International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 6 (2019), pp. 866-872

Rough Image Based Ensemble of Convolutional Neural Networks for Object Recognition

Thilagavathy Chenniappan[1] and Rajesh Reghunadhan[2]

[1] CMS College of Science and Commerce, Coimbatore, Tamilnadu, India.
[2]Department of Computer Science, Central University of Kerala, Periya-671316, Kasaragod, Kerala, India.

Abstract

Object recognition is one of the challenging computer vision problem due to the difficulty in deriving features for the effectual classification in different viewing directions, different lighting conditions, and differently colored objects. A set of feature extraction techniques for the effectual classification of a given object will not successfully classify a different object. Recently convolutional neural networks (CNNs) with more number of convolution layers and more number of filters are widely used for object classification, which has more than 10 million learning parameters. Due to these tremendous number of learning parameters, the results of CNNs are not still optimal. Hence this paper presents a rough image based ensemble of CNNs for effectual object recognition. The results on the benchmarking dataset shows promising results.

Keywords: Object recognition, rough set, ensemble of Convolutional neural networks (ECNN), rough image based ECNN (RIECNN)

I. INTRODUCTION

Objects with variable size, colors and types will look different with respect to viewing directions, reflections and illuminations. Hence, object recognition is one of the interesting, important and challenging computer vision problem.

Many feature extraction methods have been evolved for effectual object recognition including, but not limited to, HOG [1], LBP [2] etc. The Pascal visual object challenge have shown the world that a single set of features which can effectively classify an object cannot classify other objects in the world [3]. It means that separate features have to be obtained for different types of objects.

It can be seen that in most of the situations the number of features are much larger in number when compared to the total number of pixels in the image/object. Hence feature selection methods or dimensionality reduction mechanisms are usually applied before classification.

The set of feature vectors corresponding to the training images are used for training a classifier. The most common classifiers used for object classification includes, but not limited to, SVM [4], KNN [5], ANN [6], ANFIS [7], etc.

Recently it has been seen that convolutional neural network (CNN) [8] has evolved as a standard for the classification of objects/images. One of the interesting research in this direction is the voxel-based CNN for 3D object classification and retrieval by Cheng Wang et.al [9]. Accurate object localization in remote sensing images based on convolutional neural networks by Yang Long et.al is another interesting work [10]. A detailed review can be had from the object detection with deep learning review by Z.-Q. Zhao et. al [11].

There are many convolutional architectures designed as solutions for specific problems. Some of the architectures are LeNet for document recognition [12], AlexNet for ImageNet classification [13], GoogleNet for large visual recognition challenge [14], VGGNet for large scale image recognition [15], ResNet for image recognition [16], etc.

It so happens that, even convolutional neural network would not provide optimum classification results due to over learning. One option is to increase the number of layers in CNN. But the number of features/parameters to be learned increases with the number of layers, which in turn leads to under learning.

Hence in this paper, an ensemble of CNNs (with minimum number of layers) are trained using a set of rough images derived from input images, followed by majority voting for the effectual object recognition. The results on benchmarking object dataset are promising.

This paper is organized as follows. Section 2 provides a quick review of convolutional neural network. Section 3 provides the architecture and the working of ensemble of convolutional neural network using rough images. Section 4 provides the experimental results and comparison. Section 5 concludes the paper.

II. CONVOLUTIONAL NEURAL NETWORK – A QUICK REVIEW

The CNN architectures will have a set of convolutional layers. Each convolutional layer will have a set of filters and filter values are learned during training. The size of the filters will be different or same in different convolution layers. Figure 1 & 2 shows examples of 2D convolution.

1	0	1	5	3
6	7	2	9	3
0	6	1	8	2
1	4	8	5	4
7	5	3	2	4

2D Image

0	1	0
1	4	1
0	1	0

2D filter

10	11	20
13	28	23
34	27	22

Convolved Image

Figure 1: Example of convolution on a 5x5 image with zero padding using a 2D filter of size 3x3 with a stride of 2.

1	0	1	5	3
6	7	2	9	3
0	6	1	8	2
1	4	8	5	4
7	5	3	2	4

2D Image

0	1	0
1	4	1
0	1	0

2D filter

42	25	54
36	28	49
36	45	42

Convolved Image

Figure 2: Example of convolution on a 5x5 image without zero padding using a 2D filter of size 3x3 with a stride of 1.

