Convolutional Neural Network and Deep Residual Network

Pedestrian Classification

Abstract— This paper presents a pedestrian classification methods based on the use of a convolutional neural network (CNN) and deep residual network (DRN). Dataset contains 81592 infrared pedestrian and non-pedestrian images. As a preprocessing, dataset are scaled between 0-255 and converted as data type uint8 for each method. For feature extraction and classification parts, convolutional neural network and deep residual network are used. DRN has higher accuracy than CNN. Accuracies of classification CNN and DRN are 89% and 95%, respectively. Finally observations and future work plans are done in conclusion part.

I. INTRODUCTION

All over the world, a lot of people are died due to lack of attention in many areas. In daily lives, people cannot recognize other people in the traffic or a dangerous factory. This situation cause fatal accidents. On the other hand, people cannot be found in case of emergency such as earthquakes, floods or grounding. Even if high resolution cameras are proposed to solve these types of circumstances, cameras do not detect and classify very well in the dark. Therefore, thermal cameras can be used for human detection and classification because of human body radiation. As a solution, deep learning algorithms such as Convolutional Neural Network (CNN) and Deep Residual Network(DRN) have led to a series of breakthroughs for image classification. In order to understand solution better let's look at figure 1. Thermal camera detect pedestrian, then this image is tried to classify with CNN and DRN. After classification process, the result is given whether it is pedestrian or non-pedestrian. In the following sections, information about dataset, CNN and DRN methods, performance of methods and experiment results will be explained, respectively.

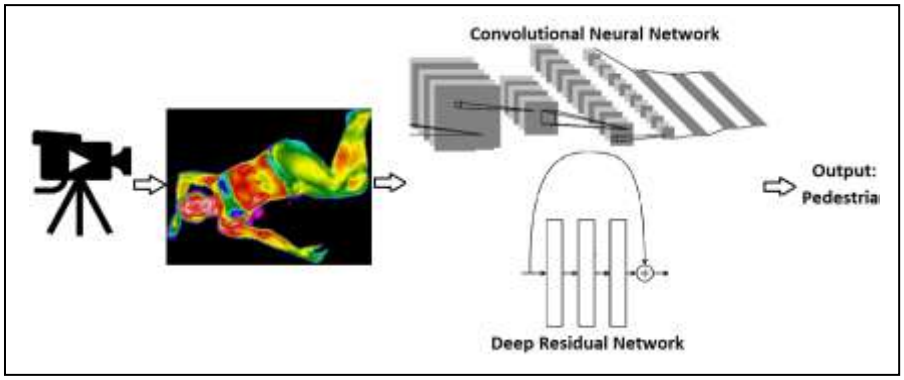

Figure 1. Block scheme of problem

II. DATASET

The proposed Convolutional Neural Network and Deep Residual Network algorithms are evaluated with LSI Far Infrared Pedestrian Dataset that consists of FIR images collected from a vehicle driven in outdoors urban scenarios [2]. Images were acquired with an Indigo Omega imager, with a resolution of 164x129 pixels, a grey-level scale of 14 bits, and focal length of 318 pixels. The camera is put on vehicle. Recorded images are labeled manually and are taken in a box. In order to prevent bias introduced by border artifacts their height is subsequently up scaled by 5%. In the dataset, there are positives and randomly sampled negatives with a fixed height-width ratio of (1/2) and rescaled to 64x32 pixels. The Database is divided in two groups which are train and a test sets. The train set includes 10208 positives which include pedestrians and 43390 negatives which does not include pedestrians. On the other hand, the test set includes 5944 positives and 22050 negatives. 65% of the dataset is train set and 35% of the dataset is test set. Some samples from dataset can be seen in Figure 2. The algorithms are implemented on windows using GTX-1070 GPU and GPU-based Pytorch platform [5].

Figure 2. Example of samples from positive and negative images

III. METHODS

In this section, brief instruction is given for Convolutional Neural Network and Deep Residual Network.

