

# A deep learning framework for time series classification using Relative Position Matrix and Convolutional Neural Network

论文链接: <https://www.sciencedirect.com/science/article/pii/S0925231219308598>

随着CNN的快速发展,基于CNN的TSC最近也不断浮现。但是这些算法的表现相对于最新的传统TSC算法相比稍显逊色。因此,论文提出了一个新颖的使用相对位置矩阵和卷积神经网络(Relative Position Matrix and Convolutional Neural Network, 简写RPMCNN)的深度学习框架。**论文研究了一种称为相对位置矩阵(RPM)的时间序列数据表示方法,将原始时间序列数据转换为2D图像,从而可以使用图像识别去处理时序数据。**论文还构建了一种改进的CNN架构,以自动学习低级原始时间序列数据的高级抽象表示。因此,将RPM和CNN组合在一个框架中有望提高TSC的准确性和泛化能力。论文对大量标准数据集使用各种现有方法进行了综合评估,并证明了我们的方法取得了显著成果,并且在很大程度上领先于当前的最佳TSC方法。

## Literature Review

近年来,已经开发了大量用于时间序列数据的新的TSC算法。在这些方法中, Bag of SFA Symbols (BOSS)和Collective of Transformation Ensembles (COTE)在UCR数据集上表现出比其他方法更好的性能。但是,通过这些传统方法来提高准确性变得越来越困难。最近,由于功能更强大的计算机的快速发展,深度学习(DL)成为一种通用的,实用的,有效的生成原始数据区分特征的方法。因此,它可能为提高TSC方法的准确性提供线索。作为最成功的DL模型之一,卷积神经网络(CNN)已应用于解决许多领域中的复杂问题。CNN的优点在于,它减少了权重数,从而更易于优化网络,并降低了基于共享权重架构和转换不变性特征而过度拟合以获得更好的泛化能力的风险。因此,CNN在许多任务(例如图像和视频识别,推荐系统和自然语言处理)中均取得了显著成功。

基于CNN的TSC方法最近不断浮现,在这些方法之中,最吸引人的有Multi-scale Convolutional Neural Network(MCNN), Fully Convolutional Networks (FCN), Gramian Angular Field (GAF)-Markov Transition Field (MTF) CNN model, and Recurrence Plots CNN model。进一步,这些方法可以依据输入到CNN中的数据分为两大类,一类是MCNN和FCN,使用原始时序数据预处理后的1D数据,另一个包括GAF-MTF和RPCNN,后者将原始时间序列数据转换为2D图像。具体而言,MCNN通过多尺度分支和多频率分支对原始时间序列数据执行各种转换。一D卷积层提取不同分支的特征,并在局部卷积阶段之后立即进行串联,然后将所有这些特征喂入完整的卷积阶段以获得最终结果。**MCNN解决了先前研究的局限性,该研究仅在单个时间尺度上提取特征。**但是,输入时间序列数据的增加将导致无用信息的增加和更大的内存消耗,这可能会导致分类精度降低。FCN由Wang等人提出,建立在TSC领域应用深度神经网络(DNN)的标准基线。在他们的研究中,包括多层感知器(MLP), FCN和残留网络(ResNet)在内的三个DNN模型无需进行任何特征工程和数据预处理即可应用于TSC,这意味着一维原始时间序列数据直接馈送到DNN模型。**FCN以简单的协议和较小的模型构建和部署复杂性实现了卓越的性能,为将来的研究提供了坚实的基准和良好的起点。**但是,Wang和Oates认为将时间序列编码为图像的问题使机器可以在视觉上识别,分类和学习结构和图案。他们使用GAF将原始时间序列数据转换为极坐标系统,并使用MTF以沿时间轴的一阶马尔可夫链的方式计算转移概率,这些时间序列被视为时间