The CNN architectures will have activation functions after convolution layers. Some of the activation function are discussed here. Sigmoidal function is shown in figure 3 and is given by

$$f(x;a,c)=\frac{1}{1+e^{-a(x-c)}}$$

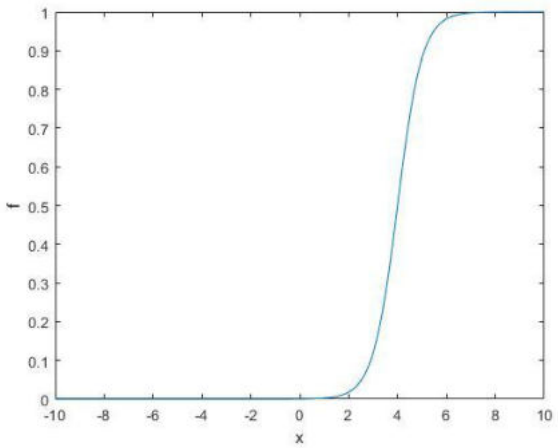

Figure 3: Sigmoid function with a=2 and c=4.

Hyperbolic tangent (tanh) is shown in figure 4 and is given by

$$\tanh(x)=\frac{e^{2x}-1}{e^{2x}+1}$$

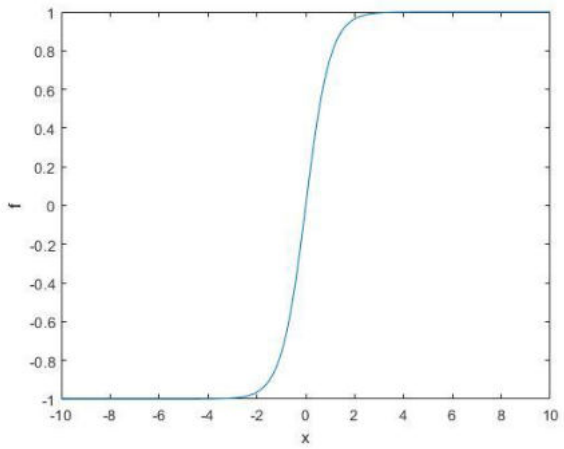

Figure 4: Hyperbolic tangent function

Linear transfer function is shown in figure 5 and is given by

$$f(x)=x$$

Rectified Linear Unit (ReLU) [17] also called positive linear transfer function is shown in figure 6 and is given by

