Proceedings of the XIII Bulgarian-Serbian Astronomical Conference (XIII BSAC) Velingrad, Bulgaria, October 3-7, 2022
Editors: Evgeni Semkov, Milan S. Dimitrijević, Momchil Dechev and Zoran Simić
Publ. Astron. Soc. "Rudjer Bošković" No 25, 2023, 137-142

PROCESSING DOUBLE STARS IMAGES USING MACHINE LEARNING

RADE PAVLOVIĆ[1], ALEKSA JOVANOVIĆ[2],
ZORICA CVETKOVIĆ[1] and OLIVER VINCE[1]

[1]*Astronomical Observatory, Volgina 7, 11060 Belgrade, Serbia*
[2]*Faculty of Mathematics, University of Belgrade, Serbia*
E-mail: rpavlovic@aob.rs, ai18272@alas.matf.bg.ac.rs,
zorica@aob.rs, ovince@aob.rs

Abstract. We use Machine Learning to process observations of double stars obtained by lucky imaging technique. First, we generate a large number of images of a double star system, whose PSF is Gaussian in the shape distribution, in a wider range of separations and position angles, which serves us for learning the system. Once we have trained the system to recognize double stars, we input our images of a particular double star, which we remade at the Astronomical Station at Vidojevica with the 1.4 m telescope attached CCD camera Andor iXon 897 Ultra, and as a result we get its separation and positional angle.

1. INTRODUCTION

Since CCD cameras were used to capture two stars at the end of the 20th century, the most common method for determining relative coordinates, separation and position angle, of double stars is the calculation of angular distance between the photocenters of the components and the angle between the lines which passes through the photocenters of components and direction to the north celestial pole calculated over the east. Here we suggest alternative method - application of machine learning i.e., convolutional neural network (CNN) for determining these double star parameters.

2. SIMULATED IMAGES OF DOUBLE STARS

To learn the system, we simulated images of binary stars. Knowing that our test star WDS 16238+6142 = STF 2054AB has a separation of about 1 arcsecond and a position angle of about 270 degrees we generated images in the range 0.7 to

1.3 arcsec with step 0.01 arcsec for separation and in the range of 250 to 290 degrees with step of 1 degree for positional angle[1]. Star profiles were 2D Gaussian

$$f_i(x,y) = A_i e^{-\left[\frac{(x-x_{0i})^2}{2\sigma_{Xi}^2}+\frac{(y-y_{0i})^2}{2\sigma_{Yi}^2}\right]}, i = 1,2$$

where A_i is the amplitude, x_{0i} and y_{0i} are the coordinates of the Gaussian center, σ_{Xi} and σ_{Yi} are variances for the given component axes. Labels are primary $i = 1$, and secondary component $i = 2$. In order to make the simulated images as realistic as possible, we added noise of the form Gaussian distribution and amplitude equal to the noise amplitude on the recorded images. The primary star is placed in the center images. The image resolution was 256 × 256 pixels. This resolution of simulated recordings was chosen to match the resolution of real images obtained on the telescope. As an example of a simulated image of double star is given in Fig. 1.

