

## Research Article

# Coincidence Detection Using Spiking Neurons with Application to Face Recognition

**Fadhlan Kamaruzaman, Amir Akramin Shafie, and Yasir M. Mustafah**

*Department of Mechatronics Engineering, International Islamic University Malaysia, P.O. Box 10, 50728 Kuala Lumpur, Malaysia*

Correspondence should be addressed to Amir Akramin Shafie; [aashafie@iiium.edu.my](mailto:aashafie@iiium.edu.my)

Received 24 February 2015; Revised 22 April 2015; Accepted 12 May 2015

Academic Editor: Wei Fang

Copyright © 2015 Fadhlan Kamaruzaman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We elucidate the practical implementation of Spiking Neural Network (SNN) as local ensembles of classifiers. Synaptic time constant  $\tau_s$  is used as learning parameter in representing the variations learned from a set of training data at classifier level. This classifier uses coincidence detection (CD) strategy trained in supervised manner using a novel supervised learning method called  $\tau_s$  Prediction which adjusts the precise timing of output spikes towards the desired spike timing through iterative adaptation of  $\tau_s$ . This paper also discusses the approximation of spike timing in Spike Response Model (SRM) for the purpose of coincidence detection. This process significantly speeds up the whole process of learning and classification. Performance evaluations with face datasets such as AR, FERET, JAFFE, and CK+ datasets show that the proposed method delivers better face classification performance than the network trained with Supervised Synaptic-Time Dependent Plasticity (STDP). We also found that the proposed method delivers better classification accuracy than  $k$  nearest neighbor, ensembles of  $k$ NN, and Support Vector Machines. Evaluation on several types of spike codings also reveals that latency coding delivers the best result for face classification as well as for classification of other multivariate datasets.

## 1. Introduction

Donald Hebb first proposed that if the synapses between two neurons effectively cooperate in an activity then the synaptic efficacy of the synapse would be strengthened. Since the cooperativeness between these neurons would be more effective if it happens within a specific period of time, the idea of “Hebbian Plasticity” could also be considered as a form of coincidence detection or neuronal synchronization between the inputs of the two neurons. Previous studies show that thalamic synchronization has significant impact on cortical responsiveness and suggest that coincidence detection plays a critical role in the sensory information transmission between different brain’s regions [1] as well as in phosphoinositide signaling [2]. Subsequently, the resulting Long-Term Potentiation (LTP), a phenomenon in which synaptic strength is enhanced following bursts of synaptic activity, is vital for learning and memory [3].

In this paper, we discuss two ways of learning and classification by coincidence detections, namely, (1) learning

by weight adaptation in the form of Supervised STDP and (2) learning by synaptic time constant adaptation in the form of a novel approach called  $\tau_s$  Prediction. These two strategies are both based on Hebbian plasticity but their implementations are quite different. Here supervised learning rules are used to form the necessary synaptic weights or synaptic time constant that represent the training data and then the trained network is used for classification.

In learning stage, the network is presented with set of positive (negative) samples, and the network will be allowed certain amount of time to fire spikes. If the designated neurons fail to fire spikes, the weights or synaptic time constant will be adjusted accordingly based on the desired spike timing. This strategy would result in higher (lower) weights or synaptic time constants for neurons that barely (easily) fire. In face identification, similar faces compete more than dissimilar faces; thus this strategy would enforce stricter conditions for spike firing on neurons dedicated to facial regions with high degree of similarity. This is done by imposing smaller weights or synaptic time constant on the synaptic

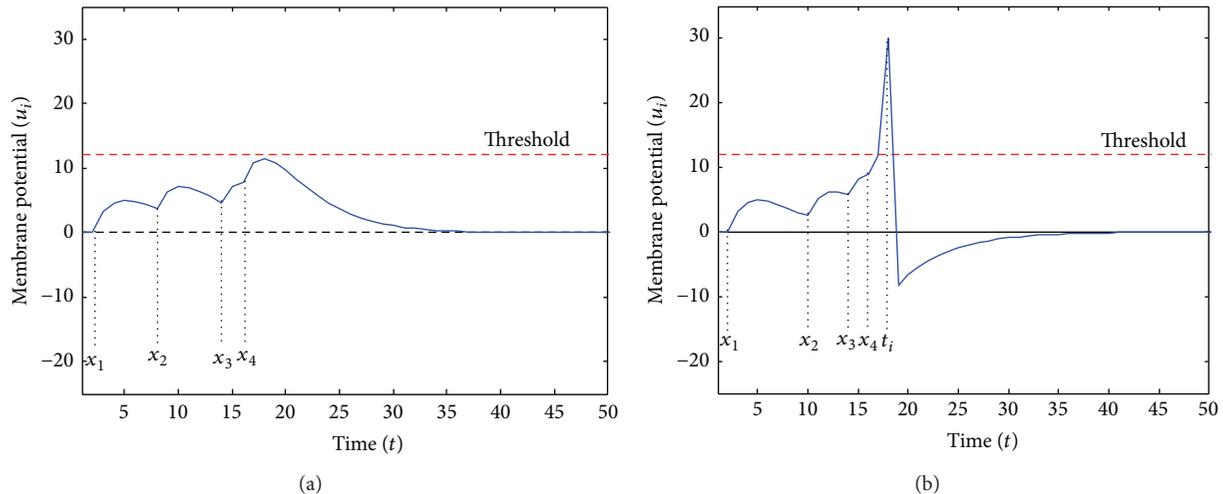


FIGURE 1: The membrane potential  $u_i(t)$  for different sets of inputs  $x_n$ : (a) the output is  $ED_4(x_1, x_2, x_3, x_4) = 0$ , and (b) the output is  $ED_4(x_1, x_2, x_3, x_4) = 1$  since  $|x_4 - x_3| \leq c_1$ .

connections. The restrictions would ensure that only highly similar facial regions would cause firing in output neuron. On the other hand, the process would also impose looser restrictions on dissimilar facial regions.

Coincidence detection strategy in practice would cause neurons connected to similar faces to fire more easily and vice versa; thus, based on coincidence detection and input synchrony point of view, it can be hypothesized that if the synaptic connection between presynaptic input pair of similar face requires larger weight or larger synaptic time constant to facilitate output spike firing, it signifies that the input pairs are connected to the facial area which possesses smaller discriminative capacity for face recognition.

Implicitly, to realize this hypothesis, it is thus assumed that (1) human beings exhibit similar intrapersonal variations, and thus (2) the learned weights and synaptic time constant from set of generic face samples could represent the actual variations that might appear on the unseen face samples.

## 2. Related Works

In this section we review briefly the idea of coincidence detection as proposed by Maass [4]. Then we highlight several supervised spiking neural network (SNN) learning methods. Subsequently, we discuss the basics of Spike Response Model (SRM) used in this paper. Finally, we take a closer look on multilayer supervised learning algorithm based on Synaptic-Time-Dependent Plasticity (STDP) approach proposed by Sporea and Grüning [5].

**2.1. Coincidence Detection Overview.** Spiking neurons can act as a coincidence detector for incoming input pulses by relaying the synchronized synaptic inputs and exact timing of spikes [4, 6, 7]. Studies of the somatosensory [8] and visual systems [9] suggested that neuronal synchronization is critical in transmitting sensory information. This relies on the

fact that synchronous inputs signals are more effective in producing higher firing rates of output spikes than asynchronous inputs signals.

Assuming the input pulses received are encoding some set of numbers, the coincidence detection can determine whether some of these numbers have equal or almost equal values. This operation if carried out on more traditional type of ANN is actually very expensive [4]. By the description of the basic idea of coincidence detection by spiking neurons [4], an output neuron  $i$  would not fire if the input neurons  $j$  fire at a temporal distance of  $\geq c_2$  but it will fire when the input neurons  $j$  fire at a temporal distance of  $\leq c_1$ . If a set of  $n$  input neuron  $j$  is used to encode  $n$  real numbers  $x_1, x_2, \dots, x_n$ , the firing patterns  $\{0 \text{ or } 1\}$  of output neuron  $i$  denoted as  $ED_n$  can be used to decode the input. For example, two cases, namely,  $ED_4 = 0$  and  $ED_4 = 1$ , are shown in Figure 1.

Recent work showed that a simple SNN model constructed by integrate-and-fire neurons and single coincidence-detector neuron can precisely read out subthreshold noisy signal [10]. The authors highlight that the two important parameters that will determine reliability and precision of the coincidence detection of the input pulses are the detection time window which can be manipulated by  $\tau_s$  and the threshold. They suggest that it is possible to obtain as much as 100% reliability of the outputs by having an optimal pair of the detection time window and the threshold.

**2.2. Supervised Learning Methods for SNN.** Supervised learning for SNN is usually performed based on the traditional gradient descent techniques. However, due to nature of spiking neurons timing, some modifications or special methods are introduced for dealing specifically with this temporal adaptation problem. Some popular methods of supervised learning in SNN includes learning method designed specifically for SNN based on the gradient descent by backpropagation of error called SpikeProp [11, 12], and Remote Supervised Method (ReSuMe) [13]. Later, Sporea and Grüning [5] extend

ReSuMe into multilayer supervised learning algorithm based on STDP where results of benchmarks on XOR problem and Iris dataset reveal successful implementation of the algorithm as well as the flexibility of this learning rule to learn different spike coding and timing patterns.

Recently an SNN learning rule called Chronotron is proposed by Florian [14]. Xu et al. [15] then proposed a supervised multispike learning rule with temporal coding based on gradient descent that aims to solve the problem of error function construction and interference among multiple output spikes during learning. Another learning rule, called Spike Pattern Association (SPAN) [16], is based on Widrow-Hoff learning rule and temporal coding that can associate multiple spatiotemporal spike patterns to desired output spike pattern. Other methods also include statistical method [17, 18], linear algebra method [19], evolutionary method [20], and Analog Spiking Neuron Approximation Backpropagation (ASNAProp) [21].

**2.3. Spike Response Model.** The formulation of spiking neuron behavior in SNN implementation described by integrate and fire model can be further simplified and represented by SRM [22]. Let  $w_{ij}$  be the weight between postsynaptic neuron  $i$  and presynaptic neuron  $j$ ,  $\tau_s$  is the synaptic time constant,  $\tau_{rec}$  is recovery time constant,  $I^{ext}$  is the external current,  $t_j^f$  is the time of presynaptic spikes, and  $\hat{t}_i$  is time of output spike, while  $\eta$ ,  $\epsilon_{ij}$ , and  $\kappa_{ij}$  are kernels,  $\delta(t)$  is Dirac delta function, and  $s = t - t_j^f$ . According to [22] the state of membrane potential  $u_m(t)$  can be computed from (1) where the kernel  $\epsilon_{ij}$  is an alpha function computed from (2):

$$u_m(t) = \eta(t - \hat{t}_i) + \sum_j w_{ij} \sum_f \epsilon_{ij}(t - \hat{t}_i, s) + \int_0^\infty \kappa_{ij}(t - \hat{t}_i, s) I^{ext}(t - s) ds, \quad (1)$$

$$\epsilon_{ij}(s) = \frac{s}{\tau_{s,ij}} \exp\left(1 - \frac{s}{\tau_{s,ij}}\right). \quad (2)$$

In SRM, each incoming spike from neuron  $j$  at time  $t_j^f$  will perturb  $u_m$  to produce presynaptic potentials (PSP) and the time course of  $u_m$  as a result of the perturbation is defined by the kernel  $\epsilon_0$ . If after the summation of PSPs  $u_m$  reaches the threshold  $v$ , output spike at time  $\hat{t}_i$  is therefore triggered. The form of the spike and the after-spike potential is described by kernel  $\eta$ . Then the zero order SRM can be constructed by neglecting the dependence of and upon the  $t - \hat{t}_i$  argument, so the kernels  $\epsilon_{ij}$  and  $\kappa_{ij}$  are set so that  $\epsilon_0(s) = \epsilon_{ij}(\infty, s)$  and  $\kappa_0(s) = \kappa_{ij}(\infty, s)$ , respectively. Assuming that there is no external current discharged into the neuron, we let  $I^{ext} = 0$ , so now (1) becomes:

$$u_m(t) = \eta(t - \hat{t}_i) + \sum_j w_{ij} \sum_f \frac{s}{\tau_{s,ij}} \exp\left(1 - \frac{s}{\tau_{s,ij}}\right). \quad (3)$$

Therefore each presynaptic spike evokes a PSP with the same time course, independent of the index  $j$  of the

presynaptic neuron and independent of the last firing time  $\hat{t}_i$  of the postsynaptic neuron. Thus the synaptic efficacies  $w_{ij}$  and  $\tau_s$  are the parameters that are responsible to scale the amplitude of the PSPs and their ‘‘effective time interval,’’ respectively.

**2.4. Supervised STDP Learning Method.** Using neurons described by SRM model in fully connected feed-forward SNN with single hidden layer, this learning rule is based on backpropagation of error [5]. The error is defined as the difference between the actual firing rate and target firing rate for all neurons. It is similar to the standard backpropagation in discrete time but derived as a functional derivative in continuous time. Assuming that the neuron has only single spike train, according to the STDP learning [22–25], the weight change between output and hidden neurons  $\Delta w_{oh}$  and between  $n_h$  hidden neurons and  $n_i$  input neurons  $\Delta w_{hi}$  can be described as

$$\Delta w_{oh} = \frac{1}{n_h} \left( A_+ \exp \frac{t_h - t_d}{\tau_+} - A_+ \exp \frac{t_h - t_o}{\tau_+} \right), \quad (4)$$

$$\Delta w_{hi} = \frac{1}{n_h n_i} |w_{oh}| \left( A_+ \exp \frac{t_i - t_d}{\tau_+} - A_+ \exp \frac{t_i - t_o}{\tau_+} \right),$$

where  $A_+ > 0$  is the amplitude,  $\tau_+ > 0$  is the time constant,  $t_h$  is hidden neuron firing time,  $t_i$  is input neuron firing time,  $t_o$  is output neuron firing time, and  $t_d$  is target neuron firing time. Note that the weight modification rules do not depend on the specific dynamics of the neuron model but only depend on the target, output, and input firing time thus making it applicable to any neuron model.

