https://doi.org/10.2352/EI.2022.34.15.COLOR-261

Image segmentation for content-color-dependent screening (CCDS) using U-net

Altyngul Jumabayeva, Department of Computer Science, School of Engineering and Digital Sciences, Nazarbayev University, Nur-Sultan, Kazakhstan
Adnan Yazici, Department of Computer Science, School of Engineering and Digital Sciences, Nazarbayev University, Nur-Sultan, Kazakhstan

Abstract

In this work, we propose to use deep learning to segment an image based on its color and content. We start by reviewing previously developed content-color-dependent screening (CCDS) presented in [1] [2]. The goal of CCDS is to apply different color assignments for the two or more regular or irregular halftones within the image depending on the local color and content of the image. If the image content locally contains high variance of color and texture, the artifacts due to halftoning will not be as visible as the artifacts in smooth areas of the image [1]. Therefore, the objective of CCDS was to detect the smooth areas of the image and apply the best possible color assignments in those areas. In order to detect smooth areas, the image segmentation algorithm involving the retrieval of the cluster-map and the segmented edge-map was proposed. The main disadvantage of the current approach is that for any given image, the result is highly dependent on the initial parameters, such as the number of clusters, low and high thresholds for edge detection, bilateral filter parameters and others. In this work, we propose to use the well-known U-net architecture to detect the smooth areas of the image, and then apply the well-known K-means algorithm to cluster the image based on color. The U-net is a type of a convolutional neural network (CNN) designed for quick, precise image segmentation, and it is used to predict a label for every single pixel [3]. The architecture of the U-net is suitable for this work because it consists of a contracting path to capture context and a symmetric expanding path that enables precise localization [3]. We believe that using the U-net to detect smooth areas of the image greatly improves the current approach and provides better results.

Introduction

The purpose of our research is to improve print quality in high end digital presses. In [4], we presented an HVS-based model for the superposition of two or more clustered-dot color halftones, which are widely used for electrophotographic printers. The model helps us decide what are the best color assignments for the two or more regular or irregular halftones that minimize the perceived error [4]. After experimentation, we concluded that for different combinations of colorant absorptance values, their corresponding best color assignments may also turn out to be different. Hence, we came up with content-color-dependent screening (CCDS), where we proposed to apply different color assignments within the image depending on the local color and content of the image [1].

Since the artifacts due to halftoning are more visible in smooth areas than they are in the non-smooth areas, our goal has been to detect the smooth areas and to apply the best color assignments in those areas. In [1], we presented a method, where we used the well-known K-means clustering algorithm along with an edge detection algorithm in order to segment an image into clusters. We then used our spatiochromatic HVS-based model for the superposition of four halftones in order to search for the best color assignment in a particular cluster. In [2], Yan and Allebach refined the CCDS approach and made improvements to the existing algorithm. First, they proposed to use the elbow method to automate the selection of number of clusters when using K-means. Next, they used the adaptive bilateral filter, so that the parameters do not need to be manually adjusted based on the visual output results. They also proposed to use the color-aware Sobel edge detector, which is more sensitive to subtle color changes than the previous method. Finally, their updated CCDS algorithm optimized the switching between screens, so that only the most essential switches were performed [2].

The concept of performing image segmentation based on the content and texture of the image was investigated before. For example, in [5], Wang et al. presented an effective method for objective fabric smoothness appearance assessment. In their work, the researchers collected 385 fabric specimens whose smoothness degree had been evaluated manually. They analyzed the relationship between the spatial masking effect in the human visual system (HVS) and the fabric smoothness perceived by the human. Their method outperformed the state-of-the-art methods for fabric smoothness assessment and a series of widely used deep learning methods [5]. In [6], Enshaei et al. designed and implemented an end-to-end UNet-based fully convolutional neural network for automated defect detection in industrial surfaces. The goal of their work was to design a model to detect defective regions inside the textured surfaces [6]. In [7], Aakroum et al. proposed another deep learning neural network to predict the irradiance associated with sky images. Their approach was reported to have an accuracy of 95% [7]. The extraction of memory colors was investigated in [8] and [9].

