計測自動制御学会東北支部 第 340 回研究集会 (2022.12.23)
資料番号 340-3

意味推定と深度推定を用いたニューラルネットワークによる TVCM の分類

Neural Network Models for TV Advertisement Classification using Semantic Segmentation and Depth Estimation

○大谷　達哉 *，森 和好 **

○Tatsuya OTANI*, Kazuyoshi MORI**

*会津大学大学院コンピュータ理工学研究科, **会津大学 コンピュータ理工学部

*The University of Aizu Graduate School, **The University of Aizu.

キーワード： テレビ CM (TVCM), 畳み込みニューラルネットワーク (CNN), 畳み込みリカレントニューラルネットワーク (CRNN), 意味推定 (Semantic Segmentation), 深度推定 (Depth Estimation).

連絡先： 〒965-8580　福島県会津若松市一箕町鶴賀字上居合９０　会津大学コンピュータ理工学部
森　和好， Tel.: (0242)37-2615， Fax.: (0242)37-2747， E-mail: k-mori@u-aizu.ac.jp

1. Introduction

Television commercials (TVCMs) provide various information about products and services to sell them [1]. TVCMs reflect the social situation and trend at that time. These often affect people and social culture. We consider that investigating many TVCMs is effective for social analysis. Thus, we have investigated a semiautomatic system for classifying TVCMs. This implementation has been based on the models of convolutional neural networks (CNNs) [2] and recurrent neural networks (RNNs) [3]. TVCM classification based on image recognition using convolutional neural networks essentially needs to prepare many training data, that is, many TVCMs. Unfortunately, it is difficult to collect TVCMs because there seems no such kind of open database. Therefore, in order to make the neural network training more efficient with not so many TVCMs, we propose adding Supplementary Data, like in Figure 1, to support TVCM classification. In this paper, we propose to employ Semantic Segmentation and Depth Estimation as Supplementary Data.

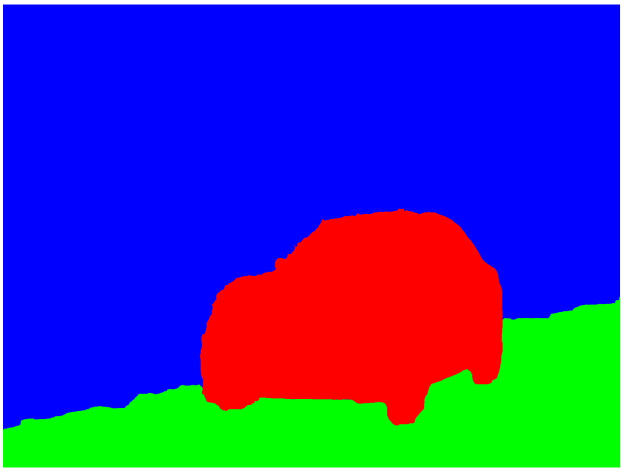

Fig. 1　One example of Supplementary Data.

2. Neural Network

The neural network is a computing system modeled on humans on the brain and nervous system, often used in pattern recognition, such as character recognition and speech recognition. A neural network consists of multiple node layers, including an input layer, one or more hidden layers, and an output layer.

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the neural networks with deep learning [2,4]. This is often used for image recognition. CNN consists of convolutional layers and pooling layers. The convolutional layer outputs feature maps from the input array. The pooling layer takes the output of the convolution layer as input and summarizes the values in a relatively small specific region. In this research, the maximum value is used as a summary.

2.2 Recurrent Neural Network

Recurrent Neural Network (RNN) is also one of the neural networks with deep learning. This method is often used to recognize time-series data. Units in the RNN are used recursively. This means the output of the units is input into the same unit along with the next data, as shown in Figure 2. There are several types of units, and Long Short-term Memory (LSTM) is one of them [3]. LSTM can hold output before the previous turn, but simple units cannot.

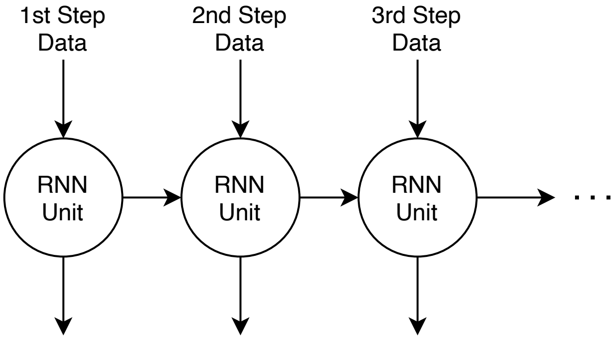

Fig. 2 Recurrent Neural Network example.

