CHALMERS
UNIVERSITY OF TECHNOLOGY

Machine Learning for Protostellar Image Fitting

A Convolutional Neural Network Approach

Master's thesis in Complex Adaptive Systems

JORGE MIFSUT BENET

DEPARTMENT OF SPACE, EARTH AND ENVIRONMENT

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Machine Learning for Protostellar Image Fitting

A Convolutional Neural Network Approach

JORGE MIFSUT BENET

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Earth, Space and Environment
Division of Astronomy and Plasma Physics
Chalmers Initiative on Cosmic Origins (CICO)
Chalmers University of Technology
Gothenburg, Sweden 2023

Machine Learning for Protostellar Image Fitting
A Convolutional Neural Network Approach
JORGE MIFSUT BENET

Supervisor: Jonathan C. Tan
Examiner: Jonathan C. Tan

Master's Thesis 2023
Department of Earth, Space and Environment
Division of Astronomy and Plasma Physics
Chalmers Initiative on Cosmic Origins (CICO)
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Protostar within the dark cloud L1527 captured by NASA's James Webb Space Telescope *NASA, ESA, CSA, and STScI. Image processing: J. DePasquale, A. Pagan, and A. Koekemoer (STScI)*

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Machine Learning for Protostellar Image Fitting
JORGE MIFSUT BENET
Department of Space, Earth and Environment
Chalmers University of Technology

Abstract

In this work we present a Convolutional Neural Network (CNN) architecture that can be utilized to regress two key physical features of massive protostars from images in the 19 μm and the 37 μm bands: the inclination angle with respect to the line of sight (θ_{view}) and the protostellar mass (m_*). The network was trained on images generated by Monte Carlo Radiative Transfer simulations following the description of massive star formation from the Turbulent Core Model [1] [2] [3]. We show by testing the network on the synthetic data that it is feasible to regress the values for the aforementioned features solely from the morphology of the protostar displayed in the images, with a mean offset of $\mu_{\delta(\theta)} = -0.43^\circ$ and standard deviation of $\sigma_{\delta(\theta)} = 4.45^\circ$ for θ_{view} and $\mu_{\delta(m)} = -0.69$ $\text{M}_\odot$ and $\sigma_{\delta(m)} = 2.31$ $\text{M}_\odot$ for m_*. The network is subsequently re-trained with added negative samples to also discern between inputs that contain protostars and inputs that contain random noise, albeit with a slight increase in the dispersion of the offsets. This new model is then tested on an image of *Cepheus A* from the SOFIA Massive Star Formation Survey and the estimations for θ_{view} and m_* agree within the error range with other estimations in the literature obtained by different methods.

Keywords: Deep Learning, Convolutional Neural Networks, Machine Learning, star formation, protostars, massive stars.

Acknowledgements

I would like to express my gratitude to my family, my girlfriend Ophélie and my friends from the master and the phadder group for their love and support throughout these two years I have spent in Sweden. I want to thank as well Prof. Jonathan Tan for his advice and supervision in this thesis. Also, I would like to thank, for their advice and assistance, to Yichen Zhang and specially for all the time spent on helping me to Duo Xu. Finally, I would like to thank Chalmers University of Technology, all the professors, teaching assistants and personnel that I encountered these two years for a very positive and unforgettable experience.

Jorge Mifsut Benet, Gothenburg, June 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ANN	Artificial Neural Network
CASI	Convolutional neural network Aproach for Shell Identification
CNN	Convolutional Neural Network
DL	Deep Learning
FIR	Far Infrared
IR	Infrared
ML	Machine Learning
NIR	Near Infrared
RT	Radiative Transfer
SED	Spectral Energy Distribution
SGD	Stochastic Gradient Descent
SOFIA	Stratospheric Observatory For Infrared Astronomy
SOMA	SOFIA Massive Star Formation Survey

Contents

List of Figures

List of Tables

1

Introduction

Machine Learning (ML), and specifically Deep Learning, are growing in popularity due to the fast and significant advancements currently being achieved at carrying out a broad range of tasks. In the context of astronomy, ML can be a promising tool to automatize the analysis, measurement and prediction of several phenomena in the ever-growing amounts of data being collected from telescopes and space missions, while being faster and more accurate than human labeling or other methods.

Some examples [4] of the implementation of Machine Learning and Deep Learning methods in the field of astronomy involve tasks such as photometric redshift estimation [5], gravitational lensing identification [6], galaxy morphology classification [7] or stellar feedback bubble identification [8].

The present work aims to implement a Deep Learning method, more specifically a Convolutional Neural Network, for the regression of key physical features of massive protostars from multi-wavelength images. This network is trained on synthetic data generated by simulations and then tested on observational data from the SOFIA Massive Star Formation Survey.

First, theoretical background will be introduced for the Machine Learning aspects, as well as for the model of massive star formation which was followed for the generation of synthetic data and which explains the target physical magnitudes that the network aims to regress. Then, the specific methods implemented along the development of this study will be described. Finally, the results of testing the network model on both synthetic data and the observational images will be presented at the end.

2

Theory

2.1 Background

2.1.1 Machine Learning

One of the earliest uses of the term *Machine Learning* is attributed to the IBM researcher A.L. Samuel in [9], where he described his methods as "programming of a digital computer to behave in a way which, if done by human beings or animals, would be described as involving the process of learning". To complement this, the following definition of learning is proposed in [10]: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". Broadly, Machine Learning consists in the application of mathematical models that make use of data and its internal structure in order to perform some task, without explicitly hard-coding the solution. Linear regression is a simple example of Machine Learning.

A common way to classify Machine Learning algorithms is according to the aforementioned experience, E, from which they learn. This way, they can be divided into supervised, unsupervised and reinforcement learning. In supervised learning, the training data contains target labels that the algorithm must aim to predict. Meanwhile, unsupervised learning consists on learning patterns in unlabeled data, such as in clustering algorithms. On the other hand, in reinforcement learning the algorithm receives feedback during the process of training according to a set reward system. Another axis along which to classify Machine Learning algorithms can be their task, T. For instance, one can find algorithms aimed at classification, regression, translation or de-noising, among many others.

2.1.1.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a subset of Machine Learning algorithms. Historically, ANNs were conceived taking biological neural networks as inspiration. Similarly to the neurons in the brain, the computational units of an ANN are artificial neurons, first postulated in [11]. These artificial neurons form a network in which they are the nodes, and their connections are defined by a series of weights and biases. In general weights and biases between different pairs of neurons assume

different numerical values, reflecting different strengths of the synaptic couplings [12]. Additionally, the output of each neuron is wrapped by some activation function, imitating the pulses of current that happen in the cellular membranes of a natural neuron.

One of the simplest architectures for an ANN is that of the fully connected feed-forward multi-layer perceptron. Here, neurons are distributed in layers and each neuron only takes as input the outputs of the neurons of the previous layer. Then, they process this information by performing a linear combination of the inputs taking into account the weights of the connections to the corresponding neurons in the previous layer. The output is then passed through the activation function and fed forward into the network. This process can be synthesized by the following expression:

$$x_i = g\left(\sum_j w_{ij} x_j - \theta_i\right), \tag{2.1}$$

where i denotes the current neuron, j are the indices of the neurons in the previous layer, g is an activation function, w_{ij} are the weights of the connections of each of the j neurons in the previous layer to the current i-th neuron, and θ_i is some bias for the linear combination.

In the context of supervised learning, the last layer (or ouput layer) of the network needs to present the same shape as the ground truth labels. This way, the two can be compared in terms of some loss function, which acts as the performance measurement of the network. The objective is to minimize the loss, and this is attained by updating the values of the weights and biases of the network according to some gradient-based optimization algorithm. These algorithms compute the gradient applying the chain rule from the loss function at the output all the way back to each of the neurons in the network, in a process called backpropagation. The act of updating the weights and biases according to the backpropagation of the loss is called training. It is repeated over a number of epochs on the same training data set. It is also common practice to test the model in every epoch with a separate validation data set, in order to assess that the model performs similarly on an unseen set of data and it is not merely learning to discern the specific instances of data in the training data set. In other words, validation is carried out in every epoch to verify that the model is good at generalizing and is not overfitting.

An attractive property of some ANN architectures such as the Multilayer Feedforward Network [13] or the Convolutional Neural Network [14] is that, given a large enough depth (i.e., number of layers), they are universal approximators of any given continuous function with arbitrary accuracy. It has been shown [15] that another crucial requirement for this Universal Approximation Theorem to be satisfied is that there exists some non-linearity in at least one hidden layer, hence why the importance of activation functions. This property of Neural Networks allows for countless applications in Science and Engineering and it is the reason for which researchers have been implementing network architectures with an ever-increasing level of complexity and depth: this is where the term *Deep Learning* originates.

2.1.1.2 Convolutional Neural Networks

Convolutional networks (CNNs) are a specific type of neural network for processing data that has a known grid-like topology [16], such as time-series data (1-D grid) and image data (2-D grid of pixels). As opposed to the layers of neurons in the feed-forward multi-layer perceptron, the building blocks of CNNs, the convolutional layers, consist of a stack of filters to be applied to the input data. These filters have the same dimensionality as the latter, and they are of a specified size (kernel size). Each filter is slid through the data with a certain stride. In contrast to equation 2.1, the operation that is carried out in each stride step is called a convolution. It consists in the sum of the resulting values in an element-wise matrix multiplication between the kernel and the specific elements of the input matrix that the filter is affecting on the certain stride. This can be expressed as follows:

$$F_{ij} = K * I = g\left(\sum_p \sum_q k_{pq} x_{p+s(i-1),q+s(j-1)} - \theta\right), \tag{2.2}$$

where I is the input matrix, K is the kernel or filter, $*$ is the convolution operation, g is an activation function, k_{pq} and x_{pq} are the (p, q)-th elements of the kernel and input matrices respectively, s is the stride and θ is a bias. As is also shown in 2.2, the numerical value produced by this operation conforms the (i, j)-th element of an output matrix F, called the feature map. As a consequence of this so called *sliding window* process, the side size of the feature map (assuming square inputs and kernels) will be:

$$F = \frac{W - K - 2P}{S}, \tag{2.3}$$

where F, W and K are the side size of the feature map, input and kernel respectively, S is the stride and P is the size of a possible padding to the sides of the input image that might be added, precisely, to control the size of the output matrix.

By construction, each element in the feature map contains information from several elements (or pixels) of input. This is called the *receptive field* of that unit. This effect transfers to each of the layers in the network. Therefore, the receptive field of the units in the deeper layers of a convolutional network is larger than the receptive field of the units in the shallow layers.

In CNNs, the trainable parameters are the elements of the kernel, as well as the bias. Through the same process of training and backpropagation described before, they are able to learn to extract features at multiple levels of abstraction. Taking into account the concept of receptive field, the first layers of a CNN might learn to recognize in an image more simple and local features, such as edges and lines, while deeper layers might learn to recognize more complex shapes or objects.

