Detection of Criminal Behavior at the Residential Unit based on Deep Convolutional Neural Network

H.A. Razak[1], N.K. Zakaria[4], N.F.M. Zamri[5]
College of Engineering, Universiti Teknologi MARA
Selangor, Malaysia

Ali Abd Almisreb[3]
Faculty of Engineering and Natural Sciences
International University of Sarajevo
Sarajevo, Bosnia-Herzegovina

Nooritawati Md Tahir[2]*
College of Engineering
Universiti Teknologi MARA
Shah Alam, Selangor, Malaysia
Institute for Big Data Analytics and
Artificial Intelligence (IBDAAI),
Universiti Teknologi MARA
Selangor, Malaysia

***Abstract*—Studies on abnormal behavior based on deep learning as a processing platform increase. Deep learning, specifically the convolutional neural network (CNN), is known for learning the features directly from the raw image. In return, CNN requires a high-performance hardware platform to accommodate its computational cost like AlexNet and VGG-16 with 62 million and 138 million parameters, respectively. Hence in this study, four CNN samplings with different architectures in detecting abnormal behavior at the gate of residential units are evaluated and validated. The forensic postures, with some other collected data, are used for the preliminary step in constructing the criminal case database. High accuracy up to 97% is obtained from the trained CNN samplings with 80% to 97% recognition rate achieved during the offline testing and 70% to 90% recognition rate recorded during the real-time testing. Results showed that the developed CNN samplings owned good performance and can be utilized in detecting and recognizing the normal and abnormal behavior at the gate of residential units.**

Keywords—Abnormal behavior; deep learning; convolution neural network; forensic posture; property crime

I. Introduction

There is an increase in the usage of closed-circuit television (CCTV) in residential units as a consequence of the upbringing awareness of sheltered zone [1]–[3]. This monotonous observation can cause fatigue and distraction, leading to negligence and being overlooked as the surveillance process is underway [4]. Currently, numerous studies are conducted to detect and track objects and people's anomalous state in developing intelligent surveillance systems [4]–[7], which is related to the changing pattern or movement of objects or humans from the original form or behavior, referred as anomalous. At present, image recognition is the most appropriate technology to utilize CCTV footage optimally, and the recognition can yield better results using deep learning techniques.

Deep learning is a hierarchical feature learning to classify multidimensional and complex data set elements. There are several types of deep learning structures that include convolutional neural network (CNN), long-short term memory (LSTM) and recurrent neural network (RNN). CNN is suitable for object detection and image recognition and has been widely used in numerous biometrics applications, namely fingerprint, iris, and gait [8]–[10]. Recall that gait biometrics can be used to identify subjects from their style or manner of walking. Due to its uniqueness, gait is also considered competent biometrics, suitable for forensic intelligent surveillance systems. This is because gait as biometric has the potential for farther distance recognition, can be possessed without the perpetrators' consent and awareness, and can also be perceived at a low-resolution camera. Combining these technologies, image recognition, CNN, and gait biometric brings us a little closer to developing a forensic intelligent surveillance system. However, the lack of data on criminal behavior in public databases leads to the problem of developing and designing adaptability features of forensic gait for recognition and detection.

Hence, the main objective of this study is to investigate and validate the forensic postures with the authorities' consent in interpreting anomalous behavior during the housebreaking crime in residential units. Four CNNs with different architectures, namely Up, Down, Up-Down, and Down-Up sampling, are developed to classify massive data on both normal and anomalous behavior. The effectiveness of the developed CNN samplings is examined with two test modes, offline and real-time detection. The offline detection is evaluated using several CCTV footage of the actual housebreaking crimes, and the real-time detection is held at the laboratory. It is essential to understand the architectures of CNN to develop a robust network at a minimum computational cost that can be a kick start in developing a robust and economical forensic intelligent surveillance system.

*Corresponding Author.

This research is funded by the Ministry of Higher Education (MOHE), Malaysia via the Fundamental Research Grant Scheme (FRGS) No: FRGS/1/2019/TK04/UITM/01/3.

II. Related Work

As mentioned earlier, CNN can be considered for gait recognition due to its outstanding realization. Currently, CNN comes in handy whenever needed to classify the image dataset without going through the pre-processing image and feature extraction step. These steps are handled by the multiple layers of nonlinear where the output from previous layers is the input to the following layers. CCN is able to learn automatically the significant features of large input image databases based on the pre-defined size of each filter in the convolution layers that establishes the convolution map uniformly [11]. The pooling layers minimize the redundant pixels based on the rescaling map, further reducing the feature matrix size of the convolution maps [12]. As for the convolution layer, the stochastic gradient descents with momentum (SGDM) optimization function and the activation function of the rectified linear unit (ReLU) are used accordingly during the learning process. This process continues with the features subset connected with each other that further developed the connection of the classification output layer. This is achieved as the first convolution maps preserved feature vectors provides learning to the second convolution layer [13]. For CNN, note that numbers of parameters are decreased accordingly during the convolution and pooling process, specifically the connections and shared weights [14].

As reported in [15], CNN is used for tracking humans based on numerous poses, viewpoints including occlusion, using ten challenging datasets. Here, CNN with five layers were developed that consists of convolutional, pooling, normalization, fully-connected and softmax layer. The 4 by 4 by k channels with two strides and zero-padding and 50 filter banks were used as the convolutional layer. As for the pooling layer, the filter size is 2 by 2 with a max operator and two strides and zero padding. For the normalization layer, the pre-defined hyperparameters are set as $k=1$, $\rho=2$, $\alpha=1/4$, and $\beta=0.5$. Next, the fully-connected layer flattened the extracted features associated with the softmax node. The classification performance was evaluated using the softmax operator and the log loss values with fixed hyperparameters during training; five as the maximum epoch, 0.001 as the learning rate, and ten as the batch size. The developed CNN was evaluated to track variation of occlusion using the women dataset since this dataset comprised numerous poses with partial occlusion of the lower and upper limb. Next, the basketball game video was used for testing body deformation variation. By averaging the Euclidean distance of the frames ground-truth positions and the person being tracked, the error of the centre location acted as the tracking result. Result attained was 91.31% using the basketball dataset and for the women dataset was 94.14%.

