BoW and CNN for Image Classification

Juan Buhagiar
juan.buhagiar@student.uva.nl
University of Amsterdam
11576014

Sarantos Tzortzis
sarantos_tzortzis@icloud.com
University of Amsterdam
11863331

https://github.com/sarantinio/BoW-and-CNN-for-Image-Classification.git
March 27th, 2018

1 Introduction

For the first part of the final project, we have designed a bag of words (*BoW*) approach to classify images into 4 categories: Airplanes, Faces, Motorcycles and cars. For the second part the goal is remains the same but we have to appply Convolutional Neural Network (*CNN*)

2 Part 1: BoW

The Bag of Words approach [1] is a popular classification method that was popular before the ground breaking research surrounding deep learning [2]. The BoW approach has many variables that can effect it's performance such as vocabulary size, proportion of images used to create the vocabulary and the classification method used.

2.1 Experiments

We split the experiments into 4 different sections Vocabulary, Classification Methods, Descriptors and Training Samples.

2.1.1 Vocabulary

To understand the relationship between the vocabulary size and the power of the classifier to discriminate between classes, we designed multiple experiments with varying vocabulary sizes. We tested the BoW approach with a vocabulary size of 150, 400, 800, 1600, 2000, 4000. Figure 1 & Table 1 show the MAP with varying vocabulary sizes. We can see a up-words trend between the MAP and vocabulary size. Unfortunately, as the size of the vocabulary increases the processing time also increases. Therefore, we have picked a vocabulary size of 800 to use in other experiments as this, we think is a good trade-off between computational time and MAP.

Table 1: MAP with Varying Vocabulary Size using the SIFT Descriptor

Vocabulary Size	**150**	**400**	**800**	**1600**	**2000**	**4000**
MAP for SIFT	0.68093	0.7325525	0.77682	0.769815	0.7736625	0.825385

To improve on our previous vocabulary selection, we looked at cluster validity indexes. The visual vocabulary, is just clusters centroids from a training set, therefore with cluster validity indexes we are able to identify the best vocabulary size. We use the dunn's index [3] to enumerate how well the clusters are created. The aim of this metric is to identify sets of clusters that are compact, with small variance between data in clusters and separated well, meaning that clusters are as far away as possible. The higher the Dunn's Index the better the clustering. We test vocabulary sizes between 600

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

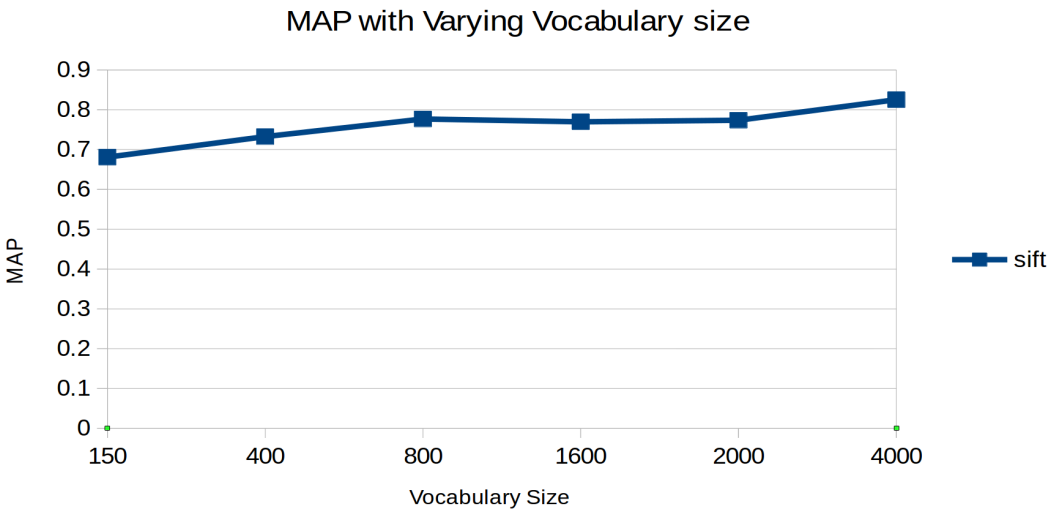

Figure 1: MAP with Varying Vocabulary Size using the SIFT Descriptor

to 1000 with intervals of 50. Figure 2 shows the results obtained with the best clustering occuring with a vocabulary size of 800 & 900.