3. Supplementary data

In classification with only images extracted from TVCMs, the training of a neural network is not good in accuracy, efficiency, and generalization performance, which means that it could not adapt appropriately to data the network has not seen. 1In addition, the training needs a large amount of data. However, there is a limit to the number of TVCM videos we can collect. Therefore, we consider creating data which assist the training of neural networks and input them along with the original. We call this created data ”Supplementary Data.”

Supplementary Data correspond to the images extracted from the TVCM videos. Figures 1 and 3 show examples of Supplementary Data. We consider it very important how to represent Supplementary Data. In the early stages of our research, we considered developing a neural network of generators of Supplementary Data, as shown in Figure 4. However, U-net [5, 6] and GAN [8], known as standard generators, require large amounts of training datasets and high-performance GPUs for development. Then, we considered that it was difficult to develop an original Supplementary Data generator using our available resources.

Therefore, we considered using the existing structure to create Supplementary data. Using this data aimed to reduce our workload and create a large amount of good-quality data. In this research, We considered two types of supplementary data.

Fig. 3 Image extracted from a TVCM video.

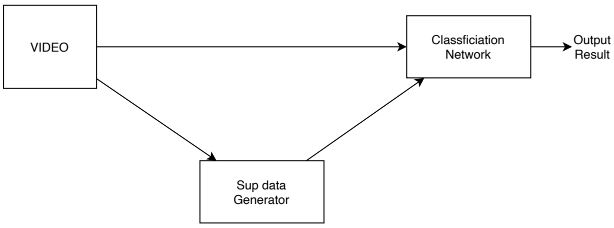

Fig. 4 Our first idea for classification network structure using the supplementary data generator.

3.1 How to adapt supplementary data

We need to consider first the way to adapt supplementary data to the neural network. Our first idea was to input supplementary data into the neural network as the fourth channel of the original image, as shown in Figure 5. However, we observed that this would not maximize the impact of supplementary data. We consider that this is because the original images and the supplementary data represent different aspects. Therefore, we decided to input the Supplementary Data into another independent CNN and combine its output with the output of the CNN of the original image.

To achieve this, we add the output of the pooling layer of the CNN of the Supplementary Data, which we call the Supplementary Stream, to the output of the pooling layer of the CNN of the original data, which we call the Original Stream, as shown in Figure 6. This addition is to increase the value of the pixels in the original image to which the Supplementary Data points. This method makes it easier for the neural network model to recognize features.

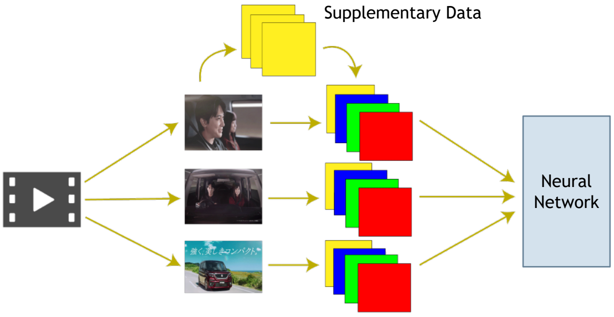

Fig. 5 First idea for adapting supplementary data for input to the neural network.

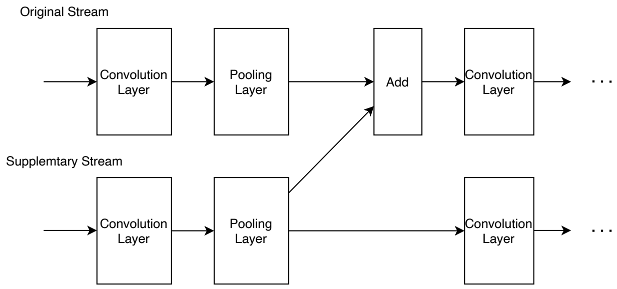

Fig. 6 CNN structure to adapt supplementary data.

3.2 Semantic Segmentation Supplementary Data

Semantic segmentation [7] predicts classes of objects to which a single pixel belongs shown in Figure 1. We used the DeepLabV3 model with ResNet-101 in PyTorch vision API for our semantic segmentation [9]. This model can classify a pixel into 21 classes. Figure 7 is an example of the semantic segmentation.

