

# Plankton Image Classification using Convolutional Neural Networks

Hussein A. Al-Barazanchi, Abhishek Verma, and Shawn Wang

Department of Computer Science, California State University, Fullerton, CA, USA

**Abstract**—*The study of plankton distribution is an important tool used for assessing the changes to marine ecosystem. Having a robust automated system for classification of plankton images plays an important role in advancing marine biology research. The images used in this study come from the SIPPER system. The challenges with SIPPER's plankton image dataset are the high degree of similarities between different classes, high variability within the same class, partial occlusion, and noise. Also, traditional computer vision techniques require tedious work to find suitable features to represent plankton. To overcome those issues, we propose the use of convolutional neural networks. Results of the experiments on SIPPER dataset show improvement in classification accuracy in comparison to other state of the art approaches. Another major advantage of our approach is the scalability for classification of new classes without the need for feature engineering.*

**Keywords**—plankton images, SIPPER system, convolutional neural networks, image search.

## 1 Introduction

The two main types of plankton: phytoplankton (drifting plants) and zooplankton (animal plankton) are considered as the main source of food for many aquatic animals. Also, carbon fixation by phytoplankton in the ocean plays an important role in the global carbon cycle. Due to their high ability to respond to changes in their environment: like pollution; plankton is considered as an alarm signal for detection of changes in marine ecosystem. Therefore, the fast mapping of plankton distribution is an important mission for oceanographic research.

In the early days, scientists were limited to the use of traditional techniques to investigate the distribution of plankton, such as Niskin bottles, towed nets, or pumps to collect samples. The counting and recognition of species was done by hand. As time progressed, use of cruise ships allowed researchers to collect bigger number of samples. However, the process of knowing the distribution of plankton remained laborious, time consuming and not elegant for real applications. Gradually, owing to the

advancement in imaging technology several underwater devices for sampling were developed such as the HOLOMAR underwater holographic camera system [1], video plankton recorder (VPR) to [2], and the shadowed image particle profiling and evaluation recorder (SIPPER) [3]. With the use these instruments it became possible to perform continuous sampling of plankton. It was a major leap on the side of data collection, but the process of analysis remained tediously manual. In more recent times, the automated analysis of the pictures collected by these devices became feasible using sophisticated computer vision algorithms.

We can trace the first work on plankton image classification obtained by using VPR [4]. In 2005, Lue et al. [5] achieved 90% accuracy on plankton images recorded using SIPPER system. Their approach was based on classification with Support Vector Machine (SVM) and they did not make use of those image features that depend on the contour information. During the following year, a new shape descriptor was proposed by Tang et al [6] and the technique was named Normalized multilevel dominant eigenvector estimation, it achieved 91% recognition accuracy. Zhao et al. [7] extended the work in [6]; they make use of random sampling and multiple classifiers to achieve about 93% of recognition accuracy.

Regardless of the success shown by the aforementioned techniques, they suffer from one major drawback, which is total dependence on features engineering, i.e., the accuracy is determined by the quality of the used features. The process of feature engineering is difficult and requires much effort. Based on previous techniques, it requires extensive work to identify new classes of plankton; new features need to be identified, which could suitably represent those new classes. Hence, scaling up poses a challenge for those techniques.

In this paper we propose the use of convolutional neural networks (CNN), which is end to end learning framework. One major advantage of convolutional neural networks is its easy scalability to classify new classes. Based on the experimental results our proposed CNN

Table 1: Plankton Types Distribution

Class No.	Class Name	Count
0	Acantharia	131
1	Calanoid	172
2	Chaetognath	450
3	Doliolid	485
4	Larvacean	529
5	Radiolaria	563
6	Trichodesmium	789

algorithm exceeds the performance of the previous methods.

This paper is organized as follows. In section 2, we describe the plankton image dataset obtained from SIPPER system. In section 3, we give details of our proposed CNN algorithm for this classification task. Section 4 discusses our implementation and gives experimental results. Concluding remarks appear in Section 5.

## 2 Plankton image dataset

The plankton images that we used in our experiment are provided by the University of South Florida (Tampa, FL, USA). They are captured by the SIPPER system. The images were collected during the years 2010 to 2014 from the Gulf of Mexico. The dataset contains 81 plankton types with more than 750 thousand images. In order to compare our method with the previous studies [5], [6], [7] we choose the exact same 7 types from the 81 types. Table 1 gives the names of these seven types and their distribution.

