average and median, the BD-BR values before rounding are used. Specifically, 10 clips are considered, and thus the median value is the average between the 5th and 6th best clips.

**Table 1** Encoding settings

| | |
|---|---|
| Codec | x265 |
| Version | HEVC Encoder version 2.1+69-c97c64ab8b8e |
| Encoding settings (AI) | –preset veryslow –tune=psnr –no-wpp –rd=6 –keyint 1 –qp $qp+3 |
| Encoding settings (RA) | –preset veryslow –tune=psnr –no-wpp –rd=6 –keyint 32 –min-keyint 32 –bframes 16 –qp $qp |

**Table 2** Video Sequences

| Sequence Number | Name | Extracted Frames | Test Subset |
|---|---|---|---|
| 201 | Ginko trees | 100 - 159 | 1 |
| 202 | Truck train | 100 - 159 | 1 |
| 203 | Cosmos flowers | 100 - 159 | 1 |
| 209 | Fountain (dolly) | 100 - 159 | 2 |
| 214 | Basketball | 100 - 159 | 2 |
| 215 | Twilight scene (zoom out) | 100 - 159 | 2 |
| 218 | Horse racing (dirt) | 100 - 159 | 3 |
| 251 | Rotating disk 59.94P | 100 - 159 | 3 |
| 259 | Colorful world A | 100 - 159 | 4 |
| 265 | Fountain (chromakey) | 100 - 159 | 4 |

### 6.1 All intra mode

The following results were obtained while encoding the movie clips with the All Intra configuration. Table 3

**Table 3** Filter results for the All Intra mode

| Sequence | Y | U | V | YUV |
|---|---|---|---|---|
| s201 (Ginko Tree) | -2.6% | -2.7% | -0.8% | -2.4% |
| s202 (Truck train) | -1.6% | -2.7% | -2.8% | -1.7% |
| s203 (Cosmos flowers) | -2.9% | -1.4% | -0.5% | -2.6% |
| s209 (Fountain) | -1.4% | -1.7% | -0.7% | -1.4% |
| s214 (Basketball) | -1.9% | -5.7% | -3.4% | -2.5% |
| s215 (Twilight scene) | 3.0% | 16.0% | 4.2% | 4.2% |
| s218 (Horse racing) | -1.1% | -1.2% | -1.5% | -1.1% |
| s251 (Rotating disk) | -4.4% | -1.8% | -1.0% | -3.9% |
| s259 (Colorful World) | -3.5% | -2.9% | -1.2% | -3.2% |
| s265 (Fountain) | -0.9% | -3.2% | -2.9% | -1.0% |
| Median | -1.7% | -2.2% | -1.1% | -2.1% |
| Average | -1.7% | -0.7% | -1.1% | -1.6% |

**Table 4** Filter results for Random Access mode

| Sequence | Y | U | V | YUV |
|---|---|---|---|---|
| s201 (Ginko Tree) | -2.1% | -2.0% | -1.4% | -2.0% |
| s202 (Truck train) | -0.8% | -0.3% | -1.1% | -0.8% |
| s203 (Cosmos flowers) | -4.7% | -0.9% | -0.3% | -4.1% |
| s209 (Fountain) | -1.5% | -0.8% | -0.3% | -1.4% |
| s214 (Basketball) | -1.8% | -2.7% | -1.5% | -1.9% |
| s215 (Twilight scene) | 3.2% | 17.2% | 6.8% | 5.1% |
| s218 (Horse racing) | -1.2% | 0.1% | -0.1% | -1.0% |
| s251 (Rotating disk) | -4.9% | -0.3% | -0.1% | -4.0% |
| s259 (Colorful World) | -4.4% | -0.2% | 1.4% | -3.1% |
| s265 (Fountain) | -1.2% | -1.4% | -1.0% | -1.2% |
| Median | -1.6% | -0.5% | -0.3% | -1.6% |
| Average | -1.9% | 0.9% | 0.2% | -1.4% |

**Table 5** PSNR values for two different QP values on each sequence for the proposed filter and reference HEVC (All Intra)

| Sequence | QP22 | | | | | | QP32 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Reference | | | Proposed Method | | | Reference | | | Proposed Method | | |
| | Y | U | V | Y | U | V | Y | U | V | Y | U | V |
| Ginko Tree | 40.69 | 40.92 | 40.28 | **40.84** | **40.97** | **40.30** | 31.74 | 35.32 | 33.96 | **31.97** | **35.46** | **34.00** |
| Truck train | 41.23 | 41.55 | 42.63 | **41.37** | **41.56** | **42.65** | 33.09 | 38.72 | 39.43 | **33.16** | **38.77** | **39.49** |
| Cosmos flowers | 40.70 | 40.27 | **40.70** | **40.93** | **40.28** | 40.69 | 32.65 | 35.37 | 36.01 | **32.87** | **35.44** | **36.04** |
| Fountain | 41.25 | 44.61 | 43.87 | **41.33** | **44.61** | **43.87** | 35.99 | 43.50 | 40.80 | **36.03** | **43.52** | **40.81** |
| Basketball | 42.17 | 44.87 | 44.17 | **42.22** | **44.99** | **44.21** | 37.43 | 39.83 | 38.86 | **37.54** | **40.16** | **39.06** |
| Twilight scene | **41.31** | **41.85** | **41.55** | 41.24 | 41.76 | 41.52 | **37.02** | **39.11** | **37.37** | 36.91 | 38.63 | 37.18 |
| Horse racing | 42.25 | 43.84 | 46.19 | **42.27** | **43.85** | **46.20** | 37.69 | 39.65 | 41.64 | **37.75** | **39.71** | **41.71** |
| Rotating disk | 41.49 | 42.50 | 43.32 | **41.68** | **42.52** | **43.34** | 35.62 | 39.59 | 38.57 | **35.91** | **39.65** | **38.62** |
| Colorful World | 40.84 | 40.96 | 41.14 | **41.04** | **41.01** | **41.17** | 33.92 | 35.51 | 34.58 | **34.14** | **35.68** | **34.66** |
| Fountain | 41.42 | 43.43 | 44.72 | **41.49** | **43.44** | **44.73** | 36.18 | 42.64 | 42.37 | **36.19** | **42.67** | **42.44** |
| Average | 41.33 | 42.48 | 42.86 | **41.44** | **42.50** | **42.87** | 35.13 | 38.93 | 38.36 | **35.25** | **38.97** | **38.40** |

**Table 6** PSNR values for two different QP values on each sequence for the proposed filter and reference HEVC (Random Access)

| Sequence | QP22 | | | | | | QP32 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Reference | | | Proposed Method | | | Reference | | | Proposed Method | | |
| | Y | U | V | Y | U | V | Y | U | V | Y | U | V |
| Ginko Tree | 38.91 | 41.09 | 41.00 | **38.95** | **41.09** | **41.06** | 32.71 | 36.28 | 34.88 | **32.85** | **36.36** | **34.92** |
| Truck train | 39.77 | 41.98 | 43.18 | **39.83** | **41.98** | **43.20** | 34.70 | 39.20 | 39.99 | **34.72** | **39.20** | **40.00** |
| Cosmos flowers | 39.22 | 39.80 | 40.57 | **39.46** | **39.80** | **40.58** | 32.04 | 35.65 | 36.22 | **32.26** | **35.67** | **36.23** |
| Fountain | 40.49 | 44.50 | 43.65 | **40.56** | **44.50** | **43.65** | 35.74 | 43.68 | 40.68 | **35.77** | **43.68** | **40.68** |
| Basketball | 41.41 | 44.87 | 43.92 | **41.46** | **44.89** | **43.94** | 37.60 | 40.65 | 39.47 | **37.65** | **40.75** | **39.53** |
| Twilight scene | **40.27** | **41.51** | **41.23** | 40.19 | 41.40 | 41.14 | **36.89** | **38.73** | **37.20** | 36.82 | 38.43 | 37.06 |
| Horse racing | 41.51 | 43.91 | 46.69 | **41.53** | **43.91** | **46.70** | 37.89 | **40.48** | 42.52 | **37.92** | 40.47 | **42.52** |
| Rotating disk | 40.54 | **42.74** | 43.91 | **40.68** | 42.74 | **43.92** | 36.36 | 40.25 | **39.53** | **36.52** | **40.25** | 39.51 |
| Colorful World | 39.57 | **41.02** | **41.78** | **39.68** | 41.00 | 41.76 | 35.21 | 36.55 | **35.81** | **35.38** | **36.58** | 35.74 |
| Fountain | 40.60 | **43.42** | 44.65 | **40.66** | 43.42 | **44.66** | 35.60 | 42.63 | 42.74 | **35.66** | **42.63** | **42.75** |
| Average | 40.23 | **42.48** | 43.06 | **40.30** | 42.47 | **43.06** | 35.47 | **39.41** | **38.90** | **35.56** | 39.40 | 38.90 |

shows the improvement in the proposed filter for each video and the results were averaged over each video. The actual PSNR values for two values of QP are shown in Table 5.

With the exception of a video (Twilight scene (zoom out)), each video exhibited an improvement and especially for luma with up to 4.4% reduction in the bitrate for a video. Chroma exhibited less gain on average. However, outside the outlier, it was always improved by at least half a percent and up to 5.7% and exhibited good performance while considering the median.

The PSNR values indicated that the filter is most efficient for low bitrate clips and that an improvement over high quality clips is more difficult.

The poor performance on Twilight scene can be explained by looking at the other videos in the current set as follows: this was the only video shot at night with low light and significant grain due to the lack of light. The characteristics of the clip (significantly differ from the others) explained why the filter failed to improve it and made it worse. This indicated that it is important that the training set was comprised of movie clips with characteristics similar to that of the target clips.

While considering $\mathrm{PSNR}_{\mathrm{YUV}}$, the improvement corresponds to a reduction in bitrate by approximately 1.7% on average.

**Figure 3** shows the improvement in the filter on the Basketball and Rotating disk sequence. The large edges due to the low bitrate encoding were smoothened and the image was less degraded. In Fig. 3, top row is from the Basketball clip, bottom from the Rotating Disk clip, the left column shows the image before using the filter, and the right column shows the image after using the filter.

### 6.2 Random access mode

The following results were obtained while encoding the movie clips with the Random Access configuration. Table 4 shows the improvement in the proposed filter for each video. With the exception of a video (Twilight scene(zoom out)), which exhibited same problems as the

(a) without the application filter (b) with the application of the filter

**Fig. 3** Comparison of images

All Intra mode, each video improved with the proposed filter with an overall BD-BR improvement corresponding to -1.4% by using $PSNR_{YUV}$. Except for a few exceptions, luma exhibited maximum improvement with an average of -1.9% and up to -4.9%.

Chroma was unfortunately improved to a lesser extent by the filter with only up to -2.7% for U and -1.5% for V and very often merely corresponded to minor improvements, which resulted in an overall negative mean change due to the outlier and a -0.3% reduction in bit rate while considering the median. The actual PSNR values as shown in Table 6 indicate the lack of improvement on chroma with higher QP values, and thus potentially requiring additional training for usability.

While the Random Access mode exhibited similar and occasionally better performance on luma when compared to the All Intra mode, performance on chroma was still poor for most clips. Only a clip exhibited an improvement over 2% on chroma and several others also exhibited some degradation. Different types of frames can make the learning more difficult. Furthermore, as chroma was typically subject to more severe degradations and especially in B frames, the recovery of information was typically more difficult than that for luma.

In order to solve the issue, a solution was assumed to involve the use of a different filter based on frame type. However, this required significantly more encoding and source material to exhibit sufficient I and P frames to form a sufficiently large training set.

### 6.3 Comparison with other methods

Other methods did not use the same datasets, and thus comparison proves difficult.

With respect to the All Intra mode (which was present in the two methods[14),15)]), the results can be compared albeit with a few misgivings. Only Dai et al.[15)]discussed results for chroma components, and thus only the luma component can be compared. In this case, the proposed method revealed inferior results to Dai et al.[15)]and Park et al.[14)], who show a reduction of approximately 4.5% in bitrate (with the latter using the same dataset for training and testing). However, the proposed filter is significantly faster than both methods.

With respect to the random access case, results reported by Li et al.[13)]and Park et al.[14)]can be compared albeit with a few caveats. In the former, the intra-frame distance is fixed to 150 frames (–keyint 150)[22)]while for the latter, it corresponds to 16 frames. The current study used 32 frames because it offers a good compromise between jumping in the middle of the stream and encoding efficiency. Furthermore, 150 frames is not typically considered random access and is most used for archiving data as the efficiency is more important than the ease of access. Hence, Park et al. indicated the highest efficiency of the three methods with a 2.6% reduction in bitrate. It is also the only in-loop filter solution of the three and required the encoder and the decoder to be modified and supplied with the network parameters. The proposed filter corresponded to the second best filter with a 1.9% reduction in bitrate, and the filter proposed by Li et al. was last with 1.6%. Both methods also exhibited an additional advantage because they use the same training and testing datasets, which means their true efficiency for different testing data is likely to be lower.

### 6.4 Processing speed

Given that other methods did not use as many optimizations to extract the maximum performance from their computers, the use of the processing speed from the naïve implementation in MatConvNet appeared as

the fairest option, especially since it was also used in Park et al.[14]. The proposed filter achieved a processing time of 0.4 s for a FullHD(1920×1080) 4:2:0 subsampled video. Park et al.[14]revealed a processing time of 1.2 s on a 832×480 sequence (only luma). Dai et al.[15]takes 0.98 s on CPU and 0.45 s on a GPU for a QCIF (176×144), 4:2:0 subsampling video.

In order to make this comparable, the times were extrapolated assuming the processing scales linearly with the number of pixels. This leads to 6.2 s per frame for Park et al. and 36 s for Dai et al. (given GPU time). This makes the proposed filter at least an order of magnitude faster than other methods.

## 7. Conclusion

In the study, a post-processing filter for HEVC was introduced. The filter, based on CNN, uses only two convolutional layers to preserve processing speed.

The results indicated that the proposed filter significantly increased the quality of the output (based on the PSNR metric). Based on the video, gains of up to 5% reduction in bit rate on luma were possible. One video exhibited negative results although a better training set can potentially fix this.

The filter exhibits good average performance for both All Intra and Random Access modes with an overall 1.5% reduction in bitrate. While other modes were not tested, similar values obtained on both modes suggest that they result in similar values. For example, Park et al.[14]indicate that the results for Low Delay P are similar to those for the Random Access mode.

The filter failed to achieve the improvements in the state of the art post-processing methods although it achieved a moderate improvement for a processing time that was ten to a hundred times lower. It was also considerably simple to run because it only required the frames and QP value, which can be passed as a global parameter to select the appropriate model. It allows the proposed post-processing filter to be embedded inside a decoder or television without excessive additional cost. Other existing filters are simply not worth the additional processing cost.

Future studies should investigate videos of different resolutions and from other sources. Results on 10-bit videos using 4:2:2 chroma subsampling should also be examined. In order to aid with the processing cost, it is important to focus on separable filters since they allow faster and easier implementation of computation for the convolutions.

## Acknowledgment

The study was partially supported by JSPS KAKENHI Grand Numbers 16H02841 and 18K19772.

## References

1) B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand: "High Efficiency Video Coding (HEVC) Text Specification Draft 9", ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) (2012).
2) P. List, A. Joch, J. Lainema, G. Bjøntegaard, M. Karczewicz: "Adaptive Deblocking Filter," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp.614–619 (2003).
3) A. Norkin, G. Bjøntegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, G. Van der Auwera: "HEVC Deblocking Filter", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp.1746–1754 (2012).
4) C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsa, C.-W. Hsu, S.-M. Lei, J.-H. Park, W.-J. Han: "Sample Adaptive Offset in the HEVC Standard", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp.1755–1764 (2012).
5) A. Chauvet, T. Miyazaki, Y. Sugaya, S. Omachi: "Adaptive Post Filter for Reducing Block Artifacts in High Efficiency Video Coding", Proc. of the 2016 International Conference on Multimedia Systems and Signal Processing (ICMSSP), pp.22–25 (2016).
6) V. Jain, S. Seung: "Natural Image Denoising with Convolutional Networks", Proc. of the Advances in Neural Information Processing Systems 21, pp.769–776 (2009).
7) K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang: "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising", IEEE Trans. on Image Processing, Vol. 26, No. 7, pp.3142–3155 (2017).
8) M. J. Du, W. Y. Wang, A. Liu: "Study for Image Optimal Filter Based on Neural Network", Proc. of the 2016 International Conference on Audio, Language and Image Processing (ICALIP), pp.391–394 (2016).
9) V. Jampani, M. Kiefel, P. V. Gehler: "Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks", Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4452–4461 (2016).
10) W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, Z. Wang: "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network", Proc. of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1874–1883 (2016).
11) W.-S. Lai, J.-B. Huang, N. Ahuja, M.-H. Yang: "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution", Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5835–5843 (2017).
12) C. Dong, C. Change Loy, X. Tang: "Accelerating the Super-Resolution Convolutional Neural Network", Proc. of the European Conference on Computer Vision, pp.391–407 (2016).
13) C. Li, S. Li, R. Xie, W. Zhang: "CNN based post-processing to improve HEVC", Proc. of the IEEE International Conference on Image Processing (ICIP), pp.4577–4580 (2017).
14) W.-S. Park, M. Kim: "CNN-based in-loop filtering for coding efficiency improvement", Proc. of the 2016 IEEE 12th Im-

age, Video, and Multidimensional Signal Processing Workshop (IVMSP), pp.1–5 (2016).
15) Y. Dai, D. Liu, F. Wu: "A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding", Proc. of the International Conference on Multimedia Modeling, pp.28–39 (2016)
16) A. Vedaldi and K. Lenc: "MatConvNet: CNNs for MATLAB", Proc. of the ACM International Conference on Multimedia, pp.689–692 (2016).
17) Intel, Math Kernel Library, http://developer.intel.com/software/ (2017).
18) MulticoreWare, Inc., x265 HEVC Encoder, http://x265.org/ (2016).
19) Institute of Image Information and Television Engineers, ITE/ARIB Hi-Vision Test Sequence 2nd Edition, http://www.ite.or.jp/contents/chart/emanual.pdf (2009).
20) G. Bjøntegaard: "Calculation of Average PSNR Differences between RD-curves", ITU-T Video Coding Experts Group (VCEG) 13th Meeting (2001).
21) J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, T. Wiegand: "Comparison of the Coding Efficiency of Video Coding Standards–Including High Efficiency Video Coding (HEVC)", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp.1669–1684 (2012).
22) Grand Challenge ICIP 2017, Grand Challenge on the Use Of Image Restoration For Video Coding Efficiency Improvement, https://storage.googleapis.com/icip-2017/index.html (2017).

(Received May 31, 2018)
(Revised March 5, 2019)

**Antoine CHAUVET**
He received M.E. degrees from Tohoku University, Japan, and INSA Lyon, France in 2017. He is currently a doctoral student in Tohoku University, Japan. His research interests include video filtering, video coding and parallel processing.

**Tomo Miyazaki**
He received B.E. degree from Department of Informatics, Faculty of Engineering, Yamagata University in 2006. He received M.E. and Ph.D. degrees from Graduate School of Engineering, Tohoku University in 2008 and 2011, respectively. He joined Hitachi Ltd in 2011, and has worked at the Graduate School of Engineering, Tohoku University from 2013 to 2014 as researcher. Since 2015 he has been an Assistant Professor. His research interests include pattern recognition and image processing. Dr. Miyazaki is a member of the Institute of Electronics, Information and Communication Engineers.

**Yoshihiro SUGAYA**
He received B.E., M.E., and Ph.D. degrees from Tohoku University, Japan in 1995, 1997, and 2002, respectively. He is currently an associate professor at the Graduate School of Engineering, Tohoku University. His research interests include the areas of pattern recognition, image processing, parallel processing, and distributed computing. Dr. Sugaya is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and the Information Processing Society of Japan.

**Shinichiro OMACHI** (*Member*)
He received B.E., M.E., and Ph.D. degrees from Tohoku University, Japan, in 1988, 1990, and 1993, respectively. He worked as a research associate at the Education Center for Information Processing at Tohoku University from 1993 to 1996. Since 1996, he has been with the Graduate School of Engineering at Tohoku University, where he is currently a professor. From 2000 to 2001, he was a visiting associate professor at Brown University. His research interests include pattern recognition, computer vision, image processing, and image coding. He received the IAPR/ICDAR Best Paper Award in 2007, the Best Paper Method Award of the 33rd Annual Conference of the GfKl in 2010, the ICFHR Best Paper Award in 2010, and the IEICE Best Paper Award in 2012. Dr. Omachi is a member of the IEEE, Institute of Electronics, Information and Communication Engineers, Information Processing Society of Japan, among others.

IIEEJ **Paper**

# Deep Learning Based Uterus Localization and Anatomical Structure Segmentation on Fetal Ultrasound Image

Yan LI[†](*Student Member*), Rong XU[††], Artus KROHN-GRIMBERGHE[†††], Jun OHYA[†††](*Member*), Hiroyasu IWATA[†]

†Department of Modern Mechanical Engineering, Waseda University,
††Global Information and Telecommunication Institute, Waseda University,
†††Department of Business Information Systems, Paderborn University

<**Summary**> This paper proposes deep learning based methods for automatically detecting the uterus in the ultrasound (US) image and segmenting the detected uterus into anatomical structures. For accurate detection of the uterus and for segmentation of multiple fine-grained anatomical structures from the US image, we use a two-tier deep learning based algorithm: (I) localizing the bounding box of the uterus, and (II) segmenting the areas of amniotic fluid and fetal body from uterine image. To achieve (I) we design and train a convolutional neural network (CNN) based bounding box regression model, which regresses candidate positions of the uterus. Then we use the cropped uterus region as the input to a semantic segmentation approach. For (II) we apply fully convolution based architecture that segments the fetal body and amniotic fluid from fetal US images in an end-to-end, supervised learning pipeline. We use additional inner layers and intermediate supervisions to improve the segmentation accuracy and smooth out the boundaries. We experimentally evaluate our methods and demonstrate the accurate uterus detection and anatomical structure segmentation results.

**Keywords**: uterus detection, anatomical structure segmentation, ultrasound image, convolutional neural network

## 1. Introduction

Accurate and automatic measurements of the anatomical structures on the ultrasound (US) image can bring precise medical treatment to the prenatal examination and solve the shortage of manpower. To achieve this goal, we seek a way to automatically localize the uterus area, and precisely extract the border lines of the fetal body and amniotic fluid areas on US images. In this work, we propose a two-tier approach of deep learning based techniques to (I) locate the bounding box of the uterus, and (II) segment the pixels in the uterine region into fetal body, amniotic fluid, and background. An example of fetal US image and its annotation is visualized in **Fig.1**.

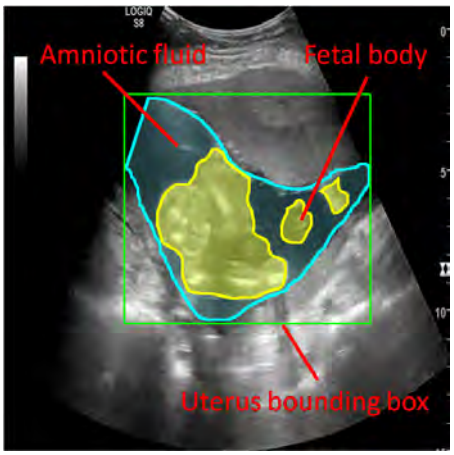



**Fig. 1** Example of the fetal US image and its annotation

Firstly, we expect to use localized uterus area to improve the semantic segmentation results. Conventional methods use manual designed feature descriptors and trained classifiers to detect the object from the images in a sliding window fashion. For instance, Albayrak et al.[5] use Histograms of Oriented Gradients (HOG) to extract the responses of detection widows from each of the positions, and scores each of the positions using a Support Vector Machine (SVM) classifier.

The deep learning based convolutional object detectors (such as F-RCNN[9] or SSD[17] etc) have shown large advantages on object detection tasks by well-designed neural network structures. They map the input image into a target dimension space which corresponds to the vectors between sets of pre-defined positions to the ground truths.

On the other hand, to achieve pixel-wise classification,

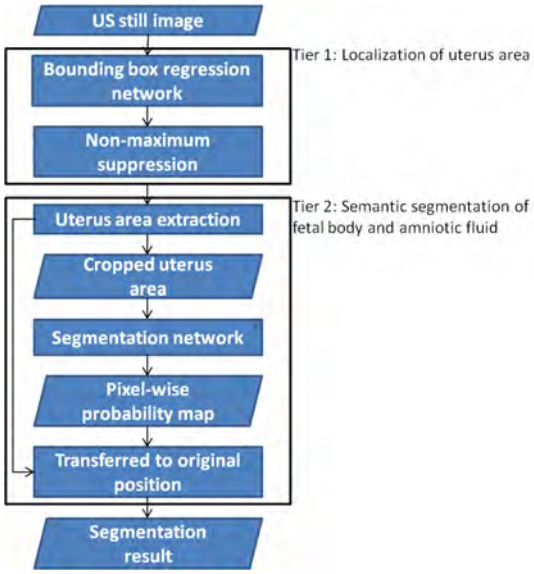



**Fig. 2** The flow map of proposed method

in the earlier study, low level patterns are used as the clues for clustering each pixel to groups or extracting the edge of objects on the US images, some of the related works[8] achieve the segmentation by morphology options or gradient filters etc. Mitchell SC et al.[10] propose the energy function and statistic model based approaches. B. Georgescu et al.[7] propose a data-driven approach to detect and segment the edge of the left ventricle (LV) from US image. Comparing with recent deep learning based works, the methods lack of accuracy and rely on regular shape information.