$$f(x)=\begin{cases}x & if\ x\geq 0\\ 0 & otherwise\end{cases}$$

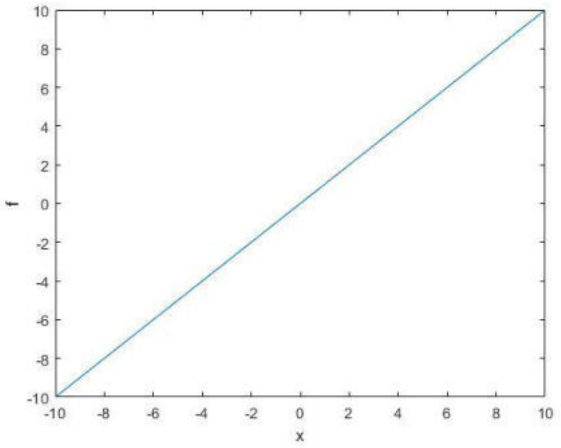

Figure 5: Linear transfer function

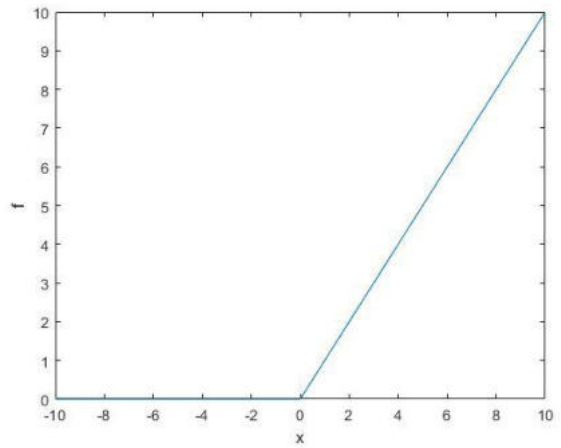

Figure 6: Rectified Linear Unit

A leaky ReLU [18] is shown in figure 7 and is given by

$$f(x)=\begin{cases}x & x\geq 0\\ scale\times x & x<0\end{cases}$$

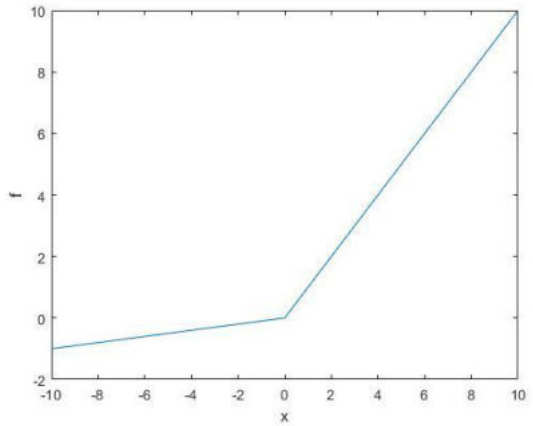

Figure 7: Leaky Rectified Linear Unit

Saturating linear transfer function is shown in figure 8 and is given by

$$f(x)=\begin{cases}0 & if\quad x\leq 0\\ x & if\quad 0\leq x\leq 1\\ 1 & if\quad 1\leq x\end{cases}$$

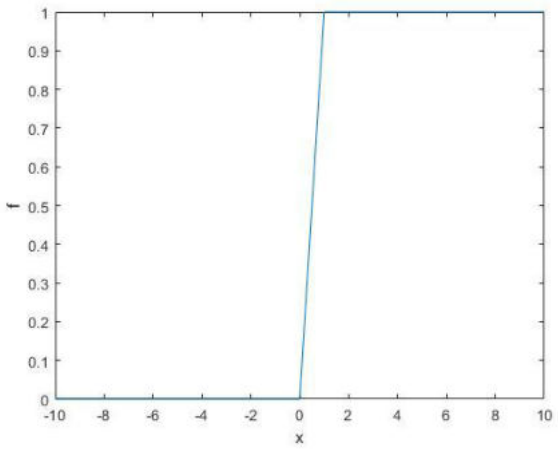

Figure 8: Saturating linear transfer function

Clipped ReLU is the modified form of saturating linear transfer function. Clipped ReLU is given by

$$f(x)=\begin{cases} 0 & if \quad x\le 0 \\ x & if \quad 0\le x\le ceiling \\ 1 & if \quad ceiling\le x \end{cases}$$

Other similar functions are parametric ReLU (PReLU) [19], randomized leaky ReLU (RReLU) [20], S-Shaped ReLU (SReLU) [21], bipolar ReLU (BReLU) [22], etc.

The CNN architectures will have pooling layers whose duty is to reduce the size of input images/volumes/features. There are various pooling layers like minimum, average, etc. but the most commonly used pooling layer is the max-pooling layer. Figure 9 & 10 shows examples of maxpooling. Figure 11 & 12 shows examples of min pooling and average pooling respectively.

After a set of convolution-activation-pooling layers, there will be a set of fully connected layers finally leading to the output layer. Figure 13 shows an example of CNN architecture.