The Supplementary Data based on the semantic segmentation is to help the neural network to understand the main object of input images. We aimed to improve accuracy and training efficiency with this Supplementary Data.

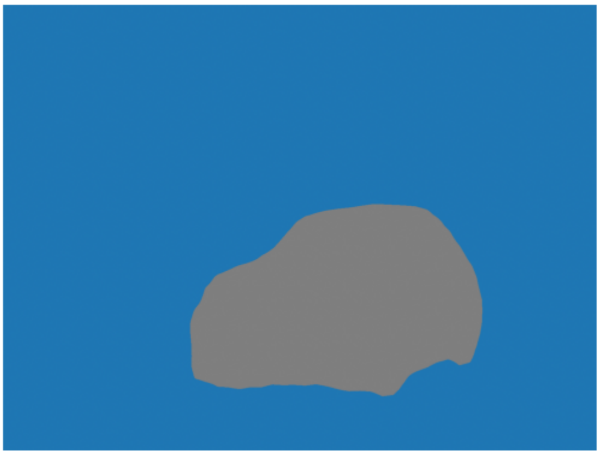

Fig. 7 An image of semantic segmentation.

3.3 Depth Estimation Supplementary Data

Depth Estimation recognizes the input image and estimates the distance from the shooting position to the object [10]. In this study, we used a model of monocular depth estimation model, which is called Midas. Figure 8 is an example of the depth estimation. The depth estimation data will enable us to clarify the shapes of objects in front of the image. We also expect to improve accuracy and training efficiency with this Supplementary Data.

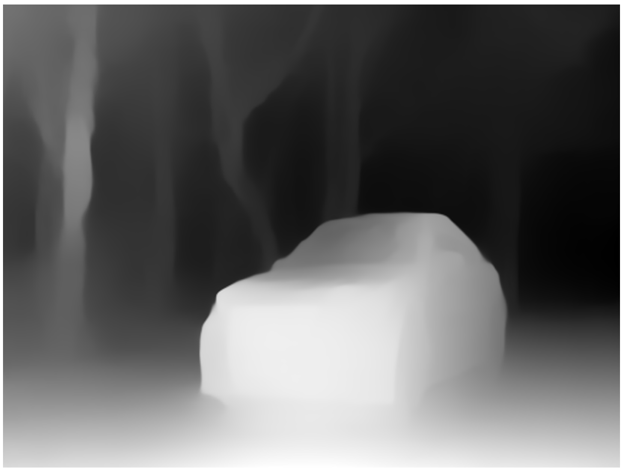

Fig. 8 An image of depth estimation.

4. Experiment

In the experiments in this paper, we trained neural network models on each type of Supplementary Data. After that, we compared the evaluation of these models using a test dataset.

4.1 Dataset

We prepared 600 TVCMs. These have two categories: “Car” and “Drink.” Each category has 300 TVCMs. We extracted 30 images from one TVCM video.

For Semantic Segmentation Supplementary Data, we input the 30 images extracted from one TVCM video into the Semantic Segmentation network. The raw output of the Semantic Segmentation network is the label numbers of the classes. If this output were directly used, the difference in values would represent some specific meaning. Therefore, we decided to use a color palette to represent the classes.

For Depth Estimation Supplementary Data, we input the 30 images extracted from one TVCM video into the Depth Estimation network. The output values are normalized between 0 and 1.

4.2 Neural Network Model

In our approach, we used Convolutional Recurrent Neural Network (CRNN) as in Figure 9. Images extracted from an input video are processed by convolution and pooling layers as the first step. Feature maps are passed to the input of LSTM units. After that, full-connected layers output the classification result. The supplementary data is input into another stream separated from the original image stream. Output values of each pooling layer on the Supplementary Stream are added to the output of the pooling layer on the original stream of the same size. About LSTM, we used the Backward LSTM and Bi-LSTM [3]. We consider that this structure can find highly accurate temporal features because humans sometimes understand the meaning of commercials by going backward.

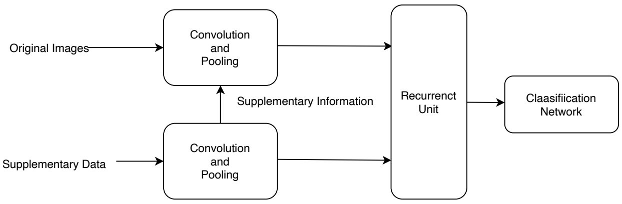

Fig. 9 Neural network structure with supplementary data.

