The predictive power of convolutional neural networks in Astrophysics as a discovery tool

Master's thesis in Physics

at Friedrich-Alexander-University Erlangen-Nürnberg

presented at August 28, 2020

by **Jonas Geyer-Ramsteck**

Supervisor: Prof. Dr. Manami Sasaki

Erklärung zur Masterarbeit

Hiermit erkläre ich, dass ich die Masterarbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.

Erlangen, August 28, 2020 Jonas Geyer-Ramsteck

Zusammenfassung

In dieser Arbeit haben wir ein vortrainiertes Convolutional Neural Network, das auf der VGG16 Architektur basiert, genutzt. Das Netzwerk wurde angepasst um Bubbles und Bubble-ähnliche Strukturen im Interstellaren Medium der Großen Magellanschen Wolke (GMW) und um Fanaroff-Reiley I (FRI) Galaxien anhand ihrer Morphologie zu detektieren.

Zur Detektion von Bubble-ähnlichen Strukturen wurde das Netzwerk mit einem Set von lediglich 83 dieser Strukturen trainiert, die manuell aus den Daten des Southern H-Alpha Sky Survey Atlas (SHASSA) selektiert wurden.

Das so trainierte Netzwerk wurde auf Beobachtungsdaten von SPITZER angewendet, um weitere Bubble-ähnliche Strukturen zu identifizieren, die dann als erweitertes Training-set verwendet wurden.

Das finale Model des Netzwerks wurde auf schmalbandige Bilder der GMW aus der Magellanic Cloud Emission Line Survey (MCELS) angewendet und fand 456 Bubble-ähnliche Strukturen in Hα, 288 in [OIII] und 267 in [SII].

Die Verteilung von Bubbles wurde mit der Verteilung von massereichen Sternen aus dem Bonanos et al. (2009) Katalog, und mit HI Shells und Supershells, Assoziationen, Sternhaufen und Emissionsnebeln aus dem allgemeinen Katalog von ausgedehnten Objekten in der GMW von Bica et al. (2008) verglichen. Die Korrelation der Verteilungen wurde mithilfe von Ripleys K Funktion (Ripley 1981) analysiert.

Zusätzlich wurde ein weiteres Convolutional Neuronal Network mit 340 FRI Objekten trainiert, die manuell in den Pilot-Beobachtungen des neuen Australian Square Kilometre Array Pathfinder (ASKAP) gefunden wurden. Das so angepasste Netzwerk wurde auf ASKAP Daten der GMW angewendet und fand insgesamt 186 FRI Kandidaten. Ein großer Teil der detektierten FRI Objekte konnte mit bereits bekannten extragalaktischen Objekten und Radioquellen assoziiert werden.

Ein vortrainiertes Netzwerk kann, mit Hilfe von Data Augmentation, bereits mit wenig initialen Trainingsdaten zu einem ersten Model führen. Dieses Model kann auf unbekannte Daten angewendet werden, um Kandidaten für die gesuchten Objekte zu identifizeren, die anschließend manuell verifiziert werden. Korrekt klassifizerte Objekte können dann als erweitertes Trainingset verwendet werden. Dieser Prozess kann solange wiederholt werden, bis die Leistung des Netzwerks zufriedenstellend ist.

In beiden Fällen haben wir mit sehr wenigen Trainingsdaten begonnen. Diese Arbeit zeigt, dass es selbst mit dieser geringen Zahl an Daten möglich ist, ein gut funktionierendes Convolutional Neural Network zu erzeugen.

Abstract

In this work we used a pretrained convolutional neural network based on the VGG16 network to detect bubbles and bubble-like structures in the interstellar medium of the Large Magellanic Cloud (LMC), as well as Fanaroff-Riley I (FRI) galaxies by their morphology.

For the detection of bubble-like structures a small training set of only 83 bubble-like structures was manually selected from data from the Southern H-Alpha Sky Survey Atlas (SHASSA). The trained network was applied to SPITZER data, and identified additional bubble-like structures that served as additional training data. The final model of the network was applied to narrow-band images from the LMC from the Magellanic Cloud Emission Line Survey (MCELS) and found 456 bubble-like structures in Hα, 288 in [OIII] and 267 in [SII]. The distribution of bubbles was compared to the distribution of massive stars from the Bonanos et al. (2009) catalog, HI shells and supershells, associations, star clusters, and emission nebulae from the general catalog of extended objects in the LMC by Bica et al. (2008). The correlation between the distributions was studied using Ripleys K function. A significant correlation was found between bubbles and massive stars, and between bubbles and emission nebulae.

Additionally a neural network based on the VGG16 was trained on 340 manually labeled FRI objects from the new Australian Square Kilometre Array Pathfinder (ASKAP) pilot survey of the Emu sky region. The trained network was applied to ASKAP data from the LMC and found a total of 186 FRI objects. A huge amount of the detected FRI galaxies can be associated to already known extragalactic objects and radio sources.

A pretrained network and data augmentation allows to generate a first model which, applied to new data, yields additional new training data. After manually evaluating this additional training data the network can be trained again on the larger set. This can be repeated until the performance of the network is satisfying.

For both cases we started with very few training samples. This study shows, that even with such a small amount of initial training data it is possible to create a well performing convolutional neural network.

Contents

1 Motivation

In 2019 the Space Telescope Science Institute (STScI) in conjunction with the University of Hawai'i Institute for Astronomy published over 1.6 petabytes of data in one batch, which was gathered in a period of four years as part of the largest digital sky survey Pan-STARRS - the Panoramic Survey Telescope and Rapid Response System. This is equivalent to 30.000 times the total text content on Wikipedia or two billion selfies NASA & ESA (2019). It is an tremendously huge amount of images with an incredible amount of information stored in it. Yet it is only one of many ongoing surveys of the sky and only a small part of the already gathered information from previous observations. The data volume of entire surveys from a decade ago can be nowadays obtained in a single night and the capability of the observation equipment is increasing rapidly. Data volumes of this size obviously can not be handled manually by individual scientists anymore and, therefore, modern Astronomy requires more and more sophisticated automatized data analysis methods. A rough roundup of the increasing gathered data volume from different sky surveys is depicted in Table 1. For a variety of scientific questions this immensely huge amount of data is actually not a problem but rather an advantage, since scientists can focus on specific information of the survey. On the other hand, if someone is ,e.g., interested in the amount of some specific recurring objects in the sky one would have to look through all these images. This is obviously not feasible. There were approaches like, for example, delegating this problem to a citizen science project where many non-scientific people voluntarily analyse images by comparing the images to a given exemplary data-set. A more efficient and maybe more reliable option is the usage of image recognition algorithms. In the recent years the field of machine learning and deep learning in particular developed rapidly. In this work, a convolutional neural network based on the famous VGG16 by Karen Simonyan and Andrew Zisserman is used in order to automatize the search for bubbles and bubble-like structures within astronomical survey data, especially in optical data from the Large Magellanic Cloud.

Survey	Approximate Data Volume
DPOSS (The Palomar Digital Sky Survey)	3 TB
2MASS (The Two Micron All-Sky Survey)	10 TB
GBT (Green Bank Telescope)	20 PB
GALEX (The Galaxy Evolution Explorer)	30 TB
SDSS (The Sloan Digital Sky Survey)	40 TB
SkyMapper Southern Sky Survey	500 TB
PanSTARRS	≈ 40 PB expected
LSST (The Large Synoptic Survey Telescope)	≈ 200 PB expected
SKA (The Square Kilometer Array)	≈ 4.6 EB expected

Table 1: Estimated Data Volume of different Sky Surveys Zhang & Zhao (2015).

2 Deep Learning Introduction

2.1 Neural Networks

The above mentioned citizen science projects are projects where amateur scientists are asked to evaluate given data like ,e.g., images of regions in the sky based on a set of criteria that were predefined by professionals. An exemplary task for such a project is to identify structures within images that are similar to a given set of samples. Imagine you are given images (Figure 1a) as examples of the objects to search for and are asked to identify this kind of objects in a wider region of the sky as it is shown in Figure 1b.

You most certainly will identify at least the marked areas and maybe also more. You will also probably be able to order the found areas by their likelihood with your training set. Even though this task seems straightforward for humans this simplicity is compelling. Trying to put your decision for this likelihood in words and, in addition to that, write it into an algorithm should reveal that it is not that easy after all. Of course, the obvious property is the circular shape but the extension is unique for every example, the borderline is hard to generalize, and every example is unique in itself.

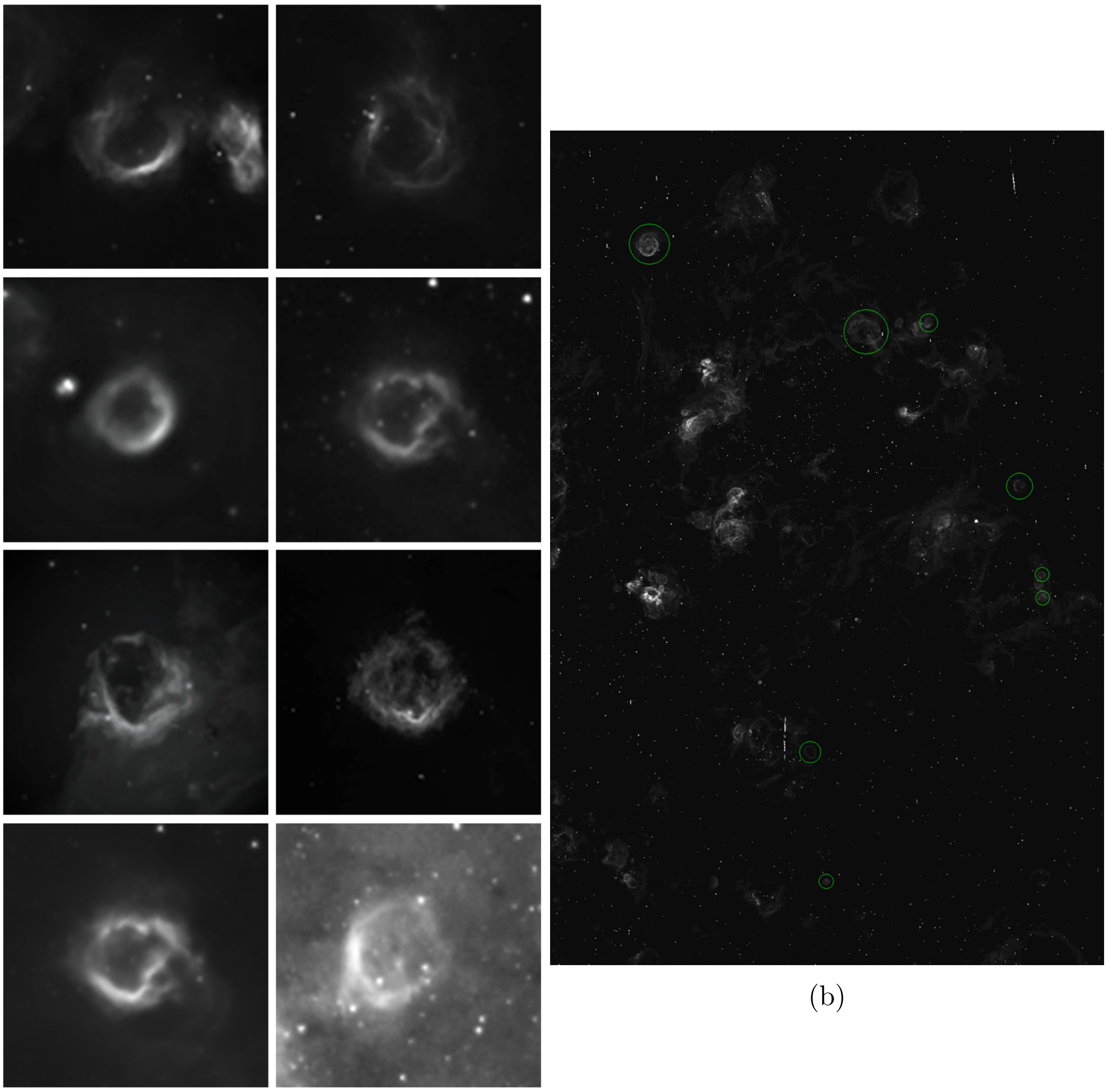
While there are some specific tasks computers are ultimately more efficient to do, like e.g. linear algebra tasks on your calculator or most other linear well defined problems, the human brain has its advantages in its incredible versatility. The primary visual cortex contains more than 140 million neurons allowing you to gather more than 10 million bits of information per second and filter them for relevant information in a heartbeat Markowsky (2017). The primary idea of Artificial Neural Networks is to recreate this competence in an algorithm. Just like the human brain, artificial neural networks also consists of billions of connected neurons. Of course, the nature of artificial and biological neurons differ, yet their working principle is strongly related.

2.1.1 The Perceptron

The first kind of artificial neuron was already introduced in the 1950s and 1960s by Frank Rosenblatt. A very simple illustration of a perceptron is depicted in Figure 2a. The perceptron takes a number of scalar inputs and has a binary output. In this case there are three input variables to this perceptron. Each of them can be imagined as a decision criterion that contributes differently to the final result of the neuron. Applied to our citizen science example these inputs could be:

1. ”Is it brighter than the surrounding?”
2. ”Is it pancake shaped?”
3. ”Is it toroidal shaped?”

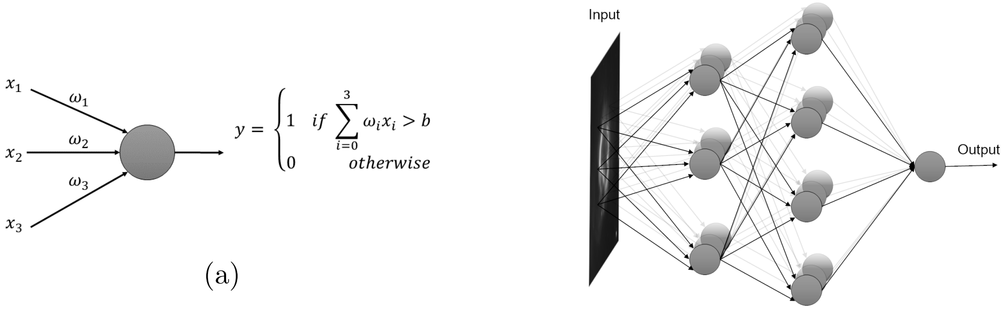
You could assume that criterion 2 and 3 exclude each other but if you look closely



(b)

(a)

Figure 1: Exemplary task for a citizen science project: Understand the concept of the objects in the left image (a) and transfer that concept in order to find similar-looking regions in a wider area of the sky like in the right image (b).


x_1
ω_1
x_2
ω_2
ω_3
x_3
$y = \begin{cases} 1 & if \sum_{i=0}^{3} \omega_i x_i > b \\ 0 & otherwise \end{cases}$
Input
Output

(a)

(b)

Figure 2: An example for a Rosenblatt Perceptron and its decision making process (a) and a simple neural network where every neuron of every layer in the network is connected with all neurons in the next layer (b). Each connection has a certain weight and every neuron an associated bias. This network allows to create complex decision criteria. Inputs to that network could be the pixel values of an image. These values are also connected to every neuron of the next layer. Note that for a better presentability not all connections are drawn.

the transition is actually diffuse. The output of the perceptron would be the classification if some object is a bubble or bubble-like structure or not. Imagine you could choose a value x_i between zero and one in order to evaluate a given image by these questions. Each of these values is multiplied with an individual weight w_i defining the importance of the criterion. The sum of the weighted criteria is compared to a offset bias b. If this bias is exceeded the neuron outputs High/One, otherwise Low/Zero. Essentially, the bias is a measure of how easy it is to activate the neuron.

$$y = \begin{cases} 1 & \text{if } \sum_{i=0}^{3} \omega_i x_i > b \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

So, for example, if an object is by far brighter than its surrounding you can assign $x_1 = 0.9$ but on the other hand it is only slightly toroidal or pancake shaped so the values for $x_2 = 0.1$ and $x_3 = 0.2$ are rather small. The decision if something is considered a bubble or bubble-like is then dependant on how each of these criteria is weighted and what threshold has to be exceeded. If the importance of the toroidal shape is high but it should also contrast to the surrounding the corresponding weights have to be high, e.g. $w_1 = 10$ and $w_3 = 8$, while the pancake shape is not so important $w_2 = 2$. For a threshold of $b = 12$ the perceptron will decide that the given object is not a bubble because: $10 \cdot 0.9 + 2 \cdot 0.1 + 8 \cdot 0.2 = 10.8 < 12$. However, if the object had been slightly more toroidal or pancake-like it would have been classified as bubble. You can see that while the inputs x_i for each individual image won't change we have free parameters w_i and b, which we can adjust in order to define our decision-making. Of course, this exemplary decision is already at a high level since you need an evaluation of its shape and brightness. Yet it illustrates how a perceptron weighs up different kind of evidences in order to make decisions.

2.1.2 Fully Connected Layers

Actually the upper example for a perceptron is only the last high-level step of a chain of such decision-making perceptrons as it can be seen in Figure 2b. Any input is connected with every neuron of the first layer and every neuron of the first layer is connected to every neuron of the next layer and so on. Each connection has a certain weight and every neuron is associated with a bias. This ultimately allows for an increasing complex and abstract decision-making with every layer. These fully connected layers are also called dense layers.

For arbitrarily many neurons every layer can be written as a vector product

$$y = \begin{cases} 1 & \text{if } \vec{\omega}\vec{x} + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

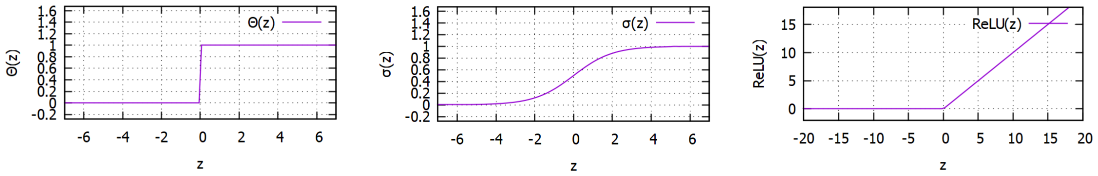
and in order to ease further computations we changed a little detail to our threshold $b \to -b$. Since this is only an arbitrary value we can switch it's sign without any change to the concept. So far we have only discussed networks with one final result y. However, if we have more neurons in our last layer the network has that many outputs and $y \to \vec{y}$ the output becomes a vector with that many entries.

2.1.3 Activation function

In order to discuss the learning part of a neural network we have to add an important detail to our neuron.

As already mentioned in Section 2.1.1 we have weights and biases as free parameters which we can adjust in order to define our decision-making. We can evaluate their impact on the final decision of the network by varying them slightly. We then will need to see a small but significant change in the output. For example if the network mistakenly classifies an object as bubble even though it is not bright or circular shaped enough, we can change the weights and biases again and again until the output of the networks converges towards the desired output. If we stick with the Rosenblatt perceptron this is not possible. The perceptron only has a binary output and any change to the weights or biases will either result in no change at all or make the neuron flip from one state to the other. It is impossible to tell if you are coming closer to the desired output or if you are moving away from it further and further. So what we need is a continuous slope in the output for the neuron - at least in the part that we want to evaluate. The function which defines the activation condition of the neuron is called the activation function. So far we essentially used the Heaviside function (Figure 3a):

$$\Theta(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$



(a) Heaviside function (b) Sigmoid function (c) ReLU function

Figure 3: Different activation functions responsible for the decision making of every neuron

The first calculation of the neuron is

$$z = \vec{\omega}\vec{x} + b \tag{4}$$

This is then evaluated by the activation function in order to decide if the neuron *fires*. Over time many different activation functions have been developed. We will only discuss those, which are the most important to this work.

Sigmoid activation

As the name already hints the sigmoid activation function is the sigmoid function (Figure 3b).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{5}$$

It is essentially a smoothed out version of the Heaviside function. This smoothness allows to see a change in the output for any small changes to the weights and biases. However, for very large or very small values of z the sigmoid function converges against zero or one which can result to the so-called vanishing gradient problem. For these values small changes to z actually result in extremely small changes in the output which happen to be below the machine precision. Nevertheless, the sigmoid was used for a long time as the default activation function.

Rectified linear unit activation

To overcome the vanishing gradient problem the Rectified Linear Unit or $ReLU$ is often used (Figure 3c)

$$ReLU(z) = \begin{cases} z & \text{if } z > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

This function essentially combines two important properties of an activation function: It is partially linear and therefore very easy to evaluate for changes to our free parameters. But it also has a non-linearity at zero, which allows decision making. The function is also not limited by zero and one, which avoids the saturation of the function as it happens with the sigmoid. The $ReLU$ is nowadays the most commonly used activation function.

2.1.4 Classification

Depending on the task of the neural network we have different options to implement the final classification of our input. In our example the classification is binary - either it is a bubble or it is not. This corresponds to a sigmoid activation function in our last layer. However, there are also multi-label tasks where the network should learn to classify the input to multiple categories. In this case our final activation function is the softmax function:

Softmax
The softmax function essentially rescales an input vector $\vec{x}$:

$$\hat{y}_k = \frac{\exp(x_k)}{\sum_{j=1}^{K} \exp(x_j)} \tag{7}$$

This essentially introduces a normalized probability distribution. For every output neuron the output value is normalized using all other parallel output neuron values.

2.1.5 Loss-function

A non-binary activation function allows us to evaluate the impact of a change to the weights and biases of a neuron on the output of the neuron. But we still need a way to evaluate if our neuron output is actually coming closer to the desired result. Therefore we need to introduce a Loss-function

$$L(\vec{x}, \vec{y}, \vec{\omega}, b) = \frac{1}{2}||f(\vec{x}, \vec{\omega}, b) - \vec{y}||_2^2 \tag{8}$$

Here $\vec{y}$ is the desired output for the input $\vec{x}$. The function $f(\vec{x}, \vec{\omega}, b)$ represents the neural network and has a certain output dependant on its weights, biases and the input. The Loss is then defined as the square of the absolute value of the difference between desired output and actual output.

For a neural network with arbitrarily many layers this function is of course depending on all weights and biases of the network

$$L(\vec{x}, \vec{y}, \vec{\omega}, b) \rightarrow L(\vec{x}, \vec{y}, \vec{\omega_1}, \vec{\omega_2}, \vec{\omega_3}, ..., b_1, b_2, b_3, ...)$$

In order to improve the outcome of our network we want to minimize the outcome of the Loss function under the alternation of the weights and biases of the network.

For multi-label problems we have to use a different loss function. In this case the cross-entropy loss is used

$$L(\vec{x}, \vec{y}, \vec{\omega}, b) = -log\left(\frac{\exp(x_k(\vec{\omega}, b))}{\sum_{j=1}^{K} \exp(x_j(\vec{\omega}, b))}\right)|_{y_k=1} \tag{9}$$

If the amount of training data is unbalanced between the labels we can introduce an additional weight that is applied to the loss for the different labels. For example if there are two times more images for one class than for the other class the loss also has to be doubled. On the other hand we can favor a certain class by weighting the loss for this class higher. The network then trains to predict this class easier.

2.1.6 Training

Without further assumptions for any activation function σ, the impact of a change to the free parameters of a network can be calculated. The output of a single neuron is given by

$$\sigma = \sigma(\vec{x}, \vec{\omega}, b) \tag{10}$$

Altering the weights $\vec{\omega}$ and the biases b results in a change of the output. The total derivative is

$$\Delta\sigma(\vec{x}, \vec{\omega}, b) = \sum_i \frac{\delta\sigma(\vec{x}, \vec{\omega}, b)}{\delta\vec{\omega}_i}\Delta\omega_i + \frac{\delta\sigma(\vec{x}, \vec{\omega}, b)}{\delta b}\Delta b \tag{11}$$

which is a linear function of the changes $\Delta\omega_i$ and Δb.

For multiple layers of neurons the input of a neuron of the layer i is actually the output from different previous neurons and therefore $\vec{x} \rightarrow \sigma(\vec{x}, \vec{\omega}_{i-1}, b_{i-1})$ can be repeatedly applied until the first layer is reached and $\vec{x}$ is the input vector.