### 3. Detailed Description on Coincidence Detection

To illustrate several cases of coincidence detections under different variants of learning parameters such as synaptic time constant  $\tau_s$ , weight  $w$ , and threshold  $v$ , consider two presynaptic inputs from 2 presynaptic neurons  $j = 1$  and  $j = 2$  fired at times  $t_1^f$  and  $t_2^f$ , respectively, where  $t_2^f \geq t_1^f$ , having temporal distance between inputs  $d = t_2^f - t_1^f$ ,  $d > 0$  were propagated into a coincidence detection (CD) neuron. Under different values of learning variables, these two presynaptic inputs can either (1) invoke a spike or more than one spike (2) or none at all at the CD neuron. Since  $d$  is effectively determined by the spatiotemporal pattern of the input, then the objective of a CD neuron would be to facilitate an output spike at only certain range of  $d$  and depress firing at other instances of  $d$ .

However, in order to achieve the specified objective of coincidence detection, behaviors of learning parameters such as  $\tau_s$ ,  $w$ , and  $v$  need to be closely examined. Descriptions on a few possible coincidence detection outcomes of presynaptic inputs  $t_1^f = 2$  ms and  $t_2^f = 8$  ms (thus  $d = 6$  ms) with respect to different learning parameters are shown in Figure 2. All spikes generated using SRM model with fixed threshold.

Consider the maximum amplitude of PSP evoked by the first presynaptic input, conveniently denoted afterwards as

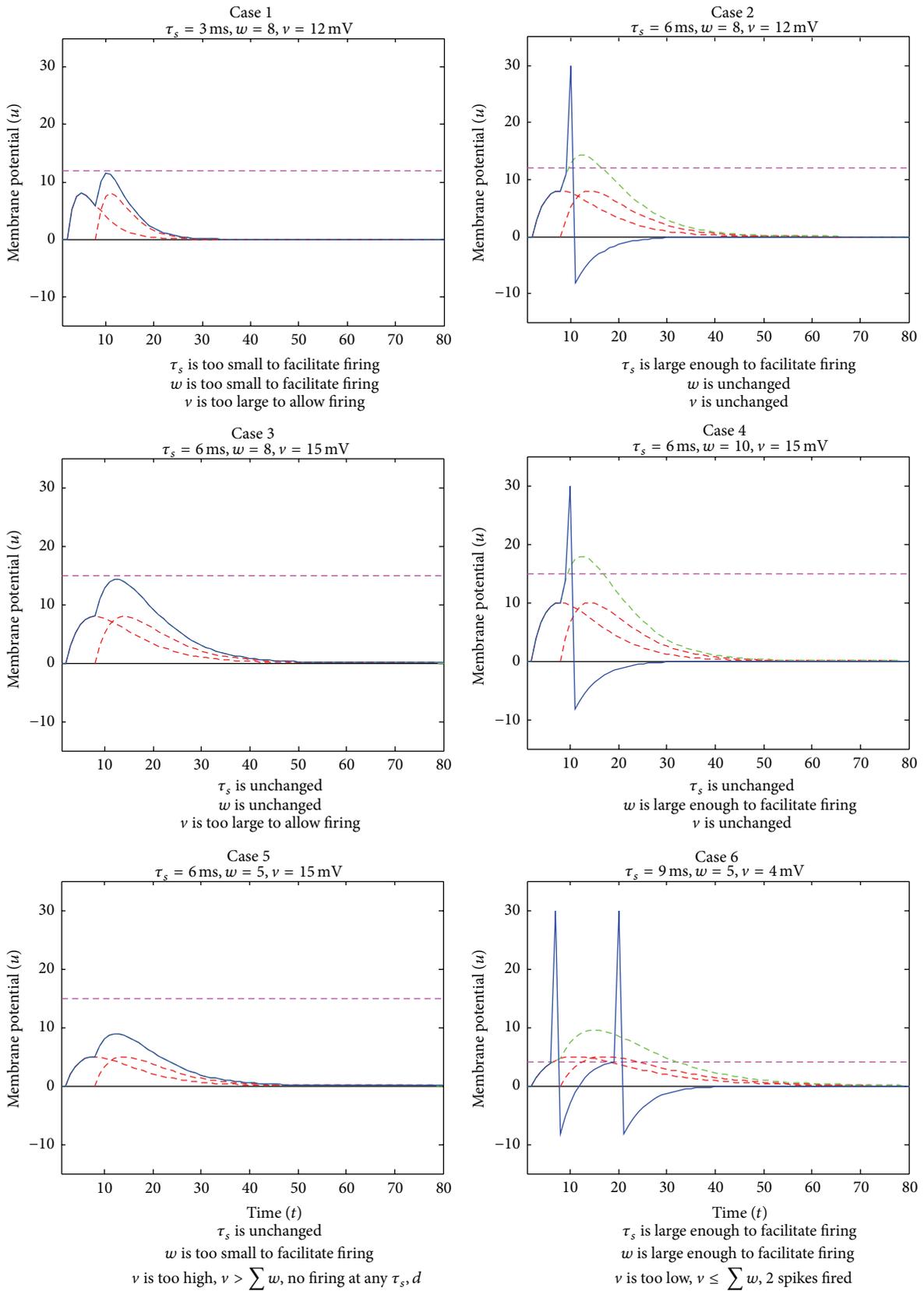


FIGURE 2: Several cases of coincidence detection between two presynaptic inputs.

PSP<sub>1,max</sub>, the weight  $w$ , and synaptic time constant  $\tau_s$ . From (2), the time at which PSP<sub>1</sub> would reach its peak  $t_{1,max}$  can be defined as  $t_{1,max} = t_1^f + \tau_s$ . Since  $s = (t_1^f + \tau_s) - t_1^f = \tau_s$ , then (2) can be written as

$$\epsilon(\tau_s) = \frac{\tau_s}{\tau_s} \exp\left(1 - \frac{\tau_s}{\tau_s}\right) = 1. \quad (5)$$

Thus, the PSP<sub>1,max</sub> can be denoted as

$$\text{PSP}_{1,max} = w \cdot \epsilon(\tau_s) = w. \quad (6)$$

For a CD neuron receiving two presynaptic spikes at a time, to ensure appropriate firing while avoiding firing facilitated by only single presynaptic spike, the proper selection of threshold should follow  $\sum w \geq v > w$ . It is assumed that there exists a minimum required synaptic time constant  $\tau_{s,min}$  that would cause a CD neuron to fire a spike. For any value of  $\tau_s$  larger than  $\tau_{s,min}$  it would definitely cause the CD neuron to fire, but with a larger delay. This can be summarized in

$$\text{Neuron spike} \begin{cases} \delta(t_i) = 1, & \text{if } \tau_{s,min} \leq \tau_s < \infty \\ \delta(t_i) = 0, & \text{if } \tau_{s,min} > \tau_s \geq 1. \end{cases} \quad (7)$$

There are two distinctive regions which can be defined for the firing behavior of a CD neuron. These two regions are (1) firing region (FG) and (2) nonfiring region (NFG). These regions are illustrated in Figure 3. According to Figure 3, there is 100% probability that any PSP that are strong enough to cause a CD neuron to fire would reach the threshold in FG region, while, in NFG, any neuron that failed to fire in FG region would absolutely not be able to fire in NFG region. The two NFG regions are discontinued by an FG region, where  $t_{fg} - t_2^f$  signifies the maximum time interval of FG region. The boundaries for these distinctive regions are given in the following equation, where we defined  $t_{fg} = \tau_s + t_2^f$ :

$$\begin{aligned} \text{FG} &= \{t_2^f, t_{fg}\}, \\ \text{NFG} &= \{0, t_2^f\}, \{t_{fg}, \infty\}. \end{aligned} \quad (8)$$

Consider if threshold  $v$  is set at the highest possible value such that  $v = 2w$ , for this threshold, only PSP resulting from two presynaptic spikes fired at the same time can reach it, where  $t^f = t_1^f = t_2^f$ . Based on (3), assuming that the CD neuron is allowed to fire only once, the refractory kernel  $\eta(t - \hat{t}_i)$  can be set to be fixed at  $\eta_0 \leq 0$ . Let  $s = (t - t^f)/\tau_s$  and  $\eta_0 = 0$ ; since  $t_1^f = t_2^f$ , (3) can be written as (9). For complete calculation readers are referred to Appendix A. Consider

$$\ln(s) - s + 1 - \ln\left(\frac{u_m}{2w}\right) = 0. \quad (9)$$

As  $\ln(s) - s$  is analytically unsolvable, we can use numerical approximation of function of  $\ln(s) - s + 1 - \ln(u_m/2w) = 0$  for  $s \in [0, 1]$ , where solution for  $s$  is found as  $s = 1$ ; hence  $t = \tau_s + t^f$ . Since  $t^f = t_1^f = t_2^f$ , and any PSP that occurs before  $t_2^f$  would yield a lower membrane potential, thus the maximum achievable membrane potential for this coincidence neuron would happen before or precisely at  $\tau_s + t_2^f$ .

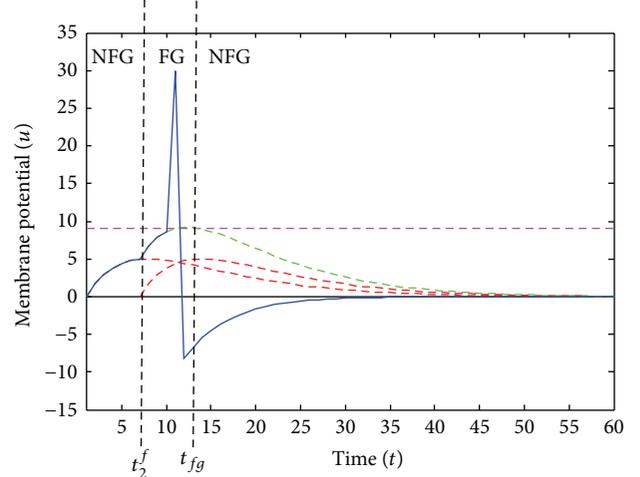


FIGURE 3: Illustration on firing region and nonfiring region of a CD neuron receiving 2 presynaptic spikes at  $t_1^f = 0$  ms and  $t_2^f = 7$  ms.

## 4. Proposed Method

**4.1. Output Spike Time Prediction.** One major problem of any spiking neuron model is the processing time taken to evaluate the exact level of membrane potential prior to any spike triggering. According to Makino [26] it is difficult to predict firing for a complex neuron such as SRM model since it involves delayed-firing and causality. Furthermore, approximation methods such as firing time prediction can be inexact while exact simulation is limited to simple models [27]. Makino [26] proposes an event-driven SRM using incremental partitioning method which uses linear envelopes of the state variable of a neuron to partition the simulated time. This would cause the firing time to be reliably calculated by implementing the bisection-combined Newton-Raphson method to each resulting partition.

In discrete-time approach, the system needs to compute accurately the level of membrane potential in each discrete time and also to update the state variables for precise output spike timing. If the numbers of discrete intervals  $\sigma$  (which is also known as sampling rate,  $f_s$  in continuous-to-discrete signal conversion) between specific period of time  $\Delta t$  are large, then the evaluations would take considerably large amount of time. One way to overcome this problem is by reducing the number of  $\sigma$  but this would reduce the precision of the resulting output spike timing. Here we propose an approximation method called Output Spike Time Prediction (OSTP) SRM to solve this problem.

Consider the simplified SRM model in (3) where the output neuron  $i$  is assumed to receive two presynaptic inputs from two presynaptic neurons  $j = 1$  and  $j = 2$  at two different times  $t_1^f$  and  $t_2^f$  where  $t_2^f > t_1^f$ . The objective of OSTP is to find the estimated spike time  $t_{i,est}$  of the output neuron. For convenience, let  $s_j = (t - t_j^f)/\tau_s$  and  $\eta(t - \hat{t}_i) = \eta_0$  so that (3) now becomes

$$\begin{aligned} u_m(s) - \eta_0 &= w [s_1 \cdot \exp(1 - s_1)] \\ &\quad + w [s_2 \cdot \exp(1 - s_2)]. \end{aligned} \quad (10)$$

Then, let  $a = t_2^f - t_1^f$ ; thus  $t_2^f = a + t_1^f$ ; consequently,  $s_2 = s_1 - (a/\tau_s)$ . Thus, we can write (10) as (11). For simplification of notation, let  $A = \exp(1)[1 + \exp(a/\tau_s)]$  and  $B = [(a/\tau_s) \cdot \exp(1 + (a/\tau_s))]$  so that the OSTP equation can be described in (12). For complete calculation readers are referred to Appendix B. Then, using the numerical approximation as discussed previously to solve for  $s_1$ , the estimated time of spike  $t_{\text{est}}$  can be obtained using (13). For  $t_1^f = 1$  ms,  $t_2^f = 3$  ms, and  $\tau_s = 5$ ,  $s_1 = [0 \ 5]$  plot of OSTP equation in (12) with 10 equally spaced values of the term  $(u - \eta_0)/w$  from 0 until 3 is illustrated in Figure 4:

$$\frac{u - \eta_0}{w} = s_1 \cdot \exp(-s_1) \cdot \exp(1) \left[ 1 + \exp\left(\frac{a}{\tau_s}\right) \right] \quad (11)$$

$$- \exp(-s_1) \left[ \frac{a}{\tau_s} \cdot \exp\left(1 + \frac{a}{\tau_s}\right) \right],$$

$$\exp(-s_1) (A \cdot s_1 - B) - \frac{u - \eta_0}{w} = 0, \quad (12)$$

$$t_{\text{est}} = s_1 \cdot \tau_s + t_1^f. \quad (13)$$

From Figure 4, several lines manage to cross 0 while others do not, which indicates that no solution of  $s_1$  could be obtained. The lines that cross 0 (blue lines) are drawn from variables that would produce spike firings while others that do not cross (red-dashed lines) are drawn from variables that would not produce any spike firing. Thus, the precise spike firing time can be approximated given that the numbers of discrete intervals between the boundaries of  $s_1$  are sufficient to provide precise estimation. Similar to discrete estimation case, OSTP relies on discrete intervals  $\sigma_{\text{OSTP}}$  between  $\Delta s_1$  in order to produce accurate estimation. However, discrete intervals between  $\Delta s_1$  in OSTP do not have significant effect on the overall processing time unlike discrete-time which will be seen later.