In this work, we propose a better and a more efficient way to obtain the image segmentation map with smooth areas for the content-color-dependent screening method that was presented in [1] and [2]. The main novelty of our approach is to implement the U-net architecture to detect smooth areas in the image and then apply the well-known K-means algorithm to cluster the image based on color. We believe that using U-net to detect smooth areas of the image greatly improves the existing CCDS method and provides better results compared to the existing method. The first advantage of our proposed approach is that since the main objective is to detect only smooth areas, our model outputs only smooth areas. Whereas in the existing method, we segment both smooth and non-smooth parts of the image. The second advantage of the proposed approach is the efficiancy in terms of run-time. Given N images, the full run-time to output segmentation images using the existing approach is about $25N$ minutes, whereas with the proposed approach it is $(23+0.003N)$ minutes. Therefore, the run-time is greatly reduced with our proposed image segmentation approach.

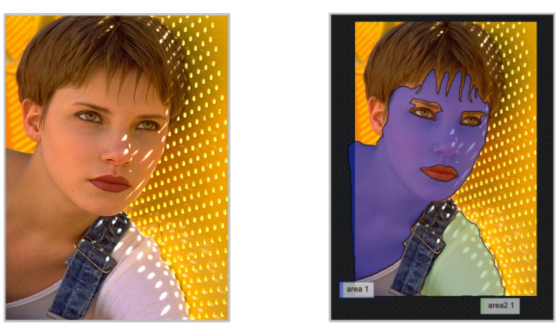
In the next section, we first describe our proposed method. Then, we provide our experimental results by using a 4-fold cross validation, and finally, we conclude the paper.

Methods

The proposed approach of our method is described in three parts. First, we explain how we collected the ground truth data for this work. Second, we provide an overview of our proposed image segmentation algorithm, and finally, we describe the details of how we applied U-net, what updates and changes we made to the existing approach in order to detect smooth areas in an image.

Collection of ground truth data

For our ground truth, we chose to work with original images from the Berkeley segmentation dataset [10]. We selected this dataset because our approach is primarily directed towards segmentation of large smooth areas and this dataset contains various images containing smooth areas with important memory colors, such as flesh tones, sky and others. We then manually annotated images using an image annotation tool called APEER [11]. An example of image annotation is provided in Fig. 1. Figure 1 (a) provides an example of an original mage from the Berkeley segmentation dataset, and Fig. 1 (b) shows the two annotated smooth areas on that image: woman's skin and a woman's t-shirt. Not all images in the Berkeley segmentation dataset contained smooth areas, therefore, out of 300 images, we annotated 128 images, which contained smooth areas. It was decided to use 96 images for training and 32 images for testing.



(a) (b)

Figure 1. *An example of obtaining ground truth using an image annotation tool called APEER: (a) Original image; (b) Annotated ground truth image contating two smooth areas.*

Proposed image segmentation approach

We propose a new image segmentation approach, which consists of two steps. The first step of the approach is summarized in Fig. 2. Here, we took all 128 selected images and annotated them using APEER tool. Every image was then output as images consisting of 0s for every pixel in the non-smooth region, and values 1 to n for every pixel in the smooth region. The maximum number of smooth areas was specified as n. Next, for every image, we combined all smooth areas into one segment, so that ground truth images only consisted of 0s for pixels belonging to a non-smooth region, and 1s for pixels belonging to a smooth region. After that we took 96 images from the ground truth data set and used them to train the U-net architecture.

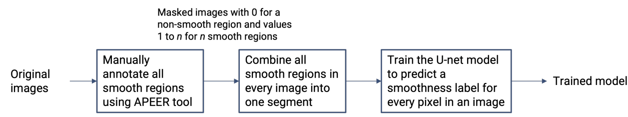


Figure 2. *The block diagram of the first step of the proposed approach.*

The second step of the approach is shown in Fig. 3. After we trained the U-net model, we then used it to predict the smooth regions in any image. Finally, to obtain the final image segmentation map, we proposed to cluster smooth regions using the well-known clustering algorithm called K-means.

