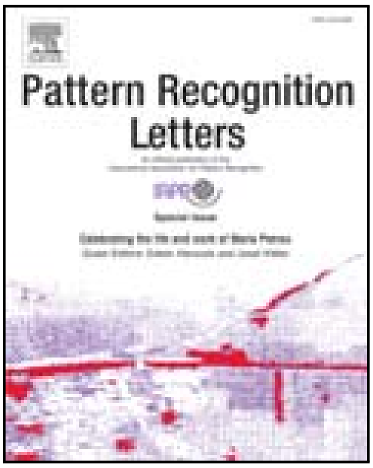
Journal Pre-proof



CNN based Spatial Classification Features for Clustering Offline Handwritten Mathematical Expressions

Cuong Tuan Nguyen , Vu Tran Minh Khuong , Hung Tuan Nguyen , Masaki Nakagawa

PII: S0167-8655(19)30378-2
DOI: https://doi.org/10.1016/j.patrec.2019.12.015
Reference: PATREC 7741

To appear in: *Pattern Recognition Letters*

Received date: 1 June 2019
Revised date: 1 December 2019
Accepted date: 21 December 2019

Please cite this article as: Cuong Tuan Nguyen , Vu Tran Minh Khuong , Hung Tuan Nguyen , Masaki Nakagawa , CNN based Spatial Classification Features for Clustering Offline Handwritten Mathematical Expressions, *Pattern Recognition Letters* (2019), doi: https://doi.org/10.1016/j.patrec.2019.12.015

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier B.V.

Journal Pre-proof

Highlights

- High performance of clustering images with 0.99 purity on CROHME
- Hierarchical representation of spatial classification features for clustering
- Propose a novel global pooling method for weakly supervised learning
- Classifying and localizing multi-scale symbols of handwritten images

Journal Pre-proof

Journal Pre-proof

CNN based Spatial Classification Features for Clustering Offline Handwritten Mathematical Expressions

Cuong Tuan Nguyen*, Vu Tran Minh Khuong, Hung Tuan Nguyen, and Masaki Nakagawa

Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588 Japan.

ABSTRACT

To help human markers mark a large number of answers of handwritten mathematical expressions (HMEs), clustering them makes marking more efficient and reliable. Clustering HMEs, however, faces the problem of extracting both localization and classification representation of mathematical symbols for an HME image and defining the distance between two HME images. First, we propose a method based on Convolutional Neural Networks (CNN) to extract the representations for an HME. Symbols in various scales are located and classified by a combination of features from a multi-scale CNN. We use weakly supervised training combined with symbols attention to enhance localization and classification predictions. Second, we propose a multi-level spatial distance between two representations for clustering HMEs. Experiments on CROHME 2016 and CROHME 2019 dataset show the promising results of 0.99 and 0.96 in purity, respectively.

2012 Elsevier Ltd. All rights reserved.

Journal Pre-proof

* Corresponding author. Tel.: +0-000-000-0000; fax: +0-000-000-0000; e-mail: ntcuong2013@gmail.com

Journal Pre-proof

1. Introduction

For promoting and evaluating the studying process of any student, several kinds of quizzes, multiple-choice questions, descriptive questions, and so on have been used depending on different subjects. From the past, teachers must mark the answers from their students, which requires substantial time and effort for marking the answers in a limit period. The more answers they must mark in a shorter period, the more they are exhausted and lose their focuses on giving feedback to their students. Multiple markers could be employed, but inconsistency among them causes another problem. Although computers have been used to reduce the workload of teachers in various aspects such as preparing teaching materials, managing students' information, communicating with them and so on, they are limitedly employed for marking. Computer Based Testing (CBT) or Computer-Assisted Assessment (CAA) has been proposed and practiced for multiple-choice questions as well as unambiguous questions for which their answers are typed in [1,2]. However, it has not been successfully employed for descriptive questions in mathematics such as those answered by Handwritten Mathematical Expressions (HMEs) although the descriptive questions are best to assess students' understanding.

There are two main approaches for applying computers for marking HME answers: Automatic Marking (AM) and Computer Assisted Marking (CaM). AM is not perfect due to the problem of HME recognition so that marking must be confirmed by teachers and/or students. Currently, the performance of computer recognition of unconstrained HMEs is from 44.55% to 46.55% [3], which is still far from the acceptable rate for AM. On the other hand, CaM avoids the above problems since their goal is not to mark the HME answers automatically but help markers mark answers efficiently and reliably. Powergrading [4] groups a large number of typed-in text answers into a smaller number of groups and let the teachers mark each group of answers by one action. Hence, the workload of teachers is significantly reduced, thus fewer teachers are employed for marking. Students' anxieties for marking will be reduced since the final marking is manually made by teachers. Hence, the clustering-based CaM for HME answers is expected to make the marking process more efficient and decrease marking errors as well as reduce score fluctuations compared to completely manual marking.

Here we focus on offline (bitmap image) HMEs which can be applied for answers on paper exams. To apply clustering for HME answers, however, there are the problems of extracting the structural representation of mathematical symbols for an HME image and defining the distance between two HME images. Recently, there are some preliminary researches on clustering HMEs using ensembled features. Ung et al. employed the recognition based features for online (pen movement trajectory) HMEs, since online recognition produces a better recognition rate than offline recognition [5]. Vu et al. combined different kinds of features from low-level features (directional features) to high-level features (bag-of-symbols, bag-of-relations, and bag-of-positions) to form a represented vector for each offline HME [6]. The method, however, encountered the problems of symbol segmentation and relation determination so that many heuristic rules were applied. These heuristic rules are not robust against ambiguities caused by various writing styles from writers.