Regarding the deep learning based US image segmentation, Carneiro et al.[4] adopt deep belief network (DBN) to localize the most likely positions on the perpendiculars of the contour of anatomical structures. The feature is extracted on each of the positions of the anchor points to obtain the responses, which makes redundant computations on overlapping areas. Chen et al.[6] use off-the-shelf fully convolution network (FCN) structure[2] to segment the LV from designated US slices. They iteratively segment the desired object in the sub region of the US image. The results show that the segmentation in the detailed areas of complex scenes is still need to be improved.

To verify the performance of deep learning based approach for uterus localization and improve the accuracy of the follow-up segmentation task, we first propose a CNN based bounding box regression model to regress the offsets of fixed number of multiple pre-defined positions (which is named as reference box), and crop the uterus area by the localized bounding box with some residuals. Then we apply CNN based architecture to segment the fetal body and amniotic fluid in an end-to-end, fully supervised learning pipeline. We use a symmetrical designed fully convolution structure[1] to achieve a pixel-wise and multiple category classification on fetal US images. To further improve the predictive performance in complex scenes and smooth out the segmentation results we modify the structure with additional layers and multi-scale supervisions. We compare the proposed method with several recently published approaches through experiments.

## 2. Method

### 2.1 Overview

The overall working flow is illustrated in **Fig.2**. The input of our system is a still US image. We first feed the raw US still image into a bounding box regression network to localize the position of the uterus. The bounding box regression network predicts the offsets between a set of pre-defined coordinates and the ground truth positions. Then Non Maximum Suppression[14] (NMS) is adopted as post-processing approach to cluster the multiple candidate bounding boxes. Regarding the tier 2, we extract the uterus area by obtained bounding box and then resize the sub region of image to a fixed size. Then we classify each of the pixels to pre-defined categories on the input region by a second segmentation network. Finally, the pixel-wise probability map is transferred back to the origin position by the localized uterus area.

### 2.2 Localization of uterus area

In this section, we describe the proposed method of how to localize the position of uterus in the pregnant patient's raw US images through bounding box regression network.

#### 2.2.1 Framework

The proposed approach is illustrated in **Fig.3**. We define the position of the uterus as tight bounding box, which can be represented by $P = (X_{\mathrm{tl}}, Y_{\mathrm{tl}}, X_{\mathrm{br}}, Y_{\mathrm{br}})$, where $X_{\mathrm{tl}}, Y_{\mathrm{tl}}, X_{\mathrm{br}}, Y_{\mathrm{br}}$ indicate the top left and bottom right x (horizontal) and y (vertical) coordinates, respectively. The ground truth position $P_{\mathrm{GT}}$ of bounding box is defined as the tight rectangle that starts from the leftmost pixel of uterus, and ends at the rightmost pixel of the uterus. Inspired by convolutional object detectors such as F-RCNN[9] or SSD[17], we use multiple reference boxes with fixed initial positions and calculate the distance to the target positions. We expect the deeply learned model learn the offsets between initial position and ground truth, and object confidence of each of the reference boxes as well.

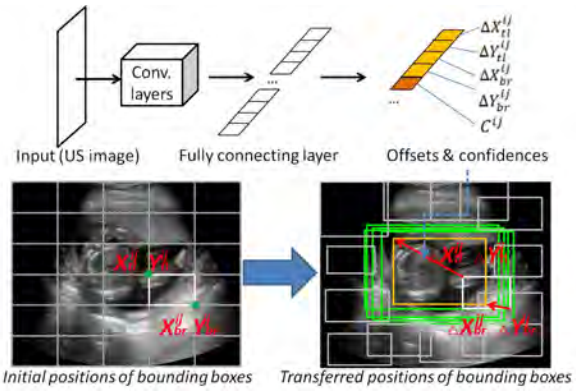



**Fig. 3** Framework of bounding box regression network

### 2.2.2 Offset regression

To predict the position of the bounding box of uterus by convolution neural network model, we design a regression model which predicts the distance between sets of pre-defined locations and the ground truth locations. We define multiple reference boxes through initial positions $P_{\text{Init}}$. For each image sample image, we equally place $n$ initial reference boxes at every column and row. Given sample image $I(v)$, the offset is defined as $P^{ij}_{\text{offset}} = (\Delta X^{ij}_{\text{tl}}, \Delta Y^{ij}_{\text{tl}}, \Delta X^{ij}_{\text{br}}, \Delta Y^{ij}_{\text{br}})$ for reference box $(i, j)$, where $(\Delta X^{ij}_{\text{tl}}, \Delta Y^{ij}_{\text{tl}})$ is the distance between the top left corner of reference box and the ground truth, and $(\Delta X^{ij}_{\text{br}}, \Delta Y^{ij}_{\text{br}})$ is the distance between the bottom right corner of reference box and the ground truth. Additionally, we also learn the confidence score $C^{ij}$, which indicates if the reference box $(i, j)$ will be used as positive or negative sample, as detailed below in the last paragraph of this section. Formally, we learn the following mapping of $I(v) \sim [(\Delta X^{11}_{\text{tl}}, \Delta Y^{11}_{\text{tl}}, \Delta X^{11}_{\text{br}}, \Delta Y^{11}_{\text{br}}, C^{11}), ...(\Delta X^{ij}_{\text{tl}}, \Delta Y^{ij}_{\text{tl}}, \Delta X^{ij}_{\text{br}}, \Delta Y^{ij}_{\text{br}}, C^{ij})]$ for each of the reference box $(i, j)$, where $i$ and $j$ range from 1 to $n$. All of the $(4+1) \times n \times n$ outputs are learned simultaneously by our network (**Fig.3**). In Fig.3, rectangles in green and orange show predicted bonding boxes. This gives us (normalized) predicted positions $P_{\text{Pred}}$ for each of the reference boxes by

$$P_{\text{Pred}} = P_{\text{Init}} + F_{\text{d}}(I(v)), \tag{1}$$

where $F_{\text{d}}(U)$ indicates the trained bounding box regression network.

In our implementations, the offset value $\Delta X$ and $\Delta Y$ are normalized by the size of original image, such as,

$$\Delta X = (X_{\text{Init}} - X_{\text{GT}})/w, \Delta Y = (Y_{\text{Init}} - Y_{\text{GT}})/h, \tag{2}$$

where $(X_{\text{Init}}, Y_{\text{Init}})$ is the coordinate of initial position while $(X_{\text{GT}}, Y_{\text{GT}})$ is the ground truth, $w$ and $h$ are the width and height of the input US image. The normalization limits the output values so that the training loss converges.

The ground truth category $C^{ij}_{\text{GT}}$ of each reference box is determined by the area $V^{ij}$ in which the reference box and ground truth bounding box overlap, such as,

$$C^{ij}_{\text{GT}} = \begin{cases} 0, \text{ if } V^{ij} > \Theta_{\text{v}} \\ 1, \text{ if } V^{ij} \leq \Theta_{\text{v}} \end{cases}, \tag{3}$$

where $\Theta_{\text{v}}$ is a pre-defined value (in our experiments we use 0.3). The confidence labels of the corresponding reference boxes are set to 1 (positive); otherwise, the label is set to 0 (negative). The specific computation of the overlapping area $V^{ij}$, in which the ground truth bounding box and reference box overlap, are performed by Eq.(4):

$$V^{ij} = (S^{ij} \cap S_{\text{GT}})/(S^{ij} \cup S_{\text{GT}}), \tag{4}$$

where $S^{ij}$ indicates the area of reference box $(i, j)$, $S_{\text{GT}}$ indicates the area of uterus bounding box, $\cap$ and $\cup$ are intersection and union of the two areas, respectively. During testing, the predicted category values are used as the confidence of each reference boxes. We treat the boxes which have higher positive confidence than negative ones as the positive positions. In our implementation, all of the outputs (the sets of regression offsets and confidences) are assigned as one vector, which is together calculated by linear combination on previous fully connecting layer. It brings advantage in robust global and context information because all of the positions and confidence scores of the reference boxes are considered jointly.

### 2.2.3 Post-processing

As introduced, the method first transfers all of the initial coordinates to the target positions $P_{Pred}$ by predicted offsets, as Eq.(1). To demonstrate the relations between initial reference boxes and prediction results, the positions of initial reference boxes, the positions of transferred reference boxes, and the final prediction results after NMS are visualized in Fig.4. Here, in Fig.4, column a) shows initial reference boxes with category annotations (green: positive, red: negative) which are assigned by overlapping area with ground truth, column b) shows the regressed positive (in green) and negative (in red) bounding boxes, and column c) shows the uterus localization results after NMS, and the yellow rectangles indicate the ground truth bounding boxes. This initial prediction contains many overlapped bounding boxes which are assigned to the same object in the image (as shown in **Fig.4** (b)). Therefore, the approach needs to further

eliminate the redundant predictions. In this paper we cluster the multiple overlapping boxes by NMS[14]. The NMS seeks for the position with the maximum confidence value in a given region which might contain the same object by eliminating all of the bounding boxes whose overlapping areas are larger than a threshold. Consequently, the merged bounding box is obtained as the final result of the uterus detection. Among the obtained multiple candidate bounding boxes, we keep the bounding box which has the largest confidence as per NMS.

### 2.3 Semantic segmentation of anatomical structure

In this section we describe the semantic segmentation model for anatomical structure segmentation. To prevent that the obtained bounding box is smaller than the actual uterus area, we extend the bounding box area with a fixed factor of 1.2 to leave some residuals for the areas near the uterine border.

To segment the pixels from the obtained uterine region into the fetal body, amniotic fluid and background, we adopt framework based on symmetrical designed "encoder-decoder"[1] CNN. The input image is scaled down by a series of pooling and a second series of up-sampling operations. **Fig.5** illustrates the overall framework. The network first maps the input image into multi-channel down-scaled feature maps then up-sample the feature maps back to the original size and target channels (which has same number of categories). In Fig.5, the input and output dimensions of feature maps are represented in *width*$\times$ *height*$\times$*channels*. The optimizations for improving segmentation accuracy and smoothness are introduced as follows.

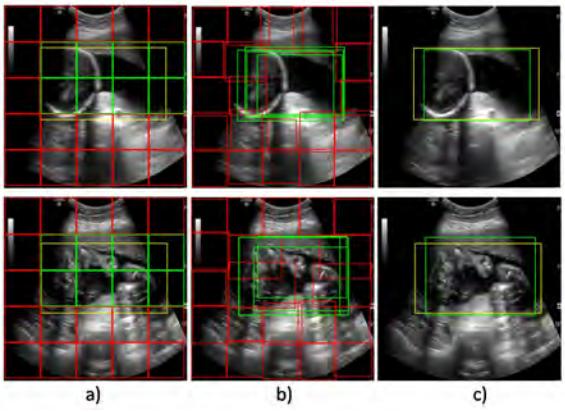



**Fig. 4** The visualized reference boxes and the prediction results

#### 2.3.1 Encoding-decoding architecture

The input image is first mapped to a set of down-scaled feature maps by convolution and pooling operations. Our work uses max-pooling, which keeps only the maximum value of each pooling window. In the first half stage of the network (encoding stage), multiple max-pooling operations are adopted. Overall, the feature maps of the last layer of the encoding stage are down scaled to $(1/2)^M$ of the raw input by M max-pooling layers with stride two.

Then, to rescale the output of the encoder stage to the same size as the input image we use un-pooling[3] operations. Each pixel is assigned to the recorded maximum position in each of the corresponding windows, while the other positions are filled with zeroes. For backward propagation, the derivatives are passed to the former layer by only keeping the largest one in the window. Each un-pooling layer is concatenated with convolution layers which have learnable convolution kernels.

#### 2.3.2 Inner layers

From the bottom layer to the top layer in the encoding stage, we can assume that the responses of higher layers represent more global information. In the first several layers, the learned convolution kernels work as low level fea-

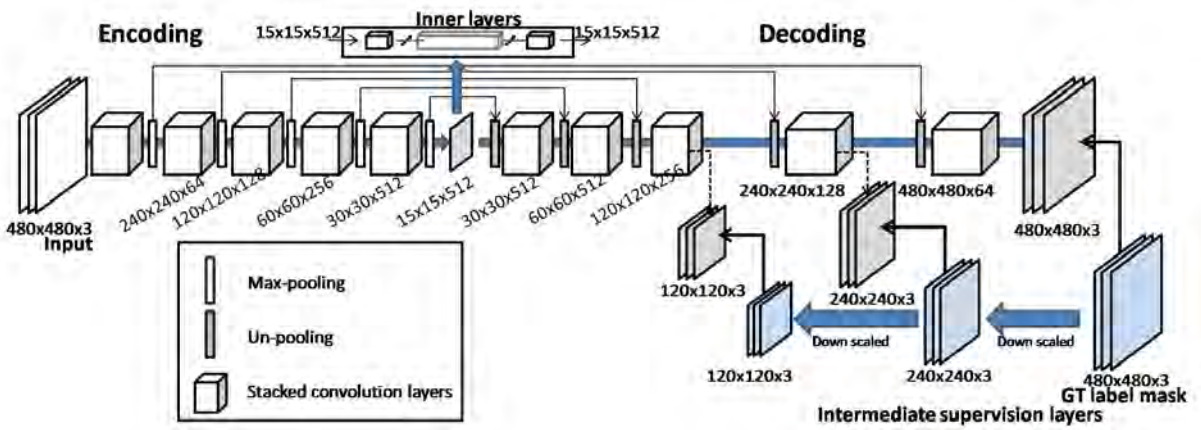



**Fig. 5** Illustration map of segmentation network for anatomical structure

ture descriptors such as edge, corner, etc. As the network gets deeper, the learned convolutional weights tend to focus on higher level representations such as shape, spatial relationships etc. To this end, we add layers called "inner layers" between the encoding and decoding stages to enhance higher level representations. The stacked inner layers are connected to the last layer of down sampling stage. The inner layer works as the smallest size local receptive fields, where we use 1x1 sized convolution operations at each position to map the data into additional hieratically feature spaces. For each of the inner layers, we widen the network by using relatively large number of convolution kernels to enhance the hierarchical representation.

#### 2.3.3 Intermediate supervision

In the decoding stage of segmentation network, we notice that the un-pooling operation causes un-smoothed segmentation in the border areas. By comparing with interpolation up-scaling, the un-pooling operation's output are relative sparse because most of the positions are filled with zeroes. It causes the final segmentation results to be very non-smooth in some areas, especially in the boundaries.

We propose to improve the smoothness by using multiple output branches with multi-scaled supervision signals. The images with different resolutions yield different detailed information in the borders. The higher resolved segmentation maps contain more information in the border area and have smoother edges. Thus, we expect to enhance the continuity of each group of convolution layers by different resolved ground truth segmentation maps. Specifically, inspired by GoogLeNet[15], we not only use the error from the last output layer, but also from inserted additional output branches among the decoding structure. The additional outputs are used as auxiliary branches for calculating the errors between the predictions and down-scaled ground truth label maps as shown in Fig.5. They are concatenated to the output of convolution layers in the up-sampling stage. We use nearest neighbor interpolation to down-scale the ground truth label map from the original size to the same size as the corresponding output branch. To map the intermediate output to the same dimension number as the ground truth, for each of the intermediate branch we use convolution layers with a fixed number of convolution kernels (the number of kernels equals to the number of channels of the output). For each of the intermediate supervision layer we target to minimize the cross-entropy loss of all of the positions on the response map. In training we treat all of the errors equally by summing the derivatives of each of the branch with same weights. We do not weight the error of each branch layer by the size of the output map since the loss is normalized by the number of pixels of the output.

During testing, the auxiliary branches are discarded, and only the output of the last layer is used as the final segmentation results.

### 2.4 Backbone network

We use the VGG16[11] structure as the base feature extractor (backbone network) for both bounding box regression and semantic segmentation networks. It has been successful in many computer vision and recognition tasks such as ImageNet[13] object classification challenge and has relative low hardware requirements. In this work we treat the backbone network as feature extractor and do not measure the effects of different backbone networks.

Regarding the structure of output vector of bounding box regression network, our model predicts the offsets and confidence scores $C^{ij}$ for each of the $n \times n$ bounding boxes from per US image. As shown in Fig.3, we organize the output (the offsets and confidences) as a one-dimensional vector. Thus we convert the $n \times n \times 4$ offsets values and $n \times n \times 1$ confidence values to one long vector $D_{\mathrm{Pred}}$ and use it as the output of the model. The Euclidean loss is adopted as our loss function of detection network by,

$$L_{\mathrm{d}} = ||D_{\mathrm{Pred}} - D_{\mathrm{GT}}||^2, \tag{5}$$

where $D_{\mathrm{GT}}$ is the ground truth vector.

The segmentation network adopts three output channels as we want to classify the pixels into three categories (fetal head, amniotic fluid, and background). We calculate the confidence for each of the pixels. For each input image with $H$ pixels, the corresponding loss function $L_{\mathrm{s}}$ of our segmentation network is:

$$L_{\mathrm{s}} = -\frac{1}{H}\sum_{h=1}^{H} \log p_h, \tag{6}$$

where $p_h$ is the predicted confidence of pixel $h$ which corresponds to its ground truth category. For each of the intermediate and the last output layer we use the same probabilistic function.

## 3. Experiment

### 3.1 Dataset

#### 3.1.1 Experimental environment

To conduct our experiments, we received approval from the Ethics Review Committee on Research with Human Subjects, Waseda University (2014-165). We use GE Voluson E8 and C1-5 linear array transducer with frequencies in the range of 4.0 Hz. The axial and lateral resolution is 2mm and 3mm, respectively. The field of views is 66 degrees and the depth setting is 15cm. The average fetal age of our subjects is approximately 22 weeks. We acquire four video clips from anonymous patients. The original resolution of each frame is $640 \times 480$ pixels. The raw data are sampled from each video clips every two frames. We use 2-fold cross validation by first randomly separating the different patients' data into the training and testing sets. The contour annotations are made by doctors who have years of US examination experiences.

#### 3.1.2 Data cleaning

The duplicate or nearly duplicate images do not contribute to our training, and bring bias to the training or testing sets. We one by one feed each of the data samples through a fully trained CNN model (in this research we use alexnet[12] network structure and trained on ImageNet dataset) and compare the Euclidean distance of output feature vector of the last fully connected layer to all of the other samples. The image pairs which have close distance are treated as the similar image samples. We discard all of the redundancy from the original data set. The cleaned dataset has about 400 samples for training and testing.

### 3.2 Training details

#### 3.2.1 Data augmentation and hyper parameters

We augment the training data by adding random disturbance to the predicted uterus bounding boxes. In particular, multiple sub regions are cropped from the original image with random transfer factor on the four points of the bounding box. The corresponded label maps also undergo the same operations. It makes our training samples increase to more than 4000 training samples in each subset. For comparison, the result without data augmentation ("_w/oAug") can be found in **Table 1**.

Regarding the hyper parameters, our experiments run on a single NVIDIA GTX 1080. The deep learning platform is modified from Caffe[16]. In each iteration the error between the ground truth and output is summed and averaged over entire batch. We stop model training after 20,000 iterations. The inference time on GPU is about 30ms for the detection and 60ms for the segmentation networks per image, respectively.

**Table 1** Structures of inner layers used by different models

| Layer name | Segnet | _1IL | _2IL | _3IL |
|---|---|---|---|---|
| Layer1 | None | 512 | 4096 | 4096 |
| Layer2 | None | None | 512 | 4096 |
| Layer3 | None | None | None | 512 |

#### 3.2.2 Domain transferred learning

Studies proved that the low level feature representations are highly similar across many domains. Compared with randomly initialized weights, the fine-tuning on the pre-trained model converges at a faster speed and achieves better performance. The weights of the convolution layers of the segmentation model are initialized from a pre-trained model[18]. During fine-tuning of our models, the newly added layers have five times as large learning rate as the fine-tuned layers. For comparison, the result without using pre-trained model ("_w/oPre") can be found in the last row of Table 1. Note that regarding the uterus detection network, we just directly train the model from scratch.

### 3.3 Evaluation criteria

We use intersection of union (IOU) to evaluate uterus detection results. Specifically, IOU is the proportion of the intersection of true positive pixels and union area of the predictions and ground truths. Since we use the relative confidences between positive and negative object, and to avoid the single threshold limitation, the experiments further adopt Average Precision (AP)[21]to evaluate the detection performance. To demonstrate the different alignment accuracy between ground truth bounding box and prediction, we use three overlapping thresholds 0.5 (AP.5), 0.6 (AP.6) and 0.7 (AP.7). The bounding boxes which have overlapping ratio larger than the thresholds are considered as true positives. The larger threshold is stricter than smaller one.

Regarding semantic segmentation, besides IOU, we demonstrate the class specified evaluation results by ROC curve. The true positive rate (TPR) and false positive rate (FPR) over classes are calculated by:

$$\mathrm{TPR} = N_{\mathrm{TP}}/N_{\mathrm{Pos}}, \mathrm{FPR} = N_{\mathrm{FP}}/N_{\mathrm{Neg}}, \qquad (7)$$

where $N_{\mathrm{TP}}, N_{\mathrm{FP}}, N_{\mathrm{Pos}}, N_{\mathrm{Neg}}$ indicate the number of true positive, false positive, positive and negative pixels of each category, respectively. We further adopt pixel-wise

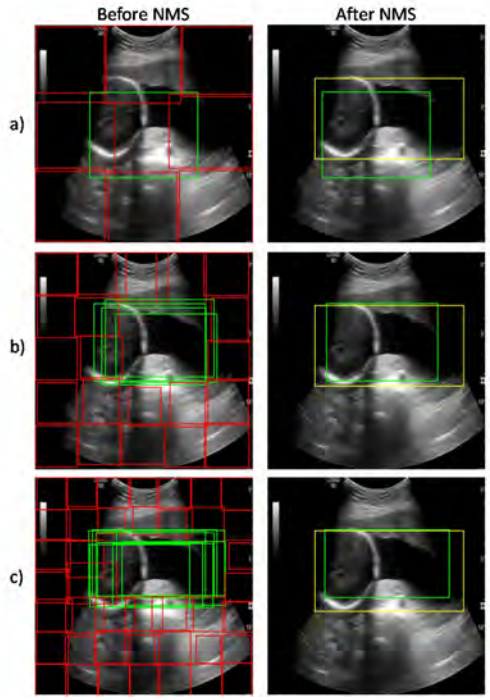



**Fig. 6** Comparison between a) $3\times3$, b) $5\times5$ and c) $7\times7$ reference boxes (green: positive, red: negative, yellow: ground truth); (left column: bounding boxes before NMS, and right column: bounding boxes after NMS)

**Table 2** Evaluation results of uterus localization (IOU)(%)

| Method | URN_ $3\times3$ | URN_ $5\times5$ | URN_ $7\times7$ | FRCNN | SSD |
|---|---|---|---|---|---|
| Sub.1 | 55.7 | 63.1 | **64.6** | 61.0 | 60.7 |
| Sub.2 | 55.0 | **61.7** | 59.7 | 60.2 | 59.1 |
| Avg. | 55.3 | **62.4** | 62.1 | 60.6 | 59.9 |

**Table 3** Evaluation results of uterus localization (AP)(%)

| Method | URN_$5\times5$ | FRCNN | SSD |
|---|---|---|---|
| AP.5 | **99.6** | 99.5 | 99.4 |
| AP.6 | **88.6** | 85.4 | 83.3 |
| AP.7 | **68.3** | 53.7 | 50.2 |

classification accuracy (Accu) to evaluate the segmentation results without the effect of different categories.

### 3.4 Results and discussions

#### 3.4.1 Localization of uterus area

The visualized examples of the bounding box regression network with different settings are shown in **Fig.6**. The IOU and AP results are shown in **Table 2** and **Table 3**, respectively. We name our proposed uterus regression network as "URN". The networks with different number of reference boxes are named with suffix "_$3\times3$", "_$5\times5$" and "_$7\times7$" in this study.

We evaluate the uterus localization results on different

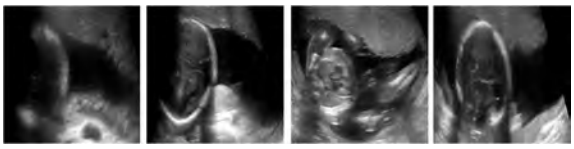


**Fig. 7** Normalized uterine images by detected bounding box

setting and compare them with F-RCNN[9] and SSD[17]. For F-RCNN we use multi-scale training scheme to randomly resize input image in 3 different scales. During testing we use original size as input. Regarding SSD, we adopt $300\times300$ as both training and testing input. For both of above methods, we adopt VGG16 as the backbone network.