**4.2. Output of Coincidence Detection Network.** As explained earlier, any sufficiently close pair of presynaptic inputs would produce output spikes. However, for reliable and accurate output spikes, two additional spikes or ‘‘cues’’ called the start cue and end cue are added for each output spike train. These two cues indicate the start and end of CD neuron simulation. Without start cue, the firing delay of each occurring spike cannot be accurately determined. The significance of the start cue has been discussed at length in literatures [11, 28]. The time of start cue is kept when the first presynaptic spike arrives at CD neuron. In contrast, without end cue, some meaningful information encoded by presynaptic spikes that are unable to evoke any output spike would be lost and this would affect the spike coding accuracy.

Another important element which is particularly important for STDP supervised learning is ‘‘imaginary spike.’’ It is important to mention that imaginary spike is not used for classification; however it is used to compute error signal for supervised adjustment of the weight. This imaginary spike

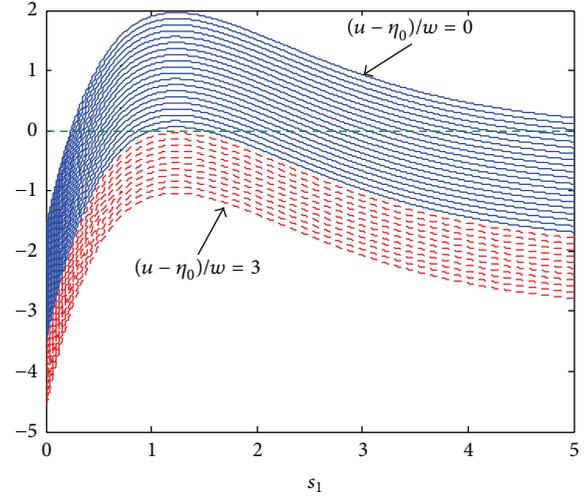


FIGURE 4: Visualization of OSTP equation with 10 equally spaced values of  $(u - \eta_0)/w$  between  $[0 \ 3]$  for  $t_1^f = 1$  ms,  $t_2^f = 3$  ms, and  $\tau_s = 5$ ,  $s_1 = [0 \ 5]$ .

indicates the end of a designated time  $t_{\text{deg}}$  if a presynaptic input fails to produce a spike within certain period. As shown earlier, the maximum time for presynaptic input pair to attain the highest membrane level and evoke a spike is  $t_{fg} = \tau_s + t_2^f$  so the designated time  $t_{\text{deg}}$  for a spike to fire is set slightly larger than  $t_{fg}$ , where

$$t_{\text{deg}} = 2\tau_s + t_2^f. \quad (14)$$

Note that the presynaptic inputs only maintain the temporal integrity of the input, while the spatial integrity is embedded into the spatial location  $\mathcal{L}$  of the presynaptic neuron. To achieve spatiotemporally reliable output spike pattern, it is vital to sequence the presynaptic inputs. This process is carried out by allowing only single pair of presynaptic inputs from neurons sharing the same spatial location to take part in evoking output spike at one time. After an output or imaginary spike is produced, another pair from next location is allowed to take part in evoking spike, if any. This process is repeated until the end of simulation; that is, all presynaptic inputs have taken part in evoking output spike.

In this implementation, the computation of exact membrane potential and the term  $\eta(t - \hat{t}_i)$  would slow the whole network down. Thus, for simplicity, the current level of membrane potential  $u_m$  (for spiking case) and the membrane reset term  $\eta(t - \hat{t}_i)$  (for nonspiking case) just before next presynaptic input pair takes part in evoking the output spike are changed to membrane reset constants  $\eta_o = \eta_{\text{no spike}}$  and  $\eta_o = \eta_{\text{spike}}$ , respectively, with respect to (10). This is actually feasible assuming that both spiking and nonspiking cases of coincidence detection are producing spikes (actual and imaginary spikes). The output spike train of a CD neuron using continuous SRM, discrete-time SRM, and OSTP SRM in a CD classification network accepting 9 presynaptic input pairs is shown in Figure 5, where the parameters used are  $w = 5$ ,  $\tau_s = 3$  ms,  $v = 7$  mV,  $\eta_{\text{no spike}} = 2$  mV, and  $\eta_{\text{spike}} = -2$  mV.

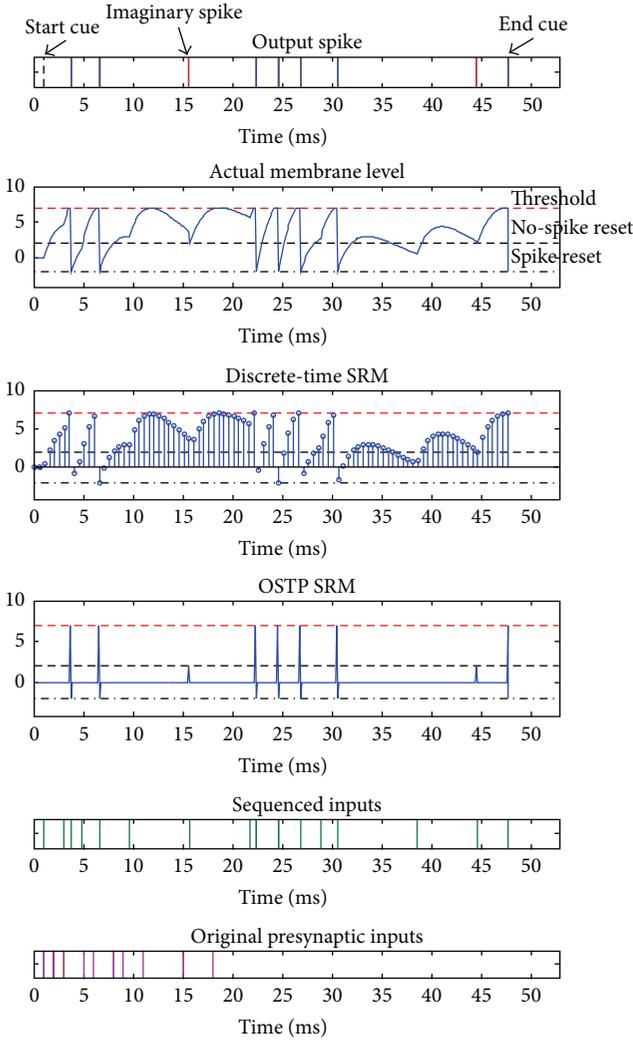


FIGURE 5: Precise spike firing time at a CD neuron by continuous SRM ( $\sigma_{\text{disc}} = 100$  ms, providing almost actual membrane level for comparison), discrete-time SRM ( $\sigma_{\text{disc}} = 2$  ms), and OSTP SRM ( $\sigma_{\text{OSTP}} = 10$ ) approximation in CD SNN classification.

Based on Figure 5 each bar represents the precise timing of output spikes (blue), imaginary output spikes (red), cues (dashed black), sequenced presynaptic inputs (green), and original presynaptic inputs (purple). The resulting output spikes for these 3 approaches are exactly the same. Imaginary output spikes are needed specifically to train the network by Supervised STDP approach and are not used for classification. There are 2 cues (start and end (cooccurring at the end of output spike)), and 7 spikes and 2 imaginary spikes are produced by the CD neuron.

The numbers of actual spikes are generally less than the number of presynaptic input pairs; however the number of total spikes (actual + imaginary) would be equal to the number of presynaptic input pairs. Note that the original presynaptic inputs do not have spatial integrity intact since they share several instances of similar presynaptic spike time and tend to clutter together in temporal neural network.

The temporal sequencing of input makes use of spatial locations by temporally rearranging the presynaptic inputs. Furthermore, it can be observed that the OSTP SRM implementation produces identical spikes as discrete-time SRM.

**4.3. Learning Process for Coincidence Detection.** We propose a novel learning approach called  $\tau_s$  Prediction, which is carried out by approximating the required synaptic time constant to produce an output spike. Unlike Supervised STDP in [5], this learning process only requires positive (+1) class sample.

The main objective of  $\tau_s$  is much simpler than Supervised STDP, that is, to evoke small-delayed spike in matching sample. Additionally, this process is assumed to be able to implicitly depress the spike or delay it longer for nonmatching sample, given that the temporal distances between the input pairs are greater than a certain range. Since changes in  $\tau_s$  could facilitate, depress, or delay the output spike, using as  $\tau_s$  learning parameter should allow this type of training in coincidence detection spiking neural network.

Assume that the training process needs to approximate the value of  $\tau_s$  that would allow a pair of inputs  $t_1^f$  and  $t_2^f$  from matching face to evoke a spike at CD neuron output at desired time  $t = t_d$ . By letting  $X = 1/\tau_s$ , from (10) we can have (15). For complete calculation readers are referred to Appendix C. Then,  $X$  can be solved by finding the zero crossing using the numerical approximation discussed earlier. After that, we can compute the approximation of synaptic time constant  $\tau_{s,\text{approx}}$  using (16). Finally, the change in synaptic time constant  $\Delta\tau_{s,m,n}$  for each pixel  $n$  belongs to local patch  $m$  such that  $n \in m$  can be computed using (17):

$$X \left[ (t_d - t_1^f) \cdot \exp(1) \exp(-(t_d - t_1^f)(X)) + (t_d - t_2^f) \cdot \exp(-(t_d - t_2^f)(X)) \right] - \frac{u - \eta_0}{w} = 0, \quad (15)$$

$$\tau_{s,\text{approx}} = \frac{1}{X}, \quad (16)$$

$$\Delta\tau_{s,m,n} = \frac{A_+}{KK_+} \sum_{k=1}^K \sum_{k_+=1}^{K_+} \tau_{s,\text{approx}}, \quad (17)$$

where  $K$  is total number of training subjects (classes),  $K_+$  is positive sample per training subject, and  $A_+$  is learning constant. Since the teacher signal  $t_d = t_2^f + \tau_s$  depends on the current state of  $\tau_s$ , it would evolve along the training process allowing it to have dynamic behavior. Additionally, there are instances of function in (15) where the plot does not produce any zero crossing, similar to no-solution case shown in Figure 5. Therefore, the approximated  $\tau_s$  can take the value of  $t_2^f - t_1^f$  since the minimum synaptic time constant  $\tau_s$  required to evoke the spike in a CD neuron is less than or equal to  $t_2^f - t_1^f$  as proven in Appendix D. Since this approach only involves estimation based on positive samples, the corresponding spike error  $e$  for  $\tau_s$  Prediction can be given

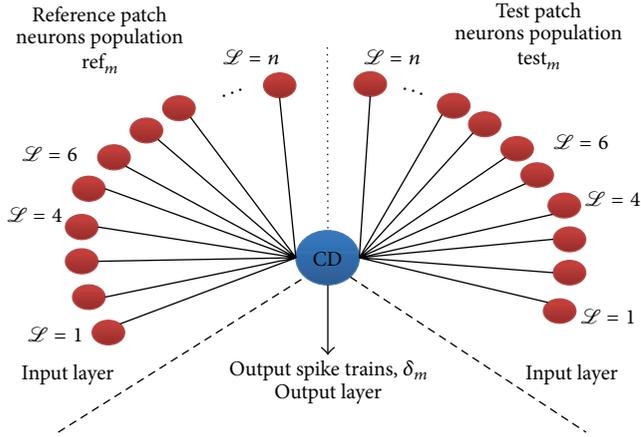


FIGURE 6: Illustration of a CD neuron (center) attached to single patch of face receiving presynaptic inputs attached to locally lateral LP  $m$  from gallery face and probe face. Each neuron connected to the CD neuron patch assumes spatially different input  $\mathcal{L}$  to ensure that spatiotemporal information is intact. This coincidence detection implementation adopts a feed-forward 2-layer neural network approach comprising an input layer and an output layer.

simply as a resulting error from sensitivity (true positive rate) which is given as

$$e = 1 - \left( \frac{e_{TP}}{e_{TP} + e_{FN}} \right), \quad (18)$$

where, for  $\tau_s$  Prediction, the spike errors  $e_{TP} = t_d - t_i$  and  $e_{FN} = t_i - t_d$ . Thus, from (18) the total spike error becomes

$$e = \frac{1}{MK} \sum_M \sum_K \frac{1}{K_+} \sum_{K_+} 1 - \left( \frac{e_{TP}}{e_{TP} + e_{FN}} \right). \quad (19)$$

**4.4. Face Classification Using Coincidence Detection.** In order to apply coincidence detection as a classifier, the CD neurons are used as output neurons in a 2-layer feed-forward neural network. Using local ensemble strategy for face recognition employed in [29, 30], each CD neuron is attached to each local patch. Thus the number of CD neurons in the classifier network would be equivalent to total number of local patches  $M$ . The number of inputs neuron however depends on the dimension,  $N$  of the local patch. Since this coincidence detection would evaluate the synchronization between a gallery and a probe input, each CD neuron would have presynaptic neurons and connections of  $2N$ . The detailed network connections and elements comprising single CD neuron as output neuron are shown in Figure 6.

