Memory Augmented Convolutional Neural Network and Its Application in Bioimages

Weiping Ding [a*], Yurui Ming [b], Yu-Kai Wang [b], and Chin-Teng Lin [b]

[a] School of Information Science and Technology, Nantong University, Nantong 226019, China

[b] School of Computer Science, Centre for Artificial Intelligence, University of Technology Sydney, NSW 2007 Australia

***Abstract*—The long short-term memory (LSTM) network underpins many achievements and breakthroughs especially in natural language processing fields. Essentially, it is endowed with certain memory capabilities that can boost the performance. Drawing inspiration from it, this paper proposes a memory augmented convolutional neural network (MACNN) utilizing self-organizing maps (SOM) as the memory module. First, we depict the potential challenge of applying solely a convolutional neural network (CNN) and highlight the advantage of augmenting SOM as memory for better network generalization. Based on this inspiration, a corresponding network architecture incorporating memory is disserted. It instantiates the distributed knowledge representation philosophy and tactically combines the SOM and CNN together. For the experiment, we test the proposed network on various datasets where conventionally only CNN is applied. It reveals that based on the same CNN structure and by incorporating memory, better results can be achieved, especially for datasets consisting of bioimages which are challenging for other models. We also illustrate the learned representations to interpret the SOM behavior and to comprehend the achieved results.**

Keywords: **Deep Learning (DL), Memory Augmented Convolutional Neural Network (MACNN), Self-organizing Maps (SOM), Memory Network (MN)**

I. Introduction

THE prosperity of deep learning (DL) triggers the retrospection of research on brain-inspired computations. For example, tremendous achievements buffeting natural language processing (NLP) in recent years are regarded as being attributed to the constructions of large long short-term memory (LSTM) networks. It encourages Google and Facebook to build more sophisticated neural networks integrating memory to solve more difficult tasks [1][2]. However, certain DL applications are not targeting sequential data, such as image classification, face recognition, cancer prediction, to name a few. These applications are based on variations in convolutional neural network (CNN) architectures, which are generally regarded as computation-oriented. Although the network interacts with all the training samples during the training process, this process is still regarded as being mainly for learning optimal filters for the targeted problem instead of learning to memorize. Hence, the impressive performance of CNN relies on the features being effectively extracted from input data via these optimal filters [3]. Alternatively, it means that the prediction is mainly based on the computation carried out on the current input sample, while the relations with historical samples are quite vague.

For a group of images containing the same object, e.g., dogs, these images share many features common to dog species. For a given CNN, categorization of the content in a specific image mainly harnesses the knowledge retrieved from the current image via applying the learned filters. Instead, humans tend to make the judgements comprehensively via cognitive processes such as familiarity or even recall if necessary, all of which involve the memory system [4]. The above consideration encourages us to devise a memory part to couple with the CNN to boost the performance.

Another drive for considering new computation paradigm involving memory originates from a specific application of CNN, i.e., bioimage classification. Unlike the datasets such as ImageNet or COCO for general-purpose application of image classification [5][6], bioimage datasets either from clinical practice or academic research tend to be small ones with imbalanced samples, all of which post challenge for effectively utilizing CNN for classification. For example, limited samples increase the risk of overfitting for deep neural network; however, shallow network may fail to effectively extract suitable features for proper classification. However, learning from a few samples to generalize on new data in sufficient accuracy is natural for humans [7]; and learning capability is regarded as interleaving with the memory system [8]. Hence, it is promising to propose new network models from a memory augmentation perspective.

But due to the limited understanding of the human memory system, inspirations from it may be not so intuitive and difficult to manage. Take Google's proposition, i.e., the differentiable neural computer (DNC) as an example [1]. It tries to establish an enriched computation model by integrating a large memory, and two extra LSTM networks have to be tailored for the memory mechanism. Although with succeeded applications to complicated problems, however, following the architecture of

* Corresponding author: Weiping Ding

E-mail address: dwp9988@163.com (W. Ding), yrming@gmail.com (Y. Ming), Yukai.Wang@uts.edu.au (Y.-K. Wang), Chin-Teng.Lin@uts.edu.au (C.-T. Lin).

modern computers avoidably adds complexity to the whole architecture and makes modifications and extensions to the memory module extremely difficult.

In fact, the idea of utilizing memory as part of computation has its own history. Kant proposed the concepts of memory elements and the interconnections between them and their functionalities in [9]. John Hopfield introduced Hopfield network, working in a content-addressable or associative manner with binary threshold nodes in [10]. In this paper, we propose using the self-organizing map (SOM) invented by Finnish professor Teuvo Kohonen as the memory module to fertilize computation [11]. SOM is based on the biological models of neural systems [12][13]; thus, it can capture the intuitive essence of memory system and meanwhile maintain its simplicity.

In this paper, we propose a neural network architecture named MACNN (memory augmented convolutional neural network) and evaluate it on datasets which conventionally only CNN are applied. We structure the subsequent sections as with our contributions. We first introduce the background knowledge of SOM to reveal its suitability as memory module, followed by the detailed explanation of the overall MACNN architecture. Then we experiment on different datasets to show the practicability and supremacy of our designed architecture via benchmarked results. Simultaneously, we also illustrate the learned memory representation to interpret the performance of our proposed network. At last, we discuss the difference between LSTM and SOM and anticipate future works to improve MACNN.

II. SOM Recap

SOM used to be a popular neural network model and could find its various applications in numerous fields [14]. It belongs to the category of competitive learning networks and implements the winner-take-all learning strategy [11]. The arrangement of neurons plus the unsupervised learning paradigm enable it to represent the features intrinsically to the input sample space in a topological and suitable manner automatically. The updating process, which drives the best-matching unit (BMU) and its neighboring neurons towards the value of the newly input sample, endows SOM the potential of tracing all past sample values, if the lattice of SOM is large enough. Nevertheless, averaging the samples belonging to the same category and differentiating the samples belonging to different categories are critical for learning. This behavior can encourage the features common to all samples to emerge, which is critical for model performance. As mentioned above, for CNN, where the learned filters act on the input data for final classification or regression, the process can be considered computation-oriented. Instead for SOM, the weights are aggregating on all input data for feature abstraction and hence can be considered memory-oriented. This perspective makes SOM apt for memory network such as in LAMSTAR [15]. Other utilizations of SOM such as visualization of sample space can be found in [16][17].

To facilitate the understanding of our designed network and its configurations in the following sections, the training process of SOM is particularly addressed below. The aspects that can be traded off for better performance, such as learning rate manipulation, neighbourhood calculation, etc., are also highlighted.