There are many challenges with plankton images represented by the differences between the species of the same class and similar appearance between different classes. Besides that, occlusion and deformation add more difficulty. Figure 1 gives a randomly chosen sample from the SIPPER dataset. A major issue is the need to find extra features to represent any extra classes added to the dataset. The classic solution to this problem is to do features engineering and to find useful features to represent the new class. To overcome this problem, we need a robust scalable approach toward feature extraction without depending upon features engineering and followed by robust classification. Our proposed solution uses a convolutional neural network.

## 3 Convolutional neural network

Visual recognition tasks require the construction of a suitable and robust feature set to represent the world

around us. Those features should be invariant to outside variations of objects and keep enough relevant information to be able to recognize objects. The challenge is how to automatically learn such features without the need for human intervention. One approach is to simulate the process by which animals perform the task of object recognition and classification. Convolutional neural networks are proved to be the best model that simulates the vision abilities in animals with end to end feature learning and classification [9].

Convolutional neural networks are models that can learn invariant features and they are inspired from the vision mechanism in animals. This mechanism discovered during Hubel et al. [10] work on cat’s visual cortex. Fukushima’s Neocognitron [11] was the first simulated program based on this architecture. LeCun et al. [12] showed a successful use of convolution networks for handwritten recognition. Figure 2 illustrates the architecture used by LeCun et al. [12]. The popularity of convolutional neural networks started after the impressive success achieved in ImageNet Competition [13].

The typical design of convolutional neural network is stacked stages one after the other. These stages are followed at the end by a fully connected neural network.

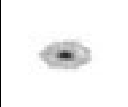










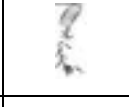
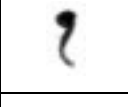


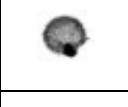
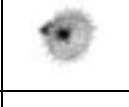




Class Name	Sample Images		
Acantharia			
Calanoid			
Chaetognath			
Doliolid			
Larvacean			
Radiolaria			
Trichodesmium			

Figure.1. Random samples from seven classes of the SIPPER dataset.

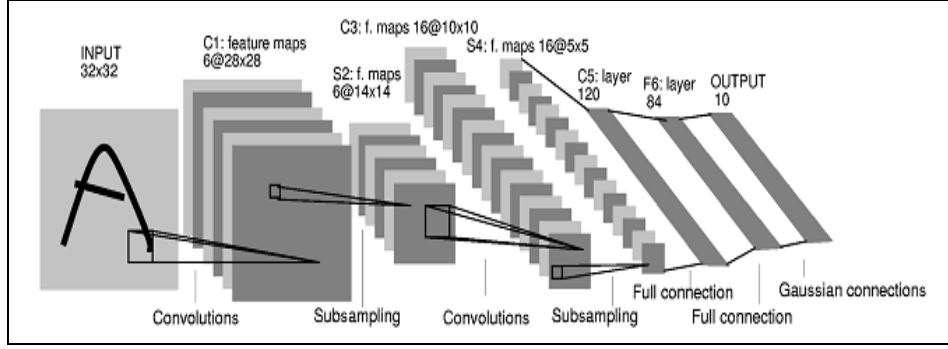


Figure 2. An example of convolutional neural network for handwritten recognition system [12].

The fully connected neural network works here as a classifier. At each level the convolutional neural networks consist typically of filters layer, non-linearity layer, and feature pooling layer [9] [12] [13]. The use of multi-level convolutional neural networks enables the system to learn the features' hierarchies. It starts from low-level features represented by the pixels, next it ascends to the mid-level features represented by edges and parts followed by the high-level features, which are objects.

### 3.1 Filter layer

The filter layer of the convolutional neural network is a variant form of neural networks in several aspects [15] [16]. First, neurons in convolutional neural network are sparsely connected to neurons in the next layer. On the other hand they are fully connected in regular neural networks. Second, convolutional neural networks' neurons follow a topographical layout. This means that connections are based on the related areas in the visual context. The regular neural networks do not make use of this feature. In our method, the images are fed to the convolutional layer in the format described in the equation (1). The symbols  $h$  and  $w$  refer to the height and width of the images while  $c$  refers to the number of color channels of the images.

$$h \times w \times c \quad (1)$$

$$y_j = b_j + \sum_i K_{ij} * x_i \quad (2)$$

We refer to each input to the layers as  $x_i$ . Where  $i$  is to indicate the filter number. Each component in the filters has the form  $x_{ijk}$ . The output will be computed by equation (2) [9]. The kernel (filter)  $K$  in the bank of filters has  $K_n \times K_m$  dimensions depending upon the specified reception field; where  $n$  and  $m$  are the size of the reception field. Also,  $*$  indicates convolution operator while  $b$  is the network bias. Each kernel finds specific features at every place on the image. This means moving the kernel spatially will look for a particular feature in an image. As

to which exact image feature a particular kernel will look for is decided dynamically by the algorithm [9].