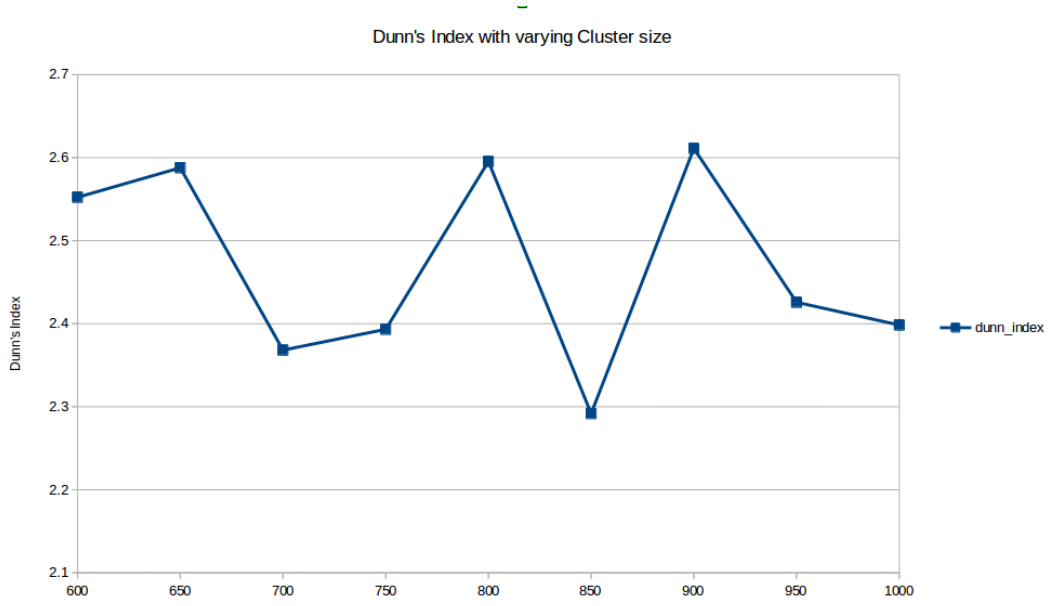

Figure 2: Caption

2.1.2 Classification Method

We are using an SVM classifier to classify images into the 4 categories. We have performed tests using linear kernel and a RBF kernel. Unfortunately, due to limited computational possibilities we were not able to tune hyper-paratmeters. Nonetheless the results are shown in Figure x. All the tests in this section have a 10% split for training images to create the visual vocabulary and a vocabulary size of 800. We can see that using the default hyper-parameters the Radial Basis Function (RBF) kernel obtains a better classification rate then the linear kernel.