A. Convolutional Neural Network (CNN)

In this paper, Convolutional Neural network is used to solve binary classification problem. Convolutional Neural Networks [7] are a special variant of multilayer perceptrons (MLP) where the first layers that are called convolution layers are configured to act as a hierarchical feature extractor. The used structure of neural network can be seen in figure 2. Convolutional neural network is structured as two main parts that are feature extraction and classification. In feature extraction part, there are 2 convolutions and 1 pooling layers. In classification part, there is fully connected layer. The details of CNN will be explained in remaining sections.

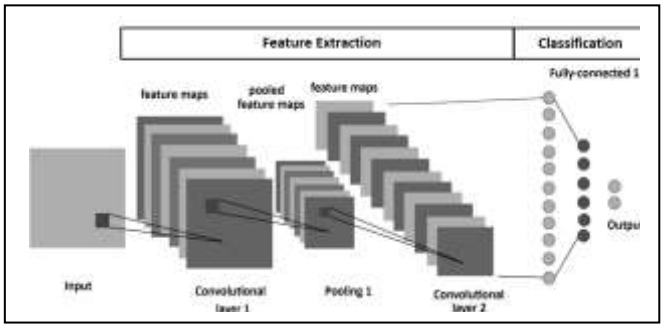

Figure 3. Structure of Convolutional Neural Network

Table 1: Layers of the Convolutional Neural Network

Network Name	*Block Name*	*Layer Name*	*Number Of Kernels*	*Kernel Size*	*Stride*	*Number Of Neurons*	*Neuron Type*
CNN	*C1*	*Conv1*	*10*	*5*			*ReLU*
		Pool1	*-*	*2*	*-*		
	C2	*Conv2*	*16*	*5*			*ReLU*
	FC	*FC1*				*520*	*ReLU*
		FC2				*130*	*ReLU*
		FC3				*2*	*Linear*

1) Preprocessing

All images in Far Infrared Pedestrian Dataset are scaled between 0-255 and converted as data type uint8. Preprocessed data is used for both Convolutional Neural Network and Deep Residual Network.

2) Feature Extraction

In the proposed algorithm CNN feature extraction part is as follows. In Convolutional Neural Network method, feature extraction is done by using its Convolution kernels instead of using traditional feature extraction methods. Multiple learnable filters in the various Convolutional layers extract discriminative features, which will be are then fed to the fully connected networks in the deeper layers of the CNN at next section. In order to make feature extraction two blocks are used as it can be seen in figure 2. In the first block, there are convolutional and pooling layers. Number of kernels in first convolutional layer is 10 and kernel size is 5 x 5. The kernel size in first pooling layer is 2 x 2. On the other hand, second block has only a Convolutional layer that has 16 kernels. The size of the kernel in second convolutional layer is 5x5 as it happens in first convolutional layer. Also, Rectified Linear Unit (ReLU) is used as an activation function. Extracted features that are also called feature maps can be seen in figure 4. In pedestrian images, there is a density in the middle of the images. On the other hand, feature map of non-pedestrian images looks like a random noise.