Another essential component of a CNN is the pooling layer, which reduces the spatial dimensionality of the input by performing downsampling of the feature maps. The most common form of pooling is Max Pooling, which only maintains the maximum value within a pool of pixels. This technique helps to reduce the number of parameters in the model and ameliorates computational efficiency.

2.1.2 Radiative transfer of models of massive star formation

The machine learning model presented in this work has been trained on a set of synthetic data generated by Monte Carlo continuum radiative transfer simulations performed for the models of massive star formation presented in [17] and [3], hereafter the model grid. This subsection will give an overview of the model grid and the model for massive star formation that it represents.

Massive stars are fundamental to the understanding of the evolution of galaxies, yet there is no consensus on how they form. Different scenarios range from core accretion models ([2], [1]), competitive models ([18]) to protostellar collisions ([19]). This incertitude is due to the relative rarity, large distance, crowded environments and high extinctions that characterize observations of massive protostars.

The main method to understand the properties of massive protostars is to analyze their broadband spectral energy distributions (SED) via radiative transfer (RT) modeling, particularly on the NIR to FIR range. All the physical features of the massive star forming system affect its SED, thus requiring a large number of parameters to describe it precisely. The Turbulent Core model [1] aims to reduce the number of independent parameters compared to previous models for massive star formation. In this model, massive stars are formed from pre-assembled massive prestellar cores supported by internal pressure provided by a combination of turbulence and magnetic fields. These cores are defined as a region of a molecular cloud that forms a single star or a closed binary by means of gravitational collapse. They are assumed to be quasi-spherical, self-gravitating, in near-virial equilibrium and in pressure equilibrium with the surrounding star-cluster-forming clump. The size of the core is defined by the mean mass surface density of the clump, Σ_{cl}. Also, the core is defined to contain a single, rotationally supported accretion disk.

This way, the model grid is constructed from two initial conditions: the initial core mass, M_c, and the environmental mass surface density, Σ_{cl}; which alongside the protostellar mass, m_*, inclination and foreground extinction conform its main free parameters. Properties of the different components of the system (namely the protostar, disk, envelope and outflow) and their evolutions can be self-consistently calculated from the two main initial conditions through several analytical and semi-analytical methods. The evolutionary history of a protostar from a specific set of initial conditions is referred as *evolutionary track* and a set of tracks is what we call the model grid. A specific moment in the evolutionary track is an *evolutionary stage* and it is specified by m_*.

The values used for the initial conditions are in a fiducial range similar to the range of values observed in most Galactic massive star-forming regions. Namely, these values are $M_c \in [10, 480]$ $M_\odot$ and $\Sigma_{cl} \in [0.1, 3.16]$ g/cm^2. As can be observed in Figure 2.1, these value ranges contain the internal clumps or cores of infrared dark clouds (IRDC cores, dark green crosses) that are thought to represent an initial stage in the process of formation of massive stars and massive star forming cores/clumps (MSF, red squares).

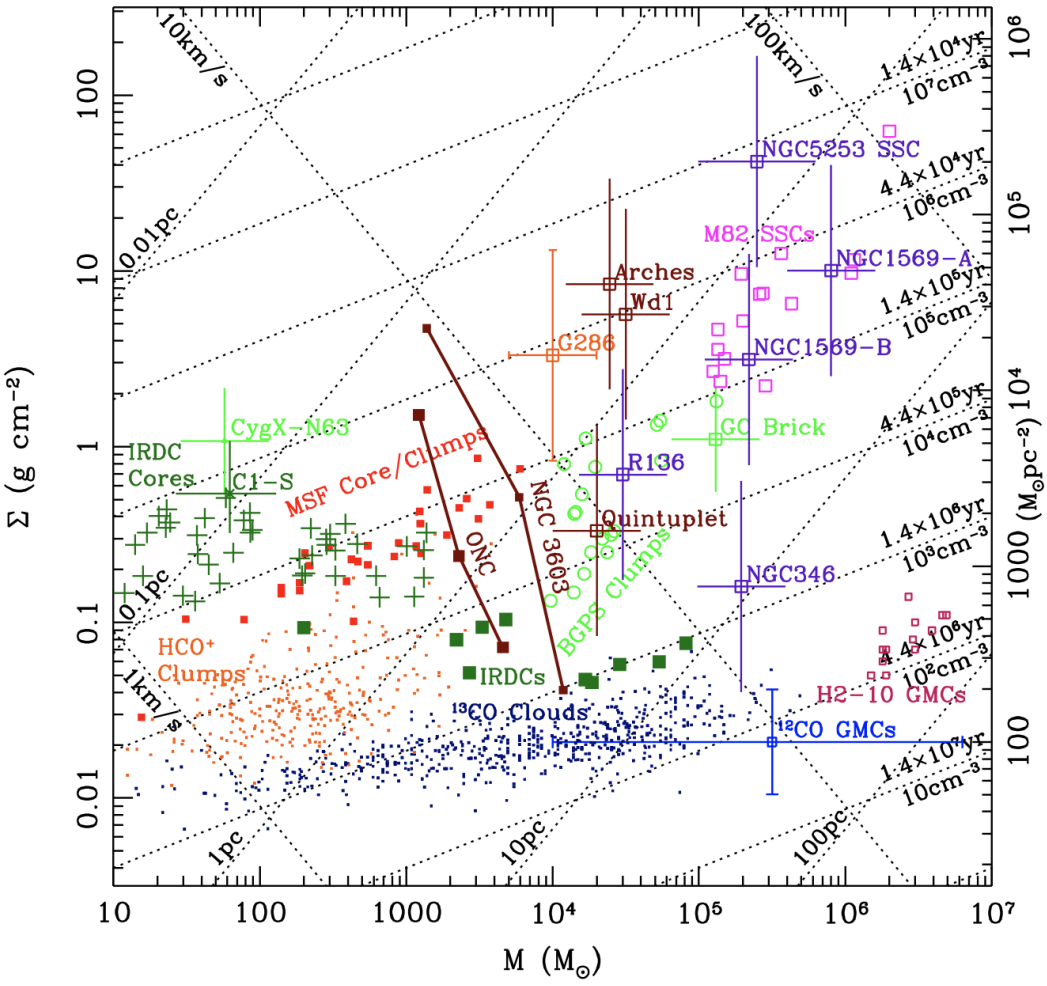

Figure 2.1: Plot of mass surface density, Σ versus mass, M, of different observed environments of star formation ([20])

The Monte Carlo continuum radiative transfer simulations for the protostellar cores in the model grid with the above mentioned initial conditions are carried out utilizing HOCHUNK3d from [21], [22]. For each stage, the images (as presented in [3]) are created by calculating the temperature profiles and producing the SEDs at 20 different viewing inclinations. The result is a set of 8640 SEDs defined by four independent parameters: M_c, Σ_{cl}, m_* and θ_{view}. Finally, from these SEDs images are produced with the surface brightness at different wavelengths, which constitute the training set for our Machine Learning model.

By comparing the SEDs from the models with observations one can asses how the IR emission of massive protostars can be explained by specific initial conditions and evolutionary stages, ultimately testing the theory of massive star formation via turbulent core accretion. In order to conduct this comparison, the observation needs to be re-scaled by distance (as the simulations are defined to be observed at 1 kpc) and adjusted by foreground extinction. Also, the model SEDs should be convolved with the transmission profiles of the instruments utilized in the observations, with

the intention of simulating the fluxes detected in said observations.

Some extra caveats of comparing the model grid to real observational data involve the fact that the clump is not included in the RT simulations (other than in Σ_{cl}) even if it affects the observed SEDs by providing additional foreground extintion at shorter wavelengths and additional emission at longer wavelengths. The model might also underestimate the emission at short wavelenghts by not taking into account emissions from small grains transiently heated by single photons. Taking these considerations into account, the estimated lower and upper bounds for the wavelength of the observations are $8 \leq \lambda \leq 70$ μm ([3]). For the training data, the specific wavelengths used were 19.7 and 37.1 μm from the F8 and L4 filters of the SOFIA FORCAST instrument, respectively.

2.1.3 SOMA data

The trained network is finally to be tested on real observation data. In particular, the images utilized are extracted from the Stratospheric Observatory For Infrared Astronomy (SOFIA) Massive (SOMA) Star Formation Survey (PI: Tan), which uses the FORCAST instrument ([23]) to image massive protostars from $\sim$ 10 to 40μm. These wavelengths identify warm dust thermal emission, which in Core Accretion models primarily originates from the interior of protostellar outflow cavities. The presentation of the survey and further details of the methods can be foud in [24]. Relevant to this study, Table 2.1 contains data for the coordinate value increment per pixel in the images obtained with this instrument, which is necessary for the re-scaling of the images to coincide with the format of the training data.

Table 2.1: Coordinate value increment deg/pixel for the instrument SOFIA FORCAST at the 19 and 37 μm bands. CDELT1 is for R.A. and CDELT2 for declination.

	CDELT1	CDELT2
19 μm	$-2.58133324992e-005$	$2.58133324992e-005$
37 μm	$-2.58346665741e-005$	$2.58560004529e-005$

SOFIA is a Boeing 747SP aircraft modified to carry a reflecting telescope. Flying into the stratosphere at 38,000-45,000 feet allows SOFIA to operate above 99 % of Earth's atmosphere, avoiding the atmospheric IR absorption and enabling astronomers to conduct studies of the universe that are not feasible with ground-based observatories. SOFIA is made possible through a partnership between NASA and the German Space Agency at DLR.

The focus of this study will be on the source *Cepheus A* (CepA, Figure 2.2) imaged at wavelengths of 19.7 and 37.1 μm in order to coincide with the simuated data. The results obtained in the present study will be compared to the latest analysis performed on this source. Table 2.2 compiles the results from [25] for the values of the key features of the protostar we are concerned with.