Conversely, as reported in [5], malicious activities were investigated for three anomalous behaviors based on six CNN layers, three convolutional layers, two fully connected layers, and one softmax layer. Filter size and total filters were fixed for all three convolutional layers. The same goes for the convolution stride, pooling and ReLU. The pooling layers utilized the max operator. Two experiments were conducted. The first fully-connected layers have 64 neurons as output, with the output neurons set as two and six accordingly for each experiment for the second fully-connected layer. Further, between the two fully-connected layers a ReLU layer was added. For each category, the probability was computed in the softmax loss layer. SGDM was used as the network optimization function with the learning rate range set from 10^{-3} to 10^{-1} whilst the epoch was set from 10 up to 100. Further, the developed CNN was tested using images based on five datasets comprised of normal and anomalous behaviors and variations. Ratio for training and testing was set as 70:30. However, for the PEL dataset, since most of the dataset consists of fighting scenes acquired from movies, the dataset was split as 43:57 as training and testing. Firstly, the algorithms classified the datasets into two categories; normal and abnormal, followed by classifying the abnormal category into three different anomalous behaviors. The three anomalous behaviors are punching, pushing and kicking. For all datasets, it was found that higher accuracy is achieved as the epoch is increase although more time is required for the learning rate to finally stabilize at 0.001. The accuracy attained was 100% for anomalous behaviors detection for both experiments based on the developed CNN.

Moreover, research on human behavior has gained the attention for safety community purposes, especially the large-scale industry since they are dealing with dozens and even hundreds of employees and equipment every day. As reported in [16], temporal information of human activities in the industry of each frame was processed using motion history image (MHI) and discrete cosine transform (DCT) to generate the 2D spatial-temporal maps. This process has successfully reduced the size of each frame from 704×576 to 88×72 with minimal information loss. These 2D maps were fed to the CNN for identifying human behavior and activity in the industrial environment. The developed CNN consists of three convolutional layers and one output layer of multi-layer perceptron (MLP). The first two convolutional layers have similar size with regards to the filter and zero padding. The number of filters in the second convolutional layer was 40, and it doubled from the first layer, which were 20 filters. The third convolutional layer had 60 filters with the dimension of 5×5. A pooling layer followed each convolution layer. Next, the final pooling layer flattened the convolutional feature vectors forming 1440-dimensional feature vectors. These feature vectors were inputs of 600 neurons of the first hidden layer of MLP and the output layer of six neurons representing the number of behaviors or activities to be classified. SCOVIS dataset contains heavy occlusion, the interaction between humans and machinery and factory environments were suitable to validate the network. 15 scenarios for training and 5 scenarios for testing were used for the SCOVIS dataset. Precision and recall were almost 99% for the training set and nearly 90% for the unseen dataset.

Recently, studies on anomalous behavior during driving environments have been very encouraging. As discussed in [17], deep learning based on CNN architectural was developed, known as DedistractedNet, that was used to classify the distracted driving behaviors like texting, drinking, putting on cosmetics and many more. This network has five sets of a convolutional layer, ReLU and max pooling layer, followed by fully connected layers with neurons corresponding to eight driver behaviors. The cross-entropy loss function computed the category loss of DedistractedNet. The learning process acquired 9840 images, with 9120 images used for training and 720 images for testing. The network was compared with two pre-trained CNN, LeNet and AlexNet. The results of similarity and F1-measure showed that DedistractedNet preceded both LeNet and AlexNet in all categories.

Based on previous work and findings by [14] that detected the criminal behavior using CNN as motivation in this work, we aim to investigate the forensic postures on anomalous human behavior at the gate of residential units as the database. Next is to develop four samplings of CNN with different architectures operated on a humble hardware platform.

III. Architecture of Convolution Neural Network

Hubel and Wiesel found that the neuron cells in the visual cortex of the cats were able to produce visual perceptions through self-organizing the image structure by learning from the experiences [18]. These cells are sensitive to a specific visual field region that is generally referred to as perceptive fields [19]. The specific tasks of the neuron cells in the visual cortex have been considered the genesis behind CNN's invention. Applying the same idea, let $F \in \text{¡}^{m_H \times n_W}$ and $W^k \in \mathbb{R}^{m \times n_{\ell\text{-}1}}$ be a matrix representing the perceptive field and neuron cell or filter size and weights in machine learning, respectively. In this study, the input type for CNN is an order 3 tensor, $X \in \mathbb{R}^{n_H \times n_W \times n_C}$ represents an image with H rows of neuron, W columns of neuron and C of color channels.

A. Convolution Layer

The convolution layers are characterized by an input map, X, a bank of independent filters or kernel, F, and biases, b. Each filter is convolved individually with the input map to produce a feature map, ϕ. indicated the relationship between the input and output of the network. Y can be written as, $Y = \varphi(WX + b)$ where X is the inputs, Y is the output, W is the weights, b is the biases, and φ is the activation parameters.

The feature map, ϕ is defined as the relationship between the input image $X \in \mathbb{R}^{n_H \times n_W \times n_C}$ and the filter or kernel, $F \in \mathbb{R}^{n_H \times n_W}$.

$$\phi = \mathrm{X}^{\otimes} \mathrm{F},\ \forall W \in F \tag{1}$$

Let the convolution of input feature maps, $X \in \mathbb{R}^{n_H \times n_W \times n_C}$ with a bank of D multi-dimensional filters, $F \in \mathbb{R}^{n_H \times n_W \times n_C \times D}$ and biases, $b \in \mathbb{R}^D$ one for each filter. The output from the convolving process of the input X by transposing the filter to implement data interpolation is given by the convolution theorem [20][21].

$$\left(x \cdot w\right)_{i\,j} = \sum_{m=0}^{H\text{-}1} \sum_{n=0}^{W\text{-}1} \sum_{c=1}^{C} w_{m,n,c} \cdot x_{i+m,\, j+n,c} + b \tag{2}$$

In essence, the convolution layer is a hierarchical model that comprises multiple convolution layers to train massive data in achieving the highest accuracy in detection and recognition. As presented in Fig. 2, there are two stages in the learning procedure of the convolution layer namely forward and backward propagation or also known as back propagation.