The last step of the calculation is the Loss-function. So what we are actually evaluating is the gradient of the Loss-function with respect to the weights and the biases of our network. For simplicity ω and b represent all existing weights and biases inside the network with arbitrary many layers. When we know this gradient we can update our weights and biases in a way that will reduce the Loss:

$$(\omega^{k+1}, b^{k+1}) = (\omega^k, b^k) - \eta\nabla_{\omega,b}L(\vec{x}, \vec{y}, \vec{\omega}, b) \tag{12}$$

with $\nabla_{\omega,b} = (\frac{\delta}{\delta\omega_1}, ..., \frac{\delta}{\delta\omega_n}, \frac{\delta}{\delta b_1}, ..., \frac{\delta}{\delta b_n})$. This is iterated until the gradient converges to zero. η is also called the learning rate. It defines how strong the weights are adjusted along the gradient. Note that the loss function is actually a chained function of all layers of the network, e.g., $L(Layer_i(Layer_{i-1}(Layer_{i-2}(...))))$. This essentially can be imagined as a multidimensional gradient descent depicted exemplary in Figure 4 for a Loss-function that is only dependent on two weights. For a real neural network this gradient descent would happen in k-dimensions with k being the amount of free parameters - the weights and biases - of the network. Since we can not evaluate the loss function for every set of parameters we need to stick to a few certain values determined by the training data. Calculating the gradient descent in several steps is also called a stochastic gradient descent. The update of these parameters is not done for every single sample but for a particular amount of samples given by the batch size. This allows to include a variety of different features into the calculation of the loss. When every sample of the training data were used once to train the network one epoch of the data was processed.

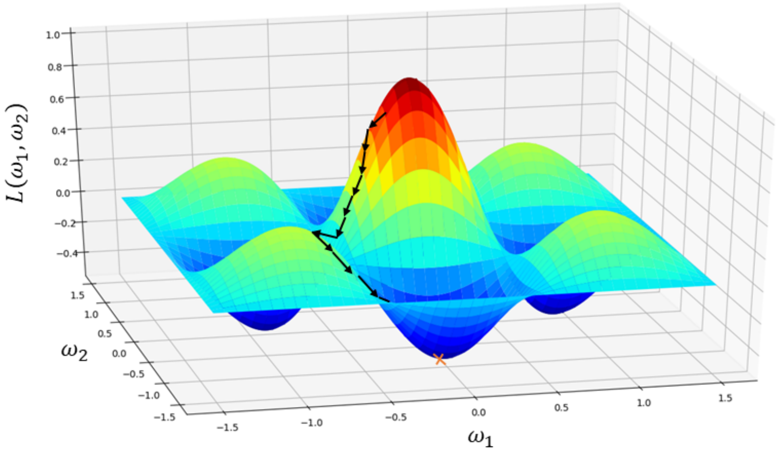


Figure 4: Gradient descent in two dimensions. The Loss function depends only on two weights ω_1 and ω_2. The plotted Loss function is actually not known like depicted here but is rather evaluated for every training step at the position of the black arrow-origin. The gradient then allows us to update the weights along the direction that the Loss function decreases most - along the black arrows. This procedure is iterated until the gradient converges to zero.

In general it can not be guaranteed that the stochastic gradient descent finds the global minimum. However, since the dimensionality is very high, it is assumed that there are many local minima which result in a similar small Loss. The learning rate has a big impact on the gradient descent. If it is chosen too high the change of weights can overshoot, while a too small learning rate will be very inefficient. There are different approaches to optimize the learning rate, e.g., high in the beginning, small the closer it gets to a minimum. The used optimizer in this work is the RMSprob.

Optimizer - RMSprop

When updating the weights of the network it occurs that some features are activated very infrequently while others are updated very often. In order to account for that it is beneficial to introduce individual learning rates for every parameter in the network. This can be implemented by updating the weights according to the following equations with the element-wise multiplication $\odot$ and the gradient $g^{(k)}$

$$g^{(k)} = \nabla L(\vec{\omega^{(k)}}) \tag{13}$$

$$r^{(k)} = \rho r^{(k-1)} + (1-\rho)(g^{(k)} \odot g^{(k)}) \tag{14}$$

$$\omega^{(k+1)} = \omega^{(k)} - \frac{\eta}{\sqrt{\vec{r}^{(k)}} + \eta} \odot g^{(k)} \tag{15}$$

With the value $r^{(k)}$ we essentially calculate the weighted mean square of the gradient from the last iteration and the current iteration. This process averages the gradients that are used to update our weights over successive mini-batches. With this value we finally can update our weights. This optimization of the learning process is called RMSProb Hinton (n.d.).

2.1.7 Dropout Layer

While training it happens that some connections between layers becoming more and more important than others. This means the classification concentrates on few features which can result in the problem that the network does not generalize to the problem very well. To avoid this dropout layers can be used while training. They randomly block connections in the network to force the feature learning to distribute more over all connections. For normal operation the dropout layer is deactivated.

2.1.8 Data Augmentation

For many problems that are approached with self learning algorithms, the amount of data that is needed to train a network efficiently is actually the limitation. One very useful method to approach this problem is data augmentation. All samples that are used as training data can be altered slightly to create additional training data. For example you could introduce a small amount of noise to the image and still expect the network to recognize it just like the original one. Of course it is mandatory that the augmented object is still recognizable. The kind of data augmentation therefor is very dependant on the problem. While the rotation of our bubbles or bubble-like structures does not alter the concept of the object, a rotation of a cat is problematic. In the end it is still a cat but the network also learns repeating properties like the spatial orientation or location. If these properties are unwanted features because the object is actually spatially invariant, the augmentation of the object to different spatial properties will teach the network to disregard this property. In the case of bubbles and bubble-like structures we can use the following augmentation possibilities:

Rotation Range The rotation range defines the rotation that can be applied for augmentation. Since bubbles are circular structures they can be rotated at will.

Width Shift Range The width shift range defines how far the image can be shifted in width. This is limited by the fact that a too high value would shift the object out of the image.

Height Shift Range The height shift range defines how far the image can be shifted in height. This is limited by the fact that a too high value would shift the object out of the image.

Brightness Range The brightness range defines how strong the brightness of the image can be scaled. In general we can use this but with caution. A too high value would distort the fundamental physical properties and may introduce new features that are not generalizable.

Zoom Range The zoom range defines the enlargement factor that can be used. A too high value can enlarge the bubble to a point where the boundaries are not inside the image anymore. A too small value can shrink the bubble to a point source.

Horizontal Flip The horizontal flip defines if the image can be mirrored horizontally. In our case this is no problem and does not distort any information.

Vertical Flip The vertical flip defines if the image can be mirrored vertically. In our case this is no problem and does not distort any information.

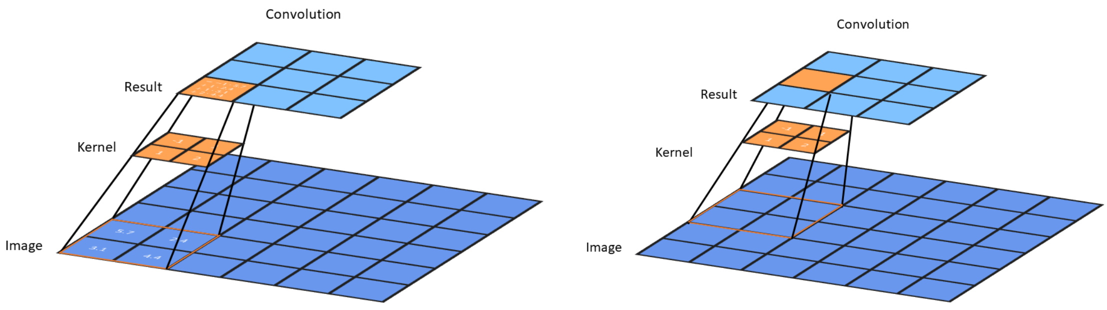
2.2 Validation

After the network is trained, it has to be tested for its performance. For a quantitative validation it is necessary to use a part of the training data and separate it. A usual fraction is 20%. These samples are not allowed to be used during the actual training. When the network is applied to this validation data the outcome of the network is compared to the actual known labels and the performance can be calculated.

At some point if the chosen network has a very high capacity there is a chance that the network starts to internalize the complete set of training data and achieves a very high performance on the training set. However, the network does not necessarily generalize to other data. Therefor it is useful to monitor the performance on the validation set after every epoch. If the performance on the validation set decreases repeatedly while the performance on the training set increases the training should be stopped. After this early-stop the weights are then restored to the state with the best validation performance.

2.3 Convolutional Neural Networks

So far we have only discussed inputs to the neural network as some kind of a vector which is distributed to every neuron in the first layer. This can also be done with images. For example you could use an image with 32×32 pixels and concatenate all pixel rows which would result in a 1024×1 input vector. For a few problems this works in general, but it is actually very inefficient. If you take a look on the examples of our citizen science project in Figure 1a again you can see that the larger part of each picture is actually black and doesn't have any impact on the concept of the object. This means that from our 1024×1 input vector only a very small amount is actually interesting. A way to analyze images with a neural network more efficient are convolutional neural networks. The concept of convolution for images



(a) First convoluted value

(b) Second convoluted value

Figure 5: Convolution of an Image. A kernel with given weights is applied to the image. The values of the pixels within the kernel reach are multiplied with the kernel weights and summed up while the kernel is walking over the whole image resulting in a new convoluted image. The increment of the kernel movement is called stride and is 2 × 2 pixels in this example.

is depicted in Figure 5a and was first described by Kunihiko Fukushima in 1980 Fukushima (1980).

The idea is, to analyze correlated pixel-areas. Therefore an arbitrary big kernel is used, which is essentially an array of weights. This array moves step-by-step over the whole image. At every position the pixel values are multiplied with the associated weight of the kernel and all weighted pixels are then summed up. This results in a new value for one pixel in the convoluted image. Convolution reduces the image size by $2 \cdot s \cdot \lfloor n/2 \rfloor$ with kernel size n and stride s which is the increment of the kernel movement.

For the example in Figure 5 you can see that the kernel has size 2×2 with weights $-1, 1, 1, 2$ while the image pixels in reach of the current kernel position are $5.7, 2.4, 3.1, 4.4$. The convoluted value then is e.g.

$$conv_{11} = -1 \cdot 5.7 + 1 \cdot 2.4 + 1 \cdot 3.1 + 2 \cdot 4.4 = 8.6$$

The weights of this kernel can be adjusted and trained and allows the network to work out different kernels for different features inside of the image. This could be for example a kernel that scans for perpendicular edges, or curvatures exemplary shown in Figure 6. Each of these kernels results in a different convoluted image, which is then essentially a heat map where straight lines or curvatures in the original image are depicted. This heat map can then be flattened to a one dimensional vector and fed into a fully connected layer. This layer than has the positional information about these features in the image as an input. These features do not only have to be spatially distributed. The kernel can have a depth which means not only spatial correlation is calculated but also channel correlation. For an RGB image this would be the channels red, green and blue and essentially allows to correlate color in the image. A kernel with size $1 \times 1 \times 3$ for example does not regard any spatial correlation but only color.

Convolution allows us to find correlated pixel-areas very efficiently. The weight and the size of the kernel define the way of this correlation. The common practice is to use more than 30 kernels at every convolutional layer but the exact number depends on the task. Applied to an image 30 kernels would produce a $k \times k \times 30$ array where k is the strided image size.

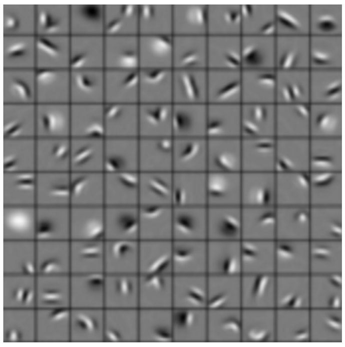


Figure 6: Different kernels for a variety of features than can be contained within the image.

Max Pooling
Max pooling allows to fuse information of input across spatial locations and decreases the number of parameters of the network. Similar to the convolution process a $k \times k$ shaped area moves over the array but instead of convoluting this area with a kernel simply the maximum value in the area is propagated. For the above example (Figure 5a) the propagated value would be 5.7. Typical choices are 2×2 or 3×3 neighborhoods with a striding equal to the neighborhood size.

2.4 VGG16

A very important example for convolutional neural networks is the *VGG16* proposed by K. Simonyan and A. Zisserman from the **V**isual **G**eometry **G**roup at the University of Oxford Simonyan & Zisserman (2014). The model achieves 92.7% test accuracy on Imagenet.

Imagenet is a data-set of over 14million labeled images in roughly 22.000 categories Deng et al. (2009). The images were gathered from the world wide web and labeled by humans. The data-set is used regularly for training neural networks. There are many challenges about which network architectures can achieve the best accuracy on this data-set.

The architecture of the VGG16 is shown in Figure 7. For the original network $224 \times 224 \times 3$ RGB images are used as input (RGB means three color channels). The images are passed through a variety of convolutional layers with 3×3 kernel size which is the smallest kernel possible that still gathers notion of right/left and up/down. Spatial pooling is realized with five max-pooling layers over a 2×2 pixel window with a stride of 2. In the last part of the network three fully connected layers follow the convolutional part. These layers can be imagined as the evaluation and decision making part of the network. While the convolutional layers mainly reduce the original image to its most important features and their spatial position this information is then used in the fully connected layers in order to decide on a certain label. The amount of channels of the last layer depends on the application. For the Imagenet this would be 22.000 channels one for each label. The detailed

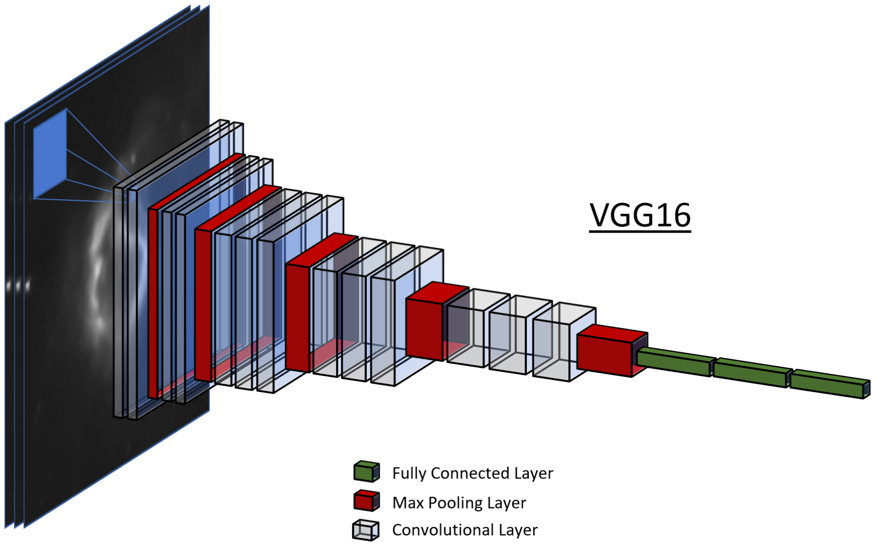


Figure 7: The VGG16 convolutional neural network.

configuration for each layer is listed in Table 8 in the appendix.

A very useful property of the VGG16 is that it can be obtained as an already trained network online Baraldi (2016). The weights and biases in this network are already trained on the images of the Imagenet data-set. As we have already discussed in the previous chapters the first part of the network actually analyses the images on a very basic level (Figure 6). Successively the deeper layers of the network learn more and more abstract properties of the image. So even if the images of a certain task differ from the images in the Imagenet this network can still be very useful. We can initiate the last part of the network, specifically all fully connected layers, new and start training the network again on a new data-set. When a new neural network is created usually all weights and biases are set to a random value. Then the training begins. However, since the convolutional part of the network is already pretrained it can find basic features like edges or curvatures in the image without any problem. The reset fully connected layers than are trained on the new task - in our work to find bubble like structures.

3 Astronomical Objects

3.1 Extended bubble-like structures

The first model of interstellar bubbles that explained UV and X-ray observations of these objects was presented by Weaver et al. (1977). Young massive stars with strong stellar winds, inside of a homogeneous interstellar medium (ISM), create a symmetrical, spherical shock front that propagates outwards. This can be seen in Figure 8 schematically.

The shock front interacts with the interstellar medium, that is surrounding the star. While the fast stellar winds expands freely in the beginning in region (a), they encounter an adiabatic stagnation shock at radius R_1 as the stellar winds accumulate in region (b). In the outer parts the ambient interstellar gas in region (d) is shocked by the expanding bubble at radius R_2, as it accumulates an increasing amount of interstellar gas in region (c). A contact discontinuity at radius R_C separate the shocked stellar wind and the shocked interstellar gas. UV radiation from the star ionizes the surrounding. When the ionized hydrogen recombines again, the relaxation of the excited, recombined hydrogen emits photons. This allows to observe the shock front, and the surrounding medium in the Hα line. Since most gas is accumulated at the shock front the radiation from the front is largest. Dependant on the concentration and evolutionary status of the massive star, different shell structures are produced. These structures have sizes ranging between more than 1000pc to less than 1pc and are differentiated into supergiant shells, superbubbles and bubbles Chu (2008).

Supergiant shells
Supergiant shells (SGS) have sizes of $\approx 10^3$ pc, dynamic ages of $\approx 10^7$ yr, and require multiple generations of star formation.

Superbubbles
Superbubbles have sizes of $\approx 10^2$ pc, dynamic ages of $\approx 10^6$ yr and require only one episode of star formation. They are powered initially by fast stellar winds and later by supernova explosions.

Bubbles
Bubbles have sizes in the order of $\approx 10^1$ pc and are powered by stellar wind of individual massive stars. According to the model, massive stars ionize the surrounding stellar medium which is visible in the Hα line. However, hardly any known main sequence O stars are surrounded by shell nebulae. When the environment of the star has a low density, no strong compression occurs. Therefor no sharp density contrasts to the complex background are produced Chu (2008).

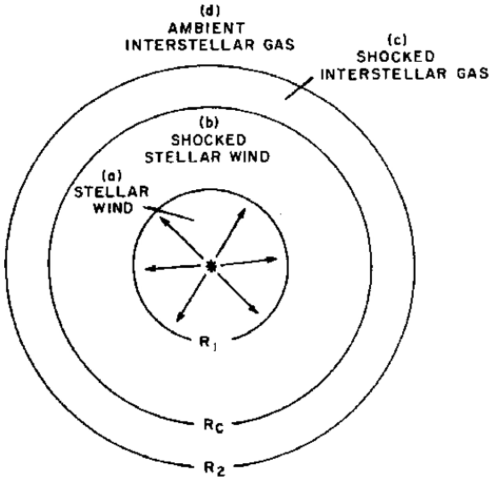


Figure 8: Schematic drawing of the structure of interstellar bubbles. Adopted from Weaver et al. (1977).

The ongoing stellar winds impinging on the outer shell imparts the out-going momentum.

For this work we classify something as a bubble only by its morphological structure, without any further information. This of course is very unspecific, but allows to search a large area of the sky in different wavelengths for interesting candidates that can be evaluated further. Including spectral information into the classification is broached in the outlook (Section 7). The method used in this work is therefore also not dependant on the size of the bubble shell.

3.2 Large Magellanic Cloud

The Large Magellanic Cloud (LMC) is a galaxy, that is only about 50 kiloparsecs away from the Milky Way (Pietrzyński et al. 2019). It is one of the closest galaxies to us. With an inclination of ≈ 33° to 45° (Westerlund 1997) it is tilted in a way, that allows to observe it almost face-on. This makes it a perfect object of investigation for Astronomy. One of these observations is the Magellanic Cloud Emission Survey - MCELS (Smith et al. 2000) - which is a deep imaging survey of the LMC in the emission of Hα, [SII] and [OIII] (Section 4.3). The LMC is an actively star-forming galaxy (Harris & Zaritsky 2009) with a huge population of young massive stars. 1750 of these stars are e.g. listed in a catalog by Bonanos et al. (2009). Since young massive stars are assumed to be the origin of bubble-like structures, we will compare their distributions as part of this work. In Figure 9 the Large Magellanic Cloud is depicted in the optical wavelength regime.

Figure 9: The Large Magellanic Cloud. Zdeněk Bardon/ESO (2017)

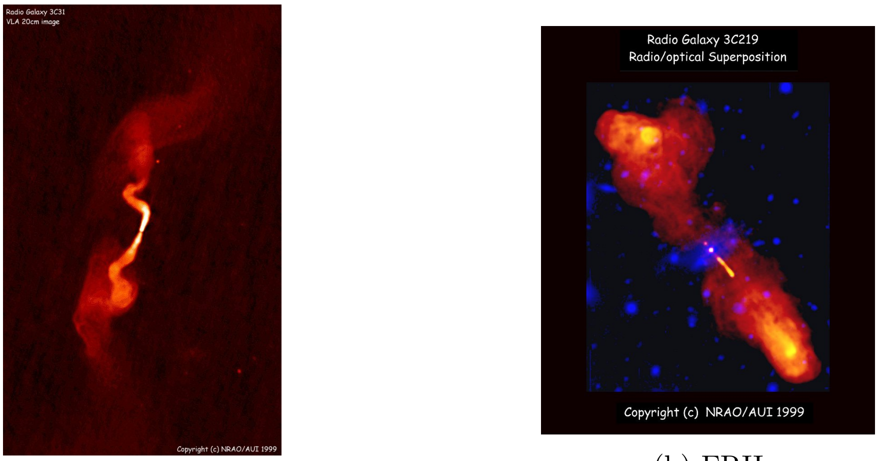
3.3 Active Galactic Nucleus

An Active Galactic Nucleus (AGN) is the center of a galaxy, that is emitting a huge amount of radiation in a broad band of wavelengths. The source of this energy is assumed to be a black hole, with a mass of over 100 million sun masses, that accretes gas and dust. The binding energy from the accreted material is set free, and partly radiated while it is falling inwards to the black hole. During the accretion, two jets of accelerated charged particles are emitted from the galactic center in opposing directions. These jets can reach length of more than a million light-years. AGNs can be differentiated by the level of activity, which is mainly given by the accretion rate and the mass of the source. Dependant on the shape of the radio emission around the jets, AGNs can be classified into FRI or FRII objects.

3.3.1 FRI and FRII

FRI and FRII galaxies are categories of galaxies that are very luminous at radio wavelength. These radio loud galaxies show a wide range of structures. The most common structure are so called lobes. Radio lobes are conically outflows on either side of the active nucleus of the galaxy that are often fairly symmetrical. They are formed around the jets that are emitted by the galactic nucleus. Dependant on the shape of the lobe, the radio galaxies can be differentiated with e.g. the Fanaroff–Riley classification, which was created by Fanaroff & Riley (1974).

With this classification, radio galaxies with active nuclei can be distinguished based on their radio luminosity in relation to their immediate surrounding. The luminosity of FRI sources decreases, as the distance from the central source increases. The


Radio Galaxy 3C31
VLA 20cm image
Copyright (c) NRAO/AUI 1999
Radio Galaxy 3C219
Radio/optical Superposition
Copyright (c) NRAO/AUI 1999

(a) FRI

(b) FRII

Figure 10: Exemplary image of a FRI and FRII radio galaxy. In Figure 10a the FRI radio galaxy 3C31 is depicted (Laing et al. 2008). The lobes are luminous close to the source and fade out towards the outer part. In Figure 10b the FRII radio galaxy 3C219 is depicted (Clarke et al. 1991). The lobes are faint close to the source and increase in luminosity towards the outer part.

luminosity of FRII sources however exhibit an increasing luminosity in some distance of the central source. Both types are depicted in Figure 10. It can be seen that the lobes, that are emitted from the FRI source, are fading outwards.

4 Astronomical Observations

4.1 SHASSA

The Southern H-Alpha Sky Survey Atlas (SHASSA) is the result of a digital imaging survey of Hα emissions from interstellar gas of the Milky Way. The observation was performed for a declination of $\delta = +15°$ to $-90°$ and each image of the observation covers 13° square at an angular resolution of approximately $0.8'$ and reaches a sensitivity level corresponding to an emission measure of 4 cm^{-6}pc (Gaustad et al. 2001).

4.2 SPITZER

SPITZER is a NASA space telescope that is orbiting the sun while tailing the earth. It was launched in 2003 and disabled in January 2020. It was observing the universe in the wavelength regime of 3 to 180 μm. The telescope can perform imaging, photometry, spectroscopy and spectrophotometry.

GLIMPSE
The Galactic Legacy Infrared Midplane Extraordinaire (GLIMPSE) of Spitzer is a survey of the inner Milky Way Galaxy. It spans 130 degrees in longitude and 2-4 degrees in latitude and therefore contains a large volume of our galaxy. The survey was performed using the Spitzer Space Telescope. The observation was performed in four different infrared wavelengths: 3.6, 4.5, 5.8 and 8 μm which we will call I1, I2, I3 and I4. Since most bubble like structures are only visible in I3 and I4 for the most part we only used these images.