Recently, there are some deep neural network-based approaches to cluster images such as Deep Embedded Clustering (DEC) [7], Deep Clustering Network (DCN) [8], Siamese Network [9], Deep Adaptive Clustering (DAC) [10]. These methods, however, focused on learning class discriminative features for clustering images, which are hard to generalize for unconstrained HMEs. Since HMEs have unlimited compositions of symbols and structured relations, learning embedded features to represent the spatial information of MEs is difficult. In this work, we proposed a deep neural network-based method to learn the structural representation of local features in HME images. The network learns to detect and classify the symbols in HME images and produce the symbol classification as local features and spatial information of these symbols as the structure of the local feature. The hierarchical combination of the structural representation is used for clustering HME images.

In the rest of this paper, section 2 briefly introduces some deep neural network-based clustering methods as related works. Section 3 describes the object localization. Section 4 describes networks and methods. Section 5 shows the experiments and results. Section 6 draws our conclusions.

2. Related works

Among the deep neural network-based clustering methods, Deep Embedded Clustering (DEC) proposed by Xie et al. [7] is an early study, which has the structure of an Auto Encoder network (AE) [11]. Its objective is to learn a non-linear mapping from a data space to a lower-dimensional feature space that could be used for clustering. First, DEC is pretrained with reconstruction loss as same as AE. Next, the encoder (or mapping function) without decoder is trained using Kullback–Leibler divergence minimization to improve the cluster results based on an auxiliary distribution computed during the pretrain stage. Recently, DeepCluster [8] is proposed for clustering without using the AE structure which iteratively clustered deep features and retrained the CNN to learn the pseudo-label by cluster assignments. Note that, these two methods are trained using clustering loss.

Another approach is based on the pairwise similarity and dissimilarity instead of clustering loss, where the similarity and dissimilarity are the distance between two samples of the same and different clusters, respectively. Siamese Network [9] and Deep Adaptive Clustering (DAC) [10] are trained to obtain the embedded features by iteratively cluster the samples by their pairwise similarity and learning to minimize the similarity within-group and maximize the dissimilarity between samples of different group. Although Siamese Network and DAC use a similar network structure, DAC has a constraint on selecting samples based on their cosine similarities while Siamese Network does not have any constraints. DAC could learn without explicit cluster labels while Siamese Network requires them to learn discriminative features.

An advantage of the deep neural network-based clustering methods is the representation ability from high dimensional feature space. Clustering results of many traditional clustering methods that are mainly based on similarity measures, for example, distance functions, are usually degraded when there is a high dimensional feature space. These deep neural network-based methods, however, are trained to extract low-dimensional features from high dimensional feature space which is driven by data. In the case of HMEs clustering problem, there is a challenge on generalization solution. These deep neural network-based clustering methods are designed to learn category discriminative features for clustering images while HMEs have unlimited compositions of symbols and structured relations. Thus, HMEs clustering requires not only symbol/object discriminative features but also spatial information of MEs.

Journal Pre-proof

3. Multi-scale object localization and classification features

In this section, we make an overview of multi-scale object localization and classification features using weakly supervised learning.

3.1. Multi-scale object localization and classification

To deal with multi-scale object detection, one approach is to use a combination of multi-level features from different Convolutional Neural Networks (CNNs) [12]. The method, however, requires a large amount of training data in different scales since the networks learn features for each scale independently. Another approach is to use a single CNN to extract features from multiple scales samples of input image [13,14] and aggregates them into a single feature. For this approach, the networks are designed to detect from a fixed scale object, and inference for multi-scale objects is obtained from the proper scale depended on input images.

Faster R-CNN [15] used ROI pooling to rescale the windows of objects into fixed-size windows for dealing with the problem of multi-scale object detection. The method also used a single CNN to learn, however, it needs location information to train object location prediction.

Fully Convolutional Networks [16] use multiple levels of CNN features to classify each pixel location in an input image. It could be used to detect multi-scale objects. The method, however, needs supervised labels for each pixel.

3.2. Weakly supervised learning for object localization and classification

Weakly supervised learning [13,17] is a learning method for CNN object localization and classification. The method applies a global max pooling layer (GMP) [13] or global average pooling (GAP) [17] to aggregate local features by CNN through the spatial dimensions. The aggregated features represent object classes exist or not in the images. Let F_c be the feature by CNN, GMP and GAP is represented as follows:

$$GMP(F_c) = \max_{xy} F_c(x, y) \tag{1}$$

$$GAP(F_c) = \text{average}_{xy} F_c(x, y) \tag{2}$$

where x, y represents the spatial location of the local features by CNN.

Without location information, weakly supervised learning uses the labels which indicate each class exists or not in the images. For each class c, there are two possibilities that it exists in the input image or not. Let t_c be the binary target of {0, 1}, where 0 indicates that the class c does not exist and 1 indicates that it does exist in the image, y_c be the classification probability of class c, the binary cross entropy loss for the class c is obtained as follows:

$$BCE_c = -\left(t_c \log y_c + (1 - t_c)\log(1 - y_c)\right) \tag{3}$$

Binary cross entropy loss BCE for all the classes calculated by (4) is used to optimize the weighting parameters for the networks.

$$BCE = \sum_c BCE_c \tag{4}$$

3.3. Class activation map

The networks trained by weakly supervised learning can be used to generate class activation maps (CAMs) [17] (or class score maps [13]) by applying to remove the global pooling layer. The output is the activation maps for every class representing the prediction of the class in each location of the output map. CAMs retain both the classification of objects and the spatial structure of these objects.

