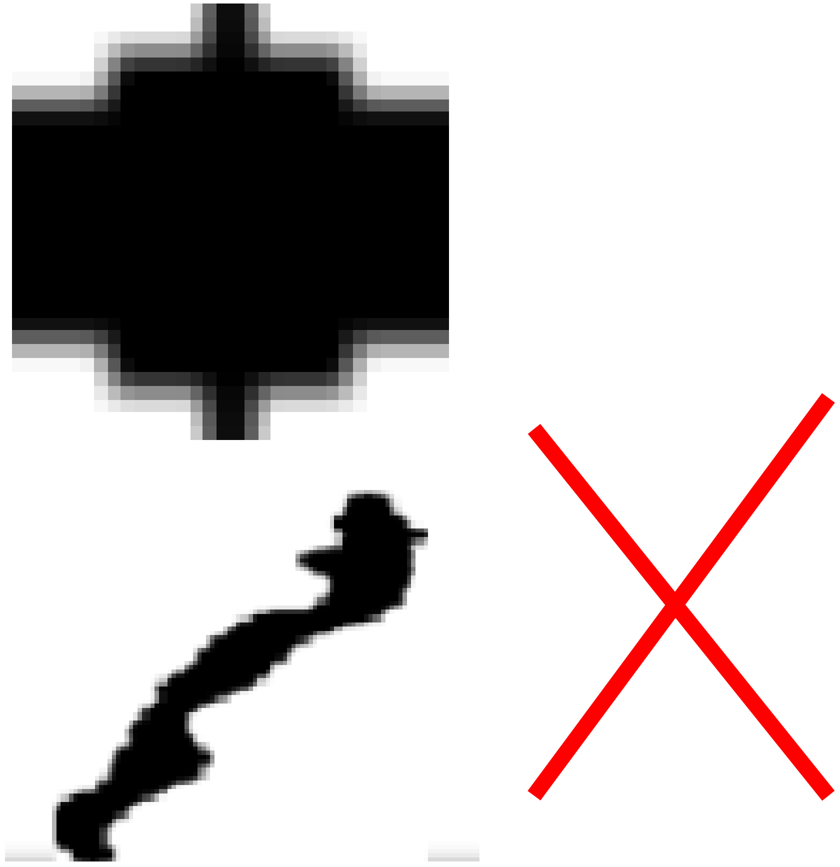
Classification of impurities in beer

Nicolas R.H. Pedersen

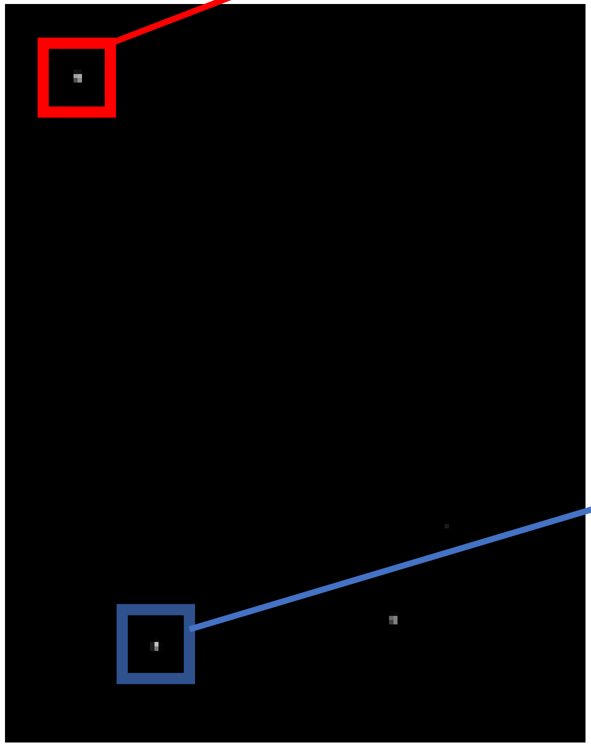


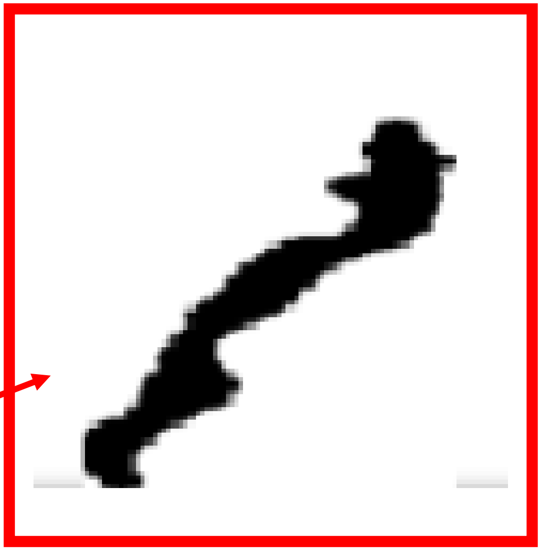
Outline

- Introduction to the data – and the problem that is solved
- Preparing the data – Labelling and preprocessing
- A naive and simple solution – applying the MNIST solution
- Building a CNN for the problem
- Analyzing results
- Conclusion

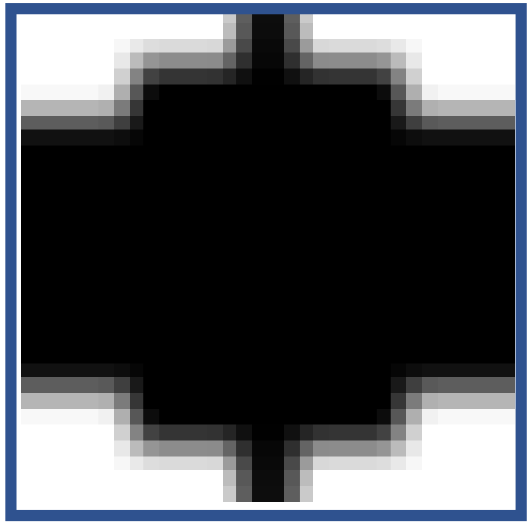
Impurities

- Carlsberg need a good way to find impurities in their beer (Carls talk)
- Impurities are in the form of either a particle or a string
- ~ 172. 000 images – 1 particle or 1 string in each
- Only black/white – but the images have different sizes
- No labeling of the images!





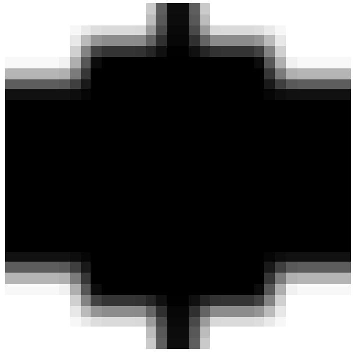
String



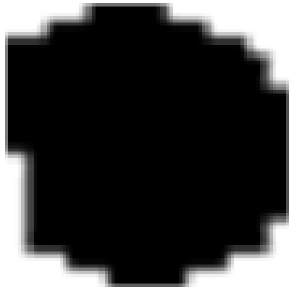
Particle

Labelling the data

- No initial labels are provided – a consistent and well functioning method is needed
- Particles look roughly the same – they only grow in size
- Strings vary a lot!
- An good labelling method can be to find all the particles and label the rest as strings
- Images have different initial sizes – resize to standard size