Table 2.2: Estimated values for the distance, location ([24]) and the four free parameters of the model grid ([25]) for the source CepA

Source	d (kpc)	R.A. (J200)	Decl. (J200)	M_c ($M_\odot$)	Σ_{cl} (g/cm^2)	m_* ($M_\odot$)	θ_{view} (deg)
Cep A	0.7	$22^h56^m17^s.98$	$+62°01'49.''39$	107^{+45}_{-31}	$0.544^{+0.626}_{-0.291}$	16^{+6}_{-5}	71 ± 31

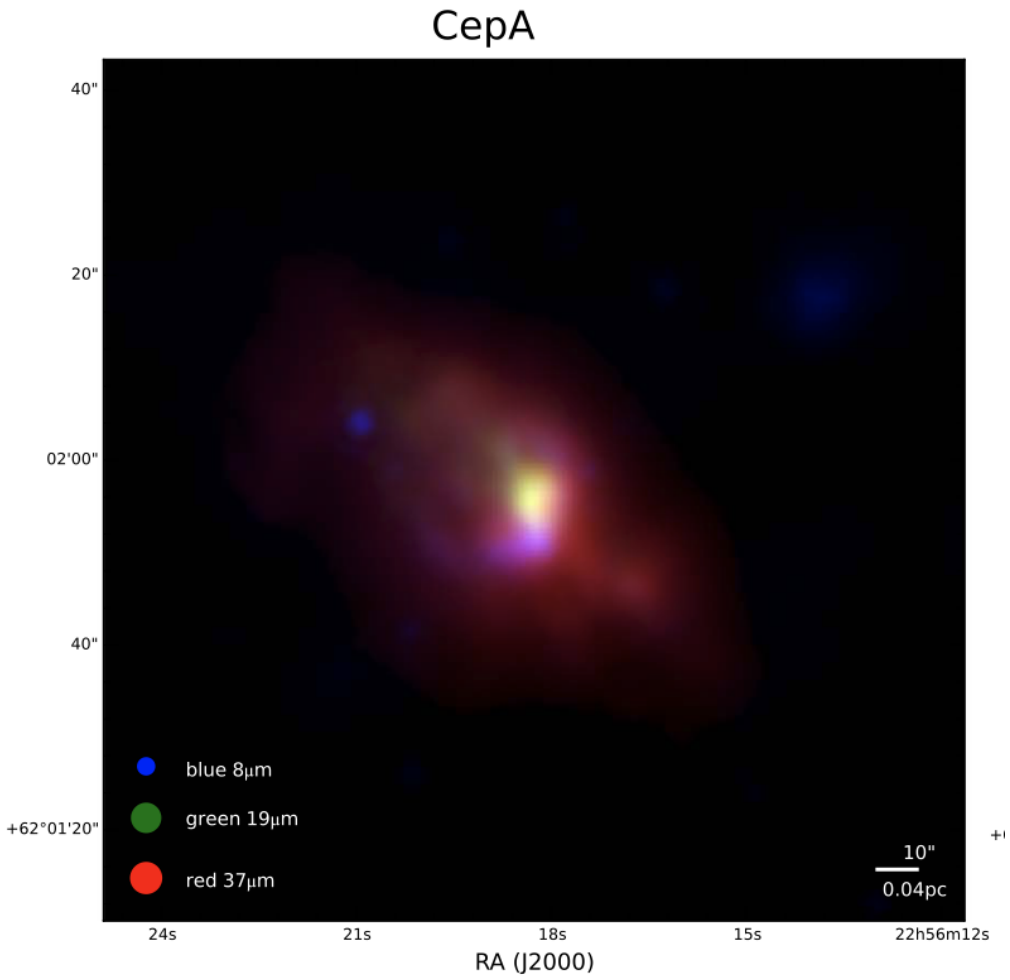

Figure 2.2: RGB image of the source Cepheus A. SOFIA-FORCAST 37 μm and 19 μm are shown in red and green respectively, and blue represents *Spitzer* IRAC 8 μm ([24])

3

Methods

3.1 Normalization and pre-processing

Normalization of the input data is essential for obtaining a good performance from ML algorithms. It ensures that all the features in the input data are distributed in the same range, preventing specific features with higher values to dominate over others. It has also been shown to improve performance and computation time [26].

In the specific case of our training data, extracted from the model grid, it is characterized by having an extremely wide dynamic range. The orders of magnitude of the surface brightness pixel values vary from 0 to $\mathcal{O}(10^9)$ MJy/Sr. As the regression of the values for the free parameters of the model was expected to be obtained solely from the morphology observed in the images and not from the absolute brightness values, it was reasonable to follow the standard practice of normalizing the images to pixel values between 0 and 1. However, due to the broad dynamic range of the images, a simple linear re-scaling by the mean and standard deviation of their surface brightness distribution did not result in a good performance of the neural network. This can be explained by the fact that, in most instances, most of the meaningful information of the images would get compressed and pushed towards 0 when scaling by the large standard deviation values, making it difficult for the network to learn the differences in the data with enough precission.

Therefore, a different method for normalization was designed and it consisted of the following steps. First, each image was treated individually. All the pixels with a value of 0 were converted to NaN, in order to not be taken into account for the next steps, as they only represent the background of the image, do not contain information about the protostar's morphology and consist of a disproportionate amount of pixels compared to the protostar, making the set imbalanced. Then, the rest of the brightness distribution was cutoff by a lower and and upper threshold, defined by the 20% and 99.95% percentiles of the distribution respectively. Afterwards, the resulting distribution was shifted by the lower threshold and rescaled by the upper threshold. Later, the NaNs are reverted to 0 and the whole distribution was multiplied by 10^6. Finally, a base 10 logarithm is applied to all the image (with the aim to decompress its dynamic range) after adding 1 to all the pixels, to ensure that there are no instances of logarithm of 0. The result is consequently divided by 6, as the largest values input to the logarithm should have been 10^6, making the new

uncompressed dynamic range be contained in the interval $[0, 1]$. Equations 3.1 and 3.2 express the previous process of normalization:

$$S'_{ij} = \begin{cases} S_{p,20} & \forall S_{ij} < S_{p=20} \\ S_{p,99.95} & \forall S_{ij} > S_{p=99.95} \\ S_{ij} & \text{otherwise} \end{cases} \tag{3.1}$$

$$S''_{ij} = \log_{10}\left(\left(\frac{S'_{ij} - S_{p=20}}{S_{p=99.95}}\right) \cdot 10^6 + 1\right)/6, \tag{3.2}$$

where S_{ij} is the ij-th pixel of the image S, the final normalized image is S'' and S_p are the values at the percentile p of the distribution.

For further pre-processing, images were also cropped to a 128 by 128 pixel square around the center of the image, in order to downscale the input to the network while still containing the whole protostar in the image. Also, 35 copies of each image were created with an added rotation, ranging from 10° to 360° in 10° steps.

Additionally, the labels are also normalized between $[0, 1]$. For the case of θ_{view} this is done by taking $\cos(\theta_{\text{view}})$ as label. For m_*, Σ_{cl} and M_{core}, the labels are normalized by dividing all labels by the maximum value among the labels in the specific range of each magnitude used in the training or testing set.

After the normalization and pre-processing, the whole new data set is split into a training and validation set with a random shuffle and validation set size of 20% of the original. The training and validation sets are then separately converted into a class object that inherits from the PyTorch Dataset class in order to be fed to a PyTorch DataLoader, which will allow us to load the data in batches, reducing the computational cost.

Figure 3.1 shows the normalization and pre-processing applied to one image from the grid, alongside the distribution of the normalized pixel values. As can be observed, the distributions are imbalanced, with the highest frequencies at value 0, hence why the previous considerations regarding this value. When removing all the pixels with value 0 (right most panels) one can appreciate by eye that the distributions are close to gaussian.

3.2 Testing on SOMA data

In order to test the SOMA images on the network trained on the grid data, it is first necessary to rescale the former in distance and in coordinate increment per pixel so that they are equivalent to the latter. This is done by taking into account that the grid images are defined to be of sources at a distance of 1 kpc and their pixels represent 0.2778 arcsec. With this and the data from Table 2.1 and 2.2 the SOMA images were re-scaled accordingly.

Next, the rescaled SOMA images were normalized following the same process described for the syntheic data. This ensures that the input to the trained network when performing the test presents a similar distribution as that of the training data, with the aim to avoid domain shift problems, as no further domain adaptation techniques were applied in this study. Figure 3.2 shows the normalization and pre-processing applied to the SOMA images of the CepA source in the $19\mu m$ and $37\mu m$ bands used for the testing of the network.

3.3 Background subtraction

In the synthetic data, only the protostar is simulated and the pixel values of the background are set equal to 0. However, in the observational data this is not the case. Other sources of IR radiation in the background and foreground of the observed source affect the measurements performed by the instrument. Thus, background subtraction is a key step for the analysis of observational data, as it aims to isolate the radiation of the source one is concerned with, which would ideally make the observed system more similar to the simulations. Therefore, two background subtraction schemes were implemented.

The first scheme consisted of subtraction of a constant background. From the Turbulent Core Model ([2]) the radius of the core is given by:

$$R_{\rm core} \rightarrow 0.057 \left(\frac{M_{\rm core}}{60\ M_{\odot}} \right)^{1/2} \Sigma_{\rm cl}^{-1/2}\ {\rm pc}. \tag{3.3}$$

In the SOMA images of Cep A, this is equivalent of a core radius of approximately 42 arcsec. Defining an annulus centered on the protostar (Table 2.2) and of inner radius $R_{\rm core}$ and outer radius of $2R_{\rm core}$, the background value is calculated as the median pixel value inside the annulus. This constant value is subsequently subtracted from the entirety of the image.

The other method of background subtraction applied in this work starts by defining the same annulus as in the previous one. Then, it is divided in annular sectors of 30°. The pairs of radially opposite sectors are then utilized to define a linear gradient filter by using the two median pixel values inside each sector as the two extremes of the gradient, and the angle that the sectors form with the horizontal axis. This gradient is then applied only on the corresponding pairs of circular sectors in the circle inscribed in the annulus that correspond in orientation angle with the pair of annular sectors. This scheme is illustrated in Figure 3.3. This process is then iterated along all the pairs of annular sectors, applying the background subtraction to the whole inner circle of radius $R_{\rm core}$.

3.4 CASI-2D

The network architecture utilized in the present study is based on that of CASI (A Convolutional neural network Aproach for Shell Identification) presented in [27].

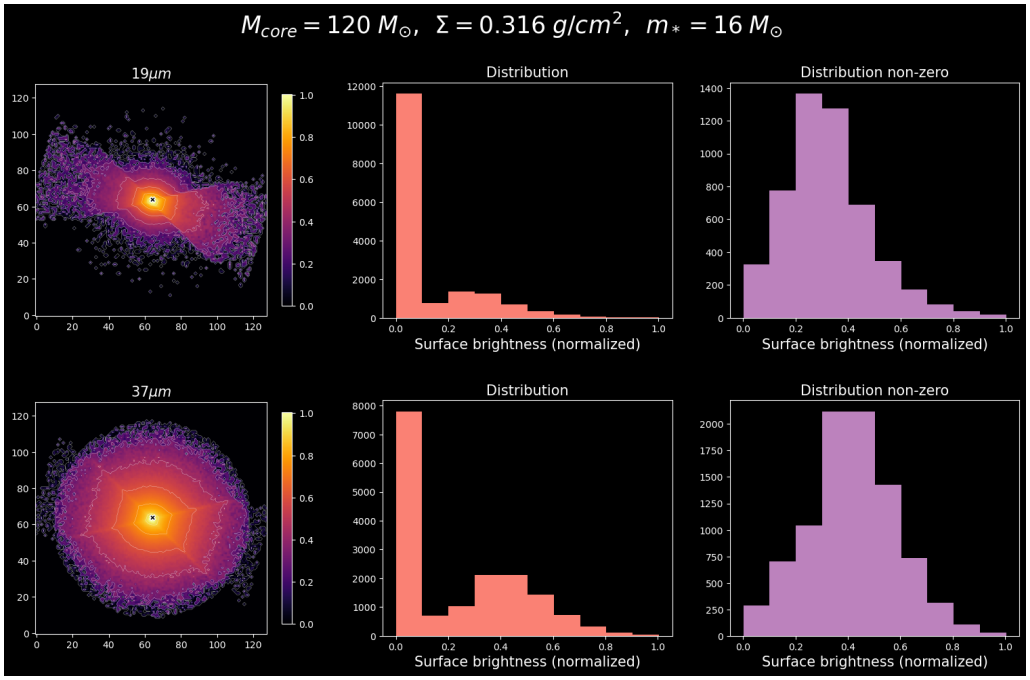

Figure 3.1: Example of one image of the grid in the 19μm and 37μm bands which has been normalized, cropped and rotated (left panels), histogram of the distribution of pixel values in the image, i.e. normalized brightness surface (middle panels) and histogram of the distribution of pixel values in the image which are non-zero (right panels).

