ISSN: 2688-8645

Research on the Method of Keras Library Combined with CNN Convolution to Classify Video

Ziqiao Qiu*

Guangdong University of Science & Technology, Dongguan Guangdong 523000, China

Abstract

With the popularization of the mobile Internet and the impact of the new crown epidemic, the time spent by netizens on the Internet is gradually increasing. Every minute, every major video website will add nearly 100 hours of video. How to help users quickly find their favorite content and achieve effective video push is the future development focus of deep learning technology. This paper mainly introduces the research direction of efficient video classification using Keras library combined with CNN.

Keywords

Keras Library; CNN Convolution; Video Classification.

1. Introduction

According to a statistical report released by the China Internet Network Information Center, Chinese netizens spend 30.8 hours per week online, with short video apps accounting for 11.0% of the time, ranking third in terms of short video usage time. With the impact of the normalcy of the new crown epidemic, the length of time netizens watch videos will inevitably increase further. Faced with rich video content, it is difficult for users to filter effectively, so they will be more critical in the choice of video. For Internet companies, how to quickly find content that users like more, so as to provide different users with featured video pushes, is a prerequisite for ensuring user stickiness. In this process, the efficient classification of videos is the basis of video push. These video classification algorithms can automatically analyze the semantic information contained in the video, understand its content, and perform motion capture, automatic labeling, classification and description of the video, in order to achieve the high efficiency and accuracy of human eye recognition. Therefore, large-scale, efficient and fast video classification algorithms will be the key research direction of deep learning technology in the future.

2. Introduction to the Keras Library

Keras [1] is considered to be one of the most characteristic machine learning libraries in python. It provides a mechanism for easier expression of neural networks. Keras also provides some of the best utilities for compiling models, processing datasets, graph visualization, etc. On the backend, Keras uses Theano or TensorFlow internally. Some of the most popular neural networks such as CNTK can also be used. When we compare it to other machine learning libraries, Keras is relatively slow because it uses the backend infrastructure to create a computational graph and then leverage it to perform operations. All models in Keras are lightweight.

Features of Keras It runs smoothly on both CPU and GPU. Keras supports almost all neural network models - fully connected, convolutional, pooling, recurrent, embedding, etc. Additionally, these models can be combined to build more complex models. Modular in nature, Keras is incredibly expressive, flexible and capable of innovative research. Keras is a completely python-based framework that makes debugging and exploration easy. Keras contains

ISSN: 2688-8645

implementations of many commonly used neural network building blocks, such as layers, objectives, activation functions, optimizers, and a range of tools to make working with image and text data easier. Also, it provides many pre-processed datasets and pre-trained models like MNIST, VGG, Inception, SqueezeNet, ResNet

3. Exploration of Main Research Methods

Compared with image recognition, videos can provide more information than static images in video classification tasks, including complex motion information that evolves over time. Videos (even short ones) contain hundreds or thousands of frames, but not all of them are useful, and processing these frames requires a lot of computation. The easiest way to do this is to treat these video frames as static images, apply a CNN to identify each frame, and then average the predictions to get the final result for that video. However, this method uses incomplete video information, thus making the classifier potentially prone to confusion.

Set the behavior representation method. The behavioral representation [2, 3]includes a series of operations such as feature encoding, pooling, and normalization, and finally forms a normalized feature vector describing the video. Feature encoding is to quantize the features in the continuous feature space to obtain the feature encoding vector. For example, in the bag-of-words model, each feature is quantified to a term in the dictionary. In order to reduce the quantization error, many quantizations of each feature to more than one term have been proposed, such as quantization to all terms constructed based on kernel functions, feature coding methods based on sparse coding, linear feature coding methods based on position constraints, and Feature encoding method based on Fisher kernel, etc. Through the evaluation of various feature codes on human action recognition datasets, the experimental results show that the sparse coding method and the Fisher kernel method achieve the highest level of recognition accuracy on different datasets respectively.

A comparative study of pooling methods. Pooling is to calculate the feature vector of the video according to all the feature codes extracted from the video, that is, the representation of human behavior. There are two commonly used pooling methods: Sum Pooling and Max Pooling. Sum pooling is equivalent to accumulating all feature codes, while max pooling is equivalent to the most statistically significant feature codes. In practical use, the pooling method used depends on the chosen feature encoding method. When the sparse coding method is used, the maximum pooling method is generally used to statistically quantify the most significant value of each term to construct a feature vector.