The IOU shows that URN_$5\times5$ achieves the best score but the gaps between different results are not obvious. The AP demonstrates that URN_$5\times5$ model achieve higher accuracy especially on more strict criteria (AP.7). It proves that our method has better alignment accuracy. The major difference is the last fully connecting operation. It introduces global feature to the offsets regression, which brings richer context information. The global regression scheme is more robust to the case of uterus localization in US image.

The advantages of usage of cropped areas can be explained from following 2 aspects: 1. constraint the location of the uterus to the area inside the uterus bounding box; 2. alleviate the unbalanced data distribution. The cropped uterus examples are shown in **Fig.7**. We notice that the predicted uterus area smaller than uterus, it is caused by unclear feature at left and right areas (as shown in Fig.6). It does not heavily affect to the segmentation scheme since we extend every bounding box with fixed factor. The quantitative results of semantic segmentation w/ and w/o cropped uterus area can be found in the column 1 ("SegNet_nodet" , here, the "_nodet" indicates the model without using uterus detection as the pre-processing. ) and 2 (SegNet) of **Table 4**. It proves that the localization scheme improves the accuracy of following segmentation work.

#### 3.4.2 Semantic segmentation of anatomical structure

We named our optimized scheme as "EDN". The base scheme without any modification is same as Segnet. Regarding the models with different settings, in this study, the models with intermediate supervisions are named with the suffix "_IS". The models with $k$ ($k$=1,2,3) inner layers are named with the suffix "_$k$IL". For exam-

ple, the "EDN_3IL_IS" is the results of EDN model with 3 inner layers and intermediate supervisions. Firstly, to demonstrate the performance of intermediate supervision layers, in **Fig.8**, we show some of the segmentation results by highlighting the edge of the segmented blobs. The smoothed boundaries which are predicted by model with intermediate supervision layers show significant improvements in the given image samples. The intermediate supervision layers also improve the overall segmentation accuracy (EDN_IS of Table 4).

**Table 4** Pixel-wise accuracy over all of the pixels (Accu) (%)

| Method | SegNet _nodet | SegNet | EDN _IS | EDN _1IL _IS | EDN _2IL _IS | EDN _3IL _IS |
|---|---|---|---|---|---|---|
| w/ cropped uterus area | | √ | √ | √ | √ | √ |
| w/ intermediate supervision | | | √ | √ | √ | √ |
| w/ 1 inner layer | | | | √ | √ | √ |
| w/ 2 inner layers | | | | | √ | √ |
| w/ 3 inner layers | | | | | | √ |
| Subset 1 | 91.99 | 92.23 | 93.43 | 93.97 | 93.81 | **94.08** |
| Subset 2 | 90.41 | 91.05 | 92.22 | 92.39 | 92.49 | **93.71** |
| Avg. | 91.20 | 91.64 | 92.83 | 93.18 | 93.15 | **93.90** |

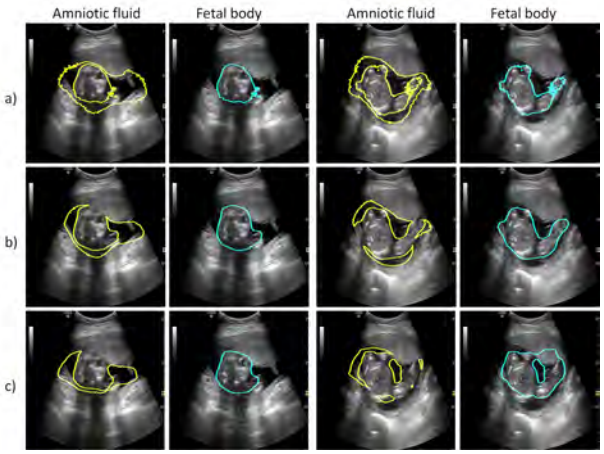



**Fig. 8** The visualized segmentation results; a) SegNet (w/o intermediate supervision layers), b) EDN_IS (w/ intermediate supervision layers) and c) PSPVGG16

Then, to verify the effectiveness of inner layers, we use several settings with different number of inner layers. We list the detail of the structures of inner layers in Table 1. The quantitative accuracy (Table 4 and **Fig.9**) and visualized results (**Fig.10**) show the additional inner layers bring higher accuracy on the segmentation results.

Concerning comparison experiments, we adopt FCN and Segnet as baseline methods. Besides the baseline methods, we also evaluate Deeplabv3+[19] and PSPNet[20] with same training and testing sets. Note that, all of the segmentation models are trained based on the cropped uterus area for fair comparison. The FCN8s model is iterative trained from FCN32s and FCN16s. Regarding the Deeplabv3+, we adopt the ASPP scheme with batch normalizations and an additional encoder-decoder layer. For fair comparison purpose, the results are firstly evaluated and compared with Deeplabv3+ and PSPNet with VGG16 backbone (V3+VGG16 and PSPVGG16). Numerical results of above models are demonstrated in **Table 5** and Fig.9. In Table 5, the suffix of "_fb" indicates fetal body, "_af" indicates to amniotic fluid, and "_bkg" indicates to background. Some visualized examples of extracted contours and segmented blobs with different approaches and proposed methods can be found in Fig.8 and Fig.10.

In addition, the paper further provides the results of

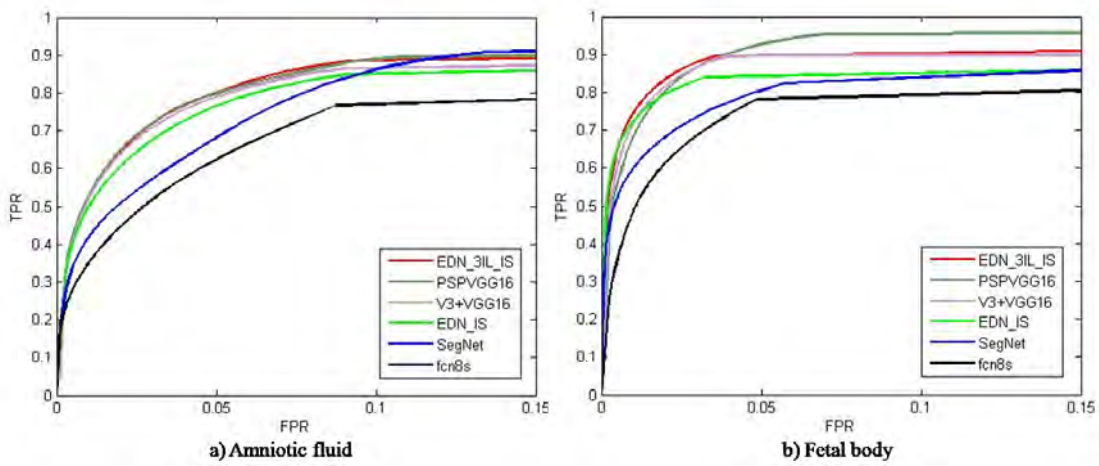



**Fig. 9** Category specified ROC curves of different models

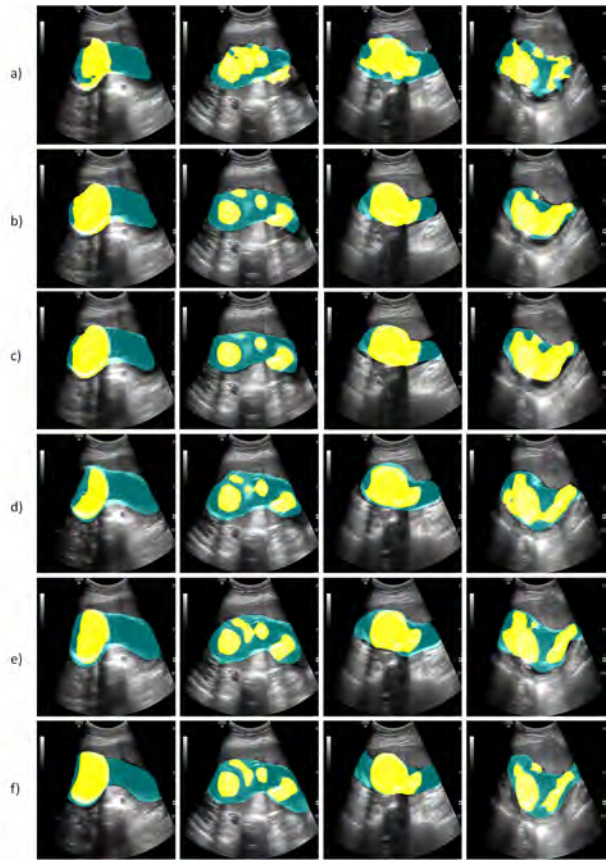



**Fig. 10** Visualized segmentation results; a) FCN8S, b) PSPVGG16, c) V3+VGG16, d) EDN_IS, e) EDN_3IL_IS, and f) ground truth

Deeplabv3+ and PSPNet with Resnet50[22] as the backbone network in **Table 6** (V3+RES50 and PSPNet). In Table 6, the suffix of "_fb" indicates fetal body, "_af" indicates to amniotic fluid, and "_bkg" indicates to background. Interestingly, the numerical results do not achieve better scores than using VGG16's model in our experiments. It might because in this paper we use relative small batch size (batch size=6 for both models) on a single GPU device. Even if we have trained the models with more iterations, it seems that the performances of networks trained with batch normalizations (i.e Resnet50, Resnet101) are heavily depended on large batch size.

The best performance in the quantitative results is our proposed optimized encoding-decoding model with 3 inner layers and intermediate supervisions. Although the result does not have large gaps with V3+VGG16 and PSPVGG16 but we notice that it can proved more complete contour of objects. The proposed inner layers bring similar effects as the atrous convolution or multi-scale pooling by introducing larger field of views to the feature maps. Compare with PSPVGG16, the additional intermediate supervision branches not only give us faster converging speed in the training, it also brings smoother segmentation results by using different scales of ground truth masks in training. From the visualized results, the FCN8s model shows under fitting to the ground truth area. Our model performs more complete and accurate contour information on the segmentation results than others. Despite this, from the visualized samples we can see that the model with intermediate supervision have more smooth segmentations in the border areas.

**Table 5** Class specified results (IOU) and pixel-wise accuracy over all of the pixels (Accu) (avg. of 2 sets.) (%)

| Method | IOU_fb | IOU_af | IOU _bkg | mIOU | Accu |
|---|---|---|---|---|---|
| FCN8S | 49.32 | 36.06 | 92.37 | 59.25 | 89.86 |
| SegNet | 55.00 | 44.83 | 93.35 | 64.39 | 91.64 |
| PSPVGG16 | 65.13 | **55.13** | 94.84 | 71.70 | 93.03 |
| V3+VGG16 | 69.17 | 52.64 | 95.19 | 72.33 | 93.51 |
| EDN_w/ oAug | 44.29 | 30.91 | 92.03 | 55.74 | 88.01 |
| EDN_w/ oPre | 61.24 | 50.81 | 93.10 | 68.38 | 89.90 |
| EDN_IS | 64.10 | 51.71 | 94.90 | 70.24 | 92.83 |
| EDN_3IL_IS | **69.36** | 54.79 | **95.44** | **73.19** | **93.90** |

**Table 6** Evaluation results of DeeplabV3+ and PSPNet with Resnet50 as the backbone network (%)

| | IOU_fb | IOU_af | IOU_bkg | mIOU | Accu |
|---|---|---|---|---|---|
| PSPNet | 61.48 | 50.41 | 94.17 | 68.69 | 92.95 |
| V3+RES50 | 61.79 | 51.80 | 94.83 | 69.47 | 93.50 |

## 4. Conclusion

This paper has proposed deep learning based uterus detection and anatomical structure segmentation frameworks. Through experiments we verify our methods and conclude the results as follows:

a) We use bounding box regression network to localize the positions of the uterus and achieve the best performance within different comparison models.
b) Compared with segmentations applied directly to raw US images, segmentations in the cropped uterine area optimize the segmentation results by relieving the imbalance issue and aligning the region of target pixels.
c) The additional $1 \times 1$ sized inner layers improve the overall segmentation accuracy by extending the network to larger dimensional space and global representations.
d) The intermediate supervision layers smooth out the segmentation results through adding multi-scaled supervisions after each group of convolution layers.

In this work we do not extend our proposed optimiza-

tions to other backbone network such as Resnet. The reason is because the scheme records geometric representation and makes use of more local position information which is obtained from max-pooling operations. However, the original Resnet does not provide such down scaling method which is required by our currently used decoding scheme. Regarding the limitation of the work, the symmetric designed segmentation scheme has drawbacks on high GPU memory cost in the training, which will cause the training process on our device meets the out of GPU memory problem. It limits us to adopt the backbone networks with less learn-able weights. Therefore, in this work we do not concern much about the backbone networks.

To deal with above limitations, as for the future work, firstly we will try to modify the base structure such as Resnet backbone to fit the decoding structure used in our segmentation scheme. Then, we plan to reduce the memory cost caused by the symmetric designed scheme, and make the network more light weight to avoid the computation limitation without losing too much performance. We are also considering integrating the proposed optimization modules with other optimization modules such as ASPP in related US image segmentation tasks.

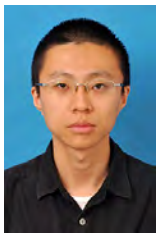
On the other hand, we notice that our current two-stage approach has redundancies in computations. We might speed up the system and optimize the learning strategy through joint learning of the bounding box regression and segmentation.

## References


1) V. Badrinarayanan, A. Kendall, R. Cipolla:"SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 39, No. 12, pp.2481–2495 (2017).
2) E. Shelhamer, J. Long, T. Darrell:"Fully Convolutional Networks for Semantic Segmentation", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 39, No. 4, pp.640–651 (2017).
3) M.D. Zeiler, R. Fergus: "Visualizing and Understanding Convolutional Networks", Proc. of European Conference on Computer Vision, Vol.8689, pp.818–833 (2014).
4) G. Carneiro, J. C. Nascimento, A. Freitas: "The Segmentation of The Left Ventricle of The Heart from Ultrasound Data Using Deep Learning Architectures and Derivative-Based Search Methods", IEEE Trans. on Image Processing, Vol.21, No.3, pp.968–982 (2012).
5) N. B. Albayrak, A. B. Oktay, Y. S. Akgul: "Prostate Detection from Abdominal Ultrasound Images: A Part Based Approach", Proc. of IEEE International Conference on Image Processing, pp.1955–1959 (2015).
6) H. Chen, Y. Zheng, J.H. Park, P. -A. Heng, S. K. Zhou: "Iterative Multi-domain Regularized Deep Learning for Anatomical Structure Detection and Segmentation from Ultrasound Images", Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 9901, pp.487–495 (2016).
7) B. Georgescu, X. S. Zhou, D. Comaniciu, A. Gupta: "Database Guided Segmentation of Anatomical Structures with Complex Appearance", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, pp.429-436 (2005).
8) R. Supriyanti, D. A. Putri, E. Murdyantoro, H. B. Widodo: "Comparing Edge Detection Methods to Localize Uterus Area on Ultrasound Image", Proc. of International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering, pp.152–155 (2013)
9) S. Q. Ren, K. M, He, R. Girshick, J. Sun: "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks", Proc. of International Conference on Neural Information Processing Systems, Vol.1, pp.91–99 (2015)
10) S. C. Mitchell, B. P. Lelieveldt, V. D. G. Rj, H. G. Bosch, J. H. Reiber, M. Sonka: "Multistage Hybrid Active Appearance Model Matching: Segmentation of Left and Right Ventricles in Cardiac MR Images", IEEE Trans. on Medical Imaging, Vol.20, No.5, pp.415–423 (2001)
11) K. Simonyan, A. Zisserman: "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556, (2014)
12) A. Krizhevsky I. Sutskever, G. Hinton: "ImageNet Classification with Deep ConvolutionalNeural Networks", Advances in Neural Information Processing Systems 25, pp.1097–1105 (2012)
13) J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, F. F. Li: "ImageNet: A Large-scale Hierarchical Image Database", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp.248–255 (2009)
14) A. Neubeck, L. V. Gool: "Efficient Non-Maximum Suppression", Proc. of IEEE Conference on International Conference on Pattern Recognition, Vol.3, pp.850–855 (2006).
15) C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich: "Going Deeper with Convolutions", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp.1–9 (2015).
16) Y. Q. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell: "Caffe: Convolutional Architecture for Fast Feature Embedding", Proc. of ACM International Conference on Multimedia, pp. 675–678 (2014).
17) W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Ful, A. C. Berg: "SSD: Single Shot MultiBox Detector", Proc. of European Conference on Computer Vision, Vol.9905, pp.21–37 (2016).
18) L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille: "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs", arXiv:1412.7062 (2014).
19) L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam: "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation", arXiv:1802.02611 (2018).
20) H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia: "Pyramid Scene Parsing Network", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 6230–6239 (2017).
21) M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman: "The Pascal Visual Object Classes (VOC) Challenge", International Journal of Computer Vision, Vol.88, No.2, pp.3030-338 (2010).
22) K. M. He, X. Zhang, S. Q. Ren, J. Sun: "Deep Residual Learning for Image Recognition", Proc. of IEEE Conference on Com-

puter Vision and Pattern Recognition, pp.770–778 (2016).

(Received April 13, 2018)
(Revised December 2, 2018)



**Yan LI** (*Student Member*)
He earned B.S. degree in Communication University of China. He obtained M.S. degree in Graduate School of Global Information and Tele-communication Studies (GITS), Waseda University. He is currently a doctoral program student in Graduate School of Creative Science and Engineering, Waseda University. His research areas include computer vision and machine learning.

**Rong XU**
He earned B.S. degree in School of Information Science and Engineering, and M.S. degree in School of Computer Science and Engineering from Southeast University in 2003 and 2009, China, respectively. He obtained his Ph.D degree in Graduate School of Global Information and Tele-communication Studies (GITS), Waseda University in 2014. Currently, he is a senior researcher in Datasection Inc., a Japanese listed company. His research focuses on image processing, video analysis, pattern recognition, deep learning, surgical image-guidance, etc.

**Artus KROHN-GRIMBERGHE**
He earned his MSc equivalent in information engineering and management from KIT, Karlsruhe Institute of Technology in 2006. After roughly a year in the industry he went back to university and obtained his Ph.D. in Machine Learning from University of Hildesheim in 2012. Afterwards he joined University of Paderborn as an assistant professor in applied Machine Learning in 2012 for six years till 2018. Currently, he is founder and CEO of LYTIQ GmbH, a Germany-based startup focused on consulting in AI (www.lytiq.de).

**Jun OHYA** (*Member*)
He earned B.S., M.S and Ph.D degrees in precision machinery engineering from the University of Tokyo, Japan, in 1977, 1979 and 1988, respectively. He joined NTT Research Laboratories in 1979. He was a visiting research associate at the University of Maryland, USA, from 1988 to 1989. He transferred to ATR, Kyoto, Japan, in 1992. Since 2000, he has been a professor at Waseda University, Japan. He was a guest professor at the University of Karlsruhe, Germany, in 2005. His research areas include computer vision and machine learning. He is a member of IEEE, IIEEJ, IEICE, IPSJ and VRSJ.

**Hiroyasu IWATA**
He earned the B.S., M.S. and Dr. of Engineering degrees in Mechanical Engineering from Waseda University in 1997, 1999 and 2003, respectively. He is currently a Professor of Department of Modern Mechanical Engineering, School of Creative Science and Engineering, Waseda University and appointed as a Next Generation Core Researcher in Key Researchers Development Program by Board of directors, Waseda University (since 2016) as well. His research fields include human assistive and augmentation robotics, skin adhesive bio-sensors, supernumerary robotic limb "Third Arm", echography-based prenatal checkup assistive robot, perception-assistive rehabilitation RT and artificial intelligence for construction machinery. He is a member of RSJ, JSME, SICE, EAJ and IEEE.

IIEEJ **Paper**

# An Ensemble Approach to Precancerous Cervical Lesion Classification from Cervigram Images

Suleiman MUSTAFA†, Akio KIMURA††(*Member*)

† Graduate School of Engineering, Iwate University, †† Faculty of Science and Engineering, Iwate University

<**Summary**> In this study, we propose a semi-automated system for detecting cervical cancer from cervigram photographs of affected cervix regions. Cervical cancer is among the most common cancers affecting women in the world particularly in developing countries where few population have access to proper screening due to high costs of laboratory testing. For this reason a simple inexpensive test by visual inspection with acetic acid (VIA) is used where the cervix region is observed with the naked eye for change in color, texture and appearance. We consider that applying adequate image processing techniques to the captured images during VIA is effective to assist gynecologist (doctor) for detecting, diagnosing and examining the cervix region based on the visual inspection observations. That is, it is possible to construct a kind of computer aided systems for detecting and diagnosing cervical cancers. In our framework we first segment an input image into lesions of interest by GrabCut algorithm, and next extract many color- and texture-based features by using image processing.Then based on these extracted features the segmented image is categorized as cancerous "malignant" or non-cancerous "benign" by using ensemble classification methods combined with 3 or 5 machine learning algorithms. We conducted some experiments using real cervigram images and found through the statistical analysis that only 10–13 extracted features can be sufficient to detect cervix cancer and our method comparatively improved the detection accuracy compared to visual eye inspection.

**Keywords**: computer aided diagnosis, cervigram classification, cervix segmentation, feature extraction, ensemble learning

## 1. Introduction

Cervical cancer is among the most common cancers affecting women worldwide with an estimated 530,000 new cases reported in 2012 and of the 270,000 deaths from the cancer in 2015 with about 85%–90% occurring in less developed countries[1]. Its incidence in Sub-Saharan Africa is increasing due to lack of or poor screening[2], while in contrast the incidence of human papillomavirus (HPV) is declining rapidly in Western countries by the reducing of the rate of cervical cancer through appropriate screening[3]. Cervical cancer screening is usually achieved by taking a papanicolaou test (Pap smear test), where cells are scraped and collected from the cervix for microscopic examination. This is followed by investigated for irregularities with the aim of detecting potentially precancerous symptoms, i.e., cervical intraepithelial neoplasia (CIN)[1].

Microscopic images of Pap smear cells have been used in various research recently[4)–7]. Several good results have been obtained, however, the capturing of cellular level images is difficult to sustain in resource-poor areas (like developing countries) because of the complicated process; in collection, sampling, preparation, staining, reading, reporting and the period of time it takes before providing test results[3]. Due to the high cost of the equipment used for Pap smear screening, it is being expected to develop low-tech and inexpensive screening tools that could significantly reduce the burden of cervical cancer deaths.[2]There is therefore a need for low cost cervical cancer screening systems that can be used in the developing countries has become an interesting research field recently.

An easier and cost effective alternative screening method is the visual acetic acid (VIA) test[3]performed at tissue level rather than cellular level. It involves applying acetic acid (vinegar) to the region and observing change in color to whitish appearance. This test works in the remotest of settings, bringing access to virtually location.[1] VIA has shown ability to correctly identified between 45%

to 79% of women at high risk of developing cervical cancer by well-trained medical personnel[8]. As well as change in color, texture appearance is also observed for irregular surface, coarseness and mosaic vessels[9]. Shape features on the other hand have been shown to be not so useful at tissue level detection classification[10]. Therefore, some suitable image processing techniques can be applied to the images captured during VIA in order to easily observe their color tone and texture differences which are two main characteristics used by experts to differentiate lesions. Furthermore, machine learning classification can also potentially improve performance in identification of precancerous cervical lesions.

There have been several related (image processing-based) studies that have used VIA image data-sets. Xue et al.[10]developed the web-based cervigram image retrieval system and the system used visual characteristics based on color, texture and shape are used to derive features. Acosta-Mes et al.[11]proposed an aceto-white temporal pattern feature extraction model to identify precancerous cervical lesion. Zimmerman et al.[12]utilized information of gradient, color saturation and intensity to detect specular reflections of cervix images. Song et al.[9]proposed a method using pyramid histogram oriented gradients features (P-HOG) on segmented regions of interest. Xu et al.[13]improved the method of Song et al. by applying the P-HOG features with color features from LAB color space and local binary pattern (LBP) texture features.

Moreover, some cancer classification techniques based on the machine learning algorithms, such as k-nearest neighbors (k-NN), support vector machines (SVMs) and deep learning have been studied recently[9),13),14)], and it has been reported that they work well but under limited conditions with only a few hundred images used for the experiments making it a small dataset. Particularly in deep learning-based method, precancerous lesion detection with a small dataset achieved relatively high accuracy, sensitivity and specificity of 88.9%(0.889), 87.8% and 90.0% respectively. However, since the method utilize the rectangular cropped ROI (region of interest) in the cervigram images for feature extraction, it will either select unnecessary additional regions outside of the cervix or not select some important cervix regions. An essential requirement to the computer aided task of investigating of cervix cancer is segmentation. Previous studies[15)–19)]have highlighted that this segmentation step allows for better representation of the interest region and computational methods can provide suitable results for identifying cervix regions. So far as the authors know, precancerous lesion detection using the above segmentation-based machine learning methods (not deep learning) are between 65.10–82.46%(0.651–0.825), 69.78–83.21% and 54.00–94.79% respectively and therefore can be improved upon.