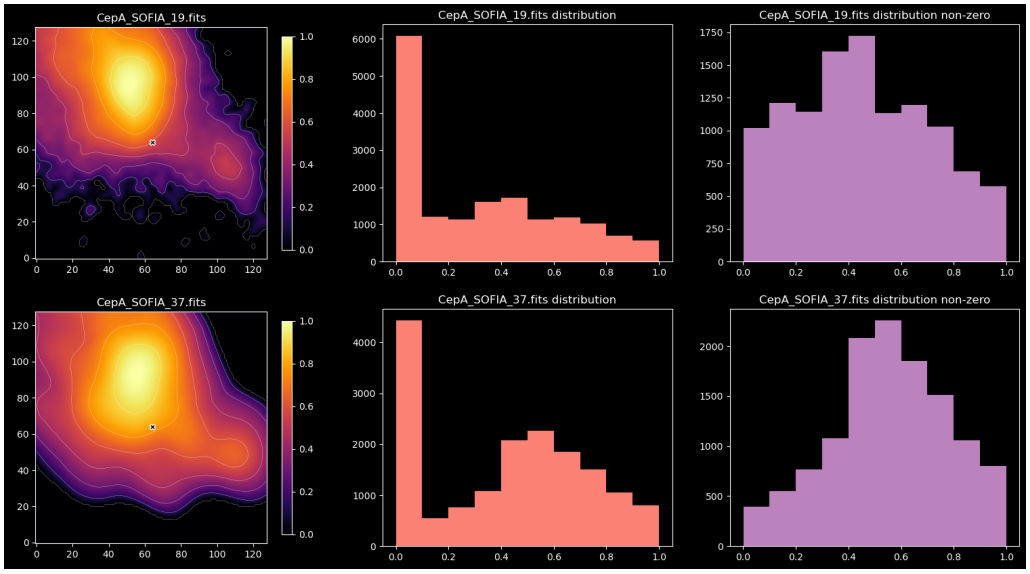

Figure 3.2: Analogous to Figure 3.1, images of Cep A in the 19μm and 37μm bands which have been normalized, cropped and rotated (left panels), histogram of the distribution of pixel values in the images, i.e., normalized surface brightness (middle panels) and histogram of the distribution of pixel values in the image which are non-zero (right panels).

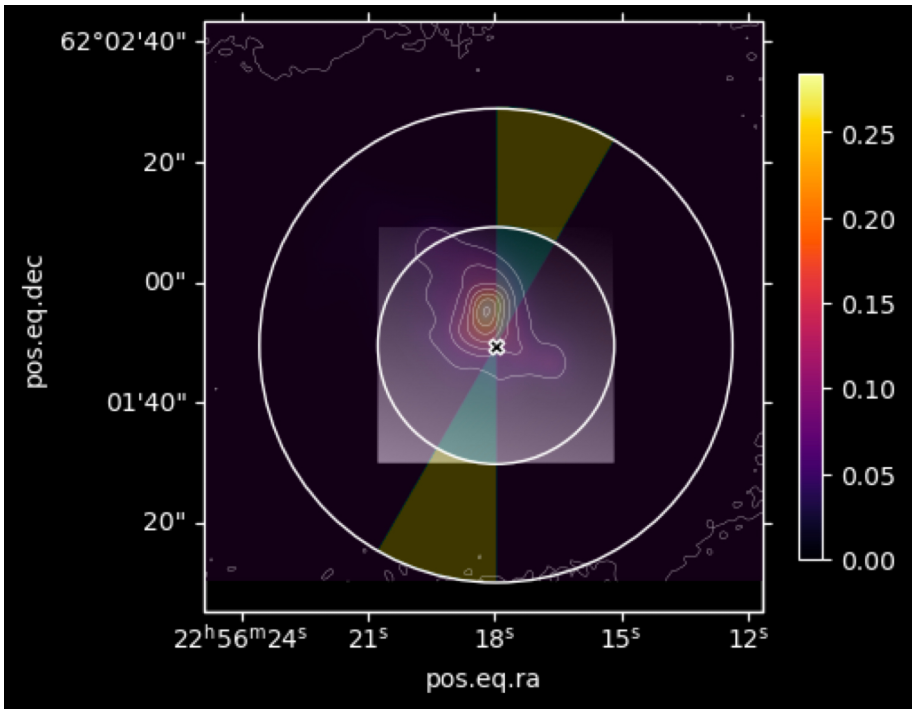

Figure 3.3: Illustration on the Cep A 37 μm SOMA image of one instance of the process of gradient background subtraction by sectors. Two radially opposed annular sectors are highlighted in yellow and the two corresponding circular sectors are highlighted in green, with the gradient applied to them superposed.

This model implements a deep neural network similar to U-Net for two different tasks: dense regression and image segmentation. Analogously to our case, CASI was also trained using synthetic data generated with magneto-hydrodynamic simulations and tested on observations, achieving good performance at both tasks.

For this study, the code provided in [27] was first translated from a Tensorflow to a PyTorch implementation, and adapted to the problem at hand.

As our only task is regression, the U-Net shape is not necessary, so only the encoder half was used (Figure 3.4). The fundamental building block of this architecture is a residual block [28], rather than a single convolution. It consists on a sequence of padded convolution operations applied consecutively in such a way that they conserve the dimensions of the input to the block. At the end of the sequence, the output is merged with the input to the block through a residual (or skip) connection. This scheme has been shown to improve performance in accuracy and to be easier to optimize [28]. The encoder is built by a concatenation of residual blocks and Instance Normalization and Max-Pooling layers. Instance Normalization was chosen, as opposed to the usual Batch Normalization, because, unlike the latter, Instance Normalization does not contain batch-size-dependent trainable parameters, which allows for testing on the single two channel SOMA image regardless of the batch size used for training. The number of sequences of these three elements in the concatenation is controlled by the *depth* parameter. Another parameter used to

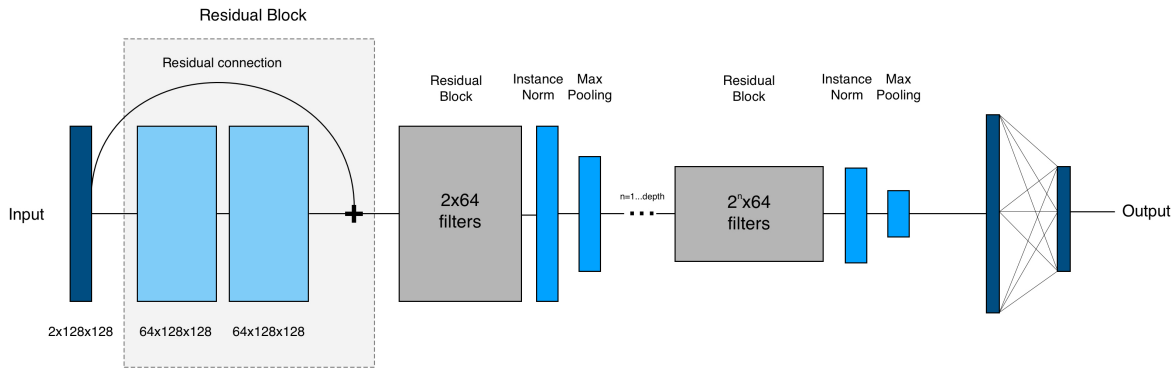

Figure 3.4: Diagram of the basic architecture of the CASI encoder, with a detailed diagram of the residual block, and the fully connected layers.

define the architecture is the number of *filters* of the convolutions in the first residual block. Then, the number of filters for consecutive blocks is doubled at each depth step.

Following the results from [29] suggesting better generalization properties, stochastic gradient descent (SGD) was chosen as the optimizer for the network. The optimizer is mainly controlled by two hyperparameters: the learning rate and the momentum intensity. The learning rate acts as a weight for the updates carried out to each of the trainable parameters in the network according to the optimization. The choice of order of magnitude of the learning rate requires finding a balance, as too high values will make the optimization diverge, while too low values will lead to premature convergence or unnecessarily long training times. Momentum consists in substituting the original update by a linear combination between the previous update and the current update, weighted by the momentum intensity parameter. It can be understood as a moving average of the gradients, reducing oscillations and promoting convergence.

The other parameter that controls the training of the network is the batch size. This parameter determines the amount of samples from the training set that are fed to the network in each training step. It can take values from 1 to the training set size. At the extreme of batch size of 1 results in updates at each sample, which introduces stochasticity to the optimization process. A training process with SGD and a batch size between the two extremes is called mini-batch SGD, which is the usual choice in the literature. The selection of the batch size presents a trade-off: larger batch sizes allow for more parallel calculations, reducing the calculation times and providing better gradient estimates resulting in faster convergence to an optimum. However, this also requires higher memory and computational costs. On the other hand, small batch sizes have been shown to have an effect of regularization and to provide an improvement at generalization. Every time training has been performed on all the batches of the data set, an epoch is finished. The number of epochs determines the duration of the overall training.