### 3.2 Non-Linearity layer

The typical activations function for the output of neurons are  $\tanh()$  and  $\text{sigmoid}()$  functions [9] [13], which are shown in equations (3) and (4). The problem with these activations is its slow speed when used with gradient descent. Using non saturating activation functions proves to be faster by many times of magnitude [13] [14]. We restrict our work to Rectified Linear Units (ReLUs), which is represented by equation (5).

$$\text{sig}(x) = 1 / (1 + e^{-x}) \quad (3)$$

$$\tanh(x) = (e^{2x} - 1) / (e^{2x} + 1) \quad (4)$$

$$f(x) = \max(0, x) \quad (5)$$

### 3.3 Pooling layer

Pooling is a technique for dimensionality reduction [16]. This layer aims to remove unrelated information and keeps only relevant ones [17]. The input to this layer is the output of the non-linearity layer. The output of this layer is the reduced version of the input [15]. This layer has pool units that are organized in topographical way and connect to local areas in the input coming from the non-linearity layer.

The replication of neurons' weights in the filter bank helps to detect features in the different regions of the image. The problem that arises in those features is that they are not translation invariant. The pooling is used to make the features invariant to translation in the input. Pooling helps to reduce the sensitivity of activations in neural network to the pixels' locations and the neural network structure [18]. The common functions used in pooling are maximum and average functions and they are usually named max-pooling and average-pooling. There are two different ways to feed the input to those functions,

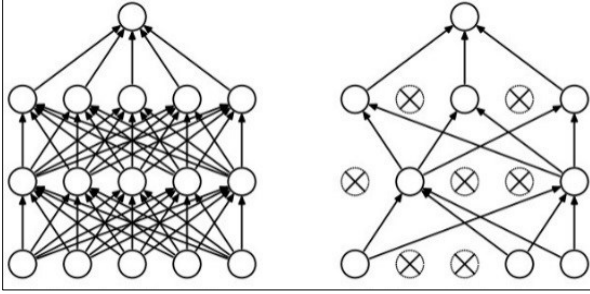


Figure 3. On the left fully connected neural network. On the right neural network after the dropout [19].

which could be either the separate or overlapping mode [15].

### 3.4 Dropout layer

Dropout is a recent technique developed by Srivastava et al. [19]. The purpose of this layer is to reduce the problem of overfitting and enhance generalization on the test data. This method works by removing random neurons with their connections during the process of learning. Fig 3 shows an example of the dropout layer.

### 3.5 Output layer

The output layer is different from all aforementioned layers. The output of this layer is in the form of probabilities that sum to one. The probability values indicate the confidence level about the chosen class where higher value means higher confidence. The common function used in this layer is the *softmax* function. This function is linear and it uses the log probability.

### 3.6 Learning algorithm

We used backpropagation algorithm for learning and

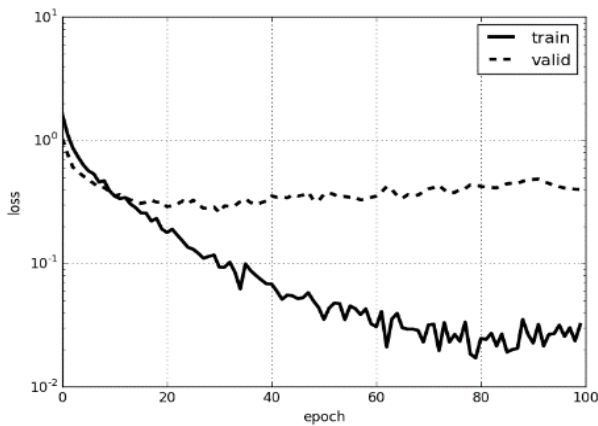


Figure 4. Training and validation loss for the highest accuracy in 1-layer convolutional neural network.