Forward propagation procedure calculates the output z, using the inputs, x. Meanwhile, the back propagation procedure takes the gradient of the loss function concerning output, ∇L_z as the input of the network and the gradients of x for the loss function, ∇L_x to implement the updating procedure of weights through convolution layers. The back propagation algorithm utilizes the effectiveness of the chain rule in handling the derivatives recursively to obtain the desired weights. Understanding that the network's input is a single vector, and the convolution layer consists of a set of neurons, each vector will convolve with each neuron during the learning process on the convolution layer. It is appropriate to understand the dimensional vector in each layer of the network in building the CNN.

There are two weaknesses during the convolution procedure. Firstly, it shrinks the image, and secondly, it discards great numbers of information near the edge of the image. Hyperparameters are introduced to solve these problems, first is f as the filter size that generally in odd size to attain symmetrical padding. Next is p, the padding by adding columns and rows of zero to preserve the spatial sizes of feature maps, and finally parameter s, known as the stride, the number of pixels that move when traversing the input during convolution. The convolution layer has various feature map sizes as the hyper parameters can modify it. Generally, feature maps and output volume can be expressed as, $\phi\,(z) \in \mathbb{R}^{n_H \times n_W}$ and $z \in \mathbb{R}^{n_H \times n_W \times n_C}$ respectively.

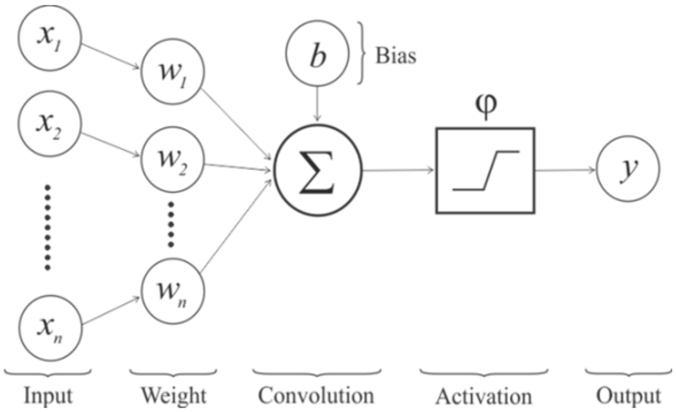


Fig. 1. The Architecture of Neural Network.

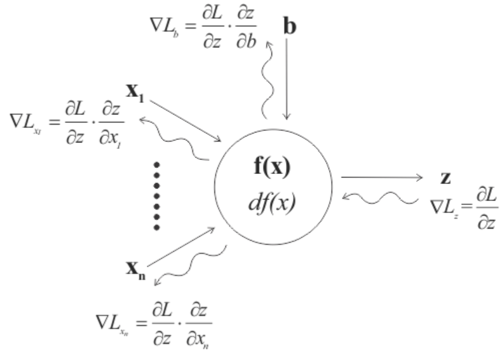


Fig. 2. Change in Vector during One Iteration of forward and backward Propagation.

B. Optimization Function

Calculus helps the learning process in machine learning to improve prediction accuracy by calculating the derivatives during the optimization procedure. The most common optimization function in CNN is gradient descent. The hallmark of gradient descent is required only the first order of derivatives of parameters concerning the loss function. The lower error value in the loss function demonstrated that better predictions had been calculated for the network. The Stochastic Gradient Descent (SGD) trains the learning algorithm to minimize errors, the calculation of the slope of error towards the negative of the gradient to find the global minima of the network. However, the downside of SGD is it complicates the convergence to potentially better global minima because the error rates keep overshooting due to the frequent updates. It results in SGD being computationally expensive and highly ineffective for memory, making SGDM a popular choice in the learning process of CNN. The velocity and friction parameters, *β*, applied in SGDM can prevent overshooting while allowing faster convergence. Adding the SGDM parameters into the gradient of the loss function to weight ∇L_w allows updating the consequences in the network. Furthermore, the parameters can steer the gradient vectors to accelerate in the right direction with the knowledge of the previous surface curve in the ravine [22]–[24]. In this study, the friction, *β*, is set to 0.9.

C. Activation Function

The primary purpose of the activation layer is to convert the input map of each neuron to the output feature map, which will then be used as the input map in the next layer. Essentially, the non-linear activation function is differentiable. Therefore, it allows the back propagation optimization technique to reduce errors by optimizing the weights using gradient descent [25]. Additionally, the function enables the neurons to learn the complex functional mapping from input data due to its curvature quality, considering the function has more than one degree. Traditionally, the sigmoid and hyperbolic tangent has been broadly used for the neural network but become irrelevant for many layers networks due to the vanishing gradient problem and slow convergence [25], [26]. The most appropriate activation function for CNN is rectified linear unit (ReLU) [27]. The advantages of ReLU are sparsely activated, that offers better predictive power and lessen over fitting to the training set, faster converging, avoids vanishing gradient problem, and the best attribute is computational economical as it excludes complicated mathematic functions. The ReLU mathematical expression can be written as [25][28];

Let $\varphi_k : \mathbb{R} \rightarrow \mathbb{R}$ be a non-linear activation function.

$$\varphi_k = max\begin{cases} 0 & for \quad x < 0 \\ x & for \quad x \geq 0 \end{cases} \left| \begin{matrix} \infty \\ 0 \end{matrix} \right. \tag{3}$$

D. Pooling Layer

The pooling layer role reduces the spatial dimension of input volume for the next layer. Therefore, the hyper parameter of the pooling layer is the stride only. Padding is rarely applied in this layer. The depths of this layer also remain the same, $n_C^{[\ell]} = n_C^{[\ell\text{-}1]}$. The feature map of the output can be calculated as follows,

$$\phi\,(z) \;=\; \left(\frac{n_H^{[\ell-1]} - f_H^{[\ell]}}{s}\right) \times \left(\frac{n_W^{[\ell-1]} - f_W^{[\ell]}}{s}\right) \tag{4}$$

E. Fully-Connected Layer

The fully-connected layer requires a vector as the input, therefore the flattening procedure is performed towards n-dimensional vectors where *n* >1. The flattening procedure is transforming spatial structure data into one dimensional feature vectors by concatenating the tridimensional tensor of convolution layer output into the monodimensional tensor that is a vector. The vectors are learned using gradient descent to ensure the class scores are accordant with the labels in the learning set of each image.

F. Classification Layer

The properties of the classification layer are the softmax activation function and cross-entropy loss function. The softmax activation function is generally applied to the final layer of the networks. It calculates a probabilistic value for every class between 0 and 1. The cross-entropy loss function measures the optimization for multi-class classification.