Table 3.1: Network Hyper-Parameters

Hyper parameter	Value
Batch size	8
Depth	5
Filters	64
Learning rate (initial)	1e-3
Noise σ	0.001
Epochs	200

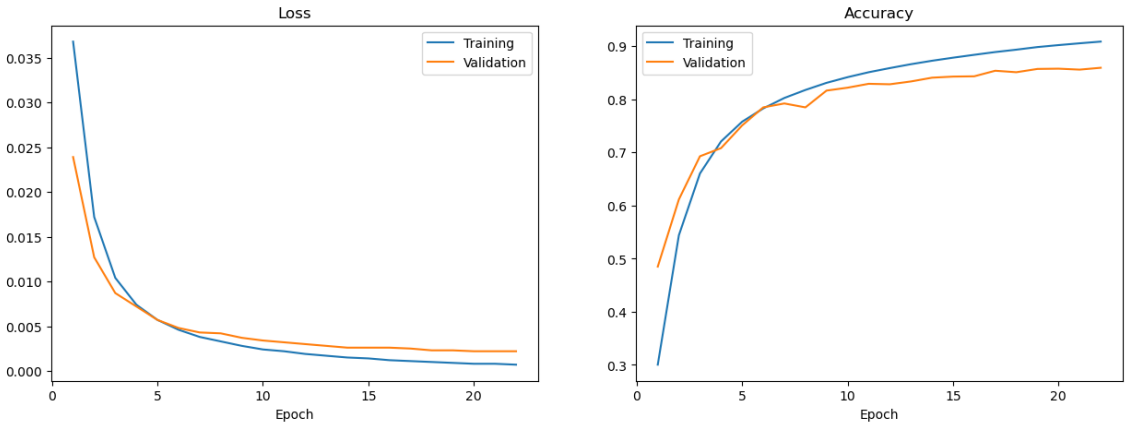

Figure 3.5: Loss and accuracy over 25 epochs for the training (blue) and validation (orange) sets during the training of the network with all the available data for the regression of M_{core}, Σ_{cl}, m_* and θ_{view}.

Additive Gaussian noise can be applied at the input by setting its specific standard deviation via the parameter σ. This can help the robustness of the model to small perturbations and reduce over-fitting.

Since hand-tuning alone produced satisfactory performance, an exhaustive hyper-parameter optimisation strategy was not carried out for this work. Table 3.1 contains the specific values utilized for the above mentioned hyper-parameters. Note that the learning rate was modified along the training with a multiplicative learning rate scheduler, which reduced its value by multiplying it by 0.5 every epoch with a number that is multiple of 50. The batch size, number of filters and initial learning rate values were chosen trying different combinations, finally these values achieved sufficient accuracy in the validation without compromising too long computation times. The number of epochs is similarly just large enough to verify that the loss has converged to a stable value.

An instance of training and validation loss and accuracy over 20 epochs of training can be observed in Figure 3.5.

4

Results

4.1 Testing on synthetic data

In order to test the hypothesis that a CNN can be utilized to regress the two physical variables m_* and $\theta_{\rm view}$ solely from the morphology observed in the images, a first approach was to verify that this was attainable on the testing data. To that end, a fiducial set of initial conditions was chosen ($M_{\rm core} = 60\ M_\odot$ and $\Sigma_{\rm cl} = 1 {\rm g\,cm^{-2}}$) on the basis that the different values of the target variables were visually discernible to some degree to the human eye merely from looking at the morphology of the simulated data corresponding to these values of $M_{\rm core}$ and $\Sigma_{\rm cl}$. This would suggest the ML algorithm could be able to achieve better precision than human labeling and also it would serve as an automated tool. Training and testing for each of the two target variables were carried out separately.

As another experiment, all the synthetic data (without initial condition restrictions) were used to train a modified version of the network with 128 initial filters and with the task of regressing the four variables: $M_{\rm core}$, $\Sigma_{\rm cl}$, m_* and $\theta_{\rm view}$.

4.1.1 Viewing angle

For this test, the network initialized with the hyper-parameters described in the previous chapter (Table 3.1) was trained on alternate inclination angles. In this way, the training set contained only the images with the aforementioned initial conditions and all the available range of m_* the labels of which corresponded only to 10 out of the 20 different values of $\theta_{\rm view}$ comprised in the synthetic data. The other 10 were saved as testing data.

Figure 4.1 shows the simulated images in the 19 μm band for a protostar with these initial conditions and evolutionary stage $m_* = 24\ M_\odot$ with the aim of showcasing the differences of morphology at different viewing angles.

Figure 4.2 shows the result of this test. It is evident from the plot in the left panel that the predictions closely follow the true value-prediction identity line, indicating that the regression task for $\theta_{\rm view}$ can be deployed successfully. Also, there is not an apparent difference between the dispersion of the predictions of the set of angles that the network was trained on with respect to the ones that were excluded. There

is, however, some dispersion in the results. In order to study this dispersion, the offset of each of the predictions with respect to the true value was calculated and the distribution of the offsets is plotted in the right panel of Figure 4.2. The mean offset is $\mu_{\delta(\theta)} = -0.43°$ with a standard deviation of $\sigma_{\delta(\theta)} = 4.45°$. This mean offset is relatively small, especially when compared with the value intervals between the discrete labels, so these results can be deemed acceptable and the regression of θ_{view} from the morphology of the simulated protostars successful.

One final observation in this test is that the offset is more apparent at the two extremes of the range. This could be explained by the fact that the variability in morphology is most evident in the middle of the range (see Figure 4.1), while the images close to the extremes present a more uniform shape. Also, with regards to the extreme of low angles, due to the non-linear nature of the transformation of the data labels ($\cos(\theta_{\text{view}})$), even if the labels were equally spaced, the translation into degrees makes this range of viewing angles less populated in terms of labeled data, making the regression more uncertain.

4.1.2 Protostellar mass

A similar experiment was carried out for the regression of m_*. In this case, one mass value ($m_* = 16\ M_\odot$) in the range available in the grid for the fiducial initial conditions was excluded from the training set, while the test set contained all the available evolutionary stages. In this way, the intention is to probe how the network will regress a previously unseen m_* value contained in the range of the training set.

Figure 4.3 from [17] shows different simulated evolutionary stages (m_*= 1, 2, 4, 8, 12, 16 and 24 $M_\odot$ at the initial conditions $M_{\text{core}} = 60\ M_\odot$ and $\Sigma_{\text{cl}} = 1\text{g cm}^{-2}$ viewed at an angle of 60° and at different wavelengths. Although not as obvious as with θ_{view}, each evolutionary stage presents a distinct morphology. Notably, the outflow cavity widens as m_* increases. This will allow the CNN to discern between the labels of this physical magnitude as well.

Figure 4.4 shows the results of the test described above. Similarly to the test for inclination angle, the predictions for m_* also follow the identity line defined with the true labels. As one might expect, the predictions for the excluded mass have a larger dispersion, but it is not significantly larger than those of the other labels. In contrast with the previous case, it is the extreme values of the range that are predicted the most accurately.

Regarding the offset, the distribution has mean $\mu_{\delta(m)} = -0.69\ M_\odot$ and standard deviation $\sigma_{\delta(m)} = 2.31\ M_\odot$, although the distribution is slightly skewed to the left, which might signify that the network tends to slightly overestimate the value of m_*, since the offset is calculated as $\delta(m) = m_{*,\text{true}} - m_{*,\text{pred}}$. These are still overall acceptable offsets relative to the spacing of the labels and, therefore, the network is able to regress the protostellar mass from the morphology observed in the images.

The previous tests have shown that, at least when treating with the synthetic images, it is possible to regress θ_{view} and m_* with a certain degree of accuracy. Would

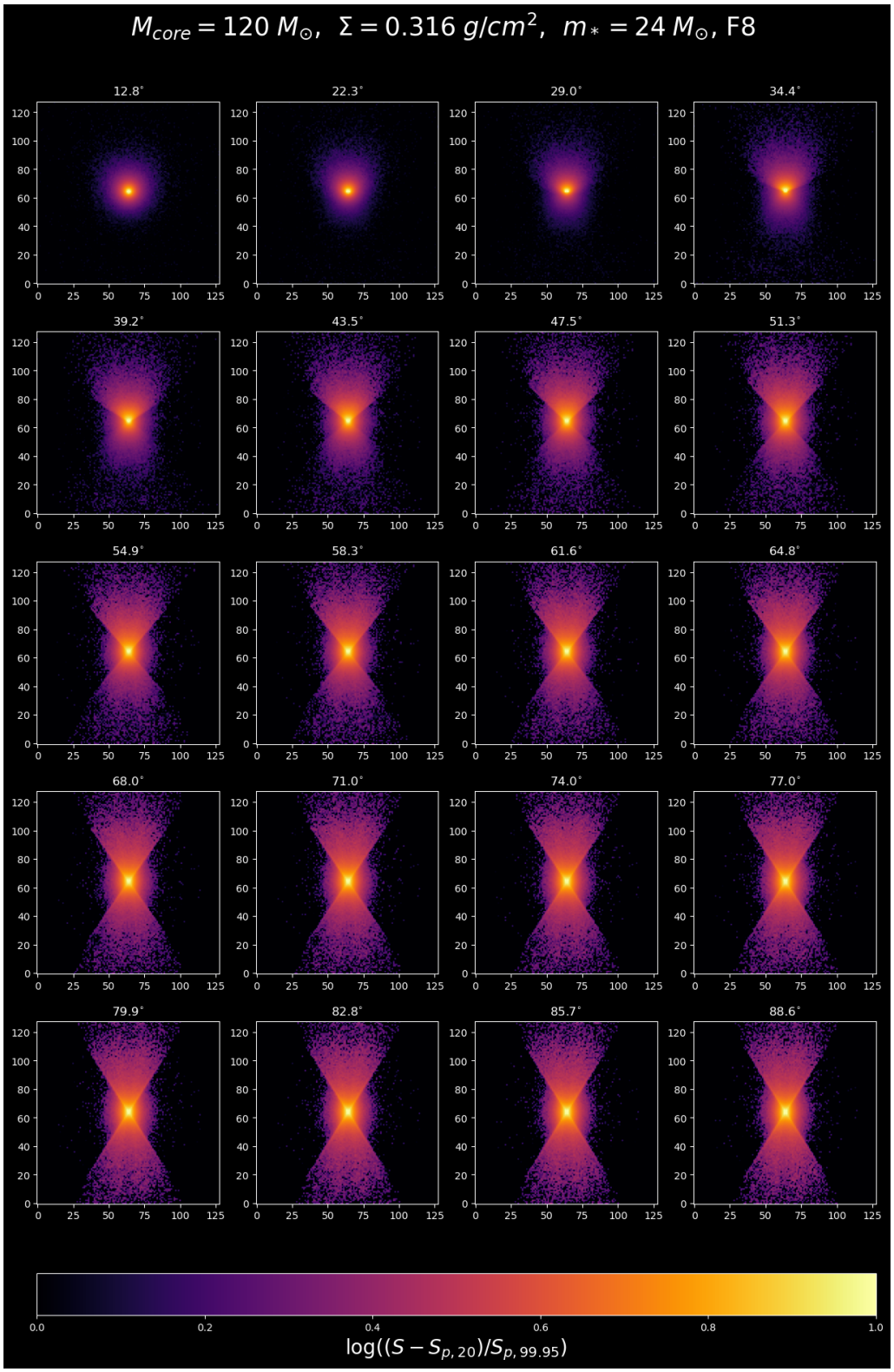

Figure 4.1: Grid of simulated images for a protostar with $M_{\rm core} = 120\ M_\odot$ and $\Sigma_{\rm cl} = 0.316 {\rm g\,cm^{-2}}$ at $m_* = 24\ M_\odot$ at 20 different $\cos(\theta_{\rm view})$ with a filter at the 19μm band, showcasing the variability in morphology at different values of $\theta_{\rm view}$.