序列的静态和动态信息。应用**Tiled-CNN模型对GAF-MTF表示后对时间序列进行分类**，但是与其他最新方法相比，它们分类的**准确性没有竞争力**，这可能是优于MTF提供对动态信息不完整导致对，此外，Hatami提出了一种基于回归图RP的表示方法，利用CNN模型将时间序列转换为二维图像，用于TSC。在他们的研究中，时间序列被看作某些不同于循环的行为，例如周期性/不规则的周期性，这是动态系统中的惊天现象，而使用RP方法的主要思想是写实某些轨迹在那些点返回到以前的状态，将包括两个卷积层和两个全脸阶层的CNN模型应用于对RP生成对图像进行分类，但是由于**CNN模型对结构简单**，他们的方法在标准数据集上对结构**表现是不平衡的**。最近，自Lecun等人以来，出现了一些先进的CNN模型，受基于CNN的计算机视觉技术的最新成就的启发，我们打算将TSC任务视为图像识别任务。具体而言，将时间序列数据首先编码为图像，然后通过CNN模型自动提取显着特征以进行分类。在这项研究中，我们提出了一种**使用相对位置矩阵和卷积神经网络（RPMCNN）进行TSC任务的新型深度学习框架**。具体而言，提出了一种称为相对位置矩阵Relative Position Matrix（RPM）的2D表示方法，将原始时间序列数据转换为2D图像，并提出了一种改进的CNN模型对这些2D图像进行分类。结果表明，我们的最新方法在UCR档案库中的20个标准数据集中有10个获得了最佳性能，而以前的和目前的最佳分类方法是9个。

## Contribution

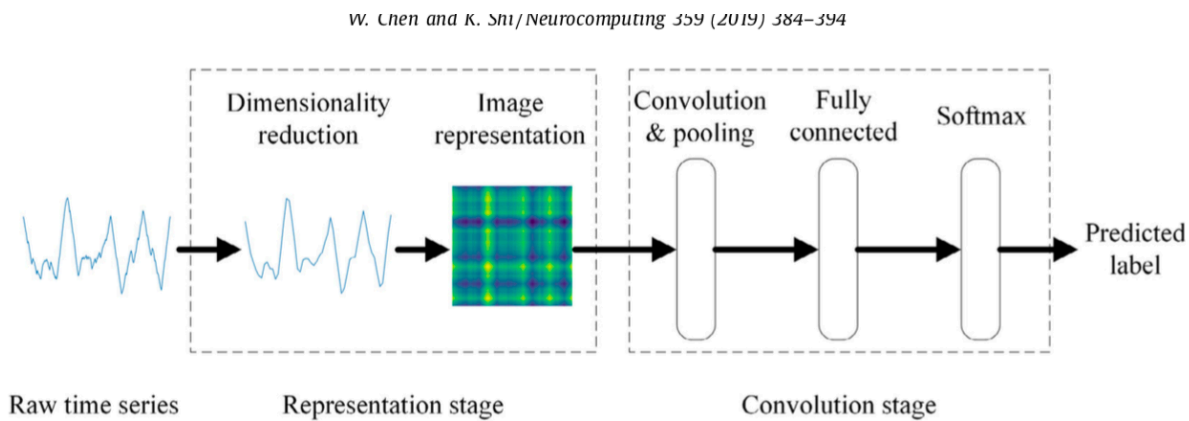
文章的主要贡献如下：

- 1、提出了一个新的简单且有效率的转化为2D图像的数据表达形式：相似位置矩阵RPM。
- 2、其次，提出了一种使用适当结构的改进的CNN模型，以**减少过拟合的风险**。同时，它仍然具有一定的模型复杂度以令人满意的泛化能力。
- 3、与其他传统和最先进的方法相比，该方法在准确性上有了显著提高。

## Methodology

### RPMCNN Framework

相似位置矩阵卷积神经网络框架如下图描述：



**Fig. 1.** Overall architecture of RPMCNN.

要有两个阶段，一个是表示阶段，在这个阶段我们首先使用降维方法降低时间序列的维度，然后在使用RPM将处理后的时间序列转化为2D图像；第二个阶段是卷积阶段，我们构建了一个改进后的CNN框架，该框架含有几个卷积层，每个卷积层后面跟了一个最大池化，还有全连接层以及一个softmax层去生成最后的结果。

## Representation stage

假设有一时间序列  $X = x_1, x_2, x_3, x_4 \dots x_n$ , where  $t_i$  is the value at time stamp  $i$  and the time series length is  $n$ .

### Z-score

首先, 将z-score归一化应用于原始时间序列数据以获得标准正态分布,  $Z = z_1, z_2, z_3, z_4 \dots z_n$

$$z_i = \frac{t_i - \mu}{\sigma}, i = 1, 2, \dots, n$$

$\mu$  is the mean and  $\sigma$  is the standard deviation (标准差) of  $T$ .

### PAA

然后, 我们使用PAA方法去减小 $Z$ 的维度, 我们选择适当的reduction factor  $k$ 来生成新的平滑时间序列, 其中  $X = x_1, x_2, x_3, x_4 \dots x_n$ , 由以下等式表示:

$$x_i = \begin{cases} \frac{1}{k} \sum_{j=k*(i-1)+1}^{k*i} z_j, i = 1, 2, \dots, m, \left\lceil \frac{n}{k} \right\rceil - \left\lfloor \frac{n}{k} \right\rfloor = 0 \\ \frac{1}{k} \sum_{j=k*(i-1)+1}^{k*i} z_j, i = 1, 2, \dots, m-1, \left\lceil \frac{n}{k} \right\rceil - \left\lfloor \frac{n}{k} \right\rfloor > 0, \\ \frac{1}{n - k*(m-1)} \sum_{j=k*(m-1)+1}^n z_j, i = m, \left\lceil \frac{n}{k} \right\rceil - \left\lfloor \frac{n}{k} \right\rfloor > 0, \end{cases}$$
$$m = \left\lceil \frac{n}{k} \right\rceil \quad (2)$$

$m$ 是时间序列 $X$ 的长度。

简而言之, 归一化的时间序列数据从 $n$ 维减少为 $m$ 维, 同时通过计算分段常数的平均值来保留原始序列的近似趋势。随后, 我们通过RPM构造一个  $m \times m$  矩阵  $M$ , 该矩阵计算两个时间戳之间的相对位置, 以将预处理的时间序列  $X$  转换为2D矩阵。

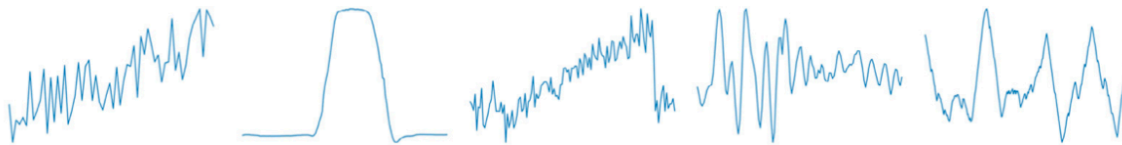
时间戳 $i$ 处的每个值分别被视为 $M$ 的每一行中的参考点, 变换方程如下:

$$M = \begin{bmatrix} x_1 - x_1 & x_2 - x_1 & \dots & x_m - x_1 \\ x_1 - x_2 & x_2 - x_2 & \dots & x_m - x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1 - x_m & x_2 - x_m & \dots & x_m - x_m \end{bmatrix}$$

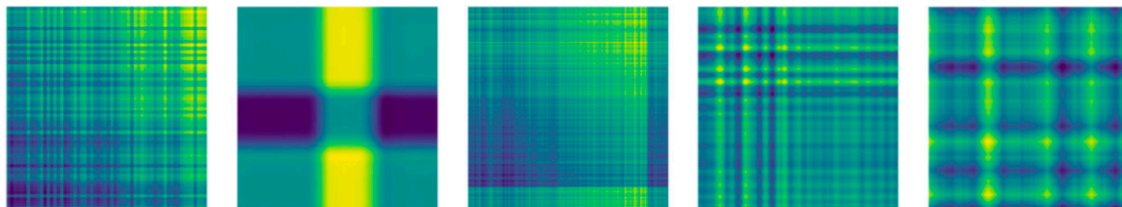
显然, 时间序列的每两个时间戳之间通过矩阵 $M$ 进行链接以获取它们的相对位置,  $M$ 每一行和每一列都以某个时间戳为参考点来包含整个时间序列的信息。更进一步, 这篇论文提出的框架优点在于, 相似位置矩阵RPM可以被视作一种数据增强的方法, 通过提供时间序列的冗余功能来提高泛化能力。 $M$ 的每一行都显示的具有不同参考点的时间序列, 而每一列显示了前者的镜像, 从而提高了相反的视觉来查看时间序列。[Each row of  $M$  shows the time series with varies reference points and each column shows the mirror of the former, which offers a reversed perspective to view the time series.] 最后, 应用最小-最大归一化将 $M$ 转换为灰度值矩阵, 并通过一下公式获得最终的矩阵 $F$  (彩色):

$$F = \frac{M - \min(M)}{\max(M) - \min(M)} \times 255$$

Fig.2.显示了来自UCR存档的前五个不同数据集上的原始时间序列数据, 以及RPM生成的图像。



(a) Raw time series data on top five different datasets from the UCR archive: SyntheticControl, GunPoint, CBF, FaceAll, and OSULeaf (from left to right respectively).