4.3 MCELS2

The main task of the Magellanic Cloud Emission Line Survey (MCELS) was the tracing of ionized gas in the Magellanic Cloud Smith et al. (2000). Therefore three different emission lines were measured with narrow band filters: The [SII]$\lambda 6716\mathring{A}$, Hα and [OIII]$\lambda 5007\mathring{A}$ lines. The survey was performed with the 0.6m CTIO Curtis/Schmidt Telescope. It produces individual images of 1.35° × 1.35° with a resolution of 2.3"/pixel.

4.4 ASKAP

The Australian Square Kilometre Array Pathfinder (ASKAP) is a synthesis radio telescope array that consists of 36 dish antennas with 12 m diameter each. The antenna positions are separated by 6km. ASKAP has an excellent imaging capability and dense UV sampling and due to the relatively small dishes a large field of view.

ASKAP is sensitive to radio waves with frequencies in the range of 700 to 1800 MHz. ASKAP became fully operational in February 2019 and is currently conducting pilot surveys.

4.5 EMU

The Emu in the Sky is a ”constellation” of nebulae in the sky that is visible by eye. It has a long history in aboriginal culture and is extensively engraved in rocks all over the Ku-ring-gai Chase National Park in the north of Sydney. The Emu has became an icon of the Australian SKA Pathfinder (ASKAP) project and was one of the first pilot observation targets. Within this constellation a huge amount of radio galaxies was discovered. The observation was performed for a period of 10h per pointing and 100h total with an observing band of 800-1088 MHz at 944 MHz centre. The resolution of the observation is 13 × 11 arcsec. It was performed in 8 different observation tiles so far. The tiles and their coordinates are listed in Table 2 adopted from https://confluence.csiro.au/display/askapsst/EMU.

Tile description	Ra	Dec	Observation time	Centerfrequency
EMU_2034-60	20:34:17.142	-60:19:18.17	10 hrs	943.491 MHz
EMU_2042-55	20:42:00.000	-55:43:29.41	10 hrs	943.491 MHz
EMU_2115-60	21:15:25.714	-60:19:18.17	10 hrs	943.491 MHz
EMU_2132-51	21:32:43.636	-51:07:6.396	10 hrs	943.491 MHz
EMU_2027-51	20:27:16.363	-51:07:6.396	10 hrs	943.491 MHz
EMU_2118-55	21:18:00.000	-55:43:29.41	10 hrs	943.491 MHz
EMU_2154-55	21:54:00.000	-55:43:29.41	10 hrs	943.491 MHz
EMU_2156-60	21:56:34.285	-60:19:18.17	10 hrs	943.491 MHz

Table 2: Performed tiles of the ASKAP observation in the EMU region.

4.6 Simbad

Simbad is the reference database for identification and bibliography of astronomical objects. It contains identifications, ‘basic data’, bibliography, and selected observational measurements for several million astronomical objects. Simbad is developed and maintained by CDS, Strasbourg. Building the database contents is achieved with the help of several contributing institutes (Wenger et al. 2000). The Simbad database has a python application programming interface which allows to access data automatized. The web presence of the database can be found here http://simbad.u-strasbg.fr/simbad/.

5 Extended bubble-like structure detection

The program described in this Section can be downloaded from https://www.sternwarte.uni-erlangen.de/gitlab/ramsteck/blobscan. It is referred to as *Blobscan* in the following.

In order to implement a neural network that is able to detect bubble-like structures in astronomical survey data a framework had to be implemented for a variety of associated tasks. Using the framework we were able to obtain a program that is able to find bubble-like structures automatized. Besides the parameter that were used for training the network there are some important parameters that vary the outcome drastically. They are described in Section 5.1.1.

For further instructions regarding the application of the program see Section A.2.4 in the appendix. In the following we talk about positive training data when there is a bubble in the image and negative if there is no bubble.

5.1 Framework

The framework that we used throughout this work was necessary to execute a variety of tasks that were associated with training, validation and application of the neural network. It mainly contains the following methods

Training Data Extraction
Methods to extract training data from labeled astronomical survey areas and to generate counter examples. Therefore a list of galactic coordinates with a certain radius is given to the algorithm and the framework allows to extract these areas from within astronomical data by cutting them out and saving them as an separate image.

Neural Network Handling
Methods for training and handling the neural network. Therefore the architecture of the network described in Section 5.2 is build up and the extracted training data is given to the network. It also allows to save and load the network and the used parameters as well as the training history of the model.

5.1.1 Bubble Detection

In order to apply the network to a large sky area we need a method to search this sky area for bubbles. In state of the art object detection algorithms the larger image is usually divided into sub images by separating regions in the image due to their color and contrast. This allows to create sub images around areas that usually belong to the same object. For astronomical images this is very hard to recreate since the contrast between objects and background vary heavily and we did not implement different colors e.g. wavelengths into the network.

So we need a different approach. In this work we applied the network to a large sky region by grating the image of this region into several tiles. Each tile is a small cutout region of the wider sky area. We then predicted every tile of the image with the network. Essentially we defined a certain box size e.g. $0.01 \times 0.01\text{degree}^2$ and a certain step size e.g. 1/2 of the box size. Then we dissect the complete image into tiles by walking over the image line by line while the step width along the line and between lines is the step size. E.g. for a 10×10 image with 2×2 box size and 2×2 step size this would result in 25 boxes - five for each line with five lines. Every box is then used as an input for our convolutional neural network. The network maps the input to two different categories - bubble or no-bubble. This method is by far more inefficient but since a real-time prediction is not necessary and every sky survey in general only needs to be evaluated once, this is okay. Also it guarantees to include every possible area in the image.

The method has two downsides though. For once we need to know the size of the bubbles beforehand in order to choose a suitable box size. It is possible to use different box sizes and superimpose the results for all box sizes. And by choosing a too large step size we can face the problem that the network does not predict reliably. Although we tried to train the network to be spatially invariant this can not be ensured completely. Objects that are cut off by the border of the box can not be predicted reliably. A large step size however will result in a higher probability to only predict cropped objects instead of complete objects. A small step size on the other hand, results in a way higher computational time.

The box size and the step size are parameters that can be set when using the *Blobscan* program.

For this work we used a complete set of box sizes during the training: 0.033, 0.050, 0.067, 0.083, 0.100, 0.133, 0.150, 0.167, 0.200, 0.233 and 0.267 degree. The step size was chosen to be 1/7 of the box size to make sure that every object in the image is given to the network in a way that the network is able to identify it as a bubble.

5.1.2 Box Merging

The framework also needs to contain methods for merging predicted boxes with different sizes (Section 5.1.1) or with the same size that appear multiple times due to a small step size. Therefore we have to differentiate between two cases: If there are two boxes close to each other because there is a bubble in the overlap of both boxes or if there are two boxes close to each other because in each of the non-overlapping part of the boxes there are objects. So a new box with twice the box size of the individual boxes is placed in the center of the overlap and this new box is predicted by the network. If the network predicts a bubble in this new box both smaller boxes are merged into the bigger one. If it does not predict a bubble both small boxes are assumed to be independent of each other and kept in the results.

However, if a small box is already contained in a bigger box it is removed.

5.1.3 Neural Network Validation

In order to evaluate our network, the framework also has to contain methods for validating the objects that were predicted by the network. Either by comparing the predicted sky regions with the SIMBAD database or by plotting the object in order to evaluate it manually. This is described more detailed in Section 5.2.6.

5.2 Network

A $VGG16$ based convolutional neural network was trained on a small amount of manually selected bubbles and then applied to new astronomical data in order to generate more training data. From iteration to iteration the detected bubbles got more and more blurred out. While we started with well defined pancake-shaped bubbles with a clear boundary the last generation of the network predicted even very diffuse toroidal shaped bubble-like structures. This led to a rapid increase in false positive predicted samples but also allowed us to find a huge variety of bubble-like structures. In order to evaluate the predicted samples we checked the sky regions that were predicted as bubble for any known objects in SIMBAD. By evaluating the ratio between found objects that were already known and the amount of unknown or unrelated objects we settled with the most promising training status of the network. This final state of the network was then applied to MCELS2 data of the Large Magellanic Cloud.

5.2.1 Structure

The network structure is based on the VGG16 architectureSimonyan & Zisserman (2014). The network is pretrained on Imagenet and can be downloadedBaraldi (2016) and used inside KerasChollet et al. (2015). Keras is a high-level deep learning API for python. Since our initial training-set was extremely small for deep learning standards the use of a pretrained network was advantageous (Section 2.4) and the major motivation to use the VGG16. As already discussed there, retraining an already pretrained network benefits from the basic feature extraction in the first parts of the network that is mostly similar for any kind of image. In direct comparison to an untrained simple convolutional neural network this approach has proven to be better. The used pretrained VGG16 network has an $35 \times 35 \times 3$ input and accordingly sized further layers. Since it is pretrained on the RGB images from Imagenet it has intrinsically three channels. The images of bubbles that were used here only have one channel - the intensity. This means that channel correlation inside the network can not be exploited but it does not restrict the power of the network in general. In order to fit to the input of the network the singular channel of the bubble images was expanded to three channels by copying the values of each

Model:	"Blobscan"	Input:	"35 × 35 × 3"
Layer	(type)	Output Shape	Param #
vgg16	(Model)	(None, 1, 1, 512)	14714688
flatten	(Flatten)	(None, 512)	0
fc1	(Dense)	(None, 1024)	525312
dropout	(Dropout)	(None, 1024)	0
prediction	(Dense)	(None, 2)	2050
Total params: 15,242,050			
Trainable params: 7,606,786			
Non-trainable params: 7,635,264			

Table 3: Architecture of the Blobscan model. The amount of weights and biases within the layers is called "Params" here.

pixel.

The fully connected layers of the network were reset and trained only on our training-set. The final structure is listed in Table 3. The layer vgg16 here represents the untouched layers from the original VGG16 architecture as it is listed in Table 8 but without the layers flatten, fc1, fc2 and prediction. Instead the newly generated layers flatten, fc1, dropout and prediction are attached. These are the only trainable layers. The prediction layer has two outputs - bubble and no-bubble. The layer "flatten" only takes a multi-dimensional array and concatenates it line by line to a one-dimensional chain. A 25 × 25 image for example would become a chain of 25 lines with 25 values each - A chain with 625 scalar values. For a detailed explanation of the fully connected layers see Section 2.1.2. The dropout layer is explained in Section 2.1.7. The used activation function for classification is the softmax function and thus the loss function is the categorical cross-entropy. This is explained in detail in Section 2.1.4 and 2.1.5.

5.2.2 Training Iteration One

Positive Training Data
The first generation of training data was gathered manually. Therefore the astronomical data of the SHASSA observation (Section 4.1) were partially searched for bubble-like structures and their positions and radii were marked. Due to this time-intensive procedure only a total of 83 bubble-like structures were marked. The complete list of this first generation training-data is listed in Table 9 in the appendix Section A.2.1. The marked regions were cutout and used as positive training set.

Negative Training Data
In order to obtain images that serve as counter example random positions within the same data that do not overlap with the marked regions were generated. These

Hyper-parameter	Value
Validation fraction	0.2
Epochs	30
Batch size	10
Class Weights	1:12

Table 4: Blobscan training parameters.

counter examples were cutout as well and used as negative training set. This works almost automatically and in general we could generate an almost arbitrary amount of counter examples. However, if the amount of negative training data is by far bigger than the amount of positive training data this leads to the following problem: In Section 2.1.6 we already discussed how training works. For every training step a batch of several images is presented to the network and the weights and biases are adapted. If all of these images happen to be actually negative samples the network will learn to simply categorize every input as negative independent of the input. By applying a certain weight to the loss of the underrepresented class this problem can be reduced to some point. If the weight difference is too high this does not work reliable anymore. Imagine the network learns many little steps into classifying everything as negative and then for one sample takes an enormous step into another direction. The probability that this result in an overshoot of the network is very high. Therefore we limited the amount of negative training sample to a maximum of 10 times the amount of positive samples. This lead to a negative training set of 830 images.

Additionally since we want the network to be more sensitive towards positive predictions we increased the class weight for the loss of positive training samples further to a total of 12. Miss-classifying a bubble as no-bubble therefore results on average in a 2 times bigger loss than miss-classifying a no-bubble as bubble. The additional weight difference of 10 only compensates the different amount of samples here. The network was trained in a first iteration with the following hyper-parameters:

As optimizer the RMSprop with a learning rate of $\eta = 1 \cdot 10^{-4}$ is used.

Since our training data is pretty small we used data augmentation (Section 2.1.8) with the parameters listed in Table 5.

In order to prevent overfitting the validation loss was monitored and after four epochs without decreasing the validation loss the training was stopped. The weights were then set to the values for the minimal validation loss. An exemplary set of positive training data is depicted in Figure 11. We only used smooth bubble like structures with a well defined border that are filled towards the middle. The network after this training iteration is called *model_1* in the following.

Augmentation-parameter	Value	Comment
Rotation Range	360°	Allowed rotation of the object
Width Shift Range	0.01	Allowed scaling for the width of the object
Height Shift Range	0.01	Allowed scaling for the height of the object
Brightness Range	0.2 to 1.9	Allowed scaling of the intensity of the object
Zoom Range	0.5 to 1	Allowed size change of the object
Horizontal Flip	True	If the object is allowed to be mirrored horizontally
Vertical Flip	True	If the object is allowed to be mirrored vertically

Table 5: Blobscan data augmentation parameters.

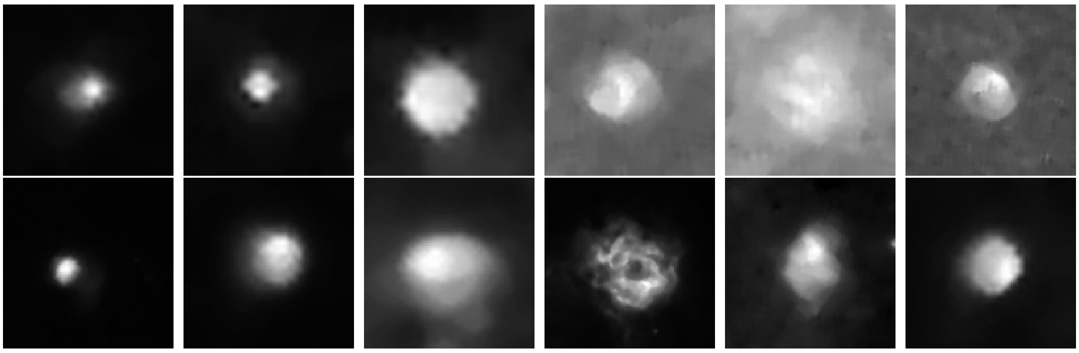


Figure 11: 12 exemplary positive training samples of the first generation of training data. The data is gathered from SHASSA. It can be seen that the structures are mainly definite with clear borders and filled towards the middle.

5.2.3 Training Iteration Two

In a second generation of the network new data were used. Therefore the network that was already trained on the first generation training data was applied to SPITZER (Section 4.2) data using the bubble detection method developed within the framework (Section 5.1.1). The resulting positive classified images were then evaluated for correctness manually. While only definite pancake-like shaped bubbles were used in the first training iteration the network still has a certain tolerance towards blurred out objects that are not always filled towards the middle. Only definite bubble-like structures were approved. This time also toroidal shaped bubble-like structures were approved. An exemplary set of accepted objects is depicted in Figure 12.

All true positive samples were additionally used as positive training set for another training iteration of the network. In order to keep the ratio between positive and negative training samples additional counter-examples were extracted randomly from

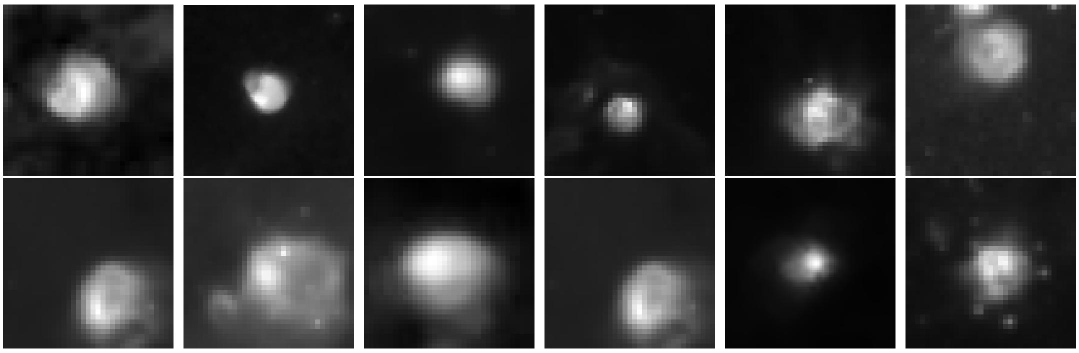


Figure 12: 12 exemplary positive training samples of the second generation of training data. The data were taken with SPITZER. It can be seen that the structures are mainly similar to generation one but the borders are not so well defined anymore and are blurred out. Using this kind of samples as additional training set will result in an even wider range of accepted shapes.

the leftover SPITZER data. The amount of training data increased to 731 positive samples and 6610 negative samples. The network that was already trained on the generation one training set was then trained on the generation two training set again. The hyper-parameters, the optimizer and the early-stop method used in the first generation were maintained. The network after this training iteration is called *model_2* in the following.

5.2.4 Training Iteration Three

In the third generation of the network the SPITZER data were again searched by the bubble detection method. This time the network already was trained on the first and second generation of training data. The resulting positive classified images were evaluated manually for correctness. Again only definite bubble-like structures were approved but toroidal as well as pancake-like shaped bubbles were accepted. An exemplary set of accepted objects is depicted in Figure 13. Due to the increased variety of allowed shapes in the second training iteration the network predicts an even bigger variety this time. Interestingly the network which was actually only trained on pancake shaped bubbles in the first place still classified an increasing number of toroidal shaped bubble-like structures as bubble. This was not expected but is in general not odd since it only means that the network weights the fact that the object is circularly shaped way more than the fact that it is filled towards the middle. This was amplified by the fact that a lot of the results from the previous generation that tend towards a toroidal shaped were selected as further training samples The amount of training data increased to 1362 positive samples and 6610 negative samples. The network that was already trained on the generation one and generation two training sets was then trained on the generation three training set again. The hyper-parameters, the optimizer and the early-stop method used in the first generation were maintained. The network after this training iteration is called

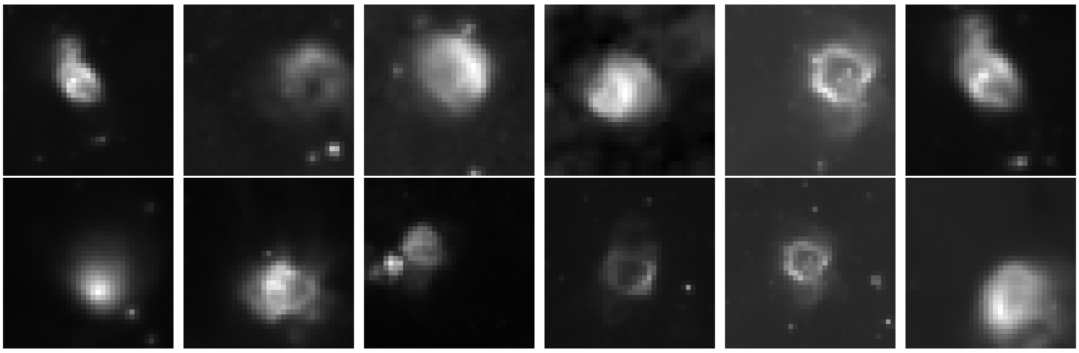


Figure 13: 12 exemplary positive training samples of the third generation of training data. The data is gathered from SPITZER. The borders of the objects became even more smeared out. Interestingly the network classified an increasing number of toroidal shaped bubble-like structures as bubble.

model_3 in the following.

5.2.5 Training Iteration Four

In the fourth generation of the network the SPITZER data were again searched by the bubble detection method. This time the network already was trained on the first, second and third generation of training data. The resulting positive classified images were evaluated manually for correctness. Again only definite bubble-like structures were approved but toroidal as well as pancake-like shaped bubbles were accepted. An exemplary set of accepted objects is depicted in Figure 14. It can be seen that the resulting samples become more and more blurred out and the borders are not well defined anymore. Even filament like structures were classified as bubbles. By pushing the decision criteria for bubbles into a more and more lenient direction the amount of results increased dramatically but also the amount of false positive classifications increased. We were able to alter these decision criteria by softening the requirement that objects had to fulfill to become a positive training sample.

The amount of training data increased to 2366 positive samples and 35663 negative samples. The network that was already trained on the generation one, two and three training sets was then trained on the generation four training set again. The hyperparameters, the optimizer and the early-stop method used in the first generation were maintained. The network after this training iteration is called *model_4* in the following.

5.2.6 Validation

Usually the performance of a network can be evaluated by withholding a certain fraction of the training data that is then used as validation set (Section 2.2). This part of the training data is not used in the actual training of the network. After training finished the network is asked to predict all validation samples and the ac-

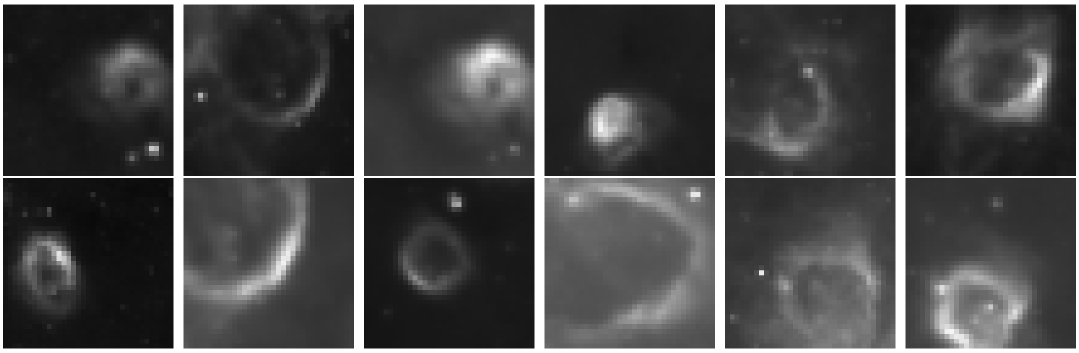


Figure 14: 12 exemplary positive training samples of the fourth generation of training data. The data is gathered from SPITZER. It can be seen that the structures are smeared out and the borders are increasingly blurred. The network started to classify even filament like structures as bubble.

curacy on the validation set is assumed to be the actual accuracy of the network. This assumption is based on the idea that the validation set is representative for all the data used. However, since our data set is extremely small for deep learning conditions this does not work very well. Additionally the shapes of bubble-like structures are diverse and our training sets do not include enough of this variety. So we needed a different validation possibility to evaluate the performance of our network. Therefore we used the astronomical database SIMBAD (Section 4.6). Every predicted sky region of the network was checked in SIMBAD for known objects. By evaluating the amount of regions that contain an object type that we associate with a bubble or bubble-like structure we are able to evaluate the performance of the network. The following categories in the SIMBAD catalog were assumed to be associated to bubble or bubble-like structure:

- Bubble
- Dense Core
- HII Regions

These categories are not the only kind of objects that are physically associated with bubbles or bubble-like structures but they were the most useful ones for validation. For each generation of the network the results differed. With every generation the total amount of results increased rapidly. This is not remarkable since for every increase of the training data the network has seen a bigger variety of objects that are classified as bubbles. Therefore the network learns more and more features that are associated to the bubble class and the classification boundaries become more and more lenient. However, this goes with the problem that also more and more objects were classified as bubbles which could not be approved as positive result manually or by SIMBAD anymore. In Figure 15 the progress of the network on the SPITZER data is depicted. While the amount of found objects that we associate with bubbles flattens the overall amount of found regions increases heavily. The

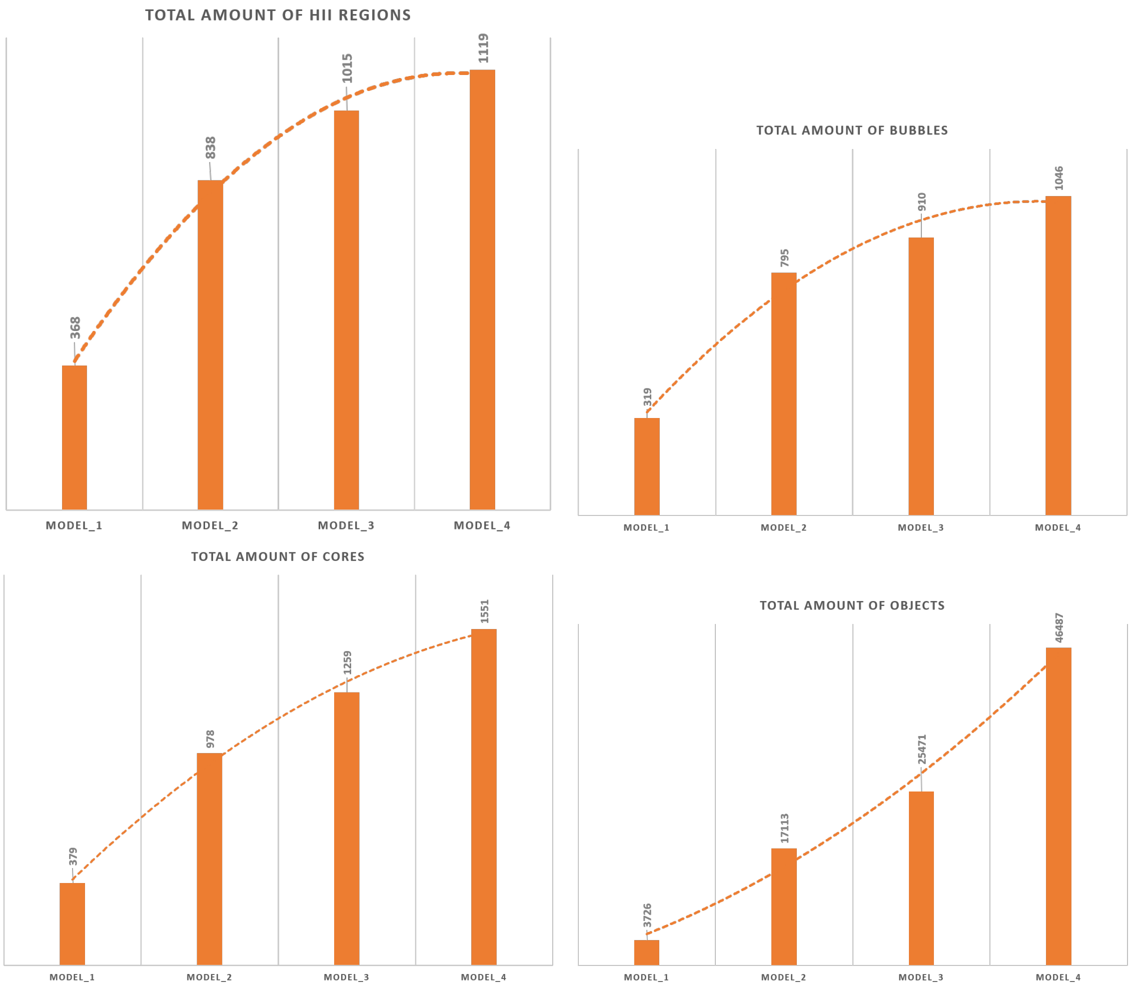


Figure 15: Progress of the evaluated results throughout the different iterations of the neural network.

detailed constituents of the result for each model is shown in Figure 16. We decided to continue the work with $model_2$. The following section will show that the results are already plenty and even this state of the network already tends to overpredict.

