🌐 ***United International Journal of Engineering and Sciences*** 🌐
(UIJES – A Peer-Reviewed Journal); ISSN:2582-5887 | Impact Factor:8.075(SJIF)
📘 *Volume 5 | Special Issue 1 | 2025 Edition*
National Level Conference on "Advanced Trends in Engineering Science & Technology" – Organized by RKCE

TRAFFIC SIGN DETECTION AND RECOGNITION

T. Pavan Kumar[1], G. Gopinadh[2], K. Deepak[3], M. Chaitanya[4]

[1,2,3,4]UG Scholars, Dept. of Computer Science, RK College of Engineering

Vijayawada, India.

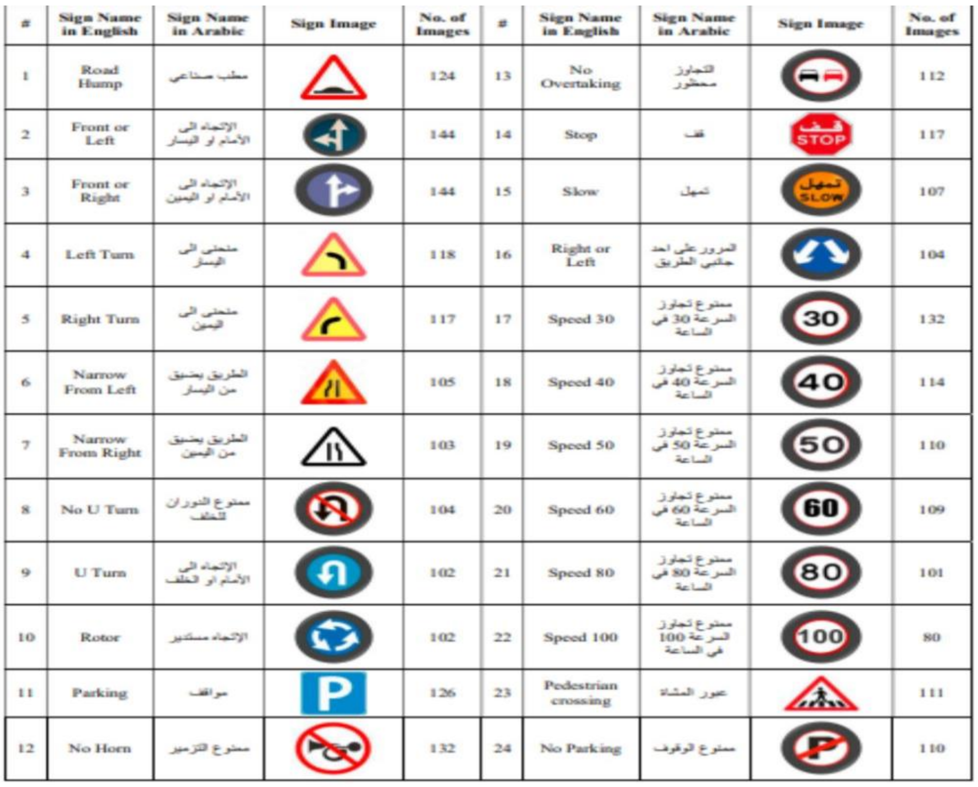
[1]tannirupavan1234@gmail.com , [2]gopiy1951@gmail.com [3]deepakkadime@gmail.com , [4]reddychaitanyamadhav@gmail.com