(b) Images generated by our framework arrange in the same order with (a).

**Fig. 2.** Images generated by RPMCNN.

这些图像指示原始时间序列数据的一些有用信息。例如，黑色区域表示原始时间序列的较低值，反之亦然；纯色区域表示原始时间序列的值没有变化或变化缓慢；颜色从暗到亮或从亮到暗的变化频率指示原始时间序列的波形，这意味着颜色变化的高频导致锯齿波形，而颜色变化的低频导致平滑的波形。此外，原始时间序列数据中嵌入的模式和功能也包含在转换后的图像中。

如果从TSC的角度去看这些图像，如Fig.3.所示，很容易从RPM生成的图像中直观地看到和解释类内和类间的相似性。

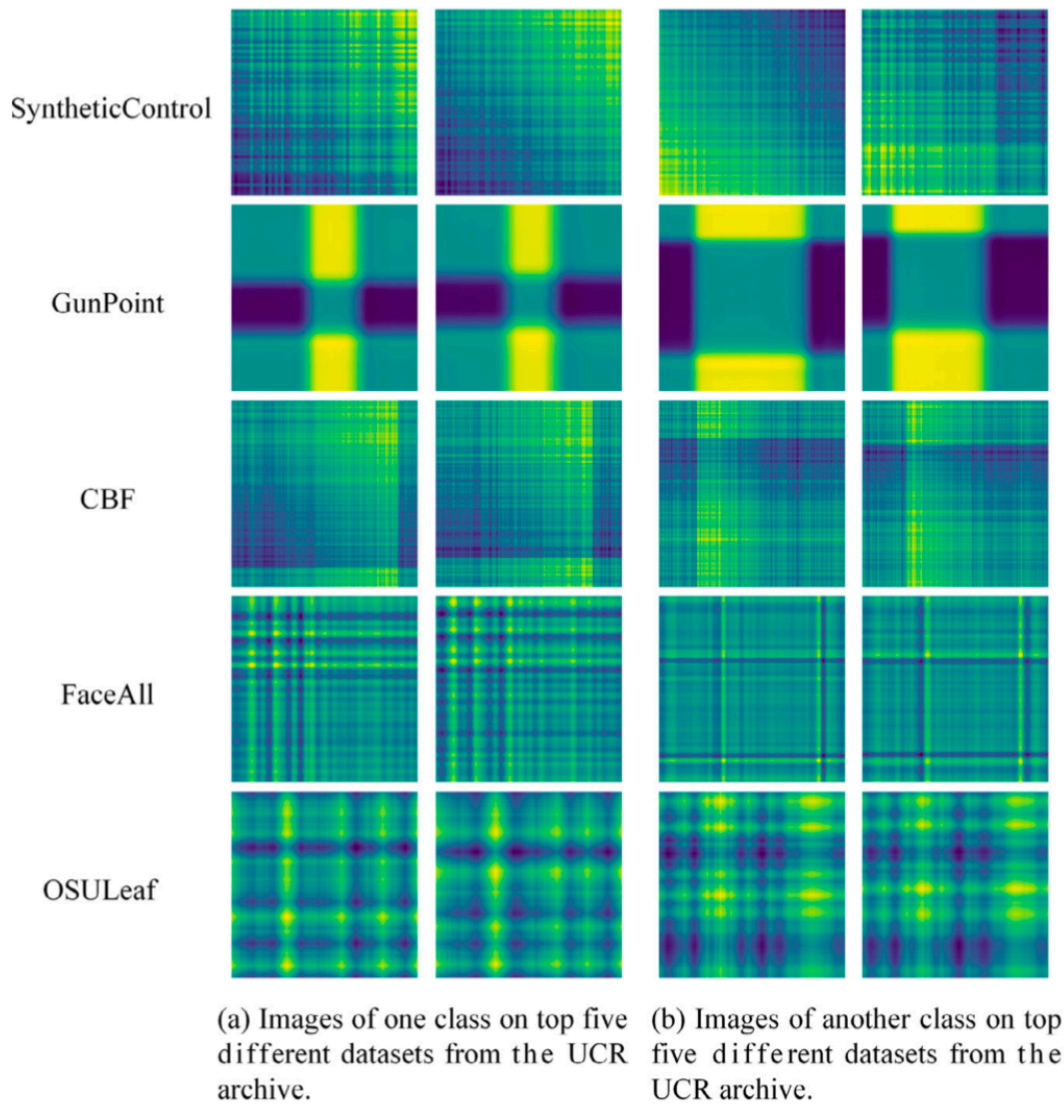


Fig. 3. Image similarity of intra-class and inter-class.