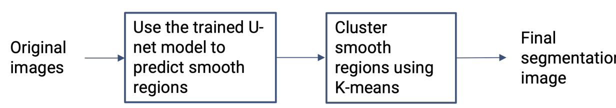


Figure 3. *The block diagram of the second step of the proposed approach.*

Application of U-net

In this work, we propose to use the U-net architecture [3] to detect the smooth areas of the image. We chose to work with the U-net architecture because it was originally designed to work well for various biomedical image segmentation applications, where thousands of training images were usually unavailable but precise segmentation was important [3]. The architecture of the U-net consists of a contracting path to capture context and a symmetric expanding path that enables precise localization [3]. The contracting path is made of the repeated application of two 3x3 convolutions, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation [3]. At each downsampling step the number of feature channels is doubled [3]. The expansive path involves upsampling of the feature map followed by a 2x2 transpose convolution that halves the number of feature channels [3]. It also involves concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU [3].

In our implementation, we followed the original design of the architecture presented in [3], but we had to make some modifications to the hyperparameters in order to train the model for the main objective of this work, which is obtaining smooth regions. The block diagram of the network that we implemented is shown in Fig. 4.

To implement the U-net architecture we used TensorFlow Core Python v2.7.0 [12]. The training images and their corresponding segmentation maps were used to train the network with the adam optimization implementation of Keras [12] [13] [14]. Adam optimization is a stochastic gradient descent method that is

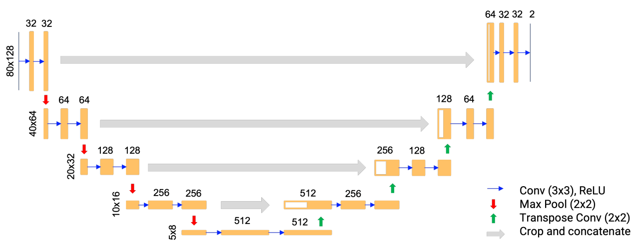


Figure 4. *The block diagram of the U-net architecture used in this work.*

based on adaptive estimation of first-order and second-order moments [13]. We used the learning rate of 0.001, then we decreased the learning rate by a factor of 0.8 every 20 epochs.

Due to the fact that the amount of ground truth data that we obtained was not big, we chose to perform data augmentation. It was decided to use random cropping and hue adjustment. For random cropping, we set the cropping box size to be 75% of the original image size in order to preserve the appearance of smooth and non-smooth regions. As for hue adjustment, the image hue was adjusted by converting the image to HSV and rotating the hue channel (H) by $\delta = (-1, 1)$.

To prevent overfitting, we applied l_2 regularization with a regularization factor of 0.001 in order to constrain neural network's connection weights. In addition, we used dropout in the contracting path. Dropout is one of the most popular techniques for regularization, where a neuron has a probability p of being temporarily "dropped out" during the current training step, but it may be active during the next step [15] [16]. In our work, we used the drop-out probability of 0.3.

For the loss function, we used cross-entropy loss for binary classification. Since, for this work, the classes were imbalanced, i.e. the portion of the non-smooth regions was usually bigger than that of the smooth regions, we used the weighted loss by adding class weights. In the next section, the experimental results of our approach are presented.

Experimental results

We evaluate our model by using cross-validation. The K-fold cross-validation involves splitting the data set into K folds, and obtaining predictions on each fold using a model trained on the remaining folds [17]. For this work, we use a 4-fold cross-validation. In each fold, 96 images were used for training, and 32 images were used for testing. Table 1 summarizes the results that we obtained for 4 folds as well as their average values.