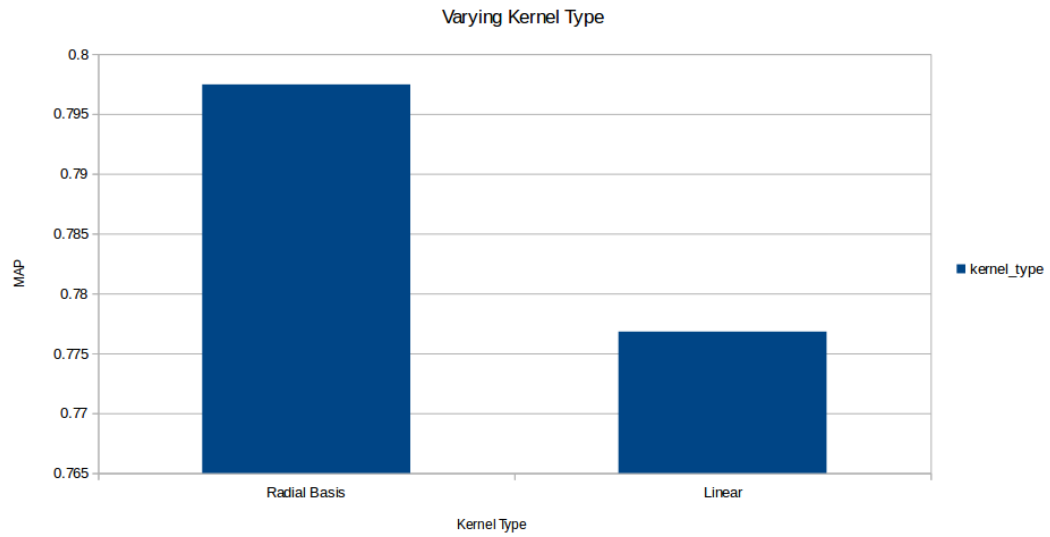

Figure 3: Varying SVM Kernel Type

2.1.3 Descriptors

We investigate different SIFT Descriptors types starting with the conventional SIFT, Dense SIFT, rgbSIFT, RGBSIFT and OPPSIFT. All the tests in this section have a 10% split for training images to create the visual vocabulary and a vocabulary size of 800. The results obtained on these different descriptors is shown in Figure 4 and Table 2. We can see that the OPPSIFT and RGBSIFT offer the most discriminate power. Although we are not able to see any difference between the two methods they obtain relatively high results. We can see that the SIFT descriptor with only key-point detection obtains the worse results. This is understandable as the other SIFT Methods have more data such as color.

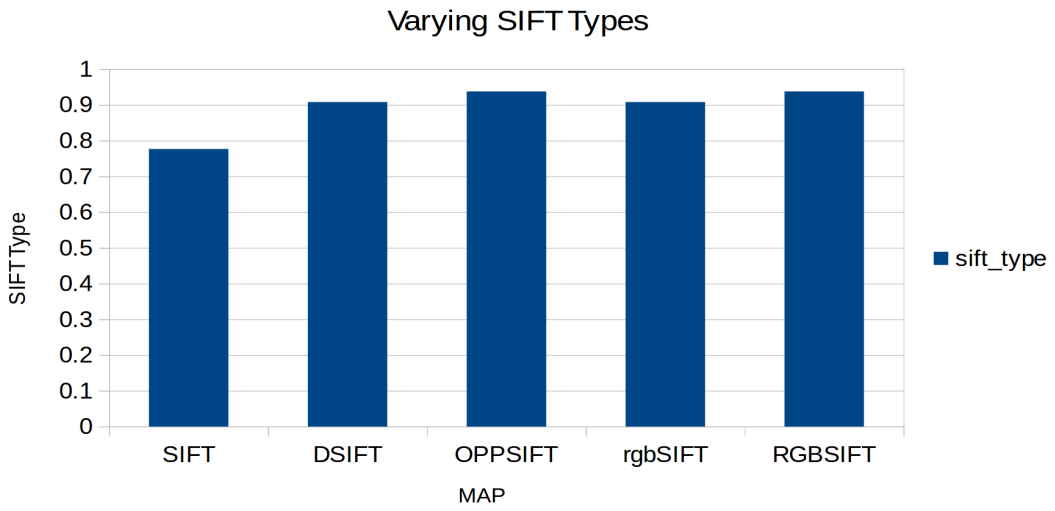

Figure 4: MAP with Varying SIFT Descriptors

Table 2: MAP with Varying SIFT Descriptors

SIFT Type	SIFT	DSIFT	OPPSIFT	rgbSIFT	RGBSIFT
MAP	0.77682	0.9080925	0.9378275	0.9080925	0.9378275

2.1.4 Training Samples

The last parameter to be tuned is the number of training samples given to the clustering algorithm. This parameter effects the creation of the vocabulary. There is a clear trade-off here between the number of samples given and the performance of the clustering algorithm. Experiments in this section have been performed with 800 as the vocabulary size and a linear kernel for the SVM. We have tested three different percentages of training samples given to the cluster 10%, 30% and 50%. Figure 5 shows the results obtained where 30% seems to be the optimal value.

