Mo-BoNet: A TIME SERIES CLASSIFICATION MODEL BASED ON COMPUTER VISION

Li Mingcheng¹*, Dong Yubo¹, Wang Hongli², Li Pengchao³

¹ School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China
² School of Foreign Studies, Xi'an Jiaotong University, Xi'an 710049, China
³ School of Mechanical and Electrical Engineering, Xi'an Polytechnic University, Xi'an 710600, China

*Corresponding author’s e-mail: lmc.xjtu@stu.xjtu.edu.cn

Abstract. Time series are widely distributed in many fields. Classical statistical methods are difficult to model the deep meaning of time series, and the deep learning methods based on recurrent neural network has great limitations when it is applied to indefinite long time series. In order to solve the above problems, a time series classification model based on computer vision is proposed, which transforms the time series classification problem into image classification problem. Firstly, three kinds of images with different linewidth corresponding to the time series are used as input to reduce the information loss in the conversion process. Secondly, the transfer learning model based on MobileNetV3-Large is used to encode the image data, and XGBoost is used for classification. The experimental results show that the classification effect of this model is better than that of the classical image classification model, and its XGBoost is also better than other ensemble methods, which proves the feasibility of computer vision method in time series classification task.

1. Introduction

The wave of the Fourth Industrial Revolution brought the world into the era of "information explosion", this is the era of data, everything can be abstracted as data. In an infinite data set, time series data is an extremely important type of existence. It has its presence in the fields of industry[1], agriculture[2], marine industry[3] and text analysis[4][5]. It can be seen that the research on time series classification methods has dual meanings in theory and practice.

Predecessors' research on time series classification methods emerges endlessly, and gradually evolved into two mainstream routes: (1) Time series classification methods based on statistics. (2) Time series classification method based on machine learning. At present, relevant research mainly combines the above two routes. For example, J Wang et al. [6] proposed a time series classification method based on auto regressive integrated moving average model (ARIMA) features and AdaBoost classification. Y Lei et al. [7] proposed a fully convolutional time series classification method based on statistical features. The salient feature of this method is the use of statistical features in data preprocessing and fine-tuning strategies in network training. Since the existing methods mainly rely on the first-order and second-order characteristics of the time series, Parker et al. [8] introduced the method of combining higher-order spectral analysis and convolutional neural networks to classify time series. At the same time, the nested use of deep learning models is also a hot research direction. For example, Kamara et al. [9] used Contextual Long Short-Term Memory (CLSTM) and Contextual Convolutional Neural Networks...
(CCNN) to extract features in an unsupervised manner. Then connecting the obtained data, input the attention the force module, through the Multilayer Perceptron (MLP) block, finally realizes the classification. Chambers et al. [10] proposed FilterNet, which uses the currently popular convolutional neural network (CNN) and long-short-term memory (LSTM) to perform activity recognition through multi-channel sensor data. Elsayed et al. [11] used gated recursive unit (GRU) instead of LSTM to create a GRU-fully convolutional network hybrid model (GRU-FCN), which can provide better performance on many time series data sets without further modification of the model. Although the above mixed models show good classification results, they still have the following problems: (1) The statistical-based modeling process is relatively cumbersome, and the stationarity of the observation sequence is relatively high, and it is difficult to accurately mine abnormal points with abrupt changing data. (2) Traditional machine learning methods have strong sensitivity to feature engineering, and the feature design process is time-consuming and laborious, which greatly depends on the designer’s experience and lacks objectivity. (3) The hybrid deep learning model is still It is difficult to parallelize the calculation, so the calculation time is longer, and it cannot be applied in the system with high real-time requirements. In response to the above problems, W Chen et al. [12] studied a time series data representation method called Relative Position Matrix (RPM), which converts the original time series data into two-dimensional images, so that computer vision technology can classify tasks in time series. Jastrzebska [13] proposed a lag time series representation based on image storage and a convolutional neural network for image classification.

In order to reduce the complexity of the time series classification model so that it can be deployed on devices with limited memory resources, while achieving higher accuracy, this paper is based on computer vision methods, using transfer learning and ensemble learning ideas to lightweight models MobileNetV3-Large [14] was optimized and proposed Mo-BoNet, which not only reduced the amount of parameters and calculations, but also achieved the best classification effect on the experimental data set. Firstly, Mo-BoNet uses three time series images with different line widths as input. After MobileNetV3-Large, a Fully Connected Neural Network (FCNN) is added to form a three-channel time series feature extractor, and then XGBoost[15] integrates the three line width features obtained by the extractor to obtain the classification result.

