

# Stellar Image Interpretation System using Artificial Neural Networks: Unipolar Function Case

F. I. Younis, A. El-Bassuny Alawy, B. Šimák, M. S. Ella, M. A. Madkour

*An artificial neural network based system for interpreting astronomical images has been developed. The system is based on feed-forward Artificial Neural Networks (ANNs) with error back-propagation learning. Knowledge about images of stars, cosmic ray events and noise found in images is used to prepare two sets of input patterns to train and test our approach. The system has been developed and implemented to scan astronomical digital images in order to segregate stellar images from other entities. It has been coded in C language for users of personal computers. An astronomical image of a star cluster from other objects is undertaken as a test case. The obtained results are found to be in very good agreement with those derived from the DAOPHOTII package, which is widely used in the astronomical community. It is proved that our system is simpler, much faster and more reliable. Moreover, no prior knowledge, or initial data from the frame to be analysed is required.*

*Keywords: neural networks, knowledge-based system, stellar images, image processing.*

## 1 Introduction

During the last three decades progress has been made in the field of observational astronomy. First, sophisticated low light level detectors were invented, developed and used for direct imaging. Among these detectors, Charge Coupled Device (CCD) chips have become the dominant detector for various applications. This is because they possess high detective quantum efficiency, a linear response and a very wide dynamic range. Secondly, fast electronic (mainframe, personal and workstation) computing machines with huge memories have greatly supported astronomy not only in automating the observational techniques but also in data acquisition, reduction and analyses.

In the field of stellar astronomy, CCD detectors and electronic computers have played a vital role since numerous frames have been taken and analysed by various packages to identify stellar images and extract their astronomical parameters. This has been achieved through modelling a stellar image either empirically [1, 2, 3] or mathematically [4, 5, 6] or semi-empirically [7, 8]. All these approaches require user intervention to set initial values for the parameters of the adopted model as well as the form of the model itself. Moreover several non-linear function-fitting processes have to be performed for many parameters, which requires much computing time. In addition, in many cases false results have been obtained, identifying images as stellar where they are not, and vice versa.

This paper deals with the development of an Artificial Neural Network approach for stellar image recognition. It is structured as follows. Section 1 introduces the project. Section 2 states the problem and the objective. Section 3 describes the developed system, specifying all relevant concepts and the implementation details. In Section 4, a case study is selected and used to verify the applicability of the system. The results of this approach are reported and discussed, taking into account standard published approaches, and are compared with the results derived by one of the best known and more widely applied methods in astronomy. Some concluding remarks are presented in Section 5.

## 2 Statement of problem and objective

CCD detectors are capable of imaging a huge number of celestial objects (stars, galaxies, etc) on a single frame under the same conditions. In such frame, each given picture element (pixel) contains the data number representing linearly the detected photons coming from one or more of several sources: 1) seen object, 2) unseen object, 3) localised image defects, 4) cosmic ray events, and 5) diffuse sources, including but not necessarily limited, for example, to the terrestrial night sky and scattered light in the camera. Hence, an obtained frame may contain many objects, some of which are actually due to stars, while the others are not, though their data-distributions may be similar to data distributions of stellar origin. This may be due to some of the sources stated above.

The main objective of the present work is to develop a simple, fast and reliable stellar image identification method that should be effectively able to identify stellar images and to differentiate between them and other objects on CCD frames. The approach deemed here is based on Artificial Neural Network concepts.

## 3 The present method

An Artificial Neural Network (ANN) can be defined as a category of mathematical algorithms that produces solutions for various specific problems. ANNs, have been biologically inspired to emulate the neural networks found in living organisms. An extremely important feature of an ANN is its learning capability, which is very useful and powerful in a wide range of applications. Supervised learning can be advantageously used, since it is a straightforward task to prepare the necessary input/output patterns for network training. This justifies the selection of multi-layer feed-forward networks for the required classification task. The error back-propagation learning algorithm will be used in the present work to train the network.

In the present work we aim at investigating the possibility of establishing a Stellar Image Interpretation System using

Artificial Neural Networks (SIIS-ANN) as a classifying technique that decides whether a given pixel array in a CCD frame represents a star image or some other objects. This decision is based on the distribution pattern of the electron contents of the given pixel and its neighbouring pixels.