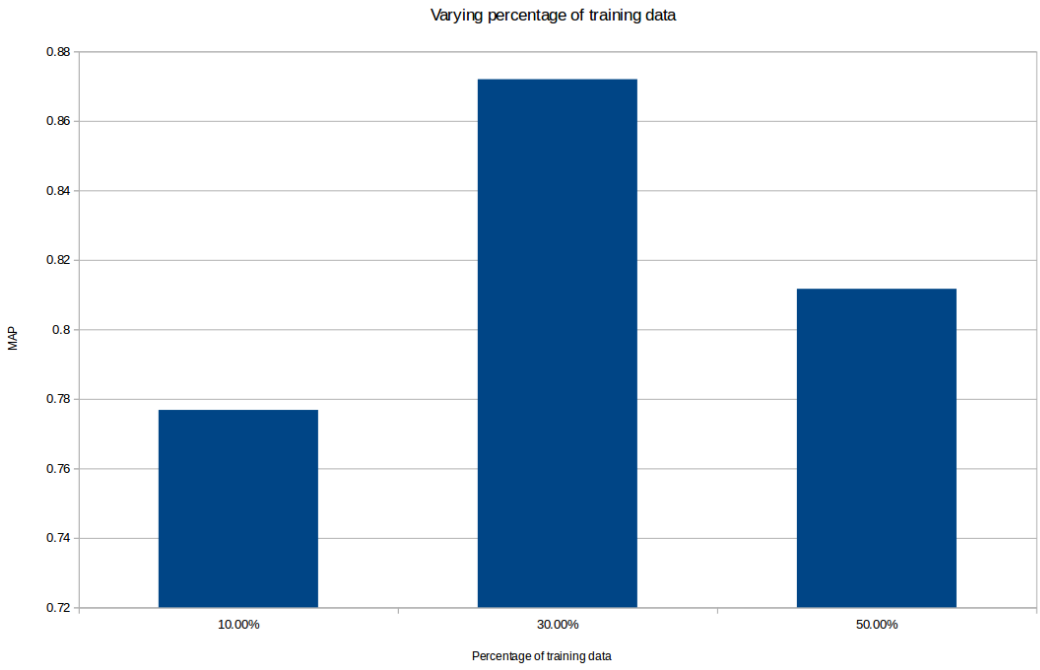

Figure 5: Varying Percentage of Training data

3 Part 2: CNN

For this part we apply Convolutional Neural Networks. This architecture make use of multilayer perceptrons. This project provide us with an already trained for classification network. The following figure indicates the structure of this network.

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13
type	input	conv	mpool	relu	conv	relu	apool	conv	relu	apool	conv	relu	conv	softmxl
name	n/a	layer1	layer2	layer3	layer4	layer5	layer6	layer7	layer8	layer9	layer10	layer11	layer12	layer13
support	n/a	5	3	1	5	1	3	5	1	3	4	1	1	1
filt dim	n/a	3	n/a	n/a	32	n/a	n/a	32	n/a	n/a	64	n/a	64	n/a
filt dilat	n/a	1	n/a	n/a	1	n/a	n/a	1	n/a	n/a	1	n/a	1	n/a
num filts	n/a	32	n/a	n/a	32	n/a	n/a	64	n/a	n/a	64	n/a	4	n/a
stride	n/a	1	2	1	1	1	2	1	1	2	1	1	1	1
pad	n/a	2	0x1x0x1	0	2	0	0x1x0x1	2	0	0x1x0x1	0	0	0	0
rf size	n/a	5	7	7	15	15	19	35	35	43	67	67	67	67
rf offset	n/a	1	2	2	2	2	4	4	4	8	20	20	20	20
rf stride	n/a	1	2	2	2	2	4	4	4	8	8	8	8	8
data size	32	32	16	16	16	16	8	8	8	4	1	1	1	1
data depth	3	32	32	32	32	32	32	64	64	64	64	64	4	1
data num	50	50	50	50	50	50	50	50	50	50	50	50	50	1
data mem	600KB	6MB	2MB	2MB	2MB	2MB	400KB	800KB	800KB	200KB	12KB	12KB	800B	4B
param mem	n/a	10KB	0B	0B	100KB	0B	0B	200KB	0B	0B	256KB	0B	1KB	0B

Figure 6: Architecture of the Pretrained Network

Observing the network can conclude to some patterns. It is consisted of repeating convolution layers for image filtering. In the intermediate layers, max/avg pooling is applied in order to reduce the dimension of the image.There are also some relu layers as activation function.
Taking this pre-trained network saves us time and expenses as we do not create this from scratch but with *transfer learning* we continue the training in order to achieve the fined- tuned network.

4 Experiments and Results

In order to find the best hyperparameters we run 6 experiments changing the *batch size* and the *number of epochs*.