In this work we aim to achieve relatively comparably high accuracy in precancerous cervical lesion classification based on simple and proper image processing. In particular, we target over 3,600 real cervical images and newly investigate a small number of effective and useful features for diagnosing cervical cancer. We previously proposed a classification framework for melanoma detection[20], and we believe a similar approach is also effective to this work. We use GrabCut segmentation on the region of interest (ROI) in cervigram images and calculate 44 features based on color and texture information from the ROI by using typical image processing techniques. Then we analyze which feature important is and how many the number of features is for correct classification by using several ensemble machine learning approaches, and finally we show from the experiments we did that only 10–13 features yield satisfactory results to determine most of precancerous cases.

## 2. Proposed Scheme

The proposed our framework has four major steps: image acquisition, segmentation, feature extraction and classification as illustrated in **Fig.1**.

### 2.1 Image acquisition

The images used for this study were provided from the National Cancer Institute (NCI) of the United States. (We could obtain test images based on the agreement for the transfer of de-identified cervical images between NCI and our university.) The data-set used here consist of 3,637 images from 1,819 patients. Among of them 2,841 images are of normal (no risk) and CIN1 (low risk) treated as our " Negatives", and 796 images of CIN2, CIN3 and CIN4 as our " Positives" (medium risk, high risk and cancer risk, respectively).

The original images have the cervicography medical equipment from which they are captured from in the background of the images as shown in **Fig.2**. Therefore, effective segmentation is required to eliminate them and outer cervix lesions but still preserve the affected part.

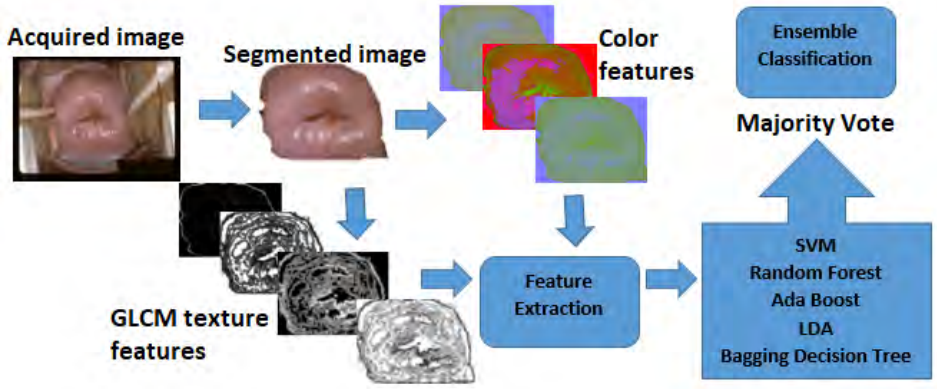



**Fig. 1** Flow diagram of our proposed algorithm

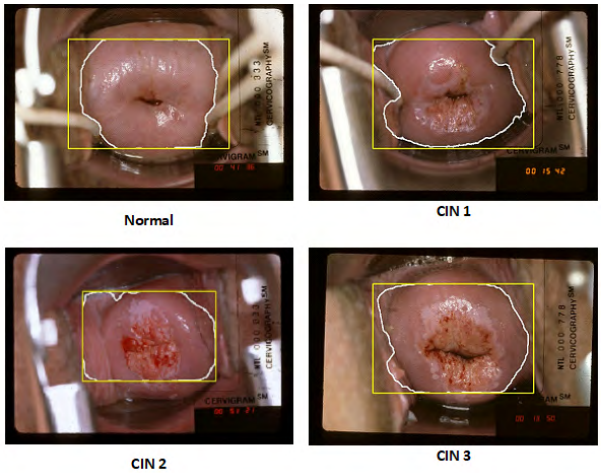



**Fig. 2** Sample of images with segmented regions of interest

### 2.2 Mask generation for GrabCut

Effective segmentation of the important regions for distinguishing cervix cancer from non-cervix cancer is very important but challenging because of similarities in structure and appearance[10]. For the acquired images segmentation is performed to separate the cervix region (foreground) from the outer region (background). We use GrabCut segmentation[21]for this purpose to capture as much of the foreground region as possible. It is derivative from graph cuts[22]with the initially identified information about the foreground and the background represented by a rectangular selection around ROI.

The affected part is located and covers most of the central region in almost all the acquired images, but the region of the affected part vary for each image. In previous studies involving cervix segmentation, semi-manual methods have been adopted[23),24]. Therefore, to eliminate the need for user intervention to determine the desired rectangle (ROI) around the foreground object, we create a segmentation image mask by morphing randomly

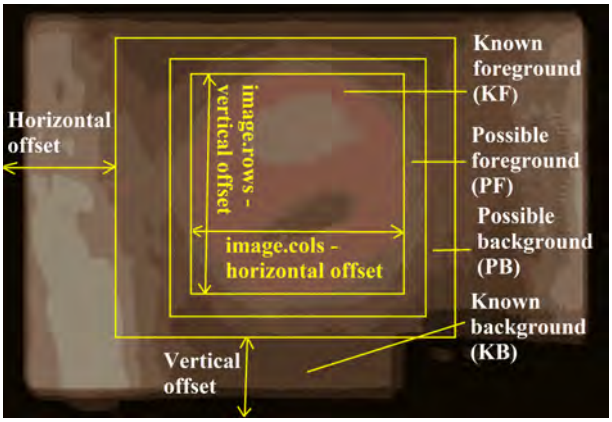



**Fig. 3** Colored quantized image and our defined ROI

selected images from our database.

Morphing is achieved by taking two of our color images $\boldsymbol{f}_u$ and $\boldsymbol{f}_v$. We create an intermediate morphed image $\boldsymbol{f}_{\mathrm{mor}}$ using the following equation

$$\boldsymbol{f}_{\mathrm{mor}}(i,j) = (1-\alpha)\boldsymbol{f}_u(i,j) + \alpha\boldsymbol{f}_v(i,j), \tag{1}$$

where $(i,j)$ represents pixel position and $\alpha$ is a parameter which controls the merger rate between $\boldsymbol{f}_u$ and $\boldsymbol{f}_v$. In RGB color space, the vector function $\boldsymbol{f}$ can be denoted $\boldsymbol{f}(i,j) = (\mathrm{R}(i,j), \mathrm{G}(i,j), \mathrm{B}(i,j))$, where R, G and B are the red, green and blue components, respectively.

We used $\alpha = 0.5$ and applied the morphing process Eq.(1) iteratively to 100 of our randomly selected images $\boldsymbol{f}_u$ and $\boldsymbol{f}_v$, then we took the average of 100 morphed images. Since the resulting averaged image being too noisy, we additionally performed color quantization using k-means clustering[25]to obtain the quantized image. The quantized image is shown in **Fig.3**. Based on this we manually determined four ROIs as shown in Fig.3; areas specified with high probability of known foreground (KF), known background (KB), possible foreground (PF) and possible background (PB). The reason why we chose these areas is that in most of our image data the fore-

ground area is inside of KF and the background is outside of KB, but in some of data the foreground or background is a little larger (inside of PF or outside of PB). Therefore we make an adjustment and use any two of ROIs as the masks of the foreground and background in the subsequent GrabCut segmentation, as necessary.

### 2.3 Cervix region segmentation using GrabCut

As already described, GrabCut[21]is an extension of graph cut algorithm used as color image segmentation method. We use the GrabCut to segment cervix region from every image data. For the details of GrabCut algorithm, please refer to the literature[21].

For each of pixels of an input image either foreground (cervix) or background (non-cervix) is labeled in advance while referring to the masks KF and KB. Also KF is regarded as an initial specified area including cervix region.

Note that GrabCut usually requires user inputs by drawing rectangles that provides three rough regions, i.e. foreground, background, and the unknown areas. But our method already estimates the four ROIs (KF, KB, PF, and PB) according to the procedure described above, and thus the method can designate possible regions easily. In other words it is easy to do the task of labelling even for a non-expert, that is, the need to directly draw the rectangles on input image for specifying rough regions is eliminated.

Then we input labeled pixels and the initial region into GrabCut algorithm. The algorithm estimates an initial Gaussian mixture distribution (GMM) derived from color information of the foreground and background pixels, and then assigns one of distributions with maximum likelihood among GMM individually to every foreground pixel. The parameters of GMM are updated after finishing all assignments, and background pixels are separated and extracted from the foreground by using minimum cut algorithm[22]. The above procedure is repeated by a several times, the GrabCut finally outputs well segmented foreground.

As shown in **Fig.4**, the affected part is well segmented by applying the GrabCut with our ROI. It should be noted that when an obtained segmentation result is not so good, we can perform the GrabCut again by replacing the masks KF, KB with PF, PB and the initial specified area KF with PF.

In this study, almost all tested images are well segmented by selecting ROIs appropriately. However, in some few cases, the result becomes either under- or over-segmentation due to noise in the image from tissue, characters, marks or light reflection as shown in **Fig.5**. In Fig.5 the automatic segmented region of our proposed model is indicated by the white boundary. Only if such cases, we manually modify the segmented boundary to be appropriate region indicated by the yellow line as shown in Fig.5.

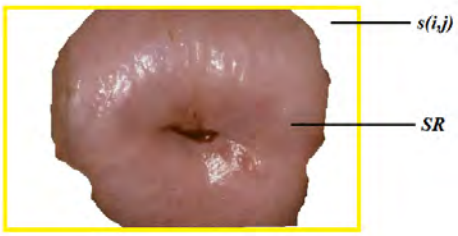



**Fig. 4** Example of segmentation results

A. Undersegmentation B. Oversegmentation

**Fig. 5** Examples of poor segmentation results

To verify the effectiveness of our segmentation in more detail, we preliminarily examined how accurate the GrabCut detects cervix region pixels in an image. For this purpose, we first randomly selected 1,000 images (500 positives and 500 negatives) and then manually segmented each correct cervix region by hand-drawing the boundary one by one. Then we regarded those segmented regions as the correct foregrounds and compared them with pixels identified as the part of foregrounds in GrabCut images by counting overlapping pixels. As a result GrabCut detected correct foreground pixels accurately about 87.1%, which is the average of 1,000 images. We think this fact shows a potential segmentation ability of GrabCut and also it suggests our method can capture sufficient lesions for color and texture feature extraction, as it is necessary later to consider both measures together.

### 2.4 Feature extraction

To determine what features are to be extracted, we use the visual inspection with acetic acid (VIA) screening of cervical cancer[3]as a guide. In VIA, the degree of whiteness and degree of pinkish are observed and therefore we consider color features. Moreover, the lesion smoothness and appearance are also noticed in VIA so we also con-

**Table 1** Extracted features based on color

| Feature | | Description |
|---|---|---|
| $x_1$ - $x_4$ | Entropy | $\sum_{k=0}^{255}\sum_{\mathrm{ch}\in\{\mathrm{R,G,B}\}} H_{\mathrm{ch}}(k)\ln H_{\mathrm{ch}}(k)$, where $H_{\mathrm{ch}}(k)$ is the histogram of channel ch of $\boldsymbol{s}(i,j)$ and $k$ is a level of pixel value integers in $[0,255]$. In RGB color space, the subscript ch means one of R,G,B channels. $x_2, x_3, x_4$ are calculated in the same manner for HSV, LAB and $\mathrm{YC_rC_b}$ color spaces, respectively. |
| $x_5$ - $x_8$ | Variance | $\frac{1}{N^3}\sum_{(i,j)\in\mathrm{SR}}\sum_{\mathrm{ch}\in\{\mathrm{R,G,B}\}}(s_{\mathrm{ch}}(i,j)-m)^2$, where $N$ is the number of pixels inside of SR, $s_{\mathrm{ch}}(i,j)$ is the value of channel ch of $\boldsymbol{s}(i,j)$ and $m$ is the mean calculated by $\frac{1}{N^3}\sum_{(i,j)\in\mathrm{SR}}\sum_{\mathrm{ch}\in\{\mathrm{R,G,B}\}} s_{\mathrm{ch}}(i,j)$. $x_5$ is calculated from RGB color space, and $x_6, x_7, x_8$ are calculated in the same manner for HSV, LAB and $\mathrm{YC_rC_b}$ color spaces, respectively. |
| $x_9$ - $x_{12}$ | Skewness | $M_3/(M_2)^{3/2}$, where $M_2$ is the second moment which is already calculated as $x_5$ (in case of RGB space). $M_3$ is the third moment and calculated by $M_3 = \frac{1}{N^3}\sum_{(i,j)\in\mathrm{SR}}\sum_{\mathrm{ch}\in\{\mathrm{R,G,B}\}}(s_{\mathrm{ch}}(i,j)-m)^3$. $x_9$ is calculated from RGB color space, and $x_{10}, x_{11}, x_{12}$ are calculated in the same manner for HSV, LAB and $\mathrm{YC_rC_b}$ color spaces, respectively. |
| $x_{13}$ - $x_{16}$ | Kurtosis | $M_4/(M_2)^2$, where $M_4$ is the fourth moment and calculated by $M_4 = \frac{1}{N^3}\sum_{(i,j)\in\mathrm{SR}}\sum_{\mathrm{ch}\in\{\mathrm{R,G,B}\}}(s_{\mathrm{ch}}(i,j)-m)^4$. $x_{13}$ is calculated from RGB color space, and $x_{14}, x_{15}, x_{16}$ are calculated in the same manner for HSV, LAB and $\mathrm{YC_rC_b}$ color spaces, respectively. |
| $x_{17}$ - $x_{28}$ | Average pixel intensity | $\frac{1}{N}\sum_{(i,j)\in\mathrm{SR}} s_{\mathrm{ch}}(i,j)$, where ch is one of channels RGB, HSV, LAB and $\mathrm{YC_rC_b}$ color spaces. $x_{17}$ is calculated regarding R channel, $x_{18}$ is G channel, ... and $x_{28}$ is $\mathrm{C_b}$ channel. |

sider texture features.

Hereafter, the area segmented as foreground (cervix region) is denoted by SR and the part of original image within SR's circumscribed rectangle by $\boldsymbol{s}(i,j)$ as shown in Fig.4. Note that $\boldsymbol{s}(i,j)$ is a vector function since it is a color image. In this study we extract 44 lesions properties from $\boldsymbol{s}(i,j)$, 28 are color-based and 16 texture-based. All features are calculated by using typical image processing techniques and every feature is described by a scalar value. These values are denoted by $x_i,\ i = 1,\ldots,44$ in this paper.

'Color features' are firstly calculated from $\boldsymbol{s}(i,j)$. We consider not only RGB as color space but HSV, LAB and $\mathrm{YC_rC_b}$ as suitable color spaces, that can provide better information for differentiating between lesions, differ as reported in many studies[13].

To analyze the cervix lesions we focus on color statistical measures; the mean, variance, skewness, kurtosis and entropy. Furthermore, we apply the individual separated channels of the four color spaces and analyze the statistical measure of average pixel intensity. Details of color-based features how to calculate and extract are summarized in **Table 1**.

'Texture features' are derived from gray-scale image $s_g(i,j)$, which can be calculated by using NTSC coefficients as follows:

$$s_g(i,j) = 0.299\cdot\mathrm{R}(i,j)+0.587\cdot\mathrm{G}(i,j)+0.114\cdot\mathrm{B}(i,j). \quad (2)$$

Note that each value of $s_g(i,j)$ is rounded to integer in $[0,255]$. We adopt the gray-level co-occurrence matrix (GLCM)[26]to examine texture features. GLCM is a well-known techniques for analyzing various texture measures and widely used for image classification.

The matrix element $Q(l_i,l_j|\Delta_x,\Delta_y)$ can be derived from $s_g(i,j)$ and it means the relative frequency with which two pixels, separated by a pixel distance $(\Delta_x,\Delta_y)$, occur within a given neighbourhood, one with gray level intensity $l_i$ and the other with $l_j$. Note that the range of $l_i$ and $l_j$ is $[0,255]$. In the case of using the notation $Q(l_i,l_j|d,\theta)$ each element contains the second order statistical probability for changes between gray levels $l_i$ and $l_j$ at a particular displacement distance $d$ and at a particular angle $\theta$. In this study we set $d=1$ and choose four directions $\theta = 0°, 45°, 90°, 135°$, i.e., the matrices are calculated in $(1,0°)$, $(1,45°)$, $(1,90°)$ and $(1,135°)$ respectively. Then we take average of the calculated results and denote it by $Q(l_i,l_j)$ again.

According to the matrix $Q(l_i,l_j)$ obtained, many textural features can be defined. The features on texture we used are summarized in **Table 2**.

After calculating our features, the ranges of values for the derived features $x_1$ to $x_{44}$ vary widely. Since feature values with wider range will cause misclassification in the latter cancer prediction process, we normalize every $x_i$ as follows:

$$x_i \leftarrow (x_i - \min\{x_i\})/(\max\{x_i\} - \min\{x_i\}). \quad (3)$$

**Table 2** Extracted features based on texture (GLCM)

| Feature | | Description |
|---|---|---|
| $x_{29}$ | Average | $\sum_{l_i=0}^{255} l_i \dot{Q}_x(l_i) (\equiv \mu_x)$, where $Q_x(l_i)$ is marginal probability matrix calculated by $\sum_{l_j=0}^{255} Q(l_i, l_j)$. |
| $x_{30}$ | Contrast | $\sum_{k=0}^{255} k^2 Q_{x-y}(k)$, where $Q_{x-y}(k)$ is calculated by $\left\{\sum_{l_i,l_j=0}^{255} Q(l_i, l_j)\right\}_{\lvert l_i - l_j \rvert = k}$. |
| $x_{31}$ | Variance | $\sum_{l_i,l_j=0}^{255} (l_i - \mu_x)^2 Q(l_i, l_j)$ |
| $x_{32}$ | Energy | square root of the angular second moment $x_{33}$. |
| $x_{33}$ | Angular second moment | $\sum_{l_i,l_j=0}^{255} Q(l_i, l_j)^2$ |
| $x_{34}$ | Correlation | $\frac{1}{\sigma_x \sigma_y} \sum_{l_i,l_j=0}^{255} \{l_i l_j Q(l_i, l_j) - \mu_x \mu_y\}$, where $\mu_y$ is the average calculated by $\sum_{l_j=0}^{255} l_j \cdot Q_y(l_j)$, and $Q_y(l_j)$ is calculated by $\sum_{l_i=0}^{255} Q(l_i, l_j)$. Further, $\sigma_x$ and $\sigma_y$ are the standard deviations respectively calculated by $\sigma_x^2 = \sum_{l_i=0}^{255} (l_i - \mu_x)^2 Q_x(l_i)$ and $\sigma_y^2 = \sum_{l_j=0}^{255} (l_j - \mu_y)^2 Q_y(l_j)$. |
| $x_{35}$ | Sum average | $\sum_{k=0}^{2\cdot 255} k \cdot Q_{x+y}(k)$, where $Q_{x+y}(k)$ is calculated by $\left\{\sum_{l_i,l_j=0}^{255} Q(l_i, l_j)\right\}_{\lvert l_i + l_j \rvert = k}$. |
| $x_{36}$ | Sum entropy | $-\sum_{k=0}^{2\cdot 255} Q_{x+y}(k) \log\{Q_{x+y}(k)\} (\equiv e_S)$. |
| $x_{37}$ | Sum variance | $\sum_{k=0}^{2\cdot 255} (k - e_S)^2 Q_{x+y}(k)$ |
| $x_{38}$ | Entropy | $-\sum_{l_i,l_j=0}^{255} Q(l_i, l_j) \log\{Q(l_i, l_j)\}$ |
| $x_{39}$ | Inverse difference moment | $\sum_{l_i,l_j=0}^{255} \frac{1}{1+(l_i - l_j)^2} Q(l_i, l_j)$ |
| $x_{40}$ | Homogeneity | $\sum_{l_i,l_j=0}^{255} \frac{1}{1+\lvert l_i - l_j \rvert} Q(l_i, l_j)$ |
| $x_{41}$ | Difference variance | variance of $Q_{x-y}(k)$. |
| $x_{42}$ | Difference entropy | $-\sum_{k=0}^{255} Q_{x-y}(k) \log\{Q_{x-y}(k)\}$ |
| $x_{43}$ | Dissimilarity | $\sum_{l_i,l_j=0}^{255} Q(l_i, l_j) \lvert l_i - l_j \rvert$ |
| $x_{44}$ | Inertia | $\sum_{l_i,l_j=0}^{255} Q(l_i, l_j)(l_i - l_j)^2$ |

That is, we re-scale each feature $x_i$ to a new range [0,1].

### 2.5 Ensemble classification

To examine which features $x_i$ are best for cervical cancer classification, we use and evaluate with machine learning algorithm. Particularly it has been reported that ensemble approaches based on majority voting using multiple learning algorithms usually achieve better performance[27], and therefore we also consider ensemble methods by stacking different learning algorithms.

As cervical cancer detection is a two class problem where there are two possible classes of the outcome "cancerous" (positive) or "non-cancerous" (negative), the class of unknown feature vector $\boldsymbol{x}$ belongs to either positive or negative. We determine the class of $\boldsymbol{x}$ by simply voting combination denoted as:

$$\text{class}(\boldsymbol{x}) = \begin{cases} \text{Positive} & \text{for } \sum_k y_k(\boldsymbol{x}) > 0, \\ \text{Negative} & \text{for } \sum_k y_k(\boldsymbol{x}) < 0. \end{cases} \tag{4}$$

where $y_k(\boldsymbol{x})$ is the $k^{\text{th}}$ classifier and outputs either 1 when $\boldsymbol{x}$ is estimated as positive or -1 as negative.

We aim to train and predict whether a class-unknown image is positive or negative based on our extracted features. For such purposes we select the following supervised learning models as classifiers $y_k$: linear discriminant analysis[28](LDA), support vector machine[29](SVM), AdaBoost[30](AB), random forest[31](RF), and bagging with decision tree[32](BDT). These methods have shown to be important and popular algorithms widely used in computer aided diagnosis for image classification and have been effective for many complex problems[27]. Note that the last three algorithms are known as ensemble classifier in even single use, but in this study we consider some combinations of the above five learning models and make decision based on majority voting from classification results obtained by multiple algorithms used.

All of these algorithms are implemented in scikit-learn libraries[33], and we utilize them in the subsequent classification experiments.

## 3. Experiments

### 3.1 Evaluating model

As already described, cervical cancer detection is a two class problem. For the experiments in this section we divided the acquired 3,637 images (2,841 negatives and 796 positives) into two sets in advance; training set with 70% of the data 2,545 images (1,988 negatives and 557 positives) and testing set with 30% of the data 1,092 images (853 negatives and 239 positives). Then we evaluated the prediction accuracy for testing data. Furthermore, to support the number of true instances we set the parameter average to weighted, due to the in-balance in our data-set having more negatives than positives samples. We then measured the performance of our model based on the accuracy defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}, \tag{5}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively. In our experiments, TP refers to the number of cervix cancer detected in cervix cancer images, TN refers to the number of non-cervix cancer detected in non-cervix cancer images, FP refers to the number of cervix cancer detected in non-cervix cancer images, and FN refers to the number of non-cervix cancer detected in cervix cancer images.

As our training data size is not so large (2,545 images) and we want results to generalize for unseen data as well, we evaluate by applying 10-folds cross-validation[34] by randomly splitting the training data-set into 10 folds, 9 folds are used for model training and 1 fold is used for testing. This processing is repeated 10 times by replacing testing data with one of the other folds, and then we take the average of 10 estimate results as the final accuracy. We apply similar cross-validation for all subsequent evaluation processes.

### 3.2 Determination of parameters for classification

First of all we need to determine parameters of some of machine learning algorithms we will use. For this purpose we executed exhaustive grid searches.

We first examined parameters of SVM (support vector machine). We decided to adopt Gaussian radial basis function (RBF) kernel for SVM, because preliminarily experiment results further confirmed better performance with RBF kernel compared to linear and polynomial kernels. Further to optimize SVM's performance we adjusted hyper parameters, i.e., cost parameter $C$ and kernel parameter $\gamma$. We searched for the values in {0.001, 0.01, 0.1, 1, 10, 100}, and found that $C = 1$ and $\gamma = 0.1$ give the best result.

Next we investigated parameters of AB (AdaBoost). Since we selected the decision tree classifier as the base estimator (weak classifier) due to the default of scikit-learn, we searched for the depth of each decision tree in {1,2,3,4,5}. Also we searched for the values in {100, 200, 500, 1,000, 2,000, 5,000, 10,000, 20,000} to determine the number of weak classifiers, and we chose 2 as depth and $1,000$ as the number of weak classifiers.

In the same way we also optimized in RF (random forest) and BDT (bagging with decision tree) by searching for the number of trees. We found that $1,000$ and 500 trees produce the best result in RF and BDT, respectively. Finally, LDA (linear discriminate analysis) is optimized by searching for the solver in {'svd', 'sqr', 'eigen'}, due to the default of scikit-learn, we searched for the tolerance (tol) in {1, 0.1, 0.01, 0.001, 0.0001, 0.00001}, and found that solver='svd' and tol=0.1 give the best result.