Experimental results: 4-fold validation

Fold	Training / Testing accuracy	R	P	S	F_1 Score
1st	0.83/0.84	0.87	0.76	0.83	0.81
2nd	0.88/0.88	0.80	0.83	0.92	0.82
3rd	0.88/0.85	0.84	0.76	0.86	0.80
4th	0.89/0.87	0.76	0.80	0.92	0.78
Average	0.87/0.86	0.82	0.79	0.89	0.80

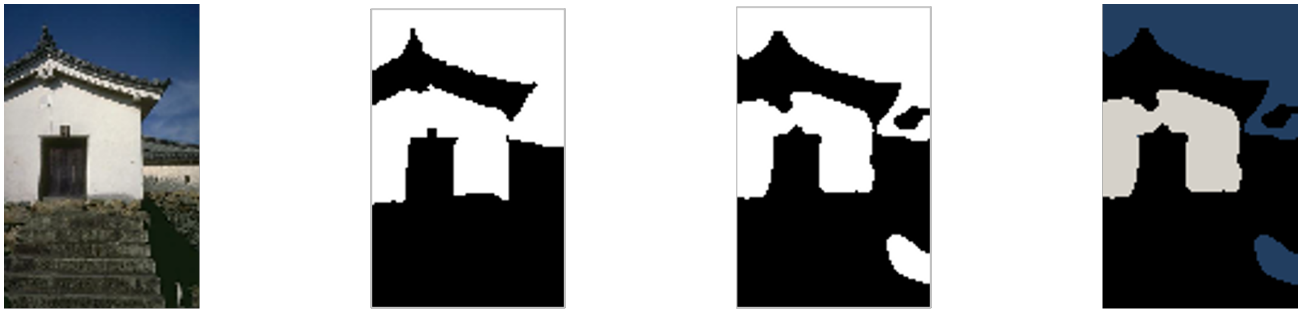
In the table, training and testing accuracies were obtained by using the categorical accuracy, which calculates how often predictions match ground truth labels. On average, the training accuracy is 0.87, and the testing accuracy is 0.86. Next, we computed the confusion matrices for each fold and obtained Recall (R), Precision (P), Specificity (S), and F_1 Score values. Recall is the ratio of positive instances that are correctly detected by the classifier [17]. In case of our experiments, on average, recall is 0.82, which means that 82% of all smooth regions were correctly detected by our model. Specificity shows how good a model is at avoiding false alarms [17]. We computed the average specificity to be 0.89. Precision is the accuracy of the positive predictions [17]. For our work, precision turned out to be 0.79, which means that 79% of all positive predictions were in fact smooth regions. Finally, the metric that combines both recall and precision is called F_1-score, and it was computed to be 80% for our experiments.

In order to visualize our experimental results, it was decided to show three examples shown in Figs. 5, 6, 7. Figure 5 (a) shows an original image of a house with stairs and sky on the background. The ground truth image in (b) shows the smooth area colored in white and the non-smooth area colored in black. The smooth area consists of the walls of the house and the sky on the backgound, whereas the non-smooth area is the rest of the image. The predicted image in (c) looks similar to the ground truth image, however, there is one additional detected smooth area in the bottom right corner. By looking at the original image in (a), we can see that there is in fact a smooth black region in the bottom right corner, which means that the predicted image looks correct with respect to the original image. Finally, Fig. 5 (d) contains the final segmentation map after we apply the clustering K-means algorithm for smooth area. We can conclude that the two main smooth areas were clustered correctly. In the second example provided in Fig. 6, it can be seen that the smooth background was detected and clustered correctly and the smooth area prediction result in (c) is very close to the ground truth image in (b). In the third example in Fig. 7, we present an image, which has a lot of smooth areas, such as the building itself, its red roof and the background sky. The ground truth image in (b) demonstrates that we selected all of the mentioned smooth areas except for the edges. The predicted image in (c) differs from the ground truth image in (b) by not including the left tower, which does include some detail and can be considered both smooth and not smooth. In the future work, we plan to add a specific measure as to what we consider to be smooth in order to avoid such errors. Finally, the final segmentation image in (d) was clustered correctly with K-means.

In the most recent version of CCDS that was presented in [2], the run-time to output a segmentation map for a single image was about 25 minutes. Using our proposed method, the run-time for training 96 images is about 23 minutes for 100 epochs. After the model is trained, the run-time to predict an output for a single image was about 0.003 minutes. Hence, given N images, the full run-time to output segmentation images using the existing approach is about $25N$ minutes, whereas with the proposed approach it is $(23 + 0.003N)$ minutes. Therefore, the run-time is greatly reduced with our proposed image segmentation approach using the U-net architecture.