Abstract **– In today's world, almost everything we do has been simplified by automated tasks. In an attempt to focus on the road while driving, drivers often miss out on signs on the side of the road, which could be dangerous for them and for the people around them. This problem can be avoided if there were an efficient way to notify the driver without having them shift their focus. Traffic Sign Detection and Recognition (TSDR) plays an important role here by detecting and recognising a sign, thus notifying the driver of any upcoming signs. This not only ensures road safety, but also allows the driver to be a little more at ease while driving on tricky or new roads. Another commonly faced problem is not being able to understand the meaning of the sign. With the help of this Advanced Driver Assistance Systems (ADAS) application, drivers will no longer face the problem of understanding what the sign says. In this paper, we propose a method for Traffic Sign Detection and Recognition using image processing for the detection of a sign and an ensemble of Convolutional Neural Networks (CNN) for the recognition of the sign. CNNs have a high recognition rate, thus making it desirable to use for implementing various computer vision tasks. TensorFlow is used for the implementation of the CNN. We have achieved higher than 99% recognition accuracies for circular signs on the Belgium and German data sets.**

I. INTRODUCTION

Traffic sign detection and recognition have gained importance with advances in image processing due to the benefits that such a system may provide. The recent developments and interest in self-driving cars have also increased the interest in this field. An automated traffic sign detection and recognition system will provide the ability for smart cars and smart driving. Even with a driver behind the wheel, the system may provide vital information to the driver, reducing human errors that cause accidents. Certainly, with such a system integrated into vehicles, it is expected that the number of car accidents will be greatly reduced, saving human lives and the monetary value associated with car accidents. Automated systems will be able to control traffic on both open roads and intersections as well.

The motivation behind developing such a system is clear due to the benefits of such a system in saving lives and saving cost. Therefore, the objective of this work is to develop an automatic Arabic traffic sign detection and recognition system based on deep learning algorithm. The proposed system has the ability to recognize the signs within images captured by cameras and processed by a Deep CNN network.



#	Sign Name in English	Sign Name in Arabic	Sign Image	No. of Images	#	Sign Name in English	Sign Name in Arabic	Sign Image	No. of Images
1	Road Hump	مطب صناعي		124	13	No Overtaking	التجاوز محظور		112
2	Front or Left	الإتجاه الى الأمام او اليسار		144	14	Stop	قف		117
3	Front or Right	الإتجاه الى الأمام او اليمين		144	15	Slow	تمهل		107
4	Left Turn	منحنى الى اليسار		118	16	Right or Left	المرور على احد جانبي الطريق		104
5	Right Turn	منحنى الى اليمين		117	17	Speed 30	ممنوع تجاوز السرعة 30 في الساعة		132
6	Narrow From Left	الطريق يضيق من اليسار		105	18	Speed 40	ممنوع تجاوز السرعة 40 في الساعة		114
7	Narrow From Right	الطريق يضيق من اليمين		103	19	Speed 50	ممنوع تجاوز السرعة 50 في الساعة		110
8	No U Turn	ممنوع الدوران للخلف		104	20	Speed 60	ممنوع تجاوز السرعة 60 في الساعة		109
9	U Turn	الإتجاه الى الأمام او الخلف		102	21	Speed 80	ممنوع تجاوز السرعة 80 في الساعة		101
10	Rotor	الإتجاه مستدير		102	22	Speed 100	ممنوع تجاوز السرعة 100 في الساعة		80
11	Parking	مواقف		126	23	Pedestrian crossing	عبور المشاة		111
12	No Horn	ممنوع التزمير		132	24	No Parking	ممنوع الوقوف		110

Figure: 1captured by cameras and processed by a Deep CNN network

Most car accidents are caused by human error, either by drivers not noticing a certain sign or by drivers driving against the direction set by a certain traffic sign (i.e. traffic sign setting speed at 100 KM and a driver driving at a greater speed). For this reason, this paper carries the major objective of developing and improving the efficiency and robustness of the traffic sign detection system for Arabic Traffic Signs as well as handling associated issues. A recognition system should also classify traffic signs into different classes in a real-time environment and avoid recognition errors.

Machine learning is divided into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In this paper, the choice of deep learning for an unsupervised learning approach is done by design because even though basic traffic signs are limited yet combined with road signs, street name signs, etc. the dataset becomes larger with endless possibilities. The ultimate goal is to have a system fitted into cars and that can detect and recognize any traffic sign to assist the driver or assist in the self-driving process. With deep learning algorithms, unlabeled data can be used and the system can extract features automatically without human intervention. The rest of the paper is organized as follows: section 2 covers the related literature review. Section 3, details the proposed system. Section 4, explains the experimental data used in this paper. Section 5, shows the results along with discussion. Section 6, concludes the work and section 7 lists all the references used in this work.

