Human Action Classification in the Epic Kitchen Scenario using Deep Learning and RGB data

Edoardo Balducci, Enrico Corradini, Alessandro Mentuccia

***Abstract*—This paper presents an automated video analysis system that recognizes human ADLs activities, related to classical daily actions inside a kitchen. The main goal is to classify human activities starting from a well-known dataset and using a deep learning method together with deep neural architecture to predict the future action.**

I. Introduction

Human actions recognition has been an active research field in computer vision because of its wide range of applications, such as smart surveillance and human-computer interactions. In recent years, we have seen significant progress in this domains due to advances in deep learning and the release of benchmarks such as [1], [2], [3], [4], [5], [6]. In this project, a convolutional neural network is used to classify human actions using EPIC KITCHENS dataset as learning input. The difference of this dataset from others is the length of videos and also the focus on first-person vision, which offers a unique viewpoint on people's daily activities. This dataset reflects people's multi-tasking ability and many different ways to perform a variety of important everyday tasks (such as cleaning the dishes).

Performance of a recognition system mainly depends on whether it is able to extract and utilize relevant information. However, extracting such information is non-trivial due to a number of complexities, such as scale variations, view point changes and camera motions. Recently, Convolutional Networks (ConvNets) have witnessed great success in classifying images of objects, scenes, and complex events [9] - [12]. ConvNets have also been introduced to solve the problem of video-based action recognition [13], [14], [15], [16]. Deep ConvNets come with great modeling capacity and are capable of learning discriminative representation from raw visual data with the help of large-scale supervised datasets. However, ConvNets have two faults: first, long-range temporal structure plays an important role in understanding the dynamics in action videos and second, in practice, training deep ConvNets requires a large volume of training samples to achieve optimal performance (high risk of over-fitting). Thus, the obstacles motivated us to considered the TSN (Temporal Segment Network), a video-level framework developed by Limin Wang et al. [8].

II. State of art

As a state of the art we have used two different tools, both useful for our project:

1) Temporal Segment Network
TSN is a framework for video-based action recognition. It combines a temporal strategy using optical flow frames and a spatial strategy using RGB frames. TSN obtains the state-of-the-art performance on the datasets of HMDB51 (69,4%) and UCF101 (94,2%). As demonstrated on two challenging datasets, TSN has brought the state of the art to a new level, while maintaining a reasonable computational cost. We use this framework in Epic Kitchen scenario.

2) Epic-Kitchen
We tried to reproduce the result obtained by Epic Kitchens maintainers related to spatial training. Table I shows the benchmark. Other results to be considered as state of the art are shown in table II, showing the percentage by which an action is recognized.
Our main goal is to improve these results considering smaller datasets and trying to implement our own ideas in order to obtain better outputs.

TABLE I
Baseline results for the action recognition challenge

	Verb Accuracy	**Verb AvgClass-Precision**	**Verb AvgClass-Precision**
RGB	45.25	54.94	23.31

TABLE II
Sample baseline action recognition per-class metrics

	put	**take**	**wash**	**open**	**cut**
Recall	65.32	51.01	80.45	60.98	74.27
Precision	35.62	41.24	63.17	72.67	69.38

III. DATASET

We used Deep Learning and RGB data to Human Action Classification in the EPIC-KITCHENS scenario [7].

The original dataset, EPIC-KITCHENS, contains data that was collected by 32 participants, in their own kitchens. The participants were asked to capture all their daily kitchen activities. The recordings, including both video and sound, show the natural multi-tasking that one performs. Data was captured using a head-mounted Go-Pro. The decision to collect narrations of the actions made by the subjects is based on the fact that they are the most qualified to label the activity compared to an independent observer, as they were the ones performing the actions. The total dimension of dataset is 1 TB.

We created our dataset starting from original but we chose only one kitchen. This choice is due to the fact that our hardware resources are significantly lower than those of the creators of the original dataset. In the EPIC KITCHEN find 'P♯♯/P♯♯_∗∗' with '♯♯' denoting the participant number and '∗∗' to identify the video number. We use only one folder "frames_rgb_flow/p01" for our project.