1. Randomly initialize the weights of neurons in the lattice, which is of dimensions $H \times W$. H denotes the height, and W denotes the width of the lattice.
2. Choose an input sample from the training set and present it to the lattice. This can be done in a batched way.
3. Calculate the Euclidean distance between the input sample $\vec{v} = (v_1, v_2, \cdots, v_n)$ and the weight vector $\vec{w} = (w_1, w_2, \cdots, w_n)$ to each neuron in the lattice, with $\|u - v\|_2 = \sqrt{\sum_{i=1}^{n}(v_i - w_i)^2}$. The node with the least distance is selected and named BMU.
4. Decide the neighbourhood of BMU for the current step. The neighbourhood is usually a disc mask, with an initial radius σ_0, and shrinks according to formula such as:

$$\sigma(t) = \sigma_0 \exp(-t/\lambda) \tag{1}$$

   λ is regarded as the time constant, where $\lambda = T/\ln \sigma_0$, with T being the total number of learning steps or training iterations.
   A boundary-condition analysis reveals that

$$\sigma(T) = \sigma_0 \exp(-T/(T/\ln \sigma_0)) = 1 \tag{2}$$

   It is within the expectation that update at the last step is confined only to the BMU itself.
5. Adjust weight vectors for the neighbouring neurons. Determine the learning rate $\eta(t)$ for the current step according to (3):

$$\eta(t) = \eta_0 \exp(-t/\lambda) \tag{3}$$

   Calculate the Euclidean distance D_i between the BMU and neuron i based on their locations in the lattice, and the corresponding distance influence factor $\Theta_i(t)$ as in (4):

$$\Theta_i(t) = \exp(-D_i^2/2\sigma^2(t)) \tag{4}$$

   The adjustment for neuron i is in accordance with (5):

$$\vec{w}_i(t+1) = \vec{w}_i(t) + \Theta_i(t)\eta(t)\big(\vec{v}(t) - \vec{w}_i(t)\big) \tag{5}$$

6. Repeat 2-5 for T steps.

Notably, in the above procedures, most formulas are not unique but can have alternatives, providing certain paradigms can be satisfied. For example, the radius of the current BMU neighbourhood can also be determined as in (6):

$$\sigma(t) = \sigma_0 - (\sigma_0 - 1)\, t/T \tag{6}$$

$\eta(t)$ can be alternatively updated as in (7):

$$\eta(t) = \eta_0(1 - (t-1)/T) \tag{7}$$

Another method for amending $\Theta(t)$ is according to (8):

$$\Theta_i(t) = \max\{(\sigma^2(t) - D_i^2), 0\}/\sigma^2(t) \tag{8}$$

Fig. 1 shows different neighborhood-modulated learning rates for a given step. Note that in Fig. 1(A), a truncation exists beyond the current radius. The choice between these formulas inevitably increases the parameter space for exploration for fine-tuning. However, on the other end it provides the

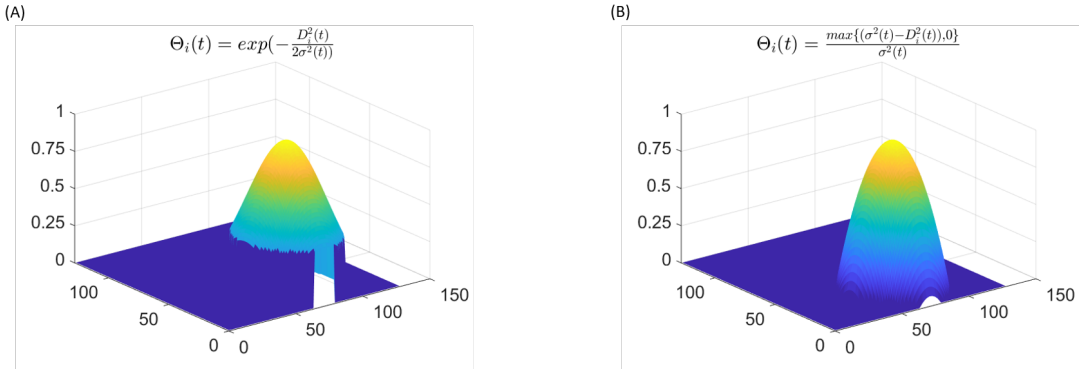

Fig. 1. Learning rate modulated by different neighborhood functions.

plausibility to adopt different formulas for different problems to achieve the best performance.

III. Network Architecture

In this work, we design a hybrid architecture that jointly combines computation and memory by incorporating a SOM into the CNN. The general structure of SOM is shown in Fig. 2(A). Each component of the input vector connects to a neuron that is arranged in a two-dimensional lattice. Assuming that the dimension of the input sample is N, let i denote the i^{th} input component and j denote the j^{th} neuron; then, the corresponding weights are $[w_{i_1}^j, w_{i_2}^j, \cdots, w_{i_N}^j]$.

To conceive a memory module based on SOM, a change in view is needed. As shown in Fig. 2(B), each neuron can be thought of storing a vector that has the same dimension as the input sample, and the weights linking the input and neurons are always of fixed value 1. In fact, at the initial stage of neural network development, the plasticity of synapses for learning is not well understood. Therefore, the weights are always set fixed, and only neurons modulate the relied signals [18]. However, to consider from the viewpoint of Fig. 2(B) can ease the conceivable barrier when adopting SOM as memory part of the network. Viewing the structure of SOM in either Fig. 2(A) or Fig. 2(B) takes the same learning or updating mechanism.

The next consideration is the content to be stored in the neurons. In applications, the data samples are usually of high dimensions, and to directly store the raw data is inapplicable. An alternative is to store some transformed features of lower dimensions, as shown in Fig. 2(C). The criterion is that the transform should be capable of extracting suitable features from the original data. Based on these reflections, the designed neural network architecture is shown in Fig. 3.

In Fig. 3, the overall architecture is a combination of CNN and SOM, abbreviated as MACNN (memory augmented convolutional neural network). We consider only the classification problem in this work, and each class is accommodated by a SOM block. The memory or SOM module consists of individual SOM blocks. To articulate the original input into features for memorization and refinement in the SOM blocks, a CNN is introduced to act as the transform based on the idea of transfer learning. To make the CNN suitable for extracting features, a pre-training stage (or circuitry) is introduced. The pre-training stage is the same as the general application of CNN. In detail, the CNN is trained first, and then the weights are fixed and transferred into the second stage (training) to extract the features for further manipulation. Subsequently, the extracted features and SOM interact to achieve the final computation goal, namely, classification.