The softmax activation function has excellent features. Firstly the normalized data increases consistency and convenience in mapping. Next, it is differentiable and practicable in calculating the loss function. Finally, it employs the natural exponential that owns the ability to identify the difference between higher and lower values. In CNN, the softmax activation function takes the actual values of a *K*-dimensional vector on both input and output vectors, transforms it to a range of between 0 and 1, which is essential for the probability distribution. The softmax activation function at the output layer of the network can be defined as;

$$\sigma_j \;=\; \frac{e^{x_j}}{\sum\limits_{k=1}^{K} e^{x_k}}, \quad \forall j \in 1 \ldots K \tag{5}$$

Hence, the vector will be trained using cross-entropy loss function, ζ, to predict which input vector belongs to one of the output classes.

$$\zeta\left(x,y\right)_j = \sum_{j=1}^{K} x_j \;\ell n\, y_j \tag{6}$$

IV. Proposed CNN Samplings

Recall that this study proposes to detect anomalous behavior at the gate of residential units using from-scratch CNNs on a standard notebook as the hardware platform. The idea of having an extensive network was to gain a more robust network with large training datasets to satisfy the nonlinear algorithms requirement. This improves the network's skill and avoids overfitting, making the CNN computationally expensive to solely operate on GPU. Four types of CNN sampling with nearly twenty thousand images as training datasets have been utilized to defy the odds.

The CNN module named sampling refers to the kernel's change in size and depth or filter hyperparameter. However, the other hyperparameters of the convolution layer are set to a fixed value. The four modules are Up sampling, Down sampling, Up-Down sampling, and Down-Up sampling. There are three sizes of the filter being employed for the experiments as in Table I.

TABLE I. Value of Hyperparameters for Convolution Layers

Layers	**Padding, *p***	**Stride, *s***	**Filter Size, *f***	**No. of filter, n_c**
Conv1	1	1	3 x 3	16
Conv1	1	1	7 x 7	32
Conv1	1	1	11 x 11	64

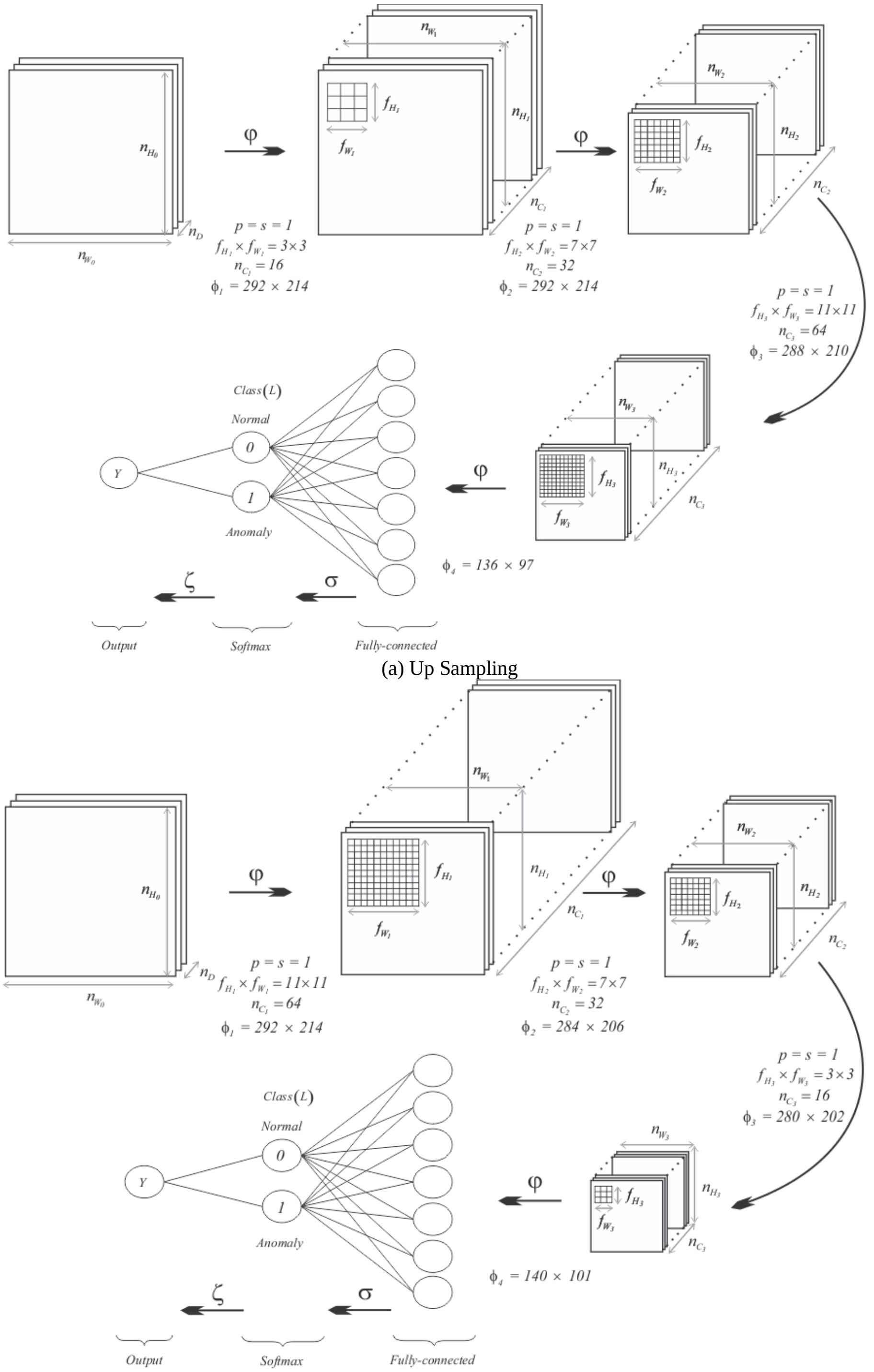
It is necessary to set the variables or hyperparameters of the training algorithm before executing each module as listed in Table II. Towards comparing each module, all hyperparameters values of the training procedure have been set to the same value. All sampling types consist of a convolution layer that alternates with the ReLU layer, the pooling layer and the networks end with the fully-connected layer followed by the softmax layer. The Up sampling is arranged in the ascending order of convolution layers namely Conv1, Conv2 and Conv3. Meanwhile, the Down sampling is organized in descending order specifically Conv3, Conv2, Conv1. As for Up-Down sampling, it starts with an ascending order and followed by descending order of the convolution layer that are Conv1, Conv2, Conv3, Conv3, Conv2, and Conv1 and the Down-Up sampling is in reverse designed with descending order at the beginning and continued by ascending order of convolution layer namely Conv3, Conv2, Conv1, Conv1, Conv2 and Conv3. In this study, the hyperparameter padding and stride of the convolution layer are set to 1 as in Table I. However, the hyperparameter stride of pooling layer can be varied. The architecture of CNN for each sampling is shown in Fig. 3. Next, Table III presents the detailed network configuration information on each developed sampling type in this study.