5.3 Result

5.3.1 Bubbles in the LMC

The final model of our *Blobscan* network was used on the MCELS2 data (Section 4.3). The used set of box sizes was 0.056, 0.083, 0.111, 0.139, 0.167, 0.222, 0.250, 0.278 0.333, 0.389 and 0.444 degree and the tiling was 1/7 of the box size. The results for the Hα image are depicted in Figure 17 and all found bubbles are listed with their position and radius in Table 11. The results for the [OIII] image are depicted in Figure 18 and all found bubbles are listed with their position and radius

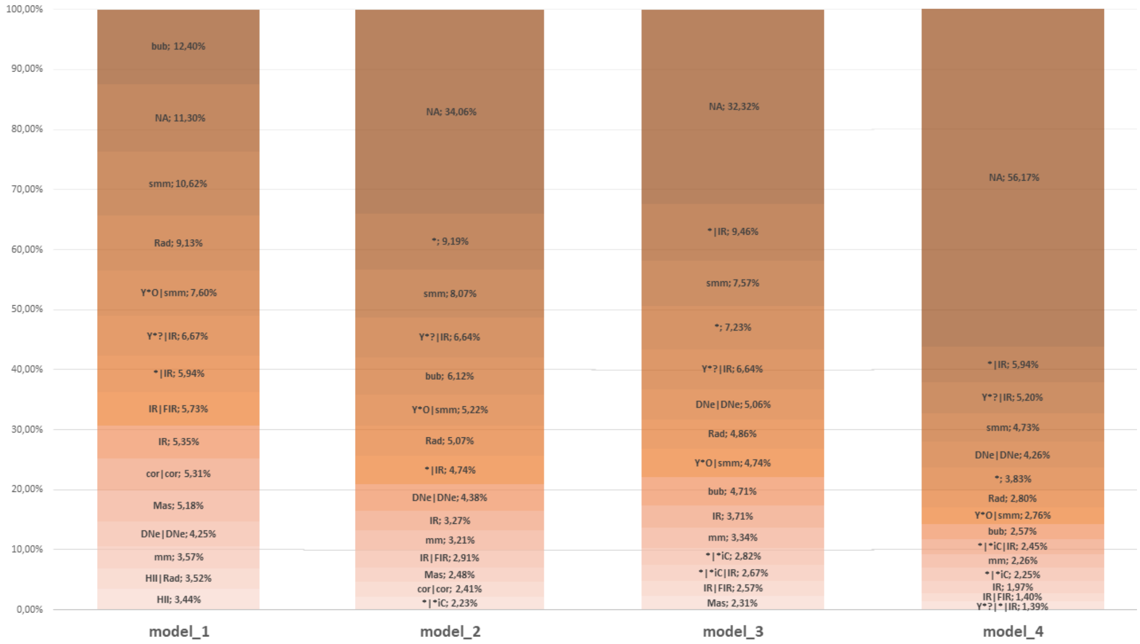


Figure 16: SIMBAD categories for the found sky regions in SPITZER data for all models. NA means that there is no known object in the sky region that was classified as bubble by the network. The complete list of abbreviations can be found in Table 10.

in Table 12. The results for the [SII] image are depicted in Figure 19 and all found bubbles are listed with their position and radius in Table 13. Even though the network was trained and used on data from various different wavelengths the network generalizes well as long as bubbles or bubble-like structures look similar across the used wavelengths.

5.3.2 Bubbles compared to star distribution

We already discussed in section 3.1 that it is assumed that stellar bubbles and superbubbles originate from massive stars. We can compare the position of the bubbles that were found by the network with the catalog of massive stars in the LMC Bonanos et al. (2009). Furthermore we can compare the distribution of found bubbles for each spectral line that we used. In order to do this we used the spatial analysis method called the bivariate *Ripley's K function* Ripley (1976). With this function one can determine if a spatial distribution is dispersed, clustered or randomly distributed. The function essentially counts the amount of objects within a certain radius of a given object. This is done for each object. A value can then be determined that correlates to the clustering of the objects. For different kinds of objects this calculation has to be done separately. Then the amount of objects of one class within a certain radius of the object of the other class is counted. Mathematically

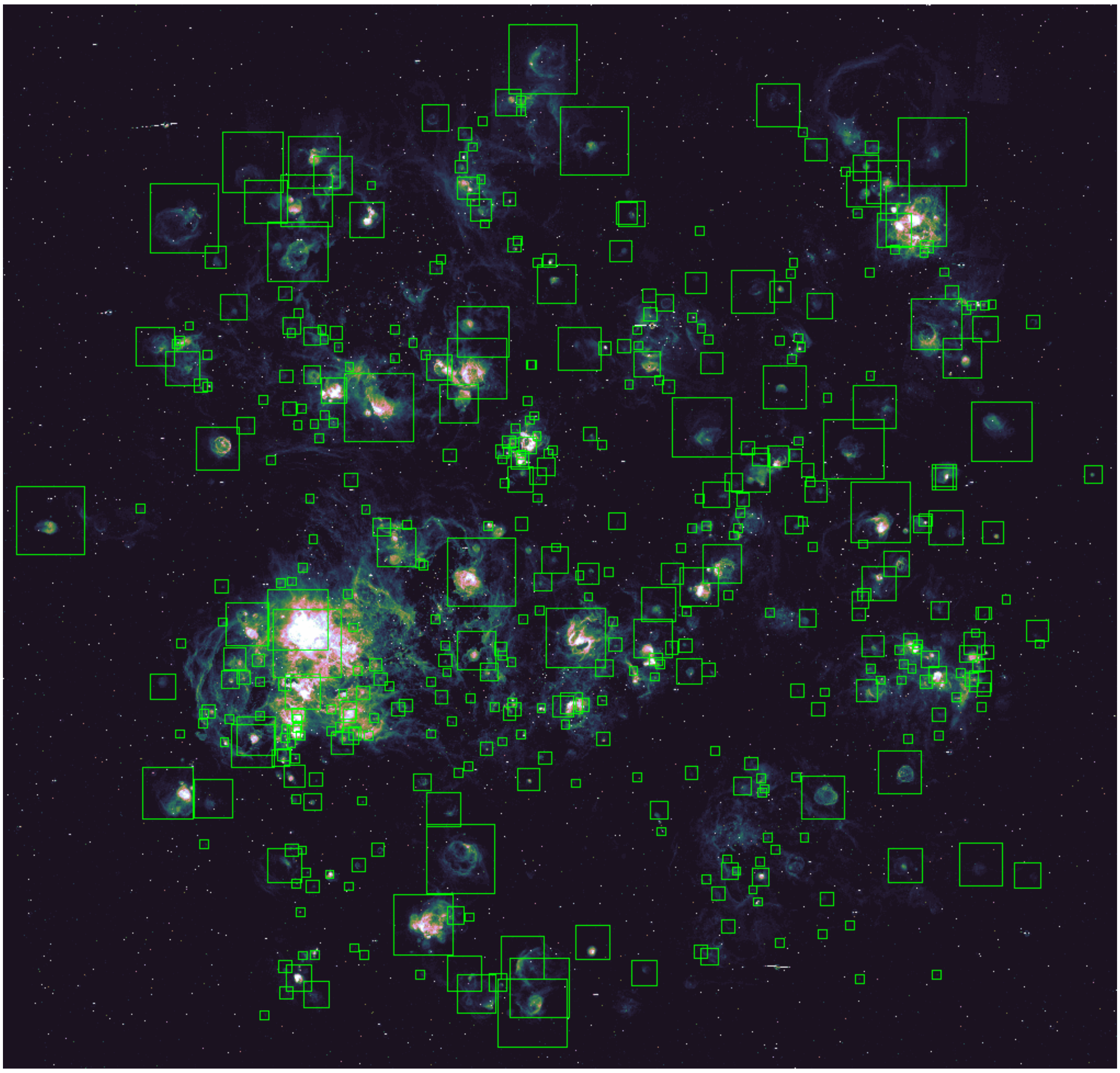


Figure 17: The LMC in the Hα line and the marked bubbles that were predicted by the network.

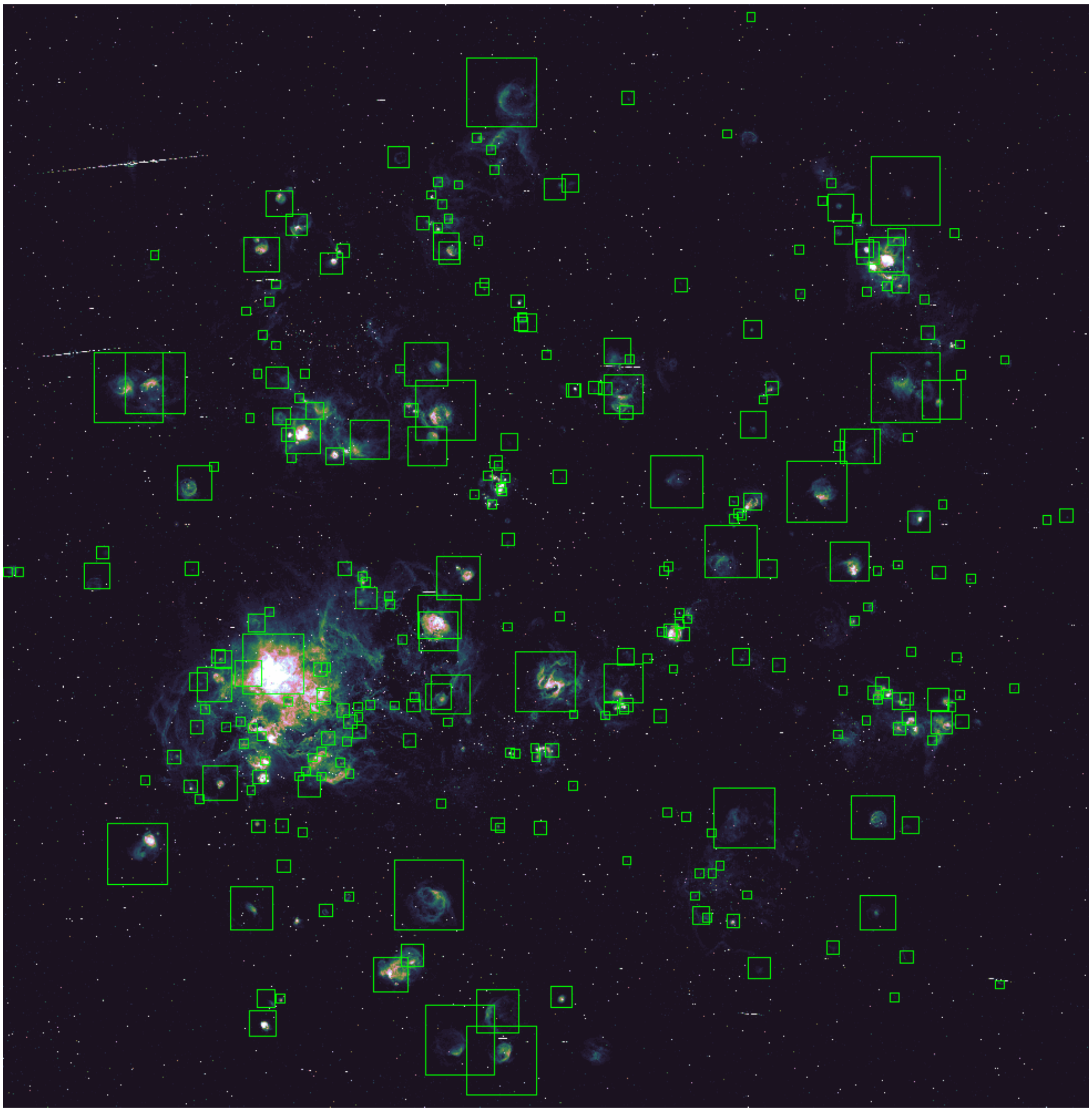


Figure 18: The LMC in the [OIII] line and the marked bubbles that were predicted by the network.

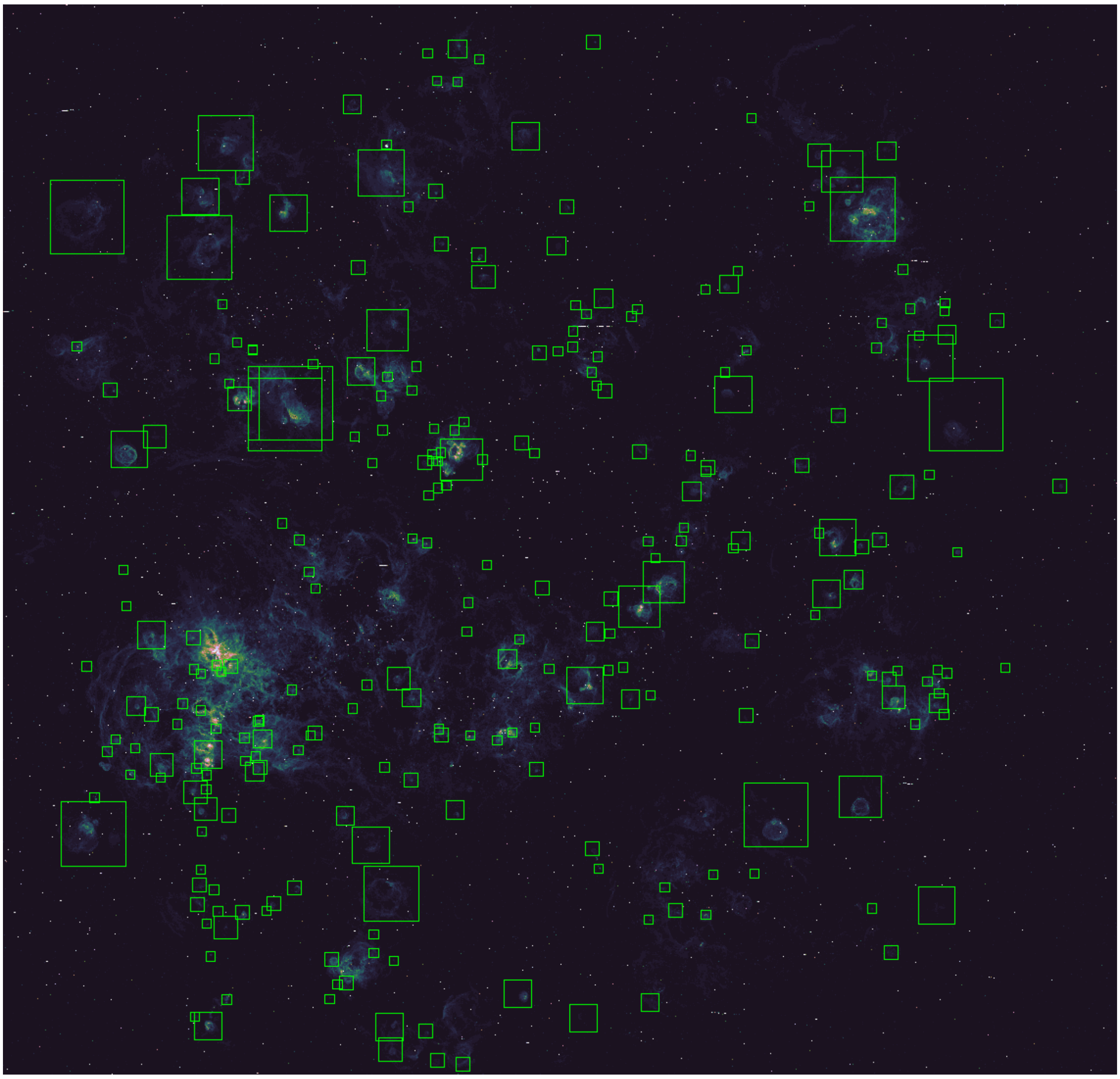


Figure 19: The LMC in the [SII] line and the marked bubbles that were predicted by the network.

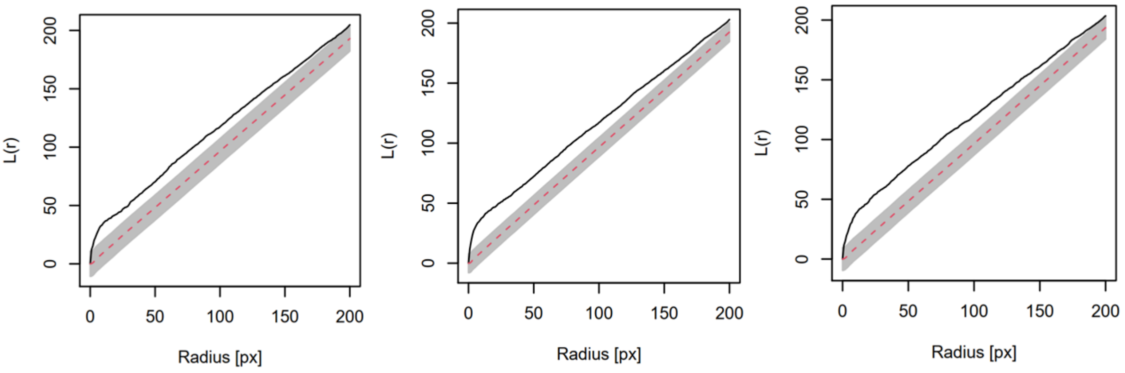


Figure 20: Bivariate Ripley's L comparing the distribution of found bubbles between Hα and [OIII] (left), Hα and [SII] (middle) and [OIII] and [SII] (right). 100 pixels correspond to 0.056 degree.

Ripley's K function is given by

$$K(t) = A \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \omega_{ij} \frac{\Theta_t(||r_i| - |r_j|| - t)}{n_1 \cdot n_2} \tag{16}$$

where A is the area of the plot, ω_{ij} the edge correction and n_1, n_2 the sample size. Θ is the Heaviside function (Figure 3a) and $||r_i| - |r_j||$ is the distance between object i and object j. The Heaviside function is one if both objects are within a radius t and zero otherwise. The edge correction is necessary since the property of clustering is dependent on the area that is considered. If e.g. four points are randomly distributed within a square this distribution would not be considered clustered. However, if we enlarge the square but keep the points position they would be clustered within a way bigger square.

For large t the estimators of K often have a high variance due to its cumulative nature. In order to mitigate a variance stabilized transform can be used

$$L(t) = \sqrt{\frac{K(t)}{\pi}} \tag{17}$$

which was first proposed by Besag Besag (1977). Usually in order to test a pattern with Ripleys K it has to be compared to another known distribution. For example it can be tested against a homogeneous Poisson distribution. If the tested pattern is clustered the value of L is bigger than for the poisson distribution. However, if we want to use the bivariate function and compare two known clustered distributions it makes no sense to compare it to a homogeneous distributed pattern. Therefore one of the clustered patterns is shifted randomly 300 times and the mean L value is calculated. The used algorithm was adapted from the thesis of Caroline Collischon Collischon (2020). The values in the following evaluation are chosen similarly so the results of both works can be compared. In Figure 20 the spatial correlation of predicted bubbles in the different wavelengths is tested. It can be seen that

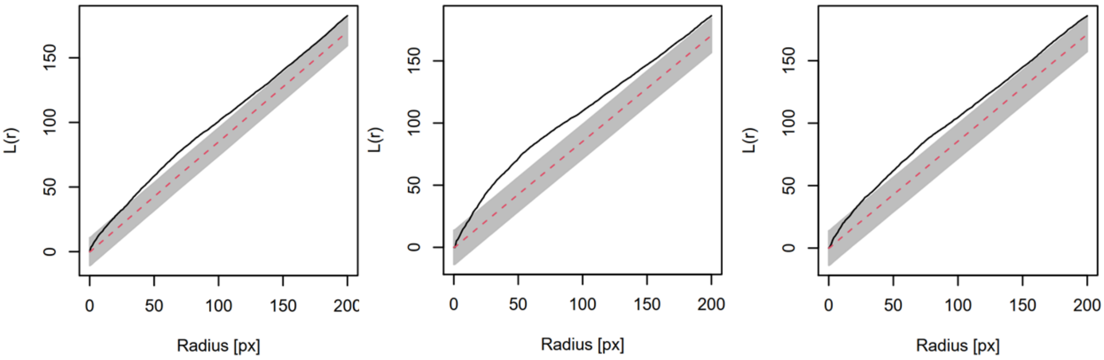


Figure 21: Bivariate Ripley's L comparing the distribution of massive stars to found bubbles in Hα (left), [OIII] (middle) and [SII] (right). 100 pixels correspond to 0.056 degree.

the bubble distribution in all three wavelengths are correlated. This can also be seen by eye if you compare the Figures 17, 18 and 19. It actually states that the majority of the bubbles is visible for the network in all three wavelengths. In Figure 21 the spatial correlation of predicted bubbles in all three wavelengths is compared to the distribution of massive stars given by the Bonanos catalog. The correlation of the patterns is significant while the biggest L values are below a radius of 100 pixels (0.056 degree). This is due to the fact that the majority of bubbles that were predicted by the network are given with a box size of 100 pixel. Also for bigger boxes the correlation states that a major part of the massive stars are clustered within 100 pixel around the center of the box. Nevertheless the correlation between the two distributions is not overwhelming.