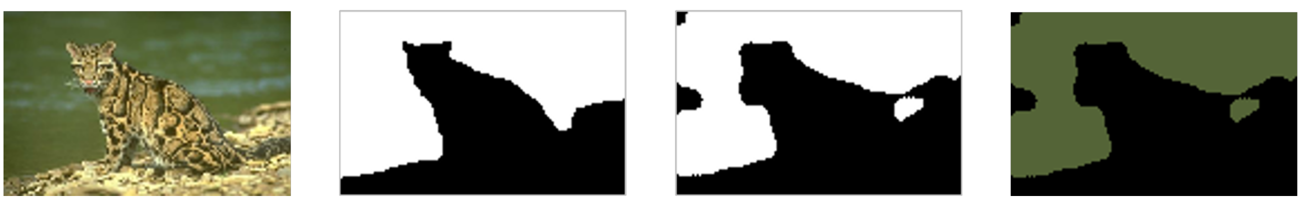
Conclusion

In conclusion, we propose a new and a more efficient approach to obtain the image segmentation with smooth areas for the content-color-dependent screening (CCDS) method, which can be used to improve print quality in high end digital presses. The main novelty of our approach is to implement the U-net architecture to detect smooth areas in an image and then apply the well-known K-means algorithm to cluster the image based on color. In this work, we showed how we collected the ground truth data, and described the details and updates of the implementation of the U-net architecture. The final F_1 score value is 80%, which can be improved by adding more ground truth data in the future. Finally, we provide experimental results and examples, and we conclude that the proposed method of detecting smooth areas is more efficient in terms of run-time than the current existing method.



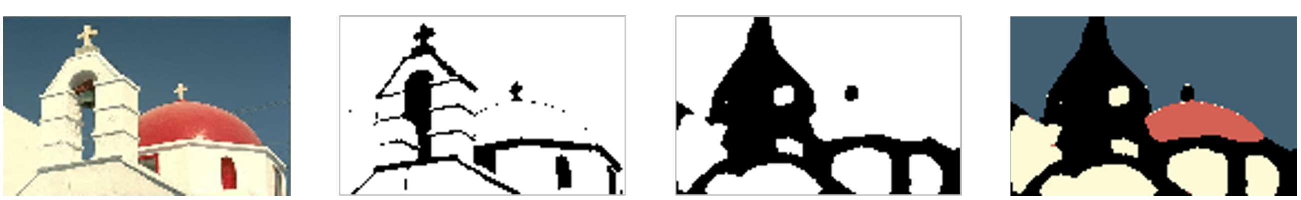
(a) Original image (b) Ground truth image (c) Predicted image (d) Final segmentation image

Figure 5. *Example 1 of obtaining the final image segmentation.*



(a) Original image (b) Ground truth image (c) Predicted image (d) Final segmentation image

Figure 6. *Example 2 of obtaining the final image segmentation.*



(a) Original image (b) Ground truth image (c) Predicted image (d) Final segmentation image

Figure 7. *Example 3 of obtaining the final image segmentation.*

References

[1] A. Jumabayeva, T. Frank, Y. Ben-Shoshan, R. Ulichney, and J. Allebach, "Content-color-dependent screening (CCDS) using regular or irregular clustered-dot halftones," in *Proc. 2018 7th Eur. Workshop Visual Inform. Process. (EUVIP)*, Tampere, Finland, Nov. 2018, doi:10.1109/EUVIP.2018.8611727.

[2] Y. Yan and J. P. Allebach, "Improved content-color-dependent screening (CCDS): Adaptive bilateral filtering and color-aware sobel edge detector," in *Proc. IS&T Electron. Imaging, 26th Color Imaging: Displaying, Process., Hardcopy, Appl.*, Jan. 2021, doi: https://doi.org/10.2352/ISSN.2470-1173.2021.16.COLOR-252.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Medical Image Comp. Computer-Assisted Intervention (MICCAI 2015)*, Munich, Germany, Oct. 2015, pp. 234–241.