Particle



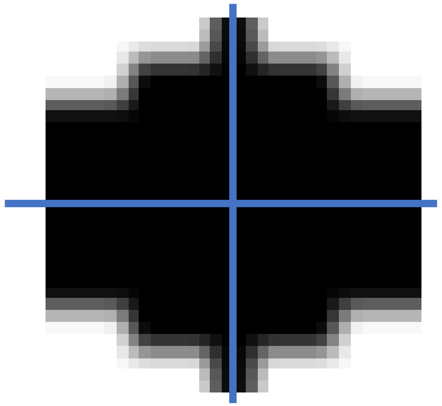
Particle that has grown

String

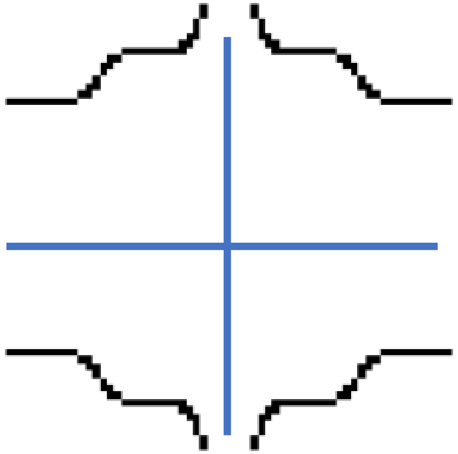
Another string

Labelling with symmetry

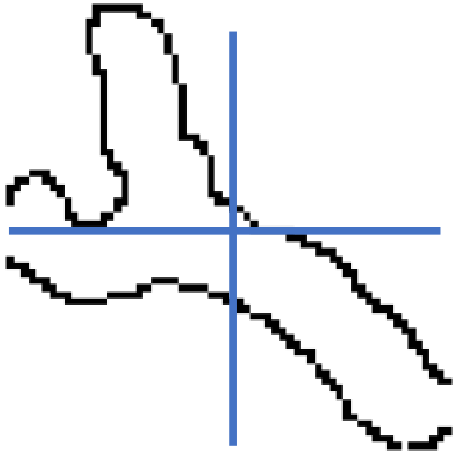
- Particles are quite symmetric – more than strings at least
- Find edges in every image – find the symmetry of these edges
- The perfect symmetry of many particles is due to the fact that they are upscaled versions of a simple pixel
- Perfectly symmetric particles are simply deemed particles and removed from further analysis



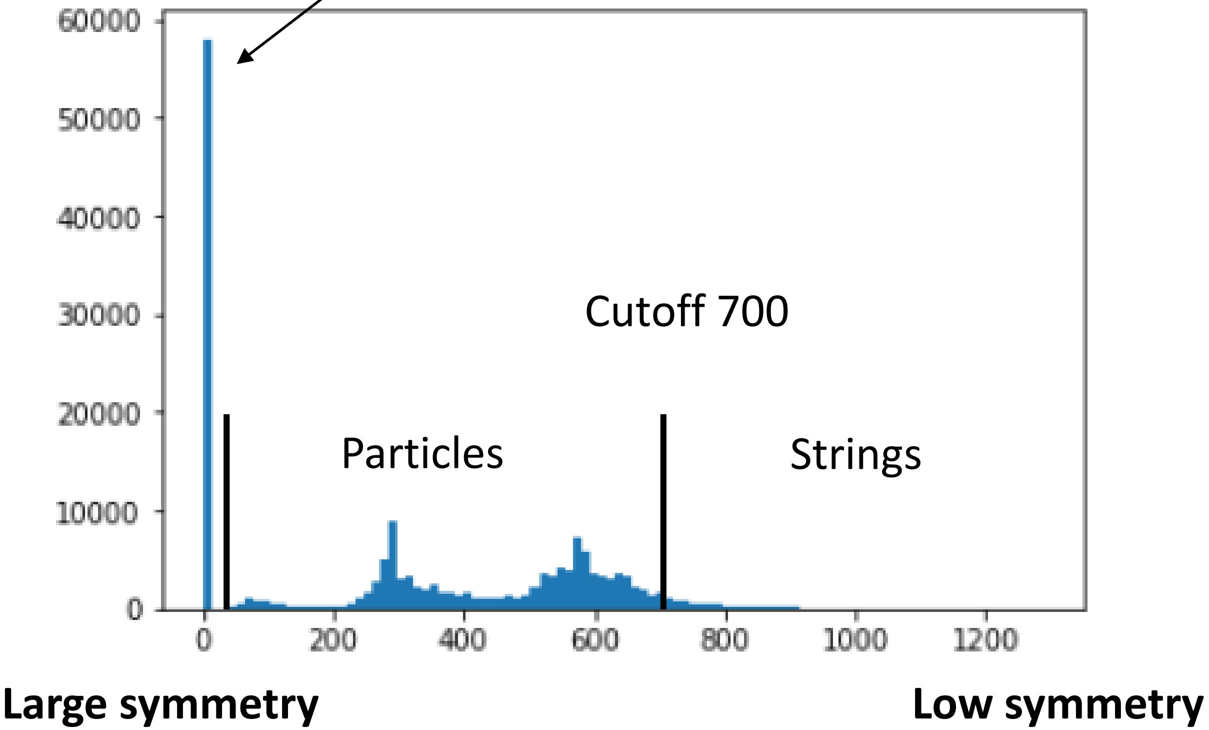
Perfect symmetric particle – “Eigenparticle”