Another important variation of Ripleys K function is the pair correlation function

$$g(t) = \frac{K(t)}{2\pi t} \tag{18}$$

The main difference to Ripleys K function is the different weighting. The PCF gives greater weight to points close to the respective object and less weight to points further away. In Figure 22 the pair cross correlation function for predicted bubbles in Hα, [OIII] and [SII] to the massive stars is calculated. It can be seen that the correlation is significant for small radii and reduces for bigger radii. This supports the conclusion we draw from Ripleys K already. The biggest correlation between the distribution of young, heavy stars and predicted bubbles is in the radius of 100 pixels and less. The envelopes that are depicted in the graphs are the Global Maximum Absolute Deviation (MAD). The simulated mean value is taken from $n_{sim}/2$ of the simulations. For the remaining patterns, the highest absolute deviation from this mean is calculated. Then, again the n-th largest of these deviation values is used as a critical value d_{crit}. The envelope then has upper/lower boundary values mean$\pm d_{crit}$ at a constant width $2{\cdot}d_{crit}$. The null hypothesis is rejected if the observed function exceeds this envelope at any value of r. The MAD test has significance level $\alpha = n/(1 + n_{sim}/2) \approx 0.046$ (Collischon 2020) (Ripley 1981).

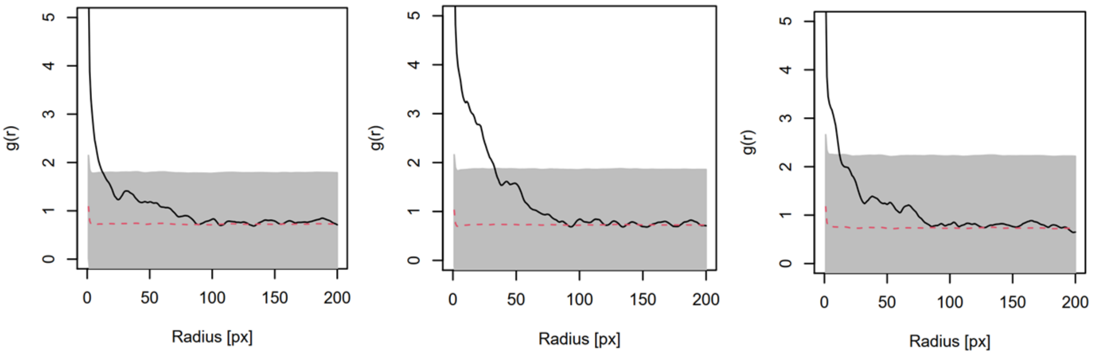
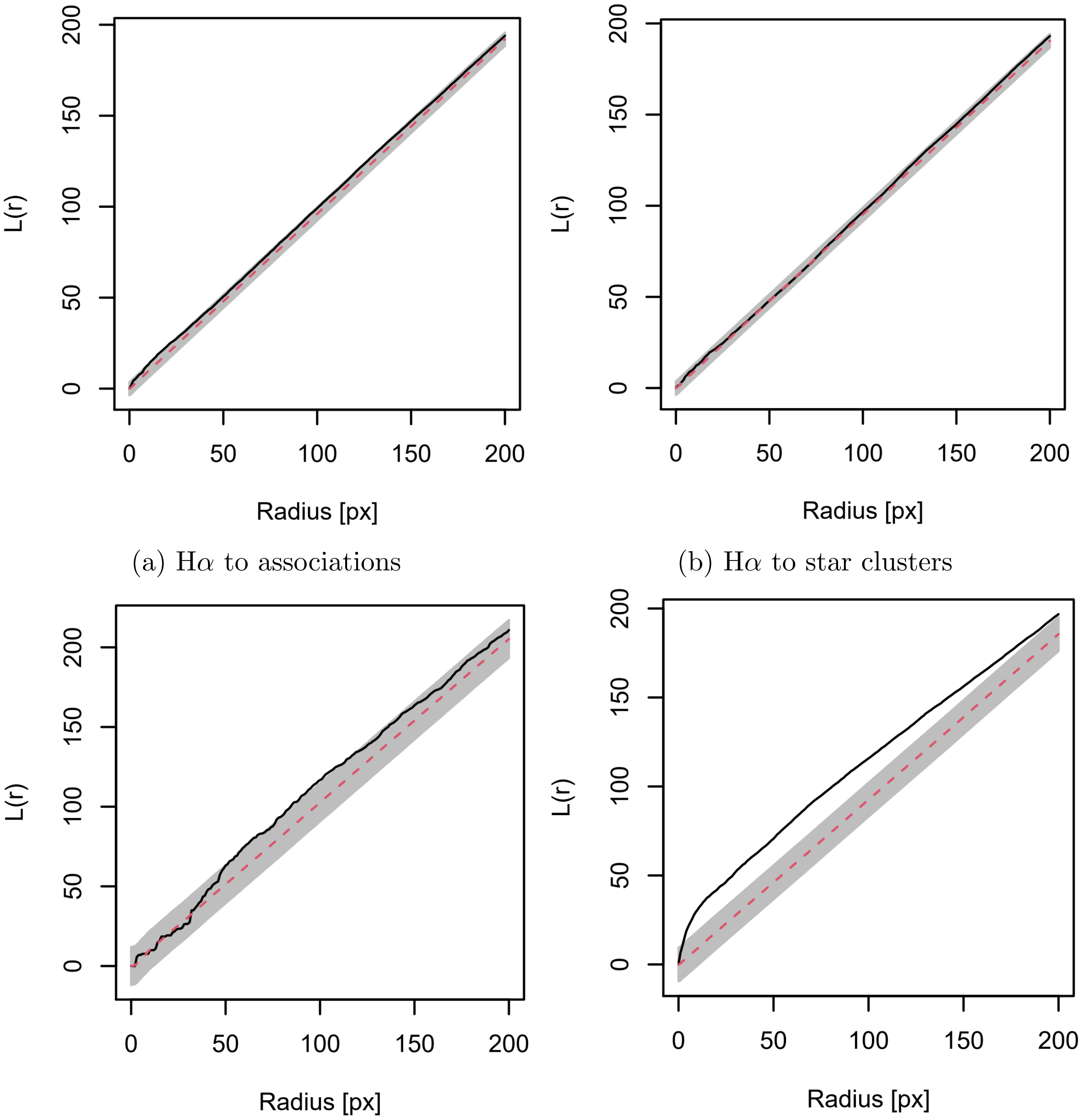


Figure 22: Cross correlation function of the distribution of massive stars and the found bubbles in Hα (left), [OIII] (middle) and [SII] (right). 100 pixels correspond to 0.056 degree.

5.3.3 Bubbles compared to other extended objects

Additionally we used the Ripleys K function, described in the previous section, to investigate the distribution of bubbles that were found by the network further. Therefor we compared the distribution with the general catalog of extended objects in the Magellanic Cloud by Bica et al. (2008). The catalog contains the categories HI shells and supershells, associations, star clusters and emission nebulae. In the Figures 23, 24 and 25 the correlation of the detected bubbles in Hα, SII and OIII with these categories is depicted. It can be seen, that only the distribution of emission nebulae is significantly correlated to the detected bubbles. This was expected since these regions mostly contain young, hot stars - the source of bubble-like structures - and are mainly HII regions. A correlation to HI shells and supershells was possible but since most of these objects were found in the HI hydrogen line they are not necessarily visible in Hα, SII or OIII. No correlation to the other two categories was expected and none was found.



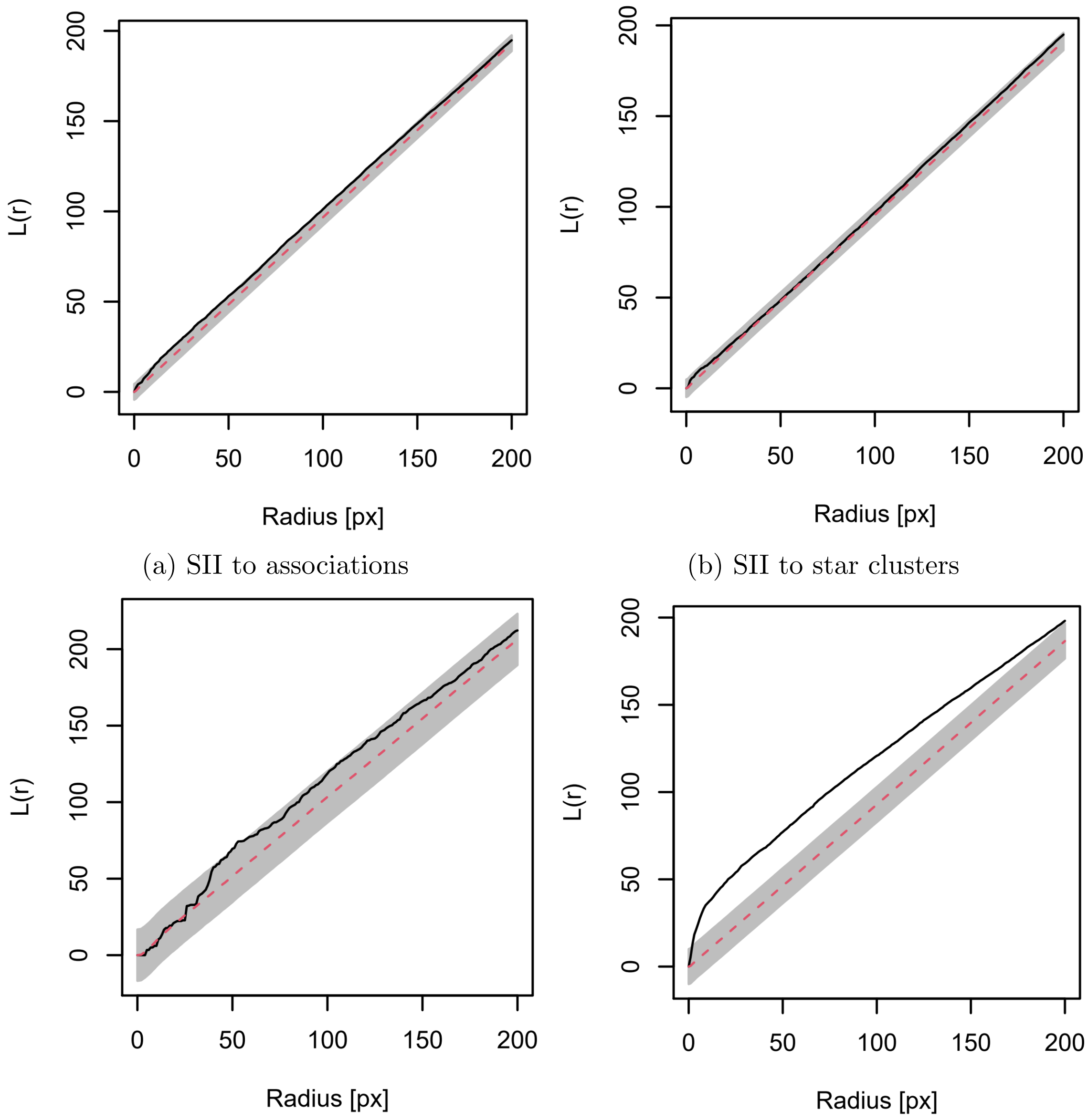
(a) Hα to associations

(b) Hα to star clusters

(c) Hα to HI shells and supershells

(d) Hα to emission nebulae

Figure 23: Correlation between bubbles found by the network in Hα and different object categories.



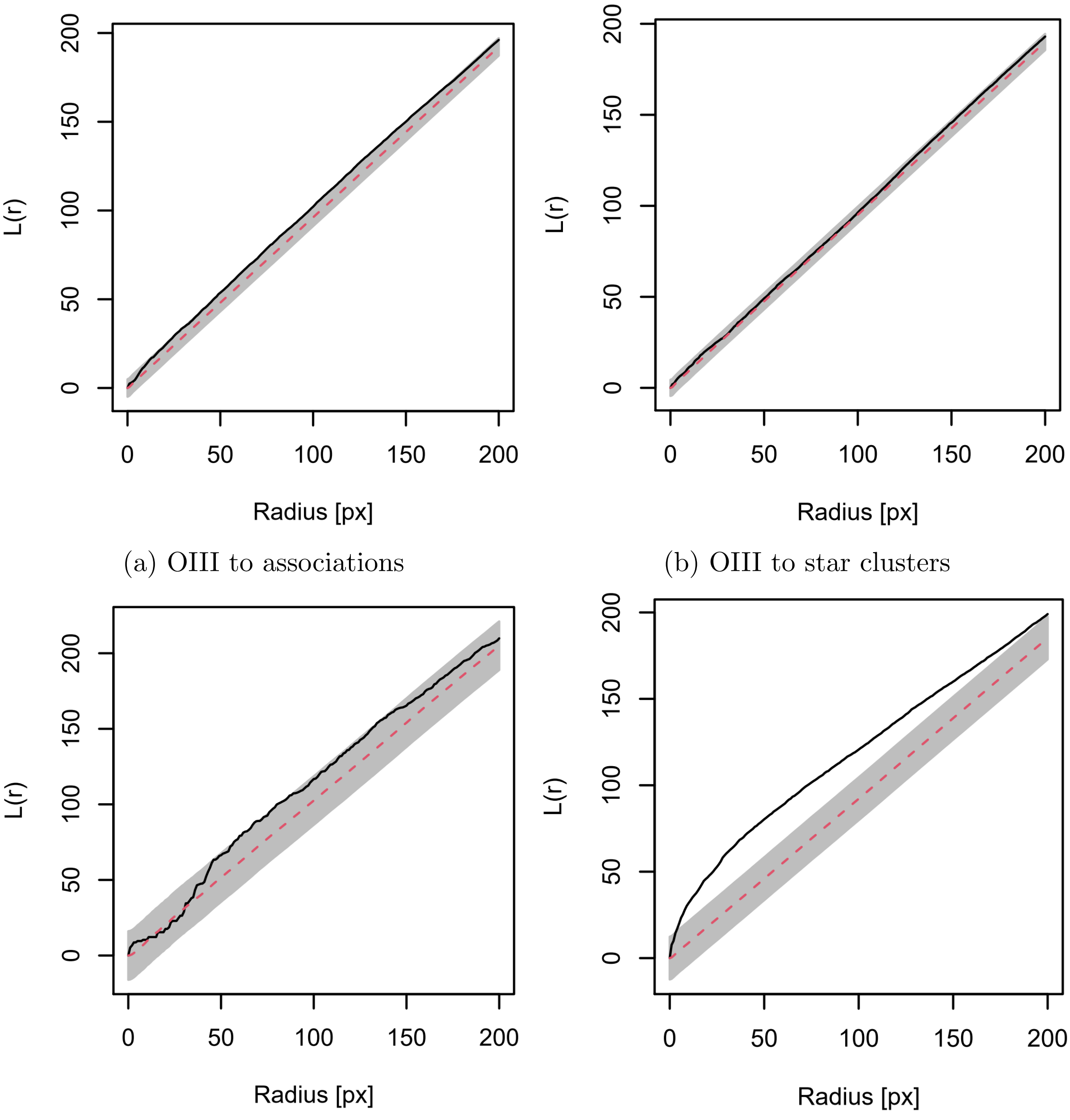
(a) SII to associations

(b) SII to star clusters

(c) SII to HI shells and supershells

(d) SII to emission nebulae

Figure 24: Correlation between bubbles found by the network in SII and different object categories.



(a) OIII to associations

(b) OIII to star clusters

(c) OIII to HI shells and supershells

(d) OIII to emission nebulae

Figure 25: Correlation between bubbles found by the network in OIII and different object categories.

6 FRI detection

Another interesting task that is similar to the bubble detection at a software level, is the detection of radio lobes in FRI galaxies (Section 3.3.1). As part of the increasing amount of data that is gathered by the new ASKAP radio telescope, a growing number of FRI objects is detected. One special region of the sky that was used here is the Emu in the sky (Section 4.5).

6.1 Framework

The framework is essentially the same as for the *Blobscan* network (Section 5). The major difference is the FRI detection method that is used to find FRI in a given region. As we discussed in section 5.1.1, the process of walking over the entire image with a given box size is very inefficient. For FRI a different approach is possible and described in the following section.

6.1.1 FRI detection

Since FRI essentially originate from a point like source we used the AEGEAN source finding tool by Hancock et al. (2012), that allows to create a catalog of sources from an astronomical image. With this catalog we can cutout the surrounding of every source and check this region with the network. This reduces the computational time immensely.

6.2 Network

In a similar way as the bubble detection that was discussed in section 5, the network that we used to detect FRI is based on the VGG16 model. The network was trained by a small amount of manually labeled data. The training data were cleaned from disturbing redundant sources and noise in the images. The network was applied to newly captured ASKAP radio data from the Large Magellanic Cloud.

6.2.1 Structure

The detailed architecture is given in Table 6. The main differences to the *Blobscan* network are the bigger input size of $50 \times 50 \times 3$, and the different output size of four different categories. The bigger input size is due to the more detailed structure which needs to be preserved. The four different classes for the output are used in order to differentiate between point sources, complex extended sources, verified and uncertain FRIs. Since this is a multi label problem also the activation function for classification has to be the softmax function and the loss function has to be the categorical cross-entropy. Again the network has the untouched layers from the original VGG16 architecture and the untrained fully connected, dropout and prediction layer.

Model:	"FRI"	Input:	"50 × 50 × 3"
Layer	(type)	Output Shape	Param #
vgg16	(Model)	(None, 1, 1, 512)	14714688
flatten	(Flatten)	(None, 512)	0
fc1	(Dense)	(None, 1024)	525312
dropout	(Dropout)	(None, 1024)	0
prediction	(Dense)	(None, 4)	4100
Total params:	15,244,100		
Trainable params:	7,608,836		
Non-trainable params:	7,635,264		

Table 6: Architecture of the FRI model.

6.2.2 Training

The Emu in the sky (Section 4.5) was one of the first pilot targets of the new ASKAP radio telescope (Section 4.4). The observed center frequency was $943.5MHz$. In the Emu region a set of 340 verified FRI and 346 uncertain FRI were already labeled manually and served as training data together with a set of 3000 other complex sources and 3000 point sources as counter examples. Complex sources here essentially include all extended sources, that are no point sources and also no FRI objects. As we want to use all of this data we either can just pool the verified and uncertain FRI as positive and the complex and point sources as negative training data. But this would ignore the fact, that there is a visible difference between uncertain and verified FRI. In order to use this additional information, we extended our two class network, that we described in the previous section, to a four class network.

Using the raw images of the FRI and their immediate surroundings did not lead to a satisfying result, since a majority of them contained disturbing redundant sources and noise. Therefore, the photutils source finding tool was used to identify and crop the central source in the image. Photutils is an affiliated package from Astropy by Bradley et al. (2019). This process can be exemplary seen in Figure 26. The left image is the original ASKAP data from the FRI object. In the second image the different sources within the image are identified with the photutils package. In the third image the central source is isolated and masked. In the fourth image the final cropped image is depicted. The cropped images where than used as training data. In Figure 27 six FRI as raw image are depicted in the first line and the corresponding cropped image that was used as training data in the second line. Due to the small training set, data augmentation was used. The data augmentation parameters are the same as for the bubble detection network and are listed in Table 5. The used training parameters are given in Table 7. As optimizer the RMSprop with a learning

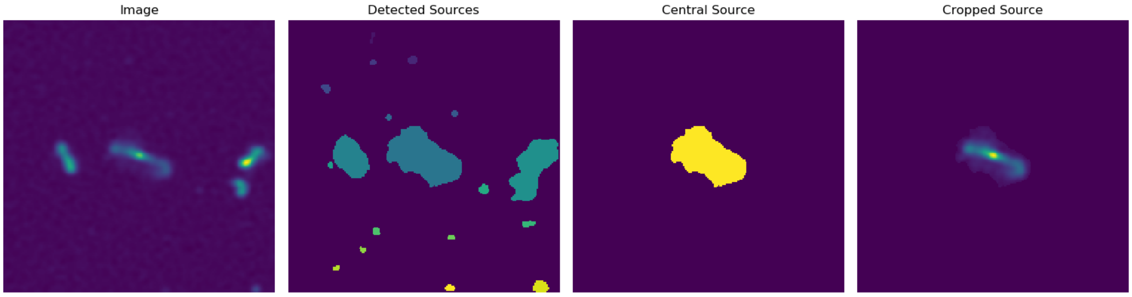


Figure 26: Exemplary cropping procedure for all sources. In the left image the original data is depicted. Using the photutils package all sources in the image can be detected and separated. The central source is separated and cropped in order to remove noise and unwanted sources.

Training parameter	Value
Validation fraction	0.2
Epochs	30
Batch size	50
Class Weights	1:1:10:10

Table 7: FRI detection training parameters. Here the class weights are associated to the classes complex, point source, FRI verified and FRI uncertain in that particular order.

rate of $\eta = 1 \cdot 10^{-4}$ is used. In order to prevent overfitting, the validation loss was monitored and after four epochs without decreasing the validation loss the training was stopped. The weights were set to the values for the minimal validation loss.

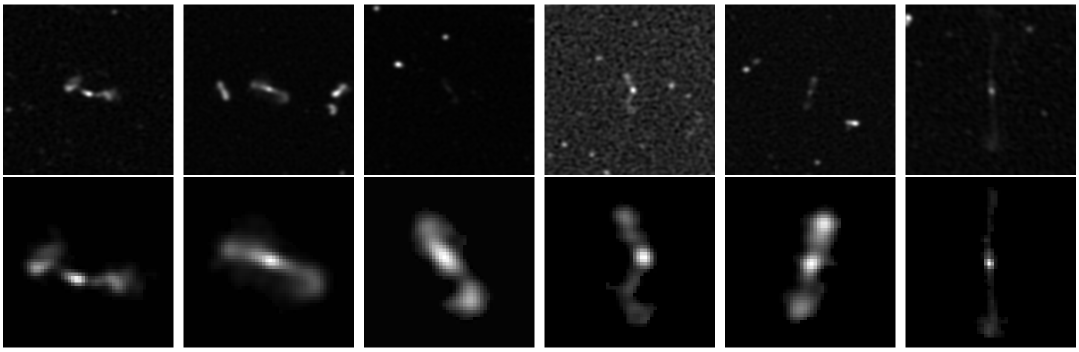


Figure 27: Exemplary set of 6 FRI objects in their original state (first line) and the clean cropped sources that were used as training data (second line).

6.3 Result

The FRI detection algorithm was applied to new ASKAP data at $888MHz$ from the Large Magellanic Cloud. A box size of 400" × 400" was tested. A total of 186 objects were classified as FRI. The detected FRI objects were checked for known objects in SIMBAD (Section 4.6). A total of 48 found FRI objects can be associated to already known galaxies, groups of galaxies, or cluster of galaxies. Three further FRI objects were associated to known active galactic nuclei. With the new ASKAP data, lobes from galaxies that were not identified as AGNs yet, might be visible.

A total of 30 found FRI objects can be associated to already known radio sources, 16 to already known x-ray sources. However, 98 found FRI objects could not be correlated to any known possible extragalactic source. Since the position of the origin source is not exactly known, a surrounding of 0.003 degree radius was queried from SIMBAD. It is not guaranteed that the found FRI object and the SIMBAD results are associated to the same source. A more thorough and detailed, individual analysis has to be done for that.

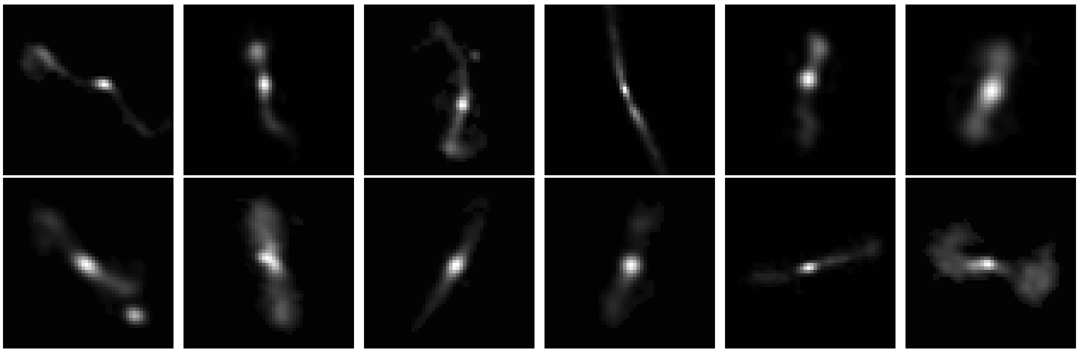


Figure 28: Exemplary set of 12 FRI objects in the LMC that were found by the network.