### 2. Time series classification method based on computer vision

Time series is also called dynamic sequence or time sequence, in which each data unit can be abstracted as a binary \((t, o)\), where \(t\) is the time variable and \(o\) is the data variable. Suppose that time series \(T = \{(t_1, o_1), (t_2, o_2), \ldots, (t_n, o_n)\}\), category set \(C = \{c_1, c_2, \ldots, c_k\}\) and data variable set \(O = \{o_1, o_2, \ldots, o_n\}\) where \(t_i < t_{i+1} (i = 1, 2, \ldots, n - 1)\), \(n\) is the number of time variables in time series and \(k\) is the total number of time series categories. The goal of traditional time series classification methods is to find the mapping \(\varphi: O \rightarrow C\), for any class \(o_i \in O\), making \(c_i = \varphi(o_i)\) equals the real class. Assuming that the data variable set \(O\) is mapped \(g: O \rightarrow O'\) to \(O' = \{o'_1, o'_2, \ldots, o'_n\}\) and mapping \(f: O' \rightarrow C\) is the image classifier where \(O'\) is iconographic set of \(O\). Then the computer vision method is used for time series classification that equals the mapping \(g\) in the traditional classification method is regarded as the composite mapping formed by the time series image mapping \(g\) and the image classifier mapping \(f\).

### 3. Mo-BoNet

As shown in Figure 1, the Mo-BoNet structure is composed of four modules: Input Layer, Encoding Layer, Ensemble Layer and Output Layer.
Figure 1. Mo-BoNet consists of four modules: Input Layer, Encoding Layer, Ensemble Layer and Output Layer.

1) Input Layer: Assuming that there are N time series in the binary data set, each time series is drawn into a pixel RGB image with dimension 224×224×3, which according to the line width of 1.0, 0.5 and 0.1, and then the RGB images with the same line width are read and spliced into an image vector matrix with dimension N×3×224×224×3.

2) Encoding Layer: The image vector matrix is divided into three parts of equal size by row, which respectively represent the image vector matrix with line width of 1.0, 0.5 and 0.1, and the dimension of each part is N×224×224×3. Three image vector matrixes representing different linewidth are fed into the transfer learning encoder composed of MobileNetV3-Large and FCNN respectively, and the Zero Padding-2D, Depthwise Separable Convolution and Batch Normalization are used repeatedly. Then, the extracted features are encoded into three N×1 column vectors by Global Average Pooling and FCNN with dropout.

3) Ensemble Layer: The three column vectors obtained in Encoding Layer are spliced into a encoding matrix with dimension N×3. For the images corresponding to each series of time series, the vector of each row in the matrix contains three kinds of line width information: 1.0, 0.5 and 0.1. Finally, XGBoost is used to classify the vectors of each row of the encoding matrix.

4) Output Layer: On the binary data set, the output of XGBoost is a column vector with the size of N×1 and the element value of 0 or 1, 0 is a negative example and 1 is a positive example.

3.1. Transfer Learning Encoder

3.1.1. MobileNetV3-Large
In the model structure, MobileNetV3 continues the techniques of Depthwise Separable Convolution of MobileNetV1[14] and Bottleneck with Residual of MobileNetV2[17], and adds Squeeze-and-Excitation Block (SE Block)[18] in SENet. According to different resource consumption and accuracy requirements, MobileNetV3 is divided into MobileNetV3-Large and MobileNetV3-Small[14]. In the experiment, MobileNetV3-Large is used.

3.1.2. Transfer Learning
By adding FCNN to the MobileNetV3-Large for transfer learning, the knowledge learned by the former can be shared with the latter, so as to speed up the learning efficiency of the model and improve the generalization ability of the model. In addition, the images of time series are fluctuation lines with low complexity. Overusing the trained deep network is often counterproductive, so only the parameters of the first Block of MobileNetV3-Large large and its previous layers are shared, and the other layers are trained together with FCNN. The FCNN structure used in the experiment is shown in Figure 2.
3.2. XGBoost
There are two kinds of ensemble learning: Bagging and Boosting. XGBoost is an excellent representative of Bagging. The cost function to be optimized by XGBoost is computed as:

\[
Cost(t) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t(x_i))
\]