The folder p01 contains 816789 frames. Those frames show all the activities carried out by a participant in his kitchen. The total size of the p01 folder is 16.1Gb. As already mentioned, even if we reduced the size of the dataset from 1TB to 60Gb, our work required much longer than expected due to the limited computational availability.

The creators of the Epic Kitchen dataset provide a csv file with details of the main properties for each video of one kitchen. The features are: uid, participant_id, video_id, narration, start_timestamp, stop_timestamp, start_frame, stop_frame, verb, verb_class, noun, noun_class, all_nouns, all_noun_classes. For our work we do not need all these properties but only three. We have implemented a python program, called "create_dataset.py", which takes the file csv, reads the property 'verb_class' and stores the 'start_frame' and the 'stop_frame'. After that, inside our folder P01, the script creates a folder called like 'id_class' for each row selected and then move all the frames starting from 'start_frame' up to 'stop_frame' to the newly created folder.

Finally we get the P01 folder containing other 'id_class' folders, each one with all the frames describing that action. Another function of the program is to merge similar classes together, i.e. we have combined the actions 'cut' and 'peel' creating the 'cut' class such that we are able to obtain a smaller number of classes but a greater number of images that they describe the class. The classes we got from all the frames contained in P01 are 12 and are cut, empty, put, move, cook, open, close, turn, sample, eat, wash, knead.

After this operation, using another python program "analyze_dataset.py", we have filtered the dataset. We only kept those classes that had a sufficient number of examples, more than 100. We kept the classes that had at least 100 frames to represent them. At the end of the analysis we decided to maintain only five classes: cut, put, open, cook, wash.

IV. METHOD

Technical specifications:
-Ubuntu 18.04 64bit
-1 GPU NVIDIA GTX 1080 8Gb
-Driver NVIDIA CUDA 396.26
-NVIDIA CUDA ver 9.2
-NVIDIA cuDNN ver 7.1

Our working method has been organized in several phases:

1) We only took a kitchen to reduce the amount of data and make it possible to train the network. This choice was also made to train the network with less actions to recognize.

2) In the epic kitchen project the classes considered are more than 100. We have chosen to focus our work on actions different from one another to obtain better results, avoiding actions that are similar.

3) We downloaded all examples of actions made in one kitchen, then we began to group those classes whose actions were very similar to each other, in order to reduce their number. At the end of this unification, we discarded all the classes that contained fewer than 100 examples.

4) Before starting the training of the network, we realized that our generated dataset was unbalanced. Instead of assigning weights to classes, we have decided to copy instances of smaller classes until we reached the same number of examples for all of them, i.e. 700.

5) We trained the Temporal Segment Networks with our own dataset, using pre-trained weights obtained from Temporal Segment Networks project.
All videos we need to extract RGB and optical flow frames are encoded at 30fps. Frames and videos have a resolution of 456x256. We trained only the spatial network, even if our own fork of Temporal Segment Networks includes all the tools for optical flow extraction, using the TV-L1 algorithm [17]. We trained each model on 1 GPU NVIDIA GTX 1080 8Gb for 2500 iterations with a minibatch size of 32. We set learning rate to 0.001.

6) The next section presents all the results of our work. To evaluate our outputs we used different metrics for all classes: Precision, Recall e F1-Score.

In order to obtain good results, we have initially reduced our problem to 3 classes and then added one class at a time until we reached 5 classes. The first experiment considered the following classes: Cut, Put, Cook and we used the pre-trained model from TSN as starting weights. Then we added the Open class, using the model obtained the previous experiment as starting weights. We considered our final results the one obtained from five classes model: Cut, Open, Put, Cook and Wash.

V. Experimental Results

As mentioned in the previous paragraph we started with three classes until up to five classes:

Experiment 1) CUT, PUT, COOK
We obtained an accuracy of 97.50%. More detailed results are shown in the table III. As can be seen from the confusion matrix in Figure 1, the class that caused lots of errors is Put.