II. SYSTEM ANALYSIS

2.1 Materials

The following hardware and software components are required to implement the Traffic Sign Detection and Recognition:

1. **Hardware Requirements:**

●	System	:	Pentium IV 2.4 GHz.
●	Hard Disk	:	40 GB.
●	Floppy Drive	:	1.44 Mb.
●	Monitor	:	15 VGA Colour.
●	Mouse	:	Logitech.
●	Ram	:	512 MB.

2. **Software Requirements:**

•	Operating system	:	Windows 7 Professional.
•	Coding Language	:	Python

2.2 Methods

2.2.1 Experimental Data

In this research, a new dataset for Arabic Traffic Signs is developed. As part of this research, 24 most common Arabic traffic signs are selected. The dataset consists of 2,728 images captured for 24 traffic signs. The images are captured from three connected cities (Khobar, Dammam and Dhahran) in the Eastern Province of Saudi Arabia. For each traffic sign, sign boards ranging 20 to 30 location were found and for each location, 5 photos were captured with different angles and distances. Table 1 describes the list all the selected traffic signs visual shape, their names in English and Arabic and the total number of images captured for each sign as part of the new database.

2.2.2 Preprocessing

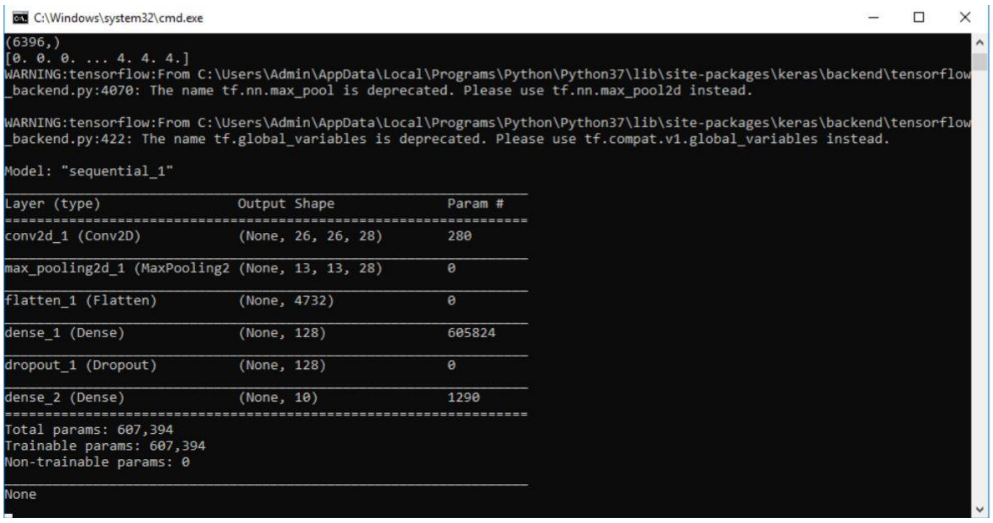
The images were RGB images with different dimensions which led to the need of pre-processing the images before inputting them to the CNN network. The images are transformed into Greyscale images and the dimension is also reduced to 30x30 pixels. Moreover, the total number of output classes is 24 classes each represents an Arabic traffic sign so all the images are carefully labeled and placed in their corresponding folders.

The number of images per class differs from one class to another as shown in Table 1. Fig. 2 shows a sample of each Traffic Sign after the preprocessing.he newly developed dataset consisting of 2728 images was randomly partitioned into 80% percent training set (2183 images) and 20% percent testing set (545) images. The training set was further partitioned; so 20% percent of the images (436 images) were used for validation.