Edges of particle
- Symmetric

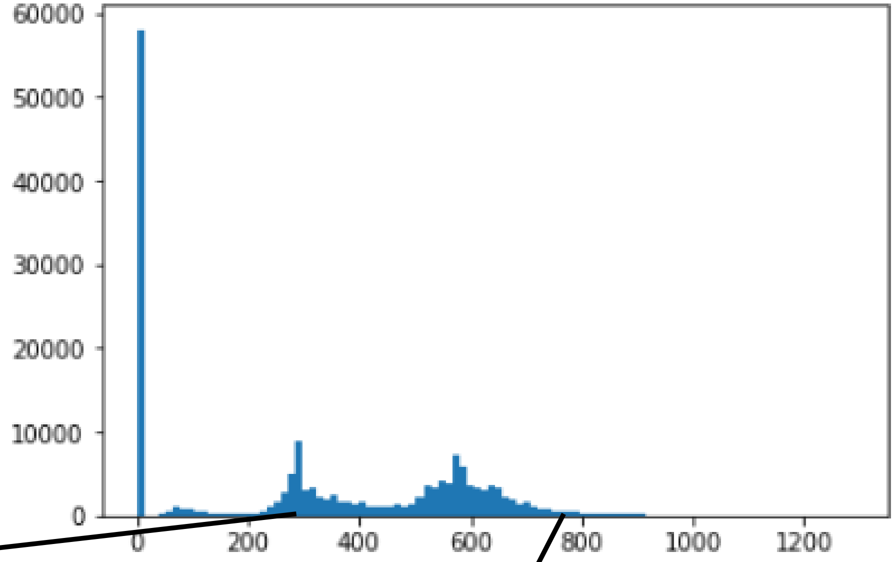


Edges of string
Non - Symmetric

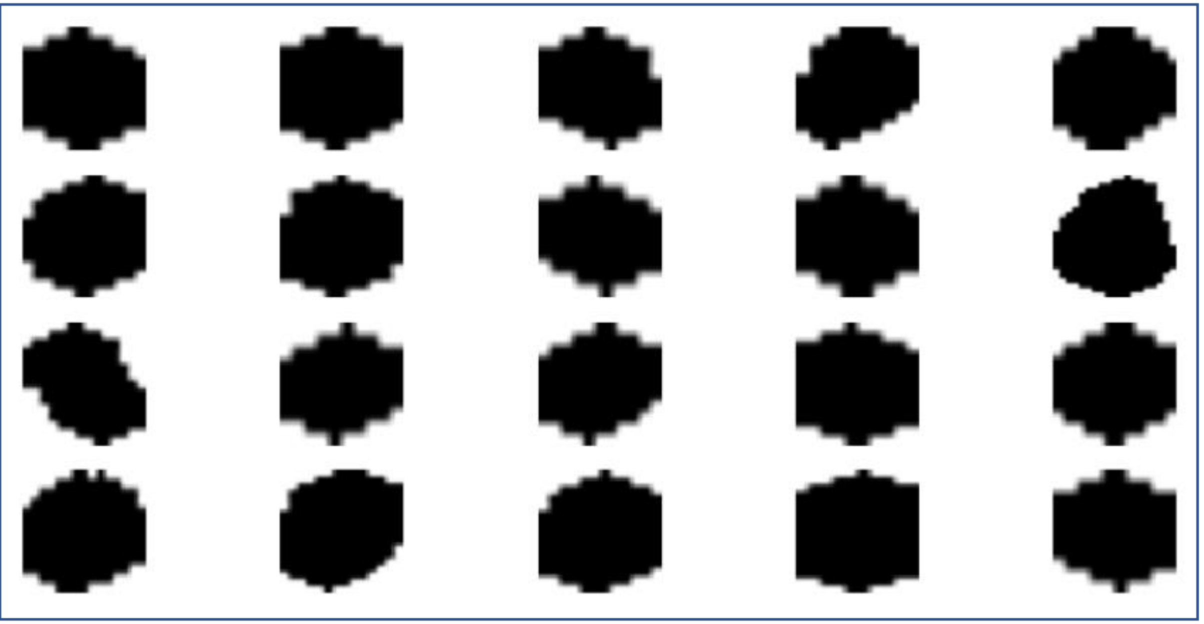


Sanity check of labelling

