Dissertation for Doctor of Philosophy

Predictive Learning with Generative Modeling

Muhammad Aasim Rafique

School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

2018

박 사 학 위 논 문

생성 모델을 위한 예측 학습

무하마드 아심 라피크

전 기 전 자 컴 퓨 터 공 학 부

광 주 과 학 기 술 원

2018

Predictive Learning with Generative Modeling

Advisor: Moongu Jeon

by

Muhammad Aasim Rafique

School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the School of Electrical Engineering and Computer Science.

Gwangju, Republic of Korea

October 12, 2017

Approved by

Professor Moongu Jeon

Thesis Advisor

Predictive Learning with Generative Modeling

Muhammad Aasim Rafique

Accepted in partial fulfillment of the requirements for the

degree of Doctor of Philosophy

October 12, 2017

Thesis Advisor _____

Prof. Moongu Jeon

Committee Member _____

Prof. Byung-geun Lee

Committee Member _____

Prof. Chang Wook Ahn

Committee Member _____

Prof. Jong Won Shin

Committee Member _____

Prof. Witold Pedrycz

Dedicated to my teachers.

PhD/EC 20132072 Muhammad Aasim Rafique. Predictive Learning with Generative Modeling . School of Electrical Engineering and Computer Science. 2018. 109p. Advisor: Prof. Moongu Jeon.

Abstract

Artificial neural networks are inspiring many state-of-the-art solutions to the problems in computer vision and machine learning. Generative modeling has gained popularity recently for their unsupervised nature. Although neural networks are taking lead in all sort of machine learning problems, the theory of the neural networks does not provide sufficient understanding for their success. They produce competitive results provided that the domain expertise and network knowledge are put together and formally device a predictive learning system for a particular problem. This thesis articulates various strategies to use generative modeling as a tool for predictive leaning. Some of the many problems which can efficiently and effectively make use of predictive learning are discussed in detail in this thesis are background subtraction and foreground segmentation in videos, speech recognition using a real memristor profile as a synapse in neural network and a vehicle license plate detection system using state-of-the-art region based convolutional neural networks. The concepts discussed are not limited to the discussed datasets.

PhD/EC 20132072 무하마드아심라피크. 생성 모델을 위한 예측 학습. 전기전자컴퓨터공학부. 2018. 109쪽. 지도교수: 전문구.

국 문 요 약

최근 딥 러닝으로 대표되는 기술의 발전에 힘입어 컴퓨터 비전과 머신 러닝에 대한 문제를 푸는 중요한 테크닉이 되었습. 최근에 생성 모델은 비지도 학습처럼 학습이 가능하게 되면서 인기가 많아 졌습니다. 인공신경망이 거의 모든 머신 러닝 문제에서 쓰이고 있지만, 인공 신경망의 성공은 이론적으로 설명이 불가능 하다. 도메인 전문 지식과 네트워크 지식이 결합되면 경쟁력있는 결과를 산출합니다. 이 논문은 생성 모델을 도구로서 예측 학습을 위해 다양한 전략을 제시합니다. 본 논문에서는 효율적으로 효과적으로 예측 학습을 사용할 수 있는 많은 문제들 중 일부를 비디오의 배경 빼기 및 전경 세그멘테이션, 실제 멤리스터 프로파일을 신경 네트워크의 시냅스로 사용하는 음성 인식 및 차량 번호판 감지 시스템은 최첨단 영역 기반의 컨벌루션 뉴럴 네트워크를 사용합니다. 이런 개념들은 언급된 데이터 셋에 한정되어 있지 않습니다.

Acknowledgements

I am grateful to Almighty Allah for his limitless blessings, and his divine guidance to take me through the immensely versed experience at machine learning and vision lab. I express my gratitude to my supervisor Prof. Moongu Jeon. It is my privilege that I get this chance as a student to benefit from his commendable wisdom, piercing insight, keen professionalism and kind mentor. His devotion toward his student's research helped me to achieve this landmark in my life.

It is one of the great pleasures of my stay at the institute is to have the company of my lab fellows. They inspired me when I needed an inspiration and they pat my back when I needed a push. I thank them for all the moments we spend together, for all the valuable discussions and their precious time they spared for me. There are many names but I would like to mention Dr. Muqeem Ahmad Sheri for his guidance and help. I also want to thank Dr. Witold Pedrycz and Dr. Byung-Geun Lee for helping me with the manuscripts I have published.

I would like to thank my grand parents, my brothers and my sister and their families for praying for me and showing confidence in me. My nephews and nice for keep pushing me to pursue my work harder. Thank you, Paa Shahbaz(late), Kashif bahi, Hajra Bhabi, Naila, Ahmed Bahi, and Bacha party.

This job could not have been possible without the continuous support of my wife and our son and daughter. They have been the energy for working hard during all my studies. Thank you, Hina, Ahsan, and Adeena.

Finally, I want to thank my parents for all they have done to make me what I am

today. Their untiring efforts and love made me stand firm for this achievement. Thank you, Aba G and Ammi G.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

"It has been suggested that the term learning defies precise definition because it is put to multiple uses. Learning is used to refer to (1) the acquisition and mastery of what is already known about something, (2) the extension and clarification of meaning of one's experience, or (3) an organized, intentional process of testing ideas relevant to problems. In other words, it is used to describe a product, a process, or a function."

-–From Learning How to Learn: Applied Theory for Adults by R.M. Smith

The definition of learning has significant importance as humans cross-reference the abilities of artificial intelligence with this. The capacity to learn by an artificial system is tested primarily on terms defined by R.M. Smith in a quotation above. The artificial system, whether it be in the hardware or software, is there, to help humans with their ongoing tasks. It may be, something basic like a dumb robot to a delicate task like an intelligent companion. The tasks may be repetitive in nature or demand ambitions. Learning process usually initiates with observation, and the artificial intelligence can emulate the process using generative modeling. Knowledge is a result formed out of the observations which are worked out as a predictive learning process.

Humans through the ages have always been inspired by nature and have tried to emulate the principles and mechanics revealed by it. Artificial intelligence researchers have been inspired by the workings of the brains as the main component of human in-

telligence. The brains physiology has been known for over a century, but the structure of learning processes in the brain has been discovered very recently in the last 40-50 years. It is debatable to claim if state-of-the-art neural networks simulate the processes of a brain, but most of the scientists agree that the artificial neural networks somehow emulate the neuronal models in the brain. Human beings possess key instincts observation and relating the observations, which can be articulated as generative modeling and using it for predictive learning.

Learning, in general, is a process of acquiring knowledge which is represented by various permutations of observations, and the representation can be assessed in a hierarchal process. The components may vary from handcrafted features to the automated learning processes in machine learning. In specific, neural networks are simple in structure and the hierarchies are designed with similar components called neurons (nodes) and the two stages in the hierarchy are connected with edges (synapses). The neurons are the decision makers while the synapses are the impact of that decision and the whole network learns what impact a decision maker should assert on the final decision. Although there is no conclusive theoretical evidence why deep learning performs state-of-the-art, the speculations are the brain like hierarchal nature of the system breaks down the complex problems in machine learning into basic elements and use the permutations of basic elements for the defined task. So, the state-of-the-art relies on experimentation and considering the fact that there is no free lunch, some experiments work well with certain problems while other behave well in a different problem. Hence there is always a need to workout a viable solution to a specific problem using

artificial neural networks.

1.1 Contribution of this thesis

The most significant research contribution of this thesis is the use of neural networks for generative modeling and predictive learning. The three data sets and the respective neural networks used are discussed in this thesis as follows:

1. Background subtraction is an image segmentation problem and this thesis presents a novel technique of using Restricted Boltzmann Machine (RBM) to solve it.

2. Speech recognition with neuromorphic devices is a challenging problem due to the limitation of computation resources. In this thesis, a hybrid neuromorphic system is discussed which uses memristor as a synapse [72].

3. Vehicle license plate detection is a common problem in computer vision and this thesis details a region generation technique to detect vehicle license plates in images and videos [74].

1.2 Summary of remaining chapters

*Chapter 2: Generative modeling and predictive learning.*This chapter gives the basic information about predictive learning and generative modeling. The chapter also contains an introduction to Restricted Boltzmann Machine (RBM) and its variant Gaussian-Bernoulli Restricted Boltzmann Machines (GRBM). GRBM is different from the Restricted Boltzmann Machine (RBM) by using real numbers as inputs, resulting

in a constrained Mixture of Gaussians (MoG), which is one of the most widely used techniques to solve problems with real value inputs.

Chapter 3: Background subtraction. This chapter presents an introduction to the background subtraction problem and the proposed solution using ANN. The background subtraction is an important technique in computer vision which segments moving objects in video sequences by comparing each new frame with a learned background model. The background subtraction method is explored using GRBM. A simple technique to reconstruct the learned background model from a given input frame and to extract the foreground from the background using the variance learned for each pixel is detailed in this chapter. Furthermore, the detailed results are also included in the end of this chapter.

Chapter 4: Hybrid neuromorphic speech recognition. This chapter details the architecture of a multilayer neural network, equipped with weight values represented by two memristors synapses, for speech recognition. The discussed neuromorphic neural network is a hybrid system which uses a Gaussian-Bernoulli Restricted Boltzmann Machine (GRBM) to transform the speech data in to sparse encoded binary data. The sparse data is used to train a Restricted Boltzmann Machine (RBM), and a two PCMO RRAM memristors synapse is used as a connection between the two layers of the RBM. The simulations are performed with the real memristor's behavioral data, for potentiation and depotentiation, to adjust learn-able parameters of the neuromorphic RBM.

Chapter 5: Vehicle license plate detection. This chapter presents vehicle license

plate (LP) detection problem and includes a new approach to solving this problem by treating the vehicle LP as an object. The primary focus of this chapter is to address following tasks associated with LP detection challenge: 1) LP detection in every single frame of a video sequence, 2) detection of partial LPs and 3) detection of LPs with moving cameras and moving vehicles. The state-of-the-art object detection techniques including convolutional neural networks with region proposal (RCNN), its successors (Fast-RCNN and Faster-RCNN) and the exemplar-SVM are used in this work to provide solutions to the problem.

Chapter 6: Conclusion. This chapter details a brief summary of the work presented in this thesis and discuss some possible future directions of research in this area.

Chapter 2

Generative modeling and predictive learning

2.1 Generative modeling

Intelligence is a context-aware phenomenon where observation plays a key role. There are many factors and hierarchies of observations, for example in a visual experience one observes the color, brightness, objects, positions, and so on in the scene. Similarly, in a hearing experience, one observes pitch, loudness, expression, tone, and so on. The factors or features, if we write in ordinary machine learning term, have global and local importance. One should be able to distinguish them well and by distinguish means knowing something in advance by having a representation of the features. It is a phenomenon of observing and recalling. Recalling is a somewhat close definition of regenerating something. Hence, generative modeling is a process of regenerating something which is observed earlier.

Generative modeling plays a vital role in modern machine learning techniques where we need observations independently. In layman terms, generative modeling tries to learn the true representation of the given data. Formally stated, if we have a distribution $P(X, y)$, where X is the observed data and y is the target or label, then the generative modeling learn the P' which is close to the true distribution. An example of comparing

the two distributions is KL-divergence $K[P||P']$ and it is solved to minimizing the following cross-entropy:

$$-\mathbb{E}_{x \in p'} log P(x) \tag{2.1}$$

There are many generative modeling techniques available in literature such as Gaussian mixture models, Hidden Markov Models (HMM), average one-depend estimation, RBM, GAN and so on. RBM and GAN are widely used in recent state-of-the-art using neural networks. RBM is used in this thesis with various datasets and is detailed in this chapter.

2.2 Predictive learning

Human beings expedite on the recalled observations and make decisions, and we call recalling as future and decision as a prediction. Predictive learning in machine learning is a similar phenomenon which predicts future extracting the information from events or stream of events from an environment.

Predictive learning a is adapted from neuroscience and is a key inspiration of an intelligent platform named as hierarchical temporal memories (HTM) [42]. It relates the artificial intelligence techniques mapping it on the human brain and the neuronal structures. The future is extracted from the different structures of neuronal patterns and their actions on events.

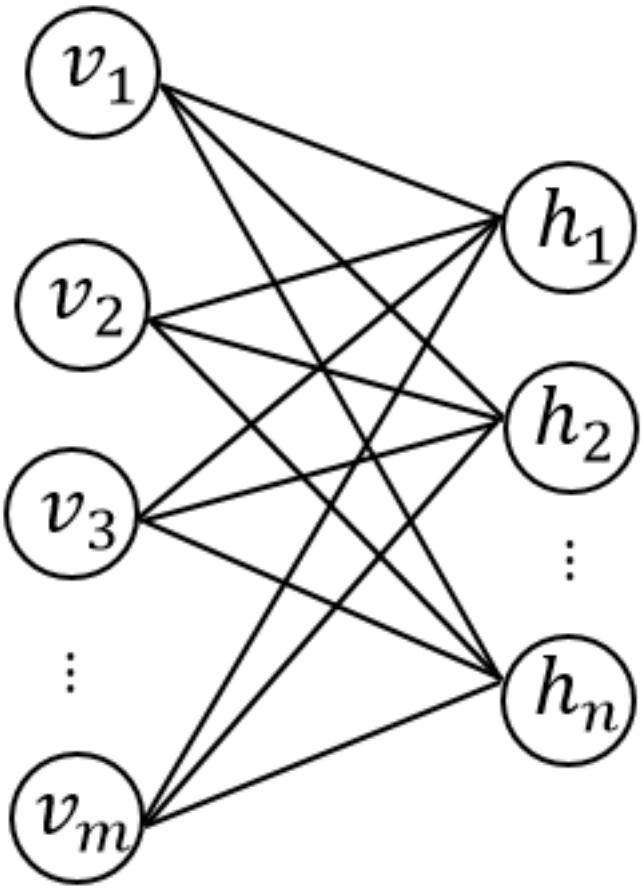
2.3 Restricted Boltzmann Machines

RBM [33] is a restricted version of the Boltzmann machine (BM), which is a special type of log-linear Markov Random Field (MRF), where hidden variables are introduced in addition to observed (visible) parameters to infer after learning from the training set. The restriction added to BM to form RBM is not to have visible-visible and hidden-hidden connections between neurons in the same layers, as depicted in Fig. 2.1(a). Visible layer neurons represent the observed data while hidden neurons are the representations to be learned from observed data. The energy function for a given state of RBM is defined [47] as:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{|\mathbf{v}|} a_i v_i - \sum_{j=1}^{|\mathbf{h}|} b_j h_j - \sum_{i=1}^{|\mathbf{v}|} \sum_{j=1}^{|\mathbf{h}|} v_i h_j w_{ij}, \tag{2.2}$$

where $\mathbf{v}$ and $\mathbf{h}$ are the vectors of the visible and hidden neurons, with their components v_i and h_j representing the binary states of the visible neuron i and the hidden neuron j, respectively, a_i and b_i are corresponding biases, while w_{ij} is the weight of the arc connecting v_i and h_j as shown in Fig. 2.1(b). $|\mathbf{x}|$ is the number of components of the vector $\mathbf{x}$. The concept of the free energy function is borrowed from physics, which is the internal energy of the system minus the amount of energy that cannot be used to perform work. The free energy function of this system for a given visible (input) vector is defined as [10]:

$$F(\mathbf{v}) = -\sum_{i=1}^{|\mathbf{v}|} a_i v_i - \sum_{j=1}^{|\mathbf{h}|} log \sum_{h_j \in \{0,1\}} e^{h_j (b_j + \sum_{i=1}^{|\mathbf{v}|} v_i w_{ij})}. \tag{2.3}$$



(a)

$$W_{m\times n} = \begin{matrix} \\ v_i \downarrow \end{matrix} \begin{matrix} \qquad\qquad\qquad\qquad h_j \longrightarrow \\ \begin{pmatrix} w_{11} & \cdots & \cdots & \cdots & w_{1n} \\ \vdots & \ddots & & & \vdots \\ w_{i1} & & w_{ij} & & w_{in} \\ \vdots & & & \ddots & \vdots \\ w_{m1} & \cdots & \cdots & \cdots & w_{mn} \end{pmatrix} \end{matrix}$$

(b)

Figure 2.1: (a) A simple RBM with m visible and n hidden neurons (b) weight matrix with m rows and n columns

Therefore, the probability assigned by the energy function to each pair of visible and hidden vectors can be given by:

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}. \tag{2.4}$$

The marginal probability of the visible vector $\mathbf{v}$ can then be computed by summing all possible hidden vectors as follows:

$$p(\mathbf{v}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}. \tag{2.5}$$

Weights and biases are learned through the gradient descent which reduces the energy of the network for a given example. This is equivalent to maximizing the log of the probability of a training vector with respect to weight and biases, given as follows:

$$\frac{\delta \log p(\mathbf{v})}{\delta w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}, \tag{2.6}$$

$$\frac{\delta \log p(\mathbf{v})}{\delta a_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model}, \tag{2.7}$$

$$\frac{\delta \log p(\mathbf{v})}{\delta b_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}. \tag{2.8}$$

Here $\langle f \rangle_l$ is the expectation of the function f with respect to variable l. Hence, $\langle v_i h_j \rangle_{data}$ and $\langle v_i h_j \rangle_{model}$ denote the results derived from the free energy function with respect to the given(training) data and the learned model of the network[1]. Thus, a

[1]The detailed derivations can be found in [56][96]

simple learning rule is gradient of the log probability of the training data [44] as follows:

$$\Delta w_{ij} = \epsilon_w(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}), \tag{2.9}$$

$$\Delta a_i = \epsilon_a(\langle v_i \rangle_{data} - \langle v_i \rangle_{model}), \tag{2.10}$$

$$\Delta b_j = \epsilon_b(\langle h_j \rangle_{data} - \langle h_j \rangle_{model}), \tag{2.11}$$

where ϵ_w, ϵ_a and ϵ_b are the learning rates corresponding to weights w and biases a and b, respectively. It is easy to obtain a sample of the data because there is no intra-layer connection which is the probability of a hidden neuron, given an example from the data and vice versa, is given by :

$$p(\mathbf{h}|\mathbf{v}) = \prod_{j \in \mathbf{h}} p(h_j|\mathbf{v}), \tag{2.12}$$

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i \in \mathbf{v}} p(v_i|\mathbf{h}). \tag{2.13}$$

With v_i and h_j of binary valued neurons with the free energy given by (2.3) and the conditional probabilities that each state of the visible and hidden neurons is 1 given by the model of biological neuron response are [10]:

$$F(\mathbf{v}) = -\sum_{i=1}^{|\mathbf{v}|} a_i v_i - \sum_{j=1}^{|\mathbf{h}|} log(1 + e^{(b_j + \sum_{i=1}^{|\mathbf{v}|} v_i w_{ij})}). \tag{2.14}$$

$$p(v_i = 1|\mathbf{h}) = sigmoid(a_i + \sum_{j=1}^{|\mathbf{h}|} h_j w_{ij}), \tag{2.15}$$

$$p(h_j = 1|\mathbf{v}) = sigmoid(b_j + \sum_{i=1}^{|\mathbf{v}|} v_i w_{ij}). \tag{2.16}$$

and sigmoid is defined as $sigmoid(x) = \frac{1}{1+\exp(-x)}$.

2.3.1 Gaussian-Bernoulli RBM (GRBM)

The binary RBM as discussed in Section 2.3 limits the network to the problems involving binary inputs. However, most of the pragmatic problems have real number as input values and the binary RBM is inappropriate to taking full advantages of such training data. A proposed variant to overcome this limitation and enhance the capabilities of the binary RBM is to use visible neurons as real valued neurons instead of the binary ones. This altered model is known as GRBM.

The energy function of GRBM is defined as:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{|\mathbf{v}|} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{i=1}^{|\mathbf{v}|} \sum_{j=1}^{|\mathbf{h}|} w_{ij} h_j \frac{v_i}{\sigma_i} - \sum_{j=1}^{|\mathbf{h}|} b_j h_j \tag{2.17}$$

where v_i is the real-valued visible neuron, h_j is the binary-valued hidden neuron, σ_i is the standard deviation of v_i. a_i, b_j and w_{ij} are the same as those defined for the binary RBM.