Depending on whether it is the training stage or test stage, the inputs are routed into the SOM in different ways. For training, the input (extracted feature) is gated by the ground-truth label. Assuming there are N classes, let x denote the input

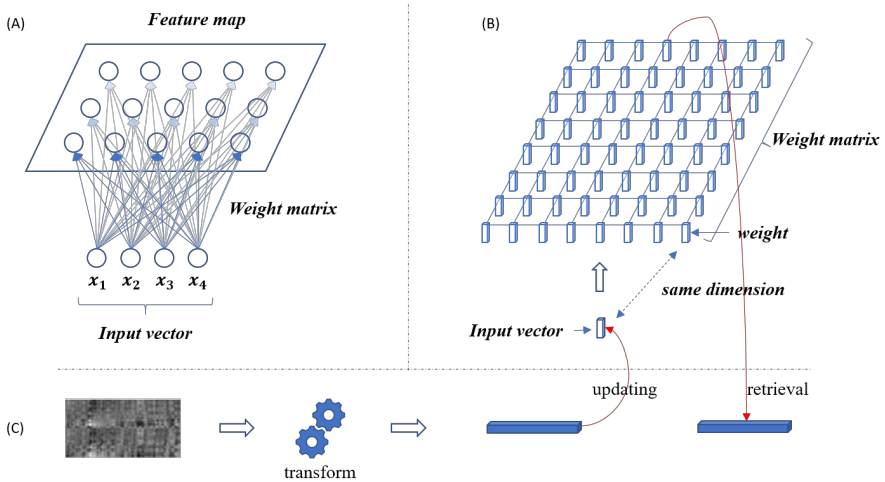

Fig. 2. SOM structure: (A) Illustration from the conceptual perspective; (B) Illustration from the pragmatic perspective (C) Dimensional reduction of EEG data and interaction with the SOM module.

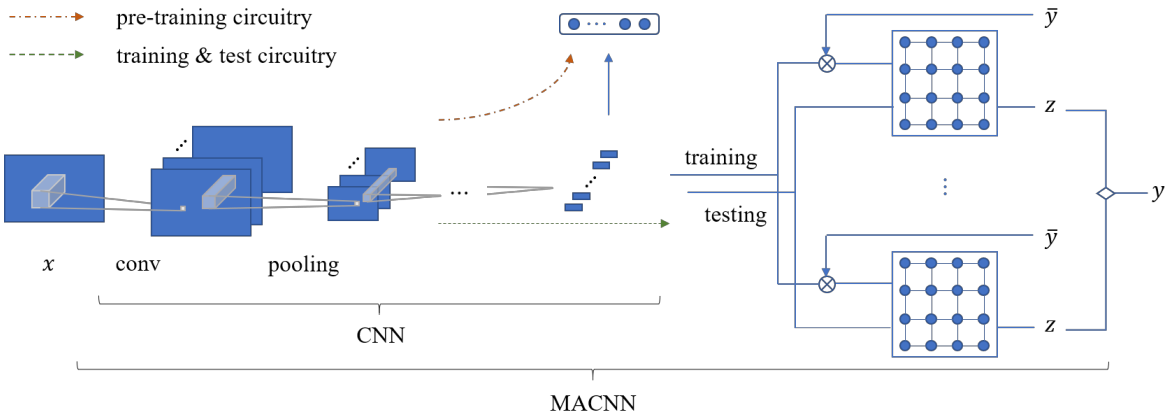

Fig. 3. The overall network architecture.

to the SOM module, and $g(x) = \{g_1(x), g_2(x), \cdots, g_N(x)\}$ the denotes the gate function, where $g_i(x) = 1$ if $x \in C_i$ and $g_i(x) = N/A$ (means unavailable) otherwise. The input to the i^{th} SOM block is $x \cdot g_i(x)$. For example, suppose the current input belongs to the first class C_1. Then, the input goes only to the first SOM block and updates the block according to the SOM learning method. All other SOM blocks are inactive for the current input. For the test stage, the input goes to every SOM block for computation.

The motivation for the design of our network architecture in Fig. 3 draws inspiration from the following biological facts closely related to neuroscience. First, the memory system is a distributed system, and knowledge are stored in different areas of the neocortex [19]. Second, long-term memory induces and depends on the modulation of synapse plasticity [20]. For a classification problem, different categories usually represent different concepts. The intention for incorporating separative SOM blocks is to cater for the need of representing different categories in a distributive manner. The gating mechanism is devised to resolve the difficulty of the learning process. SOM favors the fact that learning roots in synapse plasticity alteration. If the input is routed into all SOM blocks during training, the one corresponding to the correct label shall update the weights (or the content stored in neurons) towards the favoring direction. The rest should be updated in some adversarial directions. The problem is in high-dimensional space, it is difficult to define the adversarial direction. Therefore, an alternative is to let all SOM blocks update only towards the favoring direction. It means that only the SOM block corresponding to the correct label learns per input sample, with the others remaining inactive. Other methods for training might exist, but this method is simple and computationally efficient.

For testing, all weights involved in the whole architecture are frozen. The input is routed into every SOM block directly to find the BMU. Then the same metrics (usually the Euclidean distances) between the input and all BMU candidates are calculated, among which the minimum one is located. The index of the BMU with minimum distance is regarded as the corresponding category to which the input belongs.

The postulation underpinning the above procedure is that during updating, the SOM block corresponding to a certain category to which some samples belong, is always learning from these samples. Different features, which are supposed to be captured by different SOM blocks, differentiate samples from each other. These learned features are not from one sample but aggregate from all historical samples and get memorized by the SOM. This process resembles the memory-forming procedure, i.e., perception, consolidation, etc. Compared with the vanilla CNN, it is believed that this memory-incorporating model can exhibit better performance.