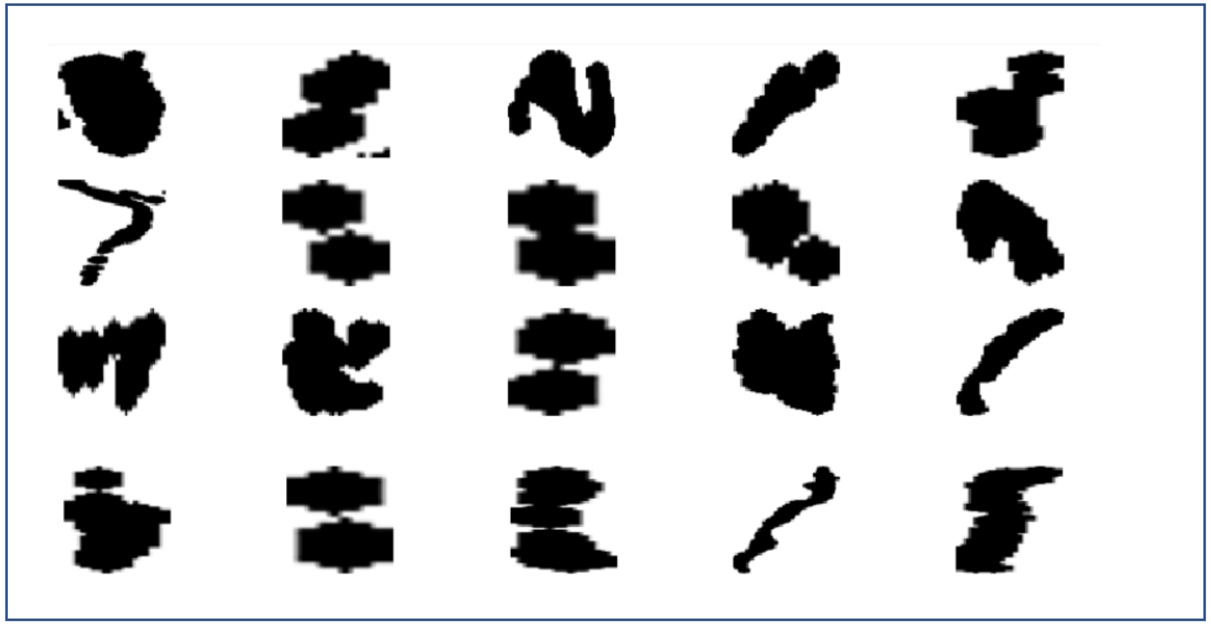
- 110. 000 particles ~ 93.5% (with “eigenparticles” removed)
- Particles are everything below 700 on the symmetry plot
- Some strings look a bit like multiple particles