4.3 Training

We used 600 TVCMs we collected and divided them into five groups. Four groups were used for training, and the remaining one group was used as the evaluation dataset. We trained and evaluated five different dataset combinations for training and testing. We then compute the average of the test results for each type of data set. We forced the neural network to train for 25 epochs. The batch size is 32.

5. Result and Conclusion

Models using Semantic Segmentation Supplementary Data and Depth Estimation Supplementary Data were trained successfully, as shown in Figures 10, 11, and 12. In our experiment, results shown in Table 1 show that the model using Semantic Segmentation Supplementary Data produced better accuracy results than the model using Depth Estimation Supplementary Data. We considered that this result was caused by the fact that the outline of the objects represented by the Depth Estimation Neural Network is often unclear. Models using Supplementary Data produced more accurate classifications than the model without Supplementary Data. According to efficiency, the results in Table 2 show that the number of parameters of each model using Supplementary Data increased by 12%, and the training time increased by 50% than models without Supplementary Data. However, about the number of epochs required for the accuracy rate to exceed 90% for the first time during training, as shown in Table 3, the models using Semantic Segmentation Supplementary Data and Depth Estimation Segmentation Supplementary Data were less than the model without Supplementary data. Therefore, we considered that supplementary data is effective in terms of learning efficiency and accuracy in our neural network structure.

6. Future Work

In this research, we found that Supplementary Data is effective in classifying TVCMs. Thus, it is necessary to consider the structure of the neural networks that can use Supplementary Data more effectively. In particular, we need to improve the CNN part by referring to structures such as VGG [11] and ResNet [12]. In addition, we classified two classes in this research: ”Car” and ”Drink.” It is necessary to add other classes and evaluate them, such as Food, Cosmetic items, and more.

References

1) 山田 奨治, テレビ・ コマーシャルと文化研究 (Television commercial as a resource for Japanese studies: past and present), **日本研究**, **35**(5), pp.527–536, 2007.

2) A. Krizhevsky, I. Sutskever, G.E. Hinton, *ImageNet classification with deep convolutional neural networks*, 2017.

3) I.J. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press 2016.

4) Y. LeCun, Y. Bengio, G. Hinton, Deep Learning, *Nature*, **521**(5), pp.436–444, 2015, DOI: 10.1038/nature14539.

5) O. Ronneberger, P. Fischer, T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015, arXiv:1505.04597 .

6) Z. Zhang, Q. Liu, Y. Wang, Road Extraction by Deep Residual U-Net, *IEEE Geoscience and Remote Sensing Letters*, **15**(5), pp.749–753, 2018, DOI: 10.1109/LGRS.2018.2802944.

7) L.C. Chen., Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, *Proceedings of the European conference on computer vision 2018 (ECCV 2018, LNCS 11211)*, 2018, DOI: 10.1007/978-3-030-01234-2_49.

8) I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Generative Adversarial Networks*, 2014, arXiv:1406.2661 .

9) DeeplabV3 resnet101, https://pytorch.org/hub/pytorch_vision_deeplabv3_resnet101/

10) R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, V. Koltun, *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer*, 2020, arXiv:1907.01341v3 .

11) K. Simonyan, A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2015, arXiv:1409.1556 .

12) K. He, X. Zhang, S. Ren, J. Sun, *Deep Residual Learning for Image Recognition*, 2015, arXiv:1512.03385 .

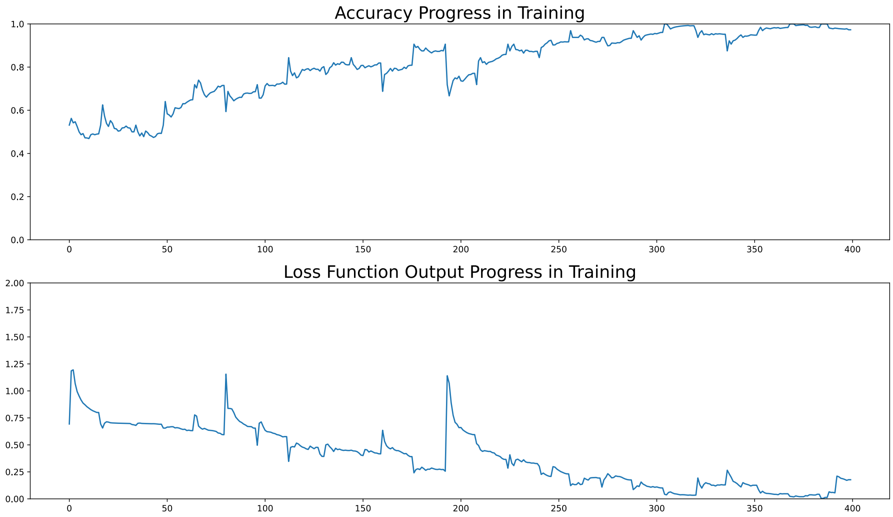

Fig. 10 Training progress of model without Supplementary Data.