An exemplary set of twelve found FRIs is depicted in Figure 28. While a majority of the found objects could be verified as actual FRI, there are also a few objects that were mistakenly classified as FRI. Especially in regions with a lot of emission in a small area, the network performs poorly. One reason for this is the fact, that we trained the network only on clearly cropped sources. An overlap of multiple different sources is therefore not included in the training data, which prevents that the network does know how to classify them. Since the photutils source finding tool separates sources based on the contrast between objects and the background, an area of overlapping emissions is not separated into different sources.

Additionally there are many further possible FRI objects that can be seen in the data but that were not found by the network. This can be explained by the fact, that the photutils source finding algorithm separates sources by a certain threshold from the background in the image. A lot of the sources that can be found in the ASKAP data have very faint lobes however. For a high threshold this can result in the problem that very faint lobes are cropped from the source as background, preventing the network to identify it as a FRI object. Since the threshold between feature and background is different for every source it is hard to generalize a value as threshold.

In general we assume that one reason for the varying performance of the network is the small amount of training data, but also the fact that by cropping the training data with the use of the photutils package a strong definite border to the object is introduced. This adds additional unwanted features that might be learned from the network.

7 Summary and Outlook

Bubbles and bubble-like structures are circular, pancake-like or toroidal-like shaped objects in star forming regions of the universe. Their size and structure varies. We initially identified 83 bubbles within the SHASSA observation data. We used this as positive training set together with 830 random regions in the same data as negative training samples. We used the well known VGG16 convolutional neural network architecture and removed the last three layers that are most important for the final classification. These layers were replaced with new randomly initiated layers that are trained on our bubble data. The main reason for the VGG16 was the matter of fact that it can be obtained with already trained weights. This training was done on the Imagenet, a database of millions of RGB pictures. It allows to make the best of our very small training set. After training the adapted VGG16 we applied it on SPITZER data in order to find additional bubble-like structures that can be used as training data for a new iteration of training of the network. This was done repeatedly until the results of the network were satisfying. The final network was applied to MCELS data of the LMC and found 456 bubble-like structures in Hα, 288 in [OIII] and 267 in [SII]. Additionally a similar model was used to detect FRI objects in new ASKAP data. Therefore a set of 340 manually labeled FRI in the EMU region of the sky were used as training data together with over 3000 other complex sources as counter examples. For better results, the objects were cleaned from noise and adjacent sources. The network was applied to new ASKAP data from the LMC and found a total of 186 FRI objects. In order to improve the network for FRI detection, the application of the photutils package needs to be evaluated. For very faint lobes the differentiation between lobe and background needs to be extremely sensitive. It might be better to still use the cropped and cleaned FRI objects as training data like before, but with specifically added noisy background. This allows to apply the network to the real data without cropping the sources from the background with the photutils source finder. It avoids the problem that very faint lobes might be cropped too.

There are some important comments to this work. Even though the used VGG16 network has actually a three channel input - Red, Green and Blue channel - we only used one input channel, the detector intensity. A way better method would be to create a individual convolutional neural network with one channel for every available wavelengths from this area of the sky. In this way spectral information could be included in the classification process. However, this leads to the problem that usually the resolution for different wavelengths varies immensely. While pixels of radio-data have a resolution of about 30"/pixel and more, optical observations can achieve 5"/pixel and less. Rescaling every image to fit each other is possible but can introduce unwanted features in the data. It would also limit the amount of training data, which is very small anyway, since including all wavelengths information restricts the possible objects that can be used to those who are actually represented in all channels. So probably only actually confirmed bubbles could be used instead of just morphological, manually classified bubble-like structures. Nevertheless this would be an interesting approach with an uncertain outcome. In this work one could achieve a similar outcome by applying the *Blobscan* network to images in different wavelengths and applying individual decision criteria. E.g. one could consider something as bubble only if the network detects a bubble like structure in infrared and in Hα data. However, this requires a similar structure throughout the different wavelengths.

We used the pretrained VGG16 simply for the fact that it can be obtained pretrained and that the size and calculation speed is irrelevant for this work. However, it is possible to train a individual network on the Imagenet database also with all images in gray-scale which have only one channel. Since the Imagenet database was under maintenance for a major part of this work we were not able to try this. Using these additional images to pretrain an individual network could enable it to benefit even from a really small amount of training data. Like the wavelength sensible network mentioned above. Also the images of bubble-like structures, that can be gathered with the *Blobscan* model, could serve as additional training data for a new network.

For a new individual network it might be advantageously to use Maxpooling only in a close confined area since it may blur out important features. The size of the kernels for convolution should be chosen bigger since we are interested mostly in large scale structures from only one object that covers the entire picture. It might be possible to introduce a simulation tool for bubble-like structures that could generate additional training data.

A Appendix

A.1 VGG16 parameters

Model:	”vgg16”	Input:	”224 × 224 × 3”
Layer	(type)	Output Shape	Param #
input1	(InputLayer)	(None, 224, 224, 3)	0
block1conv1	(Conv2D)	(None, 224, 224, 64)	1792
block1conv2	(Conv2D)	(None, 224, 224, 64)	36928
block1pool	(MaxPooling2D)	(None, 112, 112, 64)	0
block2conv1	(Conv2D)	(None, 112, 112, 128)	73856
block2conv2	(Conv2D)	(None, 112, 112, 128)	147584
block2pool	(MaxPooling2D)	(None, 56, 56, 128)	0
block3conv1	(Conv2D)	(None, 56, 56, 256)	295168
block3conv2	(Conv2D)	(None, 56, 56, 256)	590080
block3conv3	(Conv2D)	(None, 56, 56, 256)	590080
block3pool	(MaxPooling2D)	(None, 28, 28, 256)	0
block4conv1	(Conv2D)	(None, 28, 28, 512)	1180160
block4conv2	(Conv2D)	(None, 28, 28, 512)	2359808
block4conv3	(Conv2D)	(None, 28, 28, 512)	2359808
block4pool	(MaxPooling2D)	(None, 14, 14, 512)	0
block5conv1	(Conv2D)	(None, 14, 14, 512)	2359808
block5conv2	(Conv2D)	(None, 14, 14, 512)	2359808
block5conv3	(Conv2D)	(None, 14, 14, 512)	2359808
block5pool	(MaxPooling2D)	(None, 7, 7, 512)	0
flatten	(Flatten)	(None, 25088)	0
fc1	(Dense)	(None, 4096)	102764544
fc2	(Dense)	(None, 4096)	16781312
predictions	(Dense)	(None, 1000)	4097000
Total params:	138,357,544		
Trainable params:	138,357,544		
Non-trainable params:	0		

Table 8: Detailed architecture of the VGG16 convolutional neural network.

A.2 Blobscan

A.2.1 Training Set

SHASSA Field	Number	Galactic l	Galactic b	Radius
010	0	302.0728095	-44.9388960	772.708”
	1	300.8995014	-44.3170519	625.286”
013	2	276.1817830	-34.0688809	422.159”
	3	276.0649106	-35.3160122	459.551”
	4	278.9384472	-36.3335636	459.551”
	5	279.2340101	-35.9585397	498.567”
	6	277.4972789	-34.9708914	459.551”
	7	278.9129046	-35.3948865	496.061”
	8	275.9927795	-31.8460168	486.817”
	9	277.9593940	-31.3595022	623.403”
	10	278.3152691	-30.1575000	459.551”
	11	281.3192422	-31.5081031	459.551”
	12	282.3497012	-33.0588235	459.551”
	13	282.8041404	-33.2245045	459.551”
	14	282.2386265	-35.1562435	459.551”
	15	281.5373287	-35.3022212	459.551”
	16	277.2058976	-33.6937498	459.551”
	17	278.6032462	-34.4743815	558.954”
	18	279.5860708	-35.7346619	498.567”
	19	278.1503577	-36.0224225	657.146”
	20	276.4361136	-32.5721192	463.300”
	21	275.8816972	-32.2945452	423.834”
	22	276.6422344	-32.0754184	590.355”
	23	279.7719894	-34.2447635	421.688”
	24	279.6190033	-34.4400041	490.539”
032	25	286.2386693	-0.1857618	1085.482”
034	26	300.6181809	1.0598222	1197.199”
	27	299.9854249	0.4296204	1141.249”
035	28	309.3060670	-0.4186289	1417.926”
	29	308.6506879	0.5687986	1453.707”
037	30	330.0454051	-8.1798681	3071.483”
053	31	277.8946287	0.5639412	2128.990”
	32	280.1520674	0.1700679	1860.119”
054	33	285.8714592	4.4492381	1330.095”
058	34	326.2148512	0.7891228	1236.694”
117	35	358.5620557	9.0513584	2967.020”
137	36	231.4809062	-4.5128546	1515.100”
138	37	232.5970566	0.8996234	2087.357”
	38	234.8096079	-0.2180313	1083.282”
	39	234.8247115	2.4212563	1360.338”
142	40	261.1347050	32.0977603	2124.590”
153	41	11.6402179	-1.7444381	1069.095”
	42	7.0220283	-0.2767767	894.931”
	43	15.0391832	3.3401482	1949.926”
242	44	194.6389092	-15.5366587	1393.471”
245	45	205.1236450	14.2320609	1561.845”
263	46	47.1253046	-2.6435391	2109.698”
	47	46.8107420	3.8493781	1233.733”
436	48	294.1980225	-26.1493947	1577.405”
512	49	281.5599315	-34.7593580	852.029”
	50	281.5417755	-35.2788069	852.029”
	51	279.7892334	-34.2502216	439.132”

	52	279.9286861	-33.4579367	783.611”
513	53	280.5208528	-30.6542048	952.923”
	54	281.8646021	-32.0881178	952.923”
	55	279.5638966	-35.5052208	952.923”
	56	281.5555355	-34.7728328	590.874”
	57	281.5519216	-35.3008771	758.496”
	58	279.3568496	-32.7569483	749.122”
	59	279.7817972	-34.2360862	508.282”
533	60	294.1089413	-2.3456250	882.740”
	61	300.6391449	1.0354132	843.946”
534	62	309.3411276	-0.4182920	1253.213”
	63	300.6065168	1.0684040	1246.343”
553	64	280.1438593	0.1823304	1549.034”
	65	277.9537859	0.6168734	2019.926”
637	66	234.7927634	-0.2291000	1007.687”
638	67	243.1801275	0.3513620	758.363”
653	68	3.9382672	-5.6764033	1170.122”
	69	11.6070131	-1.7695818	840.278”
673	70	232.5910701	0.8532783	2256.977”
	71	234.7935290	-0.2082726	1021.376”
	72	227.8173311	-0.0204504	1182.302”
689	73	11.6380883	-1.7415817	902.176”
	74	15.0574425	3.3305641	1189.711”
	75	16.6685094	-0.3374406	617.621”
	76	16.8046269	-1.0792631	534.470”
	77	18.6630285	1.9723136	387.727”
742	78	194.6391038	-15.5513066	2323.867”
743	79	212.0235023	-1.3467613	1505.644”
	80	206.3988087	-2.0106279	4447.440”
744	81	212.0166099	-1.3146378	1442.294”
762	82	36.3690918	-1.7271053	1613.390”
	83	31.8878536	1.4060049	906.893”

Table 9: First generation of training data. Manually labeled bubbles and bubble-like structures in SHASSA data.

Abbreviation	Extended Explanation
bub	Bubble
NA	No object found in SIMBAD for this region
smm	sub-millimetric source
Rad	Radio-source
Y*O|smm	Young Stellar Object|sub-millimetric source
Y*?|IR	Young Stellar Object Candidate| Infra-Red source
*|IR	Star|Infra-Red source
IR|FIR	Infra-Red source|Far-IR source ($\lambda >= 30\mu m$)
IR	Infra-Red source
cor|cor	Dense core
Mas	Maser
DNe|DNe	Radio-source
mm	millimetric Radio-source
HII|rad	HII (ionized) region|Radio-source
HII	HII (ionized) region
*	RStar
*|*iC	Star|Star in Cluster
*|*iC|IR	Star|Star in Cluster|Infra-Red source
Y*?|*|IR	Young Stellar Object Candidate|Star|Infra-Red source

Table 10: Extended explanation for the abbreviations that are returned from SIMBAD.

A.2.2 SIMBAD Abbreviation

A.2.3 LMC Results

Number	Galactic l	Galactic b	Radius	Number	Galactic l	Galactic b	Radius
0	282.6974501	-32.5960373	1600”	229	280.2415462	-31.5091994	200”
1	281.4215943	-32.3618801	1600”	230	280.3478315	-31.9410113	200”
2	279.5638541	-31.6727496	1600”	231	280.5799498	-32.9125916	200”
3	279.2863421	-32.8749761	1600”	232	280.1071158	-31.5114376	200”
4	278.2772105	-30.1940263	1600”	233	280.2289496	-32.6426245	200”
5	277.8931722	-32.4218839	1600”	234	280.0415605	-32.4116712	200”
6	276.6759657	-36.2555811	1600”	235	279.9937435	-32.5265864	200”
7	276.144443	-34.1010505	1600”	236	280.6650069	-35.4437004	200”
8	275.4462541	-33.858716	1600”	237	279.788338	-32.5530132	200”
9	282.5211	-32.6789175	1400”	238	279.3738298	-30.8709208	200”
10	281.8514375	-32.0310037	1400”	239	279.7087122	-33.0813288	200”
11	279.9323955	-33.39178	1400”	240	279.066848	-31.5805235	200”
12	279.3706592	-31.6434903	1400”	241	279.0741379	-31.6550843	200”
13	279.4408732	-35.4972941	1400”	242	279.1484702	-32.5129526	200”
14	278.9077055	-35.4050579	1400”	243	279.1294024	-32.494619	200”
15	278.5089893	-34.4615578	1400”	244	279.2127721	-33.3856358	200”
16	277.7321577	-33.0962809	1400”	245	279.1031779	-33.8029564	400”
17	276.6045428	-32.0817884	1400”	246	278.4796739	-31.8719014	200”
18	277.1647467	-36.0828353	1400”	247	279.0565529	-34.3626105	200”
19	275.8707566	-31.8922883	1400”	248	278.492731	-33.2606112	400”
20	280.4612808	-30.5919385	1200”	249	278.7592035	-35.0638807	200”
21	277.4599602	-33.1773826	1200”	250	278.0407806	-32.0061801	200”
22	278.0479891	-36.0858785	1200”	251	277.8741063	-32.066369	200”
23	276.23145	-32.1922679	1200”	252	278.4668626	-35.3002967	200”
24	275.9503089	-32.2829306	1200”	253	278.1133244	-35.1175555	200”
25	281.5645872	-34.7491323	1000”	254	277.2219773	-32.1501584	200”
26	281.4982262	-35.2727744	1000”	255	277.3209832	-32.6172915	200”
27	280.2461046	-31.1967767	1000”	256	277.9761755	-36.2894896	200”
28	279.3298834	-31.3138112	1000”	257	277.5433255	-35.7198568	200”
29	278.604731	-35.5912952	1000”	258	276.8066757	-33.5011595	200”
30	278.3203604	-35.042195	1000”	259	276.1947613	-32.6262325	200”
31	277.7263868	-33.7776433	1000”	260	276.2922111	-33.3345114	200”
32	277.5242934	-34.9511365	1000”	261	276.030877	-33.3166731	200”
33	276.1836254	-31.9347929	1000”	262	275.7329986	-33.6738736	200”
34	276.8575799	-35.9349463	1000”	263	276.9363798	-31.6112994	600”
35	276.1058168	-35.3214761	1000”	264	280.4056037	-32.8697021	300”
36	282.4553624	-32.2670044	900”	265	280.0194028	-31.9642981	300”
37	280.5739509	-30.8700052	900”	266	279.8322644	-31.4768894	300”
38	279.8710876	-32.7411435	900”	267	276.8998412	-31.9574986	300”
39	280.0623903	-33.8861722	900”	268	282.2212671	-31.9408327	200”
40	279.7386121	-34.2452205	900”	269	281.8710519	-32.3320694	200”
41	279.5978024	-34.4207308	900”	270	282.1499608	-34.1702287	200”
42	278.9703778	-32.3619917	900”	271	280.7523157	-31.500881	400”
43	278.9326442	-34.7194023	900”	272	281.0392834	-33.19869	200”
44	277.9762688	-32.9386716	900”	273	281.0253752	-34.125695	200”
45	278.3554306	-36.2108284	900”	274	280.0287867	-31.2816699	200”
46	277.1291161	-31.066903	900”	275	281.133798	-35.8782122	200”
47	277.1920352	-33.7011049	900”	276	279.7756502	-31.3236844	200”
48	276.0676176	-32.3922844	900”	277	280.0145816	-32.9522512	200”
49	282.2848835	-32.2209693	800”	278	280.550121	-35.4358513	200”
50	282.2747075	-33.0767239	800”	279	279.5733516	-32.5200917	200”
51	281.1843421	-31.2360796	800”	280	278.9860124	-31.7399471	200”
52	282.2347386	-35.1678111	800”	281	278.6458547	-33.1324934	200”
53	281.0213995	-32.3219732	800”	282	279.0948481	-35.0910987	200”
54	279.8080839	-33.9658856	800”	283	278.4617577	-33.4804696	200”
55	279.9998322	-35.3960393	800”	284	278.2441423	-33.2707048	200”

56	279.6649483	-35.8935323	800”	285	277.2610019	-31.3896121	200”
57	277.3280697	-31.2125542	800”	286	277.8351061	-34.6098768	200”
58	277.2454267	-35.927678	800”	287	277.0163784	-31.3047517	200”
59	276.4518664	-32.5600572	800”	288	277.463759	-36.0887751	200”
60	276.87907	-35.777138	800”	289	276.7148715	-35.6774964	200”
61	282.1888891	-31.2557059	600”	290	282.5961601	-33.3597344	600”
62	282.5271497	-35.9288636	600”	291	279.4065253	-34.9282668	400”
63	279.6673999	-30.6986906	600”	292	281.9309621	-31.0992006	300”
64	280.3714471	-34.0704438	600”	293	282.3856078	-33.9596771	300”
65	279.8710809	-35.5337825	600”	294	281.2988935	-31.7058069	300”
66	279.3067405	-33.3587009	600”	295	280.6920365	-34.7301504	300”
67	279.0502062	-34.4712001	600”	296	280.996767	-35.9160436	300”
68	278.6385808	-33.2388926	600”	297	279.9413277	-32.0969331	300”
69	277.6731388	-32.8569349	600”	298	280.1626727	-33.9891719	300”
70	277.9370757	-34.1887947	600”	299	280.0442821	-33.635155	300”
71	277.7369025	-35.3624035	600”	300	279.1288933	-34.6361789	300”
72	276.7559301	-34.2625517	600”	301	282.6153672	-34.5442688	200”
73	275.7618325	-33.1104578	600”	302	281.5496319	-33.674285	200”
74	275.7356824	-33.594789	600”	303	281.419384	-34.3680615	200”
75	280.4352033	-33.2220544	500”	304	281.3589994	-34.6064372	200”
76	280.8096191	-35.1726861	500”	305	280.00708	-31.0793935	200”
77	280.9266235	-35.9079514	500”	306	279.9263276	-30.9277613	200”
78	279.5681497	-31.1920981	500”	307	280.6443482	-35.2260297	200”
79	280.4758351	-35.272866	500”	308	279.5116435	-31.9862872	200”
80	280.6186103	-35.9196194	500”	309	280.0457808	-34.6615155	200”
81	280.6248763	-36.3379313	500”	310	279.4790715	-33.6857446	200”
82	279.4623062	-33.5570859	500”	311	278.4630008	-33.1511239	200”
83	279.7795254	-36.1870366	500”	312	276.8519616	-32.9917866	200”
84	279.1782603	-35.1127016	500”	313	280.3612495	-31.7656822	500”
85	279.2758661	-35.9342592	500”	314	281.9663688	-34.0517426	400”
86	279.2808192	-35.9543851	500”	315	281.3841253	-33.6888438	400”
87	279.250987	-35.9382648	500”	316	281.3405081	-34.252831	400”
88	279.2559343	-35.9583919	500”	317	280.3766649	-31.3559364	400”
89	278.8494128	-34.9143495	500”	318	280.1504136	-34.8954565	400”
90	277.3777204	-34.5971088	500”	319	278.5824685	-33.4215872	400”
91	276.5395028	-31.5511123	500”	320	282.1179291	-33.9463256	300”
92	276.7430826	-34.2305851	500”	321	280.4527606	-33.3455448	300”
93	276.5255036	-33.3029168	500”	322	280.8433056	-35.5623214	300”
94	276.339593	-33.2461505	500”	323	280.7126172	-35.9917885	300”
95	276.505232	-35.5142232	500”	324	280.4841214	-35.5170113	300”
96	282.583007	-33.797497	400”	325	280.2040922	-35.2236791	300”
97	281.8357818	-32.2500412	400”	326	278.9815113	-33.1921743	300”
98	282.0667574	-34.2377343	400”	327	279.5740918	-35.7760231	300”
99	280.7814927	-32.2236025	400”	328	277.5272562	-31.8682519	300”
100	280.3216041	-31.829427	400”	329	277.7688704	-34.0648275	300”
101	280.1256941	-31.8470396	400”	330	277.5591559	-34.2704308	300”
102	279.7152851	-31.1372178	400”	331	277.8183695	-36.2238684	300”
103	280.8026817	-35.6437608	400”	332	275.921312	-33.279605	300”
104	280.0561763	-32.7947421	400”	333	281.9776124	-31.6146881	200”
105	280.6696655	-35.6159012	400”	334	281.8711023	-34.0496829	200”
106	280.6936982	-35.8669298	400”	335	281.3961534	-34.3893554	200”
107	280.3055793	-35.7465978	400”	336	280.1722232	-31.0184859	200”
108	280.096127	-35.2550785	400”	337	281.2559188	-35.3682104	200”
109	279.4579468	-33.2540982	400”	338	281.27905	-35.5707687	200”
110	278.641381	-33.356727	400”	339	280.0418109	-30.7495272	200”
111	278.8227744	-34.8020651	400”	340	280.1737784	-31.4985352	200”
112	277.8426452	-31.5493417	400”	341	280.000646	-30.9073237	200”
113	277.5490469	-32.0352155	400”	342	280.2690211	-33.8953002	200”
114	277.2590342	-32.0796962	400”	343	280.5338449	-35.5634381	200”