The algorithm for utilizing the proposed network for classification is summarized as follows:

Pre-training:
Input: image samples and the corresponding labels from the training set
Output: trained weights of the CNN
Procedures:
1: Train the CNN via backpropagation
2: Save the trained weights of the CNN

Training:
Input: image samples and the corresponding labels from the training set
Output: trained weights of the overall network
Procedures:
1: Load the saved weights of CNN
2: Repeat the following steps until exceeding the number of iterations:
3: Fetch an image and its label from the training set
4: Feed the image into the CNN
5: Update the weights of SOM block that corresponds to the label category using feature from the CNN
6: Save the weights of the overall network

Test:
Input: image samples and the corresponding labels from test set
Output: the predicted labels and prediction accuracy
Procedures:
1: Load the saved weights of the network
2: Repeat the following steps until exhausting the test set:
3: Fetch an image and the corresponding label
4: Feed the image into the network
5: Find the BMU of each SOM block according to the feature of the image retrieved by the CNN
6: Compute the Euclidean distances of these BMUs with the input image
7: Record the index the minimal distance
8: Find the corresponding category represented by the index and compare it with the ground-truth label

---

IV. Experiments

Three datasets are utilized to assess the practicability of the proposed network architecture. First, the clean and well-

TABLE I
NETWORK CONFIGURATION FOR MNIST EXPERIMENT

Network (CNN)					Stages		
No.	Layer	#Filter	Kernel	Act.	Pre-training	Training	Test
1	Conv2D	16	5x5	ReLU	x*	x	x
	AvgPool	-	2x2	-	x	x	x
2	Conv2D	32	5x5	ReLU	x	x	x
	AvgPool	-	2x2	-	x	x	x
3	FC	32	-	ReLU	x	x	x
4	FC	10	-	Softmax	x		
Network (SOM)							
5	Height	Width	Depth	#Blocks		x	x
	64	64	32	10			

*x indicates the presence of the corresponding layer

understood MNIST dataset is used to check the feasibility. Once the proposed model does not work, the use of clean MNIST data can narrow down the reason to the network structure rather than the dataset. The second dataset consists of breast cancer histopathology image patches, and the third dataset is an EEG dataset captured from a visual oddball task. These two imbalance bioimage datasets present high variance across samples belonging to the same category, a major challenge for most ML algorithms. Meanwhile, these datasets concern with clinical practice or neurophysiological theory, and to improve algorithms in targeting problems in these domains is a meaningful work.

A. MNIST Dataset

The first dataset for experimenting is the traditional MNIST dataset, which is a derivation of a larger handwritten digit set from NIST [21]. It contains 60,000 training samples and 10,000 test samples, which are normalized digit images with dimensions of 28 by 28 pixels. Newly constructed networks are usually run against it, because on one hand minimal efforts are required to pre-process the data, and on the other hand, the performances of variational CNN structures are well understood, a precursor for the practicability of the proposed model for other datasets.

The implementation of our proposed network architecture is based on TensorFlow, the DL library open-sourced by Google [22]. Therefore, it directly utilizes the API, which integrates the dataset internally. Because the major task in this paper is not to seek the sole CNN structure for SOTA performance but the supremacy of augmenting memory to a given CNN, hence a vanilla CNN structure is chosen here. The configuration of the network structure is given in TABLE I. Note that some validations have already been performed to determine some hyper-parameters. The number of neurons for the first FC layer is not as large as for other structures. This is to mitigate the computation in SOM module. For each digit, the length of the feature vector used to represent it is still unclear.

To objectively assess the performance of the proposed network architecture, the entire procedure is repeated 5 times. For each pre-training CNN, the transferred weights of the CNN are used 3 times to train the MACNN, and the maximal test accuracy is taken as the performance indicator for the current fold. The final comparison is between the maximal prediction accuracies of the CNN and MACNN.

With the above description, taking a learning rate of 0.001 with a batch number of 64, we pre-train the model (or train the sole CNN) for 10,000 iterations. To stabilize the pre-training process, the learning rate is decayed by a factor of 0.96 for every 100 iterations. After pre-training, with a learning rate of 0.2 for the SOM module, we train the MACNN for 10,000 iterations. Some of the test data are used to monitor both the pre-training and training processes, and no overfitting is observed. After training both the CNN and MACNN, the statistics for prediction on the whole test set are shown as in TABLE II.

TABLE II
TEST STATISTICS FOR MNIST EXPERIMENT

	CNN (↓)	MACNN (→)			Maximum
No.		1	2	3	
1	0.9881	0.9893	0.9892	0.9896	**0.9896**
2	0.9893	0.9895	0.9893	0.9898	**0.9898**
3	**0.9895**	0.9894	0.9889	0.9890	0.9894
4	0.9913	0.9922	0.9912	0.9918	**0.9922**
5	0.9897	0.9901	0.9895	0.9898	**0.9901**
Maximum	0.9913	-	-	-	**0.9922**

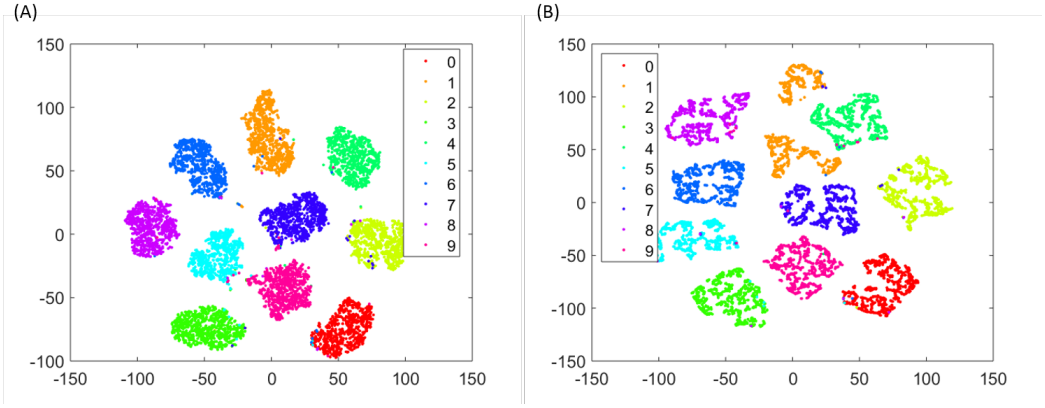

Fig. 4. Illustration of features of MNIST test samples from CNN and MACNN respectively via t-SNE (A) The CNN case; (B) The MACNN case.

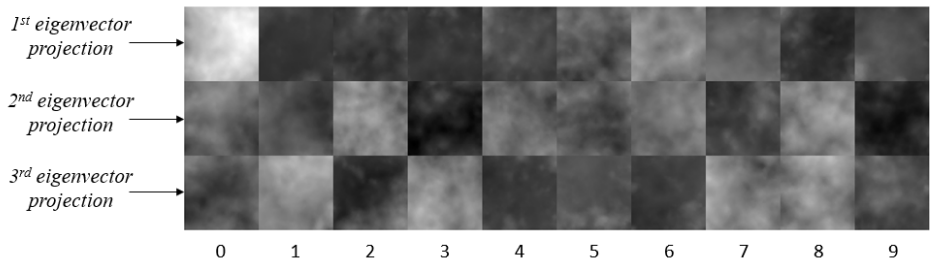

Fig. 5. SOM contents projection along a specific direction (eigenvectors of SOM block 0 in this case).

