Object and Human Action Recognition From Video Using Deep Learning Models

Padmeswari Nandiya Soentanto, Janson Hendryli, Dyah E. Herwindiati
Faculty of Information Technology
Universitas Tarumanagara
Jakarta, Indonesia
Email: ri_nandiya@yahoo.com, jansonh@fti.untar.ac.id, dyahh@fti.untar.ac.id

***Abstract*—This paper explores the deep learning models aiming at two tasks, which are classifying objects and recognizing human action from a video. The deep learning models are the convolutional neural networks and long short-term memory network. For the action recognition, the optical flow is employed as the feature representation of movement on each video. The video data simulates one person doing either taking, returning, or browsing items on a shelf. From the experiments, the model achieve accuracy of 56.41% of accuracy for the object classification task and 76.92% for the action recognition.**

I. Introduction

Nowadays, the existence of convenience stores, minimarkets or retail markets has spread throughout the country, especially in Indonesia. The people depend on these stores to buy their daily needs. One of the problems in such stores is that the shoppers have to queue and wait for relatively long time in busy hours due to limited number of cashiers.

With recent advancement in computer vision, the technology can be utilized to solve this problem and one of the ideas is to automate the checkout process by actively tracking the shoppers around the store and detecting which items are added to or removed from the shopping cart. When the customers finish shopping, they can then directly proceed to the payment process, which can also happen automatically when the system is connected to a specific payment system.

This paper explores the deep learning models to detect human action recognition of taking, returning, or browsing items from a shelf and also concurrently detect which item is taken or returned. These models are the important parts of building an automatic checkout system.

There are two models trained on the video dataset, which are the object classification model and the action recognition model. For the action recognition model, initially the system will extract information related to the human movement. The information are represented by the optical flow between two video frames and fed into the convolutional neural networks and long short-term memory networks to classify the action. Meanwhile, the object classification model uses only the convolutional neural networks and the video frames as input. Finally, we will evaluate the accuracy of the models in recognizing the human action and classifying the objects on the video.

II. Related Works

The research in video recognition has been known and famous by the advances in image recognition methods, which were often adapted and modified to handle video data, such as the promising results of [1] who extended convolutional neural network model into spatial-temporal space by operating on stacked video frames, also by [2] who compared several architectures for action recognition, and [3] who introduced an interesting two-stream approach in convolutional neural network to clasify movements. The recurrent neural networks has also been shown to be effective for handling sequential data, such as speech recognition [4] and image [5] or video description [6]. For long-term temporal modeling of the video data, [7] proposed encoder-decoder framework in long short-term memory to learn video representations in an unsupervised manner.

In [8], the authors propose a graphical Bayesian model for recognizing interactions of human and object. The work resembles the aim of this paper, which is building a system to recognize action and, at the same time, classifying the object from a video, although it only concerns with the interactions of objects and the actions.

III. Methods

A. Optical Flow

Optical flow is the pattern of moving objects from one frame to another frame caused by either the movement of the object itself or the camera. Optical flow works with the assumption that the intensity of one pixel from one object will not change between sequential frames and neighborhood pixels that have the same movement. Assume $I[x, y, t]$, an image located on x and y in t second, is a center pixel in $N \times N$ matrix that moves as far as δ_x and δ_y in δ_t second and turns into $I[x + \delta_x, y + \delta_y, t + \delta_t]$ [9]. Since $I[x, y, t] = I[x + \delta_x, y + \delta_y, t + \delta_t]$, using the first order Taylor expansion, we obtain:

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0 \quad (1)$$

where $v_x = \frac{\delta_x}{\delta_t}$ and $v_y = \frac{\delta_y}{\delta_t}$ are the optical flow. In this research, we use the Gunnar-Farnebck method which produces dense optical flow, that is flow information computed for every pixel in the frame.

978-1-7281-2177-2/19/$31.00 ©2019 IEEE

B. Deep Learning Models

Convolutional neural network, also commonly known as CNN or convnet, is a special type of multi-layer neural network. CNN is developed to recognize visual patterns from image pixels directly with a minimum process. It can recognize the pattern with extreme variability and with robustness to simplify geometric distortions and transformations. The convolutional neural network has two main algorithms which are convolution, a process using filter matrix to filter important features from input image, and pooling, a process that will turn several pixels in one neighborhood to become one single pixel. The convolution operation is generally used to blur, sharpen, detect edges, or any filtering effects on an image. This operation is accomplished using a kernel matrix or filter. There are several notable kernel matrices with their own effects. For example, using a Sobel filter [10], one can locate edges of an image. Different filter such as one derived from the Gaussian function can blur or smooth an image. In traditional image processing, such filters are defined and used to extract particular features from an input image and fed to a classification model. The main idea of a convolutional neural network is to train such filter and find the best feature representation for a classification task. Stacking several layers of convolutional image and ending with fully-connected neural networks as the classification model results is what we call the convolutional neural network [11].