1	**2**	**9**	**8**
4	**3**	**4**	**3**
8	**5**	**2**	**6**
6	**7**	**5**	**1**

Max pooling with 2x2 filters with stride 2 →

4	**9**
8	**6**

Figure 9: Example of Max pooling with a stride of 2

1	**2**	**9**	**8**
4	**3**	**4**	**3**
8	**5**	**2**	**6**
6	**7**	**5**	**1**

Max pooling with 2x2 filters with stride 1 →

4	**9**	**9**
8	**5**	**6**
8	**7**	**6**

Figure 10: Example of Max pooling with a stride of 1

1	**2**	**9**	**8**
4	**3**	**4**	**3**
8	**5**	**2**	**6**
6	**7**	**5**	**1**

Min pooling with 2x2 filters with stride 2 →

1	**3**
5	**1**

Figure 11: Example of Min pooling

1	**2**	**9**	**8**
4	**3**	**4**	**3**
8	**5**	**2**	**6**
6	**7**	**5**	**1**

average pooling with 2x2 filters with stride 2 →

2.5	**6**
6.5	**3.5**

Figure 12: Example of average pooling

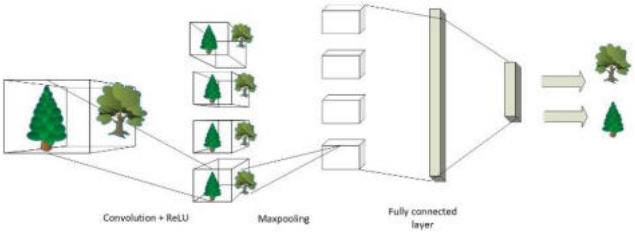

Figure 13: Convolution neural network architecture

III. ROUGH IMAGE BASED ENSEMBLE OF CNN

This section presents the proposed work for the recognition of objects using rough image based ensemble of convolutional neural networks (RIECNN). The general architecture of the proposed method is shown in figure 14.

It has three different stages, namely, rough image generation, ensemble of convolutional neural network and the majority voting.

Stage 1: Rough Image Generation

Rough sets, proposed by Pawlak in 1981, have been evolved as an effectual technique for the representation of uncertainty, vagueness and ambiguity [23][24]. Rough sets have seen wide applications across the disciplines [25].

Recently, preference based rough sets proposed by Qinghua Hu et.al. has seen its application in feature selection [26].

Rough images in the proposed work are obtained based on the concept of preference based rough set (PRS). The images are provided to PRS as vectors. Three sets of reducts on dataset (ie., pixel locations in the image that are most responsible for the classification) are obtained for the set of training images based on upwards consistency, downwards consistency and global consistency as proposed by Qinghua Hu et.al.[26].

The three sets of reducts obtained from preference based rough set (PRS) for images of size m×n are given below, where $i \ \varepsilon \ [1:m]$ and $j \ \varepsilon \ [1:m]$

$$UC=\left\{(i,j)\Big/ \begin{matrix} set & of & all & (i,j) & obtained & based \\ on & PRS & with & upward & consistency & \end{matrix}\right\}$$

$$DC=\left\{(i,j)\Big/ \begin{matrix} set & of & all & (i,j) & obtained & based \\ on & PRS & with & downwards & consistency & \end{matrix}\right\}$$

$$GC=\left\{(i,j)\Big/ \begin{matrix} set & of & all & (i,j) & obtained & based \\ on & PRS & with & global & consistency & \end{matrix}\right\}$$