The output spike train for each local patch  $\delta_m$  from the CD neuron will then be fed to a summation stage where the outputs of all CD neurons in the network will then be evaluated to produce vectors called Non-Coincidence Factor (NCF),  $d_m$  for each local patch. This NCF describes degree of coincidence between an input pair where smaller values of  $d_m$  would indicate higher coincidence between the inputs thus higher matching probability and vice versa. Different spike codings can be used to interpret the output spike trains and,

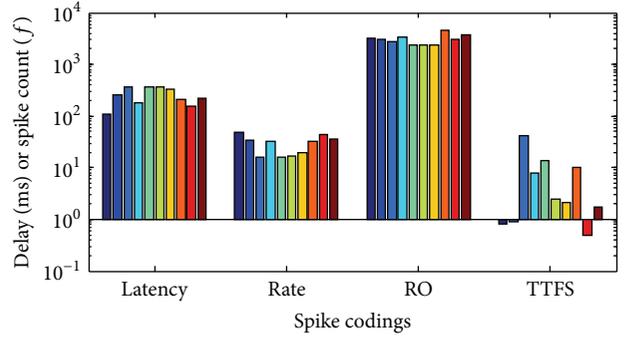


FIGURE 7: Detailed description on the classification model, showing different spike interpretations on classification of a probe patch against 10 gallery patches.

here in this paper, the performance of several spike codings, namely, latency, rate by spike count (conveniently denoted simply as “rate” afterwards), rank order (RO), and time to first spike (TTFS) is investigated. Equations (20) to (23) describe several spike codings used and the associated NCF,  $d_m$  as follows.

**Latency.** Consider the following:

$$d_m = \sum_{l=1}^{\mathcal{L}} (\delta_m(\hat{t}_l) - \delta_m(\hat{t}_{l-1})). \quad (20)$$

**Rate (Spike Count).** Consider the following:

$$d_m = 2N - \sum_f \delta_m(\hat{t}^f). \quad (21)$$

**RO.** Consider the following:

$$d_m = \sum_{l=1}^{\mathcal{L}} (\delta_m(\hat{t}_l) - \delta_m(\hat{t}_1)). \quad (22)$$

**TTFS.** Consider the following:

$$d_m = (\delta_m(\hat{t}_2) - \delta_m(\hat{t}_1)). \quad (23)$$

At the summation stage, the NCF obtained from different spike codings would indicate the gallery image which has the highest probability to be the correct match of the probe. For a more detailed description on the classification model, the delay and spike firing counts of a CD neuron’s firings caused by classification of a probe image’s patch of dimension,  $n = 49$  with 10 lateral gallery images’ patches are shown in Figure 7. Note that each bar for each spike coding case represents a gallery image patch, and the first gallery in each case is the actual match for the probe. Based on the figure, we can observe that, for latency, RO, and TTFS coding, lower delay in CD neurons’ firing signifies higher coincidence and hence higher probability of correct match, while, for

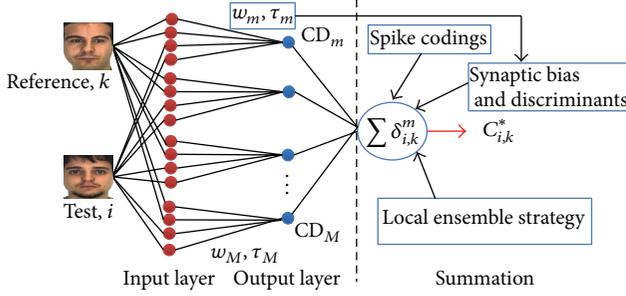


FIGURE 8: Coincidence detection classifier evaluating synchronization of two classes of inputs, namely, reference face  $k$  and test face  $i$  comprising  $M$  CD neurons as output neurons.

spike interpretation by rate coding, the firing delay is totally insignificant but higher rate of spike firings would indicate higher coincidence between the probe and the gallery image. Note also in the figure that a misclassification occurs in TTFS case, where the 9th gallery image patch is found to be having the highest coincidence.

Aside from applying the proper spike codings to interpret the output spike trains, two other processes are also carried out at the summation stage in order to add discriminative influence on the final classification. Firstly, after the spike codings are applied, local ensemble strategies are adopted to locally classify the resulting  $d_m$  and then  $d_m$  are normalized to find the local confidence vectors  $c^m = \log(\arg \min\{d_m\} + 2) / \log(d_m + 2)$ . Secondly, these confidence vectors will be weighted using discriminants denoted as  $\varphi_m$  acquired from the learnt synaptic time constants  $\tau_{s,m,n}$  computed using

$$\varphi_m = \frac{1}{N} \sum_{n=1}^N \tau_{s,m,n} \quad \text{where } n \in m. \quad (24)$$

Here, the discriminants are then normalized to ensure that  $\varphi_m$  would take the values between  $[0 \ 1]$ . Consider a probe  $i$  needing to be matched to gallery  $k$  with each local patch dimension  $n = 2$  using a CD classification network with fully trained  $\tau_s$ , the total weighted confidence  $c_k^*$  of probe  $i$  as belonging to gallery  $k$  can be obtained from (25). For illustration purpose, the whole classification network is shown in Figure 8:

$$c_k^* = \sum_{m=1}^M (\varphi_m c_k^m). \quad (25)$$

## 5. Experimental Results and Discussions

In this section, we conduct a test to validate and evaluate the accuracy of OSTP approximation and its efficiency. Then we compare the performance of coincidence detection trained by  $\tau_s$  Prediction against coincidence detection trained by supervised STDP. Subsequently, we investigate the performance of several spike codings used in our proposed coincidence detection. Then using Principal Component Analysis (PCA) and Gabor features, we assess the performance of the proposed CD classifier against several types of classifiers, namely,

$k$  nearest neighbor classifier ( $k$ NN), ensembles of  $k$ NN classifier (soft  $k$ NN) [30], Support Vector Machine (SVM), and ensembles of SVM classifiers (soft SVM) (inspired by [30]).

We adopt Single Sample per Person (SSPP) face recognition (for review, see [31]), where only single image per person is used as gallery. Four publicly available datasets are used for the experiments, namely, AR, JAFFE, FERET, and CK+ datasets. The AR dataset [32] contains frontal images of 76 males and 60 females with several types of variations such as different illumination conditions, expressions, and partial occlusions. Images were taken in two sessions (S1 and S2) with 13 images per session. We use only 8 expression-variant images (neutral, smile, angry, and scream) and 4 partially occluded images (sunglasses and scarf) from both sessions. JAFFE dataset [33] contains 212 expression-variant images from 10 female Japanese subjects. There are 7 types of expressions in this dataset and each subject portrays at least 3 images for each expression. FERET dataset [34] consists of 13,539 facial images corresponding to 1,565 subjects, which are diverse across ethnicity, gender, and age. Two subsets were used, namely,  $fa$  and  $fb$  following the standard FERET evaluation protocol [34]. Subset  $fa$ , containing 1,196 frontal images of 1,196 subjects, was specifically used as gallery, while  $fb$  (1,195 expression-variant images) was used as probes. CK+ dataset [35] contains 523 sequences from 123 subjects portraying seven basic expressions (happiness, sadness, surprise, anger, disgust, fear, and contempt). Examples of images from AR, JAFFE, FERET, and CK+ datasets are shown in Figure 9.

As standard preprocessing step, all images used are aligned and resized to  $84 \times 84$  pixels. Histogram equalization is applied on all images except for images with scarves and sunglasses in AR dataset. This is due to too much irregularity caused by the histogram equalization process on the occluded parts of the image (i.e., the scarves and sunglasses), consistent with the suggestion in [30]. Each image is partitioned into 144 square local patches of  $7 \times 7$  scanning window. This will result into total dimension of 7056 pixels per image and the vector dimension of  $N = 49$  per local patch. Hence the number of afferents that connect input neurons pair to CD neuron is equal to 7056. Each afferent is assumed to be representing a pair of neurons connected to inputs at spatial location  $\mathcal{L}$ .

**5.1. OSTP Performance Analysis.** A test is carried out to determine the accuracy and speed of the OSTP approximation by comparing  $t_{\text{est}}$  with the exact spike firing time  $t_{\text{exc}}$  obtained from computation of discrete-time SRM in coincidence neuron. Consider that the sampling frequency or discrete intervals  $\sigma$  for discrete-time SRM and OSTP SRM is measured for each 1 ms and  $\Delta s_1 = 1$ , respectively; the evaluation of discrete-time SRM uses  $\sigma_{\text{disc}} = 1$  while OSTP SRM uses  $\sigma_{\text{OSTP}} = 100$ . The error  $e$  for this test is defined as  $e = ||\lceil t_{\text{est}} \rceil - t_{\text{exc}}|$ , where  $\lceil x \rceil$  denotes the ceiling process. The test is performed on 191000 neurons, having different values of threshold, synaptic time constant, and weights. It is found that the approximation is correct 99.6% of the time and the processing time taken for OSTP is 9.2454 seconds while discrete-time SRM took 498.7757 seconds. This yields

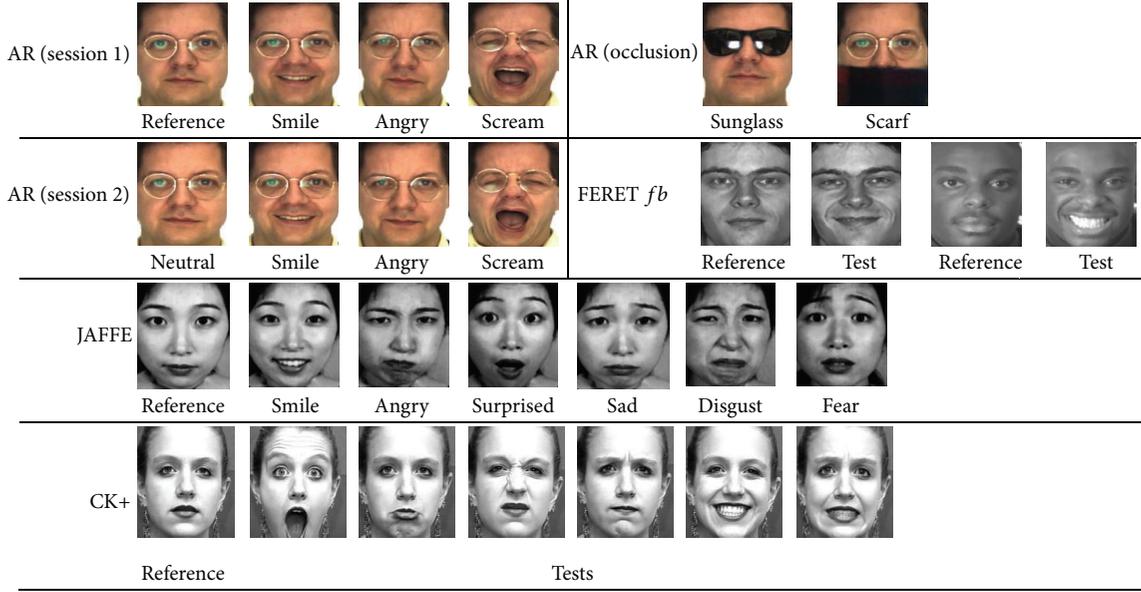


FIGURE 9: Examples of images used in this paper. The variations of probe samples in AR are neutral, smile, angry, scream, sunglasses, and scarf, while the variations of probe samples in JAFPE are neutral, smile, angry, surprised, sad, disgust, and fear. FERET probe samples' variations, however, consist of mixtures of several different expressions while CK+ consists of several FACS-coded expressions.

a significant reduction in processing speed by more than 98% of the discrete-time SRM's.

In order to investigate further on the effect of discrete intervals  $\sigma_{OSTP}$  and  $\sigma_{disc}$  on the performance of the spike firing time approximation, the accuracy, Mean Squared Error (MSE), processing speed, and number of floating points operations of OSTP and discrete-time SRM are compared for different values of  $\sigma_{OSTP}$  and  $\sigma_{disc}$ , ranging between 1 and 10000. This test uses  $n = 20500$  pairs of input neurons with different combinations of presynaptic inputs, thresholds, synaptic time constants, and weights. The accuracy and MSE of both OSTP and discrete-time SRM is compared with the exact spike timing  $t_{exc}$ , where  $t_{exc}$  is obtained by discrete-time using  $\sigma_{disc} = 1000$ . The approximated spike firing accuracy is calculated based on the following equation while MSE is computed from  $MSE = (1/n)(\sum_{n=1}^n (t_{est} - t_{exc})^2)$ :

$$\text{approximation accuracy} = \frac{1}{n} \left( \sum_{n=1}^n x_n \right) \times 100\%,$$

$$\text{where } x = \begin{cases} 1, & \text{if } |[t_{est}] - t_{exc}| = 0, \\ 0, & \text{if } |[t_{est}] - t_{exc}| \neq 0. \end{cases} \quad (26)$$

The number of floating points operations is computed as the total numerical operations from the start to the end of the test. The results on the effect of discrete intervals' size on the performance of OSTP and discrete-time SRM are shown in Figure 10.