TABLE II. Hyperparameter for CNN Samplings

Hyper-parameter	**Size**	**Task**
Learning type	SGDM 0.9	• To prevent oscillations • To navigate the gradients towards optimum global minima
Learning rate	0.001	• To determine the speed of updating the trainable parameters • To assist in constant converging
Epoch	1000	• To start network propagates in both forward and backward • To activate the neuron • To calculate the loss • To obtain the partial derivative of the loss function • To update trainable parameters
Minibatch size	20	• To achieve training stability • To improve performance
Validation frequency	50	• To validate the network at regular interval

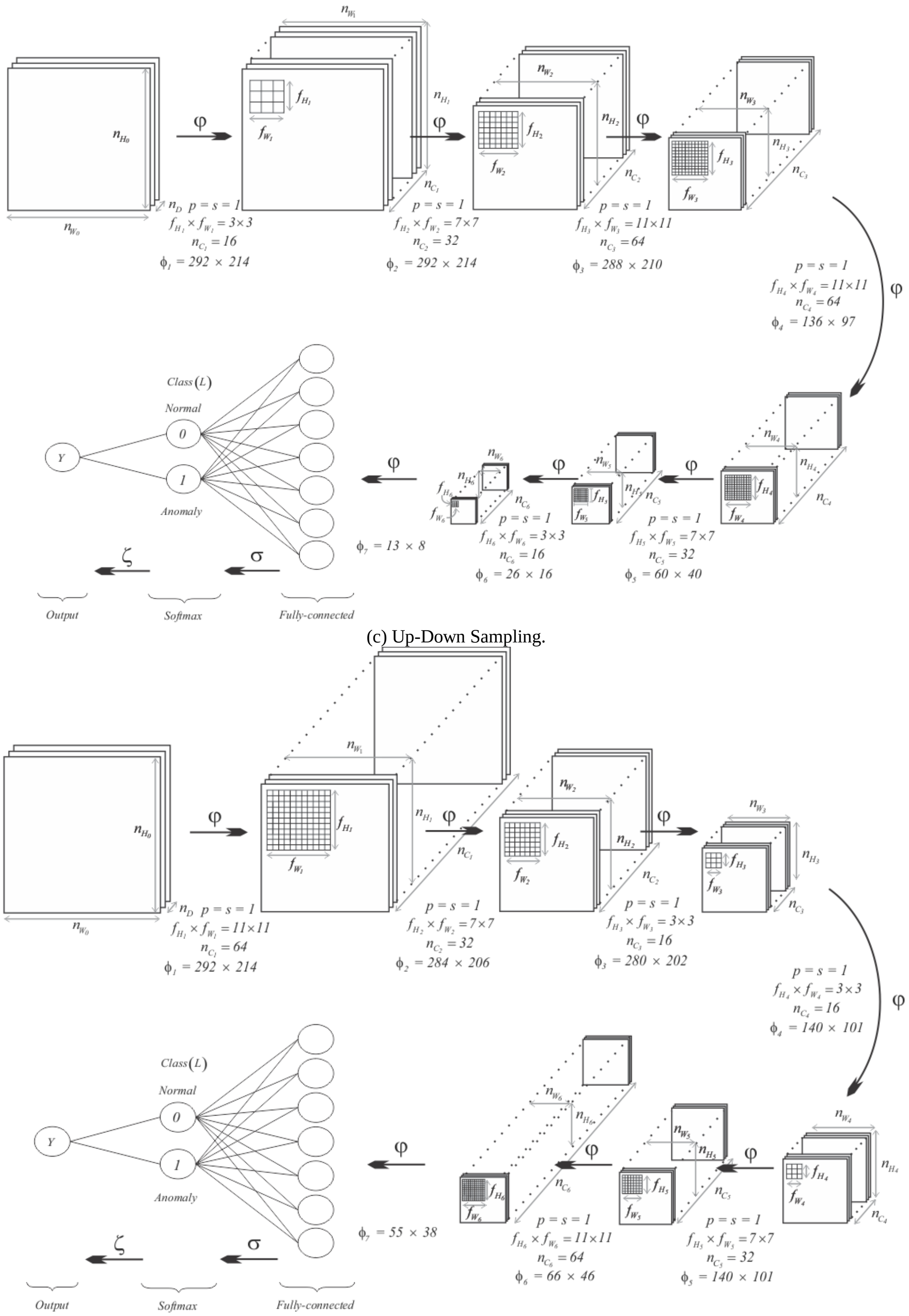
TABLE III. Network Configuration of all Samplings

Layers	**Up**	**Down**	**Up-Down**	**Down-Up**
Total layers	15	15	27	27
Image size	292 x 214 x 3			
Convolution 1	3 x 3 x 16	11 x 11 x 64	3 x 3 x 16	11 x 11 x 64
Pooling 1	1 x 1 average pool			
Convolution 2	7 x 7 x 32	7 x 7 x 32	7 x 7 x 32	7 x 7 x 32
Pooling 2	2 x 2 max pool			
Convolution 3	11 x 11 x 64	3 x 3 x 16	11 x 11 x 64	3 x 3 x 16
Classification	2 fully-connected, softmax			
Pooling 3			2 x 2 max pool	1 x 1 average pool
Convolution 4			11 x 11 x 64	3 x 3 x 16
Pooling 4			2 x 2 max pool	
Convolution 5			7 x 7 x 32	7 x 7 x 32
Pooling 5			2 x 2 max pool	1 x 1 average pool
Convolution 6			3 x 3 x 16	11 x 11 x 64
Classification			2 fully-connected, softmax	



(a) Up Sampling

(b) Down Sampling.