### 3.1 SIIS-ANN structure

The present SIIS-ANN consists of a two-layer feed-forward network, as shown in Fig. 1. The first (hidden) layer has 4 neurones, while the second (output) layer has 3 neurones. The total inputs ( $z_i$ ) to the network amount to 24. The dimension of the hidden layer weight matrix is  $v(4 \times 24)$ , while that of the output layer weight matrix is  $w(3 \times 4)$ . Note that  $v_{1,0}$ ,  $v_{2,0}$ ,  $v_{3,0}$ , and  $v_{4,0}$  are the biases of the neurones of the hidden layer, while  $w_{1,0}$ ,  $w_{2,0}$ , and  $w_{3,0}$  are the biases of the output layer neurones. A differentiating function has to be adopted for learning and discriminating purposes, on which the weights  $v$  and  $w$  are applied and by which the neurones work. We adopted one of the most common functions, namely the Sigmoid function, defined as:

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad 1 \geq f(x) \geq 0$$

where  $\lambda$  (steepness factor) is an arbitrary small positive value.

### 3.2 Input patterns

Two pattern sets were adopted, one for training and the other for testing the present SIIS-ANN. Each set contains 341 input patterns (119 stellar images, 111 cosmic ray events and 111 noises). Each pattern comprises the data of a  $5 \times 5$  – pixel array, where the central pixel represents the centre of its relevant identity. All values were normalised with respect to the central value. These patterns were randomised through each set. The chosen stellar input patterns have a wide range of brightness, while the cosmic ray event patterns resemble energies from low energy to high energy.

The input patterns were adopted as follows:

- 1. Mark the object of interest (star image, cosmic ray event, or noise) on some CCD frames.
- 2. Find the brightest pixel.
- 3. Construct a  $5 \times 5$  window with the brightest pixel at its centre.
- 4. Extract the values of the pixels corresponding to the constructed window.
- 5. Normalise the obtained values by dividing them by the value of the brightest pixel.

Figure 2 demonstrates the 24-network input values ( $z_i$ ), where  $I(x,y)$  is the value of the brightest central pixel. Examples of these input patterns are given below.