115	277.1541382	-31.9603465	400”
116	282.5307596	-33.750069	300”
117	282.3509776	-34.6239267	300”
118	281.1528804	-33.9452931	300”
119	281.1814199	-34.3402258	300”
120	280.7970177	-33.0344875	300”
121	280.6317887	-32.6768347	300”
122	280.3098902	-31.8604221	300”
123	280.0815735	-31.5267012	300”
124	280.1943242	-32.1677503	300”
125	280.3783113	-32.9091348	300”
126	280.1778299	-32.2205508	300”
127	280.3876345	-34.1936208	300”
128	280.8046912	-35.9185053	300”
129	279.8993382	-32.5310403	300”
130	280.397899	-36.0136291	300”
131	280.4010058	-36.0255644	300”
132	280.0574408	-35.2860039	300”
133	278.3969857	-32.1593526	300”
134	279.0016907	-35.0862868	300”
135	278.7349586	-34.7306365	300”
136	278.3392948	-33.2132419	300”
137	277.7715565	-31.8488201	300”
138	277.4814327	-31.3110496	300”
139	277.26889	-32.2366267	300”
140	278.1735112	-36.7058588	300”
141	276.8365609	-33.4793332	300”
142	276.4361747	-33.2644697	300”
143	283.9477971	-31.8361471	200”
144	283.0229956	-34.717433	200”
145	281.878406	-31.2245659	200”
146	281.9070914	-31.5599762	200”
147	282.6030325	-34.2607218	200”
148	282.2510895	-34.182927	200”
149	281.4412321	-31.6139638	200”
150	281.9515222	-34.2565408	200”
151	281.8798873	-34.3723507	200”
152	280.7077811	-31.4000667	200”
153	281.0967575	-33.5951443	200”
154	280.732949	-32.5360716	200”
155	280.6047867	-32.7889499	200”
156	280.5795572	-33.2107233	200”
157	280.0927663	-31.7394805	200”
158	280.3307711	-32.7864246	200”
159	280.7511016	-35.2754675	200”
160	280.7852836	-35.4056712	200”
161	279.846234	-31.9610169	200”
162	280.2782702	-33.7149363	200”
163	279.8634113	-33.6502763	200”
164	280.3273068	-36.1820008	200”
165	279.6392314	-33.5267602	200”
166	279.8480356	-34.4183685	200”
167	279.2531588	-31.9944515	200”
168	279.4081872	-34.1774702	200”
169	279.6040631	-35.020264	200”
170	279.6630243	-35.3418387	200”
171	278.7880132	-31.8429057	200”
172	279.3904902	-34.5287869	200”
173	278.8187685	-32.4571394	200”
344	280.5409772	-35.9415071	200”
345	279.4647243	-33.4966158	200”
346	279.3945851	-34.9868547	200”
347	278.8079961	-33.327479	200”
348	278.1385812	-31.6716021	200”
349	278.3343582	-33.3972154	200”
350	278.3574417	-35.6008223	200”
351	277.5395147	-32.2826508	200”
352	277.2997125	-32.1524349	200”
353	277.3606199	-35.2431557	200”
354	281.2140161	-31.8481676	300”
355	276.4764099	-33.4975064	300”
356	281.0955554	-31.3671227	200”
357	281.8397715	-34.5715359	200”
358	279.9508781	-32.8991832	200”
359	280.7346181	-36.3348702	200”
360	279.2285118	-34.6040892	200”
361	277.5572612	-32.7810271	200”
362	277.0646831	-32.0155678	200”
363	276.9828685	-34.6784968	200”
364	277.4893078	-34.3723199	400”
365	280.7490125	-33.4304257	300”
366	278.1442353	-34.2981091	300”
367	277.0021868	-33.684555	300”
368	281.7771039	-34.3363016	200”
369	281.009293	-35.7322767	200”
370	279.9958894	-31.834666	200”
371	278.0318467	-35.1932297	200”
372	277.0572881	-35.7096272	200”
373	282.363114	-35.6533216	1000”
374	277.755864	-33.9386138	300”
375	278.9478063	-32.9758089	200”
376	282.0386291	-31.1676845	600”
377	276.713037	-35.8116862	600”
378	280.933606	-32.9019502	500”
379	280.534648	-31.4199992	500”
380	277.5676309	-35.1246771	500”
381	277.0289774	-34.1525843	500”
382	280.2072324	-31.4722159	300”
383	281.1159576	-35.5953332	300”
384	282.5953045	-34.7326474	200”
385	281.5540253	-31.2691154	200”
386	281.3319371	-31.2835237	200”
387	281.9915804	-33.892776	200”
388	280.8919706	-30.7553436	200”
389	281.3125143	-34.3829774	200”
390	280.7664023	-32.4636237	200”
391	280.4319642	-32.3418443	200”
392	280.2531713	-31.7140077	200”
393	280.2237352	-32.0043707	200”
394	280.0109567	-32.1437493	200”
395	280.5695821	-35.5453492	200”
396	277.9407706	-32.1151531	200”
397	277.6693649	-31.6921886	200”
398	278.0817342	-34.2456194	200”
399	277.5414633	-32.5836653	200”
400	277.662091	-34.1646003	200”
401	278.3919459	-33.7392462	300”
402	280.4515483	-33.4780224	200”

174	278.3105478	-30.777628	200”
175	278.4813669	-33.4535911	200”
176	277.9086434	-31.8831297	200”
177	277.922251	-31.9910518	200”
178	277.7784104	-33.4449477	200”
179	277.7802177	-33.4531417	200”
180	277.7829625	-34.1558636	200”
181	277.137581	-31.1995896	200”
182	277.4480719	-32.7164477	200”
183	277.1139497	-31.2285357	200”
184	277.9455649	-35.0629162	200”
185	277.9534369	-35.1887557	200”
186	277.2050738	-31.9497981	200”
187	277.7433715	-34.576895	200”
188	277.9542732	-36.3310891	200”
189	277.3971218	-35.9057656	200”
190	276.6355623	-33.3107172	200”
191	275.846585	-33.4052446	200”
192	275.8164599	-33.6624638	200”
193	276.2051242	-31.3919916	1600”
194	278.9945163	-36.3702265	1400”
195	282.2663136	-32.6115476	1000”
196	277.9720615	-31.3471719	1000”
197	280.1738549	-31.2265984	900”
198	279.9151285	-31.5836379	800”
199	277.7011664	-32.1538026	600”
200	278.160749	-36.3903852	600”
201	280.4237548	-33.2762914	500”
202	279.5917978	-34.090772	500”
203	278.025894	-34.6039398	500”
204	282.4068768	-32.4221478	400”
205	280.2090684	-33.5340337	400”
206	278.7978947	-32.2936905	400”
207	279.6033941	-35.7451769	400”
208	278.9809094	-34.5984461	400”
209	277.0920489	-35.910213	400”
210	282.1268368	-31.0677141	300”
211	281.0786492	-31.3032926	300”
212	280.5917673	-31.5480599	300”
213	280.1673613	-31.8482058	300”
214	280.8725456	-34.8391186	300”
215	280.1700953	-32.4589574	300”
216	280.9024857	-35.8514911	300”
217	279.7165104	-31.2230209	300”
218	279.8273543	-32.082141	300”
219	279.6132197	-31.3404958	300”
220	280.2236531	-33.8725265	300”
221	279.9001487	-32.8988605	300”
222	278.35395	-32.8191088	300”
223	278.8658776	-35.0271216	300”
224	278.4014849	-33.1598124	300”
225	277.4114292	-34.2817891	300”
226	276.5575212	-35.8732171	300”
227	281.3194705	-31.5089323	200”
228	281.979201	-34.0937198	200”
403	279.2754678	-34.2747272	200”
404	278.4103893	-33.1900594	200”
405	279.4805556	-36.9005706	400”
406	281.3808388	-31.3856448	300”
407	280.8329006	-35.7950414	200”
408	279.2753809	-31.1624818	200”
409	279.9085546	-34.1314047	200”
410	277.4418663	-35.2109376	200”
411	276.1686155	-33.2219501	300”
412	280.7147812	-35.4900553	200”
413	278.0732661	-34.0480218	200”
414	277.3770595	-32.234243	200”
415	280.7524469	-34.9011904	200”
416	278.4929337	-33.8058084	200”
417	278.3506023	-33.2481979	200”
418	279.9188993	-30.9573609	300”
419	277.4211707	-36.1135428	300”
420	282.2512724	-30.8958649	200”
421	283.1292386	-35.2042764	200”
422	281.8930197	-31.3001615	200”
423	280.3691002	-32.4914236	200”
424	279.3412856	-34.5697427	200”
425	278.6404347	-32.2165364	200”
426	277.7942178	-31.92878	200”
427	276.0930064	-33.2472113	200”
428	280.1635101	-34.0798029	300”
429	279.0049061	-31.1997	300”
430	276.9057127	-32.9500535	300”
431	280.1680152	-31.425899	200”
432	280.2350414	-31.7705012	200”
433	278.4533632	-33.2748177	200”
434	277.4947055	-31.3518153	200”
435	277.974767	-36.4621921	200”
436	280.3568013	-31.3804521	300”
437	280.2836245	-31.4422249	200”
438	278.1731259	-33.4028503	200”
439	278.0527793	-33.3793004	200”
440	277.9702774	-36.4155146	200”
441	276.3453023	-35.4462657	200”
442	277.0054459	-33.604008	200”
443	279.6669901	-33.1995271	200”
444	278.3888984	-33.3266184	400”
445	280.3560688	-33.8421283	200”
446	279.2745719	-34.3677584	200”
447	278.201341	-33.3718502	200”
448	280.4019522	-31.4477645	200”
449	277.6472239	-36.1966627	200”
450	281.2671972	-31.3625585	200”
451	280.8250374	-31.8161468	200”
452	277.6407276	-34.5313843	200”
453	278.4916177	-33.267576	200”
454	280.4161597	-33.0789694	200”
455	280.3237737	-32.9045478	200”
456	280.2318955	-31.4100487	200”

Table 11: Bubbles that were detected by the network in the LMC at the Hα emission line. Note that the Radius is actually the size of the box that was used to query the network. It is not the actual size of the bubble but the bubble is contained within the radius.

Number	Galactic l	Galactic b	Radius	Number	Galactic l	Galactic b	Radius
0	282.6974501	-32.5960373	1600”	145	276.7163087	-35.826985	600”
1	282.4725237	-32.366694	1600”	146	279.4043869	-31.2077738	400”
2	277.1370971	-31.1180322	1600”	147	279.2781952	-31.3904698	400”
3	278.2170803	-36.0321311	1600”	148	280.0493378	-34.6592085	400”
4	276.6759657	-36.2555811	1600”	149	279.0591961	-31.6479735	400”
5	280.5782479	-30.5990789	1400”	150	277.9254817	-32.3405862	400”
6	281.2846701	-34.4647612	1400”	151	277.5442595	-35.1343464	400”
7	279.9323955	-33.39178	1400”	152	276.1725781	-34.1393529	400”
8	279.3840976	-31.7006256	1400”	153	282.0498684	-34.2488826	300”
9	278.894	-35.3478595	1400”	154	281.2918882	-31.7071081	300”
10	277.7321577	-33.0962809	1400”	155	279.9220681	-30.9568044	300”
11	277.1371415	-31.2925598	1400”	156	279.9307676	-32.1140719	300”
12	279.2261785	-34.731301	1200”	157	279.4963684	-31.9894588	300”
13	278.6081589	-34.4741734	1200”	158	279.7374415	-34.2488189	300”
14	282.3254662	-32.6428318	1000”	159	278.7825863	-32.2707514	300”
15	281.1533475	-31.2446616	1000”	160	278.3189632	-30.7621558	300”
16	281.4982262	-35.2727744	1000”	161	278.192706	-33.3545549	300”
17	279.2805743	-32.8079778	1000”	162	277.7603217	-33.9249449	300”
18	279.0174029	-32.9790771	1000”	163	277.7629846	-33.9369633	300”
19	277.3606184	-33.0287674	1000”	164	277.7911788	-34.131458	300”
20	279.8799775	-32.7775976	900”	165	277.0048543	-33.6779609	300”
21	280.0623903	-33.8861722	900”	166	276.2895192	-33.1615032	300”
22	279.385171	-32.7859015	900”	167	275.5924048	-34.5939077	300”
23	279.4841638	-35.4641705	900”	168	282.9169266	-35.1443272	200”
24	277.9762688	-32.9386716	900”	169	281.2271743	-31.8707207	200”
25	277.8521119	-32.5796347	900”	170	281.7961593	-34.0466067	200”
26	278.364028	-36.2473226	900”	171	281.8752989	-34.3732067	200”
27	277.8578838	-34.2438434	900”	172	280.1042669	-31.8904818	200”
28	281.8705781	-32.0273668	800”	173	280.4184852	-33.239189	200”
29	282.2347386	-35.1678111	800”	174	279.6847667	-31.7470724	200”
30	280.1839162	-31.2134865	800”	175	279.6981465	-34.3121135	200”
31	279.4509315	-31.3072823	800”	176	278.9394079	-34.8229989	200”
32	277.9277437	-31.4230431	800”	177	278.5079152	-33.2796744	200”
33	278.602796	-35.6565749	800”	178	277.5992651	-34.3191593	200”
34	278.6105407	-35.6892667	800”	179	276.721005	-31.9463815	200”
35	277.7302791	-32.1650715	800”	180	277.5541376	-36.2547737	200”
36	276.3205674	-32.1062889	800”	181	281.3446151	-32.3761611	1600”
37	279.4172842	-31.5308627	600”	182	282.0204407	-31.1541443	600”
38	278.4829765	-30.6958064	600”	183	276.4750219	-32.539252	500”
39	278.2819976	-35.0255286	600”	184	277.0235933	-36.1479208	400”
40	277.5209431	-34.2518906	600”	185	279.9070989	-32.5325328	300”
41	276.4924604	-33.2862569	600”	186	279.7119321	-36.0137005	300”
42	275.9584006	-32.2733541	600”	187	281.3455715	-34.2382654	200”
43	282.3265366	-33.0627749	500”	188	280.6848417	-32.5748005	200”
44	282.453715	-34.3446803	500”	189	280.2719798	-31.0641617	200”
45	280.3465533	-31.7682966	500”	190	279.7801066	-31.3229095	200”
46	280.8805227	-35.8320839	500”	191	280.7780417	-35.9157776	200”
47	279.0315948	-32.3716493	500”	192	279.4597439	-33.2277317	200”
48	279.2808192	-35.9543851	500”	193	278.3311238	-33.3949004	200”

49	277.2593067	-32.0683796	500”
50	276.5497087	-33.299437	500”
51	277.1118081	-35.9496265	500”
52	276.1368708	-32.3588103	500”
53	276.1999168	-34.0364935	500”
54	275.7594352	-33.0776154	500”
55	281.8411044	-31.208974	400”
56	281.954617	-34.0539495	400”
57	281.6221514	-35.4973876	400”
58	279.8683154	-33.9329733	400”
59	279.4158332	-34.9437261	400”
60	278.8737647	-34.9296949	400”
61	278.0633847	-33.4629755	400”
62	277.55256	-32.0515952	400”
63	277.1835588	-33.7153353	400”
64	277.3994805	-36.1194988	400”
65	276.9356608	-35.8141813	400”
66	282.6291591	-35.2846873	300”
67	282.4254198	-34.8351879	300”
68	280.8958539	-31.5053586	300”
69	281.0253004	-33.1682695	300”
70	280.1641653	-31.0261446	300”
71	280.1990302	-31.4685691	300”
72	280.4579644	-33.344628	300”
73	280.1630076	-32.4601995	300”
74	279.7399583	-31.1380233	300”
75	280.9089561	-35.9626647	300”
76	280.7909976	-35.6338906	300”
77	280.3717226	-34.0715308	300”
78	279.8344171	-32.0809184	300”
79	280.2356152	-33.7705836	300”
80	280.230856	-33.8712784	300”
81	279.7067775	-31.9783205	300”
82	279.692664	-31.9807505	300”
83	279.7856349	-34.321759	300”
84	278.7997088	-33.3384453	300”
85	278.3992895	-33.7386502	300”
86	278.0022982	-34.2358323	300”
87	277.6869495	-32.8802136	300”
88	277.8181063	-36.2379987	300”
89	277.1023476	-34.7305274	300”
90	276.3951279	-32.6262329	300”
91	283.0088049	-35.8222903	200”
92	281.9780184	-34.0893894	200”
93	281.6533797	-34.1840393	200”
94	281.6075776	-34.2436846	200”
95	281.4120919	-33.665515	200”
96	280.6754294	-31.6696462	200”
97	280.9606405	-32.912394	200”
98	281.1779316	-34.1004601	200”
99	280.2576992	-31.718874	200”
100	280.2917842	-31.8573956	200”
101	280.0723272	-32.0661338	200”
102	281.0022754	-35.7505383	200”
103	279.9033307	-31.5347228	200”
104	280.0834848	-32.7276055	200”
105	280.742508	-35.3620908	200”
106	279.9072037	-32.1714398	200”
107	280.2214335	-33.5290474	200”
194	278.2880013	-33.2852568	200”
195	275.9938785	-35.1842535	200”
196	277.5595995	-32.2705282	400”
197	280.6402454	-35.6448313	300”
198	279.9291454	-31.4158121	200”
199	280.8001335	-35.1630021	200”
200	279.7612896	-31.41902	200”
201	280.2812776	-33.7268977	200”
202	279.4915424	-32.0282194	200”
203	279.6761986	-34.3707764	200”
204	278.9086014	-32.3874781	200”
205	279.4667923	-36.7615658	200”
206	277.3959687	-36.0287519	200”
207	276.1913157	-31.4288392	200”
208	275.9752227	-33.6698511	200”
209	275.8180817	-33.6706774	200”
210	281.7663019	-32.1881342	500”
211	280.5936783	-31.5477156	300”
212	280.0188107	-31.9536334	300”
213	280.0141187	-32.1539678	300”
214	280.8579766	-35.5597468	300”
215	280.5386036	-35.4531585	300”
216	280.1691687	-34.8848805	300”
217	278.5613465	-31.3026916	300”
218	277.6973382	-32.0714467	300”
219	277.1817043	-33.6749965	300”
220	280.3136187	-32.0406482	200”
221	279.7792273	-31.9046381	200”
222	279.8785094	-32.4150324	200”
223	280.5951595	-35.4643478	200”
224	280.6990261	-35.982806	200”
225	280.471482	-35.2560345	200”
226	279.5989207	-35.6293052	200”
227	278.1030663	-35.1179943	200”
228	277.2678568	-32.2474968	200”
229	276.1696145	-33.3039786	200”
230	276.5124217	-35.7945086	200”
231	280.4109035	-33.0742201	200”
232	280.2818584	-35.7359027	200”
233	279.866747	-35.5248769	200”
234	278.4136193	-33.1804969	200”
235	278.5730573	-35.5425573	200”
236	280.6465088	-35.6074459	400”
237	280.9275611	-32.9047478	300”
238	276.8634496	-33.4658266	300”
239	280.4261298	-33.1078614	200”
240	279.6179673	-31.2151849	200”
241	279.7348227	-34.1908722	200”
242	278.8648326	-32.3720613	200”
243	276.8185825	-33.4870262	200”
244	276.9890569	-35.5166028	200”
245	279.4678897	-36.8867922	300”
246	277.0685786	-35.933262	400”
247	280.4825714	-35.5110406	300”
248	279.2483908	-31.3621565	300”
249	278.0240468	-35.189334	300”
250	277.7702045	-34.0708365	300”
251	281.6292472	-34.1020495	200”
252	280.0488734	-30.7460144	200”

108	280.2620308	-33.8539538	200”	253	280.2277162	-31.7678289	200”
109	279.8381989	-32.2599019	200”	254	280.2192731	-31.9943852	200”
110	279.8474927	-32.547362	200”	255	278.9694396	-34.8419397	200”
111	280.0346305	-34.216519	200”	256	278.2276459	-33.3654485	200”
112	279.9151188	-34.0667745	200”	257	278.1958971	-36.399751	200”
113	279.3939192	-32.5482315	200”	258	278.1448211	-36.6915589	200”
114	279.4549934	-33.566243	200”	259	277.9560746	-36.4238277	200”
115	278.9942037	-31.740843	200”	260	277.3327815	-35.4744124	200”
116	279.1129533	-32.5103779	200”	261	276.3409871	-33.2532583	200”
117	279.0528811	-32.5118709	200”	262	276.0835236	-33.2468529	200”
118	278.3275322	-30.1407083	200”	263	275.7094297	-33.5949976	200”
119	278.343664	-30.2141164	200”	264	280.695876	-35.8433031	500”
120	279.2697861	-34.28529	200”	265	281.8664906	-31.2978473	200”
121	279.2460091	-34.3146822	200”	266	280.7399066	-36.3362222	200”
122	278.9786987	-34.7912189	200”	267	276.4920343	-33.4947103	200”
123	279.1867116	-36.1238446	200”	268	280.1388753	-31.8299836	200”
124	278.4295011	-33.3600564	200”	269	280.5711795	-35.5354311	200”
125	278.3982357	-33.3565475	200”	270	280.5616367	-31.397566	300”
126	277.8391971	-31.563989	200”	271	280.7630921	-33.4296156	200”
127	277.8786157	-32.0656604	200”	272	279.8768116	-32.7010379	600”
128	278.607843	-35.9862451	200”	273	280.0904773	-31.5223007	200”
129	277.5227623	-31.8503368	200”	274	279.9504659	-35.423609	200”
130	277.4443255	-32.1838982	200”	275	277.4082268	-35.8962651	200”
131	277.2050738	-31.9497981	200”	276	276.6434635	-35.7195355	200”
132	277.3574565	-32.8571622	200”	277	279.8301789	-32.1796833	200”
133	277.4494081	-33.7998074	200”	278	280.4001606	-36.0115432	200”
134	277.0164257	-32.0966441	200”	279	278.8426225	-34.8102103	200”
135	276.9220362	-32.0262804	200”	280	279.8079666	-36.2106352	200”
136	277.1299791	-33.6866012	200”	281	276.2831389	-33.329604	200”
137	276.6789678	-32.1046159	200”	282	281.1171858	-33.9841424	200”
138	276.5614546	-32.1641054	200”	283	279.6201104	-34.3268965	200”
139	277.0720004	-36.5180697	200”	284	280.4865311	-33.2345708	200”
140	276.8264726	-35.9196159	200”	285	279.8312944	-31.4866845	200”
141	276.0431039	-33.4286222	200”	286	279.5865415	-35.7643588	200”
142	275.1144094	-35.4572508	200”	287	275.997288	-33.3033541	200”
143	275.3940631	-33.7982569	1600”	288	280.2861016	-31.3981513	200”
144	277.1492069	-36.0597168	800”				

Table 12: Bubbles that were detected by the network in the LMC at the [OIII] emission line. Note that the Radius is actually the size of the box that was used to query the network. It is not the actual size of the bubble but the bubble is contained within the radius.