TABLE II shows that the accuracy of the MACNN is better in most cases, although the improvement is marginal. However, a reflection of our method still indicates the improvement is an interesting achievement. The scenario for transfer learning usually involves a computational model that is developed for task A is reused for task B. Here, the features extracted admittedly suitable for the CNN are reused to train MACNN. and a better result is still achieved, which manifests the promise for incorporating memory. By future work on a more appropriate transform function as in Fig. 2(C), new prospects of the proposed model can be expected.

As mentioned above, the transform crafted via self-transfer learning might overshadow the advantage of our proposed network architecture. However, based on our design principle, two aspects are still worthy of investigation.

The first relates to the characteristics of features from the respective CNN and MACNN regarding the same test samples. In this respect, we propagate all test samples through both the CNN and MACNN, then apply t-SNE to visualize the distribution of the corresponding samples [23]. From Fig. 4, it can be noticed that features from MACNN are more diverse than those from the CNN. This diversity or higher variation usually confers the advantage of better generalization. For the MNIST dataset, this means that the SOM module can tolerate larger deformations of handwritten digits belonging to the same category.

The second is that each SOM block learns features towards a unique direction. This claim is verified in Fig. 5. To carry this out, we first apply principal component analysis (PCA) to the content of one SOM block (here the one corresponding to digit 0), then sort the eigenvectors according to the magnitude of the eigenvalues. Next, we project each SOM block along these eigenvectors. Fig. 5 shows the case of projections via the first three eigenvectors. The brightness of the image indicates the component magnitude. It is manifest that knowledge learned by SOM block 0 lies along a special direction distinct from the remains. From the first row of images, the image patch corresponding to 0 is the brightest, while the others dim from intermediate to deep dark.

B. IDC Dataset

In this experiment, the dataset investigated is an image dataset for a subtype of breast cancer [24]. It consists of patches of images scanned from invasive ductal carcinoma (IDC) tissue regions [25]. The diagnosis of malignant tissue traditionally relies on screening of the large digital histopathology image by pathologists to distinguish it from the swathes of benign areas. This process is time-consuming and challenging and can be error-prone for less-experienced technicians. Considering the outstanding achievements in utilizing DNN for image processing tasks such as classification and segmentation, applying DL to detect IDC to assist malignant tumour diagnosis is of great significance.

The dataset consists of 277,524 RGB digital image patches with the resolution of 50x50 pixels. These patches were extracted from H&E-stained breast histopathology images of 162 women who were diagnosed with IDC. The ground-truth labels for IDC regions are annotated via manual delineation of the cancer region by an expert pathologist. Direct operations on the original digitized histopathology image slides are intractable due to the huge size, which leads to the alternative patch-based analysis, i.e., the whole image is sliced into non-overlapping patches. In general, the breast tissue contains many cells, but only some of them are cancerous. For simplicity, patches containing cells that are characteristic of IDC are labelled "1", and “0” otherwise. However, this labelling strategy means that the degrees of IDC are not differentiable between patches labelled “1”. More information about the data can be found in [24]. An illustration of the respective positive and negative case is shown in Fig. 6.

To avoid redoing the experiment in [24] [26] and make direct comparison, the treatment of data is exactly as in these previous work. The patient ID files are retrieved from [27] for training, validation and testing. The ratio between negative samples and positive samples is approximately 1:3. It is common that the number of negative samples surpasses the number of positive

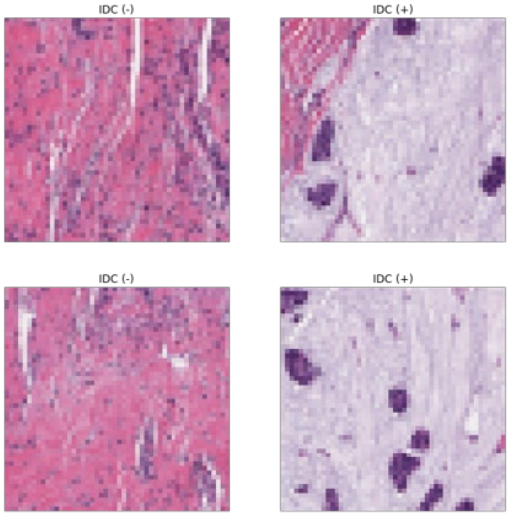

Fig. 6. Sample patch images of non-IDC(-) and IDC(+)

TABLE III
TEST STATISTICS FOR IDC EXPERIMENT

		CNN (↓)	MACNN (→)			
No.			1	2	3	Max.
1	Accuracy	0.8365	0.8419	0.8426	0.8468	**0.8468**
	F1 Score	0.7525	0.7563	0.7568	0.7611	**0.7611**
2	Accuracy	0.8372	0.8448	0.8446	0.8516	**0.8516**
	F1 Score	0.7505	0.7541	0.7536	0.7587	**0.7587**
3	Accuracy	0.8362	0.8492	0.8516	0.8560	**0.8560**
	F1 Score	0.7500	0.7537	0.7559	0.7584	**0.7584**
4	Accuracy	0.8406	0.8619	0.856	0.8474	**0.8619**
	F1 Score	0.7554	0.7748	0.7697	0.7619	**0.7748**
5	Accuracy	0.8388	0.8516	0.8454	0.8475	**0.8516**
	F1 Score	0.7549	0.7677	0.7613	0.7643	**0.7677**
Max.	Accuracy	0.8406	-	-	-	**0.8619**
	F1 Score	0.7554	-	-	-	**0.7748**

samples for bioimage datasets. However, the imbalance might incur misbehaviour of the proposed network if only the accuracy is considered. To prevent the perversion during the training process, negative samples are resampled to match the positive samples. For example, suppose that the batch number is N; $N/2$ negative samples are resampled from the total number of negative samples to blend with $N/2$ positive samples for each training iteration. There exist other alternatives to conquer the imbalance problem, but this one is simple and computationally efficient.

There is no essential difference between this experiment with the MNIST experiment; thus, the configuration of the network is identical to the MNIST experiment. In addition, distinguishing between benign and malignant tissues is a binary classification problem; hence, the number of SOM blocks is 2. The training and testing processes are the same as the MNIST experiment. We notice the tendency of overfitting in the training process; therefore, the learning rate of SOM is set to 0.001, with the neighbourhood modulation scheme governed by (7). An unconditional early stop is also adopted here (we set the iteration number to 10,000 to calculate parameters during SOM updating; however, we stop the training at 5,000). The testing statistics are shown in TABLE III.