TABLE III
Accuracy and macro-F1 for the prediction ....

	Precision	Recall	F1-Score
Cut	98,15	100	99,06
Put	100	92,48	96,09
Cook	94,66	100	97,26

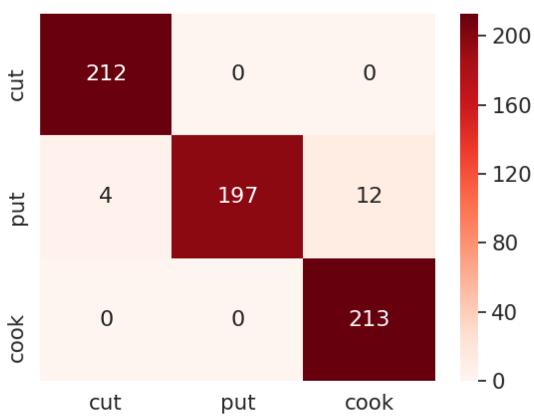

Fig. 1. Confusion matrix

Experiment 2) OPEN, CUT, PUT, COOK
We obtained an accuracy of 88,40%. More detailed results are shown in the table IV. As can be seen from the confusion matrix in Figure 2, the class that caused lots of errors is again Put.

TABLE IV
Accuracy and macro-F1 for the prediction ....

	Precision	Recall	F1-Score
Open	81,22	93,42	86,90
Cut	90,12	99,05	94,38
Put	88,38	64,31	74,45
Cook	94,49	96,71	95,59

Experiment 3) OPEN, CUT, PUT, COOK, WASH
We obtained an accuracy of 85,43%. More detailed results are shown in the table V. As can be seen from the confusion matrix in Figure 3, the class that caused lots of errors is again Put.

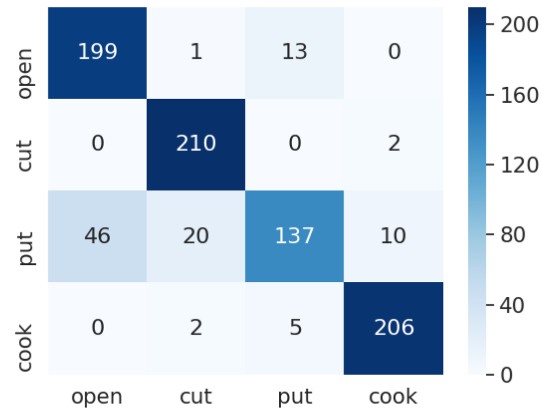

Fig. 2. Confusion matrix

TABLE V
Accuracy and macro-F1 for the prediction ....

	Precision	Recall	F1-Score
Open	82,00	92,01	86,72
Wash	83,62	91,50	87,38
Cut	86,77	99,05	92,51
Put	82,20	45,53	58,61
Cook	90,94	99,06	94,83

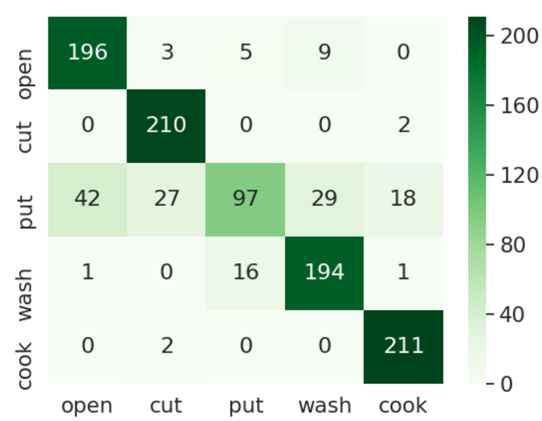

Fig. 3. Confusion matrix

Adding a class each time we have seen how the accuracy changes as seen in Figure 4.