**Table 3** Parameters used for experiments

| Classifier | Parameters |
|---|---|
| LDA | solver= 'svd', tol=0.1 |
| SVM | $C = 1$, $\gamma = 0.1$ |
| AB | max depth=2, weak classifiers=1,000 |
| RF | trees=1,000 |
| BDT | trees=500 |

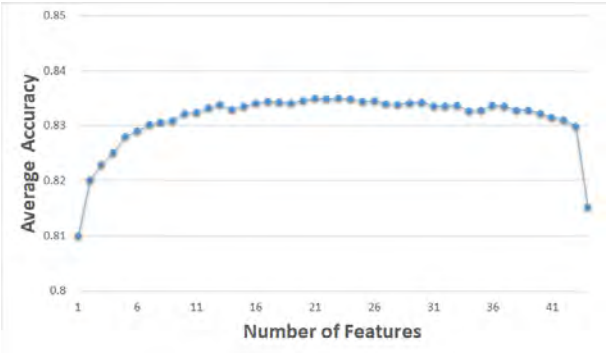



**Fig. 6** Sequential backward selection (SBS) of features

All parameters chosen from our experiments are summarized in **Table 3**.

### 3.3 Feature selection

Our aim here is to select the most useful features from those described in Table 1 and 2. All the derived features may not be of importance in cervix cancer classification and also the higher number of features will lead to an over fitting easily.

We used sequential backward selection (SBS)[33] to determine how many number of features that could be important. SBS starts from the full set of 44 features $Y = \{x_1, x_2, \ldots, x_{44}\}$ and returns a subset of features $X_{N_f} = \{x_j | j \in 1, 2, \ldots, N_f; x_j \in Y\}$; where $N_f (< 44)$ is the pre-defined number of selected features. The results are shown in **Fig.6**. From the Fig.6 we can see that performance improves from 10–13 features and peaks at 16-21 features. Therefore we concluded that ranges of 10–21 features are sufficient for cervix cancer classification.

We further analyzed the color and texture features using K-best and F-test univariate feature selection (provided scikit-learn library) to rank the features in terms of importance. Univariate feature selection examines each feature individually to determine the strength of the relationship of the feature with the response variable, and these methods are simple and are in general particularly

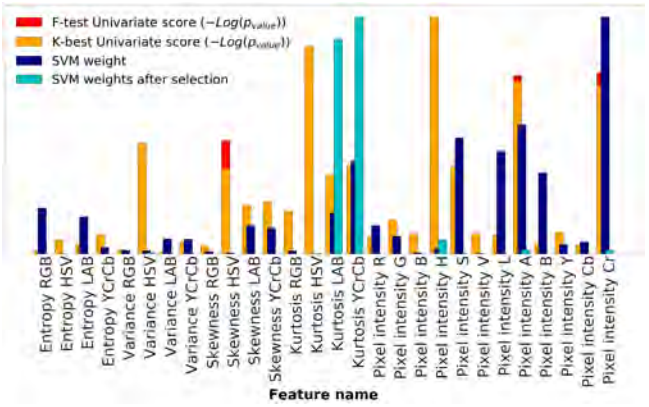



**Fig. 7** Ranking of color-based features using univariate selection

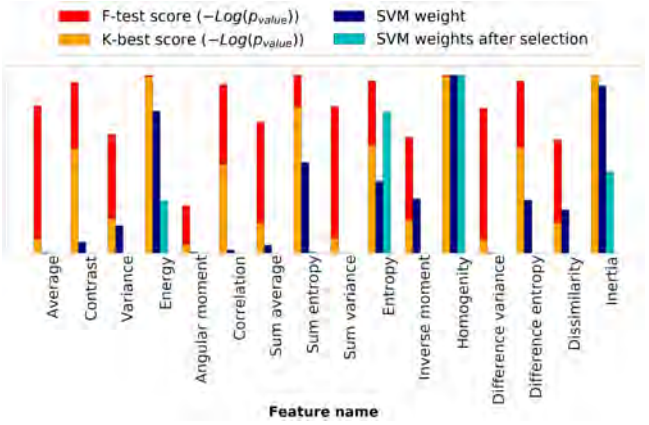



**Fig. 8** Ranking of texture-based features using univariate selection

good for gaining a better understanding of data. (For more details, please refer to the literature[33].)

Results of the color- and texture-based feature rankings are shown in **Fig.7** and **Fig.8** respectively. In the bar graphs p-values for the univariate feature selection and the corresponding weights of SVM are both plotted, and those having higher score can be regarded as informative and significant features. We found that the 21 features of which 13 are from color features: $x_1$, $x_3$, $x_6$, $x_{10}$, $x_{14}$, $x_{15}$, $x_{16}$, $x_{20}$, $x_{21}$, $x_{23}$, $x_{24}$, $x_{25}$, $x_{28}$, and 8 from texture features: $x_{30}$, $x_{31}$, $x_{32}$, $x_{38}$, $x_{40}$, $x_{42}$, $x_{43}$, $x_{44}$. These are probably important for classification.

Then we examined the best combinations of the above 21 features on each learning algorithm and the result is shown in **Table 4**. Table 4 (a) shows 10 best features selected with each algorithm, while Table 4 (b) shows features selected two times or more throughout (bold text means trice or more times occurrence). We found 13 important features with multiple occurrences across our five different classifiers: 10 features occurred more than twice are Entropies RGB ($x_1$), LAB ($x_3$), Variance LAB ($x_6$), Kurtosis HSV ($x_{14}$), Average pixel intensities a* ($x_{24}$), b* ($x_{25}$), $C_b$ ($x_{28}$) channels, Energy ($x_{32}$), Entropy ($x_{38}$), Homogeneity ($x_{40}$), and 3 features occurred twice are Kurtosises LAB ($x_{15}$), $YC_rC_b$ ($x_{16}$), Contrast ($x_{30}$). These 13 features indicate the suitability for classification and can be selected to obtain favorable results.

**Table 4** Best feature combinations

(a)

| Classifier | Features combination |
|---|---|
| LDA | $\boldsymbol{x_1}$, $\boldsymbol{x_3}$, $\boldsymbol{x_6}$,$\boldsymbol{x_{14}}$, $x_{15}$, $\boldsymbol{x_{24}}$,$\boldsymbol{x_{25}}$,$x_{30}$, $\boldsymbol{x_{32}}$, $\boldsymbol{x_{40}}$ |
| SVM | $\boldsymbol{x_1}$, $\boldsymbol{x_6}$, $x_{10}$, $\boldsymbol{x_{16}}$, $\boldsymbol{x_{24}}$, $\boldsymbol{x_{25}}$, $\boldsymbol{x_{28}}$, $x_{30}$, $\boldsymbol{x_{32}}$, $\boldsymbol{x_{38}}$ |
| AB | $\boldsymbol{x_1}$, $\boldsymbol{x_3}$, $\boldsymbol{x_6}$, $\boldsymbol{x_{24}}$, $\boldsymbol{x_{28}}$,$\boldsymbol{x_{32}}$, $\boldsymbol{x_{38}}$, $\boldsymbol{x_{40}}$,$x_{42}$, $x_{44}$ |
| RF | $\boldsymbol{x_3}$, $\boldsymbol{x_6}$, $\boldsymbol{x_{14}}$,$x_{15}$, $\boldsymbol{x_{24}}$, $\boldsymbol{x_{25}}$, $\boldsymbol{x_{28}}$, $\boldsymbol{x_{32}}$, $\boldsymbol{x_{38}}$, $\boldsymbol{x_{40}}$ |
| BDT | $\boldsymbol{x_1}$, $\boldsymbol{x_3}$, $\boldsymbol{x_6}$, $\boldsymbol{x_{14}}$, $x_{16}$, $\boldsymbol{x_{24}}$, $\boldsymbol{x_{28}}$, $\boldsymbol{x_{38}}$, $\boldsymbol{x_{40}}$, $x_{43}$ |

(b)

| Features with multiple occurrence | |
|---|---|
| trice or more | $\boldsymbol{x_1}, \boldsymbol{x_3}, \boldsymbol{x_6}, \boldsymbol{x_{14}}, \boldsymbol{x_{24}}, \boldsymbol{x_{25}}, \boldsymbol{x_{28}}, \boldsymbol{x_{32}}, \boldsymbol{x_{38}}, \boldsymbol{x_{40}}$ |
| twice | $x_{15}, x_{16}, x_{30}$ |

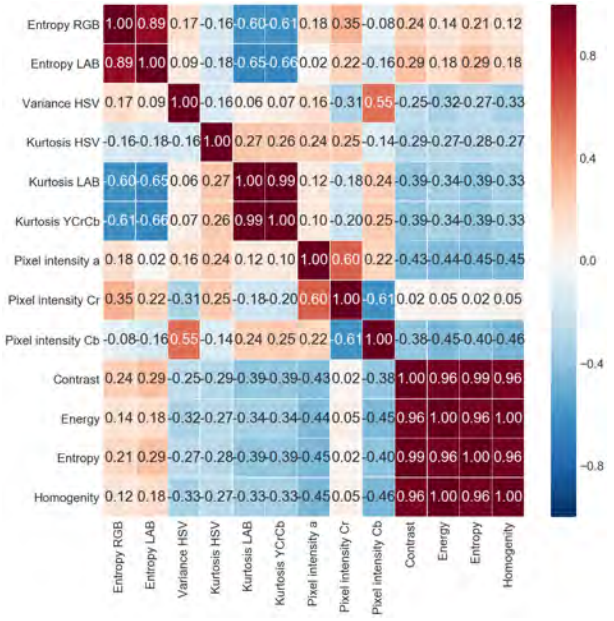



**Fig. 9** Heatmap of selected features

We further examined the relationship between the selected features using the heat map shown in **Fig.9**. In the map the redder, the higher relationship means and in contrast, the bluer, the lower. As it can be seen, the texture features are more correlated and fewer compared to the color features. Among the texture features Contrast ($x_{30}$) and Energy ($x_{32}$) are highly correlated but in our set of 10 features Contrast ($x_{30}$) does not occur. For the color features, Kurtosis LAB ($x_{15}$) and Kurtosis $YC_rC_b$ ($x_{16}$) are highly correlated with each other and both occur in the set of 13 features but not in the set of 10 features. Entropy RGB ($x_1$) and Entropy LAB ($x_3$) are also highly correlated with each other and both occur in the set of 10 features. Based on the above observations we drop

**Table 5** Three ensemble combinations we tested

| Combination | Classifiers |
|---|---|
| **1st** | **RF, AB, SVM** |
| 2nd | RF, AB, LDA |
| 3rd | RF, SVM, LDA |
| 4th | RF, AB, BDT |
| 5th | RF, SVM, BDT |
| 6th | RF, LDA, BDT |
| 7th | AB, SVM, LDA |
| 8th | AB, SVM, BDT |
| 9th | AB, LDA, BDT |
| 10th | SVM, LDA, BDT |

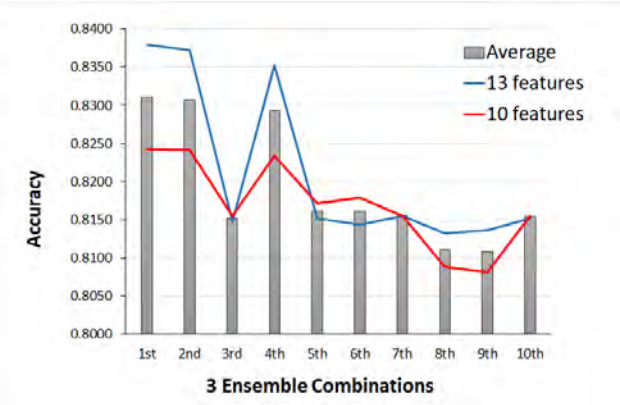



**Fig. 10** Comparison for 3 classifier combinations

Entropy RGB ($x_1$) for the set of 10 features and include Kurtosis LAB ($x_{15}$) instead, and then the new set of 10 features becomes Entropy LAB ($\boldsymbol{x}_3$), Variance HSV ($\boldsymbol{x}_6$), Kurtosis HSV ($\boldsymbol{x}_{14}$), Kurtosis LAB ($\boldsymbol{x}_{15}$), Pixel intensity a* channel ($\boldsymbol{x}_{24}$), Pixel intensity b* channel ($\boldsymbol{x}_{25}$), Pixel intensity $C_b$ channel ($\boldsymbol{x}_{28}$), Energy ($\boldsymbol{x}_{32}$), Entropy ($\boldsymbol{x}_{38}$), Homogeneity ($\boldsymbol{x}_{40}$). We decided to make use of the 10 (or 13) best features mainly in the following evaluation experiments.

### 3.4 Evaluation of ensemble approach using 3 classifier combination

As described in section 2.5, we adopt an ensemble approach for improvement of cervix cancer classification. To determine which is the best three-ensemble combination, we calculated the accuracy scores with 10 and 13 features for all 10 possible different combinations shown in **Table 5**. Obtained results are shown in **Fig.10**.

From the graph in Fig.10 we can see that 1st combination of 3 classifiers, i.e., random forest (RF), AdaBoost (AB) and support vector machine with RBF kernel (SVM) have the highest average accuracy among our tested combinations of 3 classifiers. We use this set of classifiers for further analysis.

### 3.5 Statistical analysis

We also performed statistical analysis for evaluation of our best feature combination. The analysis are specificity which expresses the ratio of correct positive observations, and sensitivity which is the ratio of correctly predicted positive events. They were respectively calculated as below:

$$\begin{aligned} \text{Specificity} &= \text{TP}/(\text{TP}+\text{FP}) \times 100\ [\%], \\ \text{Sensitivity} &= \text{TP}/(\text{TP}+\text{FN}) \times 100\ [\%]. \end{aligned} \tag{6}$$

Further, we calculated the F1 score which is the weighted average of specificity and sensitivity. The formula for F1 score is:

$$\text{F1} = 2 \times \left( \frac{\text{Specificity} \times \text{Sensitivity}}{\text{Specificity} + \text{Sensitivity}} \right), \tag{7}$$

it can give further insight into the accuracy of our model because it takes both false positives and false negatives into account.

We summarized all the calculated results for 10 features and 13 features in **Table 6** and **Table 7**, respectively. Since we did cross-validation 10 times with each classifier as already stated, the average and standard deviation is shown in each cell of the Tables. On the whole we can see from the tables that there is a tendency to show superiority of 3 and 5 ensemble classifiers compared with the other single ones.

Furthermore, we also calculated the receiver operating characteristic (ROC) curves for 3 and 5 ensemble classifier with 10 and 13 best features, and plotted them in **Fig.11** – **Fig.14**, respectively. ROC is known as a common and widely accepted method for analyzing performance of machine learning classification, and the area under the ROC curve (AUC) can be used to measure how well the method can distinguish two classes. The calculated AUC values are also shown in the above Tables 6 and 7. We can see from the figures and tables that 0.840 and 0.841 of the AUC is achieved using 10 features for 3 ensemble and 5 ensemble respectively, and 0.852 and 0.851 is achieved with 13 features respectively. Although using 13 features has slightly improved scores compared to 10 features, however, in general, there is very little statistical difference between the two approaches.

## 4. Conclusion

In this paper, we proposed a method for detecting cervical cancer from cervigraph images of affected cervix regions. Our method consist of typical image processing

**Table 6** Evaluation of classifiers using 10 best features

| Classifier | Sensitivity (%) | Specificity (%) | F1-Score | AUC |
|---|---|---|---|---|
| SVM RBF kernel | 85.0±1.8 | 81.4±2.7 | 0.824±0.024 | 0.841±0.027 |
| Random Forest | 82.2±4.0 | 82.3±2.1 | 0.822±0.021 | 0.840±0.021 |
| Ada Boost | 80.6±8.8 | 81.2±2.7 | 0.808±0.028 | 0.822±0.027 |
| **Three Ensemble** | **83.1**±1.5 | **82.5**±1.5 | **0.827**±0.015 | **0.840**±0.015 |
| LDA classifier | 86.3±2.0 | 81.6±2.6 | 0.828±0.029 | 0.840±0.026 |
| Bagging Tree | 81.6±2.0 | 82.1±2.0 | 0.818±0.020 | 0.830±0.020 |
| **Five Ensemble** | **83.7**±1.5 | **82.6**±1.4 | **0.830**±0.014 | **0.841**±0.014 |

**Table 7** Evaluation of classifiers using 13 best features

| Classifier | Sensitivity (%) | Specificity (%) | F1-Score | AUC |
|---|---|---|---|---|
| SVM RBF kernel | 83.6±2.3 | 81.4±3.1 | 0.827±0.024 | 0.848±0.031 |
| Random Forest | 82.4±1.9 | 82.7±2.0 | 0.825±0.019 | 0.851±0.023 |
| Ada Boost | 81.9±1.6 | 82.5±1.5 | 0.820±0.016 | 0.823±0.020 |
| **Three Ensemble** | **83.4**±1.6 | **83.1**±1.9 | **0.838**±0.018 | **0.852**±0.015 |
| LDA classifier | 85.9±2.3 | 81.4±2.7 | 0.825±0.02 | 0.836±0.028 |
| Bagging Tree | 82.8±1.2 | 83.3±1.2 | 0.829±0.02 | 0.844±0.013 |
| **Five Ensemble** | **83.9**±1.8 | **83.2**±1.8 | **0.834**±0.01 | **0.851**±0.018 |

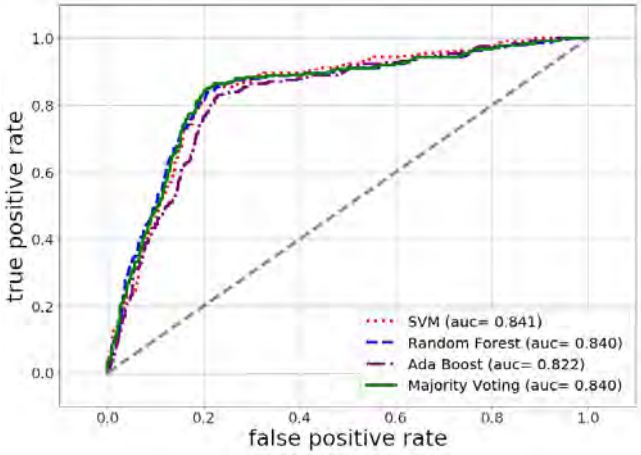



**Fig. 11** ROC of 3 ensemble classifier using 10 features (AUC=0.840)

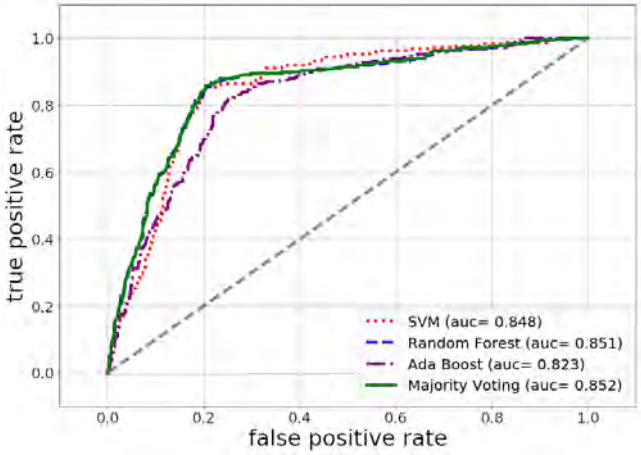



**Fig. 13** ROC of 3 ensemble classifier using 13 features (AUC=0.852)

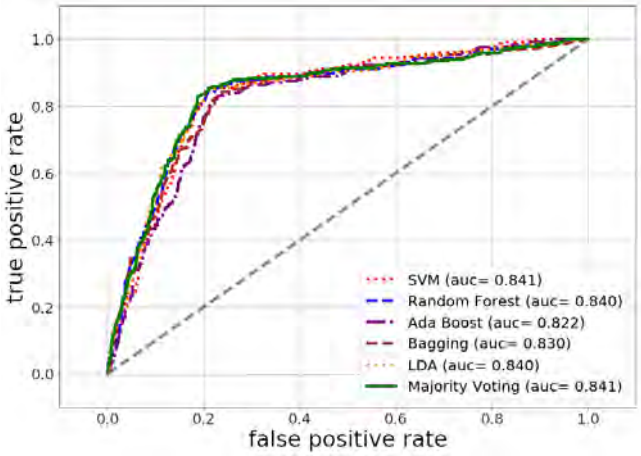



**Fig. 12** ROC of 5 ensemble classifier using 10 features (AUC=0.841)

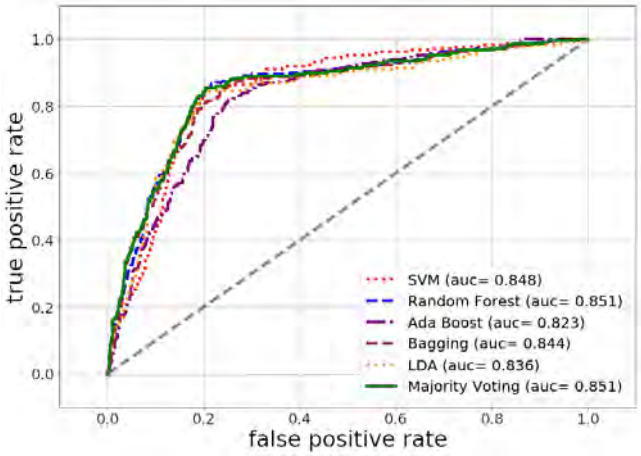



**Fig. 14** ROC of 5 ensemble classifier using 13 features (AUC=0.851)

techniques (GrabCut segmentation and features extraction) and ensemble classification model. Our proposed segmentation method can covers the affected area around the cervix and our classification experiments showed that

high accuracy can be achieved with a set of few useful features, comparative to visual inspection based on subjective opinion. The results demonstrated that better performance over 80% scores compared to VIA can be achieved through computer aided diagnosis. Particularly only 10–13 features yields satisfactory results sufficient to determine most cases and provide consistent information for determining CIN.

The use of ensembles classifier can improve the accuracy and reliability of classification. Further, the combination of 3 classifiers (random forest, AdaBoost ans support vector machine) gives the best result with very little statistical difference from using all 5 classifiers. This work also demonstrates computer aided diagnosis can potentially assist in cervical cancer diagnosis by requiring less training, fewer equipment and infrastructure, as well as fewer specialized medical personnel in remote areas around the world with high rates of cervical cancer.

The overall system can aid the doctor for making diagnosis decisions, providing quick second opinions and in prioritising patients to be attended to early due to the severity of their case. Once cancer is suspected referral to the Oncologist as early as possible is critical since early detection is important for treatment.

Future work could involve experimenting on other image data-set of varying features and qualities and improving the cervix lesion detection algorithm for eliminating the cervix boundary. Moreover, we need to examine how the segmented regions affect the classification accuracy. Also, feature selection such as using PCA (principal component analysis) for dimensional reduction of features and testing with other classification algorithms.

## Acknowledgments

We would like to acknowledge the support and advice from the National Cancer Institute (NCI) of the United States.

## References

1) J. Ferlay, I. Soerjomataram, M. Ervik, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D.M. Parkin, D Forman, F. Bray: "Cancer Incidence and Mortality Worldwide: Sources, Methods and Major Patterns in GLOBOCAN 2012", International Journal of Cancer, Vol.136, pp.E359–E386 (2015).
2) J. Jeronimo, P.E. Castle, R. Herrero, R.D. Burk, M. Schiffman: "HPV Testing and Visual Inspection for Cervical Cancer Screening in Resource-Poor Regions", International Journal of Gynecology and Obstetrics, Vol.83, pp.311–313 (2003).
3) H. S. Saleh: "Can Visual Inspection with Acetic Acid be Used as An Alternative to Pap Smear in Screening Cervical Cancer?", Middle East Fertility Society Journal, Vol.19, No.3, pp.187–191 (2014).
4) A. Sarwar, V. Sharma, R. Gupta: "Hybrid Ensemble Learning Technique for Screening of Cervical Cancer Using Papanicolaou Smear Image Analysis", Personalized Medicine Universe, Vol.4, pp.54–62 (2015).
5) K. Bora, M. Chowdhury, L. B. Mahanta, M. K. Kundu, A. K. Das: "Automated Classification of Pap Smear Images to Detect Cervical Dysplasia", Computer Methods and Programs in Biomedicine, Vol.138, pp.31–47 (2017).
6) T. Chankong, N. Theera-Umpon, S. Auephanwiriyakul: "Automatic Cervical Cell Segmentation and Classification in Pap Smears", Computer Methods and Programs in Biomedicine, Vol.113, No.2, pp.539–556 (2014).
7) H. A. Phoulady, M. Zhou, D. B. Goldgof, L. O. Hall, P. R. Mouton: "Automatic Quantification and Classification of Cervical Cancer via Adaptive Nucleus Shape Modeling", Proc. of the IEEE International Conference on Image Processing, pp.2658–2662 (2016).
8) R. Sankaranarayanan, L. Gaffikin, M. Jacob, J. Sellors, S. Robles: "A Critical Assessment of Screening Methods for Cervical Neoplasia", International Journal of Gynecology and Obstetrics, Vol.89, pp.S4–S12 (2005).
9) D. Song, E. Kim, X. Huang, J. Patruno, H. Munoz-Avila, J. Heflin, L. R. Long, S. K. Antani: "Multi-Modal Entity Coreference for Cervical Dysplasia Diagnosis", IEEE Trans. in Medical Imaging, Vol.34, No.1, pp.229–245 (2015).
10) Z. Xue, L. R. Long, S. Antani, J. Jeronimo, G. R. Thoma: "A Unified Set of Analysis Tools for Uterine Cervix Image Segmentation", Computerized Medical Imaging and Graphics, Vol.34, No.8, pp.593–604 (2010).
11) H. Acosta-Mes, N. Cruz-Ramirez, R. Hernandez-Jimenez: "Aceto-White Temporal Pattern Classification Using K-NN to Identify Precancerous Cervical Lesion in Colposcopic Images", Computers in Biology and Medicine, Vol.39, pp.778–784 (2009).
12) G. Zimmerman, H. Greenspan: "Automatic Detection of Specular Reflections in Uterine Cervix Image", Proc. of the International Society for Optical Engineering (SPIE 6144), Medical Imaging 2006: Image Processing, 61446E (2006).
13) T. Xu, H. Zhang, C. Xin, E. Kim, L. R. Long, Z. Xue, S. Antani, X. Huang: "Multi-Feature Based Benchmark for Cervical Dysplasia Classification Evaluation", Pattern Recognition, Vol.63, pp.468–475 (2017).
14) T. Xu, H. Zhang, X. Huang, S. Zhang, D. Metaxas: "Multimodal Deep Learning for Cervical Dysplasia Diagnosis", Proc. of the Medical Image Computing and Computer-Assisted Intervention (MICCAI 2016), pp.115–123 (2016).
15) J. Lui, L. Li, L. Wang: "Acetowhite Region Segmentation in Uterine Cervix Images Using Registered Ratio Image", Computers in Biology and Medicine, Vol.93, pp.47–55 (2018).
16) W. Li, J. Gru, D. Ferris, A. Poirson: "Automated Image Analysis of Uterine Cervical Images", Proc. of the International Society for Optical Engineering (SPIE 6514), Medical Imaging 2007: Computer-Aided Diagnosis, 65142P (2007).
17) S. Zhang, J. Huang, D. Metaxas, W. Wang, X. Haung: "Discriminative Sparse Representations for Rervigram Image Segmentation", Proc. of the IEEE International Conference on Biomedical Imaging, pp.133–136 (2010).
18) S. Gordon, S. Lotenberg, R. Long, S. Antani, J. Jeronimo, H. Greenspan: "Evaluation of Uterix Segmentation Using Ground Truth from Multiple Experts", Australasian Physical and Engineering Sciences in Medicine, Vol.33, pp.205–216 (2009).
19) B. Bai, P. Liu, Y. Du, Y. Luo: "Automatic Segmentation

of Cervical Region in Colposcopic Images Using K-Means", Australasian Physical and Engineering Sciences in Medicine, Vol.41, No.4, pp.1077–1085 (2018).