Notably, the prediction accuracy is usually inappropriate for imbalance data, because the training process can drive the model to misclassify the minority and still retain a high accuracy. In this circumstance, F1 score is usually taken as the performance indicator [28]. TABLE III clearly reveals that the results achieved by MACNN are better than CNN for all cases. It not only presents higher accuracies, but also displays greater F1 scores. This outcome is another good indicator of the necessity of augmenting memory for CNN.

We also compare our results with the achievements in [24][26] in TABLE IV. It is obvious that the performance in our work is the best among all. To achieve up-to-date results, previous works applied several tricks to the dataset and with more complicated network structures. However, the outcome in our work is based only on a vanilla CNN plus a memory module, without any trickery and cunning involved. We believe that by adopting more tactics and continuing exploration of SOM, the results can be further improved.

TABLE IV
TEST STATISTICS FOR IDC EXPERIMENT VIA DIFFERENT MODELS

Method	Accuracy (%)	F1-Score (%)
CNN-Based [24]	84.23	71.80
AlexNet-Based [26]	84.68	76.48
MACNN	**86.19**	**77.48**

C. EEG Dataset

The dataset used in this experiment is an EEG dataset captured during a visual oddball task [29]. Neuroscience is regarded as having a deep connection with DL; the latter tends to draw inspiration from the former. Among the auxiliaries of brain research, EEG is no doubt the most convenient and economic brain imaging technology to inspect neural activities during a specific cognitive process [30]. For most cases, EEG signals captured during certain physiological or cognitive process are serial data, which can be analyzed by constructing an RNN with LSTM cell. But it is not always the case, for instance, the P300 phenomenon in typical oddball or speller tasks [31]. The imminent EEG amplitude time-locked to the appearance of the visual stimulus (target) has a rather limited effective duration (0.7 seconds at most), and RNN is difficult to apply in this circumstance. Additionally, EEG signals are complicated bio-signals due to high intra-subject and cross-subject variance and the low signal-to-noise ratio (SNR). Those factors render sole application of CNN in this situation not as fruitful as in other cases. Hence, our proposed model is very appealing for this case.

The oddball task is based on the neurophysiological activity called visual evoked potential (VEP) [32]. The experiment is designed as image stimuli, which are of two categories (target vs non-target), presented to subjects at a rate of approximately 0.5 Hz (one image approximately every two seconds) to arouse their responses. For this experiment, the targets are enemy combatants (34 in total) while non-targets are U.S. soldiers (236 in total). The subjects were presented with all the 270 images. During the experiment, they were instructed to identify each image as being a target or not with a unique button press as quickly but as accurately as possible. The experiments were approved by the U.S. Army Research Laboratory Institutional Review Board (Protocol # 20098-10027), with the consent of subjects complying with the Federal and Army regulations.

There were eighteen subjects participated in the experiments, which last 15 min on average. The wired 64-channel ActiveTwo3 system (sample rate set to 512 Hz) from BioSemi were used to record the EEG signals, which are directly analyzed in the time domain. For pre-processing, the raw EEG data are first rectified via the PREP pipeline [33], and then subjected to the multiple artifact rejection algorithm for ICA-based artifacts removal [34]. Next, the signals are down-sampled to 256 Hz, following segmentation into [0, 1.0] second intervals time-locked to the stimulus onset. To mitigate the effect of outliers, epochs with incorrect button presses are removed. Then, the mean baseline is removed from each channel in each epoch. These procedures lead to a final 382 target trials and 3249 non-target trials.

The placement of the 64 channels is complied with the international 10-20 system. Because VEP originates from the occipital cortex, which is dominantly involved in receiving and

TABLE V
NETWORK CONFIGURATION FOR EEG EXPERIMENT

Network (CNN)					Stages		
No.	Layer	#Filter	Kernel	Act.	Pre-training	Training	Test
1	Conv2D	16	1x5	tanh	x	x	x
	AvgPool	-	1x2	-	x	x	x
2	Conv2D	32	1x5	tanh	x	x	x
	AvgPool	-	1x2	-	x	x	x
3	FC	32	-	tanh	x	x	x
4	FC	2	-	Softmax	x		
Network (SOM)							
0	Height	Width	Depth	Radius		x	x
	128	128	32	8			
1	Height	Width	Depth	Radius		x	x
	32	32	32	8			

interpreting visual signals, signals from 12 channels, which mainly cover the parietal and occipital areas are selected for analysis. As in the IDC experiment, samples of individual classes (target vs non-target) are also imbalance; hence, the resampling strategy used in the IDC experiment is adopted here as well.

However, the imbalance of the EEG dataset (1:9) is more severe than the IDC dataset (1:3). To cater to the appropriate learned knowledge representation, the dimensions of separate SOM blocks are more diverse. For the network configuration in TABLE V, the dimensions of SOM block 0 are quadruple as those of block 1. Furthermore, although the change of viewpoint from Fig. 7(A) to Fig. 7(B) makes the original EEG data no eminent difference from image data, the multi-channel property makes the convolutions along the height under dispute. Thus, the convolution is 1D and only along the width dimension. The training procedure is identical to the previous experiments. Because EEG data tend to display low SNR, the learning rate of SOM is set to 0.01 to cater for the procedural refining process of SOM update. The statistics of the test are shown in TABLE VI.

It is obvious from TABLE VI that the CNN achieves better accuracies among multiple runs. However, although MACNN fails to beat CNN in terms of accuracy, it is superior in terms of the F1 score in most cases. The results indicate that MACNN can be a promising application for imbalanced data. We also compare with the results from [36], which used the same EEG dataset to study different resampling strategies for imbalanced data. The best result of our model in TABLE VII suggests that by combining SOM and CNN together, challenging issues such as imbalanced data can be effectively addressed.

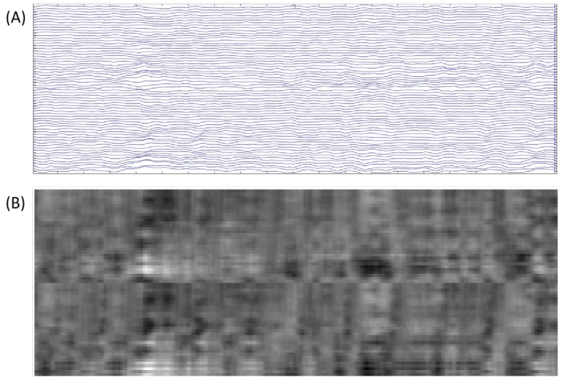

Fig. 7. Viewpoints of EEG from different perspectives. (A) Multi-channel data; (B) Image data.