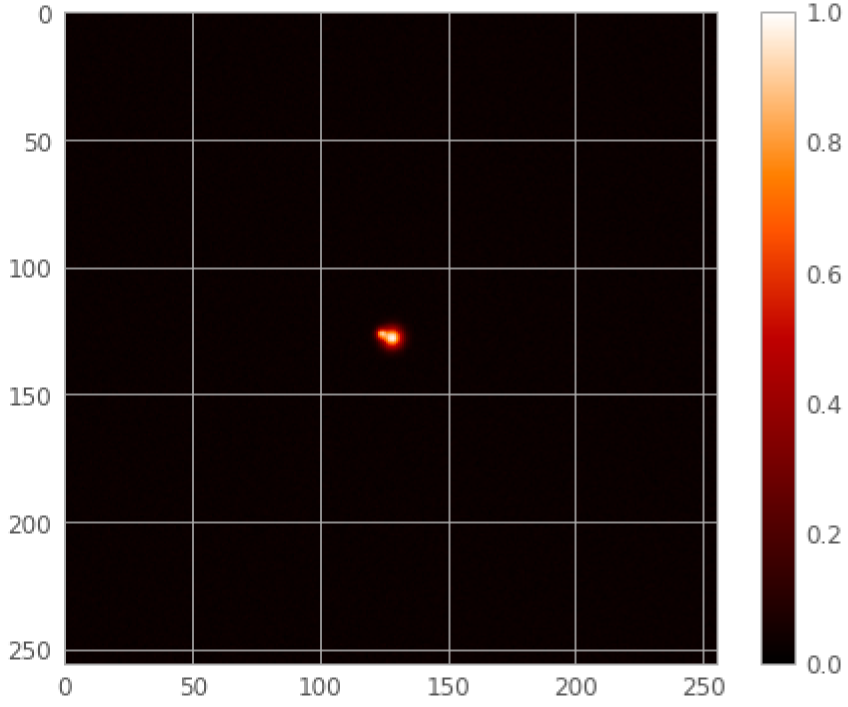

Figure 1: Example of simulated image with parameters $\rho = 0.70$ arcsec and $\vartheta = 250^o$.

[1] In our case, we measured positional angle in the local coordinate system related to the image plane.

3. REAL DOUBLE STAR IMAGES

Our equipment for lucky imaging mounted at the right port of “Milanković” telescope consists of two CCD cameras Andor iXon 897 Ultra and SBIG STXL-6303e and an Optec Perseus 4-port Instrument Selector as shown on Fig. 2. The detailed description of double star imaging procedure is given in Pavlović et al. (2021). As a test object we chose double star WDS 16238+6142 = STF 2054AB.

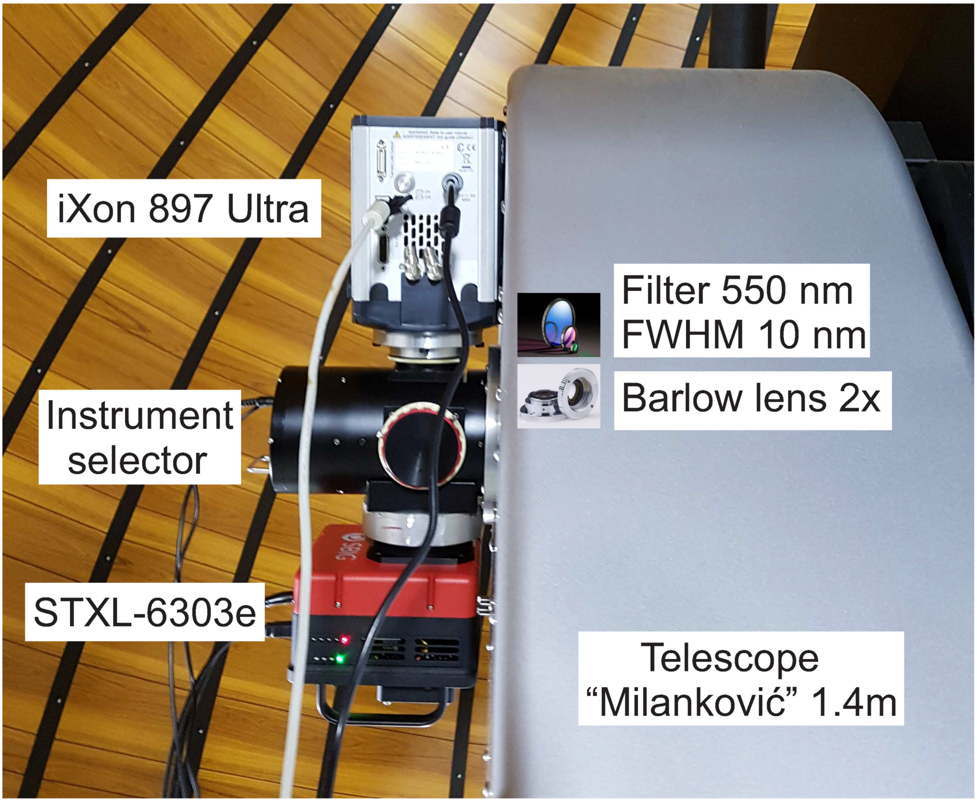

Figure 2: The equipment for lucky imaging attached to telescope “Milanković” at AS Vidojevica.