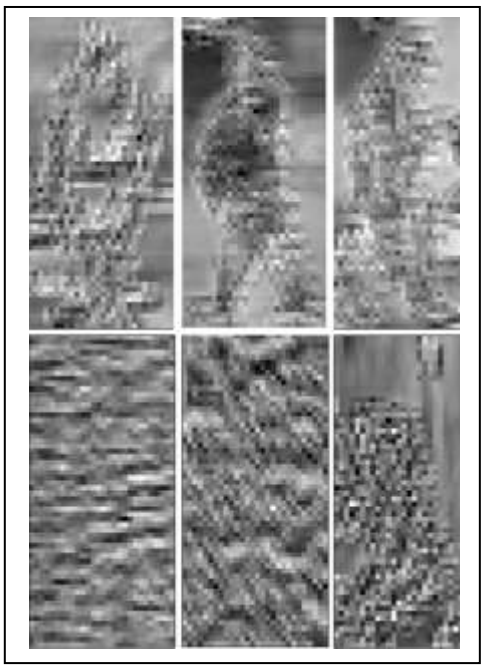

Figure 4. Feature Maps

3) Classification

In the proposed algorithm CNN classification part is as follows. The last block of the convolutional neural network includes the output neurons, which is used to classify images whether they include pedestrian or not. The last block is called fully connected (FC) block. In fully connected block, there are 3 layers. The first layer has 520 neurons with ReLU activation, second layer has 130 neurons with ReLU activation and the last layer has 2 neurons with linear activation. Using the class scores, each sample is classified as either pedestrian or non-pedestrian. CNN is trained with cross-entropy loss function and optimized with stochastic gradient descent (SGD). In SGD, fixed learning rate is 0.00001.

B. Deep Residual Network

Most important difference between Deep Residual Networks (DRN) and traditional Convolutional Neural Networks (CNN) is that DRNs provide a clear path for gradients to back propagate to early layers during training [3]. In CNN, H(x) is mapping some part of the CNN in figure 5. In DRN, H(x) is defined as F(x) + x where x is input of part of network. Therefore, values in previous layers can be identified more easily.

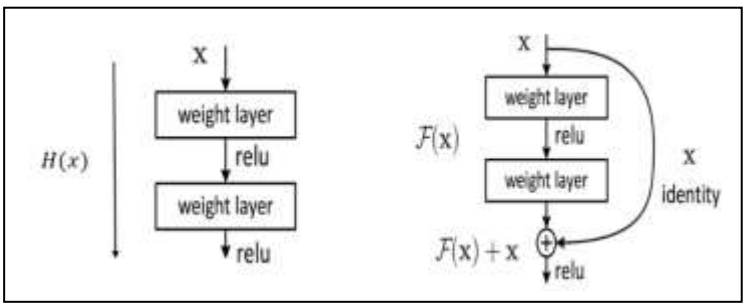

Figure 5. Left: Traditional CNN, Right: Residual Block

Deep Residual Network consists basic residual blocks in figure 5. Basic residual block architecture consists of series of layers which are convolution, batch normalization (BN), ReLU, Dropout, Convolution, BN, respectively. ReLU is applied at the end of the block after adding with identity function (X).

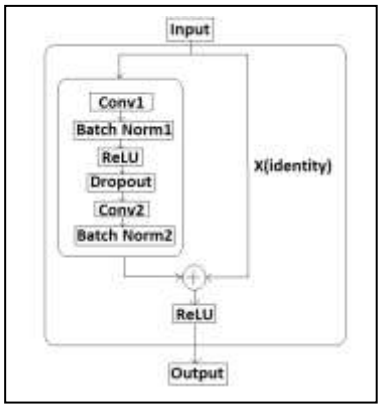

Figure 6. Basic Residual Block

Deep Residual Network architecture consists of series of layers which are convolution, BN, ReLU, Pooling, Block-1,2,3,4,5, pooling and fully connected, respectively as it can be seen in figure 7.

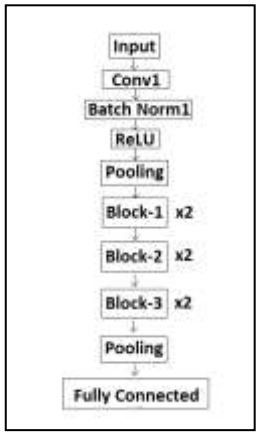

Figure 7. Deep Residual Network

Table 2: Layers of the Deep Residual Network

Network Name	*Block Name*	*Layer Name*	*Number Of Kernels*	*Kernel Size*	*Stride*	*Number Of Neurons*	*Neuron Type*
DRN	*C1*	*Conv1*	*64*	*7*	*2*		*ReLU*
		BN1	*64*	-	-		
		Pool1		*3*	*2*		
	Block1	*Conv1*	*64*	*3*	*1*		*ReLU*
		BN1	*64*				
		Drop out					
		Conv2	*64*				*ReLU*
		BN2	*64*				

Block2	*Conv1*	*128*	*3*	*1*		*ReLU*
	BN1	*128*				
	Drop out					
	Conv2	*128*				*ReLU*
	BN2	*128*				
Block3	*Conv1*	*256*	*3*	*1*		*ReLU*
	BN1	*256*				
	Drop out					
	Conv2	*256*				*ReLU*
	BN2	*256*				
FC	*Pool1*					
	FC1				*2*	*Linear*

1) Preprocessing

Preprocessing is the same with preprocessing in Convolutional Neural Network.