Long short-term memory network or LSTM is introduced by [12] to address the limitations of the recurrent networks such as vanishing and exploding error signals during the learning process [13]. The LSTM itself consists of memory blocks which composed of an input gate, output gate, forget gate, and a cell [14]. The forget gate can adaptively reset the cell's memory which essentially controls information to be retained or thrown away from the cell state. The architecture of the LSTM makes it suitable to be used for sequential data such as videos. Recently, there have been successful applications of LSTM in speech recognition [4], language modeling [15], [16], image captioning [17], and many more.

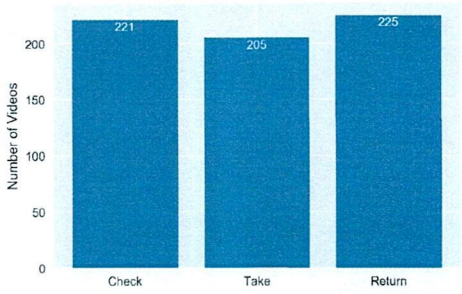
IV. Experimental Settings

In this section, the experimental settings are described, including the video data collected for the experiments and the implementation details.

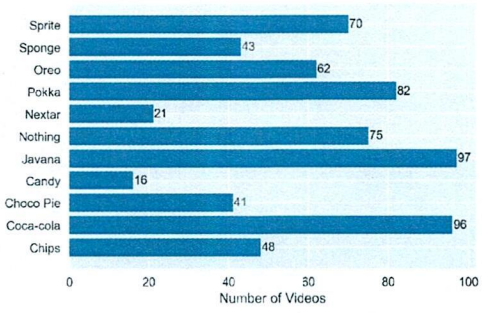
A. Data

The video data set consists of 651 simulated videos recorded from a mobile phone. There is only one human subject on each of the videos who simulates the action of taking an object or item from a shelf, returning the taken item back to the shelf, or checking the item on the shelf without taking anything. In each video, there is only one human subject, one action, one item (if the action is either taking or returning the item), and one shelf displaying all the objects.

All video clips in the data set are shot in a fixed frame rate of 29.35 fps with a resolution of 720 x 1280 pixels. It should be noted that there are several noises in the video, such as camera shaking and cluttered background.



(a) The total number of videos for the action recognition.



(b) The total number of videos for each objects or items.

Fig. 1. The number of videos for each action and object classes in the data set. 1a There are three classes for the action recognition task, which are checking items, taking an item from the shelf, and returning the taken item to the shelf. Meanwhile, 1b shows the total number of videos for the 10 objects of items. Also shown is the total number of videos where the human subject does not take any item.

The data will be split arbitrarily with 70% for training and 30% for testing in a stratified way. Fig. 1 shows the total number of videos for both action and object classification.

B. Implementation

Preprocessing. For the action recognition, the videos are initially resized to 180 x 320 pixels and converted to gray-scale images. Afterwards, the optical flow between two frames is computed. It should be noted that the length of each video in the data set varies. Consequently, the number of frames also varies. In the experiments, 60 frames are extracted from each video to get the optical flow of each consequent images. By stacking the horizontal and vertical displacement of the optical flow, each video is represented by a matrix of size 360 x 320 x 30 pixels.

The object recognition preprocessing step is identical to the movement classification above. Although, the images are not gray-scaled and the frames are resized to 360 x 320 pixels instead. The frames are also normalized by dividing each

978-1-7281-2177-2/19/$31.00 ©2019 IEEE

pixel with 256. From each video, 10 frames are selected and represented by a matrix of size 360 x 320 x 3 pixels.