2.3 Deep CNN Architecture:

The proper choice of different design hyper-parameters, such as non-linearity and pooling variants, directly affects the performance of the network. Since there is no clear guidance on how to choose the CNN hyper-parameters, many researchers tend to use trial and error experimentation in order to discover good settings [15]. We analyzed previous research studies that have been carried in the area of deep learning generally in order to choose the hyper-parameters setting that is more likely suitable for this research. So by analyzing the work done in [16-18], the hyper-parameters setting in this work is as follows.

Pooling Layer: It helps in reducing the amount of computations needed in the training process by reducing the dimension of the image, and therefore the overfitting problem is reduced. The max-pooling technique was used since the convergence rate is faster as compared to other subsampling techniques, and thus has a better generalization performance. The maximum pixel value in a non-overlapping region, equals to the window of the pooling, is the output of this layer; this is beneficial in creating position invariance.



```
C:\Windows\system32\cmd.exe
(6396,)
[0. 0. 0. ... 4. 4. 4.]
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:4070: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 28)        280
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 28)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 4732)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               605824
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 607,394
Trainable params: 607,394
Non-trainable params: 0
_________________________________________________________________
None
```

Figure: 2 Deep CNN Architecture

Non-Linearity: Since the relation between the images and their classes is not linear, introducing non-linearity in the CNN is needed. This is achieved by using non-linear activation so that the construction of the non-linear relation between the images and their classes is possible by the CNN. The Rectified Linear Unit (ReLU) is a widely used activation function in CNN. ReLU has the advantage that the CNN trains faster as compared to other functions. One of the ReLU variants was used which is the leaky ReLU since it overcomes the problem of dead neurons that is faced if the original ReLU is used. Basically, leaky ReLU does not output zero when the input values are less than zero, instead it outputs negative value. So after each convolution layer, and before the pooling layer, we added a leaky ReLU activation function

Dense Layer Activation Function and Loss Function: As mentioned above, we used the leaky ReLU activation function multiple times in the proposed network. However, the output of the fully connected layer, which is one dimensional vector, is passed to a softmax activation function to predict the label of that particular input. The input vector is transformed to a vector of the same size but the values range from zero to one only and the summation is always equals to 1. Softmax function outputs a probability distribution that is then converted to one-hot encoding vector. So the element that has the largest portion of probability distribution will have the value one in the one-hot vector and all other elements will have the value zero. To estimate the loss of softmax, the categorical cross-entropy function is used. Basically, categorical cross-entropy is used to measure the difference between the output of the softmax function and the one-hot encoding of the actual class. It is used as a part of gradient descent to evaluate CNN performance as an error measure between the true distribution and the predicted one. Softmax function and categorical cross-entropy are widely used in computer vision tasks when multiple classes are involved.