3.4. Spatial pyramid pooling

Spatial pyramid pooling (SPP) [18] divides an input image of arbitrary size into multi-level of local spatial bins, concatenates the features from the spatial bins into a fixed-size feature representation. The length of features by SPP is equal to *number_of_bins* * *feature_depth*. SPP maintains spatial information of the features by high-level spatial bins. If only the first level of 1x1 bin is used, spatial information is removed from the features.

4. Method

We propose a Hierarchical Spatial Classification (HSC) features for clustering HMEs. The CNN is trained by weakly supervised labels to extract CAMs that contain both the location and classification of symbols in HMEs. CAMs are aggregated by a multi-level of Spatial Pooling to obtain HSC features.

Note that, handwriting symbols in an HME may have different scales according to their roles. For example, the symbols in subscript or superscript are smaller than those in the baseline, which raises a challenge for CNN to detect characters at multiple scales. Thus, we design a single feature extractor for multi-scale inputs using a single CNN to extract features from multiple scales of an input HME image. Our proposed network could choose an appropriate scale for extracting useful features.

Moreover, as weakly supervised learning gathers the features from all the locations of an image, it should focus on learning from the locations that symbols exist. We use an attentive pooling layer that focuses on extracting all locations by each symbol category, which helps our network localize not only a single location but also all locations if a symbol category appears at multiple locations.

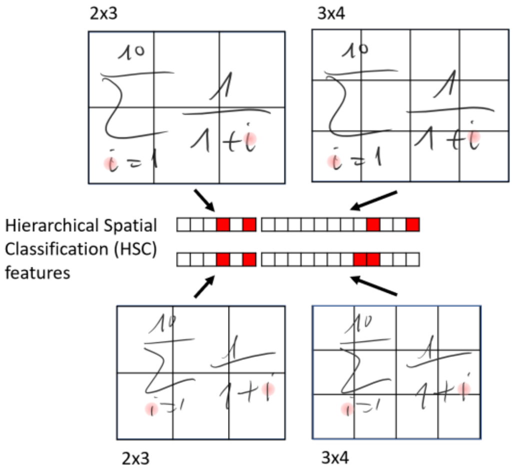


Figure 1. HSC features from Class Activation Map of class ‘i’.

4.1. Hierarchical Spatial Classification features for clustering

Inspired from SPP, we use multi-levels of spatial pooling for extracting Hierarchical Spatial Classification (HSC) feature from CAMs. These spatial pooling, however, allow various aspect ratios instead of 1:1 as in original SPP. Figure 1 shows an example of two levels spatial pooling {2x3, 3x4} for extracting HSC features from CAMs of symbol ‘i’ in two different samples of the same formula. HSC features eliminate the variance of symbol location in different writing styles since it produces the same result if the location of the object is inside the spatial bin. In

Journal Pre-proof

Journal Pre-proof

Figure 1, although the locations of symbol ‘i’ are different for the two samples, the feature by 2x3 spatial pooling of the two images are the same.

The partition size (e.g. 2x3, 3x4) of spatial pooling has effect to clustering. Features by coarse partition robust with variance of symbol locations in HME, but it also less spatial sensitivity to discriminate complex HMEs. The effect is reverse for the fine partition. Therefore, HSC features by multi-levels of spatial pooling from coarse to fine could compensate the disadvantage of each single spatial pooling level. In Figure 1, although 3x4 bins produce the different features of the symbol ‘i’ for the same formula, 2x3 bins could produce the same features so that the hierarchical of {2x3, 3x4} bins could produce more robust features.

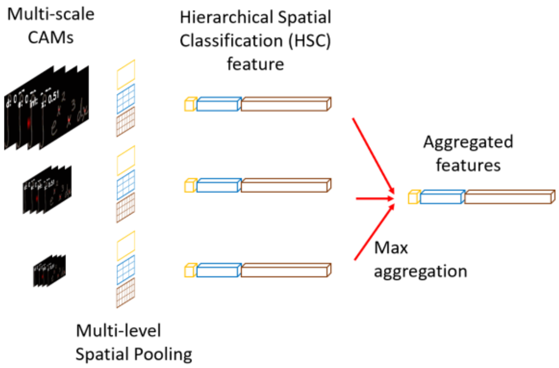


Figure 2. Multi-scale aggregated HSC features for clustering.

Multi-scale CAMs vary in their sizes since they are dependent on the input sizes of HMEs. Thus, we apply multi-level spatial pooling for each scale of CAMs feature to conform the multi-scale CAMs into the HSC features of the same size and aggregate the features by max aggregation. Figure 2 illustrates the detailed approach.

