

CODECHECK certificate 2020-002

<http://doi.org/10.5281/zenodo.3750741>





Item	Value
Title	The principal components of natural images
Authors	Peter J. B. Hancock, Roland J. Baddeley, Leslie S. Smith
Reference	Network (1992) 3:61-70 http://pdfs.semanticscholar.org/7dcf/a42cfe3b59becb441844b72558b361693608.pdf
Codechecker	Stephen J. Eglén  Daniel Nüst 
Date of check	2020-04-13 10:00:00
Summary	Matlab code written by Iain Davies to reproduce original paper; natural images provided by Peter Hancock.
Repository	https://github.com/codecheckers/Reproduction-Hancock

Table 1: CODECHECK summary

output	comment	size
Figure2.png	manuscript Figure 2	41513
Figure3.png	manuscript Figure 3	41489
Figure4.png	manuscript Figure 4	46935
Figure5.png	manuscript Figure 5	33332
Figure6.png	manuscript Figure 6	63185
Figure7.png	manuscript Figure 7	71145
Figure8.png	manuscript Figure 8	293243

Table 2: Summary of output files generated

Summary

This code was straightforward to codecheck. The code came from Iain Davies, a Cambridge mathematics student, who worked on reimplementing the Hancock et al paper. I asked him to ensure that the code for each figure could be re-run to generate a pdf.

CODECHECKER notes

The github repo <https://github.com/IainDaviesMaths/Reproduction-Hancock> contained all the necessary code. The code was written in Matlab.

Running the software to regenerate outputs.

The root Makefile contained targets to regenerate all of the figures using:

```
make -j7 all
```

This took 7m 10s to complete on a large workstation.

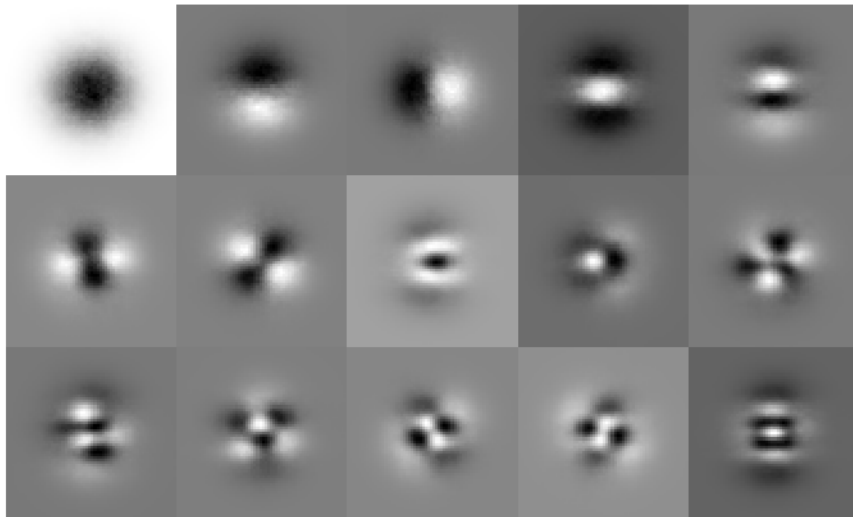


Figure C1: manuscript Figure 2

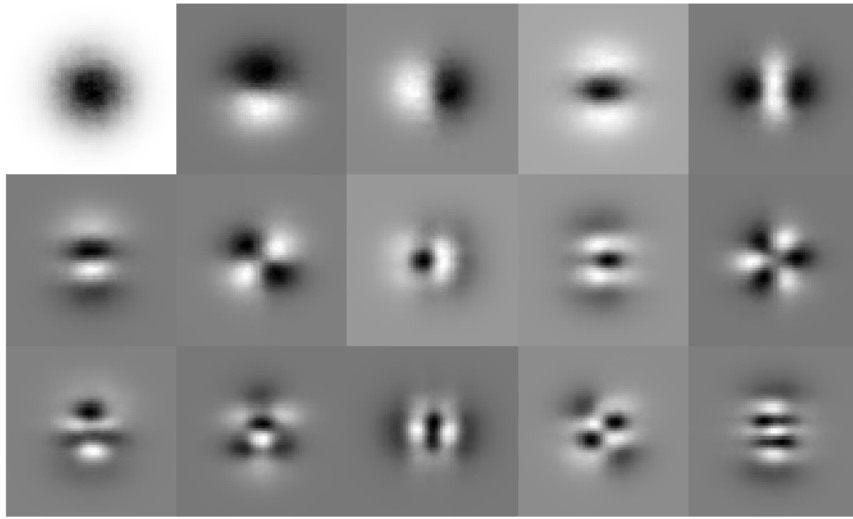


Figure C2: manuscript Figure 3

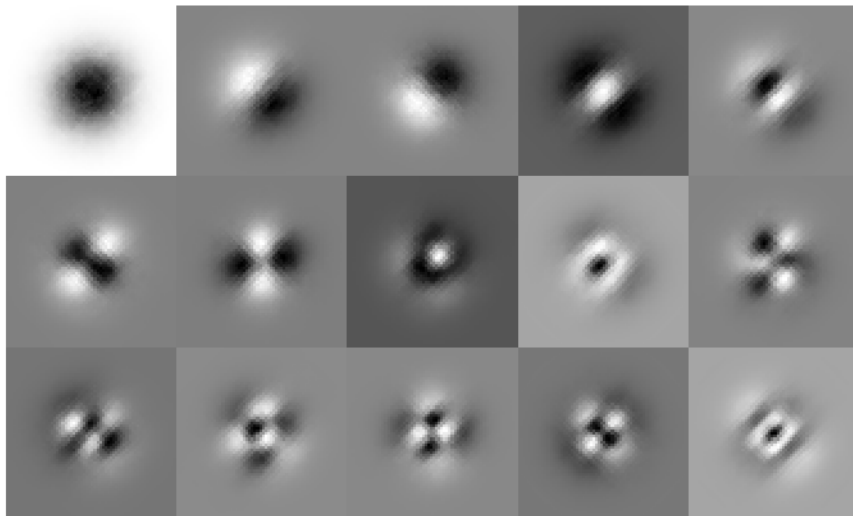


Figure C3: manuscript Figure 4

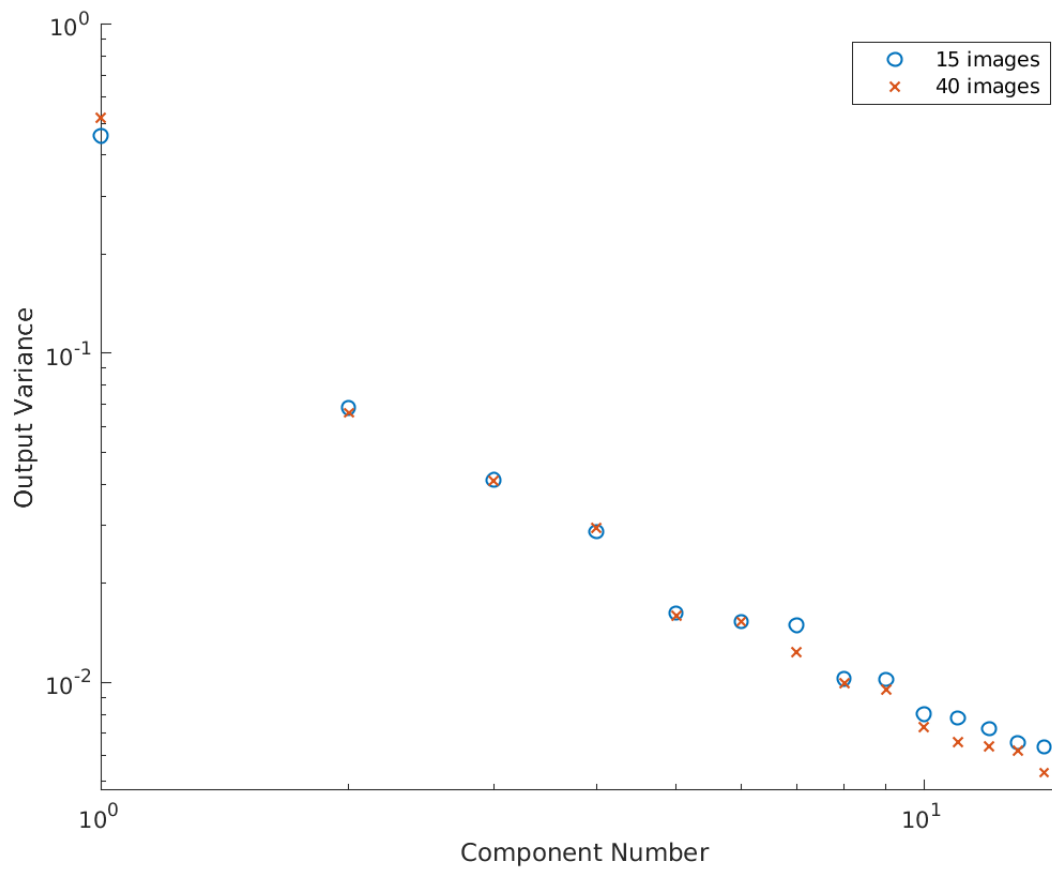


Figure C4: manuscript Figure 5

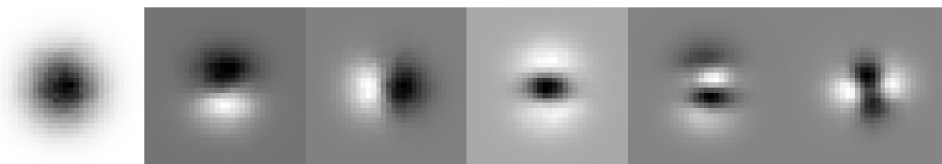
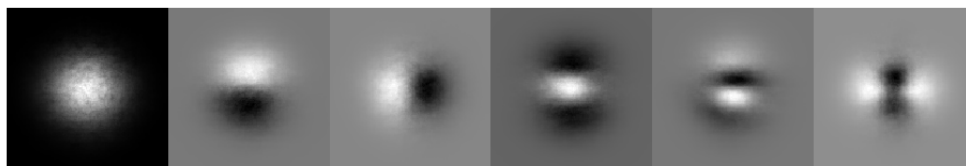


Figure C5: manuscript Figure 6

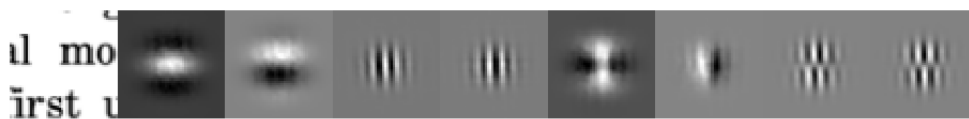
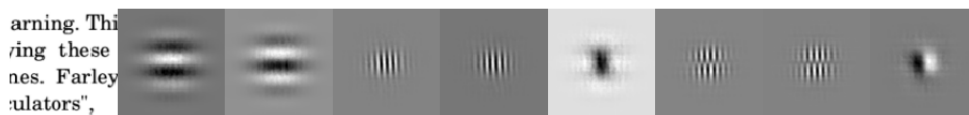
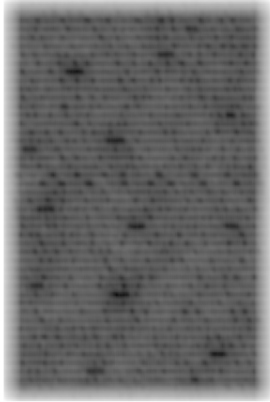


Figure C6: manuscript Figure 7

Text Image

Artificial neural networks (ANNs) or connectionist systems are computing systems that are inspired by, but not necessarily identical to, the biological neural networks that constitute animal brains. Such systems learn to perform tasks by considering examples, generally without being programmed with any task-specific knowledge. For example, in image recognition, they might learn to identify images that contain faces by analyzing example images that have been manually labeled as "cat" or "dog" and using the results of this analysis to identify new images. They do this without any prior knowledge about what they are doing. Like the human brain, they automatically process knowledge represented from the learning material that they process. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each neuron, like the neuron in a biological brain, can transmit a signal from one artificial neuron to another, an artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. To mirror a human brain's implementation, the output of each artificial neuron is computed by some nonlinear function of the sum of its inputs. The connectionist neural network is an artificial neural network and often together have a target that allows us to learn from examples. The simple neurons or elements are the nodes of the network. Artificial neurons may have a dendritic tree that the signal is only sent if the dendritic signal crosses the threshold. Typically, artificial neurons are organized into layers. Different layers may perform different kinds of computations on their inputs. Signals travel from the input layer, to the hidden layer, to the output layer, possibly also traversing the layers multiple times. The signal at each of the ANN's layers may be subject to some processing in the same way that human brain would. However, over time, signals may be refined or filtered through tasks, leading to decisions from biology. Artificial neural networks have been used in a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis. In the late 1950s, D. O. Hebb first created a learning hypothesis based on the mechanism of neural plasticity that became known as Hebbian learning. Hebbian learning is unsupervised learning. This evolved into models for long-term potentiation. Researchers started applying these ideas to computational models in 1958 with Turing's "B-type machines". Early and Chaffin's "Hebb" first used computational models that called "neural networks". In 1959, a Hebbian network model was used in a neural network computational model was used by Hebbian, Hebb, and Duda (1959). In 1961, Hebbian (1961) created the perceptron, an algorithm for pattern recognition. With subsequent neural networks, supervised learning and of the basic properties, such as the activation function that could not be processed by neural networks at the time (1). In 1968, a Hebbian network proposed by Hebbian network (Hebb and Duda) was used on their discovery of the types of cells in the primary visual cortex (area 17) and compare with (2). The first neural network with name layers was published by P. Hebbian and Duda in 1961, entitled the Group Method of Data Handling (GMDH) (3). Several neural networks designed after machine learning research by Hebbian and Duda (1961) (4). Hebbian network was later used with the unsupervised machine learning neural networks. The first was the first neural network was unsupervised ANN proposed to solve problems in the