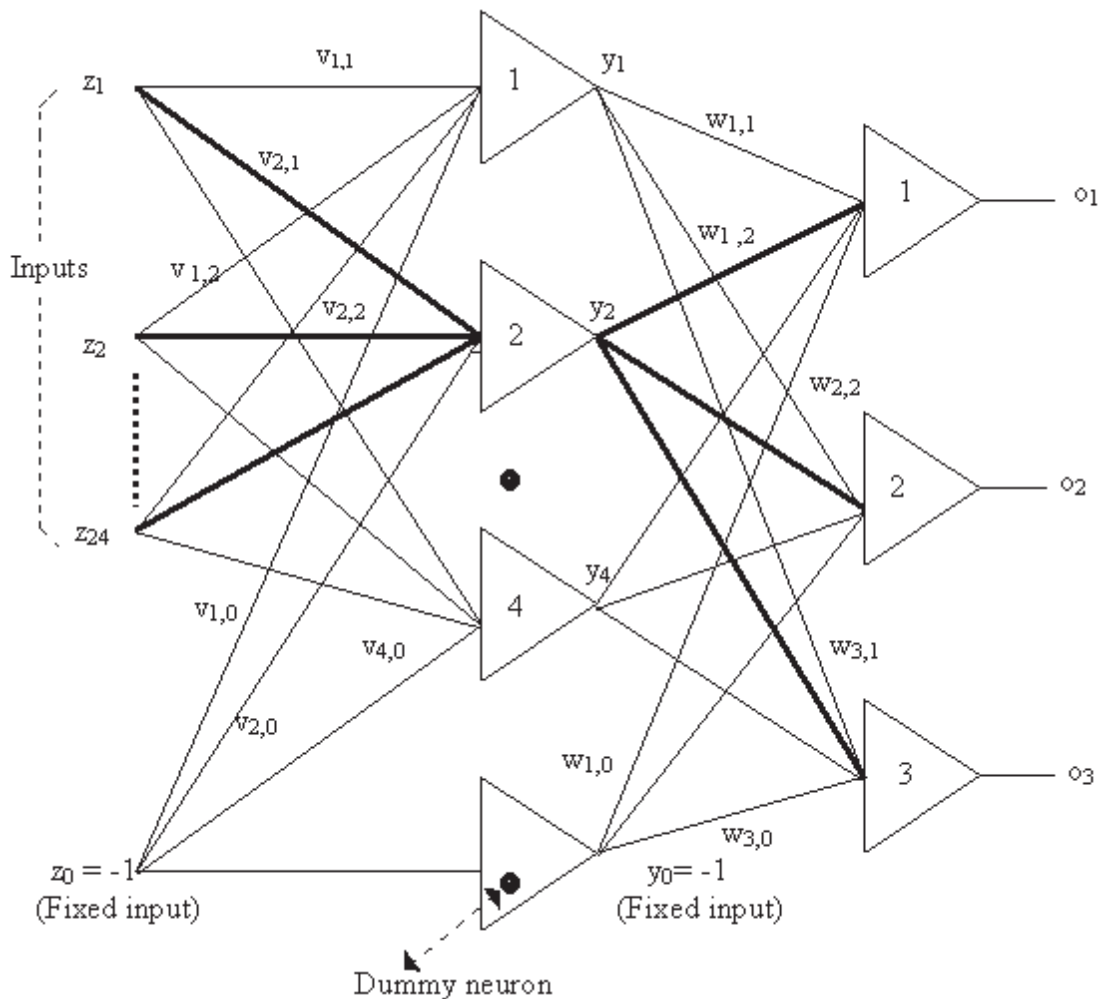


Fig. 1: The SIIS-ANN network architecture

$z_1 = I(x-2, y-2)/I(x, y)$	$z_2 = I(x-1, y-2)/I(x, y)$	$Z_3 = I(x, y-2)/I(x, y)$	$z_4 = I(x+1, y-2)/I(x, y)$	$z_5 = I(x+2, y-2)/I(x, y)$
$z_6 = I(x-2, y-1)/I(x, y)$	$z_7 = I(x-1, y-1)/I(x, y)$	$Z_8 = I(x, y-1)/I(x, y)$	$z_9 = I(x+1, y-1)/I(x, y)$	$z_{10} = I(x+2, y-1)/I(x, y)$
$z_{11} = I(x-2, y)/I(x, y)$	$Z_{12} = I(x-1, y)/I(x, y)$	$I(x, y)$	$z_{13} = I(x+1, y)/I(x, y)$	$z_{14} = I(x+2, y)/I(x, y)$
$z_{15} = I(x-2, y+1)/I(x, y)$	$Z_{16} = I(x-1, y+1)/I(x, y)$	$Z_{17} = I(x, y+1)/I(x, y)$	$z_{18} = I(x+1, y+1)/I(x, y)$	$z_{19} = I(x+2, y+1)/I(x, y)$
$z_{20} = I(x-2, y+2)/I(x, y)$	$Z_{21} = I(x-1, y+2)/I(x, y)$	$z_{22} = I(x, y+2)/I(x, y)$	$z_{23} = I(x+1, y+2)/I(x, y)$	$z_{24} = I(x+2, y+2)/I(x, y)$

Fig. 2: Diagram illustrating the input pattern values

### 3.2.1 Star input pattern:

Figures 3 and 4 represent the input patterns for a bright star and a faint star, respectively.