Models. The human action recognition and object classification model are shown in Fig. 2. After preprocessing the input video as described above, the input matrices are fed into several convolutional layers, which are 12 convolutional layers for the object classification model and 4 convolutional layers for the human action recognition model. Batch normalization is employed only in the convolutional steps of the action recognition model. The output of the convolutional layers is then flattened and fed into the next step. Before classifying the input representation using the fully-connected network, the action recognition model employs a long short-term memory (LSTM) layer with 1000 hidden units. Table I and II show the detail of each layer in the models. Additionally, in the action recognition model, the batch normalization is employed on every convolutional layer.

The models are implemented in Python 3 using the TensorFlow and Keras library. The object classification model is trained using RMSprop optimizers and categorical cross-entropy loss with learning rate of 0.00006. Meanwhile, the action recognition model is trained similarly with RMSprop and categorical cross-entropy loss, but with learning rate of 0.00008.

V. Results

A. Object Classification

There are 11 kind of items for the object classification task. Fig. 3 shows the plot of training loss and accuracy of the model. The figure shows that the training loss converges and the model accuracy is 95.61%. Testing the model with the test data results in 56.41% in accuracy as shown in Fig. 4. Clearly, the model overfit with some items such as Chocopie, which cannot be recognized most of the time. The precision, recall, and F1-score of the testing can be seen in Fig. 5.

B. Action Recognition

Fig. 6 shows the training loss and accuracy of the action recognition model. The accuracy of the model is 80.17%. Testing the model on test data, the confusion matrix in Fig. 7 and the precision, recall, and F1-score as in Fig. 8 are obtained. The average F1-score of the model is 76.94% with the checking items movement has the lowest F1-score and recall, but with the highest precision.

C. Discussions

In the action recognition task, the best model achieved accuracy of 76.92%, where it can correctly recognize 150 actions from the total of 195 actions in the testing data. From the three movements, the model can recognize the checking items movement more accurately than the other movements.

Meanwhile, the object classification model gains worse accuracy than the other model. The highest accuracy obtained by the model from the testing data is 56.41% where the model recognizes 110 items from 195 videos. The best item achieved 94.12% in accuracy, although the model clearly overfit.

TABLE I
THE MODEL ARCHITECTURE FOR OBJECT RECOGNITION TASK, WHERE PAD IS THE ZERO PADDING OPERATION, CONV IS THE CONVOLUTION PROCESS, POOL IS THE MAX POOLING, AND FC IS THE FULLY-CONNECTED LAYER.

Layer	Parameters
PAD	size = (1, 1)
CONV	filter = 64, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
CONV	filter = 64, kernel size = 3 × 3, activation = ReLU
POOL	size = 2 × 2, stride = 2
PAD	size = (1, 1)
CONV	filter = 128, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
CONV	filter = 128, kernel size = 3 × 3, activation = ReLU
POOL	size = 2 × 2, stride = 2
PAD	size = (1, 1)
CONV	filter = 256, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
CONV	filter = 256, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
PAD	size = (1, 1)
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
POOL	size = 2 × 2, stride = 2
PAD	size = (1, 1)
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
PAD	size = (1, 1)
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
POOL	size = 2 × 2, stride = 2
FC	units = 50042, activation = ReLU, dropout = 0.8
FC	units = 50042, activation = ReLU, dropout = 0.8
FC	units = 11, activation = Softmax

One of the reasons why the overfitting may occur, both in the action recognition and object classification model, is noises in the dataset, such as the camera movement. The dataset is collected from mobile phone camera, so while hand-recording the videos, the camera motion can be considered as another movement by the model although the human subject moves differently. Additionally, the number of data instance for the object classification task is also not balanced.

VI. Conclusion

Using the combination of convolutional neural network and long short-term memory network, this paper explores the object and action recognition model from videos. The videos consist of human subject who performs three actions, which are taking an item, returning the item, and only browsing the item shelf. For the object classification, there are 10 items

978-1-7281-2177-2/19/$31.00 ©2019 IEEE

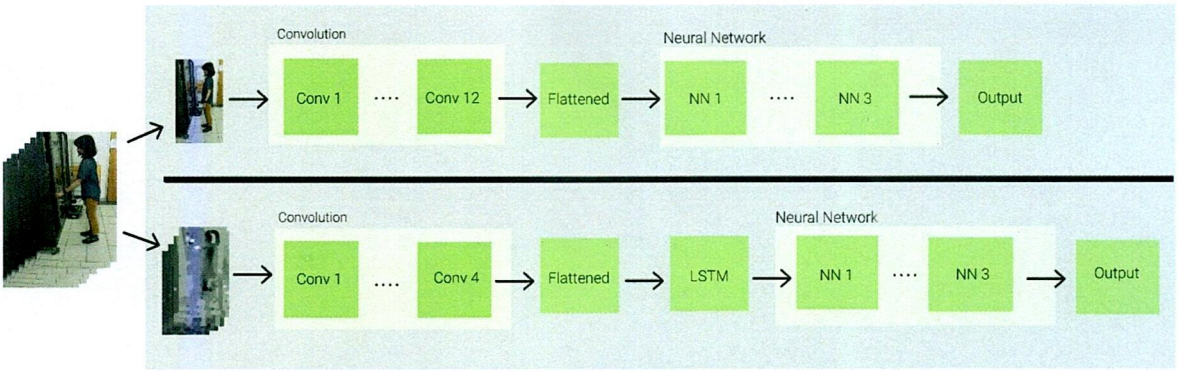


Fig. 2. The deep learning models for the object and human action recognition from video.

TABLE II
THE MODEL ARCHITECTURE FOR ACTION RECOGNITION TASK, WHERE CONV IS THE CONVOLUTION PROCESS, POOL IS THE MAX POOLING, LSTM IS THE LONG SHORT-TERM MEMORY LAYER, AND FC IS THE FULLY-CONNECTED LAYER. IN THE CONVOLUTIONAL LAYER, WE ALSO EMPLOY BATCH NORMALIZATION.

Layer	Parameters
CONV	filter = 48, kernel size = 7 × 7, activation = ReLU
POOL	size = 2 × 2
CONV	filter = 96, kernel size = 5 × 5, activation = ReLU
POOL	size = 2 × 2
CONV	filter = 256, kernel size = 3 × 3, activation = ReLU
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
CONV	filter = 512, kernel size = 3 × 3, activation = ReLU
POOL	size = 2 × 2
LSTM	units = 1000
FC	units = 524, activation = ReLU, dropout = 0.6
FC	units = 522, activation = ReLU, dropout = 0.8
FC	units = 3, activation = Softmax

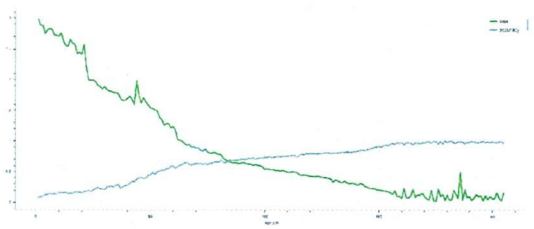


Fig. 3. The training loss and accuracy of the object classification model on each epochs.

available in the shelf. The experiments show that the object and action recognition model achieve 56.41% and 76.92% accuracy, respectively, from the testing data.

Our future concern is to improve the accuracy of the model by adding more data and variation of the movements and

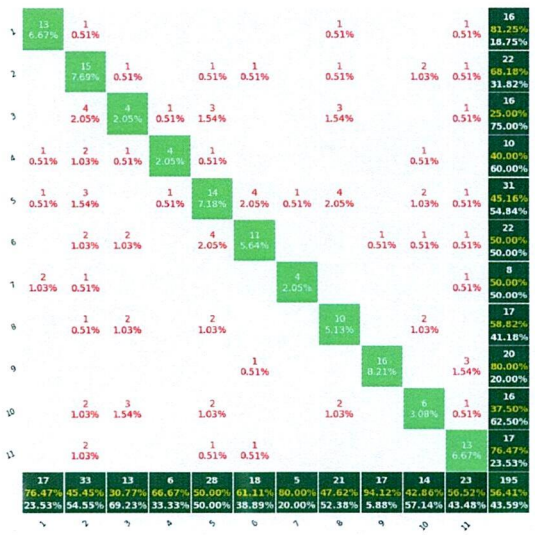


Fig. 4. The confusion matrix of the object classification model on testing data, where the number represents the first, second, third, and so on object.

objects. Hopefully, the model can be extended into an end-to-end deep learning system for an automatic checkout system.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of NVIDIA Corporation by donating Titan X GPU used for this research.

REFERENCES

[1] S. Ji, W. Xu, M. Yang and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. In ICML, June 2010, Vol. 2, No. 5, p.6.
[2] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. "Large-scale video classification with convolutional neural networks." In IEEE conference on Computer Vision and Pattern Recognition, 2014, pp.1725-1732.
[3] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," In Advances in neural information processing systems, 2014, pp.568-576.