4.2. Overall of our proposed network

Journal Pre-proof

Journal Pre-proof

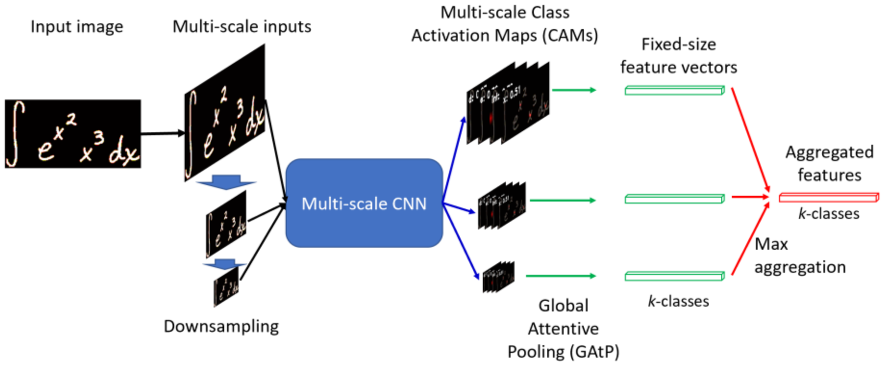


Figure 3. Overall of proposed network

The overall architecture of the proposed network is shown in Figure 3. Input images are downsampled by a half and one-fourth along each spatial dimension to obtain multi-scale inputs. Multi-scale CAMs are extracted from multi-scale inputs by a single CNN-based feature extractor that reduces the complexity of our network and learns the shared features from multi-scale inputs. Next, a global attentive pooling layer is employed to obtain the fixed size feature vectors (*k*-dimension where *k* is the number of categories) from multi-scale CAMs. These fixed feature vectors are aggregated by the max aggregation to retain the best-detected symbol by multi-scale inputs. We use weakly supervised learning to train our network without using symbol location information, which removes human labeling costs for preparing supervised locations.

4.3. Multi-scale CNN for extracting CAMs

Multi-scale CAMs represent both the location and classification of symbols in an HME image, as shown in Figure 4. The network extracts these features for object localization and classification presented in Sect. 2.1. In an HME, the dependencies between symbols are also essential to reduce ambiguity for classifying the symbols. For example, recognizing bracket symbols or fraction symbols requires not only local information of the symbols but the context of other adjacent symbols. Therefore, we concatenate the last two levels of convolutional layer outputs in order to use the local context of symbols for recognition. The symbol level block as shown in Figure 4 consist of convolution and pooling layers with the output is the symbol-level feature by concatenating the output of last two convolution layers. The symbol-level features then classified by the expression-level block. The detailed architectures of the blocks are shown in Table 1. The symbol-level block is configured to handle the input symbols in 32 x 32 to recognize small symbols in an HME. The expression-level block outputs the probability for the symbol characters. The CNN is learned by weakly supervised learning, where a global pooling layer (GAP, GMP) is added after the classifier.

Table 1: Network configuration

Module	Type	Configurations
Symbol-level block	Convolution - ReLU	#maps:64, k:3×3, s:1, p:1
	MaxPooling	#k:2×2, s:2
	Convolution - ReLU	#maps:128, k:3×3, s:1, p:1
	Convolution - ReLU	#maps:256, k:3×3, s:1, p:1
	MaxPooling	#k:2×2, s:2
	Convolution - ReLU	#maps:512, k:8×8, s:1, p:1
	Convolution - ReLU	#maps:512, k:3×3, s:1, p:1
Expression-level block	Convolution - ReLU	#maps:1024, k:1x1, s:1, p:1
	Convolution - Sigmoid	#maps:101, k:1x1, s:1, p:1

(k: kernel size, s: stride, p: padding)

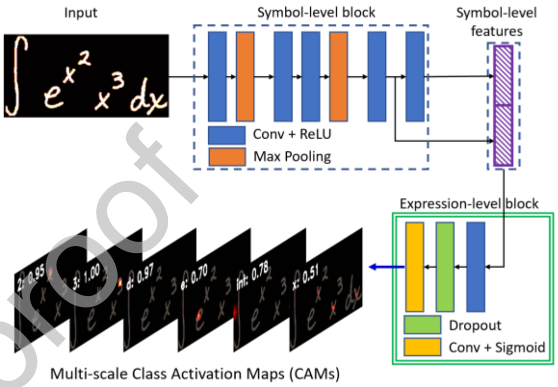


Figure 4. Multi-scale CNN for extracting CAMs

4.4. Global attentive pooling

The networks with GMP, however, only learn from the highest activation location for each class. On the other hand, the networks with GAP learns from all the locations in the CNN feature. Restricting GAP in the object region of interest (ROI) helps the model learns well [17], but it requires location information for all the objects.

To enhance localization and classification, we extract the attention of input images to train the classifier through a global attentive pooling layer (GAtP). The global attentive pooling layer aggregates the features for learning each separate class. For each class, an attentive map focuses on the locations of the symbols in the class, aggregates the local features in these locations into a single feature. The attentive pooling makes the networks learn from multiple locations where the symbols occur. Therefore, it can overcome the problem of GAP and GMP.

First, we obtain the CAMs denoted as M_c for all the classes by applying the classifier to the CNN features. For a class c, we multiply M_c with the CNN features F to extract attentive features for class c. We aggregate the features by an average pooling on the attentive features as follows:

$$Att_{F_c} = \frac{\sum_{x,y} M_c(x,y)F(x,y)}{xy} \qquad (5)$$

where x, y denotes the spatial dimensions of CNN features. The aggregated feature is fed to the classifier for training the class c.

The output y_c of the class c by the classifier f with attentive features Att_{F_c} is obtained as follows:

$$y_c = \langle f\left(Att_{F_c}\right), u_c \rangle \qquad (6)$$

Journal Pre-proof

where u_c is the one-hot vector of class c in total k classes, $\langle a, b\rangle$ is the inner product of vector a and vector b. The network is trained by applying the BCE_c loss (1) using y_c obtained by (4).