Particles - large symmetry

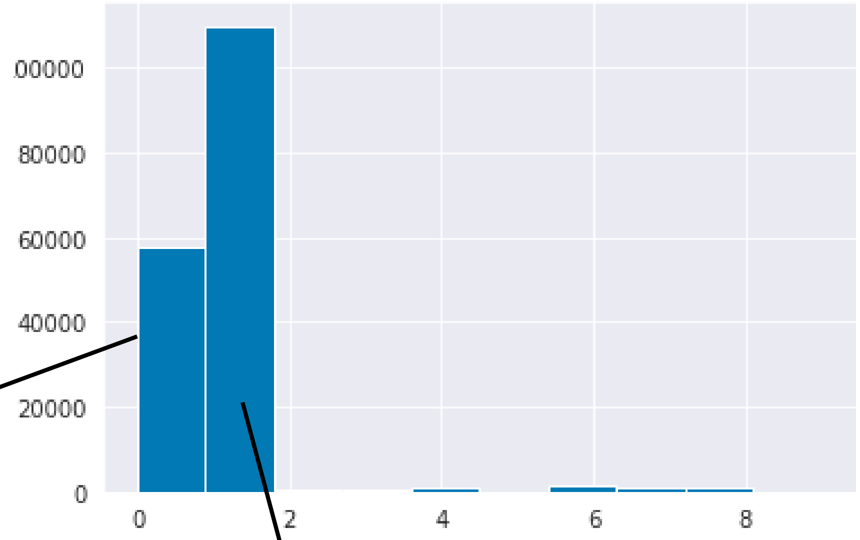


String – low symmetry

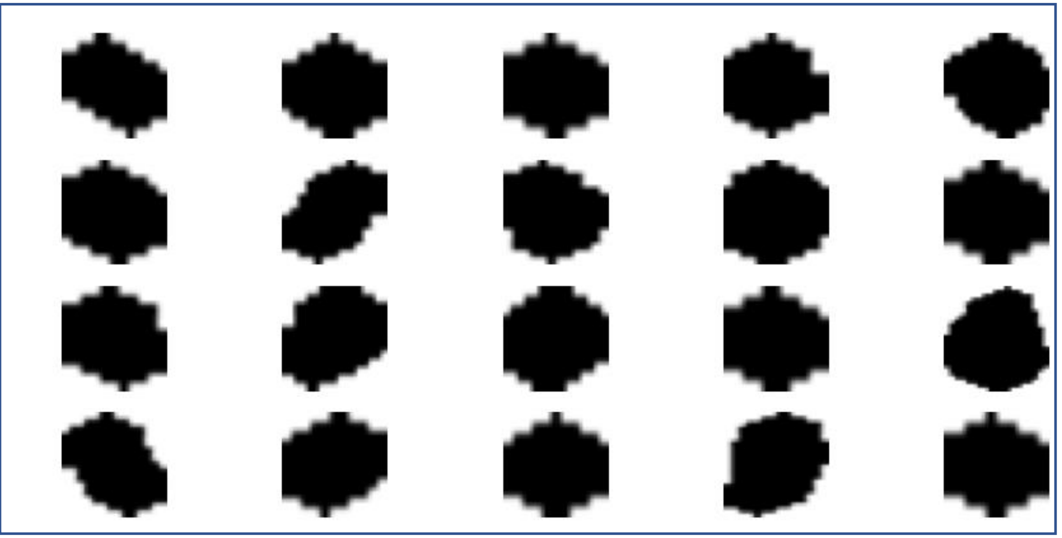


A naïve and lazy solution

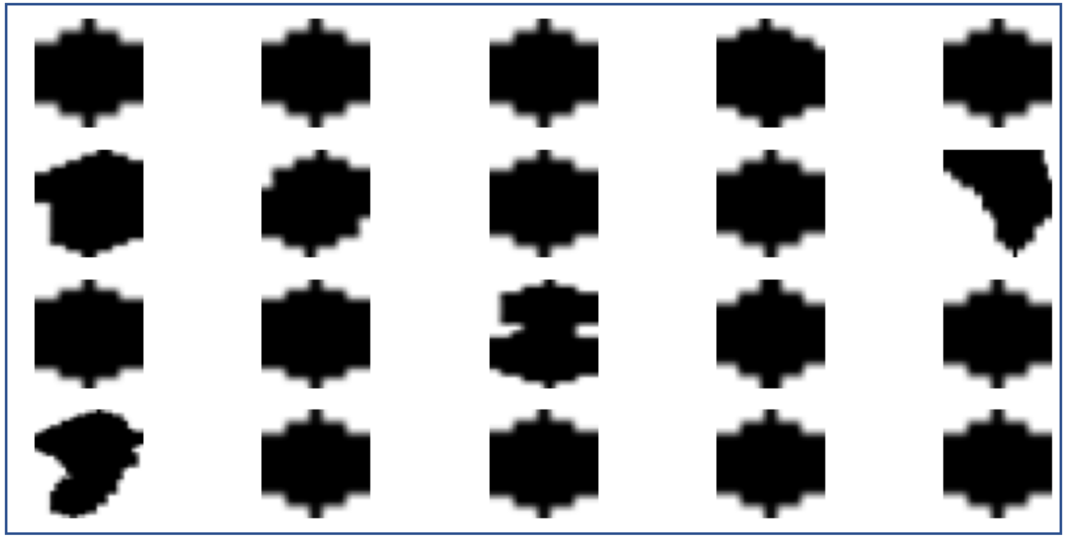
- Applying the MNIST solution (handwritten numbers)
- Particles might look like 0's and strings might look like other numbers
- Is not good at classifying this problem



Predicted as 0's



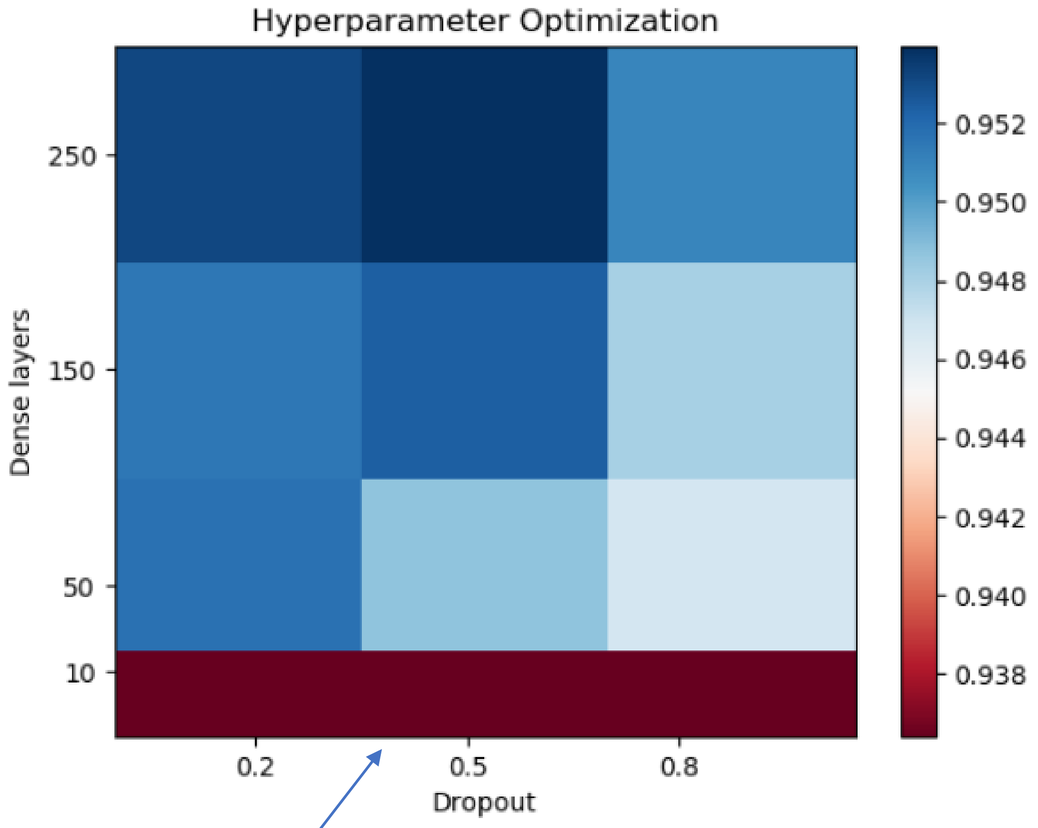
Predicted as 1's



Making a CNN using the symmetry labelling

- Keras is used to build the CNN
- 40 Epochs used
- Data is split in 70% training and 30% testing
- It takes ~ 1 hours to run on Colabs GPU
- Should it be more complex? Probably not – the images themselves are not overly complex

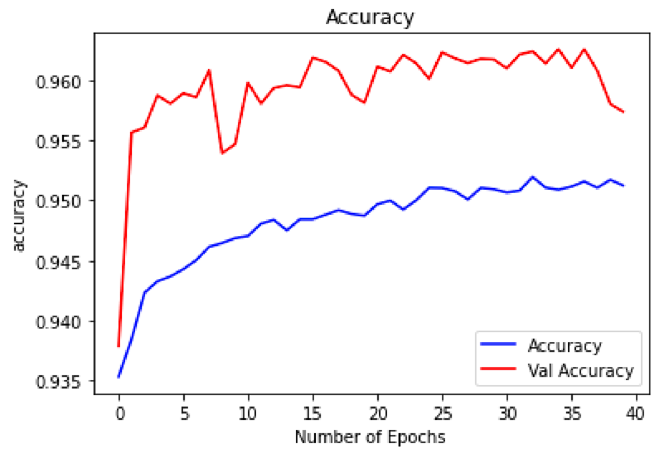
```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 64, 64, 8)         208
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 32, 8)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 32, 32, 8)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 32, 32, 8)         1608
_________________________________________________________________
flatten_2 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_3 (Dense)              (None, 150)               1228950
_________________________________________________________________
dropout_4 (Dropout)          (None, 150)               0
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 302
=================================================================
Total params: 1,231,068
Trainable params: 1,231,068
Non-trainable params: 0
_________________________________________________________________
```