The conditional probabilities assigned by the energy function in (2.17) are as fol-

lows[2]:

$$p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{|\mathbf{v}|} \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2\sigma_i^2}(v_i - b_i - \sigma_i \sum_{j=1}^{|\mathbf{h}|} h_j w_{ij})^2}, \tag{2.18}$$

$$p(h_j = 1|\mathbf{v}) = sigmoid(b_j + \sum_{i=1}^{|\mathbf{v}|} \frac{v_i}{\sigma_i} w_{ij}). \tag{2.19}$$

(2.18) represents a multivariate Gaussian distribution with the dimensionality equal to the number of visible neurons; the covariance matrix is diagonal given as $diag(\sigma_1^2, \sigma_2^2, ..., \sigma_{|\mathbf{v}|}^2)$ (where $|\mathbf{v}|$ is dimensionality of visible neurons vector) and the mean in each variable i is given by $b_i + \sum_{j=1}^{|\mathbf{h}|} w_{ij} h_j$. The conditional probability that the state of the hidden neuron j is 1 given the visible neurons is similar to that of the binary RBM with an additional term for the scaling of visible neurons by the reciprocal of σ_i. The learning rules for the weights and biases of GRBM are similar to those of RBM, as described (2.9), (2.10) and (2.11).

The learning or update rule for the variance of visible neurons in GRBM is difficult to formulate, and is one of the causes behind the slow GRBM learning process. In the literature, GRBM is used mostly with fixed variance [106], although it can be learned by maximizing the log probability of visible neurons with respect to the variance and

[2] The probabilities are derived formally in[56]

the resulting update rule is as follows [56]:

$$\Delta\sigma_i = \epsilon_\sigma(\langle\frac{(v_i - a_i)^2}{\sigma_i^3} - \sum_{j=1}^{|\mathbf{h}|} h_j \frac{w_{ij}v_i}{\sigma_i^2}\rangle_{data} \\ -\langle\frac{(v_i - a_i)^2}{\sigma_i^3} - \sum_{j=1}^{|\mathbf{h}|} h_j \frac{w_{ij}v_i}{\sigma_i^2}\rangle_{model}). \tag{2.20}$$

A very interesting derivation of GRBM is as a constrained MoG of isotropic components [96][33]. The probability model $p(\mathbf{v})$ of the visible neurons can be represented using the law of total probability as follows:

$$p(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v}|\mathbf{h})p(\mathbf{h}). \tag{2.21}$$

From (2.18) $P(\mathbf{v}|\mathbf{h})$ is a product of $|\mathbf{v}|$ isotropic Gaussian distributions $\mathcal{N}(v_i; b_i + \sum_{j=1}^{|\mathbf{h}|} w_{ij}h_j, \sigma_i^2)$. The MoG component has an associated mixing coefficient, which is the probability of selecting the corresponding component. The MOG has a direct mapping in GRBM, here the conditional probability of the visible neuron given the hidden neurons is a Gaussian component of the learned model, and the number of mixing coefficient is combination of the number of hidden states being active. Furthermore the derivation of $p(\mathbf{h})$ in [96] implies that $b_i + \sum_{j=1}^{|\mathbf{h}|} w_{ij}h_j$ is the main contributing factor, which makes $p(\mathbf{v})$ similar to isotropic Gaussians and each Gaussian distribution is centered at the mean with the variance σ_i^2 in each direction. However, MoG components are considered as independent.

It is evident from the earlier discussion that the Gaussian mean depends on the states of hidden layer neurons, which are in $\{0, 1\}^{|\mathbf{h}|}$. This implies that the total number of components in GRBM is $2^{|\mathbf{h}|}$, which is exponential in the number of hidden neurons

in $\mathbf{h}$. The selection of the mixing coefficients is automatically made by the number of active hidden states. It can be consequently realized as given the Gaussian parameters of the visible layer neurons, the components rely on the states of the hidden layer neurons and can be computed. That implies that only $|\mathbf{v}| + 1$ (where the addition of 1 is due to bias) components that sum over exactly one or zero weights can be placed and scaled independently. Following this, $2^{|\mathbf{h}|} - |\mathbf{v}| - 1$ components are determined by the choice of the $|\mathbf{v}| + 1$ components. This entails GRBM as the form of constrained isotropic Gaussian components, which suggest GRBM derivation to MoG.

Chapter 3

Background subtraction

Computer vision attempts to mimic the abilities of human vision, in particular by electronically perceiving and processing images. The intricacies involved in understanding vision are directly mapped to the applied engineering and computational challenges of today's data-hungry world. The background subtraction is one of fundamental issues in computer vision, and its solutions can be applied to many important areas in computer vision, such as object detection and classification, face detection, action recognition, video surveillance, automated driving, and sensors in robotics [6, 9, 22, 81, 94]. It is also useful for applications in medical diagnostics, astronomy, and geo-physics [15, 53, 86]. The central idea behind background subtraction is utilizing the visual properties of images in video sequences to build an appropriate model that can be used to classify parts of any new observation into foreground or background. For simple and ideal cases, background subtraction is an elementary frame difference problem, where we calculate the differences of pixel values between two consecutive frames of a video to remove the background and highlight the foreground. However, for difficult cases background subtraction is not a trivial task and the learning of a background model is a challenging task.

There are many situations that can cause unstable learning for a background model in a scene. The problems associated with background subtraction are mentioned here

[12, 92]: The **bootstrapping** problem represents the case where images with a clear background are not available for training. Images from a video scene are predominantly cluttered with foreground objects, e.g., a scene from a crowded street produces this problem. A **dynamic background** is encountered in cases where a scene contains objects in continuous motion, such as water flowing in the background, a rotating fan, waving plants, or moving escalators. **Noise** is generated while capturing a scene either unintentionally by human error or by natural interferences. **Camera jitters** are usually visible in videos and images if the camera is mounted on an unstable panel. These jitters can be a continuous process or a sudden trembling caused by random events. Each of these two sources of jitters has a different impact on background modeling. **Camouflage** is self-explanatory, and is the embedding of the foreground object in the background. A **moved object** is either the one that stays in a scene sufficiently long time to become part of the background, or the other in the background that disappears from the scene after a long time. **Illumination change** in a scene can be a part of a regular cycle, such as the sunlight at different times of the day. A light switch can also be turned on or off in the relevant scene causing this phenomenon. **Pan-tilt-zoom** cameras capture a scene by remote directional and zoom control, which makes background subtraction more difficult to tackle. Videos captured at an extremely **low frame rate** that lack sufficient information to define a background, test the learning capabilities of background subtraction techniques. **Bad weather**, such as snowfall, heavy rain, and dust storms, affects the quality of videos and images available for training. Beside optical sensors there are other sensors commonly used in cameras to

capture images, such as infrared cameras or microwave cameras for tomographic images, these days. Videos recorded at night have poor visibility and are often cluttered with strong illumination from ambient nocturnal sources of light, such as torches and vehicle headlights. All of the above-mentioned challenges [36] are difficult to address with a simple solution to the background subtraction problems, but researchers are trying to find stand-alone and inexpensive solutions to the problem in terms of resource and time consumption.

A two-layer Restricted Boltzmann Machine (RBM) has the capacity to solve background subtraction problems with most of the involved intricacies. An RBM is a generative learning technique which models features from the data with its unsupervised training capability. It can be very helpful in exploring the background model for a video sequence. The generative nature of RBM enables to extract the foreground from a test image given that RBM has seen a sufficient number of background images. Continuous learning with a controlled learning rate can solve problems of bootstrapping, dynamic background, and moved objects in a scene.

3.1 Background subtraction using GRBM

This section details the solution of background subtraction using GRBM[1]. The method is simple, and yet competes with the state-of-the-art methods. Section 3.2 presents the detailed results which endorse the efficacy of the discussed method of background subtraction. No pre-processing of image data is introduced in training

[1]This work is submitted to "IET image processing" journal

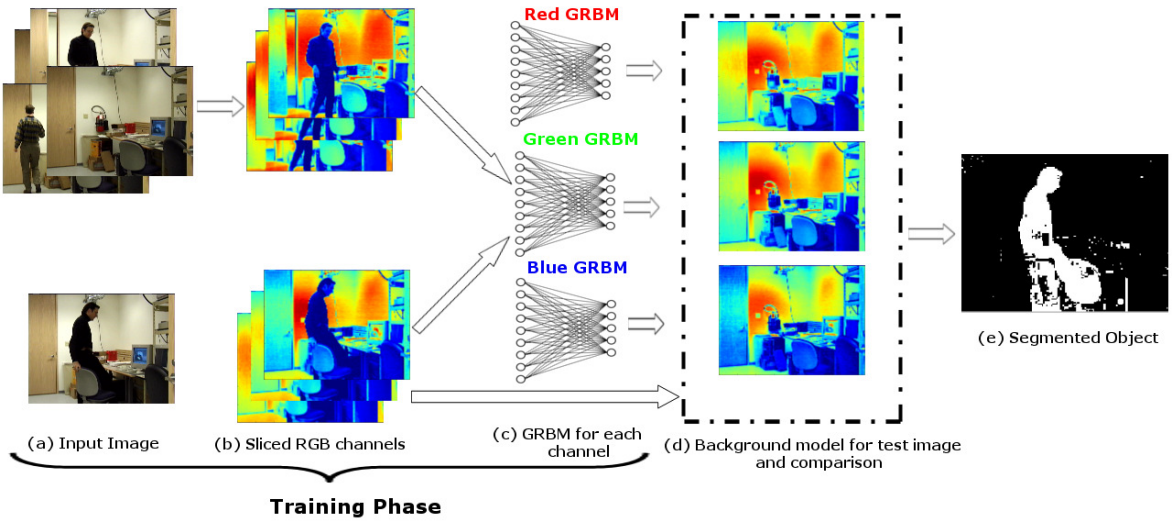


Figure 3.1: System architecture for background modeling and background subtraction phases. Modeling is composed of steps labeled (a), (b), and (c). (d) (the box bordered by broken lines) is the process of comparing the background model with the test input image respective to the RGB channel, and (e) is the foreground segmented image.

and raw RGB pixel values of images are presented as inputs to the GRBM. While simple median filtering is used as post-processing and any sophisticated post-processing technique like the graph cut minimization will further improve the results.

Background subtraction in the context of GRBM is a two-phase process. The first phase concerns background modeling, while the second one is the subtraction of the background from the test image to extract the foreground object.

3.1.1 Background modeling

The background model is composed of three independent GRBM networks i.e., one for each channel of RGB. Each channel in RGB is modeled with a separate GRBM, as shown in Fig. 3.1 (a), (b), and (c). The training image is sliced into R, G, and B

channels and each slice is used to train the respective channel's GRBM network. The pixels are considered to be random variables with associated probability distributions $p(X_i)$, where $i = 1, 2, 3$ for each channel and X is a vector of pixels. The channels are treated independently and identically in all three GRBM networks. For the sake of clarity, we describe the details of training the background model of one channel. The other two GRBM networks are trained identically to generate the background models for their respective channels. Algorithm 1 outlines the pseudo code for the discussed background modeling.

GRBM is trained using the contrastive divergence (CD) [44], which is a technique to train RBM by simulating the Gibbs sampling process. Gibbs sampling is used in Markov chains (MC) to sample the original data and generate a model, which in this paper is generating of images from a given video scene. Gibbs sampling of joint N random variables $G = (G_1, G_2, ..., G_N)$ is generated from a sequence of a data set of N sampling sub-steps in the form $G_i \sim p(G_i|G_{i-1}, G_{i-2}, ..., G_1)$. CD initializes the MC with a training example (image), and eventually generates the true model (background model). This process treats the variation in video scenes as a feature. The hidden neurons in the networks are aimed to represent the useful features and remove the infrequent features like illumination change, moving objects, natural irritations, etc.

3.1.2 Foreground extraction

Here we describe the extraction of the foreground from a test image frame. The test image is sliced into R, G, and B channels, as shown in Fig. 3.1, and their visible neurons

Algorithm 1: Background modeling algorithm

Load I_i of N frames, where $i \in \{1...N\}$;

Extract parameters: height h and width w of single frame I_1 ;

for $c \in \{R, G, B\}$ **do**

    initialize RBM_c with hxw visible and $no\ of\ Hid$ hidden neurons;

    initialize ϵ, $epochs$

end

for $noofepochs$ **do**

    for $f \in I_{\{1...N\}}$ **do**

        slice frame f in $\{f_R, f_G, f_B\}$;

        update RBM_c learning parameters with respective f_c;

    end

end

Background model update;

count fg_p $\forall$ pixel detected as foreground in successive test frames;

if $sum(fg_p \geq th_u) \geq pp$ **then**

    goto 3

end

are clamped. The network samples the test frame and reconstructs it from the learned background. The test frame is then compared with this reconstructed image frame pixel by pixel. The pixel values of the test frame falling within the scaled learned variance of the respective reconstructed frame are classified as candidates for background pixels, whereas pixels with values beyond the variance bounds are considered to be part of the foreground. Let t be the test frame and $\{t_r, t_g, t_b\}$ be the extraction of R, G and B channel, and then $rc_c(v_i)$ is the reconstruction at the visible neuron i for channel $c \in \{R, G, B\}$. The foreground of channel c (fg_c) will then be represented by following expression:

$$fg_c = \begin{cases} 1 \quad \text{if} \begin{cases} rc_c(v_i) + \alpha\sigma_i > t_c \\ \\ rc_c(v_i) - \alpha\sigma_i < t_c \end{cases} \\ \\ 0 \quad \text{otherwise} \ , \end{cases} \tag{3.1}$$

where α is the scaling parameter for variance which will be explained later in this section. (3.1) can be written in the compact form as:

$$fg_c = \begin{cases} 1 \quad \text{if } |\frac{rc_c(v_i) - t_c}{\alpha}| < \sigma_i \\ \\ 0 \quad \text{otherwise} \ , \end{cases} \tag{3.2}$$

where for all c, fg_c is computed using (3.2) and the final foreground value is determined in the form:

$$fg = \vee_{c \in \{R,G,B\}} fg_c, \tag{3.3}$$

where $\vee$ is a max operation. The variance is scaled with the parameter α, under the assumption that the data consists of un-normalized raw pixel values. The foreground value is computed separately for each channel. A pixel will be considered as a background pixel only if its all channels are considered as candidates for a background pixel. Thus, a pixel is classified as part of the foreground even if one of the three channels is classified as foreground. The resulting foreground pixels are passed through a median filter for basic smoothing to produce the pixel foreground image.

3.1.3 Model Update

As discussed in Section 3.1.2, the algorithm solely relies on the pixel information learned by the GRBM for foreground segmentation. Thus, the background model needs continuous update to incorporate the changes appear in the background scene overtime. The algorithm keeps track of the pixels classified as foreground(by (3.2)) in successive video frames and call them persistent pixels. An automatic update of the background model is triggered if the count of persistent pixels exceeds a certain predetermined threshold, and the algorithm re-learns the background model from the recent frames accommodating the recent changes in the background scene. The background model is updated in background, while the algorithm works with foreground segmentation. The learned parameters in GRBM's are weight matrix and variance vector, which are replaced online by simple exchange. The conditions for update trigger are expressed in

the form:

$$fg_p = \begin{cases} fg_p + 1 & \text{if } fg = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where fg_p represents for how long (in terms of the number of consecutive frames) the pixel is classified as foreground pixel.

$$fg_{pc} = \begin{cases} 1 & \text{if } fg_p \geq th_u \\ 0 & \text{otherwise,} \end{cases} \tag{3.5}$$

where fg_{pc} is an indicator for a pixel being a persistent foreground pixel and th_u is the update threshold whose unit is the number of frames to decide whether a pixel is consecutively classified as a foreground pixel or not.

$$ut = \begin{cases} TRUE & \text{if } sum(fg_{pc}) \geq pp \\ FALSE & \text{otherwise,} \end{cases} \tag{3.6}$$

where ut represents the indicator for updating the background model and pp is the threshold for updating the model.

The algorithm then automatically triggers the update procedure when ut is $TRUE$ to re-learn the background model. Notice that th_u and pp decide the frequency of updates, and th_u is tested with 200 frames and pp to $h \times w \times 0.1$. These values are selected empirically to produce better results with tested data sets. The update procedure reduces the effect of ghost objects and illumination changes in the scene by time, and

can be extended for crowded scenes with composite of fg_p and pixel intensity values. It is simple as well as computationally limited procedure. One challenge of using this procedure in real time applications is the inclusion of the ghost objects in the foreground segmentation results for a duration between the update trigger and parameters replacement of the model. This, however, does not exhibit a high impact as the training procedure is fast enough. The pertinent details are discussed with facts in Section 3.2.

3.2 Experiments

In this section, we provide details of a collection of experiments and results to demonstrate the performance of background subtraction using GRBM. The experiments are performed with Wallflower [92], Star [60], and Changedetection [36] data sets. This helps carry out comparison with results published in the recent literature. In addition to these sophisticated data sets, a video from a cheap CCTV camera is also tested to verify the robustness and extend the comparative analysis.

In the networks, the number of visible layer neurons is equal to the number of pixels in the image frame, and the number of hidden neurons is empirically selected as eight neurons for all data sets. We use 5 epochs for training as the inherent redundancy in video data allows it to converge very early. The learning rate for weights, ϵ_w is selected from the list $[0.01, 0.0075, 0.0050, 0.0025, 0.0001]$ in sequence for each epoch. The learning rate for variance, ϵ_σ is also empirically selected set as 0.0001 for all experiments. Value of the scaled variance variable α in equation(3.1) is computed by generating foreground segmentation with $\alpha \in \{5, 6, ..., 20\}$ for randomly selected image

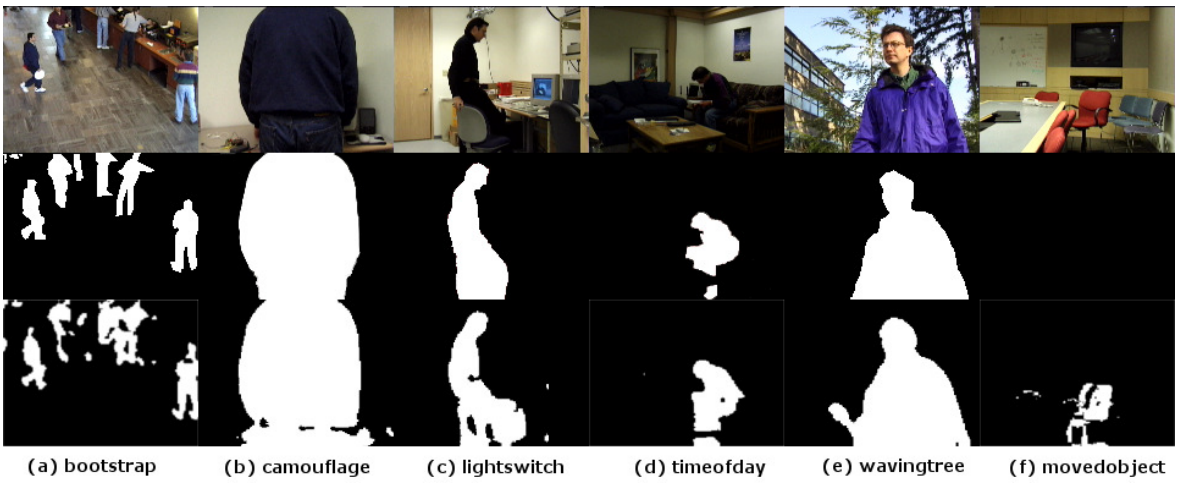


Figure 3.2: Wallflower data set results. The first row shows the actual image, the second row the ground truth, and the third row shows the background subtraction

frames coming from training images. The value of α for testing is selected as the one which gives the minimum average number of foreground pixels per frame. It is noticed that a variation of $\alpha \pm 2$ does not have any noticeable effect on the segmentation results. In the last step, the median filter is used to remove noise from the previous results. The window size of the median filter is selected by computing the noise i.e. blinking pixels from a set of $\{7, 8, ..., 13\}$, where the higher value is selected for more detected noise. This experimental setup is referred to as Ex_A in the script, henceforth.