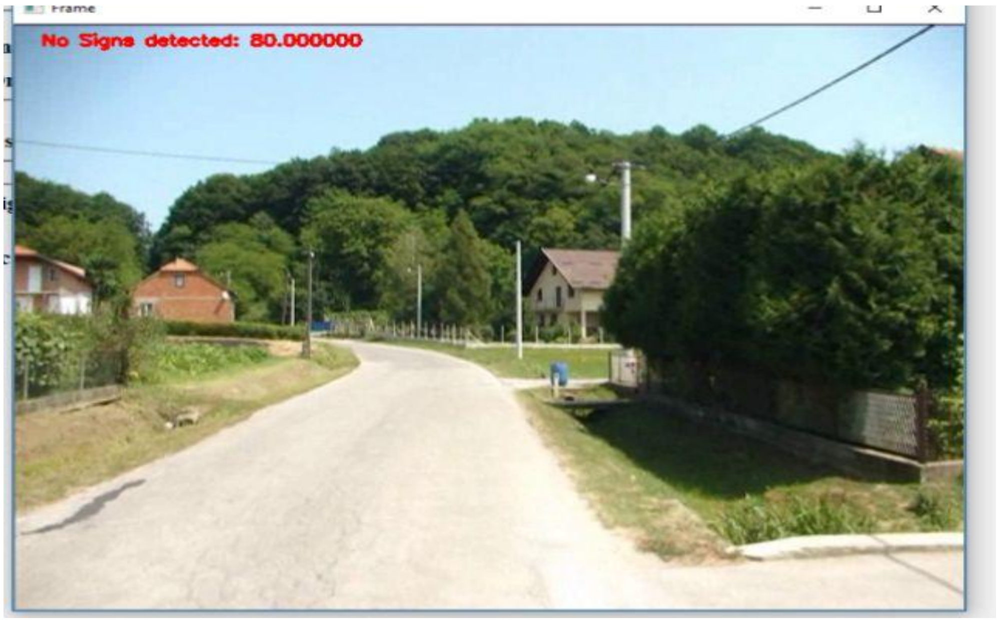


Figure: 3

After setting the needed hyper-parameters, the network was compiled using the Adam optimizer. The need of using optimization algorithms is the nature that CNN weights and biases are set automatically and they are of great importance in the field of machine learning. They work on optimizing a given function; whether maximizing or minimizing it with respect to its parameters. Since the loss function in CNN is differentiable with respect to its parameters, gradient descent based algorithms are often used; first order partial derivatives are also fast to compute. Adam is an optimization algorithm that is used to update network weights. It combines the advantages of other classical stochastic gradient descent which are Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). It is computationally efficient, requires less memory and has straightforward implementations. Given different parameters, this algorithm computes each parameter's adaptive learning rate by estimating the gradient's first and second moments [19]. Table 1 shows the summary of the parameters in each layer as well as the total parameters in the proposed network. This CNN network will be referred to as Model 1 in the results section

III. RESULTS AND DISCUSSION

The improved optimized CNN network layers are shown in Table 3. As observed in the optimized CNN architecture, the number of layers have been reduced without compromising the accuracy of the network. This CNN will be referred to as Model 2. Table 3 shows the experimental results performed on the proposed CNN architecture. The experiments were performed on epochs size of 10, 50, 100, and 150. For each epoch, the batch sizes of 50,100, 200 and 400 are used. The table for the experiment shows four columns with Validation Accuracy (Val. Acc), Validation Loss (Val. Loss), Test Accuracy (Test Acc) and finally Test Loss (Test Loss). Different design parameters were changed and the effect of the changes was tested by evaluating accuracy. A total of 16 different experiments were performed on the CNN architecture in order to identify an optimized design. Note that all the preprocessing steps remain the same as described above. The proposed CNN models are shown in the Table 1 and Table 3. Experimental results show that the number of epochs and number of batch size directly affects the test accuracies achieved for the models. The number of epochs is directly proportional with test accuracy; increasing epoch size increases the test accuracy. While the opposite is true for batch size as it is observed that increasing batch size decreases test accuracy. Overall, the test accuracy obtained for the improved CNN network-model 2 is better than those achieved for the initial CNN-model 1. In fact, model 2 results for epoch 150 achieved 100% accuracy for all batch sizes. The result is better than those reported in recent literature, however, it is pointed out here that the CNN is applied for Arabic Traffic Sign dataset and no recent literature was found targeting Arabic Traffic sign detection and recognition. No relationship was found for validation accuracy for both models in relation to epoch size and batch size.

IV. Conclusion

Automatic Arabic Traffic Sign (AATS) recognition system was designed using Convolutional Neural Networks (CNN). The training and testing dataset consists of more than 2,728 sign samples collected for 24 different traffic signs. The dataset went through a preprocessing stage before inputting it to the network. It got partitioned into training, testing and validating datasets. The initial design was based on similar previous work which was used as a base for the subsequent improved design. The final Deep CNN architecture proposed in this work consists of two convolutional layers, two maxpooling layers one dropout layer and 3 dense layers. 100% accuracy was obtained for epoch 150 for all batch sizes. We demonstrated the usefulness of the designed CNN architecture by implementing a practical system that can take real time Traffic Sign via camera attached to vehicle, classifies them to the corresponding sign, and then give voice notification to driver or take automatic decision for autonomous cars. Future work will include increasing the size of the dataset and publishing it so that it can be used by other researchers for benchmarking purposed. Work will also continue on developing more robust and computationally low-cost recognition systems.

V. References

[1] Lai, Y., Wang, N., Yang, Y., & Lin, L. (2018). Traffic signs recognition and classification based on deep feature learning. In 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM), Madeira, Portugal (pp. 622-629).

[2] M. Teague, "Image analysis via the general theory of moments," J. Opt Soc. Am., vol. 70 (8), pp. 920–930, 1980.

[3] Lee, A. (2015). Comparing Deep Neural Networks and Traditional Vision Algorithms in Mobile Robotics. Swarthmore College.

[4] Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S., & Babu, R. V. (2016). A taxonomy of deep convolutional neural nets for computer vision. Frontiers in Robotics and AI, 2, 36.

[5] Huang, Z., Yu, Y., Gu, J., & Liu, H. (2017). An efficient method for traffic sign recognition based on extreme learning machine. IEEE transactions on cybernetics, 47(4), 920-933.

[6] Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). Traffic-sign detection and classification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2110-2118).

[7] Pei, S., Tang, F., Ji, Y., Fan, J., & Ning, Z. (2018). Localized Traffic Sign Detection with Multi-scale Deconvolution Networks. arXiv preprint arXiv:1804.10428.

[8] Hatolkar, Y., Agarwal, P., & Patil, S. (2018). A Survey on Road Traffic Sign Recognition System using Convolution Neural Network.

[9] Pei, S., Tang, F., Ji, Y., Fan, J., & Ning, Z. (2018). Localized Traffic Sign Detection with Multi-scale Deconvolution Networks. arXiv preprint arXiv:1804.10428.

[10] Yang, Y., Liu, S., Ma, W., Wang, Q., & Liu, Z. (2018). Efficient Traffic-Sign Recognition with Scale-aware CNN. arXiv preprint arXiv:1805.12289.

[11] Yi, K., Jian, Z., Chen, S., Chen, Y., & Zheng, N. (2018). Knowledge-based Recurrent Attentive Neural Network for Traffic Sign Detection. arXiv preprint arXiv:1803.05263.

[12] Soetedjo, A., & Somawirata, I. K. (2018). An Efficient Algorithm for Implementing Traffic Sign Detection on Low Cost Embedded System. International Journal of Innovative Computing Information and Control, 14(1), 1-14..

[13] Hoo-Chang, S., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging, 35(5), 1285. [14] Kim, B., Park, S., Kim, K., Lim, J., & Nahm, K. (2018). Systematic process to determine DNBR limit of CHF correlation with repetitive cross-validation technique. Journal of Nuclear Science and Technology, 1-9.

[15] Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820.

[16] Latif, G., Alghazo, J., Alzubaidi, L., Naseer, M. M., & Alghazo, Y. (2018, March). Deep Convolutional Neural Network for Recognition of Unified Multi-Language Handwritten Numerals. In 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR) (pp. 90-95). IEEE.

[17] Latif, G., Iskandar, D. A., Alghazo, J., Butt, M., & Khan, A. H. (2018). Deep CNN based MR Image Denoising for Tumor Segmentation using Watershed Transform. International Journal of Engineering & Technology, 7(2.3), 37-42.

[18] Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., ... & Hodjat, B. (2019). Evolving deep neural networks. In Artificial Intelligence in the Age of Neural Networks and Brain Computing (pp. 293-312). Academic Press.

[19] Xiong, F., Xiao, Y., Cao, Z., Gong, K., Fang, Z., & Zhou, J. T. (2018). Towards good practices on building effective cnn baseline model for person re-identification. arXiv preprint arXiv:1807.11042.

.
.