[4] A. Jumabayeva, T. Frank, Y. Ben-Shohan, R. Ulichney, and J. P. Allebach, "HVS-based model for superposition of two color halftones," in *Proc. IS&T Electron. Imaging, 21st Color Imaging: Displaying, Process., Hardcopy, Appl.*, San-Francisco, CA, Feb. 2016, doi:10.2352/ISSN.2470-1173.2016.20.COLOR-311.

[5] J. Wang, K. Shi, L. Wang, Z. Li, R. Pan, and W. Gao, "An objective fabric smoothness assessment method based on a multi-scale spatial masking model," *Multidisciplinary open access journal (IEEE Access)*, vol. 15, pp. 73 830 – 73 840, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2918845.

[6] N. Enshaei, S. Ahmad, and F. Naderkhani, "Automated detection of textured-surface defects using unet-based semantic segmentation network," in *Proc. 2020 IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Jun. 2020, doi: https://doi.org/10.1109/ICPHM49022.2020.9187023.

[7] M. Aakroum, A. Ahogho, A. Aaqir, and A. A. Ahajjam, "Deep learning for inferring the surface solar irradiance from sky imagery," in *2017 Int. Renewable Sustainable Energy Conf. (IRSEC)*, Dec. 2017, doi: https://doi.org/10.1109/IRSEC.2017.8477236.

[8] B. T. Ryu, J. Y. Yeom, C.-W. Kim, J.-Y. Ahn, D.-W. Kang, and H.-H. Shin, "Extraction of memory colors for preferred color correction in digital TVs," in *Proc. SPIE, 14th Color Imaging: Displaying, Process., Hardcopy, Appl.*, Jan. 2009, doi: https://doi.org/10.1117/12.806090.

[9] M. Jaber, E. Saber, and F. Sahin, "Extraction of memory colors using bayesian networks," in *Proc. 2009 IEEE Int. Conf. System of Systems Eng. (SoSE)*, May 2009.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int. Conf. Computer Vision*, Vancouver, BC, Canada, Jul. 2001, pp. 416–423, doi: https://doi.org/10.1109/ICCV.2001.937655.

[11] (2021) APEER by Zeiss. [Online]. Available: https://www.apeer.com/home/

[12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learning Representations (ICLR 2015)*, San-diego, CA, USA, May 2015.

[14] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[15] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," https://arxiv.org/abs/1207.0580, 2012.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[17] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras and Tensorflow*. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc., 2019.

Author Biography

Dr. Altyngul Jumabayeva is a Postdoctoral Scholar in the School of Engineering and Digital Sciences Department of Computer Science at Nazarbayev University in Nur-Sultan, Kazakhstan. She obtained her B.S and Ph.D. degrees in Electrical Engineering from Purdue University, IN, USA, in 2014 and 2019, respectively. Her current research interests include image processing, halftoning, computer vision, deep learning, and others.

Prof. Dr. Adnan Yazici received a Ph.D. degree in computer science from the Department of Computer Science, Tulane University, USA, in 1991. He is a Full Professor and Chair of the Department of Computer Science at Nazarbayev University, Nur-Sultan, Kazakhstan, where he is also the Director of the Multimedia Laboratory. He has published over 200 international technical papers and co-authored/edited three books entitled Fuzzy Database Modeling (Springer), Fuzzy Logic in its 50th Year: New Developments, Directions and Challenges (Springer), and Uncertainty Approaches for Spatial Data Modeling and Processing: A Decision Support Perspective (Springer). His current research interests include data science, video data modeling and retrieval, wireless multimedia sensor networks, and fuzzy database modeling. He was a recipient of the IBM Faculty Award in 2011 and the Parlar Foundation's Young Investigator Award in 2001. He was the Conference Co-Chair of the 23rd IEEE International Conference on Data Engineering in 2007, the 38th Very Large Data Bases in 2012, and the 23rd IEEE International Conference on Fuzzy Systems in 2015. He is currently an Associate Editor of the IEEE Transactions on Fuzzy Systems and a member of ACM, the IEEE Computational Intelligence Society, and the Fuzzy Systems Technical Committee.