At the Table 3 we see that the best tuned was found for Batch Size 100 and 80 Epochs(120 epochs as well, so we keep the least).

Table 3: Hyperparameter Tuning

Batch Size	Epochs	FineTuned CNN	PreTrained SVM	FineTuned SVM
50	40	99.0	89.0	98.0
50	80	99.0	89.0	98.5
50	120	99.0	89.5	99.0
100	40	99.0	89.5	99.0
100	**80**	**99.0**	**89.5**	**99.5**
100	120	99.0	89.5	99.5

Once we obtained the best tuning of the hyperparameters, we save this setting to use for training and evaluation.

The results obtained from the training show the objective error the top1 error and the top5 error in Figure 3 respectively .

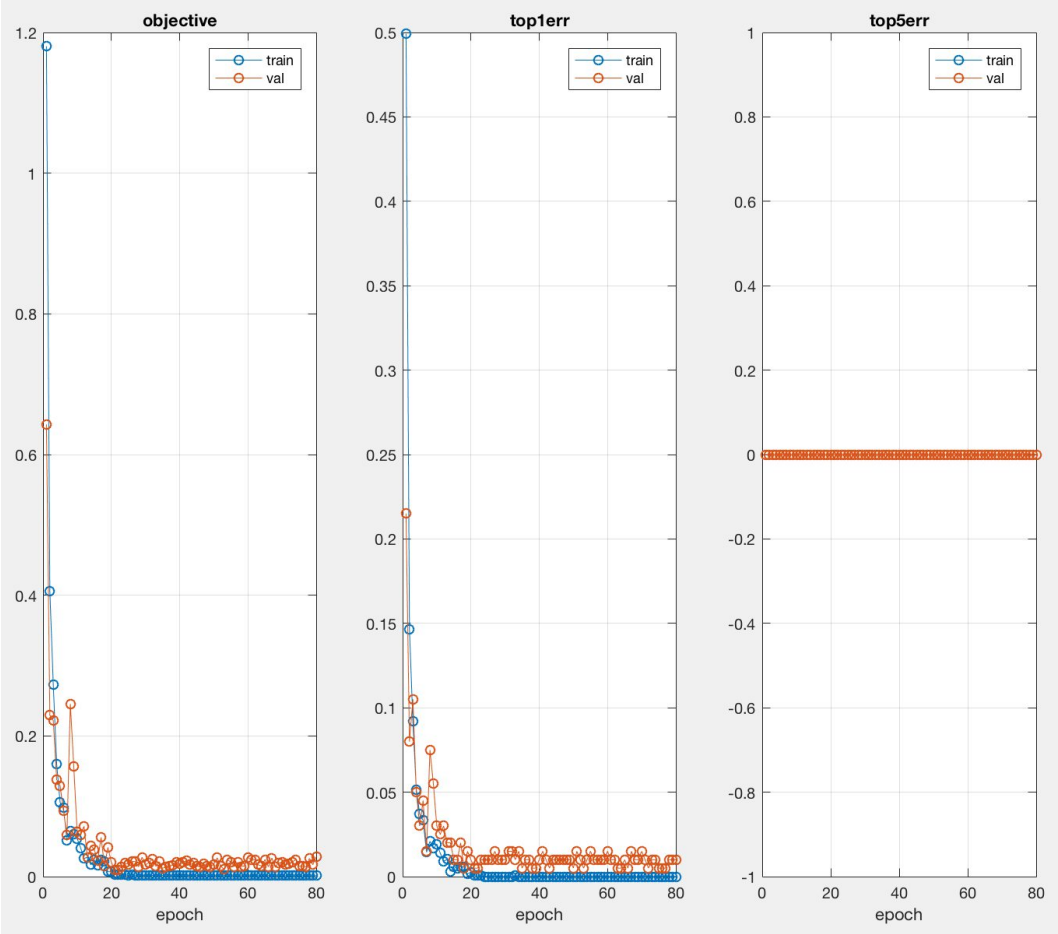

Figure 7: Training Results

Both the objective and the top1 error converge to zero as it is supposed to. The performance of the validation error is slightly worse. The top5 error is always zero as we only have 4 classes.(This network was trained for 10 classes)

After we have extracted the features of the pretrained and finetuned network, we apply dimensionality reduction with the help of TSNE. The Figure 8 and 8 show classification of the features for the pretrained and finetuned network respectively.