20) S. Mustafa, A. Kimura: "A SVM-Based Diagnosis of Melanoma Using Only Useful Image Features", Proc. of the International Workshop on Advanced Image Technology (IWAIT2018), pp.1–4 (2018).
21) C. Rother, V. Kolmogorov, A. Blake: "GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts", Proc. of the ACM Transactions on Graphics (SIGGRAPH), Vol.23, pp.309–314 (2004).
22) Y. Boykov, G. Funka-Lea: "Graph Cuts and Efficient N-D Image Segmentation", International Journal on Computer Vision, Vol.70, pp.109–131 (2006).
23) A. Alush, H. Greenspan and J. Goldberger: "Automated and Interactive Lesion Detection and Segmentation in Uterine Cervix Images", IEEE Trans. on Medical Imaging, Vol.29, No.2, pp.488–501 (2010).
24) Q. Ji, J. Engel, E. Craine: "Texture Analysis for Classification of Cervix Lesions", IEEE Trans. on Medical Imaging, Vol.19, No.11, pp.1144–1149 (2009).
25) O. Verevka, J. Buchanan: "Local K-Means Algorithm for Colour Image Quantization", Proc. of the Graphics/Vision Interface Conference, pp.128–135 (1995).
26) R.M. Haralick, K. Shanmugam, I. Dinstein: "Textural Features of Image Classification", IEEE Trans. on Systems, Man and Cybernetics, Vol.SMC-3, No.6, pp.610–621 (1973).
27) S. Dzeroski, B. Zenko: "Is Combining Classifiers Better than Selecting the Best One", Machine Learning, Vol.54, No.3, pp.255–273 (2004).
28) T. Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning: Data Mining, Inference and Prediction (2nd Edition), Springer, Section 4.3, pp.106–119 (2008).
29) C. Hsu, C. Chung, C. Lin: "A Practical Guide to Support Vector Classification", National Taiwan University Technical Report, pp.1–16 (2003).
30) J. Zhu, H. Zou, S. Rosset, T. Hastie: "Multi-class AdaBoost", Statistics and Its Interface, Vol.2, pp.349–360 (2009).
31) L. Breiman: "Random Forests", Machine Learning, Vol.45, No.1, pp.5–32 (2001).
32) L. Breiman: "Bagging Predictors", Machine Learning, Vol.24, No.2 (1996).
33) F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay: "Scikit-Learn: Machine Learning in Python", Journal of Machine Learning Research, Vol.12, pp.2825–2830 (2011).
34) S. Raschka: Python Machine Learning (1st Edition), Birmingham UK: Packt Publishing Ltd. (2016).

(Received October 24, 2018)
(Revised February 21, 2019)

**Suleiman MUSTAFA**
received both B.Sc. and M.Sc. degrees in Computer Science from the University of Hertfordshire in 2004 and 2006, respectively. He works with NASENI, Ministry of Science and Technology of Nigeria as a program Analyst. He is now a Ph.D Student at Iwate University, Japan.

**Akio KIMURA** (*Member*)
completed the master's program of Computer and Information Sciences, Graduate School of Engineering of Iwate University in 1993, and joined Sony Corporation. While at Sony, he was engaged in research and development of magnetic recording fields. In 1995, he joined Iwate University, and is now an associate professor of Faculty of Science and Engineering. He is engaged in research related to image processing and computer vision. He holds a D.Eng. degree, and is a member of IEICE, IPSJ, and the Society for Art and Science.

IIEEJ **Paper**

# Region Control in Stylized Shading Using Radial Transformation within Texture Projection

Muhammad ARIEF†, Hideki TODO††(*Member*), Koji MIKAMI†, Kunio KONDO†(*Honorary Member*)

†Tokyo University of Technology, ††Chuo Gakuin University

<**Summary**> Recent works in Non-Photorealistic Rendering (NPR) are capable of stylizing global appearance to fit shading designs by artists. Meanwhile, local appearance control is still required for the artists to depict geometric and emotional features. We propose a simple region control algorithm for an existing light-based texture projection technique. Since the shading effects of the texture projection cannot be overlayed, we consider to deform the original projection result by modifying texture coordinates. To enable artists to use familiar interface, we incorporate multiple lighting effects and painting control with the original light-based texture projection process. Our shading pipeline is implemented on GPU, which enables a real-time preview for the shading design at interactive rates. We also conducted user studies to evaluate the usability of our system.

**Keywords**: non-photorealistic rendering, stylized shading, texture projection

## 1. Introduction

Non-Photorealistic Rendering (NPR) could accent the imaginative impression to the viewers especially in cartoon animation due to the fictive story that is created for it. Artists can give imaginative impression by using stylized shading for the target 3D object. Most of 3D software provide lighting controls to change the impression of shading, which is a key factor for cinematic lighting. A lot of decorative patterns appears in stylized shading which is resulted from artist expression for portraying object features also emotions[1),2)]. The flexibility in the 2D environment gives artist to create more expressive artworks rather in 3D environment because artist does not need to consider many parameters especially for shading creation. This circumstances also need to be considered in term of creating the supporting tool for 3D stylized shading design.

Texture mapping is one of easiest way for creating hand-drawn appearance. Its simple process can also provide real-time interactions for shading design. In this paper, we focus on texture projection strategy of Lit-sphere[3)] and its extension[4)]. Thanks to the texture-based shading representation, the resulted shading includes a lot of unique features which the artist wants to preserve. While the global shading appearance is well controlled by the Lit-sphere approach, there are still significant needs for more detailed region controls.

**Figure 1** illustrates a typical multiple lighting effect designed in a 2D artwork. In Fig.1, multiple lighting effect is used for emphasizing muscle through the character body. The character is lit by several lights in order to emphasize features such as muscle in the chest and upper arm. For creating such a multiple lighting effect, we can consider to combine two texture projection of the Lit-sphere extension[4)]. However, **Fig. 2** shows that simple color overlay cannot work in this case since color transforms are applied via texture projection in advance.

**Figure 3** shows the imaginative shading shape on the character face where this shape is designed for accentuating the mysterious emotion and emphasizing shape fea-


**Fig. 1** Multiple lights in 2D artwork

tures. In Fig. 3, the left and middle picture has shading over the eyes and the nose give mysterious feeling to the character. Also round shape shading emphasizing the roundness of the head in the right picture. The artist may also need an ability to design such unrealistic shading in 3D stylized shading. Painting systems[1),5)] well handle this situation by modifying the original shading result produced from photorealistic illumination. Therefore, the integration of painting control with texture-based shading process has a potential ability to provide a more flexible design framework for the artists.

In this paper, we propose a simple shading design framework where artistic region controls can be applied to the original texture projection result. We start from considering the texture projection process of the Lit-sphere extension[4)]. Since the final shading color is affected by the relation between the input light direction and surface normal, the normal transform operator can change the underlying illumination. We will introduce a set of region control mechanisms based on the normal transform functions, while the texture-based shading style can be preserved. Comparison of three images are shown in Fig. 4. In Fig. 4, our method successfully create smmothly blended for creating multiple lighting effect. Comparing to previous method that only capable overlay texture thus produce unsmoothly blended texture. Thus, our contributions will be summarized as follows:

- Simple region control mechanisms that can be applied to the texture projection method.
- User interaction examples of multiple lighting and painting operation based on the region control mechanism.
- Simple but effective algorithms to fulfill both of stylized design and real-time interactions.

## 2. Related Work

### 2.1 Example-based stylization

Recent NPR researches focus on example-based stylization to synthesize stylized shading appearance from photorealistic illumination. Bénard et al. 2013[6)] used Image Analogies[7)] for automatic style transfer. Stylit[8)]


**Fig. 2** Multiple texture mapping limitation using extended Lit-sphere[4)]


**Fig. 3** Artistic shading shape in 2D artwork


(a) original shading (b) texture blending (c) our method

**Fig. 4** Comparison of multiple lighting effects

improved style transfer results by using more strict correspondence between rendering elements from light path expressions. Similar results can also be obtained by neural network[9)]. While these methods are capable of easy and quick stylization from a style example image, there are still significant needs for controlling the lighting shape to fulfil artistic requirements.

### 2.2 Stylized shading model

To consider lighting shape control, we review simpler shading models than example-based stylization. The most fundamental shading model is cartoon shading based on a simple 1D texture lookup process for style conversion. This 1D texture reference brought many style extensions such as illustrative rendering[10)] or dynamic stylized shading primitives[11)].

X-Toon[12)] extended the dimension of the texture reference by storing the secondary attribute effects in 2D texture. As for brush stroke style, early example-based stylization techniques for 2D texture representation[13),14)] are proposed to model brush stroke styles using samples from a painted image.

All of these stylized shading models mainly focus on global appearance derived from a 3D lighting result. Therefore, we need additional techniques for lighting shape control to fulfill artistic demands.

### 2.3 Painting system

Painting systems can provide effective user interfaces for artistic stylizations. Inverse design from user inputs is a common approach for painting systems. There are several inverse design systems to obtain the optimal illumination[15)–17)] or surface normal vector field[18)]. MatCap decomposition[19)] can also recovers expressive material

representation of the Lit-sphere[3].

Another research direction on painting system is to design strokes directly on the 3D model[20]. However, this strategy lacks the ability to animate the designed strokes by 3D lighting scheme.

User interface evaluation for 3D lighting design[21] concluded that direct interface in common 3D software such as Autodesk Maya ® and 3dsmax ® perform faster to achieve the desired design. Our method focuses on the creative side of the designed shading and also aim to integrate the painting scheme with the direct light source.

Local control through shading painting system[1] enables the user to add or erase the shading by local lighting on the 3D object. This method demonstrates seamless tweaking for the detailed shape within key-framing process. The highlight brush [5] can create various shape of highlight. While these methods are well integrated into the existing lighting scheme, the global appearance through cartoon shading is much simpler than our target style of hand-drawn shading style.

### 2.4 Texture projection

To consider the integration of hand-drawn shading style and lighting shape control, texture projection is a reasonable approach. The fundamental scheme is proposed by Lit-sphere[3], where the view-dependent shading appearance is designed with 2D texture reference. Todo et al. 2013[4] extended the original Lit-sphere approach to general lighting control with a single light source.

**Figure 5** demonstrates the comparison of stylized shading appearance. In, Fig. 5, the simple abstraction of realistic illumination by cartoon shading lacks the capability of making artistic features such as the irregular gradation in hand-drawn shading styles. The Lit-sphere extension can include irregular gradation which is commonly found in hand-drawn artworks, while the lighting control is still possible to design the lighting animation.

However, there are still technical challenges in lighting control for the Lit-sphere extension. Figure 2 shows the texture blending problem on multiple lighting effects, where two projected shading effects are overlapped through simple color overlay scheme. We will focus on achieving an intuitive lighting shape control mechanism while utilizing the expressive shading style through texture projection.


(a) cartoon shading (b) lit-sphere extension[4]

**Fig. 5** Comparison of stylized shading appearance

## 3. Shading Design via Texture Projection

In our approach, shading is designed as texture projection onto the target 3D model. To provide further flexibility in our approach, we provide two types of controls: global appearance and region control. Global appearance is determined by the common 3D lighting with key lighting elements. Region control enable the user to change the lighting area of the global appearance result through multiple lighting and painting operations.

### 3.1 Global appearance

We assume that a single key light source will generate the global appearance which consists of diffuse lighting and specular lighting. For the global appearance we consider to apply the light space normal definition described in the Lit-sphere extension [4]. With the light vector $\mathbf{L}(p) \in \mathcal{R}^3$ and the normal vector $\mathbf{N}(p) \in \mathcal{R}^3$ at the surface point $p$, light space normal $\mathbf{N}_L(p) \in \mathcal{R}^3$ is defined as follows:

$$\mathbf{N}_L(p) = (N_{Lx}(p), N_{Ly}(p), N_{Lz}(p)), \tag{1}$$

where $N_{Lx}(p) = \mathbf{N}(p) \cdot \mathbf{L}_x(p)$, $N_{Ly}(p) = \mathbf{N}(p) \cdot \mathbf{L}_y(p)$, and $N_{Lz}(p) = \mathbf{N}(p) \cdot \mathbf{L}(p)$ are the light-space normal coordinates spanned by the light coordinate frames $\mathbf{L}_x$, $\mathbf{L}_y$, and $\mathbf{L}$. **Figure 6** illustrates the light space normal representation.

The diffuse lighting effect is obtained by simple 2D texture sampling with $(u, v) = (N_{Lx}(p), N_{Ly}(p))$. **Figure 7** shows global lighting appearance. We used two difference textures for producing each type of lighting effects such as diffuse lighting, specular lighting. We can encode 2D shading details for the lighting effect. We also define the specular lighting effect effect through the simple modification of Eq. (1). Derived from phong reflectance term, we used the reflected view vector $\mathbf{V}'$ for the specular effect by replacing $\mathbf{N}$ in the definition of Eq. (1) to be in Eq. (2).

$$\mathbf{N}_s(p) = (N_{Sx}(p), N_{Sy}(p), N_{Sz}(p)), \tag{2}$$

where $N_{Sx}(p) = \mathbf{V}'(p) \cdot \mathbf{L}_x(p)$, $N_{Sy}(p) = \mathbf{V}'(p) \cdot \mathbf{L}_y(p)$, and $N_{Sz}(p) = \mathbf{V}'(p) \cdot \mathbf{L}(p)$ . These additional effects are

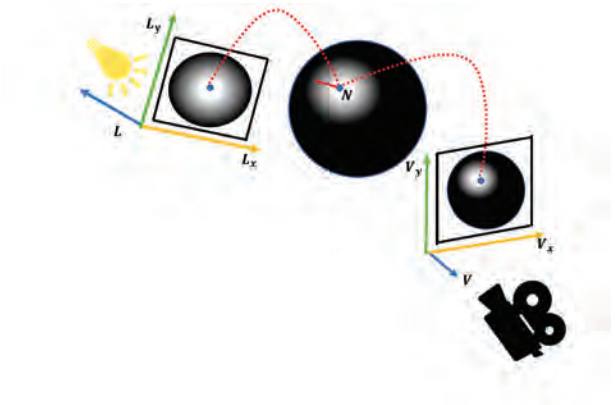



**Fig. 6** Light space normal

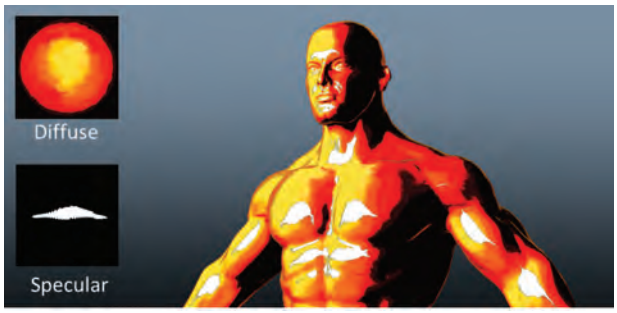



**Fig. 7** Global lighting appearance

rendered as layered effects. Final shading appearance is demonstrated in Fig. 7.

### 3.2 Region control

The main idea for region control is to modify the light-space normal coordinates depending on the target user interactions. **Figure 8** illustrates how we can change the final shading result via surface normal modifications. In Fig. 8, the modification of surface normal vector field will change the illumination result. The green curve shows the surface normal on the chest in the original lighting. The red curve shows the transformed surface normal through our region control operation. Our region control modify the normal direction to become closer to the designed light direction in this case, which leads the chest area becomes brighter.

As we can see Eq. (1) with $N_{Lz}(p) = \mathbf{N}(p) \cdot \mathbf{L}(p)$, $z$ coordinate of the light space normal is the key factor for the brightness which is computed as the usual 3D lighting models (e.g. Lambertian model for the diffuse lighting effect). Therefore, we consider to transform the $z$ coordinate to control the 2D texture mapping result for more detailed region controls. We incorporate the $z$ coordinate transformation $f : \mathcal{R}^3 \to \mathcal{R}^3$ into the light space normal $\mathbf{N}_L$ as follows:

$$f(\mathbf{N}_L(p)) = \mathbf{N}_L(p) + \sum_i w_i(p)(0, 0, 1), \tag{3}$$

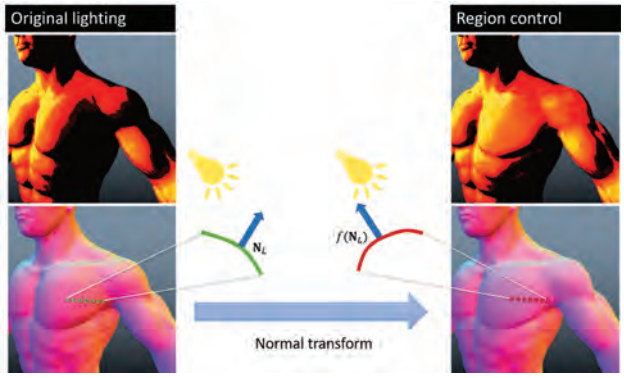



**Fig. 8** Surface normal vector transformation


**Fig. 9** Multiple lighting behavior

where the $z$ component in the $\mathbf{N}_L(p)$ is tweaked by the basic form of the weight function $w_i(p) \in \mathcal{R}$. Each $w_i(p)$ affects the texture sampling for the brightness axes (radial direction).

With the transformed normal $\mathbf{N}'_L = f(\mathbf{N}_L)$, we can finally obtain the unit-length normal $\widehat{\mathbf{N}'_L}$ for texture sampling process.

In our experiments, we apply this region control scheme for the following detailed lighting effects.

#### 3.2.1 Multiple lighting behavior

From the general formula of the region control in Eq. (3), we design the first weight function $w_1(p) = w_{ml}(p)$ as follows:

$$w_{ml}(p) = \sum_k I_{Lk} g(|\mathbf{x}(p) - \mathbf{x}_{Lk}|, R_k), \tag{4}$$

where $I_{Lk} \in [-1, 1]$ is the intensity parameter for $k$-th light, $\mathbf{x}(p) \in \mathcal{R}^3$ is the surface position, and $\mathbf{x}_{Lk} \in \mathcal{R}^3$ the light position. The distribution parameter $R_k \in \mathcal{R}$ restricts the each lighting area. The light contribution $g(d, R_k)$ is computed as follows:

$$g(d, R_k) = \begin{cases} 1 - d/R_k & 0 \le d \le R_k \\ 0 & R_k < d \end{cases}. \tag{5}$$

The function $g(d, R_k)$ is defined as a clamped linear interpolation between $[0, 1]$ influenced by the user-given lighting area $R_k$.

**Figure 9** demonstrates the multiple lighting behavior in Eq. (4). In Fig. 9, two additional lighting effects are used to modify the shading result obtained by global appearance. We visualized the distribution parameter $R_k$


**Fig. 10** Painting operations

as 3D sphere with the center $\mathbf{x}_{Lk}$ for two additional lighting effects. These guidance sphere locations are directly controlled in 3D space to update the light positions. The intensity parameter $I_{Lk}$ will adjust the overall strength of the lighting effect in the area of the guidance sphere. Negative values for $I_{Lk}$ are also possible for making the effect darker. All these parameter editing processes will affect the weight function in Eq. (4). These parameters can be freely adjusted until the desired shading result is achieved.

#### 3.2.2 Painting operation

For the painting operation, we defined the second weight function $w_2(p) = w_p(p)$ with the following simple form:

$$w_p(p) = I_p C_p(p), \tag{6}$$

where $I_p \in [0, 1]$ uniformly controls the power of the effect and $C_p(p) \in [-1, 1]$ is the painted brightness value at the surface point $p$.

As shown in **Fig. 10**, the painted brightness modifies more detailed lighting areas compared to the multiple lighting behavior. In Fig. 10, painted brightness $C_p$ (in the middle) affects the detailed shade in the particular area. The direct 3d painting process is provided for the editing process. Artist can set offset values which affected the painted map which internally updated to change the final shading result. Since this local lighting usage can be darkened, we set the painted brightness range with negative values.

## 4. Results and Discussion

### 4.1 Implementation

Our shading pipeline is implemented in Autodesk Maya ® as plug-in using CgFx shader (GPU shading).

Interactive frame rates are either possible for many lights since the method operates a simple modification of texture coordinates in texture projection. The computer specification that used to be the experiment environment was Intel ®Core ™ i7 3.4 GHz, 32GB Memory GeForce GTX 770. We can provide real-time preview at 70 FPS (average) for the animation example (with 9201 vertices and two additional lighting effects).

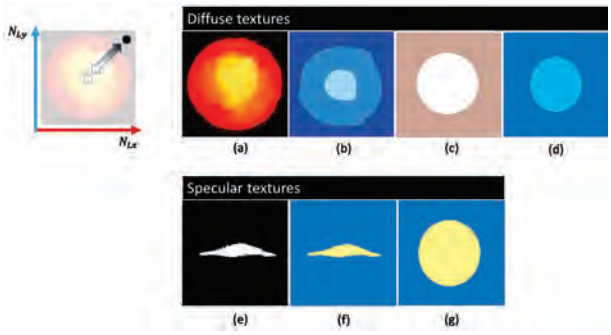



**Fig. 11** Requirement for 2D texture reference and its variations

With the implemented shading pipeline, our system provides a typical lighting design workflow summarized as follows.

First, the user designs the key lighting for the prepared input 3D scene and camera path. These settings will affect the global appearance whose shading attributes are the light vector, the normal vector, and the view vector in section 3.1

Next, the user modifies the global appearance through multiple lighting behavior in section 3.2.1 and painting operations in section 3.2.2. In our experiments, we use additional lighting effects for multiple lighting behavior. We also provide additional controls for specular by adding the control parameters for them. Painting operations are also extended to specular for more detailed controls.

In the following, we will discuss the characteristics of our method.

### 4.2 Requirement for 2D texture reference

In our shading pipeline through texture projection, we assume a circular pattern in the color distribution of a 2D texture reference. As shown in **Fig. 11**, our $z$ coordinate transformation will change the texture lookup coordinates along radial directions. Figures 11 (a)–(d) portray the impression of the center means the brightest and the outside becomes darker.

With the requirement as a shading effect, the artist can design 2D color distributions for the global shading appearance. The specular lighting texture is also designed in the same manner as the diffuse lighting texture. To composite them, the specular lighting texture is clipped with alpha regions.

### 4.3 Multiple lighting behaviour

Since the global appearance is designed by only one key lighting, the artists want to lit some dark areas resulted in the key lighting. In the experiments on Fig. 9 and **Fig. 12**, we design multiple lighting effects for static


**Fig. 12** Multiple Lighting Result for difuse lighting with Fig. 11 (b) as 2D texture reference

models and animated lighting. Figure 11, shows the additional 2 lights incorporate with the 1 key light. The first row shows the original shading with one key lighting. In the second row, the right thigh is lit by the additional light. In the third row, the left thigh is lit also by the additional light.