TABLE VI
TEST STATISTICS FOR EEG EXPERIMENT

		CNN (↓)	MACNN (→)			
No.			1	2	3	Max
1	Accuracy	0.7884	0.7858	0.7720	0.7934	**0.7934**
	F1 Score	0.4509	0.4516	0.4290	0.4569	**0.4569**
2	Accuracy	**0.8702**	0.8551	0.8564	0.8589	0.8589
	F1 Score	0.4550	0.4549	0.4818	0.4766	**0.4818**
3	Accuracy	**0.8513**	0.8387	0.8224	0.8236	0.8387
	F1 Score	0.4271	0.4285	0.4291	0.4308	**0.4308**
4	Accuracy	**0.8488**	0.8274	0.8186	0.8136	0.8274
	F1 Score	**0.4230**	0.4170	0.3999	0.3983	0.4170
5	Accuracy	**0.8299**	0.8148	0.8085	0.8047	0.8148
	F1 Score	0.3349	0.3466	0.3448	0.3459	**0.3466**
Max	Accuracy	**0.8702**	-	-	-	0.8589
	F1 Score	0.4550	-	-	-	**0.4818**

We also illustrate the distributions of features of all test samples extracted by the CNN and MACNN respectively to understand the difficulty of highly accurate classification for both models and then try to understand the behavior of MACNN, which leads to a better result. Fig. 8(A) illustrates the features from the CNN via t-SNE, which demonstrates the separability of target and non-target samples. Even after several neural network operations such as convolution and pooling, the interleaving of features is not effectively resolved into a clear condition. In contrast, in Fig. 8(B), the features are organized in a more categorical way, approximating the intended dichotomic phenomena of the oddball task. However, the intrinsic non-separability of the problem still prohibits achievement of a good prediction; nevertheless, it does not obscure that the situation in Fig. 8(B) is better than that in Fig. 8(A).

TABLE VII
TEST STATISTICS FOR EEG EXPERIMENT VIA DIFFERENT MODELS

Model	NN*	SVM	CNN	MACNN
Accuracy	0.694	0.756	**0.8702**	0.8589
F1 Score	0.309	0.288	0.4550	**0.4818**

*Multi-layer Perceptron

V. DISCUSSION

The significance of human memory in the cognitive process which shapes our daily life is self-evident. It is well-known that the human memory system consists of working memory, short-term memory and long-term memory. Long-term memory is further categorized into explicit memory and implicit memory. How to draw from human memory to design a more intelligent system is admittedly meaningful research in this DL era, just as

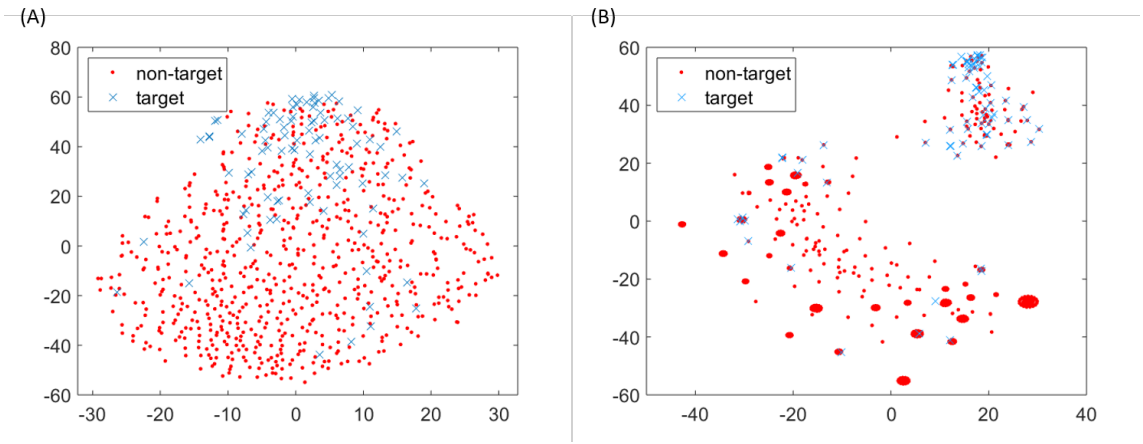

Fig. 8. Illustration of features of MNIST test samples from CNN and MACNN respectively t-SNE (A) CNN; (B) MACNN.

the work that was done in [1].

In this research, we proposed SOM that plays the role of memory in CNN, is quite different from LSTM, at least from the brain-inspired computation perspective. LSTM mainly explores the temporal correlation during recursion when processing the current sample, and the content expires for new input. On the other hand, training an LSTM network is not for storing information, but rather to tune the weights associated with the input gate, forget gate and output gate. Generally, LSTM resembles the working memory.

However, SOM does store information of the samples. A boundary condition analysis can be performed to reveal that in an extreme case, the neighborhood of the BMU is just the unit itself as in section II. Furthermore, if the learning rate is 1, the BMU can be updated to the current sample. Given a sufficiently large lattice, all the input samples can be logged; nevertheless, this is not the way that SOM is intended to be used. Actually, if we think that the current available samples (regardless of training, validation or test) are sampled from a given space, the updating process of SOM can be treated as an interpolation and/or extrapolation of that space, meanwhile, smoothing among the samples. As in Fig. 2(B), each node in the lattice can be viewed as an instantiation of a class with some learned variance, which confers a better generalization, as in the case of EEG dataset.

Our proposed network relies on a proper transform to retrieve the appropriate features to be stored in the SOM module. The current implementation is based on CNN, which is separately pre-trained. In future work, we will investigate the opportunity for directly applying an end-to-end training. Another aspect for improving our work is the consideration of learning in an adversarial direction simultaneously during training. Currently, the SOM blocks always update weights in the direction that favors the correct label and neglect all negative samples. How to draw inspiration from the biological perspective and to architect a network capable of doing so is still under investigation.

VI. Conclusion

In this paper, we proposed a new network architecture named MACNN, typically from the memory perspective, i.e., augmenting a SOM module with an existing CNN. We explained the inspiration from the neurophysiological perspective and detailed the network structure. By experiments, we showed the better results achieved by the proposed model compared with solely using the CNN and illustrated the characteristics of learned features. In future work, inspirations from additional neural structures, such as mutually inhibition networks, will be considered to enhance the learning capability of the corresponding articulated memory network.

Acknowledgements

The authors would like to express the sincere appreciation to the editor and anonymous reviewers for their insightful comments, which greatly improve the quality of this paper. This work was supported in part by the Australian Research Council (ARC) under discovery grant DP180100670 and DP180100656, NSW Defense Innovation Network and NSW State Government of Australia under the grant DINPP2019 S1-03/09, Office of Naval Research Global, US under Cooperative Agreement Number ONRG-NICOP-N62909-19-1-2058, the National Natural Science Foundation of China under Grant 61300167 Grant 61976120, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20191445, and sponsored by Qing Lan Project of Jiangsu Province.