According to Figure 10, OSTP produces comparable optimal accuracy and MSE to discrete-time SRM at  $\sigma_{OSTP} = 100$ , while consuming minimal processing speed and constant floating points operations. Discrete-time operation on the other hand, even though producing good accuracy and low

MSE, consumes exponentially increasing processing speed and floating points operations with respect to  $\sigma_{disc}$ . From Figure 10(b), for all tested  $\sigma$ , an average of 12.82 seconds is required by OSTP to produce all spikes as opposed to  $18.75 \times 10^4$  seconds required by discrete-time SRM. As a matter of fact, according to Figure 10(b), at  $\sigma_{OSTP} = \sigma_{disc} = 100$ , the processing speed achieved by OSTP is more than 99% faster than discrete-time SRM while producing comparable accuracy of spike timing as indicated by Figure 10(a). These results highlight the efficiency and performance of the proposed OSTP.

*5.2. Face Recognition Performance of  $\tau_s$  Prediction and Supervised STDP.* In order to examine the recognition accuracy of CD classifier trained with Supervised STDP and  $\tau_s$  Prediction, an experiment is conducted. Following the recommendation by Nordlie et al. [36], a tabular description of experimental setup is given in Table 1.

Each dataset is randomly split into two groups, where each group has half of total number of subjects available. Each split follows 2-fold cross validation method, where each group is interchangeably used as training and then tested once, and after that the average recognition accuracy is taken. This random split is repeated 10 times and the final average accuracy for both training and test along with the standard deviation is recorded in Table 2. For this particular experiment, the spike coding used to interpret the output spikes of CD neuron is latency coding. The Baseline accuracy in Table 2 is obtained from  $k$ NN approach.

Based on result presented in Table 2, CD classifier using either Supervised STDP or  $\tau_s$  Prediction on average delivers better recognition accuracy than the Baseline approach. For the test sets, CD classifier with Supervised STDP delivers

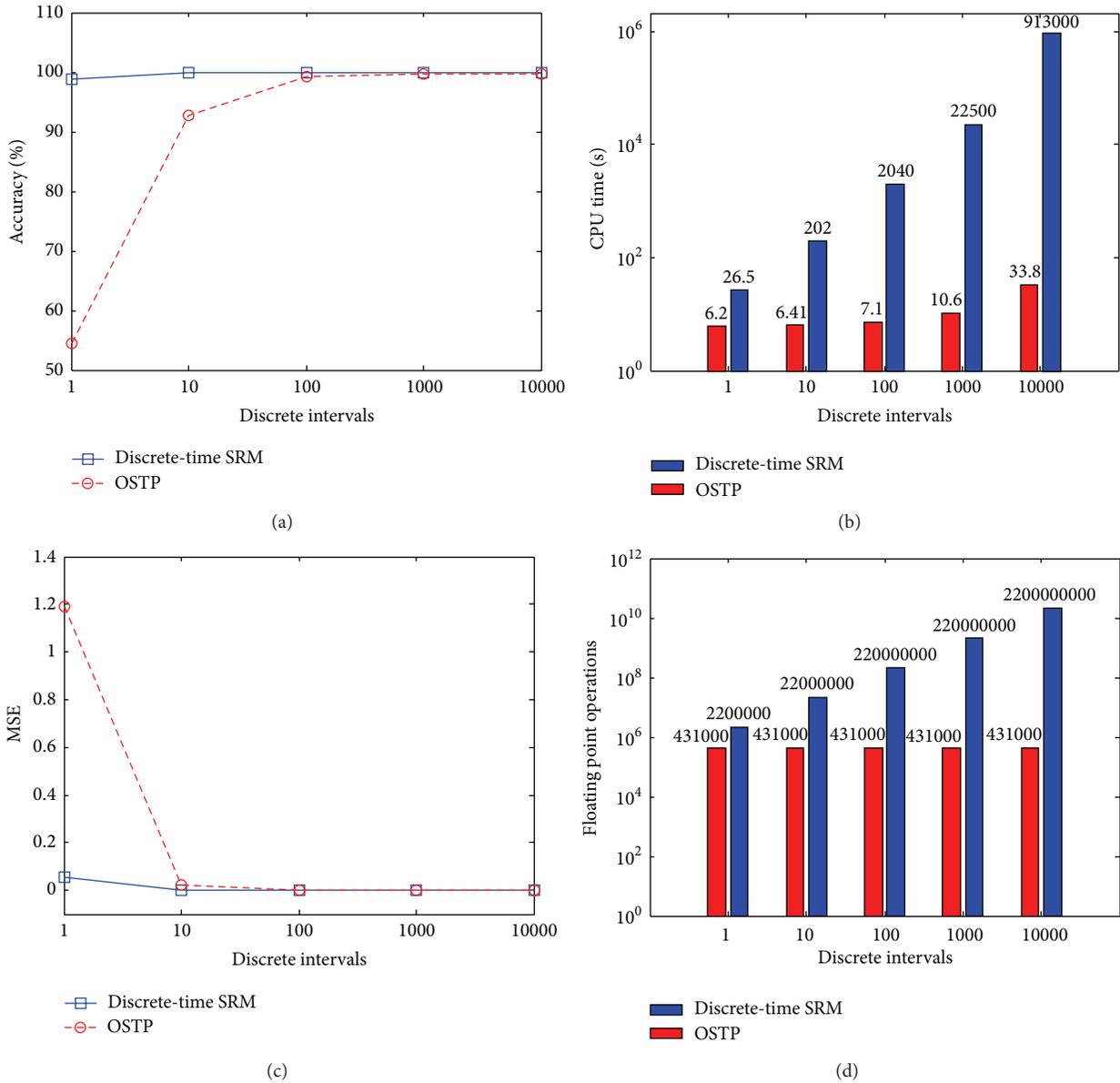


FIGURE 10: Comparisons on the (a) accuracy, (b) processing speed, (c) MSE, and (d) number of floating points operations of OSTP and discrete-time SRM for different values of  $\sigma_{OSTP}$  and  $\sigma_{disc}$  between 1 and 10000.

average recognition accuracy of 95.56%, while  $\tau_s$  prediction is at 96.31% where both are more than 20% better than Baseline accuracy. In terms of performance between training and test samples, their performances are comparable, signifying that no overfitting occurs during training process. On average,  $\tau_s$  Prediction performs slightly better than Supervised STDP by just around 1% difference in recognition accuracy.

5.3. *Convergence Analysis.* Since both learning methods are iterative algorithms, their performances with respect to different number of iterations need to be examined. Using AR Scream S1 and Scarf S1, the average recognition accuracy of the training set as the iteration grows for Supervised STDP and  $\tau_s$  Prediction is shown in Figures 11(a) and

11(b), respectively. Similarly the corresponding spike error  $e$  with  $\beta = 2$  is also presented in Figure 11(c). According to Figure 11(b), the latency coding converges to minimum recognition error immediately at epoch = 1, while RO and TTFS coding converges at epoch > 10. Special case of convergence is observed for rate coding since it converges to minimum after epoch = 30 and produces lowest error  $e_{rate} = 0$ . However, the performance of test set for rate coding is not as high, in which we found that test error of  $e_{rate} = 0.10$  is obtained (not shown in Figure 11). This indicates an *overfitting* case for rate coding, which also signifies that, in rate coding, large data with large dimension would require a very large number of spikes to reliably distinguish each individual class. Thus it is recommended to use epoch = 1

TABLE 1: Tabular description of experimental setup as suggested in [36].

	Supervised STDP	$\tau_s$ Prediction
Model summary		
Neuron model	Zero Order Spike Response Model (SRM <sub>0</sub> ) with OSTP	
Synaptic model	$\alpha$ -shaped PSP	
Synaptic function	$\epsilon_{ij}(s) = (s/\tau_{s,ij}) \exp(1 - (s/\tau_{s,ij}))$	
Input connectivity	Feed forward, with each input neuron connected to each image pixels	
Parameters		
Weight, $w$	Trained, start at constant 4	5
Synaptic time constant, $\tau_s$	3 ms	Trained, start at constant 1 ms
Learning time constant, $\tau_+$ and $\tau_-$	$\tau_+ = \tau_- = 10$ ms	none
Threshold, $v$	9 mV	5.1 mV
No-spike reset potential, $\eta_{no\ spike}$	0 mV	2 mV
Spike reset potential, $\eta_{spike}$	0 mV	-2 mV
Learning constant, $A_+$	0.3	1.0
Learning constant, $A_-$	0.1	none
OSTP discrete interval $\sigma_{OSTP}$	10	10
Positive sample per subject, $K_+$	All available	All available
Negative sample per subject, $K_-$	1 (random selection)	None
Input model		
Type	Direct conversion	
Details	Input spike time is equivalent to normalized value of image pixels Each neuron population is attached to each local patch $m$	

TABLE 2: Result of face recognition accuracy using CD classifier with Supervised STDP and  $\tau_s$  Prediction.

Datasets	Accuracy (%)				
	Baseline	Supervised STDP		$\tau_s$ Prediction	
		Train	Test	Train	Test
AR Neutral S1	82.80	94.00 $\pm$ 0.90	<b>93.40 <math>\pm</math> 1.07</b>	93.10 $\pm$ 0.99	92.80 $\pm$ 0.92
AR Smile S1	<b>100.0</b>	100.0 $\pm$ 0.00	<b>100.0 <math>\pm</math> 0.00</b>	100.0 $\pm$ 0.00	<b>100.0 <math>\pm</math> 0.00</b>
AR Angry S1	93.60	99.10 $\pm$ 0.56	<b>99.20 <math>\pm</math> 0.42</b>	99.10 $\pm$ 0.74	98.90 $\pm$ 0.88
AR Scream S1	74.00	90.90 $\pm$ 0.74	90.70 $\pm$ 0.67	93.10 $\pm$ 1.20	<b>92.90 <math>\pm</math> 1.45</b>
AR Sunglasses S1	56.10	98.20 $\pm$ 1.03	98.20 $\pm$ 1.03	98.30 $\pm$ 0.82	<b>98.40 <math>\pm</math> 0.70</b>
AR Scarf S1	7.80	94.10 $\pm$ 0.99	92.70 $\pm$ 1.34	95.50 $\pm$ 0.85	<b>95.30 <math>\pm</math> 0.67</b>
JAFFE	88.68	94.76 $\pm$ 0.90	93.68 $\pm$ 0.95	96.26 $\pm$ 0.59	<b>94.94 <math>\pm</math> 0.89</b>
<i>fb</i>	89.13	96.83 $\pm$ 0.73	96.62 $\pm$ 0.71	97.27 $\pm$ 0.60	<b>97.26 <math>\pm</math> 0.61</b>
Average	74.01	95.98 $\pm$ 0.73	95.56 $\pm$ 0.77	96.58 $\pm$ 0.73	<b>96.31 <math>\pm</math> 0.76</b>

to avoid overtraining the synaptic time constant  $\tau_s$ , where the train error  $e_{rate} = 0.08$  and test error  $e_{rate} = 0.08$  are obtained. By comparing Figures 11(a) and 11(b) the convergence to minimum recognition error is achieved faster in  $\tau_s$  Prediction for all types of spike codings. Similarly, from Figure 11(c), the spike error converges to minimum faster in  $\tau_s$  Prediction than in Supervised STDP.

To investigate the convergence of both methods further, evolution of output spikes from a CD neuron receiving inputs from 2 populations of neurons sharing similar spatial location as reference and test during Supervised STDP Learning and  $\tau_s$  Prediction learning process is shown in Figure 12. The 2 populations of neurons are acquired from images belonging to the same subject (matching samples). Based on Figure 12,

both methods start the training by producing only small number of spikes at epoch 0. However, Supervised STDP learning requires more epochs (30 to 35 epochs) before the output spikes stabilize while  $\tau_s$  Prediction learning only requires 5 to 15 epochs in order to do so. Based on the convergence alone,  $\tau_s$  Prediction would be favorable since it trains and converges faster than Supervised STDP.

**5.4. Discriminants from Synaptic Time Constant.** The resulting trained synaptic time constant  $\tau_s$  can capture the underlying variations embedded within the training faces. We observed that in  $\tau_s$  Prediction learning the trained synaptic time constants values are lower at synaptic connections attached to face feature with lower importance. This conforms

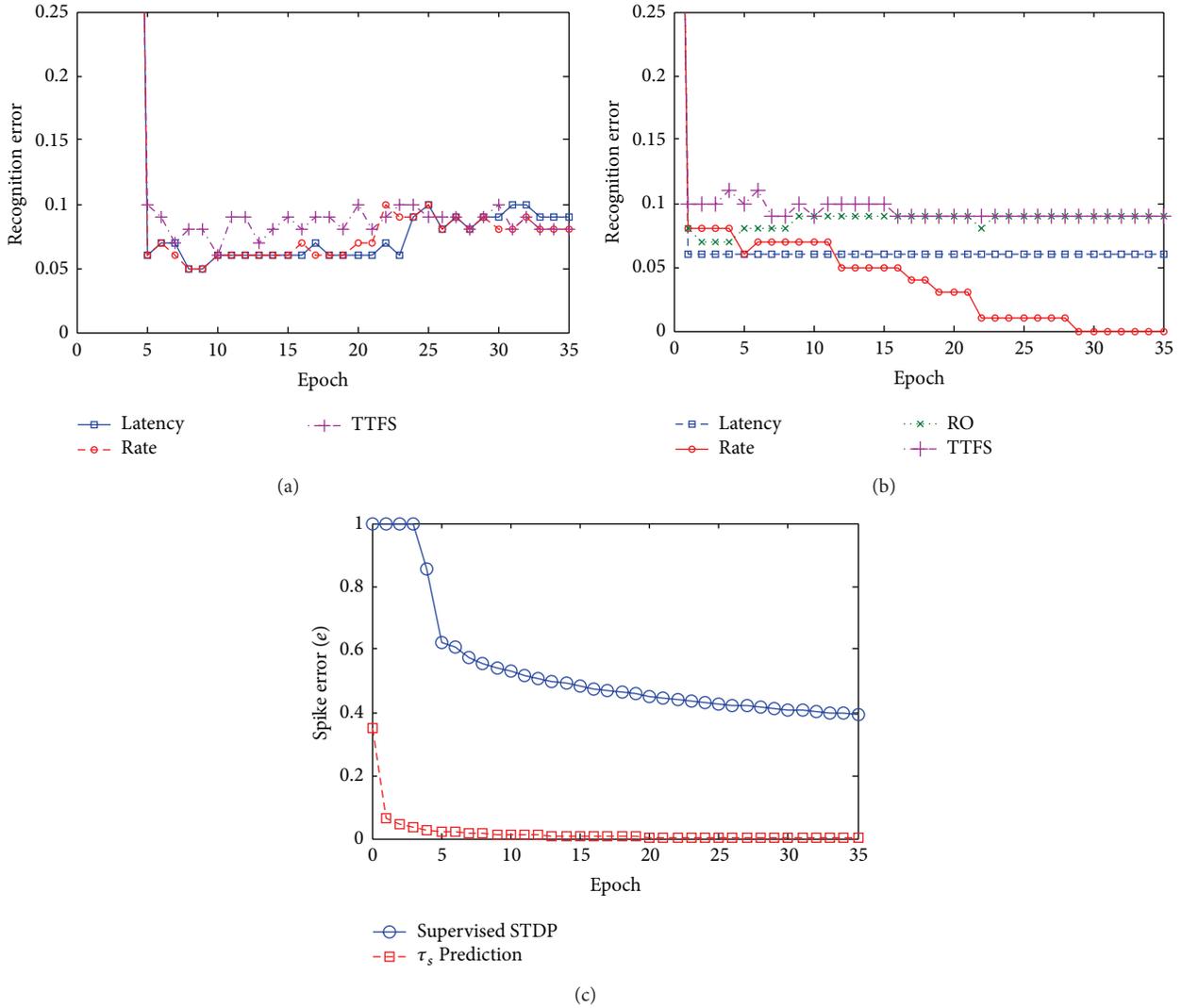


FIGURE 11: Convergence analysis on (a) Supervised STDP and (b)  $\tau_s$  Prediction using different spike codings (c) shows the comparison on spike error  $e$  of Supervised STDP and the  $\tau_s$  Prediction learning methods as the training iteration grows. The errors are taken as average of recognition error from AR Scream S1 and Scarf S1. The recognition error at  $e > 0.25$  in (a) and (b) is not shown for clarity of the graph. For Supervised STDP, recognition error for RO coding  $e_{RO}$  is not shown since  $e_{RO} = 1$  for all tested epochs.

to the initial objective of enforcing stricter conditions for spike firing on neurons attached to facial regions with high degree of similarity. This would ensure that only highly similar facial regions would cause firing in output neuron. The distribution of fully trained  $\tau_s$  is shown in Figure 13.

According to Figure 13, facial regions with low discrimination such as mouth in scream set receives higher values of fully trained  $\tau_s$  which signifies lower importance to final classification. This is used in accordance to feature selection strategy by locally rewarding or penalizing each local NCF obtained from CD classifier based on the computed discriminants  $\varphi_m$ .

5.5. Performance Comparison of Different Spike Codings. From results presented earlier, in terms of recognition accuracy and convergence, one learning method stands out from

the other.  $\tau_s$  Prediction delivers better recognition accuracy than Supervised STDP and trains faster too. Furthermore, there is a limitation on type of spike coding that could be used by Supervised STDP, where the rank-order coding does not deliver acceptable result. Thus, next analysis on performance of CD classifier using different spike codings to interpret the output spike is based only on CD classifier trained using  $\tau_s$  Prediction. Using similar experimental settings described in Section 5.2, the recognition accuracy is recorded in Table 3.

According to Table 3, on average, latency coding produces the best test result on test samples with 96.31% accuracy, followed by rate coding with 94.67% accuracy, RO coding with 94.30% accuracy, and TTFS coding with 93.42% accuracy. RO coding particularly works slightly better than latency coding in AR Scarf S1, where it produces 95.90% accuracy as opposed to 95.30% produced by latency coding. On the other hand,

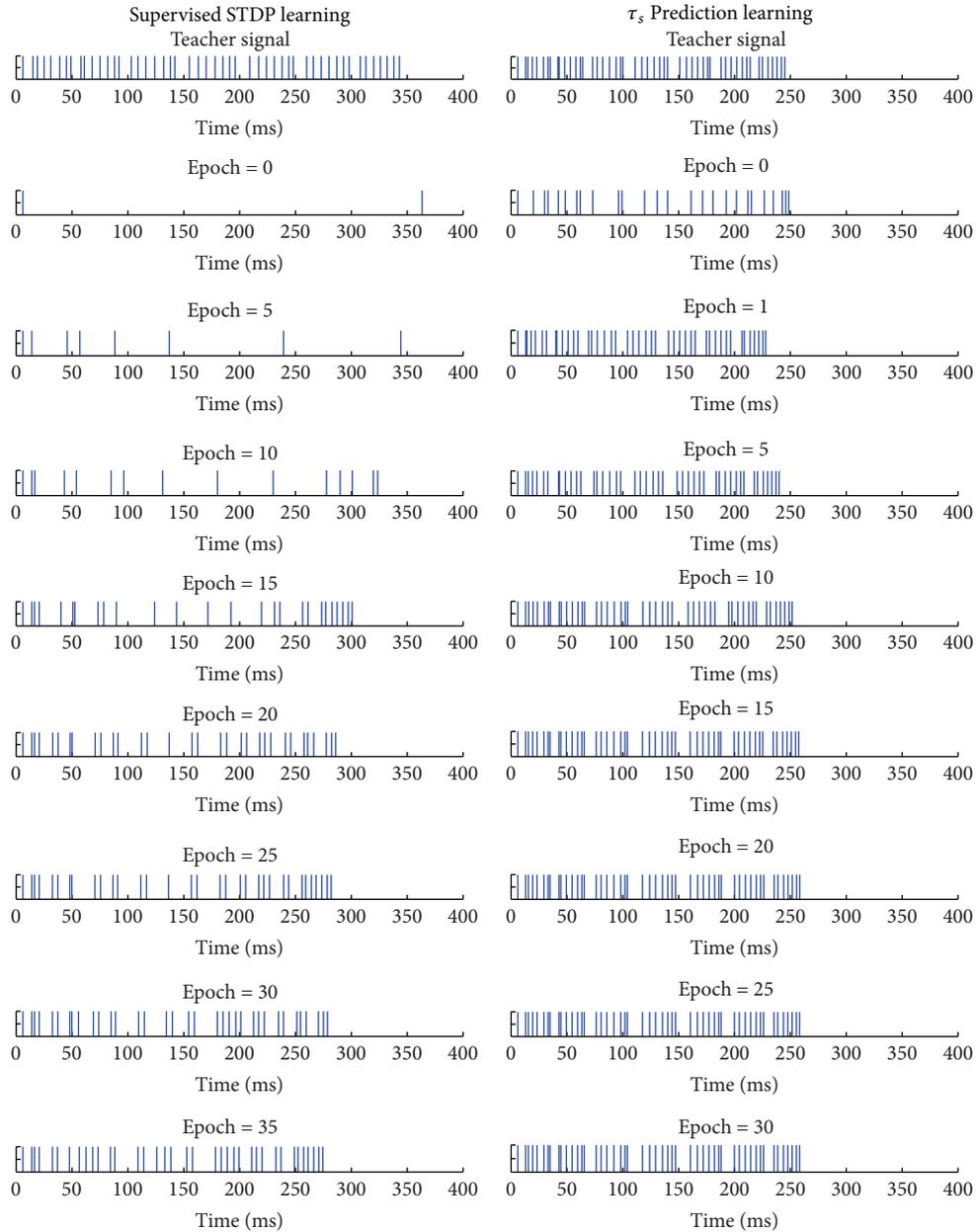


FIGURE 12: Evolution of output spikes from a CD neuron during Supervised STDP Learning and  $\tau_s$  Prediction learning process. The output spikes are shown as small bar plot indicating the exact output spike time in milliseconds. The input spikes are obtained from identical subject and the output spikes shown also include start and end cues but exclude the imaginary spikes.

considering that TTFs only uses the first output spike from each CD neuron, it delivers quite an impressive result, on average only lacks around 3% accuracy compared to latency coding.

In addition, in order to closely examine the interpretation of each spike coding on output spike distribution for both matching samples and nonmatching samples, 4 images of 2 subjects from AR Scream S1 and Scarf S1 are used. Each pair constitutes to 2 matching samples and 2 nonmatching samples, with each pair of image from the 4 different samples classified by fully trained CD classifier and the input

and output spike patterns are recorded. The input patterns and output spike interpretations of different spike codings are given in Figure 14. The codings are applied to outputs from each local population of input neurons (i.e., 144 local patches).

From Figure 14, the output spikes delays rely heavily on the coincidence of the presynaptic input sequence where matching samples appear to produce lower-delay spikes and vice versa. The rate of output spike firing is also higher in matching sample while slightly lower in nonmatching sample. In the plot, upper face parts are bound to neurons' afferents at

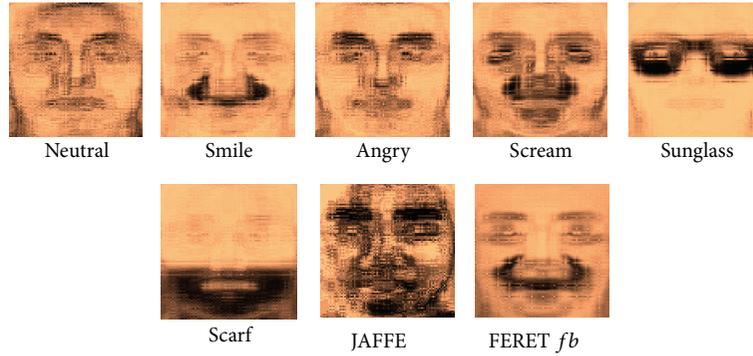


FIGURE 13: Result of learning by  $\tau_s$  prediction for several face datasets where each pixel in  $\mathcal{L}$  location within the images represents the synaptic time constant  $\tau_s$  of synaptic connection between CD neuron and input neuron attached to the  $\mathcal{L}$  location. Darker pixels represent higher  $\tau_s$  and vice versa. Number of epochs used is 1.

TABLE 3: Result of face recognition accuracy using CD classifier trained by  $\tau_s$  Prediction with different spike codings.

Datasets	Latency		Rate		Accuracy (%)		RO		TTFS	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
AR Neutral S1	93.10	<b>92.80</b>	93.40	92.20	92.60	92.70	91.30	91.10		
AR Smile S1	100.0	<b>100.0</b>	100.0	<b>100.0</b>	100.0	<b>100.0</b>	99.40	99.40		
AR Angry S1	99.10	<b>98.90</b>	99.50	98.50	97.80	98.00	96.40	95.80		
AR Scream S1	93.10	<b>92.90</b>	93.90	92.40	92.40	92.10	84.10	84.20		
AR Sunglasses S1	98.30	<b>98.40</b>	97.40	96.70	98.20	98.20	98.30	98.10		
AR Scarf S1	95.50	95.30	93.80	90.20	96.00	<b>95.90</b>	92.10	91.40		
JAFPE	96.26	<b>94.94</b>	98.12	91.31	95.83	94.35	92.78	91.70		
<i>fb</i>	97.27	<b>97.26</b>	95.98	96.05	83.22	83.18	95.70	95.65		
Average	96.58	<b>96.31</b>	96.51	94.67	94.51	94.30	93.76	93.42		

lower location, while lower face parts are attached to neurons' afferents at higher location. Note that, in both matching and nonmatching samples, the output spike delays and the spike counts are quite the same for upper afferent. However, at lower afferents, significant changes in delays and spike counts can be observed between matching and nonmatching samples. Nonmatching samples produce less spike counts and higher delay than matching sample at lower afferents. Since stricter condition is imposed on upper face part, it is much harder to evoke output spikes when the inputs actually belong to different subjects.

**5.6. Results on Face Recognition Using PCA and Gabor Features.** For the final experiment, we investigate the performance of our proposed CD classifier against several widely used classifiers. We use two popular feature representation approaches, namely, PCA and Gabor Wavelets, to represent the face. The PCA implementation follows Locally Lateral Subspace (LLS) strategy employed in [29] where the retained PCA features per local patch are 8. Local Gabor features on the other hand were acquired using approach adopted in [37] and the resulting Gabor features per local patches were further downsampled by a factor of 3. Soft  $k$ NN follows the

approach detailed in [30] while the soft SVM implementation follows the similar sum aggregation of ensembles of classifiers adopted by soft  $k$ NN [30]. SVM implementation uses LibSVM library with RBF kernel [38]. For AR, JAFPE, and FERET datasets, the trained CD classifier acquired in Section 5.2 was used, while, for CK+ dataset, 123 images at the beginning of first sequence are used as gallery while 577 peak images from each sequence are used as probe in training. Then, 4 most expressive images from each sequence resulting into a total of 2290 test images were used as test samples. The result of this experiment is given in Table 4.