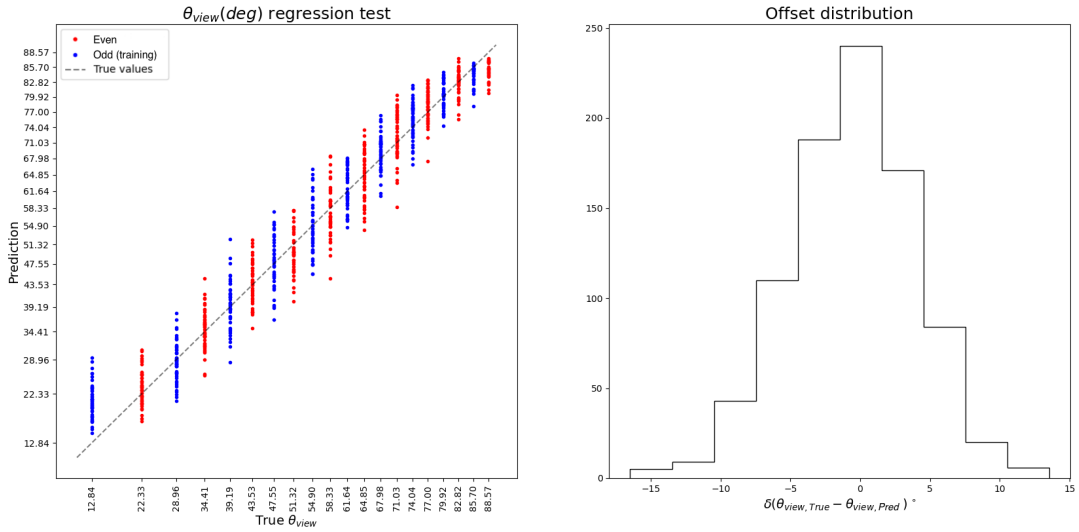

Figure 4.2: Test result for the regression of $\theta_{\rm view}$ by the network trained on images with initial conditions $M_{\rm core} = 60\ M_\odot$ and $\Sigma_{\rm cl} = 1 {\rm g\,cm^{-2}}$ and odd angles (blue) and tested on both odd and even (red) in the left panel, and distribution of the offsets of the prediction with respect to the true value shown in the right panel, with $\mu_{\delta(\theta)} = -0.43°$ and $\sigma_{\delta(\theta)} = 4.45°$

one want to achieve better precision, it might be attained with more training time, different hyper-parameter selection or more sophisticated architectures. Even during the exploration phase in this work, accuracies of up to 99% in classification tasks were achieved on the synthetic data. However, this type of task was not the main objective of this work and such high accuracies do not translate into better or more certain results as it will be discussed in the testing on SOMA observational data, below.

4.1.3 Training on all data

A separate experiment was carried out in which all 8640 synthetic images were used as training data, with no restrictions on the initial conditions. In this case, the task was to regress all four variables: $M_{\rm core}$, $\Sigma_{\rm cl}$, m_* and $\theta_{\rm view}$. To this end, the number of filters was increased to 128, but the rest of the hyper parameters were kept the same as the previous cases.

Figure 4.5 shows the distributions of the offsets between the predictions for $\Sigma_{\rm cl}$ and $\theta_{\rm view}$ and their true values, and the relative offset distributions of $M_{\rm core}$ and m_* with respect to their true values. Additionally, the offset distributions for the validation of the previous two experiments are superposed in blue for comparison in the cases of m_* and $\theta_{\rm view}$. The mean μ, and standard deviation, σ, of each distribution are also shown in each panel.

As can be observed, all distributions are centered close to 0, suggesting that on average the predictions are near the true values. The asymmetry in both mass

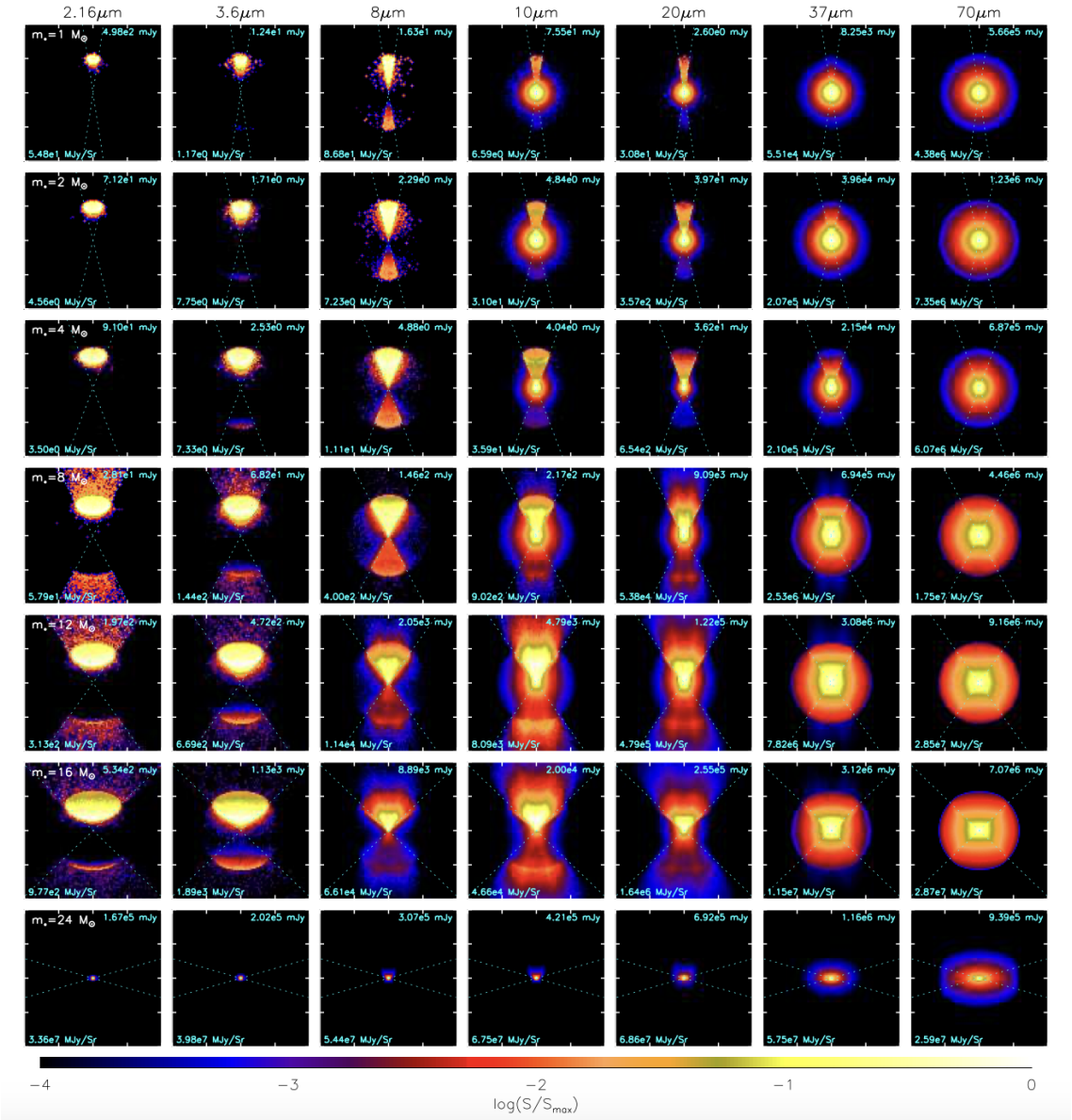

Figure 4.3: From [17]. Resolved images at different wavelengths for several evolutionary stages ($m_* = 1, 2, 4, 8, 12, 16,$ and $24 M_\odot$ from top to bottom) of initial conditions $M_{\rm core} = 60\ M_\odot$ and $\Sigma_{\rm cl} = 1\,{\rm g\,cm^{-2}}$ viewed at an inclination angle of 60°. Each image is normalized to its maximum surface brightness, which is labeled in the lower-left corner.

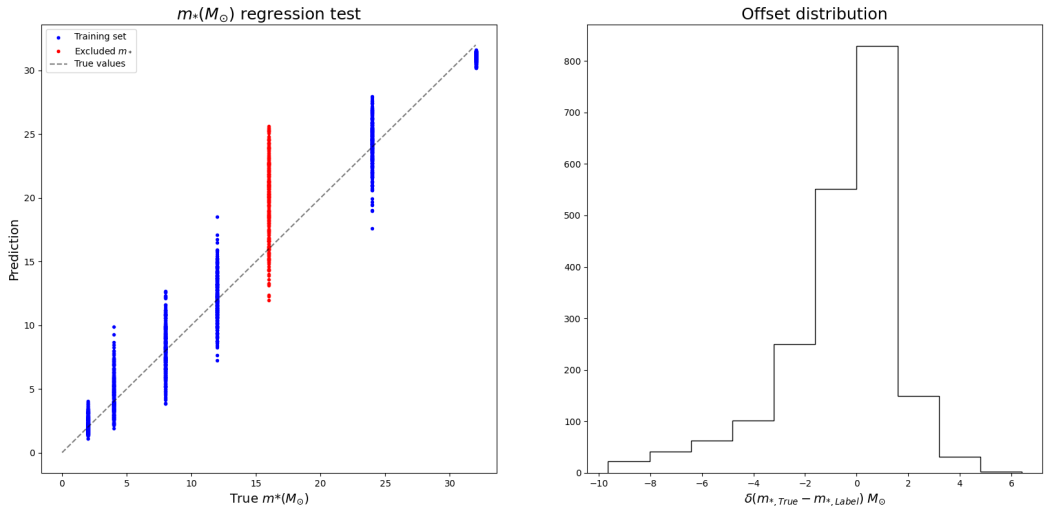

Figure 4.4: Test result for the regression of m_* by the network trained on images with initial conditions $M_{\rm core} = 60\ M_\odot$ and $\Sigma_{\rm cl} = 1\ {\rm g\ cm^{-2}}$ with $m_* = 16\ M_\odot$ (red) excluded from the training and tested on all protostellar masses, and distribution of the offsets of the prediction with respect to the true value shown in the right panel, with $\mu_{\delta(m)} = -0.69\ M_\odot$ and $\sigma_{\delta(m)} = 2.31\ M_\odot$

.

magnitudes indicates that the network tends to overestimate this values with respect to the ground truths. With regards to the width of the distributions, for the m_* offset distribution it is comparable to that of the previous experiment, and so is the one for $M_{\rm core}$. When it comes to $\Sigma_{\rm cl}$, the standard deviation is similar in size as the smallest interval between two of the label values: 0.1 and 0.316 g cm^{-2}. Finally, upon inspection of the comparison between the offset distributions of the predictions of $\theta_{\rm view}$ for the different tests, it is apparent that there has been a loss in precision, as the distribution for the restricted set network is narrower than that of the current experiment.