References

[1] Alex Graves, Greg Wayne, et al., "Hybrid computing using a neural network with dynamic external memory," Nature, vol. 538, pp, 471-476, 2016.

[2] Sainbayar Sukhbaatar, Arthur Szlam, et al., "End-To-End Memory Networks," arXiv:1503.08895v5, 2015.

[3] Matthew D. Zeiler and Rob Fergus, "Visualizing and Understanding Convolutional Networks", European Conference on Computer Vision, 2014.

[4] Andrew P. Yonelinas, Mariam Aly, Wei-Chun Wang, and Joshua D. Koen, "Recollection and Familiarity: Examining Controversial Assumptions and New Directions," Hippocampus, vol. 20(11), pp. 1178–1194, Nov 2010.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (CVPR), 2009.

[6] Lin T.Y., Maire M., et al, "Microsoft COCO: Common Objects in Context," European Conference on Computer Vision (ECCV) pp. 740–755, 2014.

[7] Jankowski N., Duch W., and Grąbczewski K., Meta-Learning in Computational Intelligence, 1st ed, 2011.

[8] Anonymous "Memory & Learning," Neuroimage, vol. 22, pp. e589-e780, 2004.
[9] Kant Immanuel, Critique of Pure Reason (The Cambridge Edition of the Works of Immanuel Kant), Cambridge University Press, 1999.
[10] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proceedings of the National Academy of Sciences of the USA, vol. 79(8), pp. 2554–2558, April 1982.
[11] Kohonen Teuvo, "Self-Organized Formation of Topologically Correct Feature Maps," Biological Cybernetics, vol. 43(1), pp. 59–69, 1982.
[12] Turing Alan, "The chemical basis of morphogenesis," Phil. Trans. R. Soc., vol. 237, pp. 5–72, 1952.
[13] Von der Malsburg, "Self-organization of orientation sensitive cells in the striate cortex," Kybernetik, vol. 14, pp. 85–100, 1973.
[14] Teuvo Kohonen, Erkki Oja, Olli Simula, Ari Visa, and Jari Kangas, "Engineering applications of the self-organizing map," Proceedings of the IEEE, vol: 84(10), pp. 1358-1384, 1996.
[15] Graupe Daniel and Kordylewski H., "Large scale memory (LAMSTAR) neural network for medical diagnosis," IEMBS, vol. 3, pp. 1332–1335, 1997.
[16] Nimmagadda Haadvitha, Sukhavasi Sindhu, and Babu S., "Self Organising Maps: An Interesting Tool for Exploratory Data Analysis," Research Journal of Engineering and Technology, vol. 3(4), pp. 319-326, 2012.
[17] S. Clark, S.A. Sisson, and A. Sharma, "Nonlinear manifold representation in natural systems: The SOMersault," Environmental Modelling & Software, vol. 89, pp. 61-76, 2017.
[18] McCulloch Warren and Walter Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity," Bulletin of Mathematical Biophysics, vol. 5(4), pp. 115-133, 1943.
[19] MARK F. BEAR, BARRY W. CONNORS, and MICHAEL A. PARADISO, "Molecular Mechanisms of Learning and Memory," in Neuroscience Explore the Brain, 4th Edition, Wolters Kluwer, 2016.
[20] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. J. Hudspeth, "Learning and Memory," in Principles of Neural Science, McGraw Hill, 2012.
[21] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges, "THE MNIST DATABASE of handwritten digits," [Online]. Available: http://yann.lecun.com/exdb/mnist/.
[22] An end-to-end open source machine learning platform. [Online]. Available: https://www.tensorflow.org
[23] Laurens van der Maaten and Geoffrey Hinton, "Visualizing Data Using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579–2605, Nov 2008.
[24] Angel Cruz-Roa, Ajay Basavanhally, et al., "Automatic detection of invasive ductal carcinoma in whole slide images with Convolutional Neural Networks," Proceedings of the International Society for Optical Engineering, Feb 2014.
[25] DeSantis C., Siegel R., Bandi P., and Jemal, A., "Breast cancer statistics, 2011," CA: A Cancer Journal for Clinicians, vol. 61(6), pp. 408–418, 2011.
[26] Janowczyk A. and Madabhushi A. "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," Journal of Pathology Informatics, vol. 7(29), 2016.
[27] Andrew Janowczyk, "USE CASE 6: INVASIVE DUCTAL CARCINOMA (IDC) SEGMENTATION," [Online]. Available: http://www.andrewjanowczyk.com/use-case-6-invasive-ductal-carcinoma-idc-segmentation/ https://github.com/choosehappy/public/tree/master/DL%20tutorial%20Code/6-idc
[28] David M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," Journal of Machine Learning Technologies, vol. 2(1), pp. 37–63, 2011.
[29] Kay Robbins, Kyungmin Su, and W. David Hairston, "An 18-subject EEG data collection using a visual-oddball task," Data in Brief, vol. 16, pp. 227-230, Feb 2018.
[30] Lopes da Silva F. H., Niedermeyer Ernst, and Schomer Donald L., Niedermeyer's electroencephalography, Wolters Kluwer/Lippincott Williams & Wilkins Health, 2011.
[31] Ali Haider and Reza Fazel-Rezai, Application of P300 Event-Related Potential in Brain-Computer Interface, Event-Related Potentials and Evoked Potentials, Phakkharawat Sittiprapaporn, IntechOpen, 2017.
[32] Odom J.V., Bach M., Barber C., et al., "Visual evoked potentials standard (2004) Documenta ophthalmologica," Advances in ophthalmology, vol. 108(2), pp. 115-23, 2004.
[33] Bigdely-Shamlo N., Mullen T., Kothe C., and Su K.M, "The PREP pipeline: standardized preprocessing for large-scale EEG analysis," Frontiers in Neuroinformatics, vol. 9, 2015.
[34] Winkler I., Haufe S., and Tangermann M., "Automatic classification of artifactual ICA-components for artifact removal in EEG signals," Behavioral and Brain Functions, vol. 7(30), 2011.
[35] David M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," Journal of Machine Learning Technologies, vol. 2(1), pp. 37–63, 2011.
[36] Chin-Teng Lin, Tsung-Yu Hsieh, et al., "Minority Oversampling in Kernel Adaptive Subspaces for Class Imbalanced Datasets," IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 5, May 2018.