We recorded it twice using lucky imaging on September 16, 2021. The first set of recordings was made at 23:40:29 and consisted of 100 exposures, and the second one recorded at 23:42:24 and consisted of 1000 exposures. All images were acquired with an exposure of 50 ms. From both sets we chose 5% and 10% of the best frames. In that manner we have prepared four files to which should be applied convolutional neural network. Fig. 3 shows all four acquired frames of the double stars WDS 16238+6142 = STF 2054AB in reduced size as the details would be better seen.

4. CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional neural networks are a class of networks for deep learning specialized for working with images. They consist from input layer, hidden layers and output layer; layers which responsible for convolution are inside the hidden layers. By convolution of input matrix and kernel, we get a new matrix that has more pronounced aspects useful for further learning, which is defined as

$$(A * B)_{i,j} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} A_{i+k,j+l} B_{k,l}\; i = 0, \dots, m-p; j = 0, \dots, n-q$$

where is $A \in R^{m \times n}$ input matrix (images or another signal) and $B \in R^{p \times q}$ is kernel (or filter).

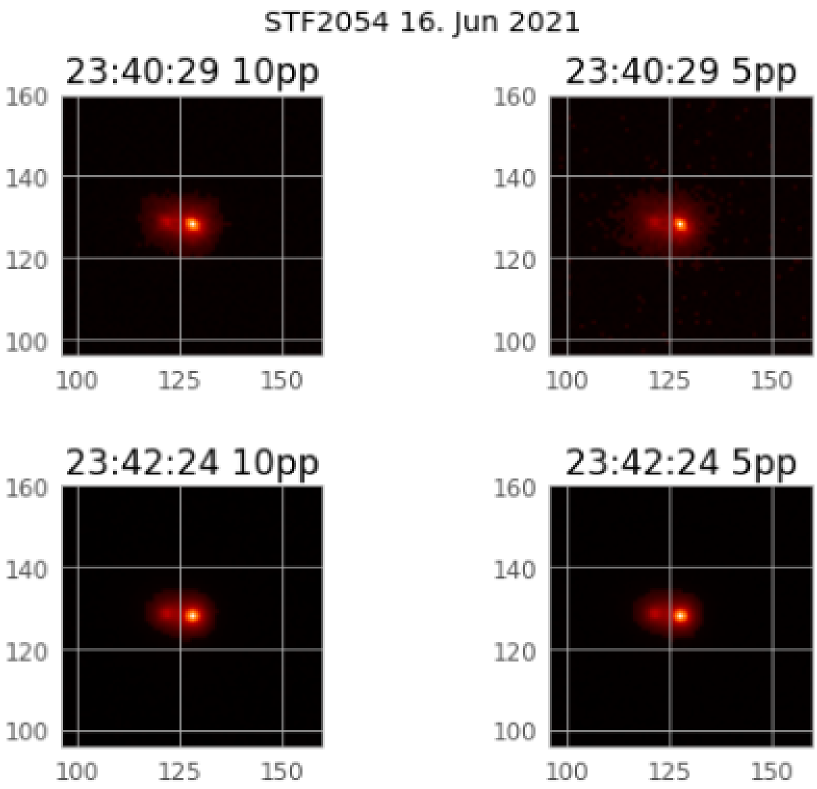

Figure 3: Four acquired frames of WDS 16238+6142 = STF 2054AB in reduced size (94 x 94 pixels).

A convolutional network is trained to minimize output errors of given inputs from the training set. During that process, kernels are learned that implement useful transformations over the input signal. In the general case, CNN works on tensors (arrays of matrices of the same dimensions), where a single matrix is a channel.

Accordingly, tensor convolution operation is defined in similar way as earlier but $A \in R^{m \times n \times c}$ and $B \in R^{p \times q \times c}$ are input tensors, and as output we get a matrix

$$(A * B)_{i,j} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} \sum_{t=0}^{c-1} A_{i+k,j+l,t} B_{k,l,t} \; i = 0, \dots, m-p; j = 0, \dots, n-q.$$

CNN learns a certain number of filters in each layer. A set of filters that act on the same input signal are called convolutional layer; it outputs a tensor so to each channel corresponds output of one filter. The outputs are further non-linearly transformed using activation function, most often ReLU (Rectified Linear

Unit) but we used a slightly more common Swish (Ramachandran et al. 2017), by applying the given activation function to each tensor channel.

