Capturing temporal information by varying frame strides as inputs in human interaction classification task

Nguyen Quang Khoi (k.q.nguyen@tilburguniversity.edu)
School of Humanities and Digital Sciences, Warandelaan 2, 5037 AB Tilburg
Tilburg, The Netherlands

Abstract

Video frames processing techniques are an essential part of human interaction classification task. Three strategies to handle individual frames have been proposed by Karpathy et al. (2014). However, they have some limitations. Applying Transfer Learning can be a way to alleviate them. I wanted to test whether varying the number of frames as inputs can improve the performance of a Transfer Learning model. In this study, I varied that variable (from two to ten) to the Inception V3 model for the Oxford TV Human Interactions dataset. Accuracy, precision, recall, and F1 score were used to evaluate the model. They were compared with Patron-Perez, Marszałek, Zisserman, and Reid (2010) model and Alexandros (2018) model. The model performed better than the former at every frame stride. The latter at value of five performed worse than my model at stride values 2, 3, 6 on every metric. Overall, the performance decreased drastically as the frame stride was increased. Furthermore, there was some fluctuation in performance between even and odd values. The reason might be due to the temporal domain information loss, data structure bias, or lacking of the negative class. Further works need to be conducted to analyze the time, memory and performance tradeoff. Additionally, deploying cross-validation strategies is also essential to avoid bias in training data.

Keywords: Deep Learning; Convolutional Neural Network; Transfer Learning; Human Interaction Classification; Temporal Information;

Introduction

Recognizing human interaction in videos has many applications in surveillance, retrieval, video games, and human interaction. There have been many attempts in machine learning at solving this task. For example, one study used SVM to recognize human interactions in TV shows (Patron-Perez et al., 2010). Since the success of deep learning in the ImageNet competition in 2012 (Krizhevsky, Sutskever, & Hinton, 2012), deep learning has risen to be one of the most powerful tools in machine learning. Convolutional Neural Network (CNN) is a well-known deep learning architecture to solve computer vision problems. It has been applied on individual frames to classify actions and interactions (Asadi-Aghbolaghi et al., 2017; Bilen, Fernando, Gavves, Vedaldi, & Gould, 2016). In order to capture information in the temporal domain, Karpathy et al. (2014) proposed three fusing strategies. They are Early Fusion, Late Fusion, and Slow Fusion. Nevertheless, they showed limitations when large data is not feasible.

Stergiou and Poppe (2018) proposed using Transfer Learning (Pan & Yang, 2010; Bengio, 2012) to partly mitigate those issues. Transfer learning is a machine learning method where a model trained for one task can be reused for another task. In the case of image data, a model that has already been trained on a larger, more general data set like ImageNet to make classification task on smaller, domain-specific image storage can be utilized. Only deeper layers need to be retrained for specific domains information. It shall lead to fewer parameters that need to be trained on task-specific data. Thus, the chance of overfitting would be reduced. This proposal was tested by applying Transfer Learning on Inception V3 model (weights trained on Imagenet) for the Oxford TV Human Interactions dataset (Stergiou & Poppe, 2018). In that study, a frame value of five was used to extract frames as inputs for the model. I wanted to test whether varying the number of frame values as inputs would help us better exploiting temporal information in videos. This question has two essential implications. Firstly, whether I would lose subtle interaction in each frame by cutting them at great values. Secondly, I wanted to know if I can increase memory and computing efficiency without undermining performance. I hypothesized that the performance would not decrease below 80% on all metrics in the range of 2 - 10.

Methods

Dataset and Code Base

Our experiment were based on two components: the Oxford TV human interaction dataset (Patron-Perez et al., 2010) and Alexandros (2018) model. The dataset consists of 300 video clips collected from over 20 different TV shows, which contain 4 different interactions: handshakes, high fives, hugs and kisses, as well as clips that do not contain any of the interactions. All videos have a duration of eight seconds, and they are 24-25 FPS. Except for the negative class, each interaction class has 50 clips. This dataset was first studied in Patron-Perez et al. (2010). For my experiment, some of the implementation code was adapted from Alexandros (2018) because I did not want to change the model architecture. I did not classify negative cases because I did not have enough computing power to train the whole dataset and to keep the data balance.

Prepossessing

Our prepossessing stage had three steps: extracting videos into images, resizing, and transforming them into arrays.

Firstly, the clips were cut into sequences of images according to a frame stride value (Figure 1, Figure 2). For instance, I took out frames indices 0, 3, 6, 9, ... etc for the number three. Afterwards, they are resized to 300 x 300 pixels. The images were then split into three subsets: training, validation and test. The proportion was 70%, 15%, and 15%.

Figure 1: First 6 video segments used for training at stride number two

Figure 2: First 6 video segments used for training at stride number six

Implementation

Transfer Learning was applied by training latter layers of the Inception V3 model, which was trained on ImageNet. The Inception V3 model is a convolutional neural network. It accepted inputs of size 300 x 300 x 3 (heights * width * channels). As to apply transfer learning, convolutional layers did not need to be trained. The data was fed to deeper layers which consist of an average pooling + non-linear 1024 units NN + softmax logistic component to output results. Adam optimizer and categorical cross-entropy loss function were employed.

The experiment was conducted on local machines and google colab. The inconsistency between them proved to be an obstacle in analyzing time, memory and performance trade-off. Therefore, the trade-off investigation was omitted. Code was written and maintained on Github (Nguyen, 2021).

Performance Evaluation

The model was evaluated using accuracy, precision, recall, and F1 score on the test batches. Accuracy is a common metric to measure how many instances were classified correctly. Precision emphasizes false positive and tells us how many positive classifications are actually correct. Copyright infringement detection software may not want to make too many false accusations against content creators because it may lead to unnecessary legal issues. Recall answers the question of whether we catch all positive instances in data. If human interaction classification is used in surveillance (e.g., detecting suspicious activities), then false negative may be an essential factor. Finally, F1 score is a harmonic mean of precision and recall, it was used as a balanced metric for both of them. This difference gave us another view of the result. I calculated precision, recall, and F1 score for each of four classes. Furthermore, macro averages were also calculated to give overviews of the whole model. It was used since the data was balanced.

Results

The results of the experiment are shown in Figure 1. The first three graphs display the corresponding metrics between interactions and macro average measurements in black. The final graph demonstrates the accuracy of the whole model at each stride. The experiment exhibits downward trends in every metrics. All metrics were higher than 0.95 at frame strides of 2 and 3. Starting from 4, there was some trade-off between precision and recall in-classes. The macro averages and accuracy lines suggest that the whole model tends to plunge at odd frame values, then it recovers at even frame values.

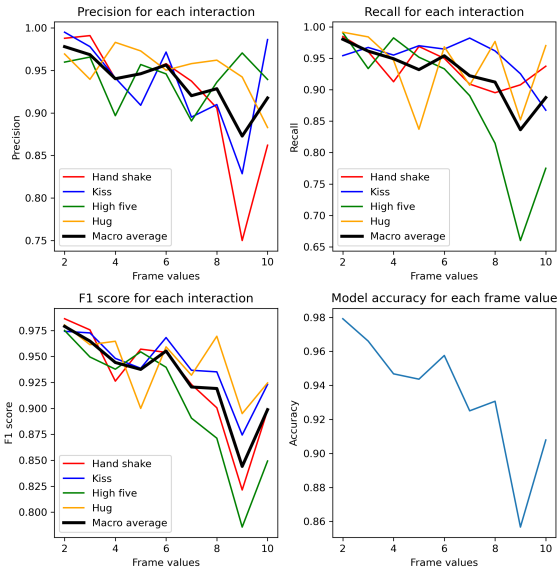

Figure 3: Experiment results, (Alexandros, 2018) model is at value five

Discussion

Our implementation was compared with that of Alexandros (2018) and SVM model proposed by Patron-Perez et al. (2010). Alexandros (2018) model at 5 performed worse than the model at stride values 2, 3, 6 on every metrics. The average precision of Patron-Perez et al. (2010) at the best setting (M + SL) was lower than the lowest average precision of my model. This result may suggest that Convolutional Neural Network and Transfer Learning were better choices than SVM in recognizing human interactions. Additionally, the fact that my model did not train the negative class may help to explain this result.

The experiment demonstrated that the performance of the model decreased drastically when I increased the frame value. Furthermore, there was some spike at even values. There are two possible explanations for those phenomenons: Lost temporal information and issues in the data structure. Firstly,

temporal information might be lost when I increased the stride number. Thus, the model did not get to study the subtle transition between frames of interactions. Secondly, key frames that define actions might be distributed evenly in systematic locations. All videos started with the onset of interactions and ended exactly at 8 seconds. Most of them also had 24 FPS. Frames which contain interaction information may appear at specific locations. This explanation is also very likely to be the cause of spikes at even frame values. Moreover, some interaction was likely to be longer or shorter than 8 seconds in real life.

Table 1: Average precision of the SVM model proposed by Patron-Perez et al. (2010)

Method	AVG
M + ID	0.5032
M + SL	0.6415
M + ID + N	0.4093
M + SL + N	0.5574
A + ID	0.3824
A + SL	0.3933
A + ID + N	0.3151
A + SL + N	0.3276

Conclusion

In this paper, I have experimented to see whether I can better capturing temporal information by varying frame strides as inputs. I varied the value when I cut our videos into individual frames. They were then fed in the Inception V3 model latter layers to apply transfer learning. The results showed a drastic decrease in performance when the frame stride values were raised. Some explanations were given as possible causes. The model proved to has better performance than Patron-Perez et al. (2010) at every frame value. It also performed better than Alexandros (2018) at 2, 3, and 6.

Future works would include trying cross-validation strategies, including the negative class in the training set, and measuring trade-off. Cross-validation is important to let the model expose to different test sets and training sets. It is useful to counteract randomness in splitting data. Including negative classes are also essential to measure how the model will respond before unusual interactions. Both of them are not available now due to a lack of computing power to conduct them. Last but not least, analyzing the trade-off between training time and performance is one of the most important tasks in the future. This experiment was trained both on google Colab free service and local machines. Thus, it was too unreliable and inconsistent to precisely measure the trade-off.

Acknowledgments

I am grateful for GPU and disk space support from my teammate, Jos Prinsen.

References

Alexandros, S. (2018). *Inception v3 - tv human interactions dataset.* Retrieved from https://github.com/alexandrosstergiou/Inception_v3_TV_Human_Interactions

Asadi-Aghbolaghi, M., Clapés, A., Bellantonio, M., Escalante, H. J., Ponce-López, V., Baró, X., ... Escalera, S. (2017). A survey on deep learning based approaches for action and gesture recognition in image sequences. In *2017 12th ieee international conference on automatic face gesture recognition (fg 2017)* (p. 476-483). doi: 10.1109/FG.2017.150

Bengio, Y. (2012, 02 Jul). Deep learning of representations for unsupervised and transfer learning. In I. Guyon, G. Dror, V. Lemaire, G. Taylor, & D. Silver (Eds.), *Proceedings of icml workshop on unsupervised and transfer learning* (Vol. 27, pp. 17–36). Bellevue, Washington, USA: PMLR. Retrieved from http://proceedings.mlr.press/v27/bengio12a.html

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., & Gould, S. (2016). Dynamic image networks for action recognition. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (p. 3034-3042). doi: 10.1109/CVPR.2016.331

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *2014 ieee conference on computer vision and pattern recognition* (p. 1725-1732). doi: 10.1109/CVPR.2014.223

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th international conference on neural information processing systems - volume 1* (p. 1097–1105). Red Hook, NY, USA: Curran Associates Inc.

Nguyen, Q. K. (2021). *Frame strides experiment - inception v3 - tv human interactions dataset.* Retrieved from https://github.com/khoinguyen19k8/Inception_v3_TV_Human_Interactions_CSAI

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345-1359. doi: 10.1109/TKDE.2009.191

Patron-Perez, A., Marszałek, M., Zisserman, A., & Reid, I. D. (2010). High five: Recognising human interactions in TV shows. In *British machine vision conference.*

Stergiou, A., & Poppe, R. (2018). Understanding human-human interactions: a survey. *CoRR*, *abs/1808.00022*. Retrieved from http://arxiv.org/abs/1808.00022