Table 2: Classification Accuracy for 1-Layer CNN

No of Filters	Reception Field	Accuracy (%)
8	2*2	88.90
8	3*3	90.71
8	4*4	90.18
<b>8</b>	<b>5*5</b>	<b>90.92</b>
16	2*2	89.86
16	3*3	89.75
<b>16</b>	<b>4*4</b>	<b>90.18</b>
16	5*5	90.07
32	2*2	88.68
32	3*3	89.75
32	4*4	92.38
<b>32</b>	<b>5*5</b>	<b>92.39</b>

stochastic gradient descent for optimization. We set the batch size to 32. Initially the momentum and learning rate are set to 0.9 and 0.01 respectively. The momentum and learning rate are continually updated as we get close to the minima. Also, we used a technique called early stopping [20] [21] to prevent overfitting problem in the neural network.

## 4 Implementation and experimental results

The total number of images in the seven class subset from the SIPPER dataset is 3119. The data used in the experiments is randomly divided into training, testing, and validation. Training consists of 56% of the images from each class, and testing and validation data is 30% and 14% respectively from each class. This gives us a total of 1745 samples for training, 437 samples for validation, and 937 samples for testing.

We divided our experiments into two phases. For the first phase we use only one convolutional layer and then extend the idea into the second phase with two convolutional layers. To standardize the testing results, we set many hyper parameters to fixed values. We used a fixed size classification layer, which is 2 fully connected layers. The first fully connected layer has 256 neurons while the second layer has 128 neurons. Each of them is followed by a 50% dropout layer. The output layer has the same number of classes which is 7 followed by the *softmax* activation function. All the convolutional layers and fully connected layers are followed by a ReLu activation function and pool layer. We limit the number of hyper-parameters to be tuned to the number of layers, number of filters in each layer, and the size of the reception field.

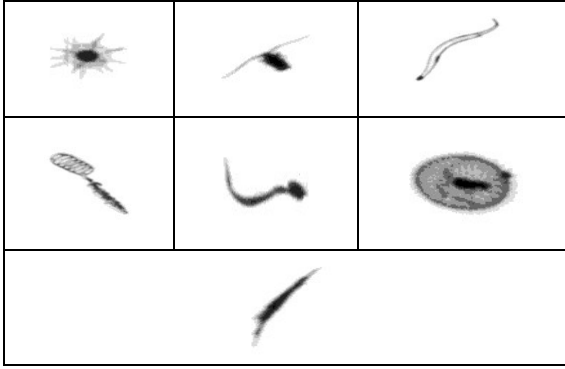


Figure 5. Correctly classified example images from 1-layer CNN, from left to right and top to bottom the images are from class number 0-6.

#### 4.1 Phase one: 1-layer CNN

To simplify the experiments in this phase, we start with only one layer for the convolutional neural network. We set the number of filters to 8, 16, and 32. The reception field is set between 2 and 5. Table 2 shows the accuracy details related with each of our configuration. We highlight the best accuracy associated with different number of filters. Our results show that our algorithm performs better than what is achieved in [5] and [6].

Overfitting is a problem in the neural network. In this case, the neural network starts overfitting on the training data giving higher accuracies while the accuracy for validation and testing data start to drop. We utilize several techniques to stop overfitting such as pooling and dropping layers. In addition, we use the aforementioned early stopping technique in conjunction with pooling and dropping methods to achieve higher testing accuracy. Figure 4 shows the loss function values associated with the number of iterations. Figure 4 relates to the highest classification accuracy in 1-layer CNN. This figure explains that with higher number of iterations the loss

Table 3: Confusion Matrix for 1-Layer CNN

Class No.	0	1	2	3	4	5	6	Classification Accuracy (%)
0	39	0	0	0	0	0	0	100.00
1	0	51	0	0	1	0	0	98.07
2	0	0	121	10	2	0	2	89.62
3	0	0	13	131	1	0	1	89.72
4	0	0	0	0	153	0	6	96.22
5	0	0	0	0	0	169	0	100.00
6	0	2	5	11	14	0	205	86.49
<b>Overall Classification Accuracy</b>							<b>92.74 %</b>	

Table 4: Classification Accuracy for 2-Layers CNN

No of Filters	Reception Field	Accuracy (%)
8	2*2	90.82
8	3*3	91.14
8	4*4	90.92
8	5*5	90.82
16	2*2	90.60
16	3*3	89.75
16	4*4	89.96
16	5*5	91.88
32	2*2	92.52
32	3*3	92.31
32	4*4	93.38
32	5*5	<b>94.26</b>

function for the training continues to drop while the loss function for validation keeps on increasing.