(c) Up-Down Sampling.

(d) Down-Up Sampling.

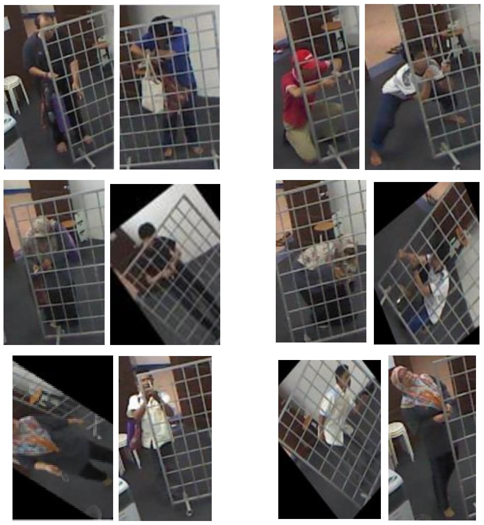
Fig. 3. The Architecture of the Developed CNN Samplings with Hyperparamters and Feature Maps of each Layer.

V. Experimental Results and Discussion

In this study, forensic postures are used, and are referred to as the postures defined by the Royal Malaysia Police (RMP). The definition is interpreted from the Criminal Procedure Code practiced by RMP. The authorities agreed that the observations from their experience are relatively in line with the Malaysian Penal Code. The four most frequent postures recognized during housebreaking crime are squatting, bending, squatting with heels up, squatting with heels down, and kneeling with heels up.

By complying with the requirement of forensic postures, input images that comprised of 9558 color images for each class representing normal and anomaly are collected during data acquisition. Some of the images collected from the footage of participants acted as criminals or otherwise are as in Fig. 4. For each category, 7000 images are randomly selected as the training images whilst the remainder as the testing images.

During experimental, AlexNet and VGG-16 were used in classifying the housebreaking crime postures, but the intention to leverage the architecture of VGG-16 was not achieved since the hardware platform was insufficient to meet the requirement; enormous memory due to the computational cost of 138 million parameters. Thus, four samplings were developed, suitable for low memory platforms to train and test a total of 19116 images together with the AlexNet. The effectiveness of all networks is investigated under two methods. First is the offline test using CCTV videos of housebreaking crime in Malaysia as inputs. Next is the real-time testing using the live feed from a webcam that replicates CCTV as inputs.



(a) (b)

Fig. 4. Input Images, (a) Normal Behaviors, (b) Anomalous Behaviors.

A. Performance of Trained Networks

Referring to Fig. 5, AlexNet with the up-down architecture of convolution layers has the highest training parameters, producing the highest ability to recognize normal behaviors with 99.27% specificity. The developed CNN, Up sampling is the best network for identifying anomalous behaviors according to the sensitivity percentages of 97.26%. Networks with higher training parameters namely AlexNet, Up sampling and Up-Down sampling achieved recognition rate of 97% to 99% in identifying normal and anomalous behavior. These results prove that the number of training parameters contributes to the performance of networks. However, the architectures and hyper parameters are more dominant considering the classification results of all developed samplings are comparable to AlexNet, which have more than 62 times higher computation costs. Here, results showed the ability of AlexNet to identify abnormal and normal behavior based on the sensitivity and specificity obtained.

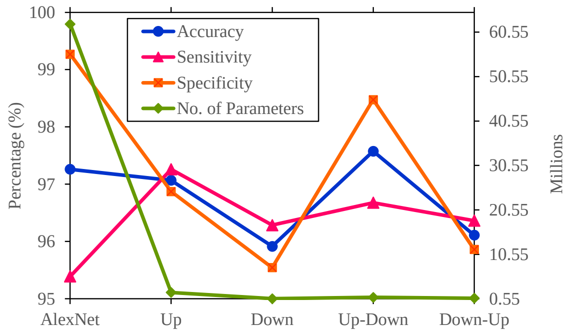


Fig. 5. Performance of Networks in Classifying Normal and Anomalous Behavior.

B. Offline Test

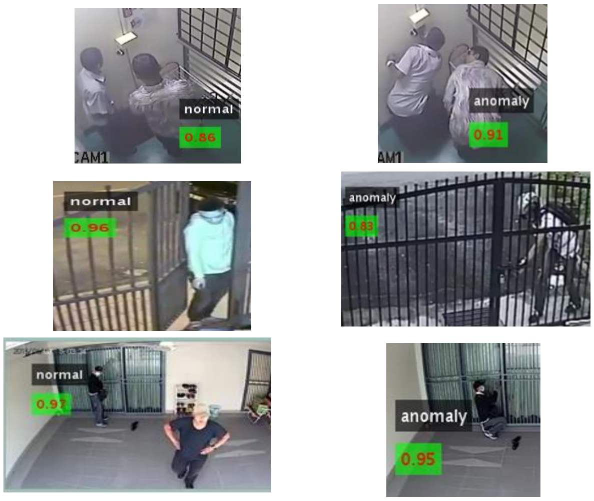
For offline testing, CCTV videos of housebreaking at the gate of residential units in Malaysia were used. The videos recorded single or multiple perpetrators while committing crimes, in broad daylight or at night. The duration of these videos is within 55 seconds to nearly three minutes for housebreaking without using tools. However, videos for housebreaking using tool have a longer video length of two to eight minutes. Each of the perpetrators clearly performs the abnormal characteristics as defined by RMP. Three networks, AlexNet, Up sampling and Down sampling, have successfully recognized both normal and anomalous behaviors from CCTV videos. All networks showed higher ability in identifying normal behaviors than anomalous behaviors following a higher percentage attained by specificity than sensitivity as in Table IV.