The hyper-parameters can be further tuned to improve the results with prior information about the test videos. Beside conducting experiments with single set of hyper-parameters in each dataset, we performed experiments with more suited selection for a category of videos. The learning rates and the number of epochs are slightly tuned for each data set. The learning rates for weights ϵ_w are 0.0001 and 0.001, where the smaller learning rate is applied to more dynamic backgrounds. Similarly, 5 to 10

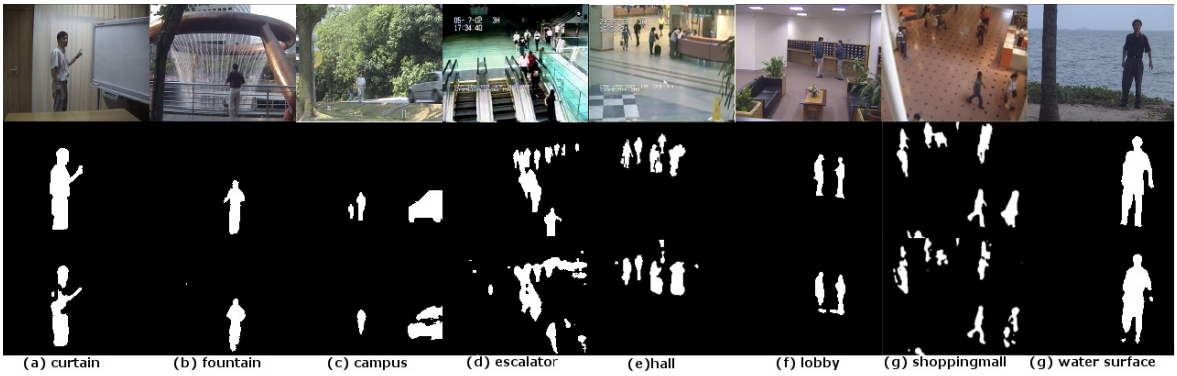


Figure 3.3: Star data set results. The first row shows the actual image, second row the ground truth, and the third row shows the background subtraction

epochs were sufficient for training, where videos with static backgrounds are trained with 5 epochs and videos with dynamic background are trained with 10 epochs. Rest of the parameters are the same as discussed in general cases. This experimental setup is referred as to Ex_B in this paper, henceforth. It is noticed that with the prior knowledge of the category of the test video, the selected parameters improves final results from 3% to 5%.

The hyper-parameters reported in the paragraph above are not adaptive in the experiments reported in this chapter. Further discussion on the intricacies involved with parameters in RBM specifically selection of learning rates is presented in literature [78][32] and will be helpful for interested users. The learning rates are usually selected manually with RBM [45] (experimental setup Ex_B). Another simple strategy is adapted, that is to start with a small learning rate and change to larger value for every epoch of training (experimental setup Ex_A). Also, there are proposals to work out the learning rate adaptively [17][18], which can be incorporated very efficiently into

the discussed idea, as there are few static learning rates reported and the domain of selection will be small.

The experiments are executed on a standard PC with two 3.40 GHz processors and regular load of general applications. The training took around 3 minutes for 254 frames with each frame being 388x278x3. The testing or the background subtraction on the same size of video frame is then experienced at the rate of more than 18 frames per second(fps). This can be further improved using GPU computation and parallelism.

Table 3.1: Contingency matrix for F-measure

		Background subtracted image	
		Foreground	Background
Groundtruth	Foreground	TP	FN
	Background	FP	TN

The results are evaluated qualitatively in visual form. The visual results are simply presented as the foreground subtracted from a given test frame using the discussed background subtraction technique. The results shown in the image also contain the handcrafted ground truth from the publisher of the data sets for better comparison.

Quantitative evaluation is carried out using the F-score or F-measure, which is an assessment of test accuracy. F-measure is the harmonic mean of precision and detection

rate (recall), where precision and recall are defined as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{3.7}$$

$$Detectionrate(DR) = \frac{TP}{TP + FN}, \tag{3.8}$$

$$F\text{-}measure = \frac{2 \times Precision \times DR}{Precision + DR}. \tag{3.9}$$

Here, TP (true positive) is the number of correctly identified foreground pixels, FN (false negative) is the number of foreground pixels incorrectly identified as background pixels, FP (false positive) is the number of background pixels incorrectly identified as foreground pixels, and TN (true negative) is the number of correctly identified background pixels, which are summarized Table 3.1 in the form of the contingency matrix. The handcrafted ground truth, which is available from the publishers along with the data set, for each data set is used for the computation of values in the contingency matrix.

For the sake of brevity and in order to dwell on the results of the discussed technique, visual results coming from the compared techniques are omitted but can be verified from the referenced material.

3.2.1 Wallflower data set

Wallflower is probably the most cited and the oldest standard background subtraction data set. It contains seven videos, each for a different background subtraction

Table 3.2: F-measure for Wallflower data set. The MovedObject data set is excluded for being undefined for most techniques.

Technique	Bootstrap	Camouflage	LightSwitch	TimeofDay	WavingTree	Average
ABSM[59]	0.6863	0.9414	0.7206	0.5911	**0.9746**	0.7828
SuBSENSE[88]	0.4192	0.9535	0.3201	0.7107	0.9597	0.6726
ViBe[8]	0.5433	0.9006	0.1888	0.3967	0.7271	0.5513
CodeBook[54]	0.4727	0.9418	0.61354	0.5132	0.9301	0.6943
PBAS[46]	0.2857	0.8922	0.2212	0.4875	0.8421	0.5457
J.G. Park[69]	0.6825	0.9296	0.5043	0.8048	0.9733	0.7789
TARBM[102]	0.6513	0.9515	0.5135	0.3087	0.8814	0.6613
GRBM(Ex_A)	0.7799	0.8879	0.8482	0.6468	0.8988	0.8123
GRBM(Ex_B)	**0.7959**	**0.9687**	**0.8372**	**0.8620**	0.9660	**0.8859**

problem category. For that data set the F-measures of the discussed method along with published results of the state-of-art background subtraction techniques are presented in Table 3.2. The experiment Ex_A uses a static α i.e. 12 to show that the GRBM background subtraction technique can perform competitively with such imposed conditions. Fig. 3.2 shows the qualitative results produced by the experiment Ex_B, where the first row contains the original test frame, the middle row shows the ground truth, and the bottom row shows the segmented foreground. It is evident from the quantitative results that the GRBM background subtraction technique performs better than other state-of-the-art background subtraction techniques. One main reason behind the better performance is that the technique here does not rely on feedback methods and

Table 3.3: F-measure for Star data set.

Technique	Curtain	Fountain	Campus	Escalator	Hall	Lobby	Shopping Mall	Water Surface	Restaurant	Average
Li 2[60]	0.1841	0.0999	0.1596	0.1294	0.1135	0.1554	0.5209	0.0667	0.3079	0.1930
Stauffer[89]	0.7580	0.6854	0.0757	0.1388	0.3335	0.6519	0.5363	0.7948	0.3838	0.4842
Culibrk[23]	0.7368	0.4636	0.5256	0.4924	0.3923	0.6276	0.5696	0.7540	0.4779	0.5600
Maddalena[64]	0.8178	0.6554	0.6960	0.5770	0.5943	0.6489	0.6677	0.8247	0.6019	0.6760
TARBM[102]	0.8174	0.6871	0.4047	0.4196	0.581	0.2033	0.6943	0.8979	-	0.5892
DP-GMM, tuned[41]	0.8411	0.7424	**0.7876**	0.5522	0.5676	0.6665	0.6733	**0.9298**	**0.6496**	0.7122
GRBM(Ex_A)	**0.8509**	**0.7791**	0.6794	**0.6206**	**0.6779**	**0.6719**	**0.7534**	0.9182	0.6120	**0.7291**

foreground modeling, which is the main strength for many of the state-of-the-art methods.

3.2.2 Star data set

The Star data set is another often-cited data set for background subtraction evaluation. There are nine videos in this data sets which magnifies the challenge for background subtraction set by Wallflower data set. The videos in this data set are provided with multiple handcrafted ground truths for extensive evaluation of background subtraction techniques. The quantitative evaluation results are thus averaged over all frames for which the ground truth is available. The results of the GRBM background subtraction technique with Ex_A are presented in Table 3.3 and the segmented fore-

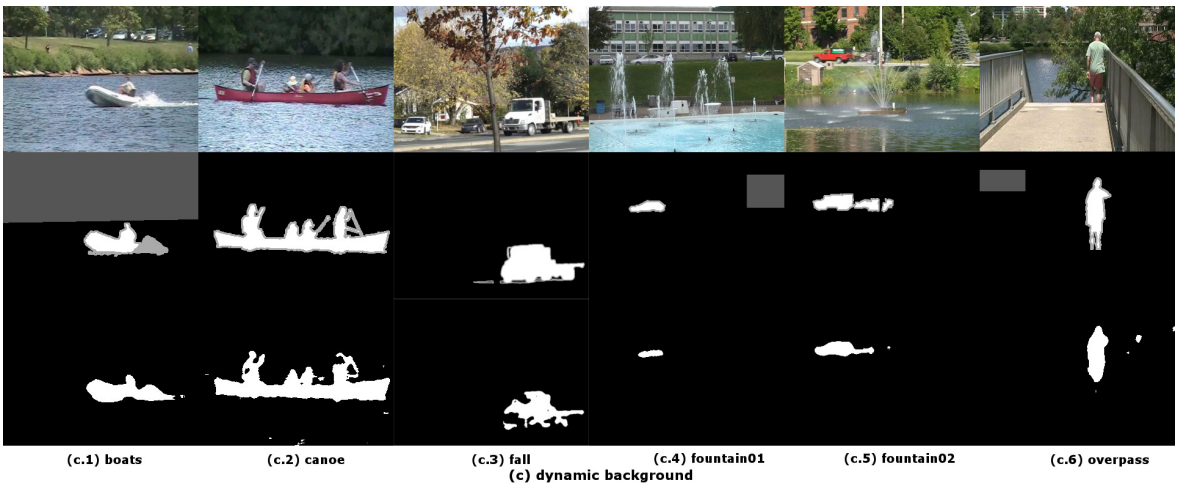


Figure 3.4: Dynamic background category for Change Detection data set

grounds are depicted in Fig. 3.3. This figure contains the original sample frame in the first row, the handcrafted ground truth in the middle row, and the segmented foreground using the GRBM background subtraction technique in the bottom row. It performs better than any state-of-the-art technique in terms of F-measure for most of the videos. The restaurant video is identical to the bootstrap video in Wallflower data set, but the additional ground truth frames are tested and the average F-measure is provided for this data set.

3.2.3 Change Detection data set

The Change Detection data set is a comprehensive data set covering most of the challenges that need to be addressed by a competitive background subtraction technique. The data set contains 11 categories in all with four to six videos in each category. The detailed results of the foreground segmented images and the quantitative evaluations for various background subtraction techniques are available on the publisher's

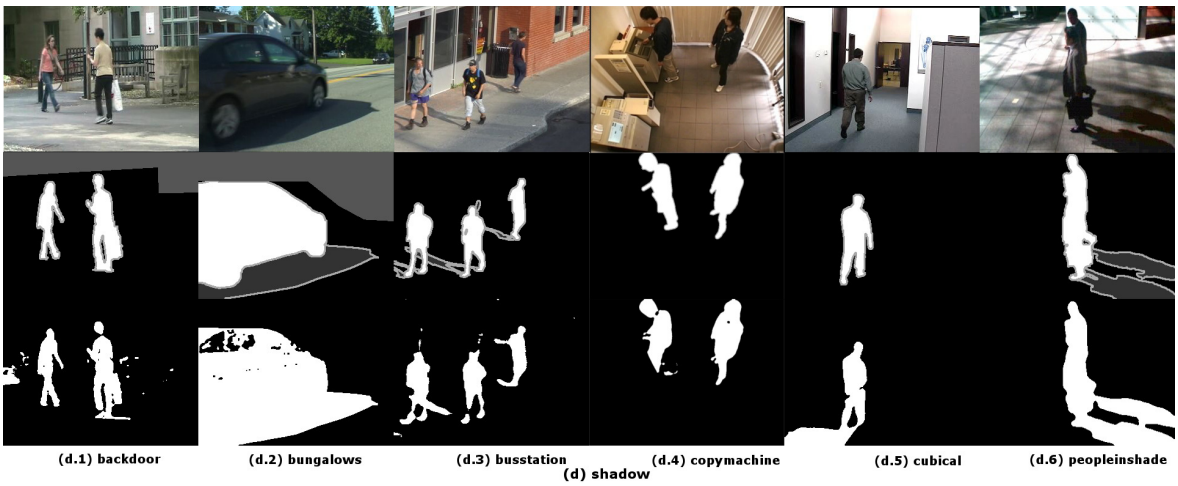


Figure 3.5: Shadow category for Change Detection data set

website[2]. We test our proposed technique every video in each category of the latest version, except for videos in the PTZ category.

The challenge is tested with experimental setups, Ex_A and Ex_B, and the quantitative results averaged over each category are presented in Table 3.4, along with the results from other state-of-the-art background subtraction techniques. The averaged results showed that our technique is among top 5 state-of-the-art background subtraction techniques. Even though it is very simple and does not rely on complex foreground modeling and ensembles. The results are published at changedetection website[3].

The foreground segmentation of the selected frames with Ex_B is shown for each video in this paper from Fig. 3.4 to Fig. 3.13. The results for all frames are also available online, along with the detailed quantitative results produced from the tools provided by the data set publisher. The evaluation results asserts that th discussed background

[2]http://changedetection.net

[3]The ranks are obtained from changedetection.net website on submission date and are subject to change

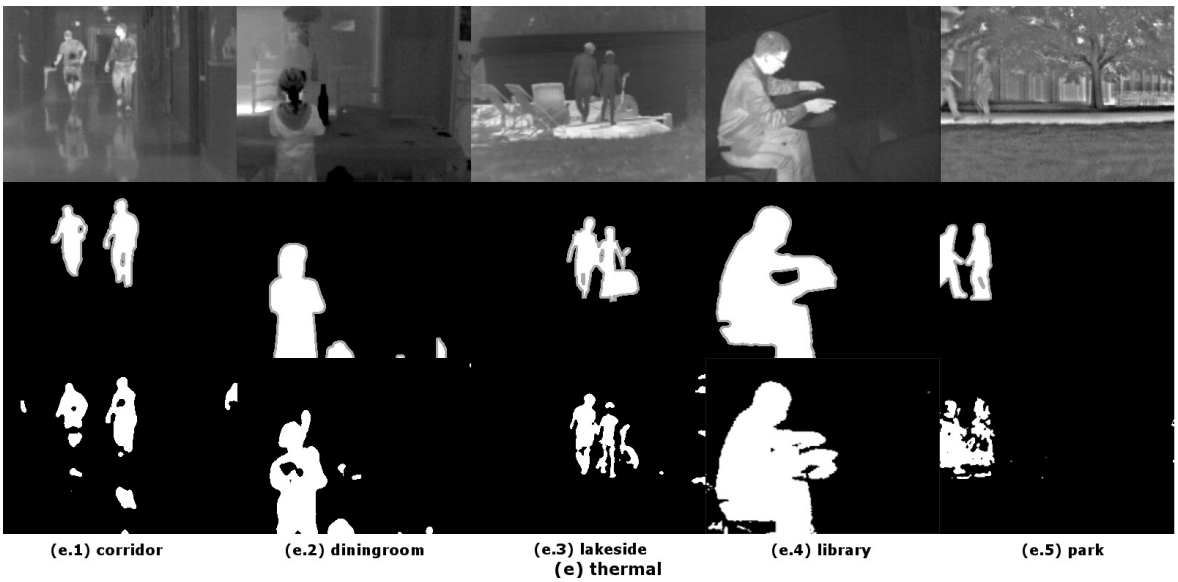


Figure 3.6: Thermal category for Change Detection data set

subtraction technique is a genuine novel addition to the state-of-the-art background subtraction techniques.

We use a couple of tricks to improve the results in dynamicBackground and cameraJitter categories. We employ the techniques mentioned in [24] to detect the blinking pixels for dynamic background detection and cameraJitters. The region with more dynamic background is compared with higher thresholds, while the videos with cameraJitter are aligned to the previous frame and tested for foreground segmentation. The changes does not effect the results in general but improves the results. We also reduce the noise by removing the small connected components in the segmented foreground. Another, trick to improve the results is to fill the holes in segmented foreground to increase the TP. These tricks helps us to improve the results in general, and it can be further studied to enhance the specific results.

Some known issues that affect the average F-measure of the background subtrac-

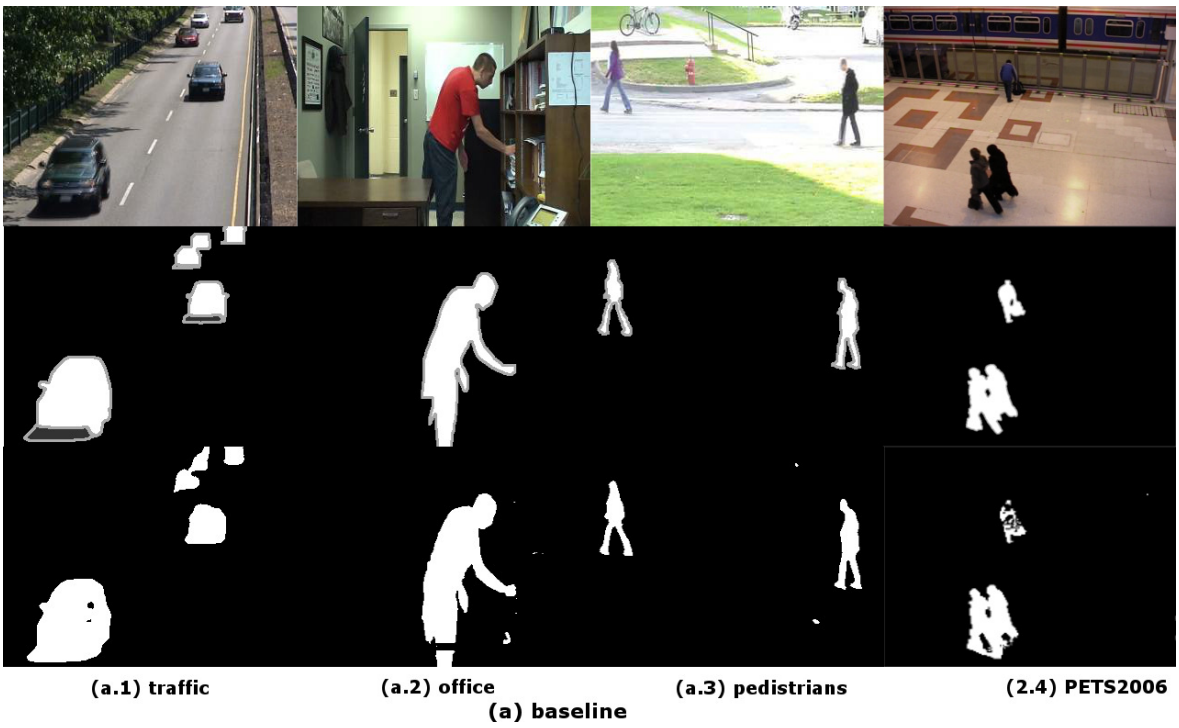


Figure 3.7: Baseline category for Change Detection data set

tion with GRBM technique are noise, shadows, abrupt change of background, and mirror reflections. The problems of noise, abrupt changes in the background, and high illumination, such as headlights in nocturnal videos, can be solved by pre-processing the training data. The shadows and mirror reflections can also be treated as special cases and removed by post-processing. The experiments performed with the discussed method uses the RGB color intensities only to train the background model, which can be extended by adding local features with an additional GRBM network. We do not employ foreground modeling or feedback techniques, which are key factors in improvements of the results in many state-of-the-art techniques. But it can be integrated to further improve the quality of the quantitative results.