Architecture	Loss(average)	Accuracy(average)
No Supplementary Data	0.58	0.83
Semantic Segmentation Supplementary Data	0.42	0.93
Depth Estimation Supplementary Data	0.70	0.88

Table 1 Averages of the evaluation score.

Architecture	Parameters	Training Time(seconds)
No Supplementary Data	23,871,810	128
Semantic Segmentation Supplementary Data	26,734,082	177
Depth Estimation Supplementary Data	26,733,506	180

Table 2 Trainable parameters and training time.

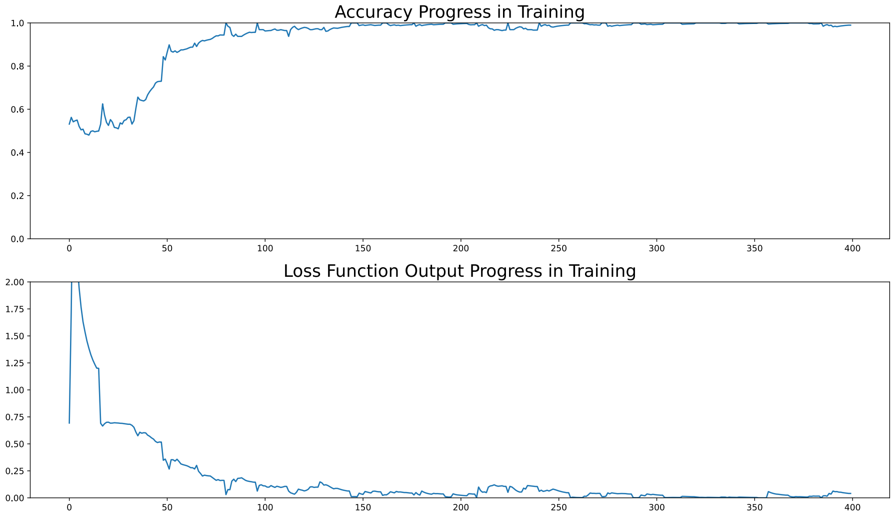

Fig. 11 Training progress of model using Semantic Segmentation Supplementary Data.

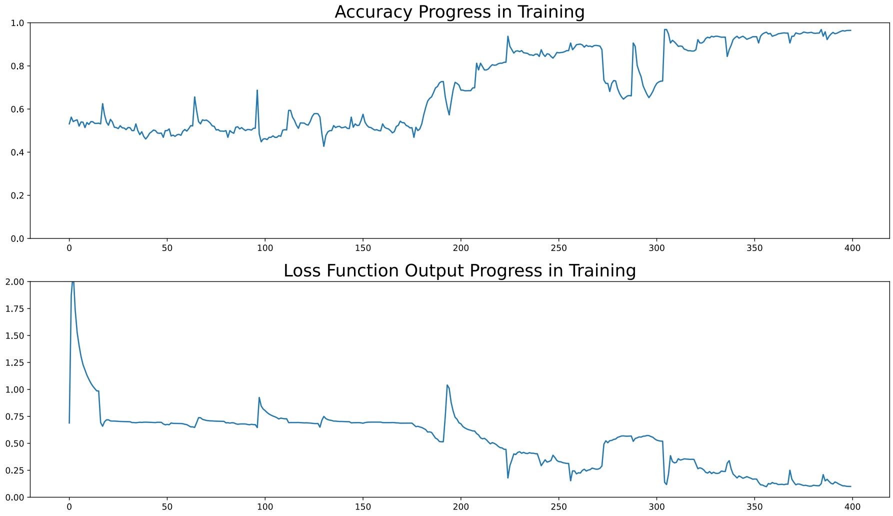

Fig. 12 Training progress of model using Depth Estimation Supplementary Data.

Architecture	Step over 90% accuracy
No Supplementary Data	13.60
Semantic Segmentation Supplementary Data	3.74
Depth Estimation Supplementary Data	8.28

Table 3 The average number of epochs required for Accuracy rate to exceed 90% during training.