TABLE IV. Performance of Trained Networks on Offline Test

Network	Sensitivity (%)	Specificity (%)	Detection Skill
AlexNet	70 to 99	90 to 99	High
Up	80 to 95	80 to 90	Moderate
Down	70 to 80	80 to 90	Moderate
Up-Down	-	100	Failed
Down-Up	-	100	Failed

Detection skill for normal behavior is around 95% to 100% due to the predictable, regular routines at the gate. However, a detection range of 75% to 100% is forecasted for abnormal behavior resulting from the perpetrator's complexity and unpredictable behavior (s). Thus, AlexNet is categorized as 'High' for Detection Skill because it has demonstrated the abilities in detecting as presumed whilst Up Sampling and

Down Sampling as 'Moderate' because lower percentages were recorded for normal and anomalous behavior than the desired results during the classification process. However, moderate detection skill achieved by Down sampling was impressive given its previous performance at the lowest place. Up-Down and Down-Up sampling are categorized as ‘Failed’ based on the testing from single behavior detection through all videos. Fig. 6 depicts the results during offline test that indicated high accuracy of detection from AlexNet and Up sampling.



(a) (b)

Fig. 6. Offline Detection towards Behaviors at the Gate, (a) Normal Behavior, (b) Anomalous Behavior.

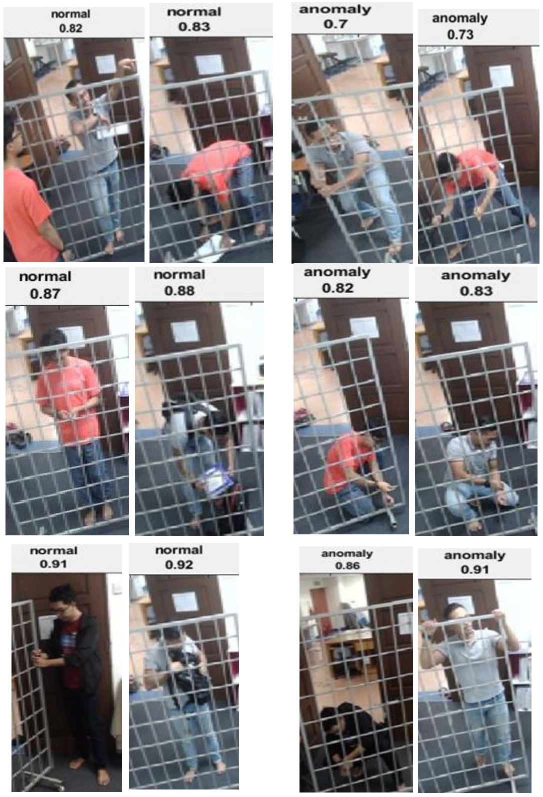
C. Real-Time Test

For the real-time test, live feed images from a webcam were used by the networks to detect within 40 milliseconds to 0.2 seconds. The trial was held in the laboratory and conducted in various situations such as different acts according to the type of gates, for instance, slide gate or push gate, sneaking or lurking, breaking locked gate using a tool for both normal and anomaly behaviors. Participants were required to behave normally at the gate, such as unlocking the padlock or latch and picking up object(s) on the ground, according to their normal habits for routine behavior detection. During anomalous behaviors detection, participants were further requested to impersonate housebreaking crime at the gate according to their interpretation, perspective, and evaluation.

Results showed that AlexNet and Up sampling successfully detecting normal and anomalous behaviors during the real-time test up to 90% recognition rate, as in Fig. 7. Refer to Table V detection achieved by AlexNet is again categorized as ‘High’ whilst Up Sampling as ‘Moderate’. However, for real-time scenarios, Down Sampling is categorized as ‘Failed” along with Up-Down and Down-Up sampling since the live feed images are detected as normal throughout the test. Similar results were identified throughout the real-time test, with squatting being highly identified as an anomaly, standing as normal and bending and kneeling were identified more as an anomaly than normal. All the normal activities at the gate were recorded less time duration than anomalous activities, including situations requiring the participants to take out keys from the backpack while holding bags using the other hand.

TABLE V. PERFORMANCE OF TRAINED NETWORKS ON REAL-TIME TEST

Network	Sensitivity (%)	Specificity (%)	Detection Skill
AlexNet	70 to 90	80 to 92	High
Up	70 to 85	85 to 90	Moderate
Down	-	100	Failed
Up-Down	-	100	Failed
Down-Up	-	100	Failed



(a) (b)

Fig. 7. Real-time Detection towards Behaviors at the Gate, (a) Normal Behavior, (b) Anomalous Behavior.

VI. CONCLUSION

In conclusion, the experimental results showed that the architectures, number of parameters, and precisely choosing the hyperparameters are the keys to developing a robust network. It is essential to understand the CNN theoretical and mathematical concepts to develop optimum architecture. The developed samplings proposed in this study are suitable for modest hardware platforms but yielded up to 97% accuracy and succeeded an offline and real-time tests with 97% and 90% recognition rates, respectively. In addition, all samplings were trained using single-subject images for both normal and anomalous behaviors. Yet, the AlexNet and Up sampling could recognize normal and anomalous behaviors for more than one subject and successfully distinguish the anomalous behavior of one person from a group of normal subjects during both offline and real-time tests. The performance of Up sampling has been proven to be on par with the renowned CNN, AlexNet and even more attractive, the computational cost of Up sampling is almost 62 times cheaper.

Finally, CNN has proven its ability in detecting and classify humans based on criminal gait or otherwise. Future work includes developing a classification method to increase the performance and the employment of Faster-RCNN that can enhance its strength in behavior detection. The next stage of work will also explore anomalous human behavior in other potential crime environments, such as banks and high-security areas at airports, warehouses, parking vehicle areas, and car robbery. These initial findings could lead to the development of forensic intelligent surveillance systems that can further help the authorities to decrease the criminal cases rate.

Acknowledgment

This research is funded by the Ministry of Higher Education (MOHE) Malaysia via the Fundamental Research Grant Scheme (FRGS) No: FRGS/1/2019/TK04/UITM/01/3). We would like to thank the Royal Malaysia Police for providing legal information and assisting in developing the forensic gait features and the College of Engineering, UiTM, Selangor, Malaysia, for the facilities and support given in this research.

References

[1] H. Iwama, D. Muramatsu, Y. Makihara, and Y. Yagi, “Gait Verification System for Criminal Investigation,” IPSJ Trans. Comput. Vis. Appl., vol. 5, pp. 163–175, 2013, doi: 10.2197/ipsjtcva.5.163.

[2] M. Ben Ayed and M. Abid, “Suspicious behavior detection based on DECOC classifier,” 18th Int. Conf. Sci. Tech. Autom. Control Comput. Eng., pp. 594–598, 2017.