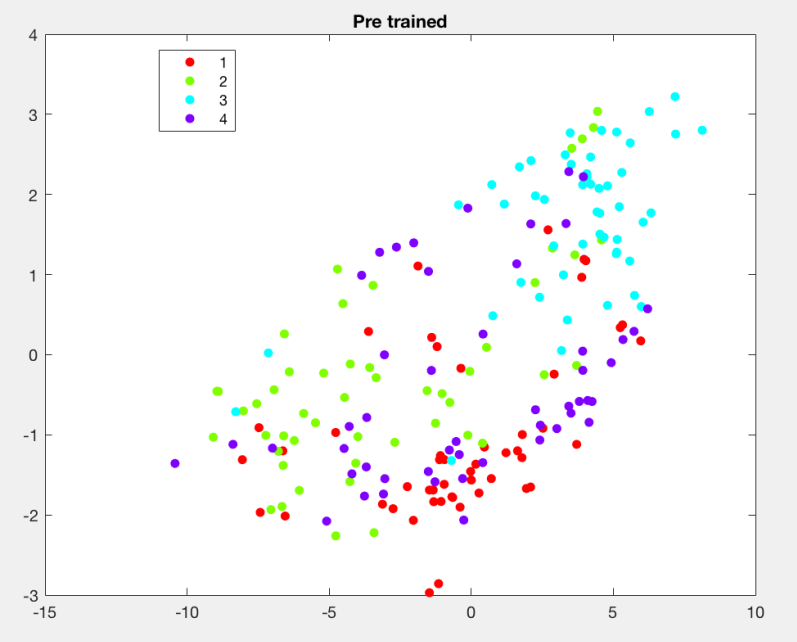

Figure 8: Pre-trained

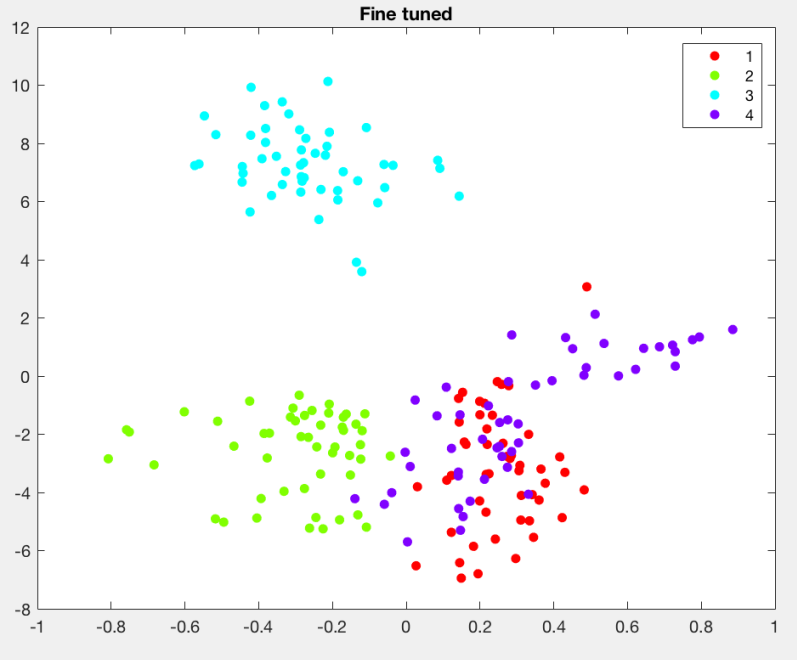

Figure 9: Fine-tuned

As expected, we observe that the fine tuned network made a better separation as it was trained upon these 4 classes. At both networks it is also noticeable that same objects(classified) tend to be clustered together.

5 Conclusion

We managed to score better accuracy (99.5) with the CNN model compared to the BoW one. Possibly though, four classes prediction is a simple task and further work should be made.

References

[1] Yang, Jun, et al. "Evaluating bag-of-visual-words representations in scene classification." Proceedings of the international workshop on Workshop on multimedia information retrieval. ACM, 2007.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[3] Dunn, Joseph C. "Well-separated clusters and optimal fuzzy partitions." Journal of cybernetics 4.1 (1974): 95-104.