244	505	827	672	219
669	1974	3171	1737	406
1593	4499	6839	2950	520
2091	3825	5474	2256	459
1369	1569	1894	941	247

a)  $5 \times 5$  window pixel (Original values)

0.035678	0.073841	0.120924	0.098260	0.032022
0.097821	0.288639	0.463664	0.253985	0.059365
0.232929	0.657845	1	0.431350	0.076035
0.305746	0.559292	0.800409	0.329873	0.067115
0.200175	0.229419	0.276941	0.137593	0.036116

b)  $5 \times 5$  window pixel (Normalised values)

Fig. 3: Bright star input pattern

23	25	28	27	21
25	35	48	38	27
25	48	64	45	23
24	43	56	37	28
25	28	35	28	24

a)  $5 \times 5$  window pixel (Original values)

0.359375	0.390625	0.437500	0.421875	0.328125
0.390625	0.546875	0.750000	0.593750	0.421875
0.390625	0.750000	1	0.703125	0.359375
0.375000	0.671875	0.875000	0.578125	0.437500
0.390625	0.437500	0.546875	0.437500	0.375000

b)  $5 \times 5$  window pixel (Normalised values)

Fig. 4: Faint star Input pattern

### 3.2.2 Cosmic ray event input pattern:

Figures 5, 6 and 7 show examples of some of the input patterns for high, intermediate and low energy cosmic ray events, respectively.

21	21	22	25	19
25	21	25	20	24
22	22	201	85	22
22	19	21	24	23
23	22	21	19	24

a)  $5 \times 5$  window pixel (Original values)

0.104478	0.104478	0.109453	0.124378	0.094527
0.124378	0.104478	0.124378	0.099502	0.119403
0.109453	0.109453	1	0.422886	0.109453
0.109453	0.094527	0.104478	0.119403	0.114428
0.114428	0.109453	0.104478	0.094527	0.119403

b)  $5 \times 5$  window pixel (Normalised values)

Fig. 5: High-Energy cosmic ray input pattern

21	21	22	22	23
18	22	23	18	19
20	23	187	27	21
21	24	93	30	23
22	16	21	19	22

a)  $5 \times 5$  window pixel (Original values)

0.112299	0.112299	0.117647	0.117647	0.122995
0.096257	0.117647	0.122995	0.096257	0.101604
0.106952	0.122995	1	0.144385	0.112299
0.112299	0.128342	0.497326	0.160428	0.122995
0.117647	0.085561	0.112299	0.101604	0.117647

b)  $5 \times 5$  window pixel (Normalised values)

Fig. 6: Intermediate-Energy cosmic ray input pattern

19	25	23	20	21
22	22	22	21	21
21	20	61	19	21
21	22	22	22	25
21	24	20	21	21

a)  $5 \times 5$  window pixel (Original values)

0.311475	0.409836	0.377049	0.327869	0.344262
0.360656	0.360656	0.360656	0.344262	0.344262
0.344262	0.327869	1	0.311475	0.344262
0.344262	0.360656	0.360656	0.360656	0.409836
0.344262	0.393443	0.327869	0.344262	0.344262

b)  $5 \times 5$  window pixel (Normalised values)

Fig. 7: Low-Energy cosmic ray input pattern

### 3.2.3 Noise input pattern:

Figure 8 provides an example of the noise input pattern.

20	19	22	25	23
23	24	22	22	24
21	21	27	22	21
21	24	25	21	22
19	21	24	23	16

a)  $5 \times 5$  window pixel (Original values)

0.740741	0.703704	0.814815	0.925926	0.851852
0.851852	0.888889	0.814815	0.814815	0.888889
0.777778	0.777778	1	0.814815	0.777778
0.777778	0.888889	0.925926	0.777778	0.814815
0.703704	0.777778	0.888889	0.851852	0.592593

b)  $5 \times 5$  window pixel (Normalised values)

Fig. 8: Noise input pattern

### 3.3 Output patterns

The present SIIS-ANN was trained with all desired output values ( $o_i$ ) set to zero, except the value for the class of the input pattern, which was set to unity. Table (1) summarises the desired output values for a star, a cosmic ray, and noise.

Table 1: The desired output values for a star, a cosmic ray, and noise

	$O_1$	$O_2$	$O_3$
Star	1	0	0
Noise	0	1	0
Cosmic Ray	0	0	1

### 3.4 Training error

For the purpose of weight adjustment in each training step, the error to be reduced is usually that computed only for the pattern currently being undertaken. To assess the quality and success of the training, however, the joint errors must be computed for the entire batch of training patterns.

It should be pointed out that networks in classification applications may perform as excellent classifiers and may exhibit zero decision errors while still yielding substantial continuous response (cumulative and root mean square) errors. In this case, the decision error adequately reflects the accuracy of the neural network classifiers. This error was adopted for the network training.

While training the present SIIS-ANN, all the desired output values were set to the values listed in Table (1). In addition, the decision error was used to terminate the training process when it reached, practically, a zero value. Such an error is defined by:

$$E_d = \frac{N_{\text{err}}}{PK}$$

where  $N_{\text{err}}$  is the total number of bit errors resulting at  $K$  thresholded outputs over the complete cycle, and  $P$  is the number patterns. In our SIIS-ANN,  $K = 3$ , and  $N_{\text{err}}$  is computed as follows:

- At the beginning of each training cycle set  $N_{\text{err}}=0$
- For an individual pattern:
  - IF the desired output = 1 and the actual output  $\leq 0.9$
  - or
  - the desired output = 0 and the actual output  $\geq 0.1$
- THEN  $N_{\text{err}}$  is incremented and the step is re-executed.

- At the end of the cycle set  $E_d = N_{\text{err}}/PK$

### 3.5 Learning factors

Implementation of the error back-propagation learning algorithm may encounter various difficulties. One of the problems is that the error minimisation procedure may produce only shallow local minima in many of the training cases. This may be sufficiently avoided by including some form of randomness in the algorithm concerning:

#### a- Initial weights

The weights of the network to be trained are typically initialised at small random values. The choice of initial weights is, however, only one of several factors affecting the training of the network toward an acceptable minimum error.

The initial weights of the present network were initialised at small random values (between 0.0 and 0.1), except the bias values ( $v_{1,0}$ ,  $v_{2,0}$ ,  $v_{3,0}$ ,  $v_{4,0}$ ,  $w_{1,0}$ ,  $w_{2,0}$ , and  $w_{3,0}$ ) for the hidden and output layers. These biases were initialised at negative random small values (between 0.0 and -0.1).

#### b- Learning constant

Various values were adopted for the learning constant  $\eta$ . It was found that the values of the cumulative, root mean square and decision errors reached their minimum and acceptable values after 400 and 5000 learning cycles for  $\eta = 0.1$  and  $\eta = 0.01$ , respectively [9]. However, the latter value ( $\eta = 0.01$ )

accelerated the present SIIS-ANN convergence without overshooting the solution.

#### *c- Momentum term*

The momentum term was adopted to be 0.5 in order to accelerate the convergence of the error back-propagation learning algorithm of our SIIS-ANN.

### 3.6 SIIS-ANN implementation

The developed SIIS-ANN deals with CCD image frames for scanning and searching for the Local Central Peaked Pixel (LCPP) whose datum is larger than those of the surrounding ones [9, 10]. This technique significantly reduces both the search space and the required computer time. When such a pixel is found the surrounding 24-inputs are adopted, normalised to the value of the LCPP, and then mapped to the network. The outputs are then evaluated and the following three cases are considered for each input pattern:

- Case 1: **IF** the first output > 0.9 **AND**  
the second output < 0.1 **AND**  
the third output < 0.1  
**THEN** classify this pattern as a **STAR image**
- Case 2: **IF** the first output < 0.1 **AND**  
the second output > 0.9 **AND**  
the third output < 0.1  
**THEN** classify this pattern as **NOISE**
- Case 3: **IF** the first output < 0.1 **AND**  
the second output < 0.1 **AND**  
the third output > 0.9  
**THEN** classify this pattern as a **COSMIC RAY**

### 3.7 SIIS-ANN test

The present method was coded in C computer language and applied to the test pattern set (Sec. 3.2). The results obtained agree exactly with the prior known input. However, it is necessary to consider a practical application in order to verify the applicability, reliability and limitations of SIIS-ANN against one of the known packages currently used by astronomers. This was performed through the test case given in the following section.

## 4 Application

### 4.1 Test case

A test case was applied in order to evaluate our method in comparison with the DAOPHOT [7], which is widely used by astronomers. A CCD frame of the star cluster M67 was available together with the recently updated version of this code (DAOPHOTII) as well as all relevant and necessary input files, that need tedious and laborious work to acquire [11]. This is a cluster that is well studied, and accurate astronomical investigations are available. Under these circumstances, a comparison between the two methods was realistic. The frame was  $320 \times 350$  pixels and was taken through the visual optical band within 30 seconds of exposure time.

### 4.2 Results and discussions

The adopted frame was reduced employing the two codes. The present code identified 134 stellar images while 137

stellar images were recognised using the other code. The two methods agree for 132 images, which are displayed by asterisks in Figure 9. On the one hand there were 2 images identified by the present method. These represent faint star images and designated by circles in the figure. The other method could not find these. On the other hand, there were 5 images are recognised by DAOPHOTII. Of these, three stellar images are located at the first or the second pixel close to the frame borders. These are plotted as filled squares in the figure. This case implies that these stars are partially imaged and hence they are of no astronomical importance. In addition, unreliable astronomical data is obtained by Gaussian fitting, as in the DAOPHOTII case, by incomplete data. Our method is not able to deal with cases for which  $5 \times 5$  array data are not available. The two other images are plotted as open squares in a position where our method identified one image only. Only one star can be seen at this location through the Palomar Observatory Sky Survey photographic plate. The ID charts given by Johnson [12], Eggen [13] and Kent et al [14] also resolve this controversy in favour of our method. Inspection of the data around this location reveals that the image is near saturation. For this case, DAOPHOTII assigns two overlapped images through a non-linear fitting process.

The computer time needed for executing SIIS-ANN code has been estimated at 45 seconds using a Pentium II (233MHz – processor) personal computer to display the image via the monitor, to identify stellar images and to encircle them on the display. By contrast a very much longer time is required for running DAOPHOTII, as reported in the manual [15]. DAOPHOTII needs a few minutes to a few hours, mainly for preparing the auxiliary input files necessary for execution. Moreover, as pointed out by Stetson (op cit.), at least 4Mb HD free space are required to reduce a  $512 \times 512$ -pixel frame. On the other hand, regardless of the frame size a few KB of HD free space is sufficient for executing our method to save the position and peak data for the images that are found of stars and cosmic rays.

## 5 Conclusions

The present study shows that, in comparison with DAOPHOTII, the method has the advantages that:

- Better reliability is provided.
- There is considerable higher recognition ability for faint images of stars.
- Extremely short execution time is needed.
- Neither user intervention nor prior knowledge about the CCD frame is needed.
- No complicated computation or numerical fitting process is performed.
- No large HD free space is required, even for large image frames.

The only limitation of the present method is its inability to identify objects having centres at the first or second pixel close to the frame borders. Such an objects, even if it is a star image is of no astronomical significance, as it is an incomplete image. Hence this limitation can be disregarded.

The SSIS-ANN method possesses remarkable features for practical applications, and it is planned to extend its capability to identify images of galaxies.

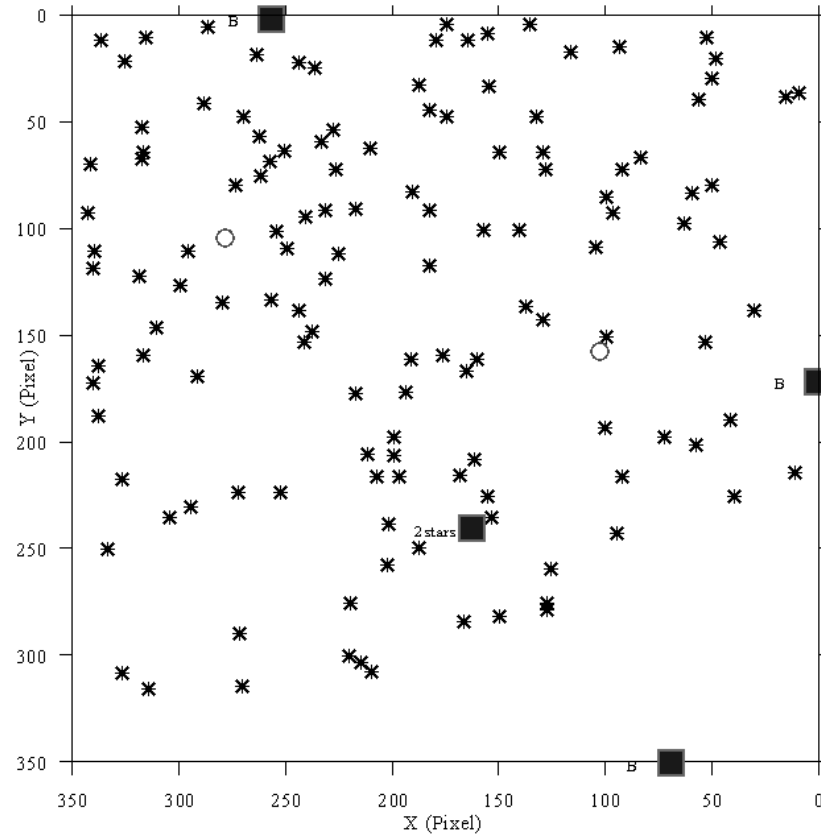


Fig. 9: Map of the images of stars recognised in the M67 cluster frame  
 North is UP and East is to the left  
 Asterisks: images identified by both methods  
 Circles: images identified by the SIIS-ANN method only  
 Squares: images identified by the DAOPHOTII method only

## References:

- [1] Tody, D.: *SPIE*. 1980, 264, p. 171
- [2] Lupton, R. et al: *Astronomical Journal*. 1986, 91, p. 317
- [3] Linde, P.: *Highlights in Astronomy*. 1989, 8, p. 651.
- [4] Bleacha, R.: *Astronomy and Astrophysics*. 1984, 135, p. 401
- [5] Penny, A. et al: *R., MNRAS*, 1986, 220, p. 845
- [6] Mateo, M. et al: *Proceeding of the first ESO/ST-ECF Data Analysis Workshop*. 1989, p. 69
- [7] Stetson, P.: *Publications of the Astronomical Society of the Pacific*. 1987, 99, p. 191
- [8] Gilliland, R. et al: *Publications of the Astronomical Society of the Pacific*. 1988, 100, p. 754
- [9] Younis, F. I.: MSc Thesis, Faculty of Engineering, Al-Azhar University, Cairo, Egypt, 1998
- [10] Alawy, A., El-Bassuny: *Astrophysics and Space Sciences*. 2001, (in press)
- [11] Bojan, D.: <http://david.fiz.uni-lj.si/DAOPHOTII>, 1996
- [12] Johnson, H. et al: *Astrophysical Journal*. 1955, 121, p. 616
- [13] Eggen, O. et al: *Astrophysical Journal*. 1964, 140, p. 130
- [14] Kent, A. et al: *Astronomical Journal*. 1993, 106, p. 181
- [15] Stetson, P.: *User's Manual for DAOPHOTII*. Dominion Astrophysical Observatory, Victoria, Canada, 1996
- Assoc. Prof. Ahmed El-Bassuny Alawy  
 e-mail: abalawy@hotmail.com  
 National Research Institute of Astronomy and Geophysics (NRIAG)  
 Helwan, Cairo, Egypt
- Doc. Ing. Boris Šimák, CSc.  
 e-mail: simak@feld.cvut.cz  
 Department of Telecommunications Engineering  
 Czech Technical University in Prague  
 Faculty of Electrical Engineering  
 Technická 2, 166 27 Praha 6, Czech Republic
- Eng. Farag Ibrahim Younis Elnagahy, MSc  
 e-mail: faragelnagahy@hotmail.com  
 Department of Telecommunications Engineering  
 Czech Technical University in Prague  
 Faculty of Electrical Engineering  
 Technická 2, 166 27 Praha 6, Czech Republic
- Prof. Eng. Mohamed Ashraf Madkour  
 Department of Systems & Computers Engineering  
 Al-Azhar University  
 Faculty of Engineering  
 Cairo, Egypt
- Assoc. Prof. Mohamed S. Ella  
 National Research Institute of Astronomy and Geophysics (NRIAG)  
 Helwan, Cairo, Egypt