For implementation, let the attentive features for all the classes $Att_F = [Att_{F_1} \quad Att_{F_2} \quad Att_{F_k}]$, the output $y = [y_1 \quad y_2 \quad y_k]$ is obtained as follows:

$$\begin{aligned} y &= [\langle f(Att_{F_1}), u_1\rangle \quad \langle f(Att_{F_2}), u_2\rangle \quad \langle f(Att_{F_k}), u_k\rangle] \\ &= \text{diag}\left([f(Att_{F_1}) \quad f(Att_{F_2}) \quad f(Att_{F_k})]\right) \\ &= \text{diag}\left(f(Att_F)\right) \qquad (7) \end{aligned}$$

where diag(A) is the operator to get elements on the main diagonal of matrix A.

Figure 3 shows the details of our global attentive pooling layer. From top to bottom, the multi-scale HSC features of each class are multiplied with symbol-level features to obtain an attentive feature vector by Eq. (5). Next, k attentive feature vectors are fed into the expression-level block to obtain k classified vectors as presented by Eq. (6) which are combined to form a fixed-size feature vector by Eq. (7).

4.5. Combination of GAtP and GMP

Global attentive pooling may not be trainable at the early epochs of training since the attentive maps produce low activations. Global attentive pooling also produces low activations for the classes which do not exist in an image. Hence, negative learning is not well performed. We overcome the problem by a combination of the output activations by both GAtP and GMP as follows:

$$y_c = \frac{1}{2}\left(\langle f(Att_{F_c}), u_c\rangle + \langle f(GMP(F)), u_c\rangle\right) \qquad (8)$$

Combining output activations makes the networks learn from both the loss provided by GAtP features and that by GMP features. At early epochs, networks benefit from the GMP loss and at the final states, networks benefit from the GAtP loss.

5. Experiments

5.1. Dataset

The main experiments are on CROHME 2016 dataset [19] and CROHME 2019 dataset [20]. The number of training, validation and test sample are 8834, 986, and 1147 in CROHME 2016 and 9993, 986 and 1199 in CROHME 2019, respectively. In CROHME 2016, we reserve the formulas from the folder ‘expressmatch’ of training set which contains the most samples per formula for clustering evaluation. In CROHME 2019, we reserve 49 most frequent formulae as a total of 1264 samples for clustering evaluation. All the samples are considered as the answers to a question. The remaining samples from training set are used for training. The validation set is the same as the original configuration. The formula of test samples is not available in train samples.

To compare with other approaches, we also do the experiments on two collected datasets [4, 5]. HME_Answer_1 was collected from 23 students. Each student answered 22 questions by writing one correct answer and two incorrect answers. The total number of samples is 1513 due to there are missing answers from the students. HME_Answer_2 was collected from 21 students. Each student wrote 3 times of 50 HMEs on 3 writing interfaces which are (i) without any guiding line, (ii) with only a center line and (iii) with a center, a top, and a bottom line. As the results, the total number of online HMEs is 3150 samples belonging to 50 classes. We consider all the samples belong to the same question. For the two databases, we use 5-fold cross-validation divided by writers for training, validation, and testing, where the ratio of train:validation:test sets is 3:1:1. Table 2 shows a summary of the testing set in all the datasets.

Table 2: Summary of the testing dataset.

Database	# Questions	# Diff. Answers per Question	# HME samples
HME_Answer_1	22	3	302
HME_Answer_2	1	50	630
CROHME 2016	1	36	620
CROHME 2019	1	49	1264

5.2. Metrics

We use purity as a measurement for evaluating the clustering task. The measurement is calculated according to (9).

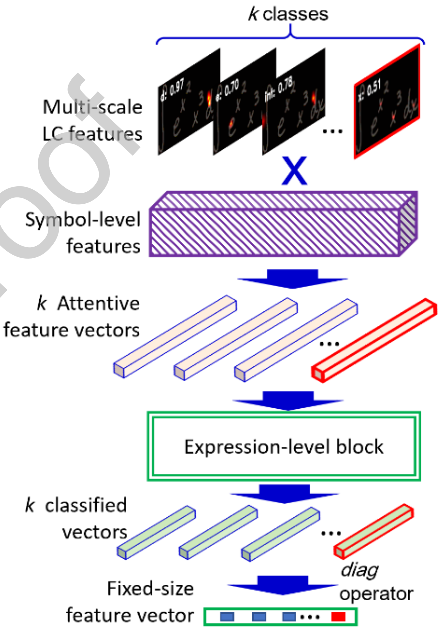


Figure 5. Global attentive pooling (GAtP)

$$Purity(G, C) = \frac{1}{H}\sum_{k=1}^{K} \max_{1\le i\le J}|g_k \cap c_i| \qquad (9)$$

where K and J are the numbers of clusters and classes respectively, H is the number of samples, $G = \{g_1, g_2, \dots, g_K\}$ is a set of obtained clusters, $C = \{c_1, c_2, \dots, c_J\}$ is a set of classes.

High purity, however, is easy to achieve when the number of clusters is huge. For example, if K equals to H, we obtain the purity of one. The number of clusters, however, should not be large since it increases the marking cost. To fairly evaluate the clustering method, we measure the purity for the number of clusters is equal to the number of formulas.