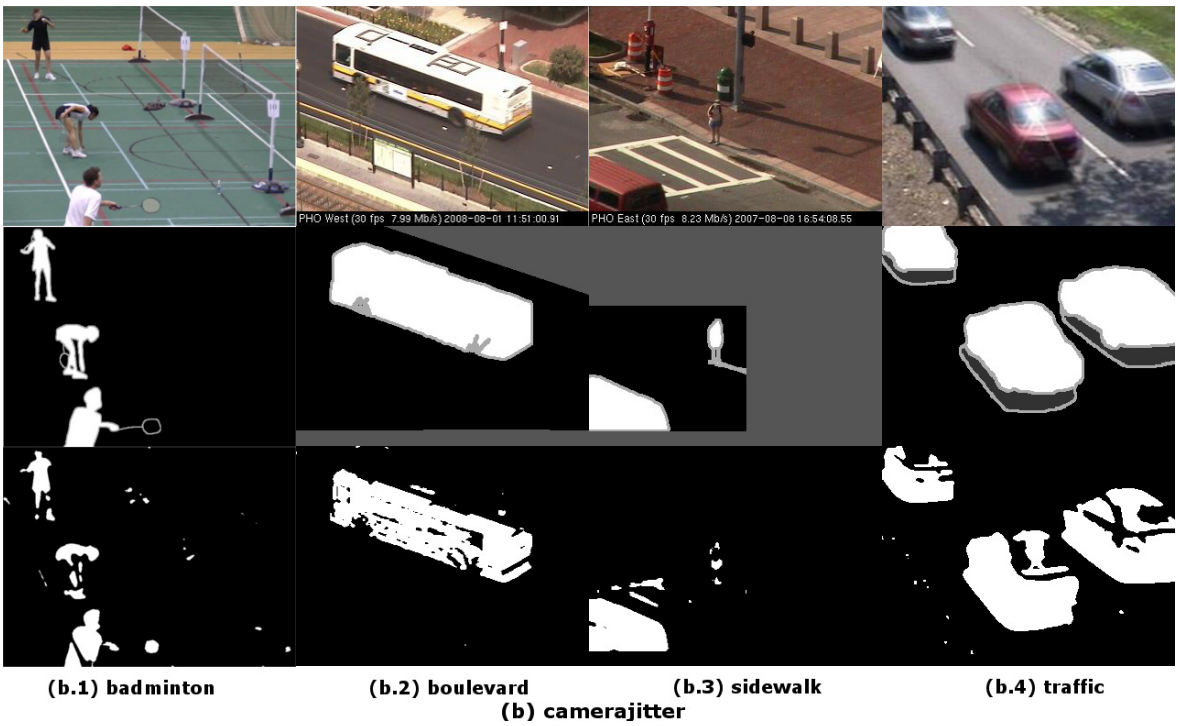


Figure 3.8: Camera jitter category for Change Detection data set

3.2.4 Additional video

The GRBM background subtraction technique is tested with a video captured by a cheap CCTV camera and data set is available on our website[4]. The color quality is very low. The background subtraction result of a test frame is shown in Fig. 3.14, where the first slice is the original test frame, middle one is the hand segmented ground truth and the final slice is the segmented foreground object using the discussed technique. The F-measure computed for this single frame is 0.83.

[4]http://mlv.gist.ac.kr/

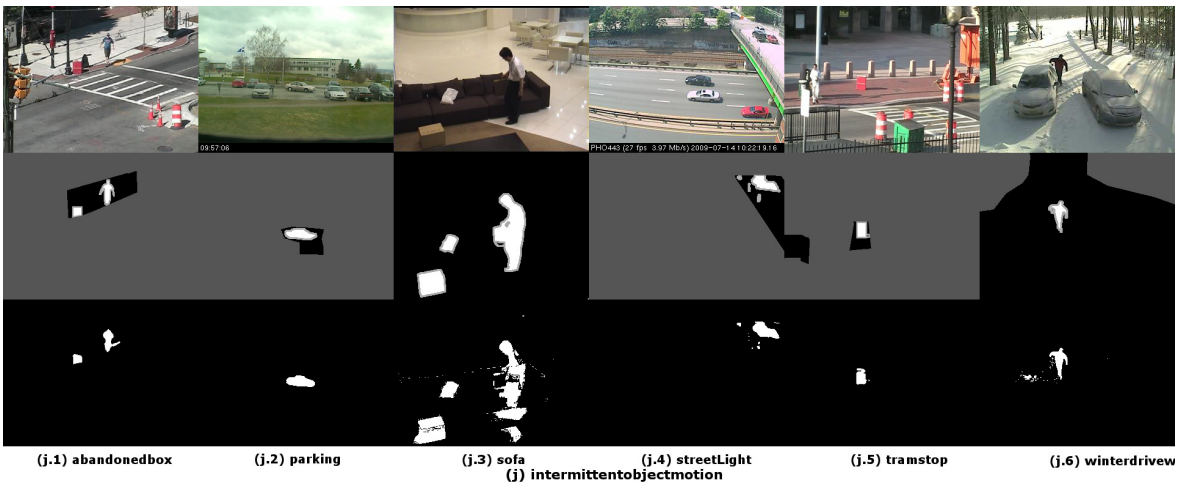


Figure 3.9: Intermittent moved object category for Change Detection data set

3.3 Related Work

Various methods have been proposed to tackle the background subtraction problems [12, 92]. One of the most widely used techniques was introduced by Stauffer & Grimson [89], which uses the Mixture of Gaussians (MoG) model for the density estimation (DE) of each pixel, and later regularizes the connected components. This method was later extended with variants of MoG implementations [108]. A recent extension [28] of MoG is to split over-dominating modes (distributions), produced by the expectation maximization (EM) learning mechanism, that dominate weaker distributions. This has been further analyzed to propose a variance control mechanism to improve results [29]. Another recent effort [40] to extend MoG for the DE of each pixel uses the Dirichlet process Gaussian mixture model (DPGMM), which automatically detects the number of mixture components required to model the background color distribution of the pixels. An extension to DPGMM uses Dirichlet process mixture models, which reduce the computational cost of continuous learning and introduce a background model update

Table 3.4: F-measure for Change Detection CDnet 2014 data set.

Technique	**Ba**	**BW**	**CJ**	**DB**	**IOM**	**LF**	**NV**	**Sh**	**Th**	**Tu**	**Average**
SuBSENSE[88]	0.9503	0.8619	0.8152	0.8177	0.6569	0.6445	0.5599	0.8986	0.8171	0.7792	0.78013
FTSG[97]	0.933	0.8228	0.7513	0.8792	0.7891	0.6259	0.513	0.8832	0.7768	0.7127	0.7687
CwisarDH[37]	0.9145	0.6837	0.7886	0.8274	0.5753	0.6406	0.3735	0.8476	0.7866	0.7227	0.71605
MBS[77]	0.9287	0.773	0.8367	0.7352	0.7568	0.6279	0.5158	0.8262	0.8194	0.5698	0.73895
PBAS[46]	0.9242	0.7673	0.722	0.6829	0.5745	0.5914	0.4387	0.8143	0.7556	0.6349	0.6906
Spec-360[79]	0.933	0.7569	0.7142	0.7766	0.5609	0.6437	0.4832	0.8187	0.7764	0.5429	0.70065
KNN[108]	0.8411	0.7587	0.6894	0.6686	0.5918	0.5491	0.42	0.7788	0.6046	0.5198	0.64219
KDE[27]	0.9092	0.7571	0.572	0.5961	0.4088	0.5478	0.4365	0.766	0.7423	0.4478	0.61836
SCSOBS[65]	0.9333	0.662	0.7051	0.6865	0.5026	0.5463	0.4503	0.723	0.6923	0.488	0.63894
Shared[16]	0.9346	0.7791	0.8173	0.8673	0.7979	0.6664	0.4333	0.8133	0.8254	0.8556	0.77902
GRBM(ExA)	0.916	0.7471	0.6901	0.5963	0.76	0.7446	0.72	0.57	0.35	0.54	0.66341
GRBM(ExB)	0.916	0.768	0.7708	0.833	0.832	0.77	0.7545	0.6469	0.3987	0.6469	0.73368

with a change of scene [41]. The GMM were effectively combined with the region based local feature for the background modeling and the results are aggregated with minimal spanning tree in [14]. The GMM is also used with two different color representation based on the results from the segmentation [77].

DE methods that use a kernel density estimate (KDE) with Gaussian kernels [80] and step kernels [8, 108] have also been developed. However, carefully selecting a best variance at each pixel location improves result and it has been picked using maximum-likelihood approach in literature [67]. Codebook construction [54] is adopted by recording the samples at each pixel in a code word that represents the background model. A novel framework was introduced in [68], where the texture, color and regional appearance of pixels are modeled using a codebook. An alternative feature vector-based

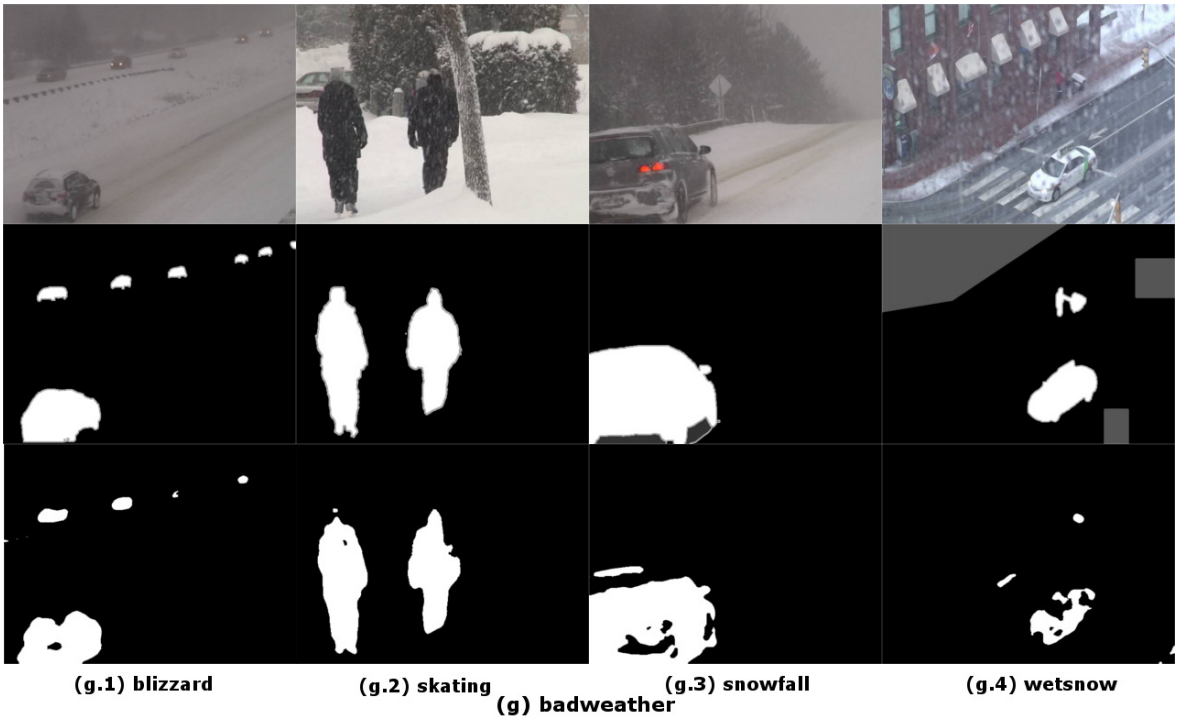


Figure 3.10: Bad weather category for Change Detection data set

technique with integrated neighborhood-assisted initialization has also been proposed [71]. It uses pixel intensity, local homogeneity, energy, and the texture mean feature vectors associated with each pixel. A recent addition to the dictionary based background modeling technique used the combination of the pixel color values, local feature and adaptive parameter in a single word for each pixel [87]. The adaptive adjustable parameters are learned by a feedback procedure based on the segmented foreground pixels [88].

Probabilistic modeling of the background has also been investigated extensively. A recent effort [75] in this regard is one where a foreground mask is generated probabilistically from pre-identified foreground and background pixel classes. Sensors and hardware implementation for background subtraction are relatively new. A sensor-based

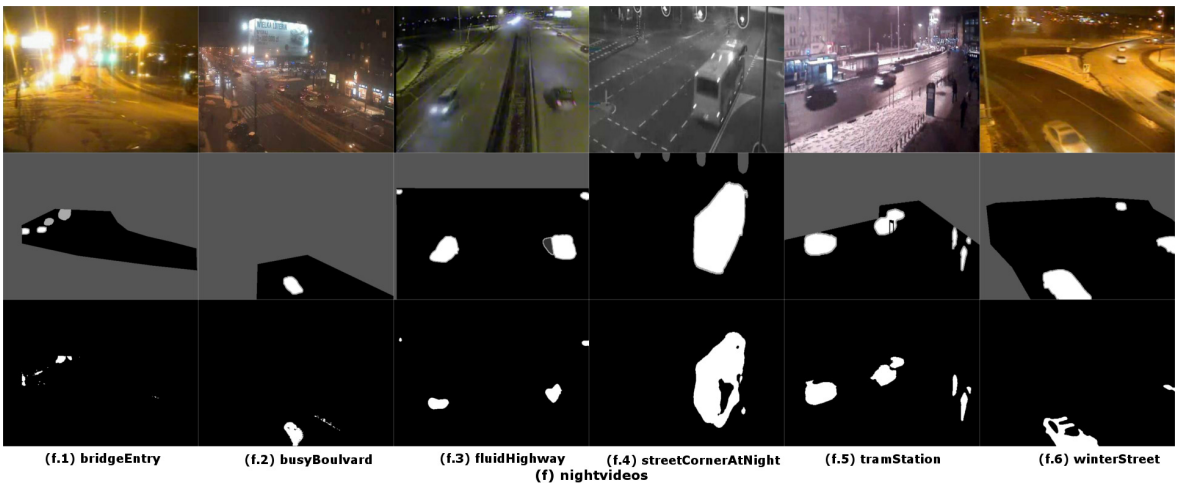


Figure 3.11: Night video category for Change Detection data set

approach has been proposed [90] that involves inpainting missing values in depth maps obtained from Kinect and related sensor. Another proposal in this regard is the use of ultra low-powered vision sensors performing pixel-level analog dynamic background subtraction [20].

Biologically-inspired techniques have also been adopted by using a feed-forward neural networks (NN) [23] to achieve background subtraction. A recent effort based on NN [31] extends [64] the idea of self-organization to automatically compensating for the egomotion of a device. Convolutional neural networks are used recently for background subtraction where the background modeling is learned using the hand segmented foreground objects in the training images.

The idea underlying our proposal is to extend biologically inspired techniques by adopting RBM to generate a background model [73]. RBM is comprised of two layers, a visible and a hidden layers of stochastic binary neurons. It learns a generative model of training data, and without supervision, extracts features to best reproduce the training

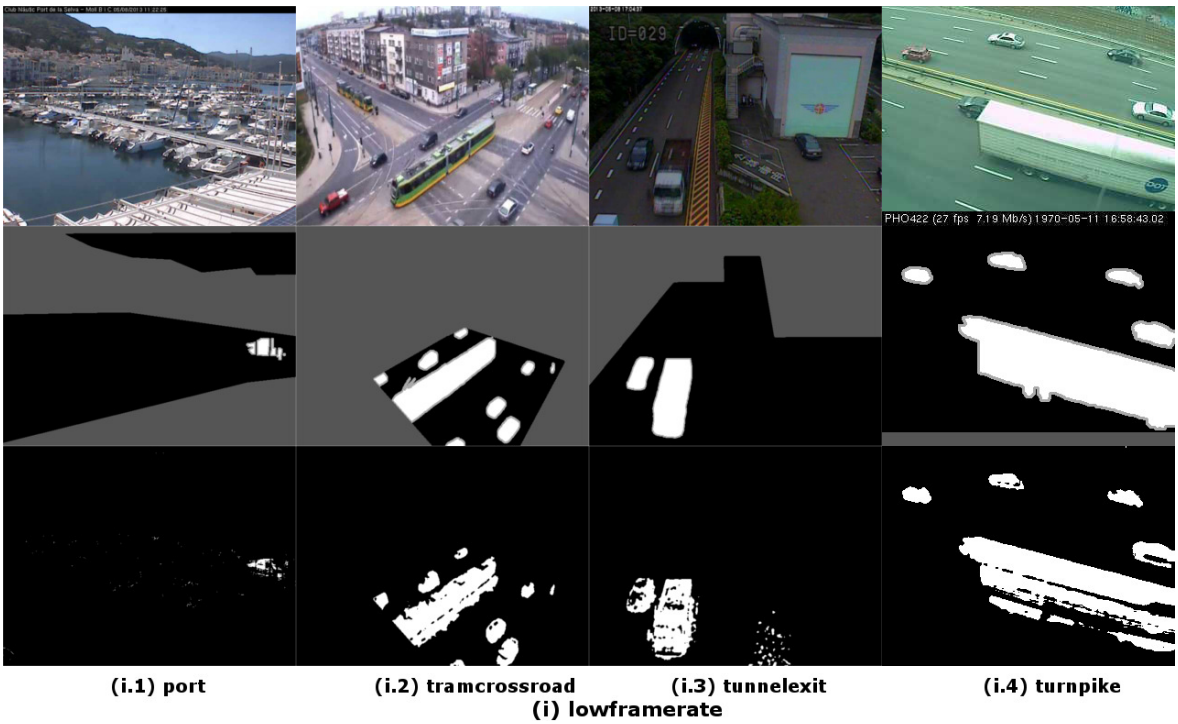


Figure 3.12: Low frame-rate category for Change Detection data set

data. In light of the common challenges in background subtraction mentioned in Section **??**, RBM with its unsupervised and generative properties offers a natural solution. Although the binary RBM offers promising capabilities, it comes with the cost of heavy pre- and post-processing of video sequences for reliable training in RBM. GRBM [100] is the alternatively proposed one that uses real numbers as input to visible neurons. GRBM extends the representation power of RBM to make better use of pixel intensity values in the RGB color domain. Other advantages of GRBM include its equivalence to the form of MoG [96], which is formulated in detail in the literature and is briefly elaborated upon in Section 2.3.1. RBM for background subtraction was earlier used in literature for background subtraction[39, 102], but the basic idea is different from the work discussed here. The technique discussed in this chapter introduce a simple GRBM

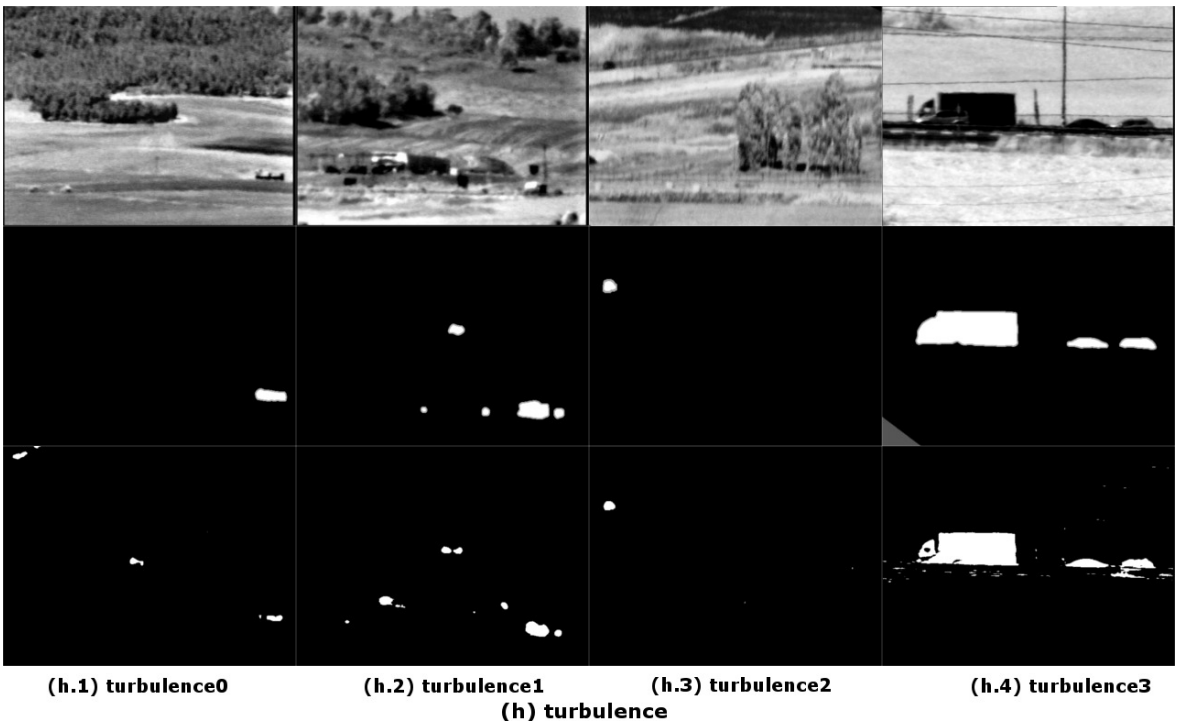


Figure 3.13: Turbulence category for Change Detection data set

which is trained using contrastive divergence learning algorithm and explore the usage of learned variance for foreground segmentation. However the earlier proposed works introduced changes in the basic RBM structure and learning techniques. They also rely on training different background scenes with separate RBM by modifying the training data.