Convolution



Superimposition

Artificial neural networks (ANNs) or connectionist systems are computing systems that are inspired by, but not necessarily identical to, the biological neural networks that constitute animal brains. Such systems learn to perform tasks by considering examples, generally without being programmed with any task-specific knowledge. For example, in image recognition, they might learn to identify images that contain faces by analyzing example images that have been manually labeled as "cat" or "dog" and using the results of this analysis to identify new images. They do this without any prior knowledge about what they are doing. Like the human brain, they automatically process knowledge represented from the learning material that they process. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each neuron, like the neuron in a biological brain, can transmit a signal from one artificial neuron to another, an artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. To mirror a human brain's implementation, the output of each artificial neuron is computed by some nonlinear function of the sum of its inputs. The connectionist neural network is an artificial neural network and often together have a target that allows us to learn from examples. The simple neurons or elements are the nodes of the network. Artificial neurons may have a dendritic tree that the signal is only sent if the dendritic signal crosses the threshold. Typically, artificial neurons are organized into layers. Different layers may perform different kinds of computations on their inputs. Signals travel from the input layer, to the hidden layer, to the output layer, possibly also traversing the layers multiple times. The signal at each of the ANN's layers may be subject to some processing in the same way that human brain would. However, over time, signals may be refined or filtered through tasks, leading to decisions from biology. Artificial neural networks have been used in a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis. In the late 1950s, D. O. Hebb first created a learning hypothesis based on the mechanism of neural plasticity that became known as Hebbian learning. Hebbian learning is unsupervised learning. This evolved into models for long-term potentiation. Researchers started applying these ideas to computational models in 1958 with Turing's "B-type machines". Early and Chaffin's "Hebb" first used computational models that called "neural networks". In 1959, a Hebbian network model was used in a neural network computational model was used by Hebbian, Hebb, and Duda (1959). In 1961, Hebbian (1961) created the perceptron, an algorithm for pattern recognition. With subsequent neural networks, supervised learning and of the basic properties, such as the activation function that could not be processed by neural networks at the time (1). In 1968, a Hebbian network proposed by Hebbian network (Hebb and Duda) was used on their discovery of the types of cells in the primary visual cortex (area 17) and compare with (2). The first neural network with name layers was published by P. Hebbian and Duda in 1961, entitled the Group Method of Data Handling (GMDH) (3). Several neural networks designed after machine learning research by Hebbian and Duda (1961) (4). Hebbian network was later used with the unsupervised machine learning neural networks. The first was the first neural network was unsupervised ANN proposed to solve problems in the

Figure C7: manuscript Figure 8

About this document

This document was created using Rmarkdown. make `codecheck.pdf` will regenerate the file.

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Manjaro Linux
##
## Matrix products: default
## BLAS: /usr/lib/libopenblas-r0.3.9.so
## LAPACK: /usr/lib/liblapack.so.3.9.0
##
## locale:
## [1] LC_CTYPE=en_GB.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_GB.UTF-8      LC_COLLATE=en_GB.UTF-8
## [5] LC_MONETARY=en_GB.UTF-8  LC_MESSAGES=en_GB.UTF-8
## [7] LC_PAPER=en_GB.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets
## [6] methods   base
##
## other attached packages:
## [1] readr_1.3.1      tibble_2.1.3
## [3] xtable_1.8-4     yaml_2.2.1
## [5] rprojroot_1.3-2  knitr_1.26
## [7] codecheck_0.0.0.9000 zen4R_0.3-1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.3      xml2_1.3.1      magrittr_1.5
## [4] hms_0.4.2       rvest_0.3.5     R6_2.4.1
## [7] rlang_0.4.2     highr_0.8       stringr_1.4.0
## [10] httr_1.4.1      tools_3.6.3     xfun_0.11
## [13] htmltools_0.4.0 digest_0.6.23   crayon_1.3.4
## [16] evaluate_0.14   rmarkdown_1.18 stringi_1.4.6
## [19] compiler_3.6.3 pillar_1.4.1    backports_1.1.6
## [22] jsonlite_1.6.1  pkgconfig_2.0.2
```