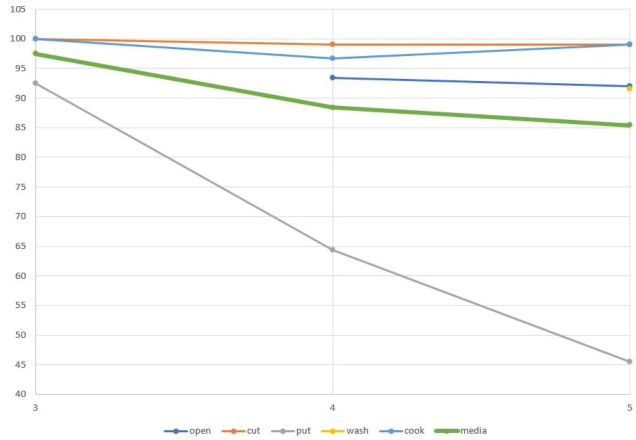

Fig. 4. Accuracy

VI. Conclusions

In conclusion, average accuracy (reported in the section Experimental result) obtained is 85,43715%, which can be considered acceptable. Remembering the average accuracy of 45,25%, obtained in EPIC-Kitchens work, using all 125 classes for the training phase obtained from all the 32 kitchens, we achieved a lot better results, but we only worked on 5 balanced classes with examples from only one kitchen. Thus, thinking about future developments, we could compute accuracy with optical flow training along RGB. Moreover, we could use all the remaining classes, after a balancing process as explained before, to train the neural network and obtain a better model for human action recognition in a kitchen. Another future work could be the generalization of the classifier so that it can recognize action of other kitchens and people. The most interesting development of this project could be a classifier which recognizes actions with objects used by the user. This could be achieved by mixing the already explained approach with an object recognition classifier, as done in epic kitchens project.

References

[1] R.Goyal, S.E.Kahou, V.Michalski, J.Materzynska, S.Westphal, H.Kim, V.Haenel, I.Frund, P.Yianilos, M.Mueller-Freitag, F.Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The "something something" video database for learning and evaluating visual common sense," in ICCV, 2017.

[2] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A Large-Scale Video Classification Benchmark," in CoRR, 2016.

[3] H. Zhao, Z. Yan, H. Wang, L. Torresani, and A. Torralba, "SLAC: A Sparsely Labeled Dataset for Action Classification and Localization," arXiv preprint arXiv:1712.09374, 2017.

[4] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A Dataset for Movie Description," in CVPR, 2015.

[5] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "MovieQA: Understanding stories in movies through question-answering," in CVPR, 2016.

[6] D. F. Fouhey, W.-c. Kuo, A. A. Efros, and J. Malik, "From lifestyle vlogs to everyday interactions," arXiv preprint arXiv:1712.02310, 2017.

[7] Scaling Egocentric Vision: The EPIC-KITCHENS Dataset, Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, Michael Wray.

[8] Temporal Segment Networks: Towards Good Practices for Deep Action Recognition Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, Luc Van Gool. https://arxiv.org/abs/1608.00859

[9] Krizhevsky, A., Sutskever, I., Hinton, G.E.:ImageNet classification with deep convolutional neural networks. In: NIPS. (2012) 1106–1114

[10] Simonyan, K., Zisserman, A.:Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015) 1–14

[11] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2015) 1–9

[12] 11. Xiong, Y., Zhu, K., Lin, D., Tang, X.: Recognize complex events from static images by fusing deep channels. In: CVPR. (2015) 1600–1609

[13] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large- scale video classification with convolutional neural networks. In: CVPR. (2014) 1725–1732

[14] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recog- nition in videos. In: NIPS. (2014) 568–576

[15] Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotem- poral features with 3d convolutional networks. In: ICCV. (2015) 4489–4497

[16] Zhang, B., Wang, L., Wang, Z., Qiao, Y., Wang, H.: Real-time action recognition with enhanced motion vector CNNs. In: CVPR. (2016) 2718–2726

[17] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in Pattern Recognition, 2007.