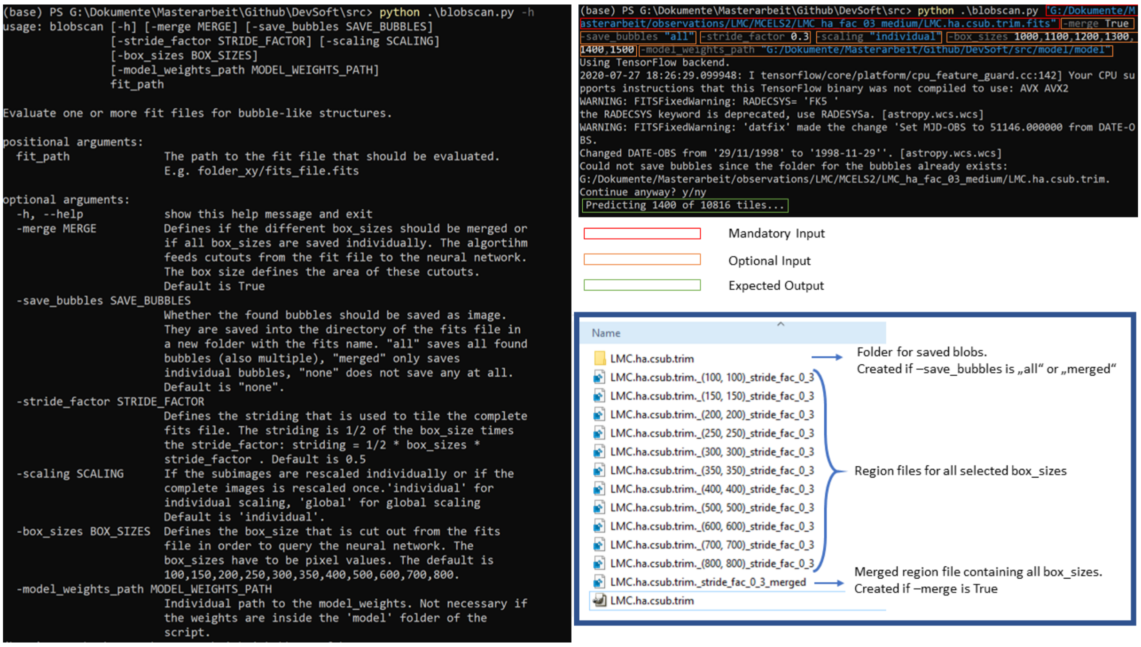
Number	Galactic l	Galactic b	Radius	Number	Galactic l	Galactic b	Radius
0	277.8788092	-32.3563761	1600”	134	279.2533934	-35.9480564	500”
1	278.8121635	-36.4146535	1600”	135	280.9130512	-32.8982827	400”
2	277.8012408	-32.3685006	1600”	136	275.7597452	-33.0826329	400”
3	277.8155471	-32.4340166	1600”	137	282.7065049	-32.6067898	300”
4	276.243532	-31.386411	1600”	138	282.6218896	-35.2860796	300”
5	280.5400999	-30.7249053	1400”	139	279.7948831	-31.2156203	300”
6	281.4451486	-34.7936835	1400”	140	279.6082466	-35.7534356	300”
7	276.5928778	-32.0243939	1400”	141	277.7715582	-33.9418677	300”
8	277.1521331	-36.0255169	1400”	142	276.9333337	-32.9504853	300”
9	281.400838	-32.4112009	1200”	143	276.9916765	-33.6860968	300”
10	275.8920994	-32.291137	1200”	144	281.788937	-32.2304781	200”
11	278.345439	-36.2665204	1000”	145	281.3256765	-31.2824217	200”
12	276.2838185	-33.186519	1000”	146	280.1814927	-31.0185383	200”

13	281.4504876	-35.3149574	900”
14	279.7386121	-34.2452205	900”
15	279.5978024	-34.4207308	900”
16	278.4326435	-33.3589904	900”
17	277.4170093	-33.0625576	900”
18	276.8457051	-35.9411159	900”
19	282.3625193	-35.6215823	800”
20	281.0297458	-32.3545256	800”
21	280.2243759	-33.8264396	800”
22	279.5271626	-35.5086318	800”
23	277.9206279	-31.3903265	800”
24	278.3368474	-35.0537676	800”
25	276.458517	-32.5928747	800”
26	276.2365875	-32.0846501	800”
27	282.3300491	-32.2264564	600”
28	282.5864071	-33.380997	600”
29	282.3047695	-33.0254006	600”
30	280.157938	-31.5017319	600”
31	279.9242043	-35.3719255	600”
32	277.6731388	-32.8569349	600”
33	276.1994649	-34.0877748	600”
34	281.3846339	-31.3900447	500”
35	280.531467	-31.4205713	500”
36	280.3988826	-31.3814341	500”
37	280.1615247	-31.2140573	500”
38	280.7779794	-35.6401368	500”
39	277.8623044	-31.5575192	500”
40	277.7102265	-32.1032968	500”
41	277.1565736	-33.6905642	500”
42	276.6982686	-35.8204981	500”
43	282.583007	-33.797497	400”
44	280.7856163	-32.2398869	400”
45	280.3438721	-31.7574341	400”
46	280.1256941	-31.8470396	400”
47	279.7152851	-31.1372178	400”
48	280.891695	-35.9004623	400”
49	280.0521613	-32.7784253	400”
50	279.9200317	-33.3967374	400”
51	279.8526697	-33.9526456	400”
52	279.8577961	-35.5501617	400”
53	279.4118392	-34.9273935	400”
54	278.9824702	-34.6915764	400”
55	278.1958894	-36.3907293	400”
56	276.7537031	-36.2279405	400”
57	275.4896012	-33.7686249	400”
58	281.9581957	-32.031718	300”
59	281.3057243	-31.5046789	300”
60	281.9705906	-34.069729	300”
61	281.1826506	-31.2528476	300”
62	281.046195	-31.2905556	300”
63	281.3945218	-33.6636244	300”
64	280.6317887	-32.6768347	300”
65	280.7490125	-33.4304257	300”
66	280.3232885	-31.7956766	300”
67	280.3640696	-32.911644	300”
68	280.1631294	-32.1607604	300”
69	280.6798966	-34.7419332	300”
70	280.1504552	-34.8710646	300”
71	279.6369016	-34.0849079	300”
147	280.2964185	-31.4699845	200”
148	280.4483928	-33.4763068	200”
149	280.0435582	-32.4198279	200”
150	280.3910209	-34.2015445	200”
151	279.8247712	-32.0797549	200”
152	280.5862628	-35.5424608	200”
153	279.6648775	-33.1908165	200”
154	280.0592552	-35.2794622	200”
155	279.4611828	-33.2337228	200”
156	279.1433677	-32.3394546	200”
157	279.4699204	-35.4036684	200”
158	278.6434697	-33.1221296	200”
159	278.6001493	-33.2394234	200”
160	278.4128084	-33.2185131	200”
161	278.3578134	-33.2442258	200”
162	278.1246462	-32.7469956	200”
163	278.1179819	-32.9220881	200”
164	277.8720878	-32.9515195	200”
165	277.1358802	-31.1914047	200”
166	277.0232932	-32.1040649	200”
167	275.7245699	-33.7340598	200”
168	279.2465493	-31.3115712	600”
169	277.5268918	-35.1493272	400”
170	282.6404726	-32.4631302	300”
171	279.5698982	-31.7430455	300”
172	279.3208268	-31.5557674	300”
173	279.4780871	-36.8851527	300”
174	281.8759205	-32.3402455	200”
175	282.0473119	-34.2470874	200”
176	281.1037247	-31.3678842	200”
177	280.0065562	-31.0772214	200”
178	280.6453739	-35.9388038	200”
179	279.2216397	-33.3886975	200”
180	278.4625072	-33.4837437	200”
181	277.8756904	-33.1418746	200”
182	277.7866543	-34.0559056	200”
183	277.7754966	-34.1468347	200”
184	277.3098046	-32.1497693	200”
185	281.7661111	-31.9744151	300”
186	280.6064569	-31.5482476	300”
187	280.6378298	-35.635609	300”
188	278.4031974	-33.1346593	300”
189	277.4846029	-31.341551	300”
190	281.9526838	-31.0910562	200”
191	281.7739183	-34.3448571	200”
192	280.9452779	-31.3142148	200”
193	281.5496319	-33.674285	200”
194	280.1481959	-34.0733645	200”
195	279.8157318	-36.1997329	200”
196	279.4543	-34.8750982	200”
197	278.6032102	-33.1880187	200”
198	279.2032829	-36.1257152	200”
199	277.7101567	-33.1928357	200”
200	278.0175541	-36.4017029	200”
201	275.6925178	-33.6119199	200”
202	275.487706	-33.5845587	200”
203	277.4637827	-34.3868499	400”
204	277.0261028	-34.1614615	400”
205	282.4186873	-32.435794	300”

72	279.4657204	-33.6896778	300”
73	278.8306519	-34.8149507	300”
74	278.9504477	-35.3807183	300”
75	278.6181014	-34.4256514	300”
76	278.3992895	-33.7386502	300”
77	278.6336344	-35.6545869	300”
78	278.1348593	-34.2899545	300”
79	276.4829493	-33.4931928	300”
80	276.1494857	-32.3519486	300”
81	282.0401721	-31.9097038	200”
82	281.6590433	-32.2549707	200”
83	281.3309346	-31.6515823	200”
84	281.7893878	-34.0393348	200”
85	281.8346278	-34.585575	200”
86	280.6805763	-31.3709754	200”
87	280.2486168	-31.1934975	200”
88	280.2336225	-31.4171081	200”
89	280.2333685	-31.7827012	200”
90	279.9908848	-30.9079126	200”
91	279.9170847	-30.9718272	200”
92	280.0927663	-31.7394805	200”
93	280.1693184	-32.1341573	200”
94	280.4483404	-33.3361055	200”
95	279.9003464	-31.356861	200”
96	281.007957	-35.7325126	200”
97	279.997474	-31.8411889	200”
98	279.9994548	-31.8493426	200”
99	279.9898528	-31.8510211	200”
100	279.8380676	-31.5121201	200”
101	279.7648412	-31.4144405	200”
102	279.5990588	-31.6808577	200”
103	279.5411703	-31.5209805	200”
104	280.7383651	-36.3410223	200”
105	279.3725624	-30.8880959	200”
106	279.8873262	-34.0374804	200”
107	278.6333103	-29.8872601	200”
108	279.0187676	-32.3222812	200”
109	279.410264	-34.3980671	200”
110	278.9787289	-33.0590334	200”
111	278.7830218	-32.3017887	200”
112	279.2336714	-34.6055107	200”
113	278.8572646	-34.8022125	200”
114	278.489583	-33.8001192	200”
115	278.2143946	-33.3517076	200”
116	278.1769437	-33.2304228	200”
117	277.5877088	-32.0602599	200”
118	277.7435312	-33.0098262	200”
119	277.9850497	-34.2350954	200”
120	278.1655721	-35.0293421	200”
121	277.5596145	-32.5808787	200”
122	277.3945162	-31.996976	200”
123	277.3852744	-32.2352433	200”
124	277.3755873	-32.2367238	200”
125	278.1925679	-35.9568503	200”
126	278.1604643	-36.2240553	200”
127	278.0181278	-36.0170365	200”
128	277.5886609	-34.574752	200”
129	277.4777232	-34.2103868	200”
130	277.0619971	-35.7089551	200”
206	281.2937987	-31.7067532	300”
207	281.2123219	-31.8467772	300”
208	279.632891	-35.639602	300”
209	276.8650544	-33.4734837	300”
210	276.7560165	-34.2597217	300”
211	279.9847737	-31.5827879	200”
212	280.9816049	-35.9170103	200”
213	279.8997187	-32.5298407	200”
214	280.6868582	-35.9871822	200”
215	280.5904178	-35.6999652	200”
216	280.1477838	-33.9843644	200”
217	280.0495088	-33.634255	200”
218	279.0082263	-31.196882	200”
219	278.7520088	-31.2215417	200”
220	279.3256728	-34.5734422	200”
221	278.6406831	-32.2176267	200”
222	278.40536	-33.1857716	200”
223	278.1679826	-33.4160927	200”
224	277.8783546	-34.2864675	200”
225	277.6629362	-34.1684246	200”
226	277.9514971	-36.2007062	200”
227	277.5358519	-35.0001979	200”
228	280.3874652	-34.0795717	400”
229	281.9519225	-31.9754841	200”
230	280.482908	-33.2363459	200”
231	279.5464306	-31.6610195	200”
232	278.3536305	-33.190045	200”
233	278.0328632	-35.1795235	200”
234	278.1579141	-36.7053783	300”
235	281.559865	-31.2634724	200”
236	280.5049593	-32.5342808	200”
237	280.2575174	-32.0437674	200”
238	279.799147	-33.4911229	200”
239	278.3384552	-32.8249731	200”
240	278.0866854	-34.2493748	200”
241	273.8052911	-31.7558679	200”
242	280.253333	-31.7173802	200”
243	275.5963177	-34.5860816	300”
244	279.2790518	-34.3749511	200”
245	278.7222446	-34.7292636	200”
246	277.5571726	-34.2645195	200”
247	277.4438071	-35.213472	200”
248	276.3446687	-35.4564774	200”
249	280.3968594	-31.4486749	200”
250	278.9261059	-32.9782388	200”
251	276.5600737	-33.3170977	200”
252	276.0877753	-33.2479449	200”
253	282.2754363	-35.2335373	200”
254	280.3138966	-32.9028791	200”
255	277.6531935	-36.2014225	200”
256	282.0346435	-31.1576584	600”
257	275.1096427	-35.4539296	200”
258	279.5811071	-31.5577034	200”
259	281.9902409	-33.9004267	200”
260	280.2917319	-30.7769497	200”
261	280.7170871	-35.8600334	200”
262	281.259862	-31.3650578	200”
263	280.821579	-35.916126	200”
264	277.9602849	-36.4147468	200”

131	275.5870512	-33.8876007	200”	265	281.8838767	-31.3007735	200”
132	282.4899754	-32.1993549	500”	266	277.6343972	-34.5306416	200”
133	279.8994967	-32.7277404	500”	267	280.4091962	-33.0836022	200”

Table 13: Bubbles that were detected by the network in the LMC at the [SII] emission line. Note that the Radius is actually the size of the box that was used to query the network. It is not the actual size of the bubble but the bubble is contained within the radius.



```
(base) PS G:\Dokumente\Masterarbeit\Github\DevSoft\src> python .\blobscan.py -h
usage: blobscan [-h] [-merge MERGE] [-save_bubbles SAVE_BUBBLES]
                [-stride_factor STRIDE_FACTOR] [-scaling SCALING]
                [-box_sizes BOX_SIZES]
                [-model_weights_path MODEL_WEIGHTS_PATH]
                fit_path

Evaluate one or more fit files for bubble-like structures.

positional arguments:
  fit_path              The path to the fit file that should be evaluated.
                        E.g. folder_xy/fits_file.fits

optional arguments:
  -h, --help            show this help message and exit
  -merge MERGE          Defines if the different box_sizes should be merged or
                        if all box_sizes are saved individually. The algortihm
                        feeds cutouts from the fit file to the neural network.
                        The box size defines the area of these cutouts.
                        Default is True
  -save_bubbles SAVE_BUBBLES
                        Whether the found bubbles should be saved as image.
                        They are saved into the directory of the fits file in
                        a new folder with the fits name. "all" saves all found
                        bubbles (also multiple), "merged" only saves
                        individual bubbles, "none" does not save any at all.
                        Default is "none".
  -stride_factor STRIDE_FACTOR
                        Defines the striding that is used to tile the complete
                        fits file. The striding is 1/2 of the box_size times
                        the stride_factor: striding = 1/2 * box_sizes *
                        stride_factor . Default is 0.5
  -scaling SCALING      If the subimages are rescaled individually or if the
                        complete images is rescaled once.'individual' for
                        individual scaling, 'global' for global scaling
                        Default is 'individual'.
  -box_sizes BOX_SIZES  Defines the box_size that is cut out from the fits
                        file in order to query the neural network. The
                        box_sizes have to be pixel values. The default is
                        100,150,200,250,300,350,400,500,600,700,800.
  -model_weights_path MODEL_WEIGHTS_PATH
                        Individual path to the model_weights. Not necessary if
                        the weights are inside the 'model' folder of the
                        script.
```

```
(base) PS G:\Dokumente\Masterarbeit\Github\DevSoft\src> python .\blobscan.py "G:/Dokumente/M
asterarbeit/observations/LMC/MCELS2/LMC_ha_fac_03_medium/LMC.ha.csub.trim.fits" -merge True
-save_bubbles "all" -stride_factor 0.3 -scaling "individual" -box_sizes 1000,1100,1200,1300,
1400,1500 -model_weights_path "G:/Dokumente/Masterarbeit/Github/DevSoft/src/model/model"
Using TensorFlow backend.
2020-07-27 18:26:29.099948: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU su
pports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
WARNING: FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to 51146.000000 from DATE-O
BS.
Changed DATE-OBS from '29/11/1998' to '1998-11-29''. [astropy.wcs.wcs]
Could not save bubbles since the folder for the bubbles already exists:
G:/Dokumente/Masterarbeit/observations/LMC/MCELS2/LMC_ha_fac_03_medium/LMC.ha.csub.trim.
Continue anyway? y/ny
 Predicting 1400 of 10816 tiles...
```

```
LMC.ha.csub.trim
LMC.ha.csub.trim._(100, 100)_stride_fac_0_3
LMC.ha.csub.trim._(150, 150)_stride_fac_0_3
LMC.ha.csub.trim._(200, 200)_stride_fac_0_3
LMC.ha.csub.trim._(250, 250)_stride_fac_0_3
LMC.ha.csub.trim._(300, 300)_stride_fac_0_3
LMC.ha.csub.trim._(350, 350)_stride_fac_0_3
LMC.ha.csub.trim._(400, 400)_stride_fac_0_3
LMC.ha.csub.trim._(500, 500)_stride_fac_0_3
LMC.ha.csub.trim._(600, 600)_stride_fac_0_3
LMC.ha.csub.trim._(700, 700)_stride_fac_0_3
LMC.ha.csub.trim._(800, 800)_stride_fac_0_3
LMC.ha.csub.trim._stride_fac_0_3_merged
LMC.ha.csub.trim
```

Folder for saved blobs.
Created if –save_bubbles is „all“ or „merged“

Region files for all selected box_sizes

Merged region file containing all box_sizes.
Created if –merge is True

Figure 29: Complete process of applying the *Blobscan* model to a fits file. On the left side the help output from the command line interface is depicted. On the right side an exemplary application is depicted. The red marked area is a mandatory input, orange marked areas are optional inputs and the green marked area is the expected outcome. The script counts all tiles that the network needs to predict and how many tiles were already predicted. In the lower right side the outcome in the folder of the analysed fits file is depicted.

A.2.4 Instructions

The *Blobscan* algorithm can be downloaded from URL https://www.sternwarte.uni-erlangen.de/gitlab/ramsteck/blobscan. It can be used with the command line interface. Therefore, navigate into the folder where the *Blobscan* program was saved and execute it with *python blobscan.py -h* . In Figure 29 the complete process is depicted. In Table 14 all arguments that can be given to the command line interface are listed.

Argument	Importance	Description
fit_path	mandatory	The complete path to the fits file that should be analysed. E.g. $C : /Dokumente/LMC\_ha.fits$
-merge	optional	True or False. Default is True. If it is True, all individual box_sizes are merged as described in Section 5.1.2 and saved as an additional region file.
-save_bubbles	optional	"all", "merged" or "none". Default is "none". For "all", all boxes that were classified as bubbles are saved as an image. A folder for every box_size is created. The images are enumerated in the same order as the region file. img_0 is the first line of the associated region file.
-stride_factor	optional	]0,1]. Default is 0.5. In general the striding, that defines the increment of the box walking over the fits file, is 1/2 of the box_size. The stride_factor is multiplied to that: $1/2 \cdot box\_size \cdot stride\_factor$. A small value essentially means that objects in the fits file are analysed by the network in many different positional variations, improving the probability for it to be detected. It increases the sensibility (Section 5.1.1).
-scaling	optional	"individual" or "global". Default is "individual". In order to analyse a cutout of the fits file the pixel values of the cutout have to be values between 0 and 255. Therefore, either the pixel values are scaled with the global maximum value of the fits file or with the individual maximum value of the cutout area.
-box_sizes	optional	Integer values separated by a comma. Default is 100,150,200,250,300,350,400,500,600,700,800. It defines the pixel size of the quadratic cutouts from the fits file that are analysed by the neural network. Too small values leads to the problem that also stars are predicted as bubble-like structures.
-model _weights_path	optional	Default is the folder model in the folder where the script is located. The complete path to the neural network model weights.

Table 14: All possible arguments that can be given to the command line interface of the *Blobscan* program.

B References

Baraldi, L. (2016), ‘Vgg16 pretrained’, https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3. Last accessed 08 Juli 2020.

Bardon/ESO, Z. (2017).
URL: https://www.eso.org/public/images/magellan-ch17-bardon-cc/

Besag, J. (1977), ‘Discussion on dr ripley’s paper’, *Journal of the Royal Statistical Society: Series B (Methodological)* **39(2)**, 193–195.

Bica, E., Bonatto, C., Dutra, C. M. & Santos, J. F. C. (2008), ‘A general catalogue of extended objects in the magellanic system’, *Monthly Notices of the Royal Astronomical Society* **389**(2), 678–690.
URL: http://dx.doi.org/10.1111/j.1365-2966.2008.13612.x

Bonanos, A. Z., Massa, D. L., Sewilo, M., Lennon, D. J., Panagia, N., Smith, L. J., Meixner, M., Babler, B. L., Bracker, S., Meade, M. R., Gordon, K. D., Hora, J. L., Indebetouw, R. & Whitney, B. A. (2009), ‘SPITZERSAGE INFRARED PHOTOMETRY OF MASSIVE STARS IN THE LARGE MAGELLANIC CLOUD’, *The Astronomical Journal* **138**(4), 1003–1021.
URL: https://doi.org/10.1088%2F0004-6256%2F138%2F4%2F1003

Bradley, L., Sipőcz, B., Robitaille, T., Tollerud, E., Vinícius, Z., Deil, C., Barbary, K., Günther, H. M., Cara, M., Busko, I., Conseil, S., Droettboom, M., Bostroem, A., Bray, E. M., Bratholm, L. A., Wilson, T., Craig, M., Barentsen, G., Pascual, S., Donath, A., Greco, J., Perren, G., Lim, P. L. & Kerzendorf, W. (2019), ‘astropy/photutils: v0.6’.
URL: https://doi.org/10.5281/zenodo.2533376

Chollet, F. et al. (2015), ‘Keras’, https://keras.io.

Chu, Y.-H. (2008), Bubbles and Superbubbles: Observations and Theory, *in* F. Bresolin, P. A. Crowther & J. Puls, eds, ‘Massive Stars as Cosmic Engines’, Vol. 250 of *IAU Symposium*, pp. 341–354.

Clarke, D., Bridle, A., Burns, J., Perley, R. & Norman, M. (1991), ‘Origin of the structures and polarization in the classical double 3c 219’, *The Astrophysical Journal* **385**, 173–187.

Collischon, C. (2020), ‘Analysis of structures in the magellanic clouds with minkowski tensors’.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009), ImageNet: A Large-Scale Hierarchical Image Database, *in* ‘CVPR09’.

Fanaroff, B. L. & Riley, J. M. (1974), ‘The Morphology of Extragalactic Radio Sources of High and Low Luminosity’, *Monthly Notices of the Royal Astronomical Society* **167**(1), 31P–36P.
URL: https://doi.org/10.1093/mnras/167.1.31P

Fukushima, K. (1980), 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position', *Biological Cybernetics* **36**(4), 193–202.
URL: https://doi.org/10.1007%2Fbf00344251

Gaustad, J., McCullough, P., Rosing, W. & Van Buren, D. (2001), 'A robotic wide-angle hα survey of the southern sky', *Publications of the Astronomical Society of the Pacific* **113**(789), 1326–1348.
URL: http://www.jstor.org/stable/10.1086/323969

Hancock, P. J., Murphy, T., Gaensler, B. M., Hopkins, A. & Curran, J. R. (2012), 'Compact continuum source-finding for next generation radio surveys'.

Harris, J. & Zaritsky, D. (2009), 'The star formation history of the large magellanic cloud'.

Hinton, G. (n.d.), 'Overview of mini-batch gradient descent'.
URL: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Laing, R. A., Bridle, A. H., Parma, P., Feretti, L., Giovannini, G., Murgia, M. & Perley, R. A. (2008), 'Multifrequency VLA observations of the FR I radio galaxy 3C 31: morphology, spectrum and magnetic field', *Monthly Notices of the Royal Astronomical Society* **386**(2), 657–672.
URL: https://doi.org/10.1111/j.1365-2966.2008.13091.x

Markowsky, G. (2017), 'Information theory'. Last accessed 08 Juli 2020.
URL: https://www.britannica.com/science/information-theory

NASA & ESA (2019), 'Usage statistics of content languages for websites'. Last accessed 08 Juli 2020.
URL: https://hubblesite.org/contents/news-releases/2019/news-2019-12.html

Pietrzyński, G., Graczyk, D., Gallenne, A., Gieren, W., Thompson, I., Pilecki, B., Karczmarek, P., Gorski, M., Suchomska, K., Taormina, M., Zgirski, B., Wielgórski, P., Kołaczkowski, Z., Konorski, P., Villanova, S., Nardetto, N., Kervella, P., Bresolin, F., Kudritzki, R. & Narloch, W. (2019), 'A distance to the large magellanic cloud that is precise to one per cent', *Nature* **567**, 200–203.

Ripley, B. D. (1976), 'The second-order analysis of stationary point processes', *Journal of Applied Probability* **13**(2), 255–266.

Ripley, B. D. (1981), *Spatial Statistics*, John Wiley & Sons, Inc.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition'.

Smith, R., Leiton, R. & Pizarro, S. (2000), 'The um/ctio magellanic cloud emission line survey (mcels)'.

Weaver, R., McCray, R., Castor, J., Shapiro, P. & Moore, R. (1977), 'Interstellar bubbles. ii. structure and evolution.', **218**, 377–395.

Wenger, M., Ochsenbein, F., Egret, D., Dubois, P., Bonnarel, F., Borde, S., Genova, F., Jasniewicz, G., Laloe, S., Lesteven, S. & Monier, R. (2000), 'The simbad astronomical database'.

Westerlund, B. E. (1997), *The Magellanic Clouds.*

Zhang, Y. & Zhao, Y. (2015), 'Astronomy in the big data era', *Data Science Journal* **14**, 11.

Acknowledgement

I want to thank Prof. Dr. Manami Sasaki for supervising this master thesis. I always got help when I needed it and I want to thank her for the possibility to visit Australia as a part of this study. I am very grateful for this experience.

Figure 9
Gaia Data Processing and Analysis Consortium (DPAC); A. Moitinho / A. F. Silva / M. Barros / C. Barata, University of Lisbon, Portugal; H. Savietto, Fork Research, Portugal.

Section 3.1
The Australian SKA Pathfinder is part of the Australia Telescope National Facility which is managed by CSIRO. Operation of ASKAP is funded by the Australian Government with support from the National Collaborative Research Infrastructure Strategy. ASKAP uses the resources of the Pawsey Supercomputing Centre. Establishment of ASKAP, the Murchison Radio-astronomy Observatory and the Pawsey Supercomputing Centre are initiatives of the Australian Government, with support from the Government of Western Australia and the Science and Industry Endowment Fund. We acknowledge the Wajarri Yamatji people as the traditional owners of the Observatory site.