(1)

where the differentiable convex function \(l(\cdot)\) is used to describe the difference between \(y_i\) and \(\hat{y}_i\), \(f_t(x_i)\) is the prediction result of the \(t\)-th decision tree for sample \(x_i\), and \(\Omega\) is the regular term. Assuming that \(\hat{y}_i^{(t)}\) is the predicted value in the \(t\)-th iteration, the cost function can be rewritten as an iterative function in the form of addition:

\[
Cost(t) = \sum_{i=1}^{n} \left( y_i, \hat{y}_i^{(t-1)} + f_t(x_i) \right) + \Omega(f_t(x_i)).
\]

(2)

By using the second-order Taylor expansion, formula (2) is transformed into an approximate function:

\[
Cost(t) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)}) + \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} f_t(x_i) + \frac{1}{2} \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2} f_t^2(x_i) + (\gamma T_n + \frac{1}{2} \tau \sum_{j=1}^{n} \omega_j^2 + C)
\]

(3)

where both \(\gamma\) and \(\tau\) are the regularization coefficients, \(C\) is constant, \(T_n\) is the number of leaf nodes in a tree, \(\omega_j\) is the fraction of leaf node \(j\). here we note

\[
P_j = \sum_{i \in I_j} \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}
\]

(4)

\[
Q_j = \sum_{i \in I_j} \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}
\]

(5)

where \(I_j\) is the sample set which is divided into the leaf node, the maximum value of the cost function can be computed as:

\[
Cost_{\text{max}}(t) = -\frac{1}{2} \sum_{j=1}^{n} \frac{p_j^2}{Q_j + \tau} + \gamma T_n
\]

(6)

4. Experiments

4.1. Experimental Data
The 23237 time series in the data set used in the experiment (as shown in Table 1) are from the user traffic data of Institute for Interdisciplinary Information Core Technology from 2018 to 2020.

| training set | test set |
|--------------|----------|
| positive     | negative |
| 10543        | 3357     |
| positive     | negative |
| 6822         | 2415     |
4.2. Experimental Design
In this paper, deep learning models contrast experiment, ensemble methods contrast experiment are carried out.

Experiment 1. Comparative Experiment of Deep Learning Models.
In order to verify the effect of Mo-BoNet on time series classification task, comparative experiments were carried out between Mo-BoNet and LeNet, InceptionV3, ResNet50 and MobileNetV3-Large.

Experiment 2. Comparative Experiment of Ensemble Methods.
Compared with MobileNetV3-Large, other ensemble methods used in this paper are Majority Voting, Logistic Regression, Support Vector Machine (SVM) and Random Forest.

4.3. Experimental Software and Hardware Environment
The operating system used in this paper is Ubuntu 16.04, the GPU is GTX 1080, the programming language is Python 3.7, and the deep learning framework is tensorflow2.0.

4.4. Experimental Results and Analysis
The experimental results of Experiment 1 are shown in Table 2, in which the image line width of input data of LeNet, InceptionV3, ResNet50 and MobileNetV3-Large(1.0) are all 1.0 pounds, and the image line width of input data of MobileNetV3-Large(0.5) and MobileNetV3-Large(0.1) are 0.5 pounds and 0.1 pounds respectively. The results show that compared with other classical deep learning models and MobileNetV3-Large with different linewidth input data, the Mo-BoNet model proposed in this paper achieves the best results in accuracy, recall and F1-score. Compared with MobileNetV3-Large with three different line width inputs of 1.0, 0.5 and 0.1 pounds, Mo-BoNet also shows better classification ability, it shows that the integration of three different linewidth inputs is helpful to improve the performance of the model.

| Deep Learning Models | Accuracy | Recall | F1-Score |
|----------------------|----------|--------|----------|
| LeNet                | 93.68    | 91.33  | 92.49    |
| InceptionV3          | 94.57    | 94.08  | 94.32    |
| ResNet50             | 94.78    | 95.67  | 95.22    |
| MobileNetV3-Large(1.0) | 94.87  | 96.32  | 95.59    |
| MobileNetV3-Large(0.5) | 94.58  | 95.96  | 95.27    |
| MobileNetV3-Large(0.1) | 93.60  | 94.29  | 93.94    |
| Mo-BoNet             | 97.01    | 96.88  | 96.94    |

Table 3 shows that the Mo-BoNet proposed in this paper uses XGBoost to ensemble the high-order features obtained from the coding layer, and the effect is better than Majority Voting, Logical Regression, SVM and Random Forest. Among them, the F1-score is improved by 2.75%, 0.40%, 0.46% and 0.23% respectively. It can be seen that XGBoost is more suitable for the high-order features obtained by the Ensemble Layer.