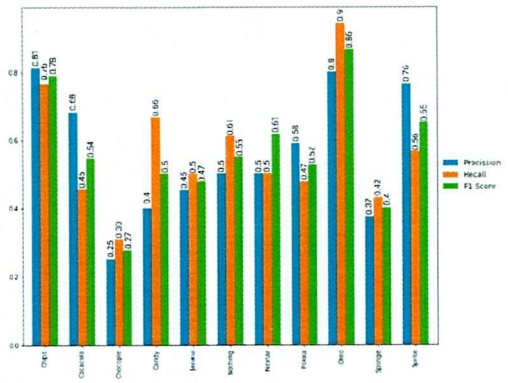


Fig. 5. The precision, recall, and F1-score for each objects from the testing data.

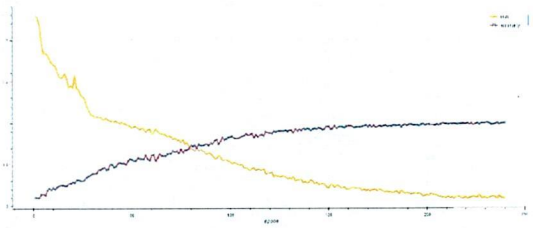


Fig. 6. The training loss and accuracy of the action recognition model on each epochs.

Fig. 7. The confusion matrix of the action recognition model on testing data, where the number represents (1) the checking items movement, (2) taking an item, and (3) returning item to the shelf.

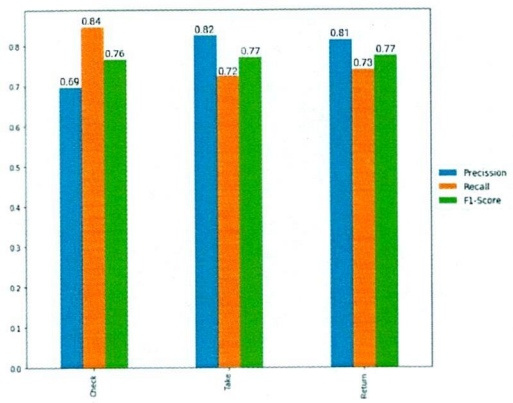


Fig. 8. The precision, recall, and F1-score for each actions or movements from the testing data.

[4] A.Graves, A.R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," In IEEE international conference on acoustics, speech and signal processing, 2013, pp.6645-6649.

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," In IEEE conference on computer vision and pattern recognition, 2015, pp.2625-2634.

[6] H. Yu, J. Wang, Z. Huang, and W. Xu, " Video paragraph captioning using hierarchical recurrent neural networks," In IEEE conference on computer vision and pattern recognition, 2016, pp.4584-4593.

[7] N. Srivastava, E. Mansimov and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," In International conference on machine learning, June 2015, pp.843-852.

[8] A. Gupta and L.S. Davis, "Objects in action: An approach for combining action understanding and object perception," In Conference on Computer Vision and Pattern Recognition June 2007, pp.1-8.

[9] M N Galabov, "A Real Time 2D to 3D Image Conversion Techniques", In International Journal of Engineering Science and Innovative Technology (IJESIT), Vol 4, January 2015.

[10] O.R. Vincent and O. Folorunso, "A descriptive algorithm for sobel image edge detection," In Informing Science & IT Education Conference (InSITE), June 2009, pp.97-107.

[11] T. Handhayani, J. Hendryli and L. Hiryanto, "Comparison of shallow and deep learning models for classification of Lasem batik patterns," In 1st International Conference on Informatics and Computational Sciences (ICICoS), November 2017, pp.11-16.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory", 1997, pp.1735-1780.

[13] Y. Bengio, P. Simard, and P. Frasconi, Learning long-term dependencies with gradient descent is difficult, In IEEE Transactions on, 1994, vol. 5, no. 2, pp.157166.

[14] H. Sak, A. Senior and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." In Fifteenth annual conference of the international speech communication association, 2014.

[15] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig and Y. Shi, "Spoken language understanding using long short-term memory neural networks," In IEEE Spoken Language Technology Workshop (SLT), December 2014, pp.189-194.

[16] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez and P.J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks," In Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[17] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," In IEEE conference on computer vision and pattern recognition, 2015, pp.3128-3137.

978-1-7281-2177-2/19/$31.00 ©2019 IEEE