According to Table 4, for PCA representation, CD classifier delivers the best result for all tested datasets, while, for Gabor representation, CD classifier gives best recognition accuracy except for CK+ dataset. On average, CD classifier is more than 5% and 11% better than soft  $k$ NN and soft SVM, respectively, in PCA representation. For Gabor representation, CD classifier is 2% and 13% better than soft  $k$ NN and soft SVM, respectively. The reason why the advantage of using CD classifier is more apparent in PCA representation rather than Gabor is due to the robustness of Gabor features against small spatial perturbations thus increasing the discriminations of facial features, while, in PCA, the noise due to variations

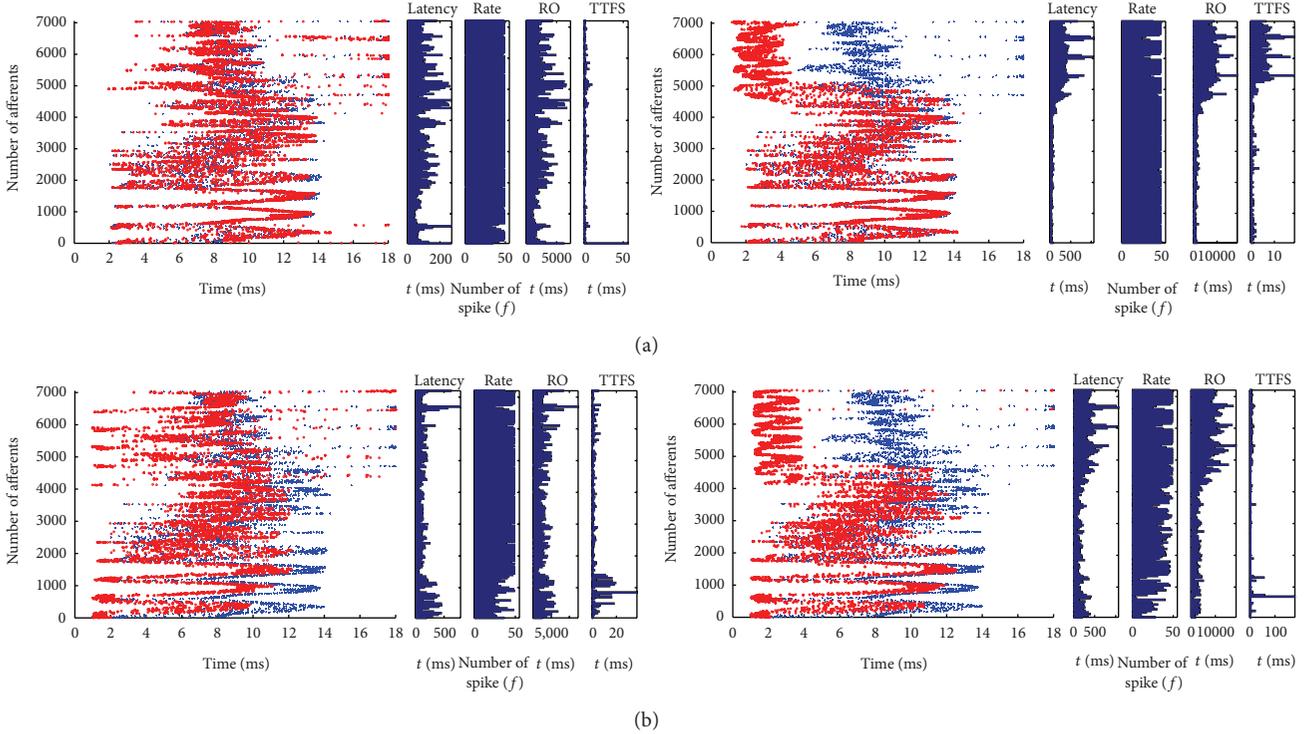


FIGURE 14: Different spike codings applied to output spikes of CD classifier fully trained by  $\tau_s$ . Prediction with epoch = 1, taking inputs from AR Scream S1 on the left and AR Scarf S1 on the right. Figures in row (a) are plotted from matching face samples, while in bottom row (b) they are from nonmatching face samples. Scatter plots are comprised of spatiotemporal input spike patterns of reference face (blue asterisk) and test face (red circle) from 7056 neurons afferents while bar plots on the right show the CD neurons latency, rate, rank order, and TTFS coding applied. The input spikes are obtained by direct conversion of pixel values to time domain which is then normalized to be between 0 and 18.

TABLE 4: Comparison of face recognition accuracy using CD classifier trained by  $\tau_s$ . Prediction against several other popular classifiers.

Face representations	Classifier	Accuracy (%)				
		AR	JAFFE	$fb$	CK+	Average
PCA	$k$ NN	55.55	83.66	83.09	94.32	79.16
	soft $k$ NN	77.09	92.08	92.97	96.86	89.75
	SVM	55.55	83.66	83.09	94.32	79.16
	soft SVM	61.00	88.12	87.11	95.19	82.86
	CD	<b>87.09</b>	<b>94.06</b>	<b>95.73</b>	<b>97.03</b>	<b>93.48</b>
Gabor	$k$ NN	58.73	89.60	87.03	98.21	83.93
	soft $k$ NN	84.00	94.55	96.65	<b>99.26</b>	93.62
	SVM	58.64	76.23	86.95	98.12	79.99
	soft SVM	58.82	87.13	87.28	98.17	82.85
	CD	<b>91.18</b>	<b>95.54</b>	<b>97.32</b>	99.13	<b>95.79</b>

is higher than Gabor features; thus CD classifier's ability to elevate the discriminations of PCA features is more obvious.

**5.7. Results on Other Multivariate Datasets.** To investigate the viability of the proposed CD classifier approach on multivariate data other than face images, another experiment is conducted using Iris dataset, Breast Cancer Wisconsin

TABLE 5: Comparison of classification accuracy of several multivariate datasets using  $k$ NN, SVM and CD classifier.

Classifier	Accuracy (%)			
	Iris	Wisconsin	Statlog	Average
$k$ NN	90.82 $\pm$ 3.5	89.04 $\pm$ 1.2	89.45	89.77
SVM	93.56 $\pm$ 2.3	90.12 $\pm$ 2.6	83.65	89.11
CD (latency)	<b>93.70 <math>\pm</math> 4.3</b>	<b>91.72 <math>\pm</math> 1.0</b>	<b>90.00</b>	<b>91.81</b>
CD (rate)	50.52 $\pm$ 4.7	85.38 $\pm$ 4.8	86.80	73.54
CD (RO)	90.07 $\pm$ 3.4	90.45 $\pm$ 1.5	87.15	89.22
CD (TTFS)	63.26 $\pm$ 6.5	72.44 $\pm$ 6.9	40.65	58.78

(Diagnostic) dataset, and Statlog (Landsat Satellite) dataset [39]. Iris dataset contains 3 classes of 50 vectors each, where each vector has dimension of 4. Wisconsin breast cancer dataset contains 569 vectors belonging to two classes, namely, "malignant" and "benign," where each vector has 32 elements. Statlog dataset contains the multispectral values of pixels in  $3 \times 3$  neighborhoods in a satellite image. There are 6435 vectors of 36 elements to be classified into 7 classes. Using 10-fold cross validations, 10 splits of training/testing are carried out, except for Statlog dataset since the training and test data are fixed, and the average and standard deviations are recorded in Table 5. CD classifier parameters used are  $w = 5$ ,  $v = 6$  mV,  $A_+ = 1.0$ ,  $\eta_{\text{spike}} = -2$  mV, and  $\eta_{\text{no spike}} = 2$  mV.

Training is done by  $\tau_s$ . Prediction learning method where epoch = 1 is used for classification by latency and rate coding, and epoch = 30 is used for classification by RO and TTFS coding.

According to results presented in Table 5, CD classifier with latency coding produces slightly superior result, which is around 2% better than  $k$ NN and SVM approach in all datasets, even though the local discrimination is not applicable since the variations within the data are not as generic as the variations found in face image and there is no clear indication on how to locally divide each piece of data into locally lateral vectors. Even if the division was done by assuming single element of the vector as a local vector, we found that no further improvements in classification can be achieved. Furthermore, variations are more random and even though the discrimination can be computed, the learnt discriminants of training data would not be able to faithfully represent the variations in the test set.

Additionally, from this experiment it is found that, for classification of multivariate data, latency coding works best with the average accuracy being 18%, 2%, and 33% better than rate, RO, and TTFS, respectively. The reason behind the inferiority of rate coding is the limitation on maximum encoding capacity,  $E_{\max} = 2^N$  of rate of firings due to relatively small number of variables when compared against the number of the samples; that is, for iris dataset only  $N = 4$  different variables were available for classification of 120 samples. In contrast, significantly better results achieved by rate coding on Wisconsin and Statlog rather than Iris dataset are due to higher number of variables, which are 32 and 36 variables, respectively, thus increasing the rate-of-firing's maximum encoding capacity. Meanwhile, RO coding is just slightly inferior to latency coding. On the other hand, worst average performance is produced by TTFS coding since it failed to capture the underlying similarities between the probe and the gallery due to only one spike per CD neuron (first spike) being considered in this type of coding.

## 6. Conclusions and Future Works

In this paper, a classifier based on SNN is proposed, namely, coincidence detection (CD) classifier, where two learning methods used to train CD classifier are also presented. A method of optimizing the discrete-time Spike Response Model (SRM) by predicting the output spike time is also discussed in details. We found that our proposed Output Spike Time Prediction (OSTP) method can produce output spike pattern from input pair identical to discrete SRM but with significantly lower floating operations and much faster processing time, with an average of 12.82 seconds as opposed to  $18.75 \times 10^4$  seconds in discrete-time SRM for all tested discrete intervals. Besides, we showed that coincidence detection can capture the degree of synchronization between two presynaptic inputs by producing lower-delay output spikes for more synchronized input pairs and vice versa. While CD classifier can produce spike based on the coincidence of inputs, the closeness between the inputs that will trigger the

output spike is explicitly determined by the training process of learning parameter  $\tau_s$ .

In addition, CD classifier trained with  $\tau_s$ . Prediction delivered comparable performance to Supervised STDP; however it can achieve convergence faster with less number of epochs required. We found that latency coding produced best recognition accuracy at 96.31% but its performance is not too far from other spike codings. Furthermore, the distribution of discriminants derived from the learning parameters revealed the ability of  $\tau_s$ . Prediction learning to capture the underlying variation within the training faces. Further investigation on the performance of CD classifier using PCA and Gabor features showed that our proposed method performs 5% and 11% better than soft  $k$ NN and soft SVM, respectively, in PCA representation, while as for Gabor representation it is 2% and 13% better than soft  $k$ NN and soft SVM, respectively. Besides, experiment on the feasibility of CD classifier on classifying other multivariate data revealed that CD classifier with latency coding is around 2% better than  $k$ NN and SVM classifiers. Additionally, for the tested multivariate data, latency coding delivers the best result which is 18%, 2%, and 33% better than rate, RO, and TTFS, respectively.

As for future work, we will explore the possibility of extending the application of proposed method into object recognition task and also for temporal recognition of faces from video sequences. We would further study how to embed the global information of face image together with local patches information so that the resulting classification is more robust against global variations such as poses, age variation, and illumination.

## Appendices

### A. Two Presynaptic Spikes Firing at the Same Time

Calculating from (3) to (9)

$$u_m(t) = \eta(t - \hat{t}_i) + \sum_j w_{ij} \sum_f \frac{s}{\tau_{s,ij}} \exp\left(1 - \frac{s}{\tau_{s,ij}}\right). \quad (\text{A.1})$$

Let  $\eta(t - \hat{t}_i) = \eta_0 = 0$ :

$$\begin{aligned} u_m(t) &= w \left[ \frac{t - t_1^f}{\tau_s} \cdot \exp\left(1 - \frac{t - t_1^f}{\tau_s}\right) \right] \\ &+ w \left[ \frac{t - t_2^f}{\tau_s} \cdot \exp\left(1 - \frac{t - t_2^f}{\tau_s}\right) \right], \end{aligned}$$

$$u_m(t) = 2w \left[ \frac{t - t^f}{\tau_s} \cdot \exp\left(1 - \frac{t - t^f}{\tau_s}\right) \right],$$

$$u_m = 2w [s \cdot \exp(1 - s)],$$

$$\frac{u_m}{2w} = s \cdot \exp(1 - s),$$

$$\begin{aligned}
\frac{u_m}{2w} &= s \cdot \frac{\exp(1)}{\exp(s)}, \\
\exp(s) \left( \frac{u_m}{2w} \right) &= s \cdot \exp(1), \\
\ln(\exp(s)) + \ln\left(\frac{u_m}{2w}\right) &= \ln(s) + \ln(\exp(1)), \\
s + \ln\left(\frac{u_m}{2w}\right) &= \ln(s) + 1, \\
\ln\left(\frac{u_m}{2w}\right) &= \ln(s) - s + 1, \\
\ln(s) - s + 1 - \ln\left(\frac{u_m}{2w}\right) &= 0.
\end{aligned} \tag{A.2}$$

## B. Proposed OSTP Equation

Complete calculation of OSTP equation (12) from (10):

$$\begin{aligned}
u_m(s) - \eta_0 &= w [s_1 \cdot \exp(1 - s_1)] \\
&+ w [s_2 \cdot \exp(1 - s_2)].
\end{aligned} \tag{B.1}$$