Figure 5 shows randomly chosen examples of correctly classified types with the 1-layer CNN. Those examples include all the seven plankton types. The confusion matrix for the 1-layer CNN based on 32 filters and 5\*5 reception field for one particular cross fold with accuracy rate of 92.74% is shown in Table 3. We perform 3 cross validation for 1-layer CNN with 32 filters and 5\*5 reception field to get the average accuracy rate of 92.39% as show in Table 2.

#### 4.2 Phase two: 2-layer CNN

In this phase our focus is to add another convolutional layer. Depending upon the results we got from phase one, we chose the configuration with the best accuracy rate to be the setting for the first convolutional

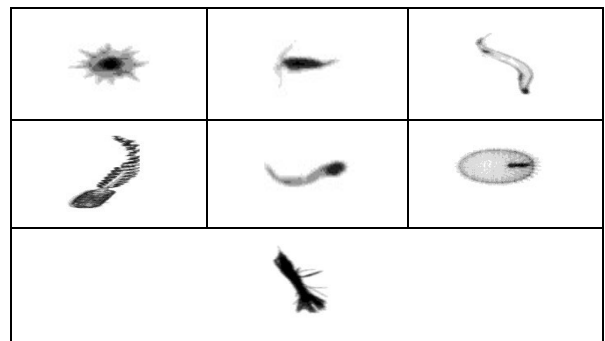


Figure 6. Correctly classified example images from 2-layers CNN, from left to right and top to bottom the images are from class number 0-6.

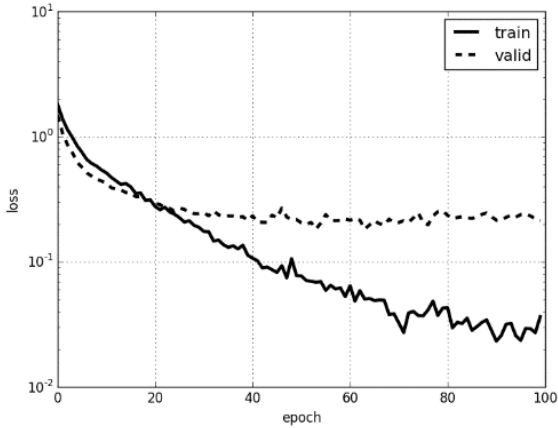


Figure 7. Training and validation loss for the highest accuracy in 2-layer CNN.

layer. In phase 2 we tune the second convolutional layer with different reception field and with different number of filters.

As shown in Table 4, overall we got better results with the 2-layers CNN in comparison with 1-layer CNN. Figure 6 shows randomly chosen examples of correctly classified types with the 2-layers CNN. Those examples include all the seven plankton types. The confusion matrix for the 2-layer CNN based on 32 filters and 5\*5 reception field for one particular cross fold with accuracy rate of 94.55% is shown in Table 5. We perform 3 cross validation for 2-layers CNN with 32 filters and 5\*5 reception field to get the average accuracy rate of 94.26% as show in Table 4.

Figure 7 relates to the highest classification accuracy in 2-layers CNN. This figure explains that with higher number of iterations the loss function for the training continues to drop while the loss function for validation keeps on increasing. It also suggests that the 2-layers CNN requires more training epochs to start converging than what is required by 1-layer CNN.

Table 5: Confusion Matrix for 2-Layers CNN

Class No.	0	1	2	3	4	5	6	Classification Accuracy (%)
0	39	0	0	0	0	0	0	100.00
1	0	51	0	0	1	0	0	98.07
2	0	0	125	9	1	0	0	92.59
3	0	0	15	129	1	0	1	88.35
4	0	1	1	0	155	0	2	97.48
5	0	0	0	0	0	168	1	99.40
6	0	3	2	6	7	0	219	92.40
<b>Overall Classification Accuracy (%)</b>								<b>94.55 %</b>

Table 6: Comparison of the Classification Performance with other Methods on the SIPPER dataset

Method	Accuracy (%)
Normalized Multilevel Dominant Eigenvector Estimation [6]	91.70
Bagging Based [7]	93.04
Random Subspace [8]	93.27
<b>Our proposed 2-layers CNN method</b>	<b>94.26</b>

Table 6 gives a comparison of the classification performance with other methods on the SIPPER dataset. We obtain classification performance of 94.26%, which is better than other previous methods.