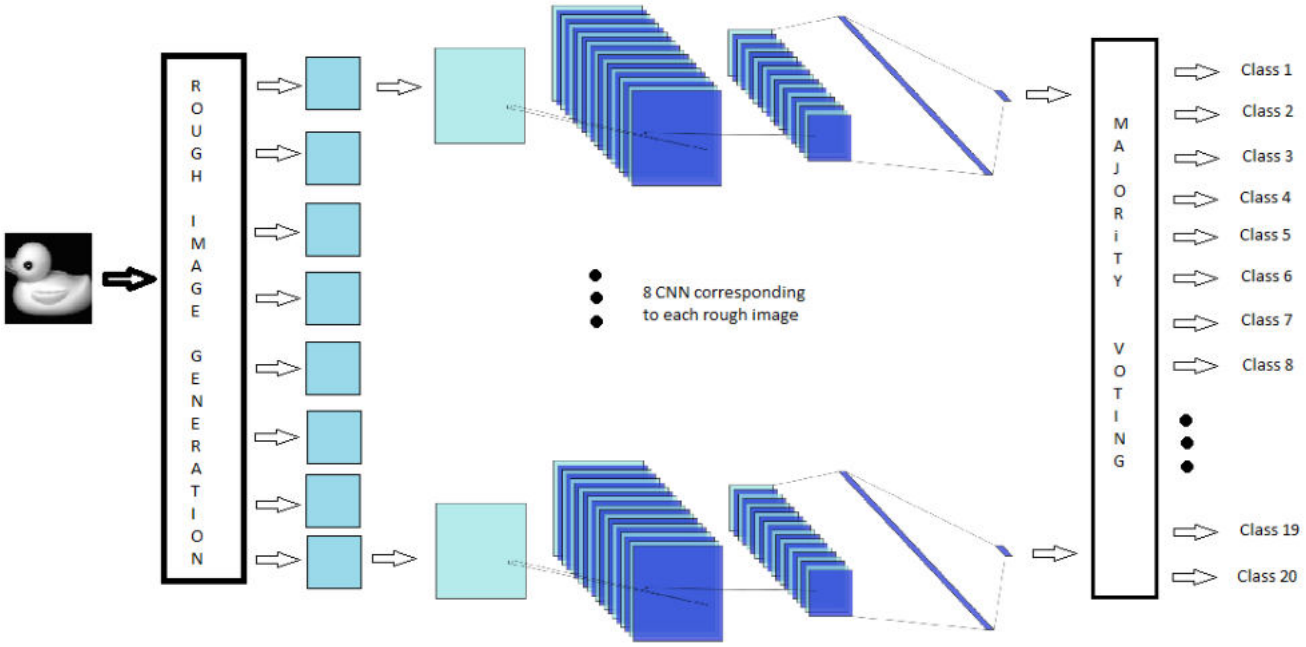

Figure 14: Ensemble of convolutional neural networks using rough images.

Then the rough images namely, RI_1, RI_2, RI_3, RI_4, RI_5, RI_6, RI_7, RI_8 can be obtained as given below.

$$RI_1(i,j)=\begin{cases}0 & if \quad (i,j)\varepsilon UC\\ I(i,j) & otherwise\end{cases}$$

$$RI_2(i,j)=\begin{cases}0 & if \quad (i,j)\varepsilon DC\\ I(i,j) & otherwise\end{cases}$$

$$RI_3(i,j)=\begin{cases}0 & if \quad (i,j)\varepsilon GC\\ I(i,j) & otherwise\end{cases}$$

$$RI_4(i,j)=\begin{cases}255 & if \quad (i,j)\varepsilon UC\\ I(i,j) & otherwise\end{cases}$$

$$RI_5(i,j)=\begin{cases}255 & if \quad (i,j)\varepsilon DC\\ I(i,j) & otherwise\end{cases}$$

$$RI_6(i,j)=\begin{cases}255 & if \quad (i,j)\varepsilon GC\\ I(i,j) & otherwise\end{cases}$$

$$RI_7(i,j)=\begin{cases}0 & if \quad (i,j)\varepsilon GC\, or \quad (i,j)\varepsilon UC \quad or \quad (i,j)\varepsilon DC\\ I(i,j) & otherwise\end{cases}$$

$$RI_8(i,j)=\begin{cases}255 & if \quad (i,j)\varepsilon GC\, or \quad (i,j)\varepsilon UC \quad or \quad (i,j)\varepsilon DC\\ I(i,j) & otherwise\end{cases}$$

Stage 2: Ensemble of CNN

For each of the eight type of rough images, separate CNNs are trained. The architecture of CNN is shown in figure 15. In the convolutional layer, 20 filters of size 5x5 is applied. The activation function used after convolutional is ReLU.