4. Video Classification with Keras and Deep Learning

Image dataset benchmarks have played a very important role in promoting image classification problem solving. From the earliest small-scale annotated datasets Caltech101/256, MSRC, PASCAL, when larger datasets such as ImageNet and SUN were released, the research on visual algorithms for image understanding progressed rapidly. Especially ImageNet and its Large-Scale Visual Recognition Challenge (ImageNet Large Scale Visual Recognition Challenge, ILSVRC) has greatly promoted the development of deep feature learning technology, and network architectures such as AlexNet, VGGNet, Inception, ResNet have emerged one after another, and finally the recognition error rate is lower than the human eye, indicating that CNN has basically solved ImageNet Image classification problem on datasets.

Every data science task requires data. Specifically, clean and understandable data entered into the system. When it comes to images, computers need to see what the human eye sees. For example, humans have the ability to recognize and classify objects. Likewise, we can use computer vision to interpret the visual data it receives. That's what image annotation does. Image annotation plays a vital role in computer vision. The goal of image annotation is to

ISSN: 2688-8645

provide task-related, task-specific labels. This may include text-based labels, labels (borders) drawn on images, or even pixel-level labels. We'll explore this range of different annotation techniques below. AI requires more human intervention than we think. To prepare high-accuracy training data, we have to annotate the images to get correct results.

Figure 1. Video Annotation Results

A video can be understood as a series of individual images; therefore, many deep learning practitioners are quick to view video classification as performing a total of N image classifications, where N is the total number of frames in the video. But there are problems with this approach. Video classification is more than simple image classification - for videos, we can often assume that subsequent frames in the video are related to their semantic content. If we can exploit the temporal properties of videos, we can improve our actual video classification results.

Neural network architectures such as Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN) are suitable for time series data - we will cover both topics in later tutorials - but in some cases they may be overkill. As you can imagine, they also require a lot of resources and time when training thousands of video files. Instead, for some applications, all you might need is a rolling average of the forecasts.

When performing image classification, we: Feed the image to our CNN Get predictions from the CNN Pick the label with the highest probability Since a video is just a sequence of frames, a simple video classification approach is: loop through all the frames in the video file for each One frame, pass the frame through the CNN to classify each frame individually and independently select the label with the highest probability to label the frame and write the output frame to disk But there is a problem with this approach - if you ever try to apply simple image classification to For video classification, you may encounter a kind of "predictive flicker" that we would like our entire video to be labeled like this - but how can we prevent the CNN from "flickering" between these two labels? A simple and elegant solution is to utilize a rolling forecast average. Our algorithm now becomes: Loop through all the frames in the video file For each frame, pass the frame through the CNN Get the predictions from the CNN Maintain a list of the last K predictions Calculate the average of the last K predictions, select the label with the highest probability Mark the frame and write the output frame to disk.

ISSN: 2688-8645

5. Summary

The current mainstream methods of video classification are mainly deep learning methods. These methods are mainly derived from popular deep models in the image and speech domains. The complex nature of video data, including large amounts of spatial, temporal, and audio information, makes existing deep models inadequate for video-related tasks. This makes a strong need for new models to efficiently acquire the spatial and audio information of videos, and most importantly, to model the dynamic process of the space. In addition, training CNN/LSTM models requires a large amount of labeled data, which is usually expensive and time-consuming to obtain. Therefore, it is very important to make full use of unlabeled data and rich contextual information to build better video description models. Promising research directions.

Acknowledgments

This work was supported by Natural Science Project of Guangdong University of Science and Technology (GKY-2021KYYBK-24 & GKY-2021KYQNK-6).

References

[1] Wang Hengtao. Image recognition integrated system based on TensorFlow, Keras and OpenCV [J]. Electronic Testing,.2020.24.019.

[2] Yanxuan Lu, Qing Gao, Jinhu Lu, Maciej , Jin Zheng. A Quantum Convolutional Neural Network for Image Classification [C]//. Proceedings of the 40th China Control Conference (11). [Publisher unknown] 2021.027534.

[3] Bendong Zhao,Huanzhang Lu,Shangfeng Chen,Junliang Liu,Dongya Wu.Convolutional neural networks for time series classification[J].Journal of Systems Engineering and Electronics, 2017, 28 (01): 162-169.