| Ensemble Methods | Accuracy | Recall | F1-Score |
|------------------|----------|--------|----------|
| Majority Voting  | 94.85    | 93.53  | 94.19    |
| Logistic Regression | 96.97 | 96.12  | 96.54    |
| SVM              | 96.90    | 96.07  | 96.48    |
| Random Forest    | 96.86    | 96.56  | 96.71    |
| XGBoost           | 97.01    | 96.88  | 96.94    |
5. Conclusion
In this paper, we propose a time series classification model based on computer vision (Mo-BoNet). MobileNetV3-Large and transfer learning method are used to encode the three channel input, and XGBoost is used as the ensemble method to classify the encoding results, it shows the feasibility of transforming column classification task into image classification task. The next step is to study the application of this model on small data sets and extremely unbalanced data sets.

Acknowledgments
This work was supported by the [National Social Sciences Fund (China)] under Grant [number 17XXW009].

Reference
[1] W Yu, I Y Kim, C Mechefske. (2021) Analysis of Different RNN Autoencoder Variants for Time Series Classification and Machine Prognostics[J]. Mechanical Systems and Signal Processing, 149.
[2] M. R. da Silva, O. A. de Carvalho, R F Guimarães, et al. (2020) Wheat Planted Area Detection From the MODIS NDVI Time Series Classification Using the Nearest Neighbour Method Calculated by the Euclidean Distance and Cosine Similarity Measure[J]. Geocarto International, 35(13): 1400-1414.
[3] K. Gundersen, G. Alendal, A. Oleynik, et al. (2020) Binary Time Series Classification with Bayesian Convolutional Neural Networks When Monitoring for Marine Gas Discharges[J]. Algorithms, 13(6), 145.
[4] Z Liu, H Kan, T Zhang, et al. (2020) DUKMSVM: A Framework of Deep Uniform Kernel Mapping Support Vector Machine for Short Text Classification[J]. Applied Sciences, 10(7), 2348.
[5] J Xu, Y Cai, X Wu, et al. (2020) Incorporating Context-relevant Concepts into Convolutional Neural Networks for Short Text Classification[J]. Neurocomputing, 386:42-53.
[6] J Wang, S Tang. (2020) Time Series Classification Based on Arima and Adaboost[J]. MATEC Web of Conferences, 309, 03024.
[7] Y Lei, & Z Wu. (2020). Time Series Classification Based on Statistical Features. EURASIP Journal on Wireless Communications and Networking, 2020(1).
[8] P. A. Parker, S. H. Holan, N. Ravishanker. (2020) Nonlinear Time Series Classification Using Bispectrum - based Deep Convolutional Neural Networks[J]. Applied Stochastic Models in Business and Industry, 36(5).
[9] A. F. Kamara, E. Chen, Q Liu, et al. (2020) Combining Contextual Neural Networks for Time Series Classification[J]. Neurocomputing, 384:57-66.
[10] Chambers, R. D., & Yoder, N. C. (2020). FilterNet: A many-to-many deep learning architecture for time series classification. Sensors, 20(9), 2498.
[11] N. Elsayed, A. S. Maida, M. Bayoumi. (2019) Deep Gated Recurrent and Convolutional Network Hybrid Model for Univariate Time Series Classification[J]. International Journal of Advanced Computer Science and Applications (IJACSA), 10(5).
[12] W Chen, K Shi. (2019) A Deep Learning Framework for Time Series Classification Using Relative Position Matrix and Convolutional Neural Network[J]. Neurocomputing, 359:384-394.
[13] A. Jastrzebska. (2020) Lagged Encoding for Image - based Time Series Classification Using Convolutional Neural Networks[J]. Statistical Analysis and Data Mining: The ASA Data Science Journal, 13(3):245-260.
[14] Howard, A., Sandler, M., Chu, G., et al. (2019). Searching for MobilenetV3. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1314-1324.
[15] T Chen, & C Guestrin. (2016). XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining.
[16] Howard A. G., M Zhu, B Chen, et al. (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://arxiv.org/abs/1704.04861.

[17] M. Sandler, A. Howard, M Zhu. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. https://arxiv.org/pdf/1801.04381v3.pdf.

[18] J Hu, L Shen, G Sun, et al. (2017) Squeeze-and-Excitation Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42:2011-2023.