5.3. Experiment setup

All the input images are scaled into 128-pixel height with maintaining aspect ratio. There are images whose width larger than 800 pixels after scaling, they are rescaled to 800-pixel width. Multi-scale images are created by applying average pooling to the input images. There are three input images of the 128-pixel height, 64-pixel height and 32-pixel height. The number of output classes for all three databases is 101 classes as the number of symbols in CROHME dataset.

For spatial pooling, we extract multi-level pooling size of (1x1), (3x5), (3x7) and (5x7) feature map bins, where (1x1)

Journal Pre-proof

spatial pooling does not retain spatial information, the other spatial pooling could retain the spatial information.

We apply cosine distance to measure the dissimilarity between the HSC features extracted from two HME images. We adopt Kmeans++ as a clustering method to clustering the HME images. In order to cluster the HME images by Kmeans++, for each HME image, we calculate the cosine distances of it to all other HME images and use the distances as the representation of the HME images.

5.4. Clustering experiments

In order to evaluate the feature fairly, we set the number of clusters for hierarchical clustering by the same with the number of formulas in the testing sets. For each question in HME_Answer_1, we cluster the answers into three groups and report the average purity overall the questions. For HME_Answer_2, we cluster the answers into 50 groups. For CROHME 2016 and CROHME 2019 we cluster them into 36 groups and 49 groups, respectively.

The clustering result is shown in Table 3. Our method using HSC features of CNN outperforms the online Bag of Features (Online BoFs) [5] and the offline Bag of Features (Offline BoFs) [6] by the large margins on both the HME_Answer_1 and HME_Answer_2. For CROHME 2016 and CROHME 2019, our method also yields the purity of 0.99 and 0.96, respectively. The result of CROHME dataset shows the robustness and high generalizability of the learned features over handcrafted features. We also reproduced the experiments with other image clustering methods as in Table 3. The first method applies PCA with the number of components set to the number of clusters to reduce the dimensionality of the images. Then the reduced features are clustered by Kmeans++. The second method is DAC, which learn to cluster directly from the unlabeled dataset so that we run the experiment on test set without using training dataset. The third method is Siamese models where we learn the model on training set and use it to extract the features on test set then clustering them by Kmeans++. We use backbone of CNNs for both DAC and Siamese models. DAC and Siamese consistently outperform the baseline of Kmeans+PCA due to the learned deep features. HSC features yield the best purity for all the datasets. The result confirms the effectiveness of spatial classification features over the deep features of the whole image.

Table 3: Clustering results (purity)

Method	**HME_ Answer_1**	**HME_ Answer_2**	**CROHME 2016**	**CROHME 2019**
Online BoFs [5]	-	0.92	-	-
Offline BoFs [6]	0.76	0.87	-	-
Kmeans+PCA	0.51	0.45	0.48	0.47
DAC	0.64	0.61	0.77	0.76
Siamese	0.65	0.79	0.88	0.76
HSC feature	**0.88**	**0.98**	**0.99**	**0.96**

5.5. Effect of multi-level spatial features

We evaluate the clustering performance for all three datasets with different configurations of multi-level spatial pooling. The results are shown in Table 4.

First, we confirm the effectiveness of localization information in HSC features. The single-level HSC features with spatial information (spatial sizes different with (1x1)) yield the purity of 0.99 and 0.96 on CROHME 2016 and CROHME 2019, outperform the HSC feature without spatial information where the purity are 0.92 and 0.86. For HME_Ans_1 and HME_Ans_2, however, spatial information does not show its effectiveness. The reason is that in both HME_Ans_1 and HME_Ans_2 the answers are composed of different symbol sets. Thus, classification features are enough for clustering them. On the other hand, varying writing styles make the purity by large-sized spatial pooling HSC features worse than the small-sized spatial pooling HSC features. For CROHME, there are formulas with similar set of symbols but different on structures, thus, classification information is not enough for clustering them.

Second, we confirm the effectiveness of combining multi-level HSC features over single-level HSC features. For both CROHME and HME_Ans_2, multi-level HSC features yield higher purity than single-level features. The best combination in CROHME 2016 is {1x1, 3x5} with purity of 0.99, the best in HME_Ans_2 and CROHME 2019 is {1x1, 3x7} with purity of 0.98 and 0.96. These are the results in Table 3. Combining more than three levels of HSC features is not effective as the purity is reduced. This is due to there are a lot of redundant information in the combined HSC features.

Table 4: Clustering results (purity)

	Spatial Pooling sizes				**Datasets**			
	1x1	3x5	3x7	5x7	HME_ Answer_1	HME_ Answer_2	CROHME 2016	CROHME 2019
Classification features	✓				**0.88**	0.95	0.92	0.86
Single-level HSC features		✓			0.75	0.93	0.96	0.93
			✓		0.73	0.92	0.97	0.92
				✓	0.70	0.91	0.96	0.92
Multi-level HSC features	✓	✓			0.85	0.98	**0.99**	0.95
	✓		✓		0.83	**0.98**	0.98	**0.96**
	✓			✓	0.80	0.97	0.98	0.93
	✓	✓	✓		0.80	0.97	0.98	0.96
	✓		✓	✓	0.79	0.97	0.98	0.93
	✓	✓	✓	✓	0.77	0.97	0.98	0.95

5.6. Effect of attentive pooling

We measure the clustering performance on CROHME 2019 of the network which is trained with GAtP, GMP and GAP, respectively. The clustering performance in purity is shown in Table 5. The network with GAtP yields slightly better purity than GMP and outperform GAP by a large margin when clustering with large-sized single-level HSC features. For clustering with multi-level HSC features, the network with GAtP also yields better results than GMP and GAP. The results suggest that GAtP produce better classification and localization than GMP and GAP. The network with GAP yields low purity of 0.28 when only classification features are used. Although the poor performance of classification features, HSC features with localization information still produce high purity of over 0.87 for clustering.