2) Feature Extraction

In the proposed algorithm DRN feature extraction part is like CNN. It means that there is no traditional feature extraction method. Convolutional layers are used for feature extraction in DRN. So as to make feature extraction two types of blocks are used as it can be seen in figure 6. First one is convolutional (conv1) layer. This convolutional layer 64 kernels and size of kernels is 7 x 7. Second one is in blocks. Each block has same architecture that includes 2 convolutional layers (conv1 and conv2). Even if architecture of blocks is same, number of kernels in blocks is different from each other. Block-1 has 2 convolutional layers and number of kernel is 64 for each convolutional layer. Block-2 has 2 convolutional layers and number of kernel is 128 for each convolutional layer. Block-3 has 2 convolutional layers and number of kernel is 256 for each convolutional layer. The size of the kernels is 3 x 3 in all convolutional layers in these blocks. On the other hand, activation function is ReLU for all convolution layers. After feature extraction one more pooling is applied and there will be classification part of DRN.

3) Classification

In the proposed algorithm DRN classification part is as follows. The last block of the convolutional neural network includes the output neurons, which is used to classify images whether they include pedestrian or not. The last block is called fully connected (FC) block. In fully connected block, there is 1 layer that has 2 neurons with linear activation. Using the class scores, each sample is classified as either pedestrian or non-pedestrian. DRN is trained with cross-entropy loss function and optimized with adaptive momentum (ADAM) In ADAM, learning rate is 0.0001.

IV. PERFORMANCE CRITERIA

Accuracy and process time is main metrics in order to evaluate performance of CNN and DRN. Both algorithms are run in NVIDIA GTX 1070 GPU. CNN has 100 epoch and process time of it is almost 0.5 hours. On the other hand, RDN has 100 epoch and process time of it is almost 1 hours.

Accuracy and confusion matrix are used as performance criteria for CNN and DRN. Confusion matrix has 4 parameters that are true positive, true negative, false positive and false negative.

Accuracy = (correct classified / number of total sample) x 100

V. EXPERIMENTAL RESULTS

Experimental result starts with parameter initialization. Weight and bias parameters are initialized according to formula in below [6].

$$\mathcal{U}(-\sqrt{k}, \sqrt{k}) \text{ where } k = \frac{1}{C_{\text{in}} * \prod_{i=0}^{1} \text{kernel\_size}[i]}$$

In the training of both CNN and DRN, cross entropy function is used as a loss function. For pedestrian dataset, there are 2 classes so cross entropy loss function can be accepted as binary entropy loss function. Cross entropy loss function and binary loss function are below, respectively. When M equals 2, first and second equation become equals [4].

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

$$-(y \log(p) + (1-y) \log(1-p))$$

In order to calculate gradients SGD is used for CNN and ADAM is used for DRN. Equations of SGD and ADAM optimizer are below, respectively [1].

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

In CNN method, the accuracy is 89% for test set and 88% for train set. In DRN method, the accuracy is 95% for test set and 98% for train set. Also test accuracy vs loss for DRN and CNN is in figure 8.