It is worth mentioning that it is known that some of the simulated data do not present distinct morphologies at different values of the model's variables, yet all the available images were used for training. This might hinder the training of the network and better accuracy could be achieved by manually selecting the images of the grid that present discernible changes in morphology in the selected ranges of values for each of the variables.

4.2 Testing on SOMA observational data

The main objective of this work is to develop a tool that will analyze observational data of massive protostars and regress their $\theta_{\rm view}$ and m_* in an automated manner. To that end, the last experiment that is carried out in this chapter is to test the network on the Cep A image from the SOFIA Massive (SOMA) Star Formation Survey.

A first step to perform this test was to retrain the network with synthetic images corresponding to a different set of initial conditions. More precisely, $M_{\rm core} = 120\ M_{\odot},\ \Sigma_{\rm cl} = 0.316\ {\rm g\,cm^{-2}}$, which more closely matches the results obtained by spectral energy distribution fitting by [25]. When doing this and testing on the synthetic data, the results were analogous to those from the previous sections, but in order to effectuate the testing on the SOMA image some further considerations are needed.

An established problematic in machine learning algorithms is that of the domain shift [30] and many efforts have been done to overcome it. When the training data and the test data are not drawn from the same distribution, the test performance can be negatively affected. This might be the case with the Cep A image. Upon visual inspection it is apparent that it does not have such a clear morphology as its synthetic counterparts.

This can pose a problem on the interpretability of the results. Although promising, the results obtained when inputting the image to the trained network are not unequivocally reliable. In fact, when an image of uniform noise was input through the trained network, the results were not discernable from what would be a result to a real source.

To overcome this problem, a different training scheme was implemented. Negative samples were added to the training sets in the form of empty images (all pixel values equal to 0) and images with random uniformly distributed pixel values with random noise intensity. They were assigned the extreme values of 0 and 1 and the labels of synthetic images were distributed along that range. Also, since it is not needed to do further tests on the synthetic data, no specific label was excluded from the training.

Figures 4.6 and 4.7 show the results of testing the network trained with this new scheme with a validation set of the synthetic data. As can be observed, the introduction of the negative samples has presented a trade-off, where a highly accurate distinction of the zeros and noise images suppose an increase in dispersion for the regression of the labels of the real images. However, they still follow the identity line and the offset distributions are comparable.

With the network being able to very accurately predict whether the input is noise, if the Cep A images do not get classified as such, the prediction will be considered reliable for the purposes of this work. Tables 4.2 and 4.1 compile the results of the regression of each of the variables of our interest. The Cep A image is fed raw and with the two background subtraction methods to the network 36 times applying a rotation of 10° each time. The results are expressed as:

$$x = \mu_{\rm pred} \pm \sqrt{\left(\sigma^2_{pred} + \mu^2_{\delta}\right)}, \tag{4.1}$$

where $\mu_{\rm pred}$ is the mean value predicted throughout the 36 predictions, $\sigma_{\rm pred}$ is the standard deviation, and μ_δ the mean offset of the network. Although there are discrepancies of around 10% with the results from [25] and up to 30% with the

results of m_*, the ranges of error of both overlap.

Also, the results do not seem to be much affected by the background subtraction, which could be expected as the pixel values comprised in the annulus are very low compared with the pixel values that conform the protostar.

Table 4.1: Results for the regression of m_* on the Cep A image, compared with the results from [25]

	m_* $(M_\odot)$
Fedriani et al. (2023)	16^{+6}_{-5}
Raw image	25 ± 6
Constant Background	24 ± 6
Sector Background	25 ± 6
Noise (label)	0.90 ± 0.03
Zeros (label)	0.003 ± 0.001

Table 4.2: Results for the regression of θ_{view} on the Cep A image, compared with the results from [25]

	$\theta_{\text{view}}(^\circ)$
Fedriani et al. (2023)	71 ± 12
Raw image	79 ± 9
Constant Background	79 ± 9
Sector Background	80 ± 10
Noise	87.4 ± 1.3
Zeros	3.0 ± 0.4

A more complete tool would result from the separate experiment performed by training on all data in order to obtain an estimation for all four variables of the grid. However, it was not possible to overcome the domain shift problems when testing on the SOMA images in this case. Improvements on this regard are left for future work.

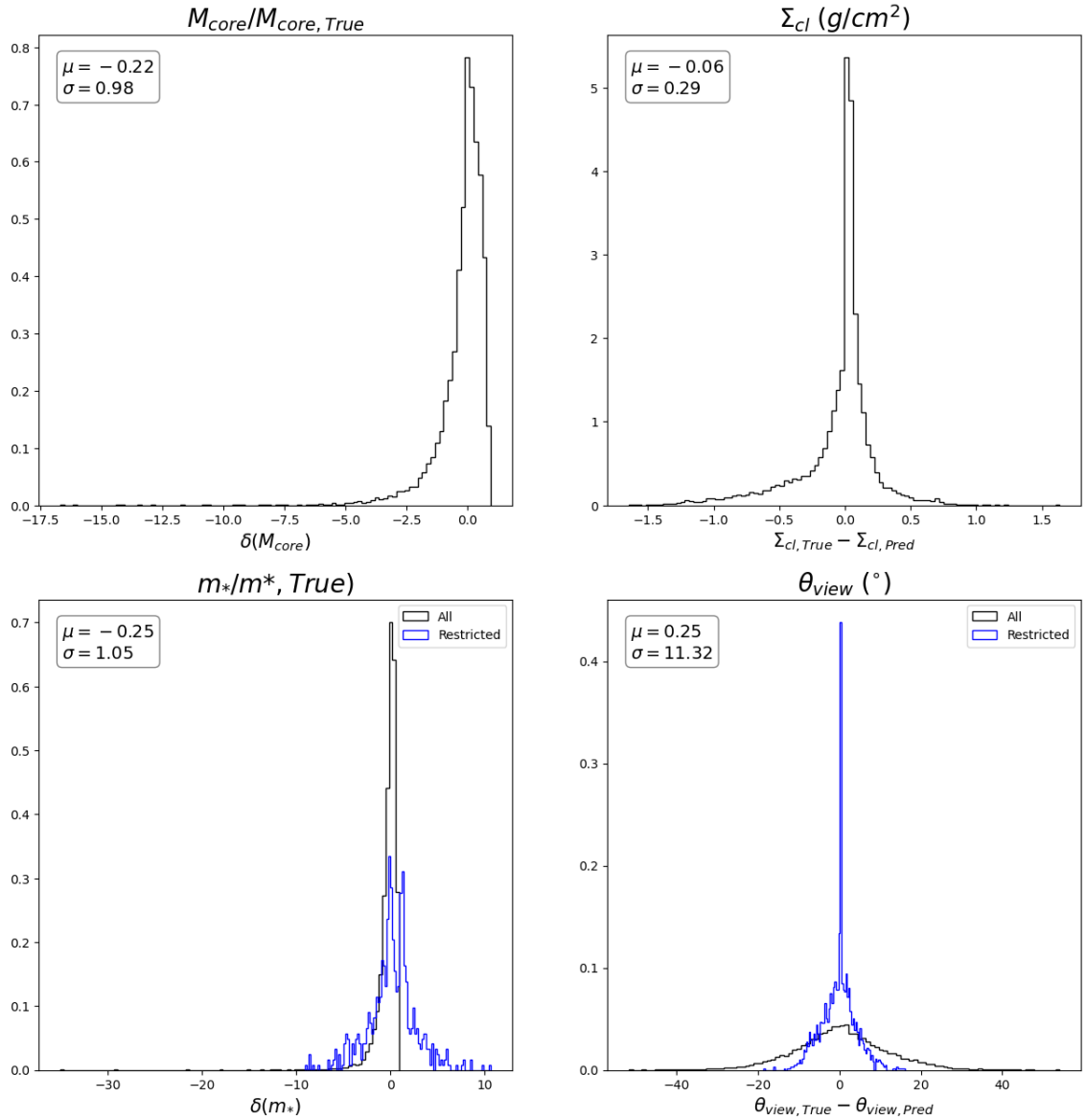

Figure 4.5: Offset distributions of the predictions for M_{core}, Σ_{cl}, m_* and θ_{view} with respect to their true values (black). In blue, offset distributions for the prediction of m_* and θ_{view} with the restricted initial conditions trained networks. The offsets for M_{core} and m_* are relative to the true value.

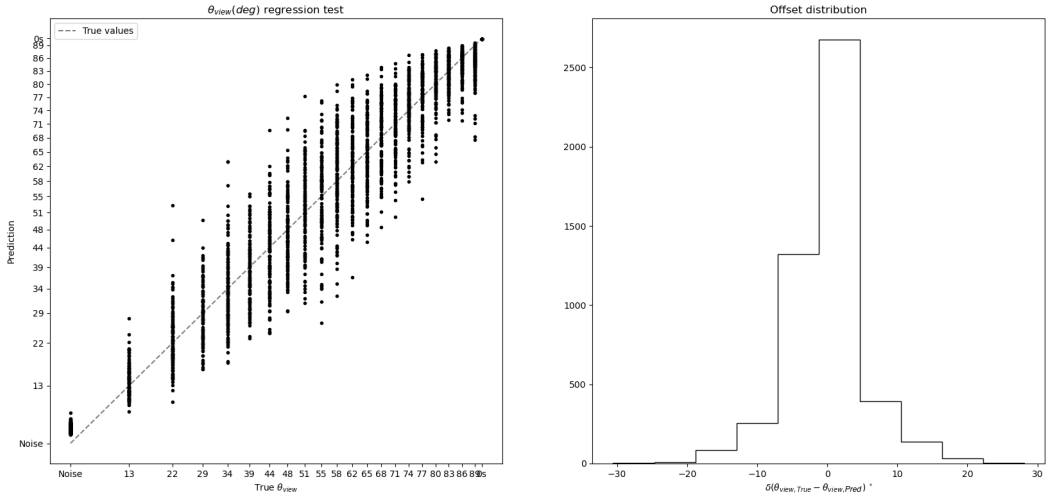

Figure 4.6: Regression of θ_{view} with negative samples of noise (labeled as $\cos(90) = 0$) and zeros (labeled as $\cos(0) = 1$) in the left panel and offset distribution of the predictions in the right panel.