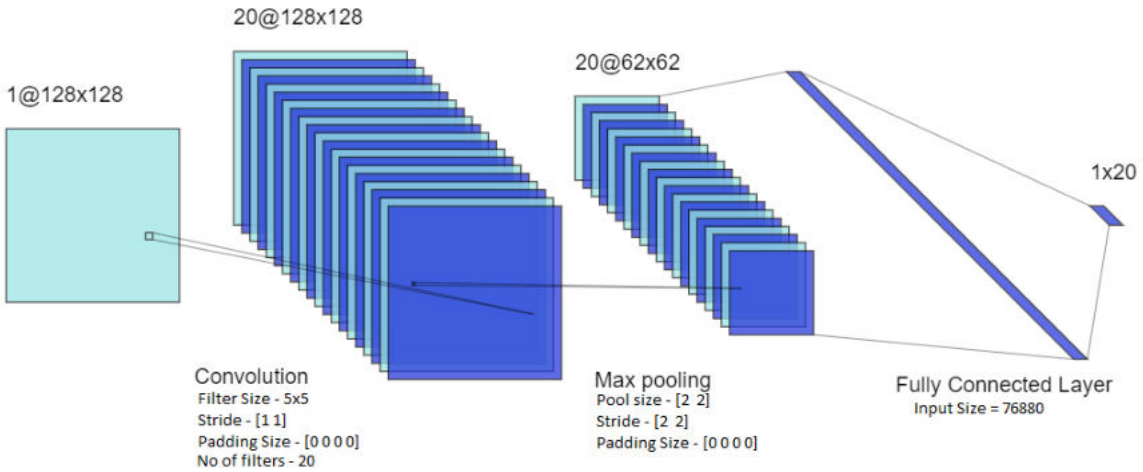

Figure 15: Convolutional Neural Network Architecture

A pool size of 2x2 with stride [2 2] is applied in the max pooling layer. It has a fully connected layer with an input size of 76880. Softmax is used after the fully connected layer. The output layer has 20 neurons equivalent to the number of classes.

Stage 3: Majority voting.
Each CNNs in the ensemble of CNN will provide/predict the output. All these predicted outputs are considered together and a majority voting scheme is adopted to get the final predicted class.

IV. RESULTS AND DISCUSSIONS

"Columbia Object Image Library (COIL) – 20" is used for the study reported in this paper. COIL–20 is image collection consisting of 20 objects imaged in 72 directions per object – making it a real challenging 3D object recognition [27]. Figure 16 shows the twenty objects in the dataset. Figure 17 shows the 72 directional image of one of the objects.

Figure 16: The twenty objects from the COIL-20 dataset.

Rough image based ensemble of CNNs is used for the recognition of objects. The learnable parameters include 520 parameters in the convolutional layer (5x5x1x20 weights + 1x1x20 bias) and 1537620 parameters in the fully connected layer (20x76880 weights + 20x1 bias) totaling to 1538140 parameters. Stochastic gradient descent with momentum of 0.9 is used to train the network with an initial learning rate of 0.0001 for a maximum of 20 epochs with mini-batch for each training iteration fixed as 128.

The result of object recognition using CNN and the result of object recognition using rough image based ensemble of CNNs is given in table 1 along with other results from the literature. The results show that rough image based ensemble of CNNs provide comparatively better recognition rate.

Table 1: Recognition rate of COIL-20 dataset

	Algorithms/Methods	**Accuracy**
1	Pose-Free Descriptors (PFD) [28]	67.40
2	Semi-supervised Two dimensional Classification (SSTC) [29]	76.80
3	Visual Attention and Object Recognition With a Biologically Plausible Retina (NIMBLER) [30]	78.87
4	Discriminative Pose-Free Descriptors (DPFD) [28]	82.20
5	Aligned Discriminative Pose Robust Descriptors (ADPR) [31]	83.00
6	Appearance Based 3D Object Recognition Using IPCA-ICA [32]	87.88
7	Convolutional Neural Network	98.16
8	Ensemble of CNN using Rough Images	99.48

V. CONCLUSION

In this paper a rough image based ensemble of CNNs for effectual object recognition is proposed. In this method a set of rough images from input images are obtained and then passed it through an ensemble of CNN (with minimum number of layers) followed by majority voting for the effectual object recognition. The results on the benchmarking dataset shows promising results.

ACKNOWLEDGEMENT

The authors would like to acknowledge Central University of Kerala, Bharathiar University and CMS College of Arts and Commerce for the research support.

REFERENCES

[1] Dalal N. and Triggs B. 2005. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1(June): 886–893.

[2] Ojala T., Pietikainen M., Maenpaa T. 2002. Multiresolution Gray Scale and Rotation Invariant Texture Classification With Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24(7): 971-987.

[3] Mark E., Eslami S.M., Gool L.V., Christopher K.I.W., John W., Andrew Z. 2015. The PASCAL Visual Object Classes Challenge: A Retrospective. International Journal Computer Vision. 111:98–136.

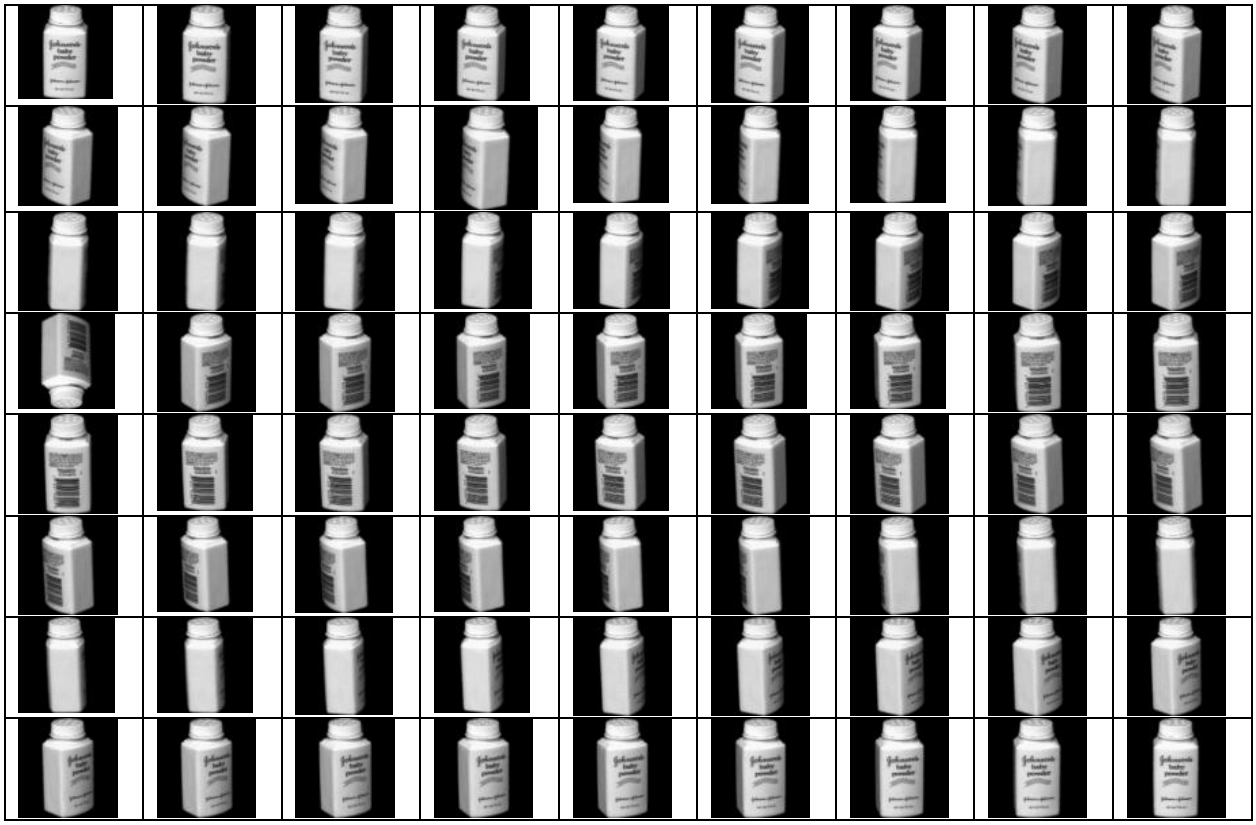

Figure 17: 72 directional image of one of the objects from the COIL-20 dataset.

[4] Rachid K., Karim O., Abdallah M. 2019. The cluster correlation-network support vector machine for high-dimensional binary classification. Journal of Statistical Computation and Simulation. 89(6): 1020-1043.

[5] Zheng P., Zhang J. 2019. Application of Variational Mode Decomposition and k-Nearest Neighbor Algorithm in the Quantitative Nondestructive Testing of Wire Ropes. Shock and Vibration. vol. 2019, Article ID 9828536. https://doi.org/10.1155/2019/9828536.