Another type of frequent layers used in CNNs are pooling layers. These layers perform aggregation data in order to reduce calculations and the number of network parameters. They are realized by changing part of tensor channel in certain dimensions with one value, in relation to the type of aggregation which we use. The most common type of aggregation is maximum pooling; with this, high values are preserved i.e. patterns detected by kernels, but is lost its specific position. The importance of this compromise i.e. model size/training time and accuracy depends on specific applications.

5. SHORT DESCRIPTION OF USED MODEL

The convolutional network was implemented in Python using Keras[2] and the Scikit-learn[3] library. The input data is simulated signals, described in section 2, in the form of tensors with dimension 256x256x1 (one-channel tensor). The output layer consists of only two neurons: one for angular separation and the other for position angle. Division of the set into training, test and validation subset was performed randomly, in the ratio 80:10:10, respectively.

Minmax normalization was applied to the recorded frames because of improving performance and stability of the model

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where x_{min} and x_{max} are minimal and maximal pixel intensity on the frame. In order to avoid overfitting and improving generalization, we used two regularization techniques: early stopping and the checkpoint model. When we train a model, an insufficient number of epochs leads to underfitting, but an excessive number leads to overfitting. Compromise is an early stop - an

[2] https://keras.io/

[3] https://scikit-learn.org/stable/

interruption of training at the moment when performance of validation set begins to decline. During training process, we monitor the validation loss and we remember those model wich have currently the smallest validation error. As the error function for our model, we used standard mean squared error (MSE).
The complete calculation process is shown by the block diagram in Fig. 4.

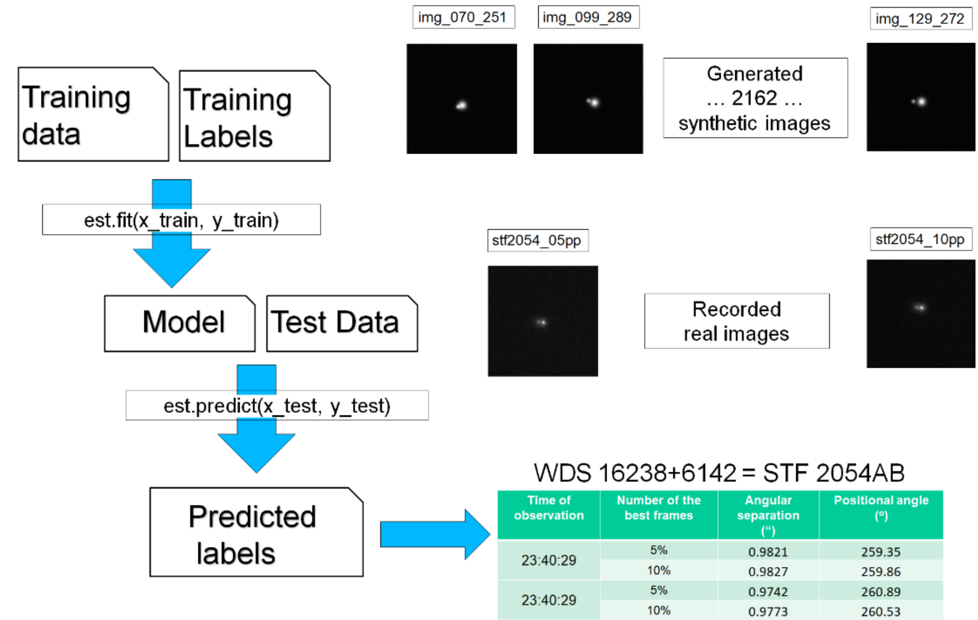

Figure 4: Block diagram of the calculation process.

6. CONCLUSIONS

The calculated MSE for our model was $4.52 \cdot 10^{-6}$ which indicates stable and accurate model. The final values for the separation and position angle with the corresponding standard deviations for double star WDS 16238+6142 = STF 2054AB are

$$\rho = (0.9791 \pm 0.0040),\ \vartheta = (260.16 \pm 0.69)^{o}.$$

The obtained small values of the standard deviation for separation and positional angle indicate that convolutional neural network can be applied to "measure" these quantities.

References

Pavlović R., Cvetković Z., Damljanović G., Jovanović M.D.: 2021, Lucky Imaging at Vidojevica, *Publ. Astron. Obs. Belgrade*, 100, 339-343.
Ramachandran P., Zoph B., Le Quoc V.: 2017, *Searching for Activation Functions* https://arxiv.org/abs/1710.05941