[3] L. He, D. Wang, and H. Wang, “Human abnormal action identification method in different scenarios,” Proc. 2011 2nd Int. Conf. Digit. Manuf. Autom. ICDMA 2011, pp. 594–597, 2011, doi: 10.1109/ICDMA.2011.148.

[4] W. Lawson and L. Hiatt, “Detecting Anomalous Objects on Mobile Platforms,” 2016 IEEE Conf. Comput. Vis. Pattern Recognit. Work., pp. 1426–1433, 2016, doi: 10.1109/CVPRW.2016.179.

[5] N. C. Tay, C. Tee, T. S. Ong, K. O. M. Goh, and P. S. Teh, “A Robust Abnormal Behavior Detection Method Using Convolutional Neural Network,” in Computational Science and Technology. Fifth International Conference on Computational Science and Technology. Lecture Notes in Electrical Engineering, vol. 481, R. Alfred, Y. Lim, A. Ibrahim, and P. Anthony, Eds. Singapore: Springer Nature, 2019, pp. 37–47.

[6] A. Al-Dhamari, R. Sudirman, and N. H. Mahmood, “Abnormal Behavior Detection In Automated Surveillance Videos : A Review,” J. Theor. Appl. Inf. Technol., vol. 95, no. 19, pp. 5245–5263, 2017.

[7] B. Delgado, K. Tahboub, and E. J. Delp, “Automatic Detection Of Abnormal Human Events On Train Platforms,” IEEE Natl. Aerosp. Electron. Conf. (NAECON 2014), pp. 169–173, 2014.

[8] M. Alotaibi and A. Mahmood, “Improved Gait Recognition based on Specialized Deep Convolutional Neural Networks,” 2015 IEEE Appl. Imag. Pattern Recognit. Work., pp. 1–7, 2015.

[9] J. M. Shrein, “Fingerprint Classification Using Convolutional Neural Networks and Ridge Orientation Images,” IEEE Symp. Ser. Comput. Intell., pp. 1–8, 2017.

[10] W. Zhang and C. Wang, “Application of Convolution Neural Network in Iris Recognition Technology,” 2017 4th Int. Conf. Syst. Informatics (ICSAI 2017), pp. 1169–1174, 2017.

[11] S. Dara and P. Tumma, “Feature Extraction By Using Deep Learning: A Survey,” 2018 Second Int. Conf. Electron. Commun. Aerosp. Technol., pp. 1795–1801, 2018, [Online]. Available: https://acadpubl.eu/hub/2018-120-6/1/20.pdf.

[12] F. T. George, V. S. P. Patnam, and K. George, “Real-time deep learning based system to detect suspicious non-verbal gestures,” I2MTC 2018 - 2018 IEEE Int. Instrum. Meas. Technol. Conf. Discov. New Horizons Instrum. Meas. Proc., pp. 1–6, 2018, doi: 10.1109/I2MTC.2018.8409864.

[13] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 10, pp. 1–20, 2016.

[14] H. Xu, L. Li, M. Fang, and F. Zhang, “Movement Human Actions Recognition Based on Machine Learning,” Int. J. Online Biomed. Eng., vol. 14, no. 4, pp. 193–210, 2018.

[15] L. Zhang and P. N. Suganthan, “Visual Tracking with Convolutional Neural Network,” 2015 IEEE Int. Conf. Syst. Man, Cybern., pp. 1–6, 2015.

[16] K. Makantasis, A. Doulamis, N. Doulamis, and K. Psychas, “Deep Learning Based Human Behavior Recognition in Industrial Workflows,” 2016 IEEE Int. Conf. Image Process., pp. 1–5, 2016.

[17] Y. Pang, S. Syu, Y. Huang, and B. Chen, “An Advanced Deep Framework for Recognition of Distracted Driving Behaviors,” 2018 IEEE 7th Glob. Conf. Consum. Electron., pp. 802–803, 2018.

[18] D. H. Hubel and T. N. Wiesel, “Receptive Fields, Binocular Interaction and Functional Architecture in the Cat’s Visual Cortex,” J. Physiol., vol. 160, no. 1, pp. 106–154, 1962.

[19] D. H. Hubel and T. N. Wiesel, “Receptive Fields and Functional Architecture of Monkey Striate Cortex,” J. Physiol., vol. 195, no. 1, pp. 215–243, 1968.

[20] M. Mathieu, M. Henaff, and Y. Lecun, “Fast Training of Convolutional Networks through FFTs,” arXiv:1312.5851v5 [cs.CV], pp. 1–9, 2014.

[21] A. Lavin and S. Gray, “Fast Algorithms for Convolutional Neural Networks,” IEEE Conf. Comput. Vis. Pattern Recognit., pp. 4013–4021, 2016.

[22] K. Ochiai, N. Toda, and S. Usui, “New Accelerated Learning Algorithm to Reduce the Oscillation of Weights in Multilayered Neural Networks,” Int. Jt. Conf. Neural Networks, vol. 1, pp. 914–919, 1992.

[23] S. K. Lenka and A. G. Mohapatra, “Gradient Descent with Momentum Based Neural Network Pattern Classification for the Prediction of Soil Moisture Content in Precision Agriculture,” 2015 IEEE Int. Symp. Nanoelectron. Inf. Syst., pp. 63–66, 2015, doi: 10.1109/iNIS.2015.56.

[24] S. Ruder, “An Overview of Gradient Descent Optimization Algorithms,” arXiv:1609.04747v2 [cs.LG], pp. 1–14, 2016.

[25] R. Zaheer, “GPU-based Empirical Evaluation of Activation Functions in Convolutional Neural Networks,” 2018 2nd Int. Conf. Inven. Syst. Control (ICISC 2018), pp. 769–773, 2018.

[26] M. Wang, “Look-up Table Unit Activation Function for Deep Convolutional Neural Networks,” 2018 IEEE Winter Conf. Appl. Comput. Vis., pp. 1225–1233, 2018.

[27] A. A. M. Al-saffar, H. Tao, and M. A. Talab, “Review of Deep Convolution Neural Network in Image Classification,” 2017 Int. Conf. Radar, Antenna, Microwave, Electron. Telecommun., pp. 26–31, 2017.

[28] The MathWork, “Softmax layer - MATLAB,” 2016. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.softmaxlayer.html. [Accessed: 20-May-2019].