3.4 Discussion

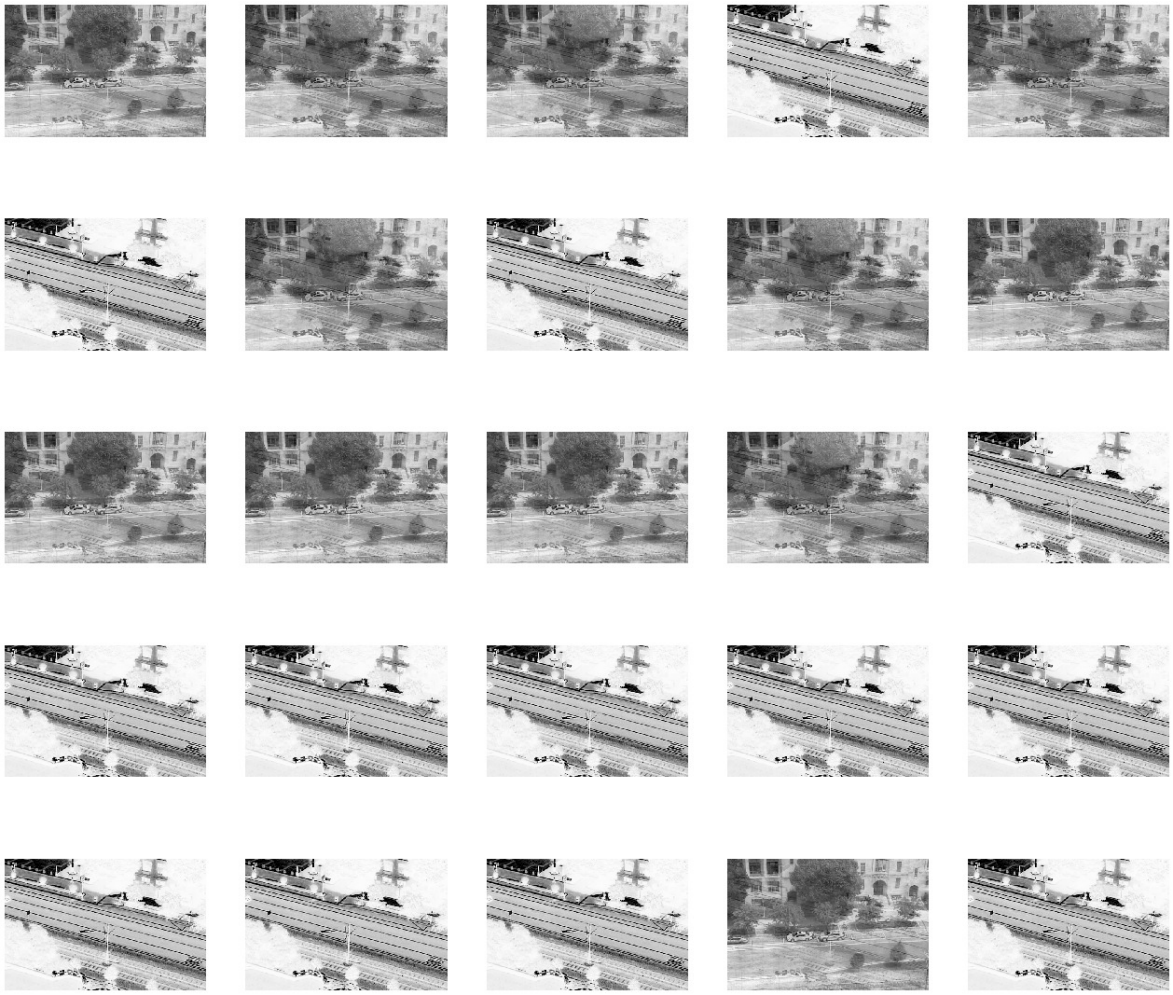
Background subtraction using GRBM presented in this chapter showed results which are comparative to the state-of-the-art. However, there are some deficiencies which are observed during experimentation which is important to mention to pursue this work in future. RBM and GRBM have enough capacity to learn from the scenes

Figure 3.14: Background subtraction results obtained for low cost CCTV camera by nightowl security system

which are provided for background modeling. The solution provided in this chapters make use of the reconstruction of the frame at the visible units which are often biased towards some scenes. It is also visible in Fig. 3.15 that the receptive fields represent most of the available scenes in the tested video. It is important to make use of the whole knowledge learned by the network to conclude the final foreground segmentation.

The background subtraction problem has a challenging task of updating the background model to cope with objects movement. The technique presented in this chapter updates the whole model by relearning the RBM with current video frames of the scene. Although learning with RBM with this specific problem is fast, but relearning the whole model can be avoided by partial learning of the network. There is some recent proposal for RBM [19] and CNN [62] partial network updates which can be tried to design a more elegant solution.

(a)



(b)

Figure 3.15: (a) five different views from a video captured with PTZ camera (b) receptive fields of RBM trained with this video

Chapter 4

Hybrid neuromorphic speech recognition

Speech recognition is a fundamental yet complex problem for the intelligence striving modern era. Speech is the most common means of communication among human race, so the automated speech recognition (ASR) system has numerous applications. Verbal communication is a trivial task in every day life, but this becomes a complex phenomenon when ported to the machines. In addition to hundreds of thousands of words in the vocabulary of a single language, the pronunciations and dialects of the same words make ASR a complex problem in the field of artificial intelligence. Although there are many advances reported on this front with software simulations, the solutions are not scalable to port it to the hardware of an intelligent machine.

The memristor is potential candidate to embed intelligence in the circuits, as their properties machinated engineering of long awaited neuromorphic devices. The memristor is defined as a basic circuit element which can hold the information of the last current passed through it. The device is in use, lately, to build prototype of intelligent systems. Automatic speech recognition systems are discussed with memristors in recent literature. A binary memristor is proposed for speech recognition, along with a crossbar, which can recognize five vowels [93]. The system used 4-bit 64 input channels with each hardwired memristor crossbar channel working as a filter, and it reported 89.2% results with 2500 speech samples. Another template matching technique is used

with predefined template with memristor cells, and it performed bit by bit comparison representing by a memristor [63]. This template-matching system is tested for speech recognition task by computer simulations with individual words from TIMIT dataset. The accurate prediction is reported from 91% to 95%. The memristor is used as a synapse in neural networks by its originator and proposed a multilayer neural network with a learning scheme [2]. The network learns separately in software, and each neuron layer, in hardware, is trained from it independently. [26] used the old cellular neural networks with memristors and tested its efficacy with image processing techniques.

The asymmetric behavior of a single memristor, discussed originally, is far hard to train the state-of-the-art neural networks. Therefore, a two memristors synapse [82] is proposed by using two PCMO RRAM devices [70]. Later, the synapse was used to train the restricted Boltzmann machine (RBM) with contrastive divergence (CD) algorithm [83]. A hybrid neuromorphic speech recognition system is designed with the help of two memristor synapse and CD algorithm which will be discussed later in this chapter. The system uses a multilayer neural network, where the first layer of the network encodes sound signals into a binary sparse representation and the second layer is neuromorphic network for feature extraction and classification.

4.1 Hybrid neuromorphic system for speech recognition

The physical properties of a resistive random access memory (RRAM) memristor makes it a suitable candidate to represent the memory as a circuit element. Although the ideal characteristics of a memristor are desired to make it near precise alternate

of the synapse, the available devices with asymmetric properties can also be used to achieve the connection feasibly. A two memristors synapse is used in literature to overcome the deficient symmetric behavior of the real device where the capacitance of the two memristors is exploited by potentiating only. The weight representation of a synapse is computed by taking the difference of the capacitance at two component memristors of this synapse, at that point in time. We use a AI/$Pr_{0.7}Ca_{0.3}MnO_3$(PCMO) device for RRAM based synapse [70].

An artificial neural network is composed of the neurons and the synapses, and the connection between two neurons can be emulated by a two-memristors synapse to build a neuromorphic device. We propose a multilayer neural network in which the second layer of the neural network is replaced with a neuromorphic chip. The neural network consists of two separate layers of RBM, stacked on top of each other. The base layer is a Gaussian-Bernoulli RBM (GRBM), and the second layer is a standard binary RBM. The architecture of the neuromorphic network is depicted in Fig 4.1, and is termed as deep belief network in neural network's literature [1]. The details of RBM and GRBM are give in Chapter 2.

The RBM is trained with CD [44] algorithm which brings the network in a low energy state for training data. The two RBMs are trained separately to each other using contrastive divergence (CD) algorithm. The bottom network, GRBM, encodes the real data of the sound files to a sparse binary representation. It is important to notice that encoding is learned into the network rather than a fix representation. The advantages are two fold: Firstly, the system is scalable and secondly, it is robust to

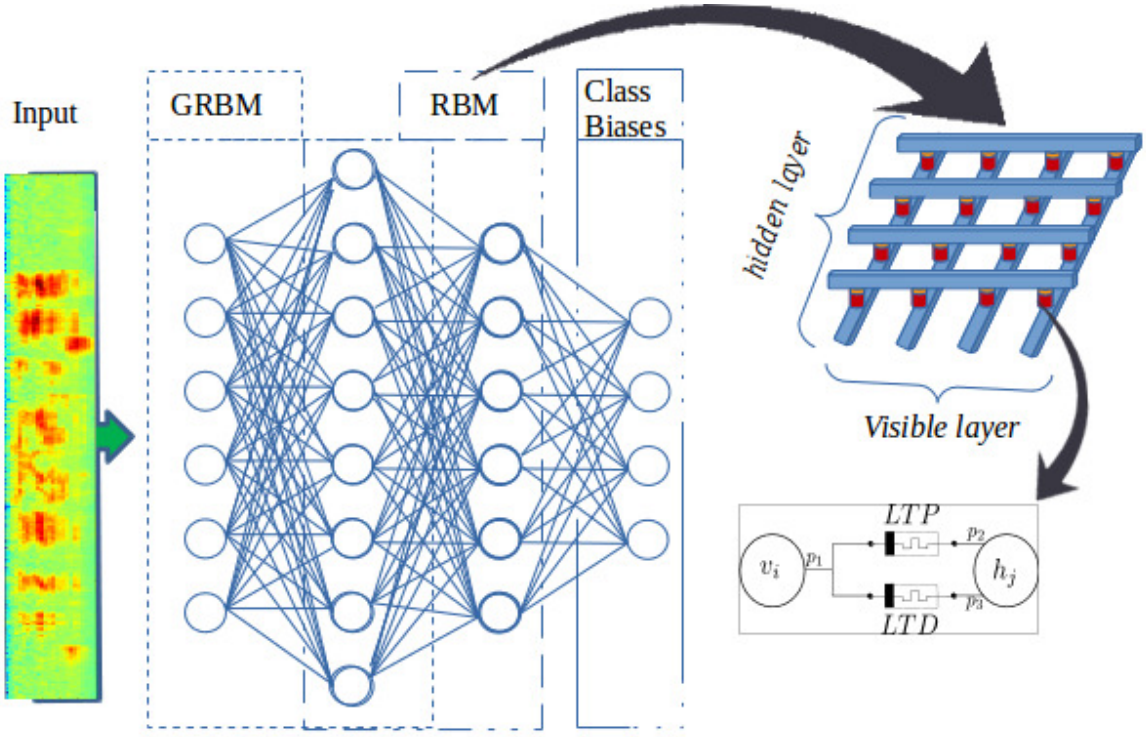


Figure 4.1: This figure shows the hybrid neuromorphic ASR system diagram. It takes the input from the sound MFCCs, and GRBM encodes coefficients to binary values, whereas output of the hidden nodes from GRBM is passed as input to neuromorphic RBM. Class biases are part of visible layer of RBM during training. The right side shows the cross-bass switch and its two memristors synapse component.

noise in the data. The second layer of the network, the standard RBM, is trained with the encoded data. The standard RBM is fitted with memristors synapse, wherein synapse values are adjusted by CD weights adjust mechanism. The two memristors synapse changes the capacitance of the first memristor on increment of the weights while the second memristor's capacitance remains unchanged. On decrement of the weight changes the procedure is vice versa. The potentiation and the depression are represented by different memristors in a single synapse.

The classification is achieved by appending a biased vector to encoded sparse vector, where biased vector represents class of the sound file. The neuromorphic RBM is trained with this extended input data. The class of the tested sound is gauged by computing the free energy [10] of the second RBM. The final class is identified as the one which is represented by the added class biased code and gives the lowest free energy, among all the inputs appended as biases of the classes.

4.2 Experimentation and Results

The experiments are thoroughly designed to test the discussed neuromorphic architecture, in which robust testing is performed to verify the efficacy of this system. The data set[1] is acquired from a previous study of the characteristics of the vowels from most of the American speakers [43]. The data set contains recordings of vowels from 45 men, 48 women and 46 children; Children includes boys and girls aged between 10 to 12. The vowels are recorded by reading each word from the list: {ae="had", ah="hod", aw="hawed", eh="head", er="heard", ei="haid", ih="hid", iy="heed", oa= "hoyed", oo="hood", uh="hud", uw="who'd"}. The voice was low-pass filtered at 7.2 kHz and digitized at 16kHz with 12 bits of amplitude resolution. Sound files are used for training of neural networks, but each file is converted to its Mel-frequency cepstral coefficients (MFCCs). The top 12 values of MFCCs and their first and second derivatives are combined in a single vector. The data is normalized with zero mean and unit variance.

[1] http://homepages.wmich.edu/ hillenbr/voweldata.html

Table 4.1: Results of the speech recognition for an average of 10 runs with each formation of the hidden layer

		Neuromorphic System		Standard weights matrix	
Hidden Layer 2	Test Examples	Correct Classifications	Percentage range	Correct Classification	Percentage range
500	168	143	(85.11±2) %	159	(94.16±1.0) %
500	1668	1338	(80.23 ± 1.5) %	1568	(94.11±1.0 %
1000	168	132	(78.3±3.0) %	160	(95.2±2.0) %
1000	1668	1312	(79.1±2.0) %	1553	(93.2±2.0) %

The normalized MFCCs data is used to train neural network. The first layer is trained with Gaussian-Bernoulli RBM, with 2160 visible layer neurons and 2000 hidden layer neurons; Visible neurons are, $(12 \times 3 \times 60)$, 12 MFCCs, their first and second derivatives and the constrained power spectrum of 60 values. The encoded sparse representation is used to train binary-binary neuromorphic RBM. The visible layer of neuromorphic RBM consists of $2000 + 144$ neurons where the additional 144 are class code of each of the 12 classes of vowels representing by 12 bits each. However, hidden layer of the second RBM is tested with different formations of neurons. Furthermore, training parameters of two constituent networks are noted as follows: GRBM uses learning rate (0.0001), momentum (0.9), sigma (0.07) and CD step size (5). The neuromorphic layer uses the learning rate (0.001) and CD step-size (1). The test sounds are classified by generating free energy of the second layer.

The results produced by the hybrid neuromorphic system, to our knowledge, are the first of its kind. Hence, Table 4.1 gives the quantitative results of this system

and equivalent standard RBM network. The results are also compared with human recognition results of the same data, and it is reported by the publisher as 95.4% averaged from 20 listeners. Fig. 4.2 and 4.3 provide insight in to the performance lacking due to the use of memristive synapse. The images illustrate the transition of weights from one visible neuron connection to ten hidden neurons. The behavior is evident from graph of the updates of the weights representing gradient decent for learning. Table 4.2 illustrates confusion matrix for classified vowels, and it can be noticed that wrong classification are confused consistently with, more or less, the same vowels. Another, important observation to note is that convergence of neuromorphic ASR is much faster than standard ASR neural networks i.e. former converges in less than 5 epochs with same training parameters while later converges in around 50 epochs.

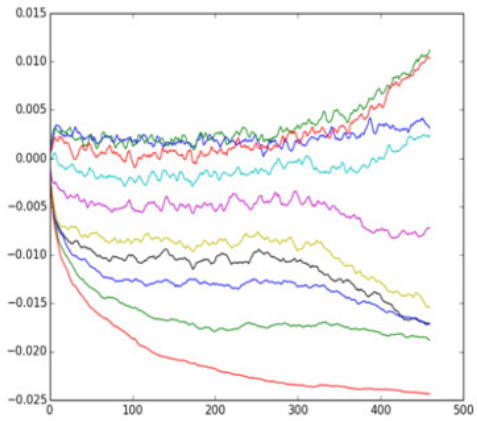


Figure 4.2: Real weights transition

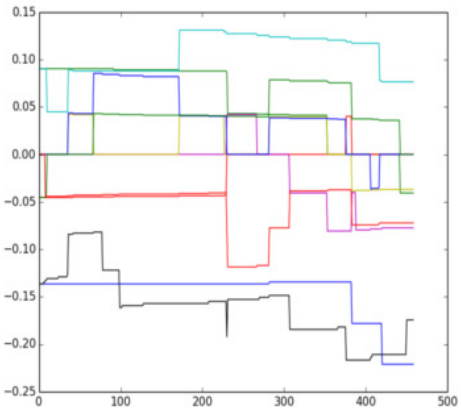


Figure 4.3: Memristor synapse plasticity transition

4.3 Discussion

The idea presented in this chapter is the speech recognition using a neuromorphic system, where the system uses memristor as a synapse. The neural network used in this system is learned on-line and the weights values represented by the memristor are updated with training. The dataset tested with the designed system contains 12 vowels which is relatively easy to train with memristors with unstable properties. However, the actual speech is much complex in nature, and it becomes hard to train it with available memristors as a synapse. A strategy to overcome the inherited limitation in memristors is to train the neural networks offline (in a computer simulation), and memristors are tuned to take those weight values.

As the real memristor can not produce the real value for weights, hence it is required to approximate the weights values. The approximation of the weights values is tested in recent experiments in [21, 50, 107]. The memristors can represent weight values which can be taken by $2, 3, 4, ..., 8$ bits. A future prospect with the given work in the chapter and the cited resource is to train the deep neural network in simulation and quantize the learned weight with values represented by the real memristor.

Table 4.2: Confusion matrix for 168 test examples with neuromorphic ASR of 2000 x 500 crossbar

	had	hod	hawed	head	haid	heard	hid	heed	hoyed	hood	hud	who'd
had	0.93	0.07										
hod		0.79	0.21									
hawed		0.07	0.93									
head	0.07			0.93								
haid					0.71		0.07	0.21				
heard						1.0						
hid					0.07		0.79	0.14				
heed								1.0				
hoyed			0.07						0.86			0.07
hood									0.07	0.71		0.21
hud		0.07	0.07						0.07		0.79	
who'd			0.07				0.07					0.86

Chapter 5

Vehicle license plate detection

Automated transportation systems are getting more important these days due to the development of autonomous driving and its applications to the transportation system, and its generic functional component is to identify a vehicle. Since a vehicle is instantaneously identified by its license plate (LP), its detection and recognition are considered of paramount relevance. Although LP detection has been an important issue in many research studies over the decades, we still face many challenges brought up by the advent of new technologies such as autonomous driving, Internet of things, big data, deep learning, and so forth. Hence, a revision of the problem and solutions without involving prior constraints is needed.

Most of the currently available methods used to detect LP in images and videos are realized under a certain constrained environment. The region of interest (ROI) is one of such constraints, as the LP is assumed to be localized in a specific region of the test image frame. Camera parameters and illumination conditions are other widely assumed constraints. A static camera is extensively used in practical applications, while the illumination conditions are considered as those offering high visibility. The aspect ratio of the license plate's dimensions, orientation and color patterns also narrow down the domain of the problem. Moreover, prior knowledge about different features of a LP permits us to design simple solutions which leverage the real time performance of the

system.

Since we intended to redefine the LP detection problem to counter upcoming challenges posed by the automation of the system, the constraints mentioned earlier must be revised or relaxed to build functionally viable solutions. An example where most of the constraints are violated is a mobile traffic monitoring system and requires a robust, accurate and fast LP detection. Another pragmatic situation is that LP may be partially visible or occluded, therefore, causing difficulties in detection. In particular, if the detection is used to hide the identity of a vehicle during live telecasts for privacy protection, there is not enough margin of time and error in detection. These examples suggest that LP detection becomes a complex problem and often demands very high accuracy. Moreover, the ROI in the above-mentioned examples is dynamic and LP is not located at fixed coordinated of the image frame.