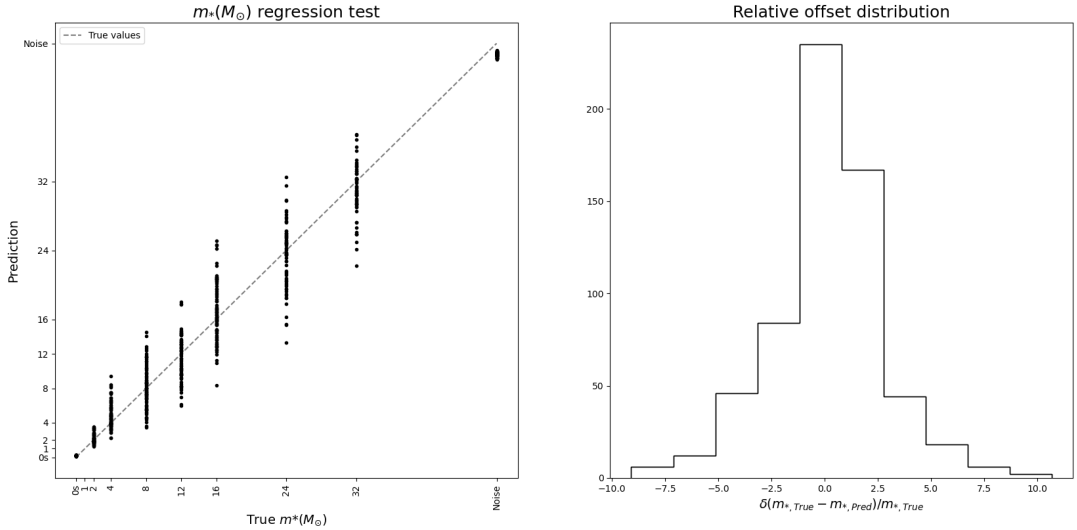

Figure 4.7: Regression of m_* with negative samples of noise (labeled as 0) and zeros (labeled as 64 $M_\odot$) in the left panel and offset distribution of the predictions normalized by the true mass in the right panel.

5

Conclusions

In this study it has been shown that, starting from synthetic images of massive protostars generated by Monte Carlo Radiative Transfer simulations following the Turbulent Core Model of massive star formation, it is possible to implement Convolutional Neural Network architectures trained on these images to successfully regress some of their key physical features, namely, their angle of inclination with respect to the line of sight, θ_{view} and their protostellar mass, m_* which defines their evolutionary state. It has also been investigated that better performance can be achieved through modifications in the architecture or hyper-parameter tuning. Additionally, the performance of the network was explored in classification tasks and it was found to have an excellent accuracy of up to 99% in case that the continuous value regression of these variables would not be necessary.

On the other hand, applying the trained network to observational data from the SOFIA Massive Star Formation Survey of the protostar Cepheus A was not so straightforward. Possible domain shift problems and lack of robustness at handling noise required for the network to be retrained adding negative samples (noise and zeros). The new model was highly precise at discerning the negative samples, but at the cost of decreasing slightly the precision of the regresion of the labels. The results when feeding the observed system agree within the range of error with the results found in the literature for the same source found with other methods, i.e., spectral energy distribution fitting, which is promising. However, the settings for the training required some previous knowledge of these results, which is not ideal for implementing this method in other sources.

All things considered, this method has the potential to be highly useful for the analysis of observational data. Specially if one manages to achieve more similarity between the training and testing data, either by further pre-processing the data by, for example, applying convolutions of the optical profile of the instrument used for the observations, or by obtaining better quality observations, for instance with new facilities such as the JWST, the results will be more reliable. For this project in particular, immediate further developments should include implementing a more sophisticated architecture in order to regress all 4 parameters that define the model grid instead of merely one at a time, which would provide much better generalization for measurements in observed systems without previously known data and the implementation of transformer architectures for better performance.

Bibliography

[1] C. F. McKee and J. C. Tan, "Massive star formation in 100, 000 years from turbulent and pressurized molecular clouds," *Nature*, vol. 416, no. 6876, pp. 59–61, Mar. 2002. DOI: 10.1038/416059a. [Online]. Available: https://doi.org/10.1038/416059a.

[2] C. F. McKee and J. C. Tan, "The formation of massive stars from turbulent cores," *The Astrophysical Journal*, vol. 585, no. 2, pp. 850–871, Mar. 2003. DOI: 10.1086/346149. [Online]. Available: https://doi.org/10.1086/346149.

[3] Y. Zhang and J. C. Tan, "Radiation transfer of models of massive star formation. IV. the model grid and spectral energy distribution fitting," *The Astrophysical Journal*, vol. 853, no. 1, p. 18, Jan. 2018. DOI: 10.3847/1538-4357/aaa24a.

[4] A. Kembhavi and R. Pattnaik, "Machine learning in astronomy," *Journal of Astrophysics and Astronomy*, vol. 43, no. 2, Oct. 2022. DOI: 10.1007/s12036-022-09871-2. [Online]. Available: https://doi.org/10.1007/s12036-022-09871-2.

[5] A. D'Isanto and K. L. Polsterer, "Photometric redshift estimation via deep learning," *Astronomy &amp Astrophysics*, vol. 609, A111, Jan. 2018. DOI: 10.1051/0004-6361/201731326.

[6] T.-Y. Cheng, N. Li, C. J. Conselice, A. Aragón-Salamanca, S. Dye, and R. B. Metcalf, "Identifying strong lenses with unsupervised machine learning using convolutional autoencoder," *Monthly Notices of the Royal Astronomical Society*, vol. 494, no. 3, pp. 3750–3765, Apr. 2020. DOI: 10.1093/mnras/staa1015. [Online]. Available: https://doi.org/10.1093/mnras/staa1015.

[7] M. Walmsley, L. Smith, C. Lintott, *et al.*, "Galaxy zoo: Probabilistic morphology through bayesian CNNs and active learning," *Monthly Notices of the Royal Astronomical Society*, vol. 491, no. 2, pp. 1554–1574, Oct. 2019. DOI: 10.1093/mnras/stz2816. [Online]. Available: https://doi.org/10.1093/mnras/stz2816.

[8] D. Xu, S. S. R. Offner, R. Gutermuth, and C. V. Oort, "Application of convolutional neural networks to identify stellar feedback bubbles in CO emission," *The Astrophysical Journal*, vol. 890, no. 1, p. 64, Feb. 2020. DOI: 10.3847/1538-4357/ab6607.

[9] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959. DOI: 10.1147/rd.33.0210.

[10] T. Mitchell, *Machine Learning* (McGraw-Hill series in computer science), en. New York, NY: McGraw-Hill Professional, Mar. 1997.

[11] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943. DOI: 10.1007/BF02478259.

[12] B. Mehlig, *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers*. Cambridge University Press, 2021, ISBN: 9781108849562.

[13] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8.

[14] D.-X. Zhou, "Universality of deep convolutional neural networks," *Applied and Computational Harmonic Analysis*, vol. 48, no. 2, pp. 787–794, 2020, ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2019.06.004.

[15] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, Dec. 1989. DOI: 10.1007/bf02551274. [Online]. Available: https://doi.org/10.1007/bf02551274.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[17] Y. Zhang, J. C. Tan, and T. Hosokawa, "Radiation transfer of models of massive star formation. iii. the evolutionary sequence," 2013. DOI: 10.48550/ARXIV.1312.3370. [Online]. Available: https://arxiv.org/abs/1312.3370.

[18] M. Y. Grudić, D. Guszejnov, S. S. R. Offner, *et al.*, "The dynamics and outcome of star formation with jets, radiation, winds, and supernovae in concert," *Monthly Notices of the Royal Astronomical Society*, vol. 512, no. 1, pp. 216–232, Mar. 2022, ISSN: 0035-8711. DOI: 10.1093/mnras/stac526. eprint: https://academic.oup.com/mnras/article-pdf/512/1/216/42894596/stac526.pdf. [Online]. Available: https://doi.org/10.1093/mnras/stac526.

[19] I. A. Bonnell, M. R. Bate, and H. Zinnecker, "On the formation of massive stars," *Monthly Notices of the Royal Astronomical Society*, vol. 298, no. 1, pp. 93–102, Jul. 1998. DOI: 10.1046/j.1365-8711.1998.01590.x. [Online]. Available: https://doi.org/10.1046/j.1365-8711.1998.01590.x.

[20] J. C. Tan, M. T. Beltran, P. Caselli, *et al.*, "Massive star formation," 2014. DOI: 10.48550/ARXIV.1402.0919. [Online]. Available: https://arxiv.org/abs/1402.0919.

[21] B. A. Whitney, R. Indebetouw, J. E. Bjorkman, and K. Wood, "Two-dimensional radiative transfer in protostellar envelopes. III. effects of stellar temperature," *The Astrophysical Journal*, vol. 617, no. 2, pp. 1177–1190, Dec. 2004. DOI: 10.1086/425608. [Online]. Available: https://doi.org/10.1086/425608.

[22] B. A. Whitney, T. P. Robitaille, J. E. Bjorkman, *et al.*, "THREE-DIMENSIONAL RADIATION TRANSFER IN YOUNG STELLAR OBJECTS," *The Astrophysical Journal Supplement Series*, vol. 207, no. 2, p. 30, Jul. 2013. DOI: 10.1088/0067-0049/207/2/30.

[23] T. L. Herter, J. D. Adams, G. E. Gull, *et al.*, "FORCAST: A mid-infrared camera for SOFIA," *Journal of Astronomical Instrumentation*, vol. 07, no. 04, Dec. 2018.

[24] J. M. D. Buizer, M. Liu, J. C. Tan, *et al.*, "The SOFIA massive (SOMA) star formation survey. i. overview and first results," *The Astrophysical Journal*, vol. 843, no. 1, p. 33, Jun. 2017. DOI: 10.3847/1538-4357/aa74c8. [Online]. Available: https://doi.org/10.3847/1538-4357/aa74c8.

[25] R. Fedriani, J. C. Tan, Z. Telkamp, *et al.*, "The SOFIA massive (SOMA) star formation survey. IV. isolated protostars," *The Astrophysical Journal*, vol. 942, no. 1, p. 7, Dec. 2022. DOI: 10.3847/1538-4357/aca4cf.

[26] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on Nuclear Science*, vol. 44, no. 3, pp. 1464–1468, 1997. DOI: 10.1109/23.589532.

[27] C. M. V. Oort, D. Xu, S. S. R. Offner, and R. A. Gutermuth, "CASI: A convolutional neural network approach for shell identification," *The Astrophysical Journal*, vol. 880, no. 2, p. 83, Jul. 2019. DOI: 10.3847/1538-4357/ab275e.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[29] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/81b3833e2504647f9d794f7d7b9bf341-Paper.pdf.

[30] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, no. 1, pp. 151–175, 2010. DOI: 10.1007/s10994-009-5152-4. [Online]. Available: https://doi.org/10.1007/s10994-009-5152-4.

DEPARTMENT OF SPACE, EARTH AND ENVIRONMENT
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY