Angel Ayala

Cleber Zanchettin - David Macedo

Deep Learning

29 de setembro de 2019

Advances in project

In general terms que article base has been implemented and tested with their respective dataset. Achieving around 93% of accuracy during training which is the value proposed in their paper. This neural network has been made in Keras with Tensorflow as backend without any troubles.

During testing over the neural network I been notice that this model present some issues for fire recognition with a image in black. When a image in black is presented to the model, present a label of fire with 98% of accuracy,  such as image below.

Reviewing the dataset, the presence of some images with a night scenario and showing smoke only, caused by a forest fire, I think its the problem, see the next image. This is only one of the four datasets selected for this work.

The dataset which has been previously used for this kind of task are:

- FiSmo: Available at https://drive.google.com/drive/u/1/folders/19oH22E7dMyJHUifaFSyn6F8pxS032NBf with no use found in previous works.
- CairDataset: Available at https://github.com/cair/Fire-Detection-Image-Dataset
  Used in J. Sharma, O.-C. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in Engineering Applications of Neural Networks, G. Boracchi, L. Iliadis, C. Jayne, and A. Likas, Eds. Cham: Springer International Publishing,

2017, pp. 183–193

https://link.springer.com/chapter/10.1007/978-3-319-65172-9_16

- FireSense: Available at https://zenodo.org/record/836749

  Used in K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, no. 2, pp. 339–351, 2 2015.

  https://ieeexplore.ieee.org/document/6857396

- FireNet, used as base article: Available at

  https://github.com/arpit-jadon/FireNet-LightWeight-Network-for-Fire-Detection

  Used in A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "Firenet: A specialized lightweight fire & smoke detection model for real-time iot applications," CoRR, vol. abs/1905.11922, 2019.

  Available: http://arxiv.org/abs/1905.11922

I would like to work the neural network on PyTorch, I already have the architecture in this backend but only gets an accuracy of 84% at top. I already check the initialization values, the images used for training purpose, which are the same, and the default values used for each hyper-parameter such as the training process itself. The unique issue which differ from one to another what I found until now is the loss function. For Keras model, the loss function is names "*sparse_categorical_crossentropy*". In PyTorch is only "*CrossEntropy*". I think this difference can not be solved due to difference of the backend itself.

I would like to improve the results working in PyTorch changing some layers among others things. I think this problems of the backend would not affect due I can compare with the same base code used in their work.