接下来，论文在Convolution stage 应用改进的CNN架构，以基于2D图像处理TSC任务，该2D图像也可以视为图像识别任务。

## Convolution stage

一旦原始时间序列数据已转换为图像，就可以训练CNN模型对这些图像进行分类。

论文设计了一种基于VGGNet的改进的CNN模型。它由较少的卷积层和完全连接的层组成，以减少模型复杂性，并应用批处理归一化层以减少过度拟合的风险，这更适合于TSC任务的发生。

VGGNet由牛津大学计算机视觉组合和Google DeepMind公司研究员一起研发的深度卷积神经网络。它探索了卷积神经网络的深度和其性能之间的关系，通过反复的堆叠33的小型卷积核和22的最大池化层，成功的构建了16~19层深的卷积神经网络。VGGNet获得了ILSVRC 2014年比赛的亚军和定位项目的冠军，在top5上的错误率为7.5%。目前为止，VGGNet依然被用来提取图像的特征。

如Fig.4.所示，括号中的符号“c”，“p”，“f”，“o”分别表示卷积层，池层，完全连接层和输出层。符号D表示输入图像的大小，符号N表示输入图像的类别数。“@”之前和之后的数字分别表示本层中的特征图的数量和特征图的形状。注意，归因于空间的限制，归一化和整流层未在图中示出。



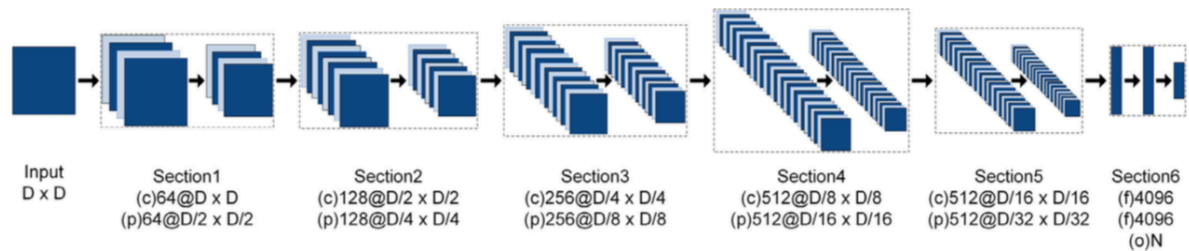


Fig. 4. Architecture of RPCNN.

论文提出的CNN模型包括六个部分。对于前五个部分，每个部分包括以下四层。第一层是卷积层，是整个CNN结构的核心层。它将前一层的输出与一组内核进行卷积，以提取原始图像不同位置的一组特征，并获得称为特征图的表示。第二层是批归一化（Batch Normalization, 简称为BN）层，用于归一化前一层中不同特征图的值。通过将先前未固定的数据分布转换为固定的数据分布，可以有效解决反向传播过程中梯度消失和爆炸的问题。第三层是整流（rectification layer）层，它通过称为SELU的激活函数映射前一层的输出。SELU的优点是引入了自归一化属性，即使在存在噪声和干扰的情况下，也可以更快、更可靠地学习深度神经网络。最后一层是最大池化层，它通过压缩由先前层处理的特征图来降低计算负担。最后一部分由三层组成，其中包括两个完全连接的层和一个输出层。全连接层的作用是将要素集成到要素图中，以获得原始图像要素的高级表示。输出层通过softmax函数实现原始图像的分类，该函数提供与训练样本的真实标签一致的分类结果的后验概率，并预测测试样本的标签。建立了整个RPCNN框架后，论文进行了全面的实验。