In this study, we establish a basis of using object detection techniques for vehicle LP detection. The main contributions of the study are outlined as follows: 1) annotation of training images dataset by adding the category of a LP bounding box, 2) training and testing of state-of-the-art object detection techniques which include Region-based CNN and exemplar-SVM and 3) elimination of frequently used assumptions and consideration of practically sound conditions to design a general solution for LP detection. Additional aspects concern a detection of partial LPs, detection of the LP in every frame of a video sequence from videos coming from a camera mounted on a movable platform and Pan-Tilt-Zoom (PTZ) camera. Since the general applications of the LP detection needed LP recognition and the camera is mounted on a static platform with

known surroundings, so the earlier studies assumed the constrained environments. The main focus here is the detection of LP to identify or hide the identity of a vehicle for security purposes. Furthermore, the case-study of real-time performance of LP detection is covered as well.

5.1 Methodology

In the automatic transportation systems, vehicle LP detection is one of the most widely tackled problems that attracts the attention of various applications. We explore the problem without making prior constraints and extend the functional requirements which are discussed in Sections **??** and 5.3. An image processing technique with simple classification algorithm is discussed here. Furthermore, two distinct advanced object detection methods are considered for the LP detection. The formal formulation of the problem is posed as follows: Given an image or a frame coming from a video sequence, I_t, detect LPs $\{p_1, ..., p_n\}$, where n stands for the number of vehicles with visible plates in I_t. The position of vehicle and number of vehicles are not constrained. There are no bounds imposed on the size of LP, and it is assumed that the camera is not stationary. Another functional requirement is that the LPs are needed to be detected in every single frame of a video sequence.

5.1.1 Conventional Image Processing Technique

The conventional technique adopted is presented in the form of Algorithm 2. This technique uses color thresholding and image morphing techniques to search for probable

LPs region and then classifies the detected regions using SVM trained with LP dataset. We briefly elaborate on the successive steps of the algorithm: 1) Vehicles are segmented from the test image. If the data is a video sequence captured by a static camera, the background subtraction technique is helpful to reduce the search space carried out by the algorithm. 2) The test image is converted into gray scale image. 3) Noise reduction and smoothening of the test image is realized with the use of the Gaussian blur. 4) The smoothened image is binarized by applying thresholding. Global, adaptive and cluster thresholding techniques are suggested in case of varying illumination conditions. 5) Fine contours from the binarized image. 6) The contours are scrutinized for false positives to detect the actual LP. False positives are removed using heuristics like the aspect ratio of the contour, overlapping contours, and enclosed contour frequency. Moreover, a linear SVM is used to classify the remaining candidates for a final LP detection.

This approach outlined above does not perform well in varying conditions in which images are captured. It only performs well under certain assumptions such as static camera location for capturing the image, the position of the vehicle in an image, fixed illumination conditions and LP background color and characters information. The general issues inherent to the LP detection problem calls for a solution formulated in an unconstrained environment.

5.1.2 Support Vector Machine

Support vector machine (SVM) is a parametric model which is widely used for regression and classification problems. We deal with the automatic LP detection as

Algorithm 2: Use of conventional image processing techniques for LP detection

1: Select region using background subtraction technique with static camera from a video sequence

2: Convert Image I_t to gray scale G_t

3: Blur G_t and smoothen it, which will be B_t

4: Binarize B_t by applying dynamic thresholding th_t

5: Find contours C in th_t **for** *each contour c in C* **do**

6: remove c if it does not comply with the aspect ratio of height and width

7: remove c if enclosed contours count is less than the minimum characters

8: remove c if it does not comply with minimum and maximum sizes

9: remove c if it is not classified using SVM

10: Return C

an object detection problem. As discussed earlier, the automatic LP detection is a complex problem and the associated complexity include illumination conditions, large search space, various print style, digits, and color etc. These features allow us to regard LP as a separate object.

The ensemble of exemplar-SVM [55] showed better performance than single and kernel SVMs in object detection tasks. The ensemble of exemplar-SVM is used to detect LP in an image and video sequence. The ensemble of exemplar-SVMs exploits each individual positive example which distinguishes it from negative examples and classifies the similar objects for unseen data. A separate linear SVM (exemplar-SVM) is trained for each positive example along with many negative examples coming from the training set. Every exemplar SVM with parameters (W, b) for the positive example p and the set of negative examples $\{n_1, n_2, ..., n_m\}$ optimizes the following objective function to construct the decision boundary [66].

$$\omega(W, b) = ||W||^2 + C_1 h(W^T p + b) + C_2 \sum\nolimits_{n \in N} h(-W^T - b) \tag{5.1}$$

$$h(x) = max(0, 1 - x) \tag{5.2}$$

where C_1 and C_2 are constants and h is a hinge loss function. SVM is trained with hand crafted histogram of gradient (HOG) features of positive and negative training examples.

5.1.3 Region-based Convolutional Neural Networks (RCNN)

Recently CNN [58] has gained a reputation in a variety of computer vision applications due to its comprehensive feature representation capabilities and superior accuracy in object classification problems. Although CNN produced the best results among the competing classification methods, it is slow because of its architecture and a way of learning. With advances in hardware and implementation techniques, CNN can now be used for object detection problems with much higher accuracy and almost real-time processing speed.

CNN relies on filtering mechanism which designates the term convolution in the convolutional neural network. There are many layers of the neural network and hence, called as deep neural networks (DNN). In CNN, layers are a hierarchy of the convolutional layers with interleaving pooling layers. The convolution layer automatically detects features in the training data and the pooling layer reduces the size of feature images filtered by the convolutional layer.

Many different architectures have been recently proposed to improve results of object detection with reduced processing time. A recently proposed RCNN [34] architecture, to detect LP in an image, is discussed here. RCNN utilizes CNN together with a region proposal technique, where the proposed regions can be classified as detected objects. The idea of using selected (proposed) regions reduces the search space as compared to the conventional sliding window based detection methods, and thus decreases the execution time for detection. There are various techniques to compute the region proposals discussed in details in [13, 48]. RCNN also takes advantage of the semantics

of region proposal techniques and improves the detection accuracy. Therefore, it is one of the choices which helps to achieve the functional requirements of the discussed LP detection problem, i.e. assure 100 percent accuracy.

The other functional requirement is the real time execution of the method which is hard to achieve with the initially proposed RCNN. But an immediate successor of RCNN, Fast-RCNN [35] overcomes the slow execution time of its predecessor. Fast-RCNN adds an additional pooling layer to fine-tune ROI, which reclaims the lapse of separate stages in earlier proposed RCNN architecture. The pooling layer reduces the CNN execution time, but Fast-RCNN still uses a separate region proposal technique. The latest proposal is the Faster-RCNN [76], which uses a faster method of region proposals called as region proposal network (RPN). RPN uses features from the convolution layers to predict the bounding boxes for the object proposal and shares those features with the classification network i.e. Fast-RCNN. RPN is trained separately to the object detection network and combined with later in the test phase. The use of same CNN layers for region proposals and classification reduces the execution time for real-time performance.

5.2 Experimentation Studies

The suggested methods are rigorously tested to assess the performance of the LP detection in dynamic environments. To correctly classify the data we need not only design a good classifier but also collect a well-balanced number of data. In particular, for LP detection as an object detection problem, there are no standard datasets

available. Even in the other cases, where the prior training is not involved and image processing techniques are used, there are very few standard datasets available. The available datasets usually contain LPs from the same region of the world.

As discussed in 5.1 the use of image processing techniques and sophisticated object detection methods for vehicle LP detection is discussed here. There are numerous options available to try in each technique. We apply different combinations of image processing techniques in these experiments to detect LP. Similarly exemplar-SVM and RCNN have different options available at component levels. RCNN and its successors i.e. Fast-RCNN and Faster-RCNN are tested with different CNN configuration: ZF[104] and VGG16[84] CNN.

5.2.1 Dataset

We extend Pascal VOC2007 [30] data by combining a publicly available LP dataset (lpdatabase)[1] [5] and a Korean LP dataset (KoreanLP)[2]. Images are transformed into training set format published by a public VOC2007 dataset. Test Images includes application-oriented license plate database (AOLPD) [49], lpdatabase and KoreanLP. AOLPD includes three categories AC(the camera is stationary and the vehicle in the image is also stationary or passing the camera with reduced speed), LE (the vehicle in the image is captured for a traffic violation) and RP (the vehicle in the image is captured with moving camera). Moreover, two test videos are included from the real world scenarios: The first video is captured using a PTZ camera installed at a

[1] http://www.medialab.ntua.gr/research/LPRdatabase.html

[2] http://mlv.gist.ac.kr

driveway and sample frames from different views are shown in Fig. 5.1. The second video is captured using Toshiba BU238MC camera mounted on the windshield of a vehicle while driving the vehicle through the university campus on a sunny day. Its sample images are shown in Fig. 5.2.

5.2.2 Detection Results

Techniques considered to detect LP are prudently tested with all datasets discussed in Section 5.2.1. There are not many datasets available for LP detection, so the comparison is limited to the publicly available datasets. For the evaluation criteria, the accuracy metrics is quantified by manual verification of the bounding boxes for the detected LPs. However, an automated system can be used with sophisticated datasets and can compare the results of the intersection-over-union method with an available ground truth.

The details of the programing codes used for experimentation is as follows: The conventional technique is implemented using C++ with OpenCV libraries. The exemplar-SVM is trained with the Matlab code provided by the author of the original work. Faster-RCNN and Fast-RCNN are also trained and tested with the code provided by the authors of the original work, who used Caffe [52] libraries for implementation. The training images data is same for all object detection techniques to make a fair comparison.

Table 5.1 summarizes test results for all datasets of images mentioned in section 5.2.1. Tests are performed with exemplar-SVM and Faster-RCNN with VGG16 and

ZF CNNs and their accuracies are given in the table. It is evident from the results that there is no single winner, but Faster-RCNN with ZF network beats the others in accuracy on average. Moreover, the percentage of correct detection is very high with every single object detection technique and difference is insignificant among them. Table 5.2 gives a comparison of test results of several methods on the AOLPD dataset. Although Gee-Sern Hsu's method beats VGG16 result in RP class, VGG16 stays on top on average. Table 5.3 displays comparative results of test results on lpdatabase with object detection techniques. It becomes apparent that the exemplar-SVM comes with the highest values of accuracy in comparison with accuracy reported for other methods.

Two videos are tested with an additional region proposal technique named as selective search [95] with Fast-RCNN. Table 5.4 gives details of images in the driveway video sequence and the respective results with each tested technique. RCNN with selective search as region proposal method and VGG16 network suggests the best results for the entire video, and it happened to be almost 100 percent. Faster-RCNN also performs well with both ZF and VGG16 networks, whereas exemplar-SVM and the conventional approach performs close when the plates are clearly visible. The second video is a hard case for all techniques because of the lens glare and vignetting effects. Exemplar-SVM performs well in this video, while Faster-RCNN also performs better if the threshold of comparison is reduced. The dataset has 90 examples with visible LPs, and the exemplar-SVM ranked on top by detecting 41 right plates. It can be seen in Fig. 5.2 that the plates are hardly visible with the human eye.

5.2.3 Tuning

Results discussed in section 5.2.2 from object detection techniques are trained with the same training set and threshold comparison. However, results can be improved in exemplar-SVM and Faster-RCNN with certain changes. It is noted that the exemplar-SVM detects false positives from headlight and chassis regions of the vehicle. We train SVM with additional negative examples containing headlights and chassis regions. This phenomenon along with improved results is depicted in Fig. 5.3. Additional training with more positive examples improves results further as given in table 5.5. The exemplar-SVM is trained with additional 8 positive examples from AOLPD dataset and significant improvement in detection can be seen in the bar chart in Fig. 5.4.

The Faster-RCNN detection results presented in earlier section are from a comparison threshold of 0.5 with detection score. The detection results can be improved by using a smaller value for the threshold comparison, but it will increase the number of false positive in each image as shown in Fig. 5.6 and Fig. 5.7. Table 5.6 gives an effect of the threshold value on detection results, while graphs in Fig. 5.5 shows an exponential increase in false positives. It can be noted that false positive shoots after a threshold value of 0.1, but detection results improve.

5.3 Related Studies

Vehicle LP detection exhibits various applications in automated traffic management, surveillance, portable gadgets, automated parking ticketing, video editing, automated street view generation, and others. There are generalized as well as application-specific

solutions proposed for the target problem, and a body of literature can be found instantly with a single search query on the Internet. Although we limit this section to recent publications, references of comprehensive review papers compensate for the limited citations. Studies [5][4] offer detailed reviews on the LP detection techniques; Additionally, a novel adaptive image segmentation technique using sliding concentric windows is presented to detect ROI at high speed. It also discussed a method which describes local irregularity in an image using standard deviation and mean value of an image. This idea is tested with 1,334 input images and resulted in 1,287 correct segmentations i.e. 96.5% accuracy and reported an increase in performance when certain limitations are applied. A comprehensive review covers each aspect of the problem and is considered informative for interested users [25]. Considering the fact that LP detection is one of the most worked problems, it is unusual that there aren't any standard dataset and evaluation criteria available.

The LP detection problem is frequently tackled by conventional image processing techniques. A recently proposed approach used a vertical edge detection algorithm to gain speed efficiency in vehicle LP detection [3] with an adaptive thresholding for binarization to produce candidate regions. It was reported that computational speed is 47.7 $ms/image$ and a success rate is 91.4% reported on 664 tested images. Optimized Gabor filters are used to detect LP in varying illumination conditions, where the filter parameters are estimated using fuzzy sets [91]. It reported 94.4% accuracy of detection on 933 tested images. Also, [101] presented an implementation of the mean shift filtering method with delineation of the cluster and simple post-processing for LP detection.

Another conventional approach, discussed in [57], reported 94.12% accuracy when processing 425 test images. It proposed a region-based filtering method for LP detection. The filtering method sifts selected regions generated by edge detection and morphological operations techniques. [7] is an example of the use of template-matching to detect the LP of Iranian vehicles. It discussed a modification of the color template-matching technique by analyzing the target pixel's color. The specific design of the Iranian LP is exploited with a strip search to fine tune the results. $1,150$ images were tested showing 96.6% detection rate and the average execution time of 0.70 $sec/image$. Gradient analysis of transition in pixel intensity is used to localize LP in [11] and considered the detection of geometrically distorted, unaligned and occluded LPs. The algorithm was tested with 2,500 natural-scene gray-level vehicle images with different backgrounds and ambient illumination. It was reported that a detection rate is about 94%.

Geometrical representations are also applied for detecting LP in images; An illustration is a LP localization method using label-movable maximally stable external region clique with a focus on characters and the template of the LP [38]. It used predetermined features of the LP color and character structure. Test results on 99 images containing Chinese LP showed 97% correct detections with 77 $ms/image$ computation time. Edge clustering is also used to identify regions for probable LP [49]. The clusters are formed with the use of application-oriented parameters and expectation-maximization (EM) algorithm. The edge clustering technique reported 93.33% correct detections in 2049 tested images and a maximum execution time of 0.32 $sec/image$. Another geometric

framework is given in [61], which proposed a rectangular shape detection in images. LP detection is a direct application of the rectangular shape detection algorithm. Furthermore, it introduced an *interestness* measure for candidate LPs, and the test performed on 100 images showed 97% correct detection with a computation time of 1.1 $sec/image$.

Detection of LP was treated as an object detection problem using Adaboost and SVM. These approaches mostly relied on a conventional ROI selection. An example of the former using gradient information and the classifier to detect LP is given in [99]. Moreover, candidates given by the classifier are scrutinized using certain heuristics and a voting-based method. A recall of 87.29% and precision of 62.31% were reported from test results on 4,087 images. Another use of Adaboost is combined with a tracker; the tracker keeps track of the ROI in a video sequence, and the detection is realized by cascading Adaboost classifier [98]. The detection algorithm is improved by the inclusion of hierarchal information fusion which reduces false positives. The improved technique is tested with 950 images showing 90.8% correct detections [103]. Traffic video datasets are targeted for detection and recognition of LP in [105]. Videos are tested with a method that uses edge features and Haar-like features to construct a cascaded Adaboost classifier. The published performance showed a 94% correct detection. It is important to refer to an unconventional way of using genetics algorithm for the LP detection which is tested with a comprehensive dataset of 3000 images with an average of 98% detection results in [85].

Here the problem of LP detection is explored without any constraints. It consider that the distinct features of a LP constitute a separate object. Thus, it reformulate the

LP detection problem and utilize current best object detection techniques to determine a viable solution. With this regard, it explore variants of support vector machines (SVM) and convolutional neural networks (CNN). The LP detection is rarely considered as a part of object detection problem in the existing literature, and there are not any standard datasets available for training. To address this issue, a transformation of existing LP database into a standard dataset is useful for object detection.

5.4 Discussion

Image processing techniques rely on simple features like color maps, edges, color histograms and binarization, so they are vulnerable to changing conditions. As mentioned in section 5.1.1, this technique is tested with different options. The adaptive binarization performs better in varying conditions and densities of the vehicle colors. Blobs detection applied to color features and edge features behave similarly in quantitative numbers as we do not consider the presupposed ROI. The final algorithm, which is empirically found best suited for LP detection uses the Otsu binarization with grayscale blur images. Accuracy depends on texture in test image like dust and vehicle color patterns, as blobs and false positives are increased. Another factor with a high impact on final accuracy is pattern matching technique, which is a linear SVM using color histograms. SVM is used to remove false positives from the final filtered blobs in initial detection. It is noted that careful selection of features and training examples have a positive impact on the final detection. It takes around 0.35 *sec* to detect LPs in an 640×480 image using Algorithm 2.

Ensemble of exemplar-SVMs also depends on the quality and features of the positive examples. It uses each example to train a separate SVM, so its processing time goes up with the addition of every positive example. However, the experiments show that this impact is negligible if positive examples are below a certain number, which is tested as 180 examples. Furthermore, it can be rightly assumed that the parallelization can define the upper bound on the processing speed as each exemplar is compared separately for the match. Another experiment performed, is to train the exemplar-SVM with LPs from a certain region (i.e. European LPs) and test with a different region (i.e. KoreanLP), which performed very well. This experiment explores the ability of object detection techniques to exploit the features of a LP more than its contents. Although the ensemble of exemplar-SVMs performed remarkably well with accurate detections, but it is not suitable for real-time systems due to its slow computation time i.e. 3 $sec/image$. The accuracy is improved with continuous training as the examples can be added and removed on-line.

Faster-RCNN are best suited for real times systems. The region proposal methods play a major role in accurate detection as shown by the results with tested videos. Faster-RCNN with a ZF CNN network performs below par with a video dataset because RPN uses rectangular regions for the proposal. But Faster-RCNN with ZF CNN is the fastest with 0.07 $sec/image$, while VGG16 CNN is acceptable with 0.16 $sec/image$. The selective search is computationally expensive, thus does not qualify for a real-time system. However, the accuracy with selective search is comparatively higher than RPN with tests performed. Considering the real-time performance on an image dataset, ZF

being the less deep network produced better results than VGG16 which is a relatively deep network in some datasets.

License plate detection using RCNN relies on the proposal generated by a region proposal technique, which computes the region proposals based on the current test frame. It can be noted that in videos the information in consecutive frames is redundant, and that information can help in robust and accurate region proposal. Considering the fact that an LP detected in the first frame of the video sequence, the successive frames can use the detection results to generate robust region proposals. A feedback system can be introduced to make use of redundant information and generate regions proposals. The proposals generated using some feedback mechanism has potential use in object tracking. The computation time can also be reduced by generating fewer and more accurate proposals to check for final object classification.

Figure 5.1: Sample images from driveway video captured using a PTZ camera.

Figure 5.2: Sample images from campus video which are captured with a camera mounted on a moving vehicle.

Table 5.1: LP detection rates for tested datasets.

		Total Images	**Exemplar SVM**	**RCNN**	
				VGG16	ZF
AOLPD	AC	681	52 (92.36 %)	13 (98.09 %)	**6 (99.11 %)**
	LE	757	25 (96.69 %)	46 (93.92 %)	**18 (97.62 %)**
	RP	611	101 (83.46 %)	**67 (89.03 %)**	110 (81.996 %)
lpdatabase		710	**7 (98.79 %)**	18 (96.90 %)	10 (98.28 %)
KoreanLP		112	12 (89.28 %)	10 (91.07 %)	**6 (94.64 %)**
Total		2871	197 (92.81 %)	154 (94.38 %)	**150 (94.53 %)**

Table 5.2: Comparison of AOLPD with VGG16. Other results are borrowed from [49]

.