The process of creating Fig. 9 start with inputting the texture reference as shown in Fig. 11 (a) for the global appearance as a single diffuse lighting which lit from the upper part of the character, then we put 2 additional lightings in chest and upper arm. In making Fig. 12, the texture reference in Fig. 11 (b) is used for the diffuse lighting effect, which moves from left to right in the upper part of the character. We then design 2 additional lightings in the right thigh and left thigh. Same process for creating Fig. 12, Fig. 11 (b) as an animated single difuse lighting

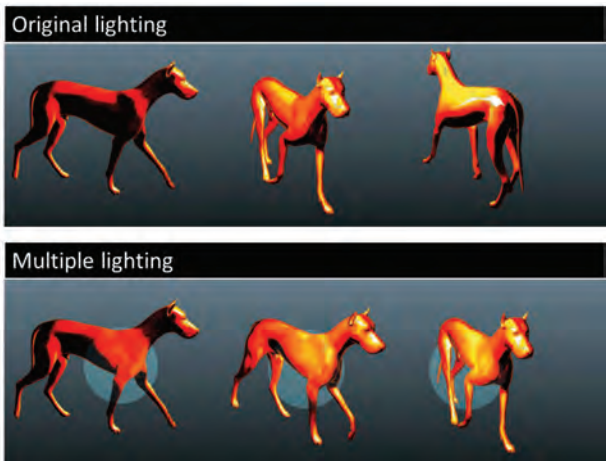



**Fig. 13** Multiple lighting case in animated scene

which moving from left to right in the upper part of character, after that we add 2 additional lightings in the right thigh and left thigh.

Figure 9 and Fig. 12 demonstrate the use case of the multiple lighting result to emphasizing muscle as the features of the character, thus the additional lighting can lit the other parts of character that lead to the stronger impression. Our function can be acted as a local multiple lighting for the character.

This kind of style is difficult to achieve using simple multiple texture projection computation since those method can occurs blending problem. Also the additional light can also move to the other parts of the surface so that give interactive interaction to the artist.

The multiple texture mapping blending problem as shown in Fig. 2 occurs because multiple 2D textures are simply composed together. This limitation can be solved using our proposed method to produce the multiple lighting by transforming the texture sampling for getting new color within radial direction with designated intensity parameter and distribution parameter.

**Figure 13** shows how our method is applicable for the dynamic scene including the character animation. In Fig. 13, the first row use only one key lighting and the second row use one additional lighting for lit the leg. Artist set Fig. 11 (a) as single static lighting for global lighting as diffuse lighting then add additional one lighting in the leg in order to emphasize the leg area. The additional lighting can also be seamlessly blended over the key lighting and make it easier for the animator to create seamlessly multiple lighting effects in the animation. The lighting interface is created to be stick on the surface so that easier for controlling in animated object rather than hovering like usual 3D lighting editing operation.

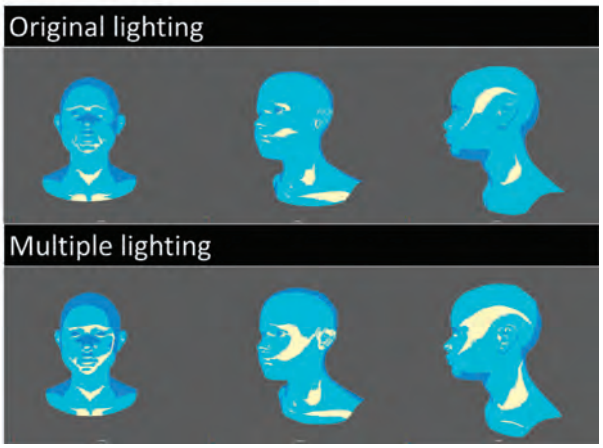



**Fig. 14** Multiple lighting result for specular lighting with Figs. 11 (d) and (f) as texture reference

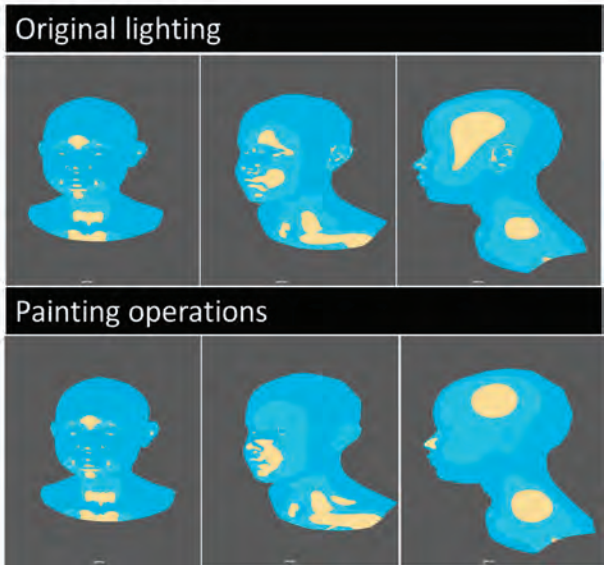



**Fig. 15** Painting operation results for specular lighting with Fig. 11 (d) and Fig. 11 (g) as texture reference and the scene of animated light and animated object

To full fill the artist intention for achieving multiple lighting looks certain looks in particular frame, the additional lighting can be incorporated with key framing technique with animating the light position with common interface and tweak the color with slider interface using intensity parameter for each light also distribution parameter.

Multiple lighting also can be applied into the specular lighting in the animated lighting and object as shown in **Fig. 14** by inputting Figs. 11 (d) and (f) animated diffuse lighting and specular lighting which move from bottom to up, then the artist add 1 additional light to broaden the specular lighting. In Fig. 14, the first row shows the original shading with one specular lighting. In the second row, one additonal lighting is added to make the specular ligthting bigger. The result shows that the method can expand the specular lighting for changing the character mood into stronger impression while maintaining artistic result.

### 4.4 Painting operation

Sometimes artists want to make imaginative shading to change the mood of the character so that familiar paint interface is also provided to modify the shading in arbitrary areas for making various shapes. These modifications act as local lighting effects, which will be seamlessly integrated with the texture projection result in global appearance.

Compared to the multiple lighting behavior, the painted shading effects can adjust more local lighting in details. By using the same scene as used in Fig. 14, we demonstrate the usability of painting operations in **Fig. 15** and **Fig. 16** as shown in Fig. 10, our paint interface enables the artist to design the unrealistic shape around the nose area to create more mysterious feeling while result in Fig. 15 (second row - third column) shows the capability of the operation to create particular shape for giving accent. In Fig. 15, negative value is inserted in the painting operation to erase unwanted shading and getting certain shape.

Fig. 16 and Fig. 15 (second row - second column) show that our system also applicable for editing the unwanted shading in animated scene toward diffuse lighting and specular lighting respectively. In Fig. 16, the first row shows the shade before the painting operation and the second row shows shading after painting operation. Artist set the negative value in the painting operation so that darker area can be achieved to get certain shape.

Such modifications are difficult to achieve by the multiple lighting behavior where the lighting effects are affected by the model's geometry. Our system provides different levels of lighting controls depending on the artistic requirement.

While our system provides a controllable single brightness map for the whole animation, we can observe that the painted lighting effects are useful for cut shots. Depending on the demands of artists, we require key-framing scheme like Todo et al. 2007[1) ].

### 4.5 User study

The questionnaire was conducted in order to get the viewer's impression to our result which seamlessly blends multiple lighting effects. 5-point Likert rating scales are used to gather the subjective opinions of the participant

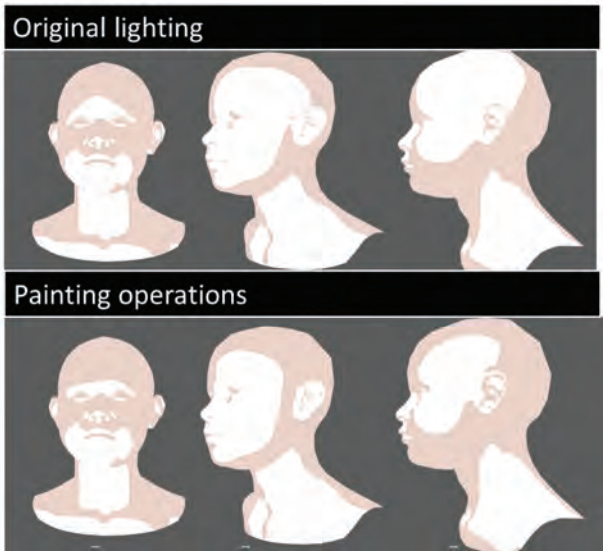



**Fig. 16** Painting interface result for difuse lighting using Fig. 11 (c) as 2D texture reference and the scene contains animated light and animated object

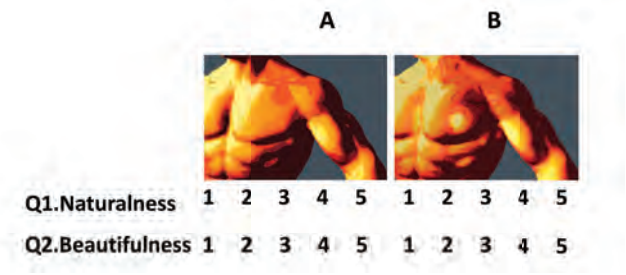



**Fig. 17** Questionnaire for Fig. 4

towards the resulted shading in Fig. 4 and relation to each question.

The questionnaire as shown in **Fig. 17** consists of Fig. 4 picture and two questions with 5 point Likert rating scales in order to collect subjective evaluation from each question. The participant chooses the scale according to them though about the relation between the picture and each question with 1 is the lowest mark and 5 is the highest mark. We focus on the aesthetic perspective such naturalness and beautifulness. Thus, the questions are defined as follows:

- Which one is natural as shading effect?
- Which one is beautiful?

We conducted the experiment with 12 participants, mainly form the university students with various art skills. Each participant was introduced to the problem briefly then give the freedom to decide their choices. From that, we collected the values and analyzed the result.

Our result can receive good impression about the aesthetic perspective. In **Table 1**, we can see the actual

**Table 1** Quistionnaire result

| | A | | | | | B | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| Naturalness | | | 2 | 8 | 2 | | 8 | 2 | 1 | 1 |
| Beautifulness | | | 3 | 6 | 3 | | 3 | 3 | 3 | 3 |


**Fig. 18** Artist's results

result shows that many participants chose our result for the questions about the naturalness and beautifulness.

We also tried to conduct a user study with an artist to consider more practical scenarios and get opinion about the usability of the proposed method. The artist is from the undergraduate student who has the best skill among the others. As for the introduction to the method, we briefly explained to the artist about the use of the shading also the user interface to operate.

Roughly the experiment done in 30 minutes. Firstly the artist created one 2D texture for making the global appearance. After applying the 2D texture, artist put 3 additional lighting prior to one global lighting, which mainly placed on the head specifically on the mouth, eye, and ear. These additional lightings are intended to lit the head that cannot be lit just from 1 global lighting. Thus, these additional lightings can enhance the shape of the head. In addition, the artist made more specific tweak using painting interface for sparsely erase shading on the character. This produces a stronger emotion impression toward the viewers. The result of this experiment can be seen on **Fig. 18**. We asked the artist to evaluate the usability of our system. The impression was positive: the operation is enough intuitive to perform editing as she expected. The slider interface in the method makes artist easier to change the look of each additional lighting also in the painting interface that uses for choosing the color.

## 5. Conclusion

In this paper, we have discussed region control for a 3D lighting scene with hand-drawn appearance in NPR scenarios. We propose a simple $z$ coordinate transformation in dynamic texture projection to provide the user with lighting modifications such as multiple light effects

and painting operation. Our implementation fits the existing pipeline with the familiar user interface and run at interactive rates for real-time previewing.

## 6. Limitation and Future Works

While we have shown how our method is applicable for several practical scenarios to satisfy the artistic requirements, there is still room for improvement.

Since we have focused on lighting behavior for the texture projection, our transformation only handles radial directions in texture mapping process as shown in Fig. 11. We need to discuss a more general transform along arbitrary directions with corresponding practical scenarios. For example, several affine transformations are discussed in previous Lit-sphere extension[4].

Another limitation is that we assume only single 2D texture reference for our lighting controls. As the future works, we also artist want to explore the multiple texture projection with various styles from a set of 2D texture references so that these shading effects can be seamlessly blended and animated.

The current painting framework is limited to a single brightness map for the whole animated scene. We may require the ability to modify the painted brightness map dynamically including dragging operations or key-framing schemes. In addition, our scheme restrict the artist to combine the different levels of lighting controls. Inverse decomposition schemes may be desirable to directly animate the painted shading from scratch.

In all future directions, we consider that providing intuitive shading design user interface is essential for animating hand-drawn shading in NPR.

## References


1) H. Todo, K. A'njyo, W. Baxter, T. Igarashi: "Locally Controllable Stylized Shading", ACM Trans. on Graphics, Vol.26, No.3, Article No.17 (2007).
2) P. Wisessing, J. Dingliana, R. McDonnell: "Perception of Lighting and Shading for Animated Virtual Characters", Proc. of Applied Perception 2016, pp.25–29 (2016).
3) P.-P. J. Sloan, W. Martin, A. Gooch, B. Gooch: "The Lit Sphere: A Model for Capturing NPR Shading from Art", Proc. of Graphics Interface 2001, pp.143–150 (2001).
4) H. Todo, K. Anjyo, S. Yokoyama: "Lit-Sphere Extension for Artistic Rendering", The Visual Computer, Vol.29, Issue 6-8, pp.473–480 (2013).
5) R. Pacanowski, X. Granier, C. Schlick, P. Poulin: "Sketch and Paint-based Interface for Highlight Modeling", Proc. of Sketch-Based Interfaces and Modelling (SBIM) 2008, pp.17–23 (2008).
6) P. Bénard, F. Cole, M. Kass, I. Mordatch, J. Hegarty, M. S. Senn, K. Fleischer, D. Pesare, K. Breeden: "Stylizing Animation by Example", ACM Trans. on Graphics, Vol.32, No.4, pp.119:1–119:12 (2013).
7) A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, D. H. Salesin: "Image Analogies", Proc. of SIGGRAPH 2001, pp.327–340 (2001).
8) J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, D. Sýkora: "StyLit: Illumination-guided Example-based Stylization of 3D Renderings", ACM Trans. on Graphics, Vol.35, No.4, pp.92:1–92:11 (2016).
9) L. A. Gatys, A. S. Ecker, M. Bethge: "Image Style Transfer Using Convolutional Neural Networks", Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) 2016, pp.2414–2423 (2016).
10) J. Mitchell, M. Francke, D. Eng: "Illustrative Rendering in Team Fortress 2", Proc. of Non-Photorealistic Animation and Rendering (NPAR) 2007, pp.71–76 (2007).
11) D. Vanderhaeghe, R. Vergne, P. Barla, W. Baxter: "Dynamic Stylized Shading Primitives", Proc. of Non-Photorealistic Animation and Rendering (NPAR) 2011, pp.99–104 (2011).
12) P. Barla, J. Thollot, L. Markosian: "X-toon: An Extended Toon Shader", Proc. of Non-Photorealistic Animation and Rendering (NPAR) 2006, pp.127–132 (2006).
13) C.-R. Yen, M.-T. Chi, T.-Y. Lee, W.-C. Lin: "Stylized Rendering Using Samples of a Painted Image", IEEE Trans. on Visualization and Computer Graphics, Vol.14, No.2, pp.468–480 (2008).
14) C. D. Kulla, J. D. Tucek, R. J. Bailey, C. M. Grimm: "Using Texture Synthesis for Non-Photorealistic Shading from Paint Samples", Proc. of Computer Graphics and Applications 2003, pp.477–481 (2003).
15) J. Obert, J. Křivánek, F. Pellacini, D. Sykora, S. Pattanaik: "iCheat: A Representation for Artistic Control of Indirect Cinematic Lighting", Proc. of Eurographics Symposium on Rendering (EGSR) 2008, pp.1217–1223 (2008).
16) Y. Kim, J. Noh: "LightShop: An Interactive Lighting System Incorporating the 2D Image Editing Paradigm", Proc. of International Symposium on Visual Computing (ISVC) 2009, pp.59–70 (2009).
17) W.-C. Lin, T.-S. Huang, T.-C. Ho, Y.-T. Chen, J.-H. Chuang: "Interactive Lighting Design with Hierarchical Light Representation", Proc. of Eurographics Symposium on Rendering (EGSR) 2013, pp.133–142 (2013).
18) M. T. Bui, J. Kim, Y. Lee: "3D-look Shading from Contours and Hatching Strokes", Computers & Graphics, Vol.51, No.C, pp.167–176 (2015).
19) C. J. Zubiaga, A. Muñoz, L. Belcour, C. Bosch, P. Barla: "MatCap Decomposition for Dynamic Appearance Manipulation", Proc. of Eurographics Symposium on Rendering (EGSR) 2015 (2015).
20) K. Bassett, I. Baran, J. Schmid, M. Gross, R. W. Sumner: "Authoring and Animating Painterly Characters", ACM Trans. on Graphics, Vol.32, No.5, pp.156:1–156:12 (2013).
21) W. B. Kerr, F. Pellacini: "Toward Evaluating Lighting Design Interface Paradigms for Novice Users", ACM Trans. on Graphics, Vol.28, No.3, pp.26:1–26:9 (2009).


(Received November 29, 2018)

**Muhammad ARIEF**
He received the B.S. from Institut Teknologi Sepuluh Nopember and M.S. from Tokyo University of Technology. His research interests include non-photorealistic rendering, and game design.

**Hideki TODO** (*Member*)
He is an assistant professor at Chuo Gakuin University. His work experience includes working as a researcher in animation industry at OLM Digital, Inc. from 2011 to 2013. He received his Ph.D. degree of Information Science and Technology from the University of Tokyo in 2013. His research interests lie in computer graphics, image processing, and computer vision, including stylized rendering, image stylization, inverse rendering, and image or video editing.

**Koji MIKAMI**
He worked at Nissho Iwai Corporation and MK Company as a producer. In 1998, he established "Creative Lab." at the Katayanagi Advanced Research Institute of Tokyo University of Technology (TUT), where research on Animation, Game and CG production technology are conducted. Currently, he is a professor at the School of Media Science, TUT. He received his Ph.D. degree at KEIO University in 2018. Now his research interests lie in game design and production technology for game and animation.

**Kunio KONDO** (*Honorary Member*)
He is a professor at the School of Media Science, Tokyo University of Technology. He received his Ph.D degree from the University of Tokyo in 1988. He was associate professor of Department of Information and Computer Sciences, Saitama University, and Technical staff of Nagoya University. He is the president of Asia Digital Art and Design Association. He was former President of TheInstitute of Image Electronics engineers of Japan, former President of The Society for Art and Science, and Chair of SIG on Computer Graphics and CAD of Information Processing Society of Japan.

IIEEJ **Short Paper**

# Region Mining of Fetal Head in Ultrasound Image Based on Weakly Supervised Annotations and Deep Learning

Yan LI[†](*Student Member*), Rong XU[††], Artus KROHN-GRIMBERGHE[†††], Jun OHYA[†,††](*Member*), Hiroyasu IWATA[†]

† Department of Modern Mechanical Engineering, Waseda University,
†† Global Information and Telecommunication Institute, Waseda University,
††† Department of Business Information Systems, Paderborn University

<**Summary**> To locate the fetal head in ultrasound (US) images, this paper proposes a deep learning based method for weakly supervised learning from image-level annotation. We first modify and train fetal head classification models based on existing backbone structures, then adopt the feature maps and learned weights to visualize the high response areas of fetal head. In order to improve the localization accuracy, this paper further optimizes completeness of the salient area of the fetal head by adopting multiple feature maps from different feature levels. The final bounding box of the fetal head is obtained from mined regions through threshold. We evaluate both fetal head plane classification and weakly learned localization results in US images. In the experiments we compare several backbone structures and verify the effectiveness of the proposed method.

**Keywords**: weakly supervised learning, fetal head localization, ultrasound, convolutional neural network

## 1. Introduction

Ultrasound (US) imaging is widely used in fetal prenatal diagnosis due to its low cost and low harm to the human body. Recently, automatic fetal care systems are desired by medical areas. Such a system requires a technology that locates the fetal head so as to infer the gesture and position of the fetus; then, the system can perform subsequent processes such as guiding the US probe to the desired positions for further measurement.

Classification of fetal head plane from US image has been studied for a long time, e.g. by L Zhang et al.[1] and Hao Chen et al[2]. However, in many cases of automatic medical treatments, classification of the category of US plane is not enough. We need to locate the position of the targets in the image in order to provide more reliable references. Normally, object detection model can be obtained by fully supervised learning, such as in the work proposed by Albayrak et al[3]. However, these methods rely on accurate manual annotation of the target position, such as pixel-wise classification mask or bounding boxes, which cost a large amount of human resources.

The hieratical feature can be used to visualize the discriminative region by the method proposed by Zhou et al[4]. The convolution neural network maps the input data into hieratical feature spaces, then the feature maps can be adopted as salient image of the target object through linear combination. In case of medical image processing, recently, some of the publications directly use the method to visualize the interesting area of the illness, for instance, Xiaosong Wang et al's work[5]. They mainly target at the classification tasks. The extracted response of the last convolution layers have small size and high feature level, which cause the response map only reflects on the regions which have the most discriminative information. Yunchao Wei et al.[6] propose to solve the problem by iteratively masking the pixels which have high confidence then re-classify the image from the rest of the pixels. However the method cannot provide a reliable termination method, the mining region has risk to be over-segmented. Therefore, the issue of the completeness of the weakly supervised region mining is still need to be further solved.

The targets of the paper are illustrated in **Fig.1**. Given US images, through learning from image level annotations, we aim at mining the region of the fetal head from the learned models. The region of the fetal head is represented as a tight bounding box of the fetal head area. We first propose a fetal head classification framework based on exist backbone networks, then use discriminative maps

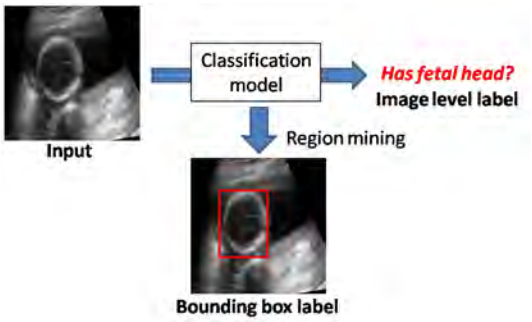



**Fig. 1** Localization of fetal head by learning from image level annotations

of fetal head which are merged from multi-scale feature maps to achieve the target.

## 2. Proposed Method

### 2.1 Fetal head plane classification from US images

The modified CNN models are used to train classifier from US slice and image level fetal head annotations, which are manually specified if each image includes a fetal head. To classify the input US image into different categories, the CNN model uses hieratical designed backbone network to extract the feature. In this paper we treat off-the-shelf networks as the feature extractors and their base structures remain unchanged.

In order to visualize the discriminative area from learned classification models, the modifications come from the output of the network structure. In order to map the output into one dimensional vector, which indicates whether a fetal head exists, the normal CNN classification model first needs to reshape the responses of the convolution layers into a vector and then connects the feature vectors to inner production layers. However, the inner production changes the geometric feature, because all of the pixels are mapped into one through linear combination. We need to use discriminative feature maps to localize the fetal head area. Therefore, we replace the reshape and inner product operations of the selected backbone structures by a single global average pooling layer.

During the training phase, we use typical cross entropy loss to update the weights of our binary classification model. In the sample US video sequence, fetal heads exist only in a small number of the sequences. This leads to the situation in which the data distribution has severe imbalances. Therefore, we modify the loss function with weights of negative category. The weight $\omega$ is calculated by $\omega = P/N$, where $P$ and $N$ are number of positive and negative samples. For each batch with $n$ samples, the loss function is written by

$$L = -\frac{1}{n}\sum_{n=1}^{n}[y_n \log p_n + \omega(1-y_n)\log(1-p_n)], \quad (1)$$

where $y_n$, $p_n$ indicate binary values of the label and prediction of the model, respectively.

### 2.2 Localization of fetal head by multi-scale discriminative maps

We adopt the feature maps' responses of the last convolution layer and the weights of the output layer to obtain the discriminative area from the learned model. In particular, the input blob of the output layer is calculated from global averaging over the feature maps $f$ of the last convolution layer. Then the compressed feature is directly mapped into the shape of the output vector by linear combination with the weights corresponding to the two categories. The weights of the output layer $w$ and feature maps $f$ are adopted for visualizing the discriminative maps of the desired categories.

The highest feature level in the deep learning model is learned from the most discriminative areas of the input feature maps. However, the most discriminative area cannot be seen as the complete area of the target object. The mined region lacks completeness due to presence of some of not so important areas. This makes finally located fetal head area is not good enough.

The pooling layers are used multiple times to narrow down the original image through averaging or maximum operations. In our research, the response maps for the discriminative localization are extracted from the last output of the hieratical layers. As the feature level becomes deeper; the semantic features get higher. The less important feature areas are progressively ignored by the network. We think the ignorance could degrade the completeness of target shape. To deal with the issue, we propose to adopt the outputs from the multi-level features. The response maps from different feature levels represent different discriminative locations.