## Experiments

我们评估了UCR时间序列分类档案1的数据集上所有方法的性能。我们选择与其他最新方法相同的20个数据集。请注意，数据集已经标准化为零均值和单位方差。

**Table 1**  
Error rates of different approaches on 20 UCR time series datasets.

Dataset	1NN-DTW	BOSS	SAX-VSM	LTS	COTE	FCN	MCNN	GAF-MTF	RPCNN	RPCNN
50 Words	0.31	0.301	N/A	0.232	0.191	0.321	<b>0.19</b>	0.301	0.26	0.193(4)
Adiac	0.396	0.22	0.38	0.437	0.233	<b>0.143</b>	0.231	0.373	0.28	0.182(3)
Beef	0.367	0.2	<b>0.033</b>	0.24	0.133	0.25	0.367	0.233	0.08	0.133(3)
CBF	0.003	<b>0</b>	0.02	0.006	0.001	<b>0</b>	0.002	0.009	0.005	<b>0(6)</b>
Coffee	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	0.036	<b>0</b>	<b>0</b>	<b>0(6)</b>
ECG	0.232	<b>0</b>	0.14	<b>0</b>	<b>0</b>	0.015	<b>0</b>	0.09	<b>0</b>	<b>0(2)</b>
FaceAll	0.192	0.21	0.2	0.217	0.105	<b>0.071</b>	0.235	0.237	0.19	0.189(2)
FaceFour	0.17	<b>0</b>	<b>0</b>	0.048	0.091	0.068	<b>0</b>	0.068	<b>0</b>	0.034(2)
Fish	0.177	<b>0.011</b>	0.017	0.066	0.029	0.029	0.051	0.114	0.085	0.034(8)
GunPoint	0.093	<b>0</b>	0.007	<b>0</b>	0.007	<b>0</b>	<b>0</b>	0.08	<b>0</b>	<b>0(1)</b>
Lightning2	0.131	0.148	0.19	0.177	0.164	0.197	0.164	0.114	<b>0</b>	<b>0(8)</b>
Lightning7	0.274	0.342	0.3	0.197	0.247	<b>0.137</b>	0.219	0.26	0.26	0.247(4)
OliveOil	0.167	<b>0.1</b>	<b>0.1</b>	0.56	<b>0.1</b>	0.167	0.133	0.2	0.11	<b>0.1(16)</b>
OSULeaf	0.409	<b>0.012</b>	0.107	0.182	0.145	<b>0.012</b>	0.271	0.358	0.29	0.186(4)
SwedishLeaf	0.208	0.072	0.25	0.087	0.046	<b>0.034</b>	0.066	0.065	0.06	<b>0.034(2)</b>
SyntheticControl	0.007	0.03	0.25	0.007	<b>0</b>	0.01	0.003	0.007	<b>0</b>	0.003(1)
Trace	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	0.01	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0(4)</b>
TwoPatterns	0.096	0.016	0.004	0.003	<b>0</b>	0.103	0.002	0.091	0.17	<b>0(4)</b>
Wafer	0.02	0.001	0.001	0.004	0.001	0.003	0.002	<b>0</b>	<b>0</b>	<b>0(4)</b>
Yoga	0.164	0.081	0.16	0.15	0.113	0.155	0.112	0.196	<b>0</b>	0.11(6)
Wins	2	9	5	4	5	9	5	3	9	<b>10</b>
MR	7.45	3.65	5.75	5.55	3.8	4.6	4.65	6.4	3.7	<b>2.35</b>
ME	0.171	0.087	0.114	0.131	0.081	0.086	0.104	0.14	0.09	<b>0.072</b>
MPCE	0.034	0.0137	0.0229	0.0238	0.0152	0.0182	0.019	0.0265	0.0108	<b>0.0106</b>

## Evaluation Metrics

在本文中，所有TSC方法的性能都通过四个指标进行评估，包括达到最低错误率（Wins），平均排名（MR），平均错误（ME）和平均每类的数据集数量 错误（MPCE）。具体来说，MPCE被定义为每类错误（PCE）的算术平均值。对于具有K个数据集的数据集池，第k个数据集具有类别数 $c_k$ 和相应的错误率 $e_k$ ，则MPCE的计算如下：

$$PCE_k = \frac{e_k}{c_k}$$

$$MPCE = \sum_{k=1}^k PCE_k$$

## Comparison

然后，论文分别与当前先进的方法、不同PAA reduction factor、其他2D-CNN的表现形式（GAF-MTF and RP at the representation stage to compare with RPM）、其他CNN结构进行对比。实验结果证明RPMCNN达到了最先进的性能，并且在很大程度上优于现有方法。我们还分别对二维representation和CNN结构的进行了控制变量的研究（ablation studies），证明了RPM和CNN结构的组合更适合TSC任务。

## Future Work

---

在未来的工作中，我们将探索更深层的结构，以处理来自实际应用程序的更大和更复杂的时间序列数据，并将我们的方法应用于更困难的TSC任务中，例如多元时间序列场景。我们感兴趣的另一项重要的未来工作是，不同的深度学习架构如何执行TSC任务，并探索不同架构的整体学习方法，以实现更好的性能。