## 5 Conclusions and future work

The study of plankton distribution is an important tool used for assessing the changes to marine ecosystem. Efficient analysis and classification of huge amounts of plankton data requires robust algorithms. Traditional computer vision techniques require tedious work to find suitable features to represent plankton. In our paper, we proposed the use of convolutional neural networks. Results of the experiments using the SIPPER dataset show improvement in classification accuracy in comparison to the previous approaches from other research groups. Another major advantage of our approach is the scalability for classification of new classes without the need for feature engineering.

In the future we plan to further improve the performance of our method by expanding the number of layers in the convolutional neural network. We also plan to explore the combination of convolutional neural network and other classification algorithms such as SVM or Random Forest to improve the overall efficiency of the classification methodology.

## Acknowledgment

We would like to thank Dr. Kendra L. Daly, Andrew Remsen, and Kurt Kramer (USF) for the validated SIPPER image dataset. The SIPPER imaging work was supported by a National Science Foundation grant OCE-0526545, a University of South Florida Sponsored Research Foundation grant, a Florida Institute of

Oceanography/BP grant, and a Gulf of Mexico Research Initiative (GOMRI) grant to K.L. Daly.

## 6 References

- [1] J. Watson, G. Craig, V. Chalvidan, J. P. Chambard, A. Diard, G. L. Foresti, B. Forre, S. Gentili, P. R. Hobson, R. S. Lampitt, P. Maine, J. T. Malmo, H. Nareid, G. Pieroni, S. Serpico, K. Tipping, and A. Trucco, "High resolution in situ holographic recording and analysis of marine organisms and particles (Holomar)," in Proc. IEEE Int. Conf. OCEANS, 1998, pp. 1599–1604.
- [2] S. Davis, S. M. Gallager, and A. R. Solow, "Microaggregations of oceanic plankton observed by towed video microscopy," *Science*, vol. 257, pp. 230–232, Jul. 1992.
- [3] S. Samson, T. Hopkins, A. Remsen, L. Langebrake, T. Sutton, and J. Patten, "A system for high-resolution zooplankton imaging," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 671–676, Oct. 2001.
- [4] X. Tang, W. K. Stewart, L. Vincent, H. Huang, M. Marty, S. M. Gallager, and C. S. Davis, "Automatic plankton image recognition," *Artif. Intell. Rev.*, vol. 12, pp. 177–199, 1998.
- [5] T. Luo, K. Kramer, S. Samson, A. Remsen, D. B. Goldgof, L. O. Hall, and T. Hopkins, (2004, August). "Active learning to recognize multiple types of plankton". In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (Vol. 3, pp. 478-481). IEEE.
- [6] X. Tang, F. Lin, S. Samson, and A. Remsen, "Binary plankton image classification," *IEEE J. Ocean. Eng.*, vol. 31, no. 3, pp. 728–735, Jul. 2006.
- [7] F. Zhao, F. Lin, and H. Seah, "Bagging based plankton image classification," in *Proc. IEEE Int. Conf. Image Process.*, 2009, pp. 2517–2520.
- [8] L. Zhifeng, Z. Feng, L. Jianzhuang, Q. Yu, "Pairwise Nonparametric Discriminant Analysis for Binary Plankton Image Recognition," *IEEE J. Ocean. Eng.*, vol. 39, no. 4, pp. 695–701, 2014.
- [9] Y. LeCun, K. Kavukcuoglu, and C. Farabet, (2010, May). Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on* (pp. 253-256). IEEE.
- [10] D. Hubel, and T. Wiesel, (1968). Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195, 215–243.
- [11] K. Fukushima, (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193–202.
- [12] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- [13] A. Krizhevsky, I. Sutskever and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, *Proc. Neural Information and Processing Systems*, 2012.
- [14] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. 27th International Conference on Machine Learning*, 2010.
- [15] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [16] J. Van, "Analysis of Deep Convolutional Neural Network Architectures". 2014. Retrieved from <http://referaat.cs.utwente.nl/conference/21/paper/7438/analysis-of-deep-convolutional-neural-network-architectures.pdf>.
- [17] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 111–118, 2010.
- [18] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.
- [19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, pages 1929–1958, 2014.
- [20] L. Prechelt, "Automatic early stopping using cross validation: quantifying the criteria", *Neural Netw.*, 11 (1998), pp. 761–767
- [21] L. Prechelt, (1998). "Early stopping-but when?". In *Neural Networks: Tricks of the trade* (pp. 55-69). Springer Berlin Heidelberg.