Technique	**AC**	**LE**	**RP**	**Average**
J. Ilonen[51]	81%	70%	72%	74.25%
C. Anagnostopoulos[5]	87%	81%	77%	81.80%
Gee-Sern Hsu[49]	93%	93%	**94%**	93.29%
VGG16[our]	**98.09%**	**93.92%**	89.03%	**93.85%**

Table 5.3: Comparison of lpdatabase with our tested techniques

Technique		**Total Images**	**Correct Detection**	**Accuracy Percentage**
C. Anagnostopoulos[5]		685	657	95.91 %
RCNN	VGG16	710	691	97.32 %
	ZF	710	701	98.73 %
Exemplar-SVM		710	**703**	**99.01 %**

Table 5.4: Driveway video frames categories and test results from techniques suggested

Video Frames	Images (5561)	Conventional Technique	Exemplar-SVM	RCNN		
				Selective Search-VGG16	RPN-VGG16	RPN-ZF
Visible LPs	**2702**	1952 (71.76%)	2520 (93.62%)	**2661 (96.6%)**	2423 (89.67%)	2309 (85.45%)
Partially Visible LPs	160	50 (31.25%)	75 (46.88%)	**151 (94.37%)**	142 (88.75%)	22 (13.75%)
Angled LPs	161	19 (11.8%)	25 (15.52%)	**141 (87.57%)**	88 (54.65%)	13 (8.07%)
No LPs	2538	-	-	-	-	-
Total LPs	3023	1955 (64.67%)	2620 (86.67%)	**2903 (96.03%)**	2653 (87.76%)	2433 (80.56%)

Figure 5.3: Left column shows results of exemplar-SVM before training with additional negative examples and right column shows detection results after additional training.

Table 5.5: Exemplar-SVM detection results with additional positive training images coming from AOLPD dataset

		Total Images	**Exemplar-SVM** (Undetected)	**Exemplar-SVM with additional training** (Undetected)	**Percentage improvement**
AOLPD	AC	681	52	43	1.32 %
	LE	757	25	17	1.05%
	RP	611	101	38	10.31%
lpdatabase		710	7	2	0.7 %
KoreanLP		112	12	12	0

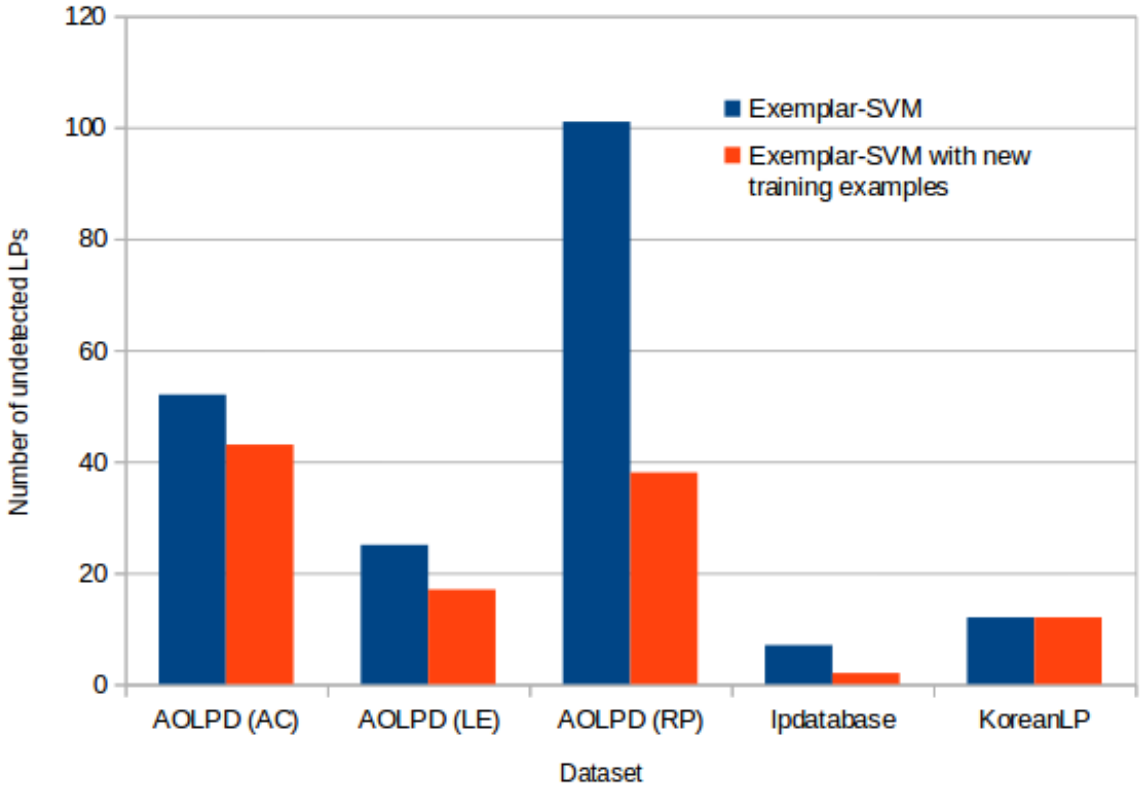


Figure 5.4: Exemplar-SVM bar chart showing the comparison of results with old and new training sets.

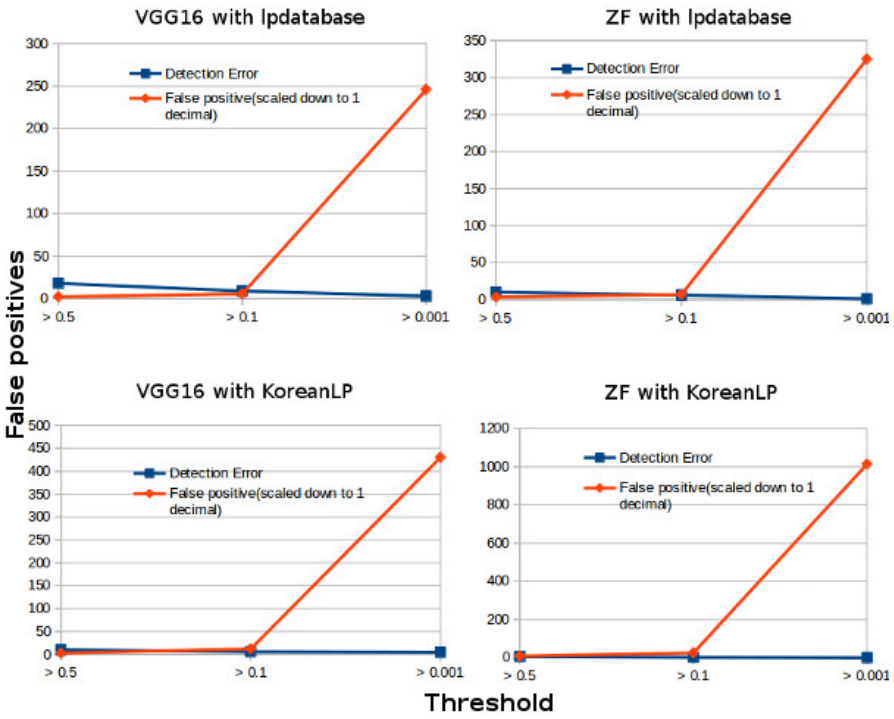


Figure 5.5: False positive vs detection error graphs with different threshold values in Faster-RCNN. x-axis is threshold values and y-axis is number of false positives.

Figure 5.6: Each row shows detection results of same image with different threshold value for Faster-RCNN with VGG16 network.

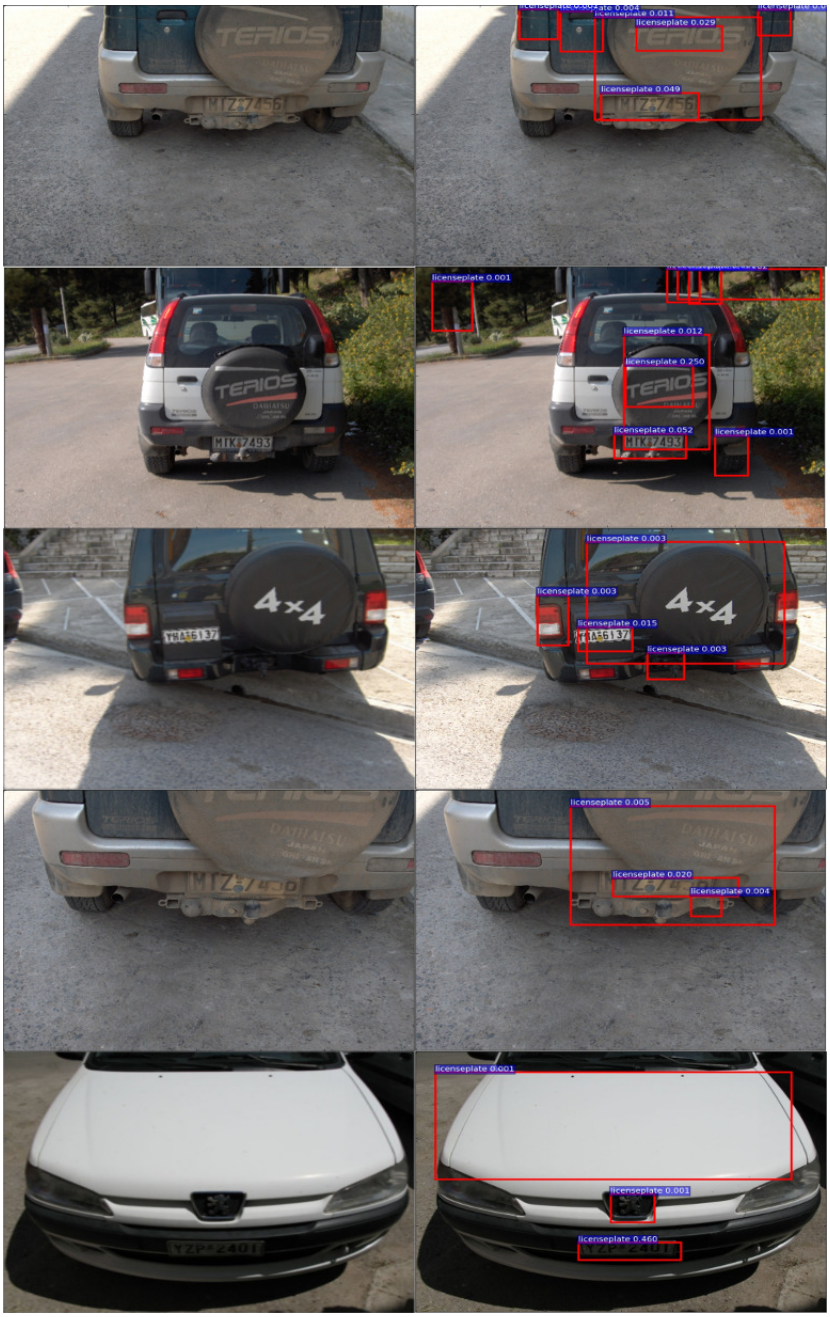


Figure 5.7: Left column shows detection results with threshold value of 0.5 and left column shows results with 0.001 threshold from Faster-RCNN with ZF.

Table 5.6: Detection and false positive results with different threshold values of Faster-RCNN with VGG16 and ZF networks.

	Total Images	threshold >0.5				threshold >0.1				threshold >0.001			
		Detection Error		False positive		Detection Error		False positive		Detection Error		False positive	
		VGG16	ZF	VGG16	ZF	VGG16	ZF	VGG16	ZF	VGG16	ZF	VGG16	ZF
lpdatabase	710	18	10	22	33	9	6	56	68	3	1	2460	3250
KoreanLP	112	10	6	3	8	6	2	12	25	5	0	430	1014

Chapter 6

Conclusion

6.1 Conclusions

Generative modeling, as an observation process for predictive learning, is a natural solution presented by artificial intelligence and machine learning. Computer vision, being the beneficiary of neural networks techniques, leaps forward through long promised solutions to the challenges. Generative modeling fills the gap in complex machine learning tasks like object detection, object classification, speech recognition, optical character recognition, and, so on, where enough labeled data is not available. Predictive learning using neural networks and generative modeling as the key structural components requires right choices. This thesis worked-out solutions to some common problems and achieved state-of-the-art results presenting the right choices for predictive learning using generative modelings techniques.

6.1.1 Background subtraction

The generative modeling capacity of GRBM for the background subtraction problem is discussed in this thesis. GRBM behaves as a constrained version of MoG, which and whose variants are widely used to solve the background subtraction problems. GRBM with implicit learning properties exploits the strength of MoG and learns the

variance associated with Gaussian visible neurons. It accepts raw pixel values of the RGB channels as input and trains a separate GRBM for each color channel only. We used the learned mean and variance provided by the GRBM for each pixel and channel to generate the background model in a video. The foreground extraction is the resultant of the combination of the output from three GRBMs and that result is processed by the simple post processing.

Experimental results show that this simple technique competes the state-of-the-art ensemble and feedback based background subtraction techniques. In addition, GRBM does not require any hand-crafted feature selection, and has the flexibility of working with various data sets by tuning a small number of parameters, and can continue learning to update the background model for changes in the environment cheaply. The research reported here can be extended to solve vision problems in deep neural networks.

The RBM has the capacity of representing the variation in the video sequence, but it may not reconstruct variant scenes because of the involved biases. A robust technique is required to make use of the information captured by the RBM. Furthermore, we can extend this research is to update the background model by updating the RBM partially.

6.1.2 Neuromorphic speech recognition

We discuss a neuromorphic multilayer neural network for speech recognition in this thesis. The network uses GRBM for sparse data representation of speech, and encoded data is used to train a neuromorphic RBM, on top of the first network. The

synapse connections between two layers of second RBM are PCMO RRAM memristors synapses. The connections are trained with the example data set to adjust synapse final weight as a real number. This proposed hybrid multilayer ASR, rightfully, constitute the basis for generative neuromorphic learning models, and the system can be extended with more deep neural networks and extensive experimentation with speech data. The network is tested with a comprehensive dataset of the American vowels in common use and compared with human performance. The technique discussed in this thesis proved itself, in a fair competition, considering the limitations of a memristive synapse.

The limitation of the memristive synapses is an apparent bottleneck in training the DNN. However, the learn-able parameters obtained from a network trained with software simulations can be approximated to the values representable by the two memristor synapse.

6.1.3 Vehicle license plate detection

we have discussed the state-of-the-art object detection algorithms to detect vehicle LPs. LPs have peculiar features which can be exploited in various environments to detect them as objects. We extended a standard dataset used for object detection and included LPs in object categories for the training of object detection methods. Exemplar-SVM and RCNN are trained with the dataset. Vehicle LP detection problems are assumed under constrained environments which are relieved in this discussion and have introduced additional functional requirements to cater to pragmatic situations. It is required that LP is detected in every single frame of the video obtained from a

moving camera and a partially visible plate is also detected.

The results obtained with the use of the developed techniques are superior to the results produced by the state-of-the-art LP detection techniques encountered in the literature. The object detection techniques are equally viable for LPs coming from any region and any environment. The convolution neural networks are found efficient in terms of the detection time to any simple image processing technique with additional graphics card hardware. A prospect for videos is to use the redundant information in a sequence of frames to reduce the region proposals in RCNN and increase efficiency in terms of reduced execution time.

References

[1] *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, 2011. IEEE. ISBN 978-1-4577-0539-7.

[2] Shyam Prasad Adhikari, Changju Yang, Hyongsuk Kim, and Leon O. Chua. Memristor bridge synapse-based neural network and its learning. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1426–1435, 2012. ISSN 2162237X. doi: 10.1109/TNNLS.2012.2204770.

[3] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail. Vertical-edge-based car-license-plate detection method. *IEEE Transactions on Vehicular Technology*, 62(1):26–38, Jan 2013. ISSN 0018-9545. doi: 10.1109/TVT.2012.2222454.

[4] Christos Nikolaos E Anagnostopoulos, Ioannis E. Anagnostopoulos, Ioannis D. Psoroulas, Vassili Loumos, and Eleftherios Kayafas. License plate recognition from still images and video sequences: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):377–391, 2008. ISSN 15249050. doi: 10.1109/TITS.2008.922938.

[5] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, V. Loumos, and E. Kayafas. A License Plate-Recognition Algorithm for Intelligent Transportation System Applications. *IEEE Transactions on Intelligent Transportation Systems*, 7(3): 377–392, 2006. ISSN 1524-9050. doi: 10.1109/TITS.2006.880641. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1688109.

[6] Kofi Appiah, Andrew Hunter, Patrick Dickinson, and Hongying Meng. Accelerated hardware video object segmentation: From foreground detection to connected components labelling. *Computer Vision and Image Understanding*, 114, 2010. doi: 10.1016/j.cviu.2010.03.021. URL http://dx.doi.org/10.1016/j.cviu.2010.03.021.

[7] A. H. Ashtari, M. J. Nordin, and M. Fathy. An iranian license plate recognition system based on color features. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1690–1705, Aug 2014. ISSN 1524-9050. doi: 10.1109/TITS.2014.2304515.

[8] O. Barnich and M. Van Droogenbroeck. Vibe: A powerful random technique to estimate the background in video sequences. In *ICASSP 2009.*, pages 945–948, 2009. doi: 10.1109/ICASSP.2009.4959741.

[9] A. Bayona, J.C. SanMiguel, and J.M. Martinez. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. In *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pages 25–30, Sept 2009. doi: 10.1109/AVSS.2009.35.

[10] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009. ISSN 1935-8237. doi: 10.1561/2200000006. URL http://dx.doi.org/10.1561/2200000006.

[11] Michael Broitman, Yuri Klopovsky, and Normunds Silinskis. License

plate detection algorithm. 9067:90670Z, 2013. doi: 10.1117/12.2051027. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2051027.

[12] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *CVPR, 2011*, pages 1937–1944. IEEE, June 2011. ISBN 978-1-4577-0394-2. doi: 10.1109/cvpr.2011.5995508. URL http://dx.doi.org/10.1109/cvpr.2011.5995508.

[13] Neelima Chavali, Harsh Agrawal, Aroma Mahendru, and Dhruv Batra. Object-proposal evaluation protocol is 'gameable'. *CoRR*, abs/1505.05836, 2015. URL http://arxiv.org/abs/1505.05836.

[14] Mingliang Chen, Qingxiong Yang, Qing Li, Gang Wang, and Ming-Hsuan Yang. *Spatiotemporal Background Subtraction Using Minimum Spanning Tree and Optical Flow*, pages 521–534. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10584-0. doi: 10.1007/978-3-319-10584-0{_}34. URL http://dx.doi.org/10.1007/978-3-319-10584-0{_}34.

[15] Mingqing Chen, Yefeng Zheng, K. Mueller, C. Rohkohl, G. Lauritsch, J. Boese, and D. Comaniciu. Enhancement of organ of interest via background subtraction in cone beam rotational angiocardiogram. In *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*, pages 622–625, May 2012. doi: 10.1109/ISBI.2012.6235625.

[16] Yingying Chen, Jinqiao Wang, and Hanqing Lu. Learning sharable models for

robust background subtraction. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, June 2015. doi: 10.1109/ICME.2015.7177419.

[17] KyungHyun Cho, Alexander Ilin, and Tapani Raiko. Improved learning of gaussian-bernoulli restricted boltzmann machines. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I*, ICANN'11, pages 10–17, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21734-0. URL http://dl.acm.org/citation.cfm?id=2029556.2029558.

[18] KyungHyun Cho, Tapani Raiko, and Alexander Ilin. Enhanced gradient and adaptive learning rate for training restricted boltzmann machines. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 105–112, 2011.

[19] Marc-Alexandre Côté and Hugo Larochelle. An infinite restricted boltzmann machine. *CoRR*, abs/1502.02476, 2015. URL http://arxiv.org/abs/1502.02476.

[20] Nicola Cottini, Massimo Gottardi, Nicola Massari, Roberto Passerone, and Zeev Smilansky. A 33 μ w 64×64 pixel vision sensor embedding robust dynamic background subtraction for event detection and scene interpretation. *IEEE Journal of Solid-State Circuits*, 48, 2013. doi: 10.1109/JSSC.2012.2235031. URL http://dx.doi.org/10.1109/JSSC.2012.2235031.

[21] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect:

Training deep neural networks with binary weights during propagations. *CoRR*, abs/1511.00363, 2015. URL http://arxiv.org/abs/1511.00363.

[22] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1337–1342, Oct 2003. ISSN 0162-8828. doi: 10.1109/TPAMI.2003.1233909.

[23] Dubravko Culibrk, Oge Marques, Daniel Socek, Hari Kalva, and Borko Furht. Neural network approach to background modeling for video object segmentation. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 18(6):1614–1627, 2007. doi: 10.1109/TNN.2007.896861. URL http://dx.doi.org/10.1109/TNN.2007.896861.

[24] M. Van Droogenbroeck and O. Paquot. Background subtraction: Experiments and improvements for vibe. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 32–37, June 2012. doi: 10.1109/CVPRW.2012.6238924.

[25] Shan Du, Mahmoud Ibrahim, Mohamed Shehata, and Wael Badawy. Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):311–325, 2013. ISSN 10518215. doi: 10.1109/TCSVT.2012.2203741.

[26] Shukai Duan, Xiaofang Hu, Zhekang Dong, Lidan Wang, and Pinaki Mazumder. Memristor-Based Cellular Nonlinear/Neural Network: Design, Analysis, and Ap-

plications. *IEEE transactions on neural networks and learning systems*, pages 1–12, 2014. ISSN 2162-2388. doi: 10.1109/TNNLS.2014.2334701. URL http://www.ncbi.nlm.nih.gov/pubmed/25069124.

[27] Ahmed Elgammal, David Harwood, and Larry Davis. *Non-parametric Model for Background Subtraction*, pages 751–767. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-45053-5. doi: 10.1007/3-540-45053-X{_}48. URL http://dx.doi.org/10.1007/3-540-45053-X{_}48.

[28] R.H. Evangelio, M. Patzold, and T. Sikora. Splitting gaussians in mixture models. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 300–305, Sept 2012. doi: 10.1109/AVSS.2012.69.

[29] Rubén Heras Evangelio, Michael Patzold, and Thomas Sikora. Adaptively splitted GMM with feedback improvement for the task of background subtraction. *IEEE Transactions on Information Forensics and Security*, 9, 2014. doi: 10.1109/TIFS.2014.2313919. URL http://dx.doi.org/10.1109/TIFS.2014.2313919.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2007 (voc2007) results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[31] Alessio Ferone, Lucia Maddalena, and Alfredo Petrosino. Neural moving object detection by pan-tilt-zoom cameras. In *Neural Nets and Surroundings*, volume 19 of *Smart Innovation, Systems and Technologies*, pages 129–138. Springer, 2013.

ISBN 978-3-642-35466-3. doi: 10.1007/978-3-642-35467-0_14. URL http://dx.doi.org/10.1007/978-3-642-35467-0_14.

[32] A Fischer and C Igel. Empirical Analysis of the Divergence of Gibbs Sampling Based Learning Algorithms for Restricted Boltzmann Machines. *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 208–217, 2010.

[33] Yoav Freund and David Haussler. Unsupervised learning of distributions of binary vectors using 2-layer networks. In *NIPS*, pages 912–919, 1991.

[34] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(1):142–158, Jan 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2437384.

[35] Ross Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[36] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. Changedetection.net: A new change detection benchmark dataset. In *CVPR Workshops*, pages 1–8. IEEE, 2012. ISBN 978-1-4673-1611-8. URL http://dblp.uni-trier.de/db/conf/cvpr/cvprw2012.html#GoyetteJPKI12.

[37] M. D. Gregorio and M. Giordano. Change detection with weightless neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 409–413, June 2014. doi: 10.1109/CVPRW.2014.66.

[38] Qin Gu, Jianyu Yang, Lingjiang Kong, and Guolong Cui. Multi-scaled license plate detection based on the label-moveable maximal MSER clique. *Optical Review*, 22(4):669–678, 2015. ISSN 1340-6000. doi: 10.1007/s10043-015-0103-8. URL http://link.springer.com/10.1007/s10043-015-0103-8.

[39] Rui Guo and Hairong Qi. Partially-sparse restricted boltzmann machine for background modeling and subtraction. In *2013 12th International Conference on Machine Learning and Applications*, volume 1, pages 209–214, Dec 2013. doi: 10.1109/ICMLA.2013.43.

[40] Tom S. F. Haines and Tao Xiang. Background subtraction with dirichlet processes. In *ECCV (4)*, volume 7575 of *Lecture Notes in Computer Science*, pages 99–113. Springer, 2012. ISBN 978-3-642-33764-2. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2012-4.html#HainesX12.

[41] Tom S.F. Haines and Tao Xiang. Background subtraction with DirichletProcess mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 2014. doi: 10.1109/TPAMI.2013.239. URL http://dx.doi.org/10.1109/TPAMI.2013.239.

[42] Jeff Hawkins. *On Intelligence (with Sandra Blakeslee)*. Times Books, 2004.

[43] J Hillenbrand, L a Getty, M J Clark, and K Wheeler. Acoustic characteristics of American English vowels. *The Journal of the Acoustical Society of America*, 97 (5 Pt 1):3099–3111, 1995. ISSN 0001-4966. doi: 10.1121/1.411872.

[44] Geoffrey Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018. URL http://dx.doi.org/10.1162/089976602760128018.

[45] GeoffreyE. Hinton. A practical guide to training restricted boltzmann machines. In Grégoire Montavon, GenevièveB. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pages 599–619. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35288-1. doi: 10.1007/978-3-642-35289-8_32. URL http://dx.doi.org/10.1007/978-3-642-35289-8_32.

[46] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38–43, June 2012. doi: 10.1109/CVPRW.2012.6238925.

[47] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8): 2554–2558, April 1982. ISSN 1091-6490. URL http://www.pnas.org/content/79/8/2554.abstract.

[48] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine In-*

telligence, 38(4):814–830, April 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2015.2465908.

[49] G. S. Hsu, J. C. Chen, and Y. Z. Chung. Application-oriented license plate recognition. *IEEE Transactions on Vehicular Technology*, 62(2):552–561, Feb 2013. ISSN 0018-9545. doi: 10.1109/TVT.2012.2226218.

[50] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *CoRR*, abs/1609.07061, 2016. URL http://arxiv.org/abs/1609.07061.

[51] J. Ilonen, J.-K. Kamarainen, P. Paalanen, M. Hamouz, J. Kittler, and H. Kalviainen. Image feature localization by multiple hypothesis testing of gabor features. *Image Processing, IEEE Transactions on*, 17(3):311–325, March 2008. ISSN 1057-7149. doi: 10.1109/TIP.2007.916052.

[52] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[53] Daniel D. Kelson. Optimal techniques in twodimensional spectroscopy: Background subtraction for the 21st century. *Publications of the Astronomical Society of the Pacific*, 115(808):pp. 688–699, 2003. ISSN 00046280. URL http://www.jstor.org/stable/10.1086/375502.

[54] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, and Larry Davis. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging*, 11, 2005. doi: 10.1016/j.rti.2004.12.004. URL http://dx.doi.org/10.1016/j.rti.2004.12.004.

[55] T. Kobayashi. Three viewpoints toward exemplar svm. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 2765–2773, June 2015. doi: 10.1109/CVPR.2015.7298893.

[56] Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, University of Toronto, April 2009.

[57] Mahmood Ashoori Lalimi, Sedigheh Ghofrani, and Des McLernon. A vehicle license plate detection method using region and edge based methods. *Computers & Electrical Engineering*, 39(3):834–845, 2012. ISSN 00457906. doi: 10.1016/j.compeleceng.2012.09.015. URL http://dx.doi.org/10.1016/j.compeleceng.2012.09.015.

[58] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

[59] Jeisung Lee and Mignon Park. An adaptive background subtraction method based on kernel density estimation. *Sensors*, 12(9):12279–12300, 2012. doi: 10.3390/s120912279. URL http://dx.doi.org/10.3390/s120912279.

[60] Liyuan Li, Weimin Huang, Irene Gu, and Qi Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 13(11):1459–1472, 2004. doi: 10.1109/TIP.2004.836169. URL http://dx.doi.org/10.1109/TIP.2004.836169.

[61] Qi Li. A geometric framework for rectangular shape detection. *IEEE Transactions on Image Processing*, 23(9):4139–4149, Sept 2014. ISSN 1057-7149. doi: 10.1109/TIP.2014.2343456.

[62] Zhizhong Li and Derek Hoiem. *Learning Without Forgetting*, pages 614–629. Springer International Publishing, Cham, 2016. ISBN 978-3-319-46493-0. doi: 10.1007/978-3-319-46493-0_37. URL http://dx.doi.org/10.1007/978-3-319-46493-0_37.

[63] A.K. Maan, A.P. James, and S. Dimitrijev. Memristor pattern recogniser: isolated speech word recognition. *Electronics Letters*, 51(17):1370–1372, 2015. ISSN 0013-5194. doi: 10.1049/el.2015.1428.

[64] Lucia Maddalena and Alfredo Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 17(7):1168–1177, 2008. doi: 10.1109/TIP.2008.924285. URL http://dx.doi.org/10.1109/TIP.2008.924285.

[65] Lucia Maddalena and Alfredo Petrosino. The sobs algorithm: What are the

limits? In *CVPR Workshops*, pages 21–26. IEEE, 2012. ISBN 978-1-4673-1611-8. URL http://dblp.uni-trier.de/db/conf/cvpr/cvprw2012.html#MaddalenaP12.

[66] Tomasz Malisiewicz. *Exemplar-based Representations for Object Detection, Association and Beyond*. PhD thesis, The Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania 15213, 7 2011.

[67] Manjunath Narayana, Allen R. Hanson, and Erik G. Learned-Miller. Background modeling using adaptive pixelwise kernel variances in a hybrid feature space. In *CVPR*, pages 2104–2111. IEEE, 2012. ISBN 978-1-4673-1226-4. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#NarayanaHL12.

[68] SeungJong Noh and Moongu Jeon. A new framework for background subtraction using multiple cues. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part III*, pages 493–506. Springer-Verlag, 2013. ISBN 978-3-642-37430-2. doi: 10.1007/978-3-642-37431-9_38. URL http://dx.doi.org/10.1007/978-3-642-37431-9_38.

[69] Jong Geun Park and Chulhee Lee. Bayesian rule-based complex background modeling and foreground detection. *Optical Engineering*, 49(2):027006–027006–11, 2010. doi: 10.1117/1.3319820. URL http://dx.doi.org/10.1117/1.3319820.

[70] Sangsu Park, Seungjae Jung, Manzar Siddik, Minseok Jo, Joonmyoung Lee, Jubong Park, Wootae Lee, Seonghyun Kim, Sharif Md Sadaf, Xinjun Liu, and Others. Memristive switching behavior in Pr0. 7Ca0. 3MnO3 by incorporating

an oxygen-deficient layer. *physica status solidi (RRL)-Rapid Research Letters*, 5 (10-11):409–411, 2011.

[71] Chiranjeevi Pojala and Somnath Sengupta. Neighborhood supported model level fuzzy aggregation for moving object segmentation. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 2013. doi: 10.1109/TIP.2013.2285598. URL http://dx.doi.org/10.1109/TIP.2013.2285598.

[72] M. A. Rafique, B. G. Lee, and M. Jeon. Hybrid neuromorphic system for automatic speech recognition. *Electronics Letters*, 52(17):1428–1430, 2016. ISSN 0013-5194. doi: 10.1049/el.2016.0975.

[73] Muhammad Aasim Rafique, Ahmed Sheri, Oh June, and Moongu Jeon. Background subtraction with restricted boltzmann machine. In *SSP'14 (2014 IEEE Statistical Signal Processing Workshop (SSP)) (SSP'14)*, Gold Coast, Australia, June 2014.

[74] Muhammad Aasim Rafique, Witold Pedrycz, and Moongu Jeon. Vehicle license plate detection using region-based convolutional neural networks. *Soft Computing*, Jun 2017. ISSN 1433-7479. doi: 10.1007/s00500-017-2696-2. URL https://doi.org/10.1007/s00500-017-2696-2.

[75] Vikas Reddy, Conrad Sanderson, and Brian C. Lovell. Improved foreground detection via Block-Based classifier cascade with probabilistic decision integration. *IEEE Transactions on Circuits and Systems for Video Technology*, 23, 2013.

doi: 10.1109/TCSVT.2012.2203199. URL http://dx.doi.org/10.1109/TCSVT.2012.2203199.

[76] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015.

[77] H. Sajid and S. C. S. Cheung. Background subtraction for static amp; moving camera. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4530–4534, Sept 2015. doi: 10.1109/ICIP.2015.7351664.

[78] H. Schulz, A. Müller, and S. Behnke. Investigating Convergence of Restricted Boltzmann Machine Learning. In *Advances in Neural Information Processing Systems (NIPS), Deep Learning and Unsupervised Feature Learning Workshop*, 2010.

[79] M. Sedky, M. Moniri, and C. C. Chibelushi. Spectral-360: A physics-based technique for change detection. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 405–408, June 2014. doi: 10.1109/CVPRW.2014.65.

[80] Yaser Sheikh and Mubarak Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 27 (11):1778–1792, 2005. doi: 10.1109/TPAMI.2005.213. URL http://dx.doi.org/10.1109/TPAMI.2005.213.

[81] Yiran Shen, Wen Hu, Junbin Liu, Mingrui Yang, Bo Wei, and Chun Tung Chou. Efficient background subtraction for real-time tracking in embedded camera networks. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 295–308, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1169-4. doi: 10.1145/2426656.2426686. URL http://doi.acm.org/10.1145/2426656.2426686.

[82] Ahmad Muqeem Sheri, Student Member, Hyunsang Hwang, Moongu Jeon, and Byung-geun Lee. Neuromorphic Character Recognition System With Two PCMO Memristors as a Synapse. *IEEE transactions on industrial electronics*, 61(6): 2933–2941, 2014. ISSN 0278-0046. doi: 10.1109/TIE.2013.2275966.

[83] Ahmad Muqeem Sheri, Aasim Rafique, Witold Pedrycz, and Moongu Jeon. Contrastive divergence for memristor-based restricted Boltzmann machine. *Engineering Applications of Artificial Intelligence*, 37:336–342, jan 2015. ISSN 09521976. doi: 10.1016/j.engappai.2014.09.013. URL http://www.sciencedirect.com/science/article/pii/S0952197614002334.

[84] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*, pages 1–13, 2014. URL http://arxiv.org/abs/1409.1556.

[85] G. Abo Smara and F. Khalefah. Localization of license plate number using dynamic image processing techniques and genetic algorithms. *IEEE Transactions*

on Evolutionary Computation, 18(2):244–257, April 2014. ISSN 1089-778X. doi: 10.1109/TEVC.2013.2255611.

[86] Raffaele Solimene, A. Cuccaro, A. DellAversano, Ilaria Catapano, and Francesco Soldovieri. Ground clutter removal in GPR surveys. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7, 2014. doi: 10.1109/JSTARS.2013.2287016. URL http://dx.doi.org/10.1109/JSTARS.2013.2287016.

[87] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin. A self-adjusting approach to change detection based on background word consensus. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 990–997, Jan 2015. doi: 10.1109/WACV.2015.137.

[88] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin. Subsense: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, Jan 2015. ISSN 1057-7149. doi: 10.1109/TIP.2014.2378053.

[89] Chris Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR, 1999.*, volume 2, pages 246–252 Vol. 2. IEEE, 1999. ISBN 0-7695-0149-4. doi: 10.1109/cvpr.1999.784637. URL http://dx.doi.org/10.1109/cvpr.1999.784637.

[90] Martin Stommel, Michael Beetz, and Weiliang Xu. Inpainting of missing values in the kinect sensor's depth maps based on background estimates. *IEEE Sensors*

Journal, 14, 2014. doi: 10.1109/JSEN.2013.2291315. URL http://dx.doi.org/10.1109/JSEN.2013.2291315.

[91] Vladimir Tadic, Miodrag Popovic, and Peter Odry. Fuzzified gabor filter for license plate detection. *Engineering Applications of Artificial Intelligence*, 48: 40 – 58, 2016. ISSN 0952-1976. doi: http://dx.doi.org/10.1016/j.engappai.2015.09.009. URL http://www.sciencedirect.com/science/article/pii/S0952197615002092.

[92] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: principles and practice of background maintenance. In *IEEE International Conference on Computer Vision*, volume 1, pages 255–261 vol.1. IEEE, 1999. ISBN 0-7695-0164-8. doi: 10.1109/iccv.1999.791228. URL http://dx.doi.org/10.1109/iccv.1999.791228.

[93] Son Ngoc Truong, Seok-Jin Ham, and Kyeong-Sik Min. Neuromorphic crossbar circuit with nanoscale filamentary-switching binary memristors for speech recognition. *Nanoscale research letters*, 9(1):629, jan 2014. ISSN 1931-7573. doi: 10.1186/1556-276X-9-629. URL http://www.nanoscalereslett.com/content/9/1/629.

[94] Du-Ming Tsai and Shia-Chih Lai. Independent component analysis-based background subtraction for indoor surveillance. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 18(1):158–167, 2009.

doi: 10.1109/TIP.2008.2007558. URL http://dx.doi.org/10.1109/TIP.2008.2007558.

[95] J R R Uijlings, K E A Sande, T Gevers, and A W M Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104 (2):154–171, 2013. ISSN 1573-1405. doi: 10.1007/s11263-013-0620-5. URL http://dx.doi.org/10.1007/s11263-013-0620-5.

[96] Nan Wang, Jan Melchior, and Laurenz Wiskott. Gaussian-binary restricted boltzmann machines on modeling natural image statistics. *CoRR*, abs/1401.5900, 2014.

[97] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan. Static and moving object detection using flux tensor with split gaussian models. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 420–424, June 2014. doi: 10.1109/CVPRW.2014.68.

[98] Runmin Wang, Nong Sang, Rui Huang, and Yuehuan Wang. License plate detection using gradient information and cascade detectors. *Optik - International Journal for Light and Electron Optics*, 125(1):186–190, 2014. ISSN 00304026. doi: 10.1016/j.ijleo.2013.06.008. URL http://dx.doi.org/10.1016/j.ijleo.2013.06.008.

[99] Runmin Wang, Nong Sang, Ruolin Wang, and Liangwei Jiang. Detection and tracking strategy for license plate detection in video. *Optik - International Journal for Light and Electron Optics*, 125(10):2283–2288, 2014. ISSN

00304026. doi: 10.1016/j.ijleo.2013.10.126. URL http://linkinghub.elsevier.com/retrieve/pii/S0030402614002253.

[100] Max Welling, Michal Rosen-zvi, and Geoffrey E. Hinton. Exponential family harmoniums with an application to information retrieval. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1481–1488. MIT Press, 2005.

[101] Xiong Xing, Byung-Jae Choi, Seog Chae, and Mun-Hee Lee. Design of a recognizing system for vehicle's license plates with english characters. *International Journal of Fuzzy Logic and Intelligent Systems*, 9(3):166 – 171, 2009.

[102] Linli Xu, Yitan Li, Yubo Wang, and Enhong Chen. Temporally adaptive restricted boltzmann machine for background modeling. In *AAAI Conference on Artificial Intelligence*, 2015. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9857.

[103] Zhenjie Yao and Weidong Yi. License plate detection based on multistage information fusion. *Information Fusion*, 18(1):78–85, 2014. ISSN 15662535. doi: 10.1016/j.inffus.2013.05.008. URL http://dx.doi.org/10.1016/j.inffus.2013.05.008.

[104] Md Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *Computer Vision–ECCV 2014*, 8689:818–833, 2014. ISSN 978-3-319-10589-5. doi: 10.1007/978-3-319-10590-1{_}53. URL http://link.springer.com/chapter/10.1007/978-3-319-10590-1{_}53.

[105] Lihong Zheng, Xiangjian He, Bijan Samali, and Laurence T. Yang. An algorithm for accuracy enhancement of license plate recognition. *Journal of Computer and System Sciences*, 79(2):245–255, 2013. ISSN 00220000. doi: 10.1016/j.jcss.2012.05.006. URL http://dx.doi.org/10.1016/j.jcss.2012.05.006.

[106] Xin Zheng, Zhiyong Wu, Helen M. Meng, Weifeng Li, and Lianhong Cai. Feature learning with gaussian restricted boltzmann machine for robust speech recognition. *CoRR*, abs/1309.6176, 2013.

[107] Shuchang Zhou, Zekun Ni, Xinyu Zhou, He Wen, Yuxin Wu, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *CoRR*, abs/1606.06160, 2016. URL http://arxiv.org/abs/1606.06160.

[108] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27, 2006. doi: 10.1016/j.patrec.2005.11.005. URL http://dx.doi.org/10.1016/j.patrec.2005.11.005.