The proposed method replaces the output with multiple output branches which split from different level of backbone network's outputs and merge the multiple discriminative maps into one, as shown in **Fig.2**. In particular, for each output branch, we add specified number of convolution layers and global average pooling operations. During the training, the output of each branch is compared with the ground truth labels, and the weights of each output branch are independently updated. By merging multiple outputs, our models learn discrimina-

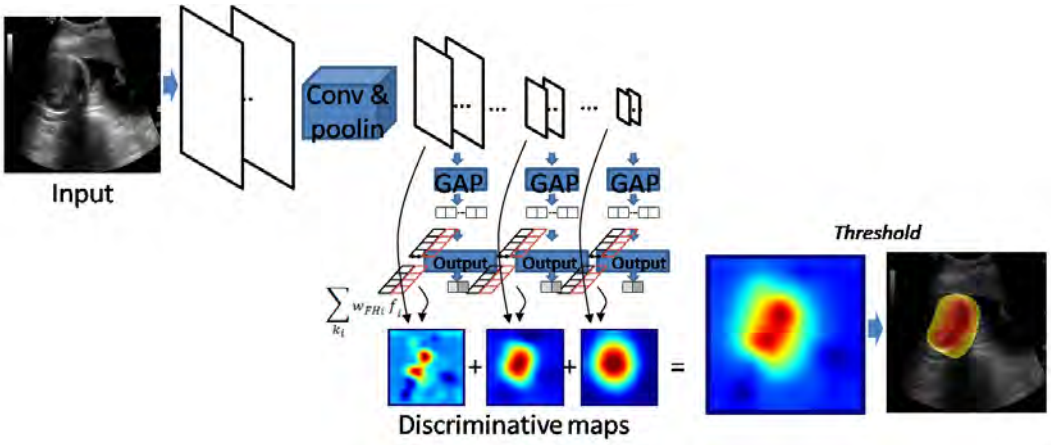



**Fig. 2** Merging of multi-scale discriminative maps

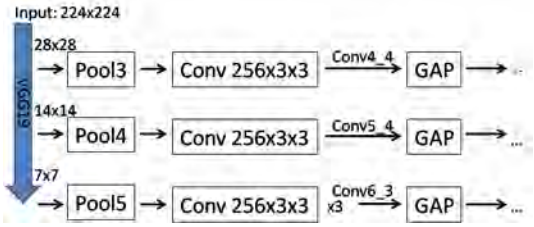



**Fig. 3** Detailed structure of output branches added on VGG19

tive response maps from layers with different feature levels. The branch structure with detailed parameters on VGG19 structure can be found in **Fig.3**. During the localization, we extract the response maps of the last convolution layers from each output branch and use the weights of each branch separately to linear combine them as multiple discriminative maps. At last we use a sum calculation of each pixel to obtain the final discriminative image $M$ of fetal head. The optimized approach can be presented as

$$M = \sum_i \frac{1}{k_i} \sum_{k_i} w_{Fi} f_i, \tag{2}$$

where $w_{Fi}$ and $f_i$ indicate the weights that correspond to the fetal head and feature maps of the $i$th output branch, respectively; $k_i$ is the number of kernels of the ith output branch.

## 3. Experiment

### 3.1 Dataset and training details

#### 3.1.1 Data collection

We conduct preliminary experiments on clinical US dataset to verify the proposed method. The research acquires four clips of US pregnant examinations from a hospital as the metadata. For each US video clip, anonymous patient with different fetal weeks that arranged from 19 to 23 are used for the experiments. The 2-folder cross validation is adopted by alternatively training and testing on randomly selected groups of US clips (each group has two clips) with different subjects. We select the clips that have fetal body perpendicular to the US scan plane. After pruning and interval sampling operations, the US video clips are stored in sequential image format. The frames that with observe-able fetal head are manually identified from the image sequences. Then, we assign pixel-wise labels to each of the images containing fetal heads. Note that the pixel-wise annotations are only used for evaluation purpose. The bounding boxes of the fetal head are calculated from thus obtained segmentation masks.

#### 3.1.2 Data augmentation

Our dataset has severe data imbalance and lacks of diversity issues. Therefore, we use related large scaled data augmentation operations on the training set. We add random crop, rotation, scale transform, and horizontal flip to the raw US image to obtain disturbance to each of the training samples. The augmented samples are resized to the fixed size in order to fit the input of the backbone network.

#### 3.1.3 Domain transferred learning

Compared with learning from random initialed weights, better initializations can be obtained from pre-trained low-level representations that learned from large scale cross domain images. Most of weights of the convolution layers are updated from imagenet[7] pre-trained models while the additional layers with learn-able weights are learned from scratch.

#### 3.1.4 Threshold

The discriminative map is normalized by rescaling the values from 0 to 1. In our experiments we use a fixed value (0.8) to obtain binary masks. To determine the threshold

value, we quantitative evaluate the IoU of region mining results on the training sets and roughly select the values which have the best performance on each sub folder. The bounding box of the fetal head is obtained from connected domain. We only keep the bounding box which has the largest area if multiple areas are obtained.

### 3.2 Results and discussions

#### 3.2.1 Metrics

First, to evaluate the model for fetal head plane classification, we adopt precision, recall and f1 score. They can be defined as

$$\mathrm{f1} = 2 \times \frac{Recall \times Precision}{(Recall + Precision)}, \qquad (3)$$

where $Recall = N_{TP}/(N_{TP} + N_{FN})$, $Precision = N_{TP}/(N_{TP} + N_{FP})$, the $N_{TP}$, $N_{FN}$ and $N_{FP}$ are numbers of true positive, false negative and false positive predictions, respectively. We also provide the Area Under Curve (AUC) of Receiver Operating Characteristics (ROC). Regarding the localization accuracy, we obtain the bounding box from manually annotated segmentation masks of the fetal head. We adopt Intersection Of Union (IOU) to evaluate the results. The IOU is defined as

$$\mathrm{IOU} = \frac{(S \cap S_{GT})}{(S \cup S_{GT})}, \qquad (4)$$

where $S$ and $S_{GT}$ are obtained area of the predictions and the ground truth. Besides IOU of bounding box (Bbox), we also provide the pixel-wise IOU (Pwise). The calculation of pixel-wise IOU is similar as bounding box IOU, except we judge the classification accuracy at each of the pixels of the input image.

#### 3.2.2 Results

Regarding the classification model, we compare several popular backbone networks, which are alexnet[8], resnet50[9] and VGG19[10]. For both Alexnet and VGG19 structures we add batch normalization[11] after each convolution layer. As we mention in section 2.1, we replace all of the fully connecting layers with the convolution layer and global average pooling operations.

The comparison results of the three models for the fetal head plane classification task can be found in **Table 1**. From the result we can see that, among the selected three models, the VGG19 with batch normalize structure achieves the best performance. Although the alexnetGAP has simple structure and the smallest network scale, it still can get acceptable classification rate because of the batch normalization and the pre-trained weights. In the following experiments we build our optimizations based on VGG19 structure.

**Table 1** Classification results of different backbone networks (%)

| Network | AlexnetGAP | Resnet50GAP | VGG19GAP |
|---|---|---|---|
| Recall | 87.79 | 86.92 | **90.92** |
| Precision | 88.65 | 90.73 | **90.26** |
| F1 score | 87.62 | 88.39 | **90.35** |
| AUC | 95.87 | 96.34 | **96.80** |

**Table 2** Localization results (IOU) with different backbone networks in VGG19GAP_OutputMerge (%)

| Network | Alexnet GAP | Resnet50 GAP | VGG19 GAP | VGG19GAP _Output Merge |
|---|---|---|---|---|
| Bbox | 41.4(37.0) | 40.3(36.1) | 61.2(54.8) | **65.3(58.0)** |
| Pwise | 60.7(58.8) | 57.7(55.0) | 72.0(70.8) | **76.5(73.1)** |

**Table 3** Localization results (IOU) with different output layers (%)

| Output layer | Conv4_4 | Conv5_4 | Conv6_3 | Output Merge |
|---|---|---|---|---|
| Pwise _FetalHead | 19.83 | 51.51 | 54.95 | **57.94** |
| Pwise _Bkg | 92.80 | **95.40** | 94.83 | 95.13 |
| Bbox | 53.20 | 62.24 | 55.71 | **65.25** |
| Pwise | 56.32 | 73.46 | 74.89 | **76.53** |

The comparison of the IOU of the bounding box of the three backbone structures can be seen in **Table 2**. Here, values not in bracket show w/o false negative classifications, and values in bracket show w/ false negative classifications in Table 2. In Table 2, to decouple the localization and classification performance, the scores (not in brackets) do not count false negative classification samples, while the scores in brackets are the IoU results with false negative classifications. The results of resnet50GAP are worse than alexnetGAP probably because it has too small feature map size. Same as the classification results, the VGG19 achieves the best localization results. By adding our proposed optimization method (VGG19GAP_OutputMerge), the localization performance is further improved. We compared the result of only using the last feature map output and the two optimization methods in the **Table 3**. The name of each result indicates the output of different layers in VGG19GAP. In Table 3, pixel-wise accuracy of fetal head ("_FetalHead") and other areas ("_Bkg") are demonstrated separately. Regarding the proposed multiple discriminative outputs, we adopt the conv4_3, conv5_3 and conv6_3 from VGG19 structure (the detail is shown in Fig.3). The result obtained by merging the above-

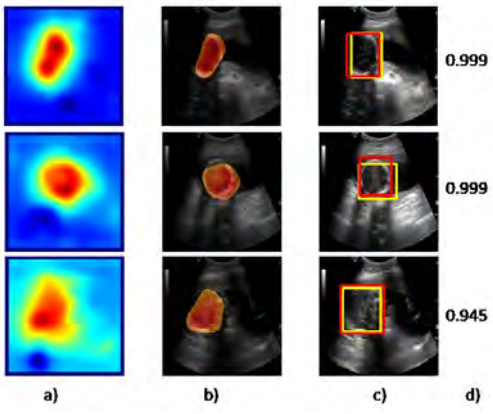



**Fig. 4** Visualized results obtained by VGG19GAP_OutputMerge; a) merged discriminative map, b) predicted fetal head region on input image, c) obtained bounding box (yellow) and ground truth (red) of fetal head region, and d) classification probability of positive category for each sample

**Table 4** Localization results (IOU) of weakly and fully supervised methods (%)

| Method | Weakly supervised | Fully supervised | Bbox (IOU) |
|---|---|---|---|
| VGG19GAP | √ | × | 61.2 |
| VGG19GAP _OutputMerge | √ | × | **65.3** |
| F-RCNN[12)] | × | √ | 72.2 |

mentioned three outputs is indicated by "OutputMerge". From the quantitative results we can see that the merged feature maps bring significant improvements by enriching the integrity of the fetal head area. Some of the visualized results are shown in **Fig.4**.

In addition, to evaluate the accuracy difference with the fully supervised approach, we further demonstrate the results which are obtained from Faster RCNN[12)](F-RCNN). Note that, the model of F-RCNN is learned from bounding box annotations. Compared with weakly supervised methods, it is impossible to learn from image level annotations. To train this model, we adopt the manually labeled bounding box annotations, which are originally used for evaluation. In particular, to the structure of F-RCNN model we embed the original VGG16 as the backbone structure and use $600 \times 600$ pixels as the input resolution for both training and testing. The comparison results are shown in **Table 4**. From the results we can see that, comparing with the baseline weakly supervised method (first row), the method with our proposed optimizations (second row) shorten the accuracy gaps between the models which learn from image level annotations and from object position annotations. On the other hand, the weakly supervised method has strong advantages over learning only from image level annotations, in that the weakly supervised method can avoid complex and time-consuming medical image annotation tasks.

## 4. Conclusion

This paper has proposed a method for training fetal head plane classification models for US pregnant images. The weakly supervised method is adopted to obtain the fetal head area. This paper discusses the insufficient of the current method, and tries to inspire the future research to merge multi-scaled discriminative maps with different feature levels to get more complete salient areas. The effectiveness of proposed method is verified through preliminary experiments. More experiments on larger scaled clinical US dataset need to be conducted for future works.

## References

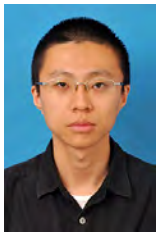
1) L. Zhang, S. Chen, C. T. Chin, T. Wang, S. Li: "Intelligent Scanning: Automated Standard Plane Selection and Biometric Measurement of Early Gestational Sac in Routine Ultrasound Examination", Medical Physics, Vol. 39, No. 8, pp. 5015–5027 (2012).
2) H. Chen, D. Ni, J. Qin, S. Li, X. Yang, T. Wang, P. Heng: "Standard Plane Localization in Fetal Ultrasound via Domain Transferred Deep Neural Networks", IEEE Journal of Biomedical Health Informatics, Vol. 19, No. 5, pp. 1627–1636 (2015).
3) B. A. Nur, B. O. Ayse, S. A. Yusuf: "Prostate Detection from Abdominal Ultrasound Images: A Part Based Approach", Proc. of IEEE Conference on International Conference on Image Processing, pp. 1955–1959 (2015).
4) B. -L. Zhou, B. , A. Khosla, A. Lapedriza, A. Oliva, A. Torralba: "Learning Deep Features for Discriminative Localization", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016).
5) X. -S. Wang, Y. -F. Peng, L. Lu, Z. -Y. Lu, M. Bagheri, R. M. Summers: "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3462–3471 (2017).
6) Y. -C. Wei,J. -S. Feng, X. -D. Liang, M. -M. Cheng, Y. Zhao:"Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 6488–6496 (2017).
7) J. Deng, W. Dong, R. Socher, L. -J. Li,K. Li, F. -F. Li:"ImageNet: A Large-scale Hierarchical Image Database", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009).
8) A. Krizhevsky I. Sutskever, G. Hinton: "ImageNet Classification with Deep ConvolutionalNeural Networks", Advances in Neural Information Processing Systems 25, pp. 1097–1105 (2012).
9) K. -M. He, X. -Y. Zhang, S.-Q. Ren, J. Sun: "Deep Residual Learning for Image Recognition", Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016).
10) K. Simonyan, A. Zisserman: "Very Deep Convolutional Net-

works for Large-Scale Image Recognition", arXiv:1409.1556 (2014).

11) S. Loffe and C. Szegedy: "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Proc. of the 32nd International Conference on Machine Learning, pp. 448–456 (2015).

12) S. -Q. Ren, K. -M, He, R. Girshick, J. Sun: "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks", Proc. of International Conference on Neural Information Processing Systems, Vol.1, pp. 91–99 (2015).

(Received June 22, 2018)
(Revised December 4, 2018)



**Yan LI** (*Student Member*)
He earned B.S. degree in Communication University of China. He obtained M.S. degree in Graduate School of Global Information and Tele-communication Studies (GITS), Waseda University. He is currently a doctoral program student in Graduate School of Creative Science and Engineering, Waseda University. His research areas include computer vision and machine learning.

**Rong XU**
He earned B.S. degree in School of Information Science and Engineering, and M.S. degree in School of Computer Science and Engineering from Southeast University, China in 2003 and 2009, respectively. He obtained his Ph.D degree in Graduate School of Global Information and Tele-communication Studies (GITS), Waseda University in 2014. Currently, he is a senior researcher in Datasection Inc., a Japanese listed company. His research focuses on image processing, video analysis, pattern recognition, deep learning, surgical image-guidance, etc.

**Artus KROHN-GRIMBERGHE**
He earned his MSc equivalent in information engineering and management from KIT, Karlsruhe Institute of Technology in 2006. After roughly a year in the industry he went back to university and obtained his Ph.D. in Machine Learning from University of Hildesheim in 2012. Afterwards he joined University of Paderborn as an assistant professor in Applied Machine Learning in 2012 for six years till 2018. Currently, he is founder and CEO of LYTIQ GmbH, a Germany-based startup focused on consulting in AI (www.lytiq.de).

**Jun OHYA** (*Member*)
He earned B.S., M.S and Ph.D degrees in precision machinery engineering from the University of Tokyo, Japan, in 1977, 1979 and 1988, respectively. He joined NTT Research Laboratories in 1979. He was a visiting research associate at the University of Maryland, USA, from 1988 to 1989. He transferred to ATR, Kyoto, Japan, in 1992. Since 2000, he has been a professor at Waseda University, Japan. He was a guest professor at the University of Karlsruhe, Germany, in 2005. His research areas include computer vision and machine learning. He is a member of IEEE, IIEEJ, IEICE, IPSJ and VRSJ.

**Hiroyasu IWATA**
He earned the B.S., M.S. and Dr. of Engineering degrees in Mechanical Engineering from Waseda University in 1997, 1999 and 2003, respectively. He is currently a Professor of Department of Modern Mechanical Engineering, School of Creative Science and Engineering, Waseda University and appointed as a Next Generation Core Researcher in Key Researchers Development Program by Board of directors, Waseda University (since 2016) as well. His research fields include human assistive and augmentation robotics, skin adhesive bio-sensors, supernumerary robotic limb "Third Arm", echography-based prenatal checkup assistive robot, perception-assistive rehabilitation RT and artificial intelligence for construction machinery. He is a member of RSJ, JSME, SICE, EAJ and IEEE.

## *Call for Papers*

# Special Issue on CG & Image Processing Technologies for Generation and Post-Processing of Image/Animation

IIEEJ Editorial Committee

In computer graphics (CG), not only realistic images but also animations can be generated in short time and with low cost, thanks to the improvement of hardware performance in recent years. In addition, creation/post-processing of animations can easily be achieved using image/video editing software and game engine under integrated environments for its development. Furthermore, production companies equipped with studios for motion capture have been increasing, so natural motion of human and animal can easily be reflected to 3D characters.

Yet, time and task for generation and editing of images/animations are still quite large and expensive. To speed-up and facilitate these processes, various methods have been proposed in worldwide. Especially, learning-based techniques (e.g. AI and deep learning, etc) have been often used just like other research areas. In the image processing field, image completion and colorization, pencil drawing generation are achieved using these learning-based techniques. The techniques are also used for accelerating physics-based rendering and simulation, in computer graphics. On the other hand, several methods using crowdsourcing are proposed, which can easily create the user-desired image/animation. As above, various techniques are being applied to facilitate/accelerate generation and editing of images and animations, and further development is expected.

Based on this background, we look forward to receiving your papers, system development papers, and material papers in this special issue.

1. Topics covered include but not limited to
   Computer graphics, Modeling, Rendering, Simulation, Stereoscopic, Visualization, AR/MR/VR, Game, Anime, Manga, Entertainment, Motion capture, Wearable sensor, Image processing, Image analysis, Object detection, Image recognition, Motion analysis, Image encoding, 3D shape reconstruction, Computer vision, Big data, Database, Crowdsourcing, Image/video retrieval, Usability, Interface, Interaction, AI, Deep learning, Machine learning, Other related fundamental / application / systemized technologies
2. Treatment of papers
   Submission paper style format and double-blind peer review process are the same as an ordinary contributed paper. If the number of accepted papers is less than the minimum number for the special issue, the acceptance paper will be published as an ordinary contributed paper. We ask for your understanding and cooperation.
3. Publication of Special Issue:
   IEEJ Transactions on Image Electronics and Visual Computing Vo.8, No.1 (June, 2020)
4. Submission Deadline:
   **Friday, November 29, 2019**
5. Contact details for Inquires:
   IIEEJ Office E-mail: hensyu@iieej.org
6. Online Submission URL: http://www.editorialmanager.com/iieej/

# Guidance for Paper Submission

## 1. Submission of Papers

(1) Preparation before submission

- There are three categories of manuscripts as follows:

- Full Paper: It should be a scholarly thesis on a unique study, development or investigation concerning image electronics engineering. This type of paper is classified as an ordinary thesis or a system development thesis for a proposed new system. As a general rule, you are requested to summarize a paper within eight pages.

- Short Paper: It is not yet a completed full paper, but instead a quick report of the result obtained at the preliminary stage as well as the knowledge obtained from said result.

- Data Paper: A summary of data obtained in the process of a survey, product development, test, application, and so on, which are the beneficial information for readers even though its novelty is not high. The length of this paper shall be summarized within four printed pages.

- We prohibit duplicate submission of a paper. If a paper with the same content has been published or submitted to other open publishing forums by the same author or, at least, one co-author, it will not be accepted as a rule. Open publishing implies internal or external books, magazines, bulletins and newsletters from government offices, schools, company organizations, etc. This regulation does not apply to a preliminary draft to be used at an annual meeting, seminar, symposium, conference, and lecture meeting of our society or other societies (including overseas societies). The regulation is not applicable to patent gazette, dissertation and specified publications that the editing committee of our society has approved. But the announcement of such a disclosure must be clarified in the document. A paper that was once approved as a short paper and being submitted again as the full paper after completion is not regarded as a duplicate submission.

(2) Submission of a paper

- To prepare of a paper for submission, an author can download Guidance for Paper Submission or Style Format from our homepage.
  (Note: Delete all author information at the time of submission. But, deletion of reference information is author's discretion. )

- At first, please register your name on the paper submission page of the following URL:
  **http://www.editorialmanager.com/iieej/**

Then, log in again and fill in the necessary information. Use Style Format to upload your manuscript.

- There are TeX and MS Word versions in the Style Format. An applicant should use PDF format (converted dvi of Tex) or MS Word format for the manuscript. In principle, charts shall be inserted into the manuscript.
  (A different type data file, such as audio and video, can be uploaded at the same time for reference.)

- For those who want to submit a paper by E-mail or do not use the Style format, or want to post-mail a paper by hard copy, please consult the editor at our office.

Contact:
Person in charge of editing
The Institute of Image Electronics Engineers of Japan
3-35-4-101, Arakawa, Arakawa-ku, Tokyo 116-0002, Japan
E-mail: hensyu@iieej.org
Tel: +81-3-5615-2893, Fax: +81-3-5615-2894

## 2. Review of Papers and Procedures

(1) Review of a paper

- A manuscript is reviewed by professional reviewers of the relevant field. The reviewer will deem the paper “approved”, “conditionally approved” or “returned”. The applicant is notified of the result of the review by E-mail.
  In the review, novelty, reliability and utility are discussed and evaluated.

(2) Procedure after a review

- In case a paper is accepted, the author prepares a final manuscript (as mentioned in 3. ).

- In the case of a paper that is accepted with comment by the reviewer, the author may revise the paper in consideration of reviewer’s opinion and proceeds to prepare the final manuscript (as mentioned in 3).

- A paper that is not worthy of publishing as it is, but may be published after modification is classified as conditionally accepted. In case of conditional acceptance, the author shall modify a paper based on the reviewer’s requirements by a specified date (within 60 days),

and submits the modified paper for approval.

- In case a paper is returned, the author cannot proceed to the next step. Please look at the reasons the reviewer lists for the return. We expect an applicant to try again after reviewing the content of the paper.

(3) Review request for a revised manuscript

- If you want to submit your paper after it has been conditionally approved, please submit the reply letter, the revised clean manuscript and the reviewed manuscript to the submission page of our site. Please observe the designated date for submission. Outdated manuscripts may be treated as new applications.

- In principle, a revised manuscript shall be reviewed by the same reviewer. It is judged either approved or returned.

- After the judgment, please take the same procedure as (2).

**3. Submission of final manuscript for publication**

(1) Submission of a final manuscript

- An author who received our adoption notice shall prepare a final manuscript and submit it by attaching it to E-mail or by way of CD-ROM. (The contact is the same as mentioned in 1. Paper submission (2))

- The final manuscript shall contain a source file (TeX edition or MS Word version) and a PDF file, eps files for all drawings, an eps file for author's photograph (L:W ratio 2:3, more than 300dpi, upper part of the body) for introduction. Please submit these in a compressed format, such as a zip file.

- Before submission, please fully confirm the attachment and check the items being sent.

Particularly, please check that the description method of a chart and writing method of cited references meet our society's designated style.

In the final manuscript, write the name of the author, name of an organization, introduction of an author, and if necessary, an acknowledgment of appreciation. (In the case of TeX, cancel macros in the Style file.)

An author whose paper is accepted shall pay a page charge before publishing. It is the author's

decision whether to purchase offprints. (ref. Page charge and offprint price information) At least one member or one student member of our society must be included as an author, before publishing.

(2) Galley print proof

- The author is requested to check the galley (hard copy) a couple of weeks before the paper is published in the journal. Please check the galley by the designated date (within one week). After making any corrections, scan the data and return it to our office in PDF file form. At that time, fill in the Offprint Purchase Slip and Copyright Form and return the scanned data to our office in PDF file form.

- In principle, the copyrights of all articles published in our journal, including electronic form, belong to our society.

- You can download the Offprint Purchase Slip and the Copyright Form from the journal on our homepage. (ref. Attachment 2: Offprint Purchase Slip, Attachment 3: Copyright Form)

(3) Publication

- After final proofreading, a paper is published in the online journal (in the electronic database). It is also posted on our homepage.

IIEEJ Trans. on Image Electronics and Visual Computing
Vol. 7, No. 1 2019

Editor in Chief: Mei Kodama
The Institute of Image Electronics Engineers of Japan
3-35-4-101, Arakawa, Arakawa-ku, Tokyo 116-0002, Japan

Print: ISSN 2188-1898
Online: ISSN 2188-1901
CD-ROM: ISSN 2188-191x