Table 5: Effect of global pooling methods for clustering (purity) on CROHME 2019

	Spatial Pooling sizes				**Global Pooling methods**		
	1x1	3x5	3x7	5x7	GAtP	GMP	GAP
Classification features	✓				0.86	0.82	0.28
Single-level HSC features		✓			0.93	0.93	0.74
			✓		0.92	0.92	0.77
				✓	0.92	0.91	**0.81**
Multi-level HSC features	✓	✓			0.95	0.94	0.76
	✓		✓		**0.96**	**0.95**	0.80
	✓			✓	0.93	0.93	0.80

Journal Pre-proof

Journal Pre-proof

We analyze the classification prediction of the networks on CROHME 2019 by mean average precision (mAP). First, the per-class performance is measured using average precision by calculating the area under the precision-recall curve. The precision and recall of classification performance are for evaluating if a class exists in the HMEs or not. The average precision across all classes is summarized to calculate mAP. The mAP by the networks with GAtP, GMP, and GAP is 0.93, 0.92, and 0.17, respectively. The results confirm that the network with GAtP yields better classification performance than GMP and GAP.

5.7. Effect of multi scale feature extraction

We make the experiments to confirm the effectiveness of multi-scale feature extraction. We train the proposed networks to obtain the HSC feature by 1-scale (original input), 2-scale (original input, 1/2 scaled input) and 3-scale (original input, 1/2 scaled input, 1/4 scaled input). The results of clustering on CROHME 2016 and CROHME 2019 are presented in Table 6. The result confirms the effectiveness of multi-scale HSC features. HSC feature with 3-scale achieves the highest purity as compare with 1-scale or 2-scale HSC feature. The 2-scale HSC features also yield high purity of 0.95 on both the datasets. This suggests that 2-scale features could cover almost the scale of symbols in HMEs.

Table 6: Effect of multi-scale HSC feature (purity)

	CROHME 2016	**CROHME 2019**
HSC feature (1-scale)	0.77	0.75
HSC feature (2-scale)	0.95	0.95
HSC feature (3-scale)	**0.99**	**0.96**

5.8. Visualizing class activation maps

The class activation map for an HME is shown in Fig. 5. We extract the output of CNN for multiple scales of input images as presented in Fig. 1. We resize all the outputs to the original image size and obtain the combined multi-scale output by taking maximum values across the output images. The activation map for all the classes is also presented by taking the maximum output for across the classes.

From the visualization, the symbols in different sizes are handled correctly by the three scales of input images. Small size symbols such as ‘n’, ‘j’, ‘x’, ‘a’ in the formula are detected by the first scale, medium size symbols such as ‘(’, ‘)’, ‘=’, ‘P’, ‘sum’ are detected by the second scale, only symbol ‘sum’ is detected by the third scale. The locations of symbols detected by the networks are also correctly predicted. There are cases in which both ‘(’ and ‘)’ are located in the same position. Since ‘(’ and ‘)’ are always presented in pairs in ME, there is no information to discriminate them by using only weakly supervised label.

6. Conclusion

We presented a CNN-based model to learn both class and location representations of symbols for clustering HME. The networks could detect symbols in various scales by a combination of features from a multi-scale CNN. Training the networks with a combination of global attentive pooling and global max pooling improves classification and location prediction. Clustering by multi-level spatial representations extracted from CNN prediction outperforms the online and offline Bag of Features by a large margin. The method also achieves high performance (purity of 0.99) for the CROHME dataset with 36 clusters.

A drawback of the method is that it is not feasible to group the answers with multiple mathematical representations. For example, the formula “1/2” could be represented as “0.5”, “1/2”, “1÷2”, etc. This may need the recognition of the whole formula.

Acknowledgment

This research is partially supported by JSPS KAKENHI Grant Number JP 18K18068.

Conflict of Interest and Authorship Conformation Form

Please check the following as appropriate:

○ All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

○ This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

○ The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript

Journal Pre-proof

Journal Pre-proof

References

[1] G. Brown, J. Bull, M. Pendlebury, Assessing student learning in higher education, Routledge, 1997.
[2] J. Bull, C. McKenna, C. McKenna, A Blueprint for Computer-Assisted Assessment, Routledge, 2003. https://doi.org/10.4324/9780203464687.
[3] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, L. Dai, Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition, Pattern Recognit. 71 (2017) 196–206. https://doi.org/10.1016/J.PATCOG.2017.06.017.
[4] S. Basu, C. Jacobs, L. Vanderwende, Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading, Trans. ACL. (2013).
[5] H.Q. Ung, K.T.M. Vu, A.D. Le, C.T. Nguyen, M. Nakagawa, Bag-of-features for clustering online handwritten mathematical expressions, in: ICPRAI, Concordia, 2018: pp. 127–132.
[6] K.T.M. Vu, H.Q. Ung, C.T. Nguyen, M. Nakagawa, Clustering Offline Handwritten Mathematical Answers for Computer-Assisted Marking, in: ICPRAI, Concordia, 2018: pp. 121–126.
[7] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: ICML'16 Proc. 33rd Int. Conf. Int. Conf. Mach. Learn., New York, 2016: pp. 478–487.
[8] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep Clustering for Unsupervised Learning of Visual Features, in: ECCV, 2018.
[9] S.J. Rao, Y. Wang, G.W. Cottrell, A Deep Siamese Neural Network Learns the Human-Perceived Similarity Structure of Facial Expressions Without Explicit Categories, in: CogSci, 2016.
[10] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, Deep Adaptive Image Clustering, in: 2017 IEEE Int. Conf. Comput. Vis., IEEE, 2017: pp. 5880–5888. https://doi.org/10.1109/ICCV.2017.626.
[11] R.R. Salakhutdinov, G.E. Hinton, Reducing the Dimensionality of Data with Neural Networks, Science (80-.). 313 (2006) 504–507. https://doi.org/10.1126/science.1127647.
[12] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: 2015 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2015: pp. 5325–5334. https://doi.org/10.1109/CVPR.2015.7299170.
[13] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Is object localization for free? - Weakly-supervised learning with convolutional neural networks, in: 2015 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2015: pp. 685–694. https://doi.org/10.1109/CVPR.2015.7298668.
[14] D. Yoo, S. Park, J.Y. Lee, K. Inso, Multi-scale pyramid pooling for deep convolutional representation, in: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., 2015. https://doi.org/10.1109/CVPRW.2015.7301274.
[15] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Trans. Pattern Anal. Mach. Intell. (2017). https://doi.org/10.1109/TPAMI.2016.2577031.
[16] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. 28th IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2015: pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.
[17] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning Deep Features for Discriminative Localization, in: 2016 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2016: pp. 2921–2929. https://doi.org/10.1109/CVPR.2016.319.
[18] K. He, X. Zhang, S. Ren, J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2015) 1904–1916.
[19] H. Mouchere, C. Viard-Gaudin, R. Zanibbi, U. Garain, ICFHR2016 CROHME: Competition on Recognition of Online Handwritten Mathematical Expressions, in: 2016 15th Int. Conf. Front. Handwrit. Recognit., IEEE, 2016: pp. 607–612. https://doi.org/10.1109/ICFHR.2016.0116.
[20] M. Mahdavi, R. Zanibbi, H. Mouchère, ICDAR 2019 CROHME + TFD : Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection, in: 15th IAPR Int. Conf. Doc. Anal. Recognit., Sydney, 2019.

Journal Pre-proof

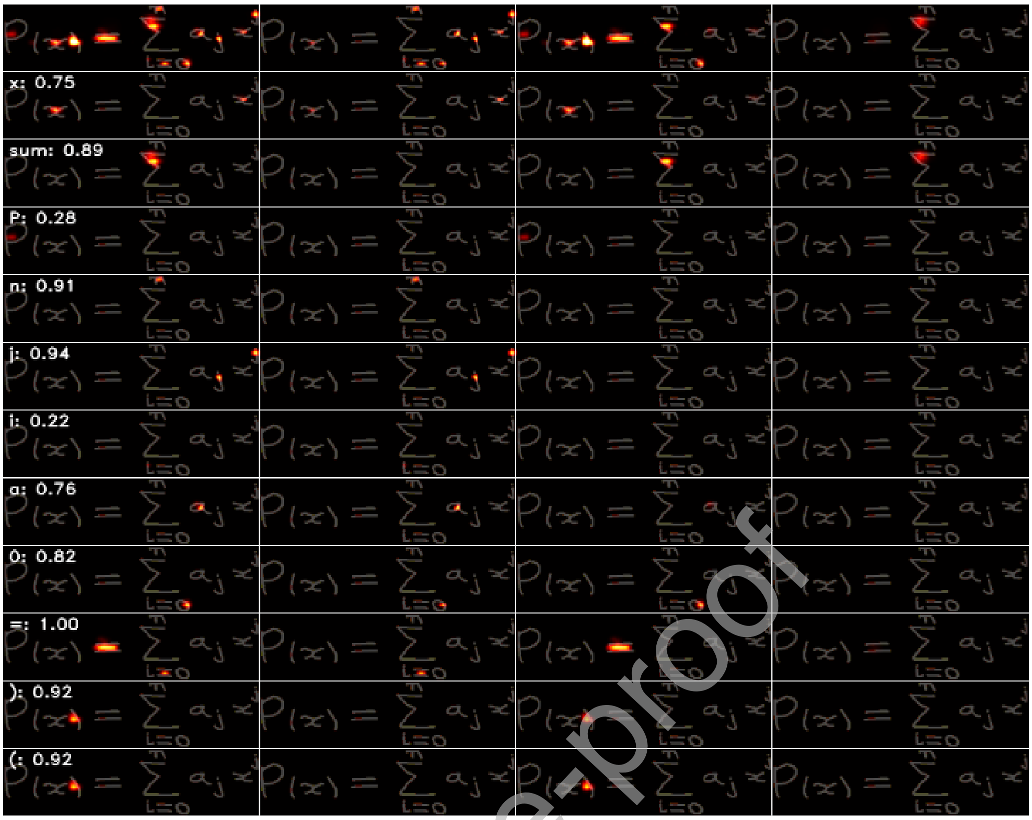


Figure 5. CAMs by multi-scale CNN with attentive pooling.
The columns from left to right are the aggregated of multi-scale CAMs with class label and predictive score, the first, the second and the third level CAMs, respectively. The first row shows the combination CAMs results for all the classes.

Journal Pre-proof