Table 3: Accuracy comparison of CNN and DRN

Accuracy	CNN	DRN
Train	88%	98%
Test	89%	95%

Table 4: Confusion Matrix of CNN and DRN

		CNN		DRN	
		Predicted Class			
		P	**Non-P**	**P**	**Non-P**
Actual Class	**P**	88%	12%	95%	5%
	Non-P	11%	89%	5%	95%

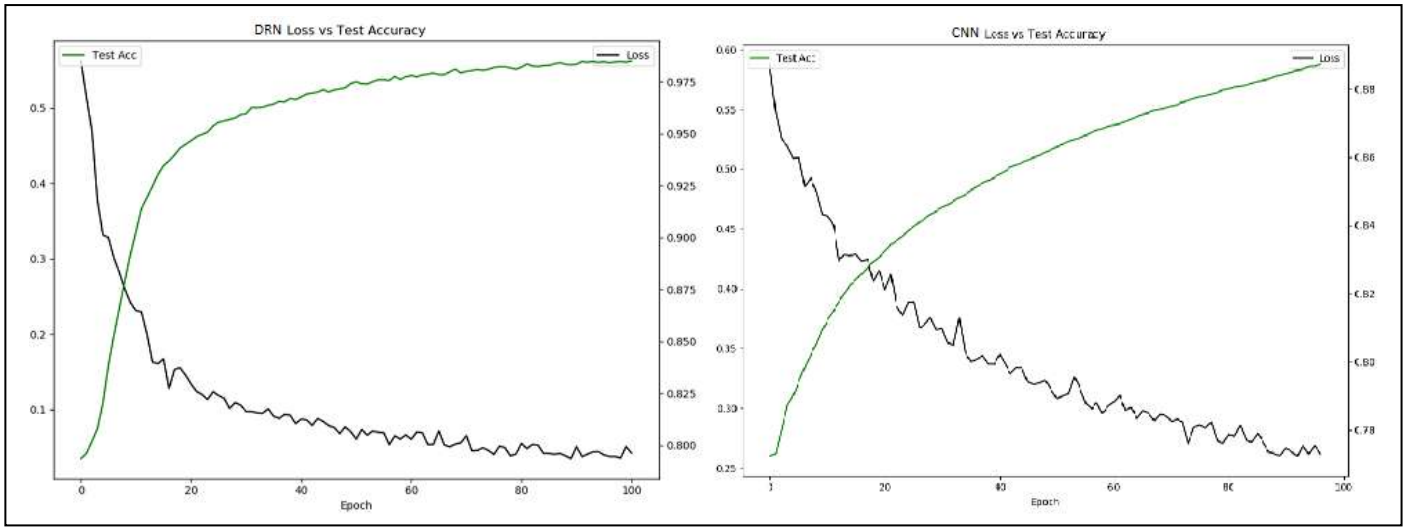

Figure 8. Deep Residual Network

VI. CONCLUSION

In this paper, CNN and DRN methods are applied and results are compared with each other for both train and test sets. Accuracies of train and test sets of each result are close to each other. It can be concluded that both methods are worked successfully and there is no over or under fitting. In order to increase the accuracy, deepness of methods can be increased as a feature work. However, it will also increase the process time as a disadvantage. Apart from these, in order to increase accuracy transfer learning and data augmentation methods can be tried in the future. Classification of pedestrian or human image will be very useful for Emergency situations and rescuing people.

REFERENCES

[1] Darken, C., Chang, J., & Moody, J. (1992). Learning rate schedules for faster stochastic gradient search. Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop, (September), 1–11.

[2] D. Olmeda, C. Premebida, U. Nunes, J.M. Armingol and A. de la Escalera. Pedestrian Classification and Detection in Far Infrared Images. Integrated Computer-Aided Engineering 20 (2013) 347–360

[3] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 770–778.

[4] Ml-cheatsheet.readthedocs.io. (2019). *Loss Functions — ML Cheatsheet documentation*. [online] Available at: https://ml cheatsheet.readthedocs.io/en/latest/loss_functions.html [Accessed 3 Jan. 2019].

[5] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam, "Automatic differentiation in PyTorch", NIPS-W, 2017

[6] Pytorch.org. (2019). *torch.nn — PyTorch master documentation*. [online] Available at: https://pytorch.org/docs/stable/nn.html#torch.nn.ConvTranspose1d [Accessed 3 Jan. 2019].

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

RELATED LINKS

Dataset:
http://portal.uc3m.es/portal/page/portal/dpto_ing_sistemas_automatica/investigacion/IntelligentSystemsLab/research/InfraredDataset