Then, let  $a = t_2^f - t_1^f$  and thus  $t_2^f = a + t_1^f$ . Consequently,  $s_2$  becomes

$$\begin{aligned}
s_2 &= \frac{t - t_1^f - a}{\tau_s} - \frac{a}{\tau_s}, \\
s_2 &= \frac{t - t_1^f}{\tau_s} - \frac{a}{\tau_s}, \\
s_2 &= s_1 - \frac{a}{\tau_s}, \\
u - \eta_0 &= w [s_1 \cdot \exp(1 - s_1)] \\
&+ w \left[ \left( s_1 - \frac{a}{\tau_s} \right) \cdot \exp\left(1 - \left( s_1 - \frac{a}{\tau_s} \right)\right) \right], \\
\frac{u - \eta_0}{w} &= s_1 \cdot \exp(1 - s_1) + \left( s_1 - \frac{a}{\tau_s} \right) \\
&\cdot \exp\left(1 - \left( s_1 - \frac{a}{\tau_s} \right)\right), \\
\frac{u - \eta_0}{w} &= s_1 \cdot \exp(1 - s_1) + s_1 \cdot \exp\left(1 - s_1 + \frac{a}{\tau_s}\right) \\
&- \frac{a}{\tau_s} \cdot \exp\left(1 - s_1 + \frac{a}{\tau_s}\right), \\
\frac{u - \eta_0}{w} &= s_1 \cdot \exp(1 - s_1) + s_1 \cdot \exp\left(1 - s_1 + \frac{a}{\tau_s}\right) \\
&- \frac{a}{\tau_s} \cdot \exp\left(1 - s_1 + \frac{a}{\tau_s}\right),
\end{aligned}$$

$$\begin{aligned}
\frac{u - \eta_0}{w} &= s_1 \cdot \exp(1) \cdot \exp(-s_1) + s_1 \cdot \exp(1) \\
&\cdot \exp(-s_1) \cdot \exp\left(\frac{a}{\tau_s}\right), \\
\frac{u - \eta_0}{w} &= s_1 \cdot \exp(-s_1) \cdot \exp(1) \left[ 1 + \exp\left(\frac{a}{\tau_s}\right) \right] \\
&- \exp(-s_1) \left[ \frac{a}{\tau_s} \cdot \exp\left(1 + \frac{a}{\tau_s}\right) \right].
\end{aligned} \tag{B.2}$$

Let  $A = \exp(1)[1 + \exp(a/\tau_s)]$  and  $B = [(a/\tau_s) \cdot \exp(1 + a/\tau_s)]$ :

$$\begin{aligned}
\frac{u - \eta_0}{w} &= A \cdot s_1 \cdot \exp(-s_1) - B \cdot \exp(-s_1), \\
\frac{u - \eta_0}{w} &= \exp(-s_1) (A \cdot s_1 - B), \\
\exp(-s_1) (A \cdot s_1 - B) - \frac{u - \eta_0}{w} &= 0.
\end{aligned} \tag{B.3}$$

## C. Approximating Synaptic Time Constant from SRM Model

Complete calculation from (10) to (15):

$$\begin{aligned}
u_m(t) - \eta_0 &= w \left[ \frac{t - t_1^f}{\tau_s} \cdot \exp\left(1 - \frac{t - t_1^f}{\tau_s}\right) \right] \\
&+ w \left[ \frac{t - t_2^f}{\tau_s} \cdot \exp\left(1 - \frac{t - t_2^f}{\tau_s}\right) \right], \\
\frac{u - \eta_0}{w} &= \frac{1}{\tau_s} \left[ (t - t_1^f) \cdot \exp\left(1 - \frac{t - t_1^f}{\tau_s}\right) + (t - t_2^f) \right. \\
&\cdot \exp\left(1 - \frac{t - t_2^f}{\tau_s}\right) \left. \right], \\
\frac{u - \eta_0}{w} &= \frac{1}{\tau_s} \left[ (t - t_1^f) \right. \\
&\cdot \exp(1) \exp\left(-\left(t - t_1^f\right) \left(\frac{1}{\tau_s}\right)\right) + (t - t_2^f) \\
&\cdot \exp\left(-\left(t - t_2^f\right) \left(\frac{1}{\tau_s}\right)\right) \left. \right].
\end{aligned} \tag{C.1}$$

Let  $X = 1/\tau_s$ , and

$$\begin{aligned}
\frac{u - \eta_0}{w} &= X \left[ (t - t_1^f) \cdot \exp(1) \exp\left(-\left(t - t_1^f\right) (X)\right) \right. \\
&+ (t - t_2^f) \cdot \exp\left(-\left(t - t_2^f\right) (X)\right) \left. \right], \\
X \left[ (t - t_1^f) \cdot \exp(1) \exp\left(-\left(t - t_1^f\right) (X)\right) + (t - t_2^f) \right. \\
&\cdot \exp\left(-\left(t - t_2^f\right) (X)\right) \left. \right] - \frac{u - \eta_0}{w} &= 0.
\end{aligned} \tag{C.2}$$

## D. Proof of Minimum $\tau_s$ Required to Evoke a Spike

Given a presynaptic input pair  $t_1^f$  and  $t_2^f$ , where  $t_2^f > t_1^f$ , and that the threshold  $v$  would make the firing possible, that is,  $v \leq \sum w$ , then we can let  $\tau_s \leq t_2^f - t_1^f$ . To prove this, based on (10) assuming that the spike would fire at time  $t = t_d$ , where  $t_d \approx t_2^f$ , thus (10) becomes

$$u_m(t_d) - \eta_0 = w \left[ \frac{t_d - t_1^f}{\tau_s} \cdot \exp\left(1 - \frac{t_d - t_1^f}{\tau_s}\right) \right] + w \left[ \frac{t_d - t_2^f}{\tau_s} \cdot \exp\left(1 - \frac{t_d - t_2^f}{\tau_s}\right) \right]. \quad (D.1)$$

Since  $t_d \approx t_2^f$ , the second term on the right is almost 0. In order to ensure definitely that a spike would be evoked,  $u_m(t_d) - \eta_0$  should be at least equal to  $w$ ; thus by setting  $u_m(t_d) - \eta_0 = w$

$$w = w \left[ \frac{t_d - t_1^f}{\tau_s} \cdot \exp\left(1 - \frac{t_d - t_1^f}{\tau_s}\right) \right], \quad (D.2)$$

$$\left[ \frac{t_d - t_1^f}{\tau_s} \cdot \exp\left(1 - \frac{t_d - t_1^f}{\tau_s}\right) \right] = 1.$$

Taking  $\tau_s = t_2^f - t_1^f$ , then  $\tau_s \approx t_d - t_1^f$  so that

$$\left[ \frac{\tau_s}{\tau_s} \cdot \exp\left(1 - \frac{\tau_s}{\tau_s}\right) \right] = 1. \quad (D.3)$$

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

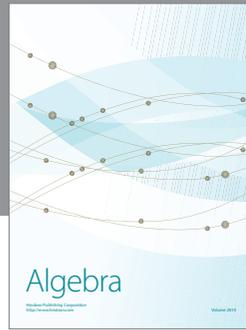
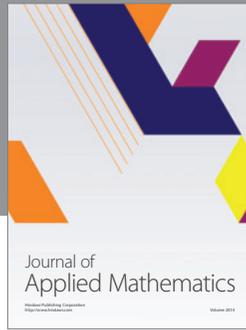
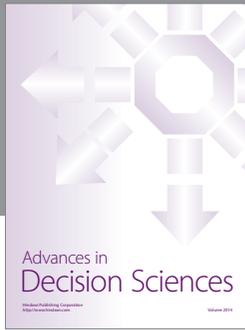
## Acknowledgments

The authors would like to thank Dr. Aleix M. Martinez for providing the AR Face Dataset. Some portions of the research in this paper also use Extended Yale Face Database B. The work presented here is cooperatively sponsored by the Ministry of Higher Education Malaysia, under MyRA Incentive Grant Scheme (MIRGS) no. MIRGS13-02-001-0001 and University of Technology MARA, Malaysia.

## References

- [1] S. A. Roy and K. D. Alloway, "Coincidence detection or temporal integration? What the neurons in somatosensory cortex are doing," *The Journal of Neuroscience*, vol. 21, no. 7, pp. 2462–2473, 2001.
- [2] J. G. Carlton and P. J. Cullen, "Coincidence detection in phosphoinositide signaling," *Trends in Cell Biology*, vol. 15, no. 10, pp. 540–547, 2005.
- [3] C. J. Tabone and M. Ramaswami, "Is NMDA receptor-coincidence detection required for learning and memory?" *Neuron*, vol. 74, no. 5, pp. 767–769, 2012.
- [4] W. Maass, "Computing with spiking neurons," in *Pulsed Neural Network*, W. Maass and C. M. Bishop, Eds., pp. 55–85, MIT Press, Cambridge, Mass, USA, 1999.
- [5] I. Sporea and A. Grüning, "Supervised learning in multilayer spiking neural networks," *Neural Computation*, vol. 25, no. 2, pp. 473–509, 2013.
- [6] M. Abeles, "Role of the cortical neuron: integrator or coincidence detector?" *Israel Journal of Medical Sciences*, vol. 18, no. 1, pp. 83–92, 1982.
- [7] P. König, A. K. Engel, and W. Singer, "Integrator or coincidence detector? The role of the cortical neuron revisited," *Trends in Neurosciences*, vol. 19, no. 4, pp. 130–137, 1996.
- [8] R. C. DeCharms and A. Zador, "Neural representation and the cortical code," *Annual Review of Neuroscience*, vol. 23, pp. 613–647, 2000.
- [9] W. M. Usrey, J. M. Alonso, and R. C. Reid, "Synaptic interactions between thalamic inputs to simple cells in cat visual cortex," *The Journal of Neuroscience*, vol. 20, no. 14, pp. 5461–5467, 2000.
- [10] Y. Chen, H. Zhang, H. Wang, and L. Yu, "The role of coincidence-detector neurons in the reliability and precision of subthreshold signal detection in noise," *PLoS ONE*, vol. 8, no. 2, Article ID e56822, 2013.
- [11] S. M. Bohte, J. N. Kok, and H. L. Poutre, "SpikeProp: backpropagation for networks of spiking neurons," in *Proceedings of the 8th European Symposium on Artificial Neural Networks (ESANN '00)*, pp. 419–424, Bruges, Belgium, April 2000.
- [12] S. M. Bohte, J. N. Kok, and H. L. Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, pp. 17–37, 2002.
- [13] F. Ponulak, *ReSuMe—New Supervised Learning Method for Spiking Neural Networks*, Institute of Control and Information Engineering, Poznan University of Technology, 2005.
- [14] R. V. Florian, "The chronotron: a neuron that learns to fire temporally precise spike patterns," *PLoS ONE*, vol. 7, no. 8, Article ID e40233, 2012.
- [15] Y. Xu, X. Zeng, L. Han, and J. Yang, "A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks," *Neural Networks*, vol. 43, pp. 99–113, 2013.
- [16] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "Training spiking neural networks to associate spatio-temporal input-output spike patterns," *Neurocomputing*, vol. 107, pp. 3–10, 2013.
- [17] J.-P. Pfister, D. Barber, and W. Gerstner, "Optimal hebbian learning: a probabilistic point of view," in *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003*, vol. 2714 of *Lecture Notes in Computer Science*, pp. 92–98, Springer, Berlin, Germany, 2003.
- [18] J.-P. Pfister, T. Toyozumi, D. Barber, and W. Gerstner, "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning," *Neural Computation*, vol. 18, no. 6, pp. 1318–1348, 2006.
- [19] A. Carnell and D. Richardson, "Linear algebra for time series of spikes," in *Proceedings of the 13th European Symposium on Artificial Neural Networks*, pp. 363–368, 2005.
- [20] A. Belatreche, L. P. Maguire, M. McGinnity, and Q. X. Wu, "A method for supervised training of spiking neural networks," in *Proceedings of the IEEE Cybernetics Intelligence—Challenges and Advances (CICA '03)*, pp. 39–44, Reading, UK, 2003.
- [21] B. Schrauwen, *Towards applicable spiking neural networks [Doctorate]*, Electronics and Information Systems, Ghent University, Ghent, Belgium, 2008.

- [22] W. Gerstner and W. M. Kistler, *Spiking Neuron Models*, Cambridge University Press, Cambridge, UK, 2002.
- [23] W. Gerstner and W. M. Kistler, "Mathematical formulations of Hebbian learning," *Biological Cybernetics*, vol. 87, no. 5-6, pp. 404–415, 2002.
- [24] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: taming the beast," *Nature Neuroscience*, vol. 3, pp. 1178–1183, 2000.
- [25] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.
- [26] T. Makino, "A discrete-event neural network simulator for general neuron models," *Neural Computing & Applications*, vol. 11, no. 3-4, pp. 210–223, 2003.
- [27] R. Brette, "Exact simulation of integrate-and-fire models with synaptic conductances," *Neural Computation*, vol. 18, no. 8, pp. 2004–2027, 2006.
- [28] I. Sporea and A. Grüning, "Reference time in SpikeProp," in *Proceedings of the International Joint Conference on Neural Network (IJCNN '11)*, pp. 1090–1092, August 2011.
- [29] F. K. Zaman, A. A. Shafie, and Y. M. Mustafah, "Face recognition robust against expressions and partial occlusions," *International Journal of Automation and Computing*. In press, <http://www.ijac.net/EN/abstract/abstract1650.shtml>.
- [30] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft  $\kappa$ -NN ensemble," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 875–886, 2005.
- [31] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Face recognition from a single image per person: a survey," *Pattern Recognition*, vol. 39, no. 9, pp. 1725–1745, 2006.
- [32] A. M. Martinez and R. Benavente, "The AR face database," CVC Technical Report 24, Purdue University, 1998.
- [33] M. J. Lyons, M. Kamachi, and J. Gyoba, "Japanese Female Facial Expressions (JAFFE)," Database of Digital Images, 1997.
- [34] P. Jonathon Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [35] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): a complete dataset for action unit and emotion-specified expression," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '10)*, pp. 94–101, June 2010.
- [36] E. Nordlie, M.-O. Gewaltig, and H. E. Plesser, "Towards reproducible descriptions of neuronal network models," *PLoS Computational Biology*, vol. 5, no. 8, Article ID e1000456, 18 pages, 2009.
- [37] Y. Su, S. Shan, X. Chen, and W. Gao, "Hierarchical ensemble of global and local classifiers for face recognition," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1885–1896, 2009.
- [38] C.-C. Chang and C.-J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.
- [39] K. Bache and M. Lichman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Sciences, Irvine, Calif, USA, 2013.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