[6] German I., Parisia R.K., Jose L.P., Christopher K., StefanW. 2019. Continual lifelong learning with neural networks: A review. Neural Networks. 113(May):54-71

[7] Jaafari A., Zenner E., Panahi M., Shahabi H. (2019). Hybrid artificial intelligence models based on a neuro-fuzzy system and metaheuristic optimization algorithms for spatial prediction of wildfire probability. Agricultural and Forest Meteorology. 266-267: 198-207.

[8] Shelly S., Avi B.C., Orit S., Michal M.A., Hayit G., Eyal K. 2019. Convolutional Neural Networks for Radiologic Images: A Radiologist's Guide. Radiology. 290(3).

[9] Cheng W., Ming C., Ferdous S., Mohammed B., Jonathan L. 2019. NormalNet: A voxel-based CNN For 3D object classification and retrieval. Neurocomputing. 323: 139–147

[10] Yang L., Yiping G., Zhifeng X., Qing L. 2017. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. IEEE Transactions on Geoscience and Remote Sensing. 55(5): 2486-2498

[11] Zhao Z.Q., Peng Z., Xu S.T., Xindong W. 2019. Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems. 1-21.

[12] Yann L., Leon B., Yoshua B., Patrick H. 1998. Gradient based learning applied to document recognition. Proc. of the IEEE. Nov: 1-46

[13] Alex K., Ilya S., Geoffrey E.H. 2017. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM. 60(6): 84-90

[14] Christian S., Wei L., Yangqing J., Pierre S., Scott R., Dragomir A., Dumitru E., Vincent V., Andrew R. 2015. Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[15] Karen S., Andrew Z. 2015. Very deep convolutional networks for large scale image recognition. ICLR. 1-14

[16] Kaiming H., Xiangyu Z., Shaoqing R., Jian S. 2016. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]. CVPR. 770-778

[17] Nair V., Hinton G.E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. 27th International Conference on Machine Learning,

ICML'10, USA, Omnipress. ISBN 9781605589077. 807–814

[18] Andrew L.M., Hannun A.Y., Andrew N.Y. 2013. Rectifier nonlinearities improve neural network acoustic models. Proc. ICML. 30 (1).

[19] Kaiming H., Zhang X., Ren S., Sun J. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. ICCV. 1026-1034.

[20] Xu B., Wang N., Chen T., Li M. 2015. Empirical Evaluation of Rectified Activations in Convolutional Network. arXiv:1505.00853 [cs.LG]

[21] Jin X., Xu C., Feng J., Wei Y., Xiong J., Yan S. 2015. Deep Learning with S-shaped Rectified Linear Activation Units. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. 1737-1743

[22] Eidnes L., Nøkland A. 2018. Shifting Mean Activation Towards Zero with Bipolar Activation Functions. International Conference on Learning Representations (ICLR) Workshop. 1-13.

[23] Pawlak Z. 1981. Rough Sets. Research Report PAS 431, Institute of Computer Science, Polish Academy of Sciences.

[24] Pawlak Z. 1982. Rough sets. International Journal of Parallel Programming. 11 (5): 341–356.

[25] Qinghua Z., Qin X., Guoyin W. 2016. A survey on rough set theory and its applications. CAAI Transactions on Intelligence Technology. 1(4): 323-333.

[26] Qinghua H., Daren Y., Maozu G. 2010. Fuzzy preference based rough sets. Information Sciences. 180: 2003–2022

[27] Nene S.A., Nayar S.K., Murase H. 1996. Columbia object image library (COIL-20). Technical report, CUCS-005-96, Department of Computer Science, Columbia University.

[28] Soubhik S., Sivaram P.M., Soma B. 2015. Discriminative Pose-Free Descriptors for Face and Object Matching. ICCV. 3837-3845

[29] Zhigang M., Yi Y., Feiping N., Nicu S. 2013. Thinking of images as what they are: compound matrix regression for image classification. Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China. 1530-1536

[30] Christopher K. 2019. NIMBLER: A Model of Visual Attention and Object Recognition With a Biologically Plausible Retina.

[31] Soubhik S., Devraj M., Soma B. 2017. Aligned discriminative pose robust descriptors for face and object recognition. Proceedings of ICIP. 820-824

[32] Ananthashayana V.K., Asha V. 2008. Appearance Based 3D Object Recognition Using IPCA-ICA, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B1. Beijing, 1083-1090