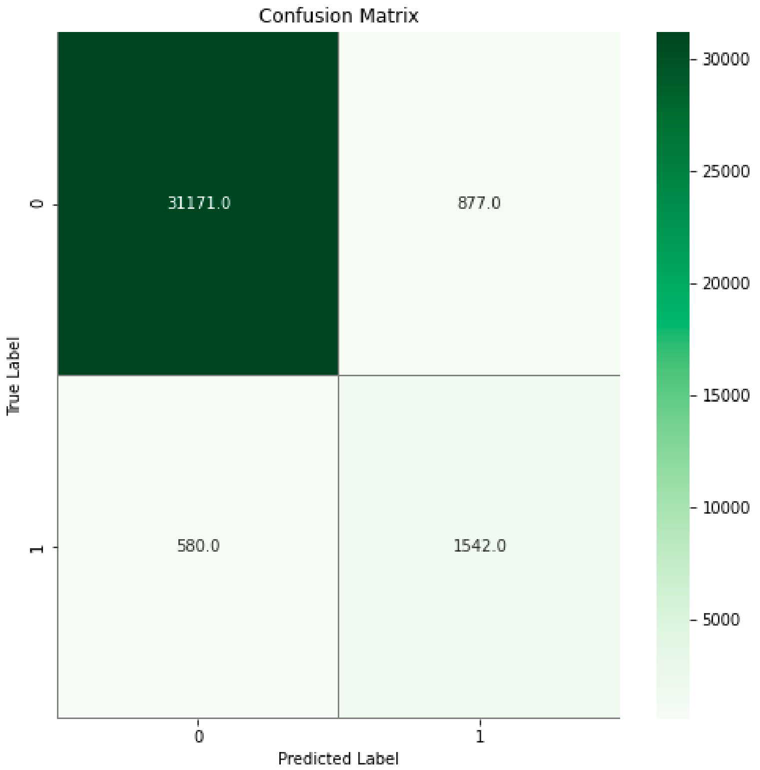


Here the model is not learning anything – it simply guesses everything as a particle.

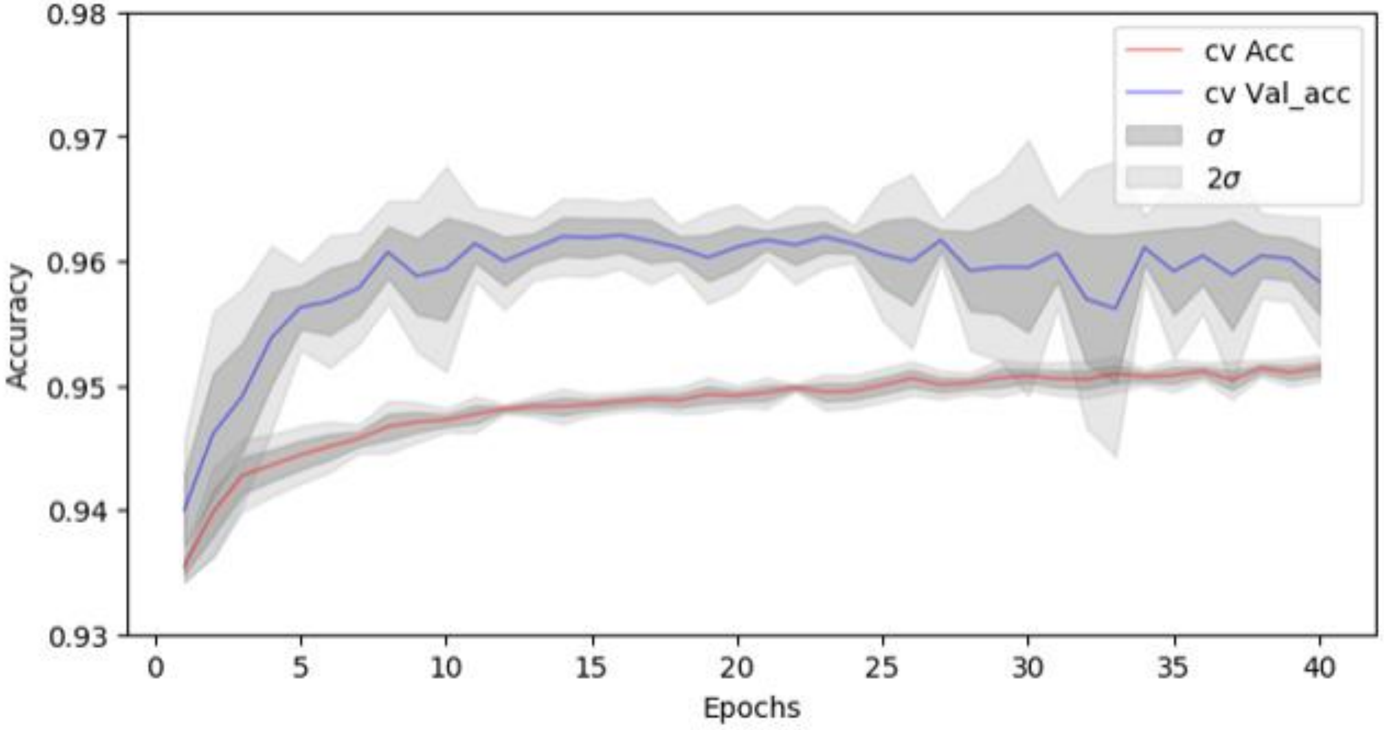
Results of the CNN

- Accuracy is better than just guessing from the initial distribution (93.5% particles)
- Seem to have somewhat converged with the number of epochs used
- CNN is not extremely complicated – It does however look as though little is to be gained!



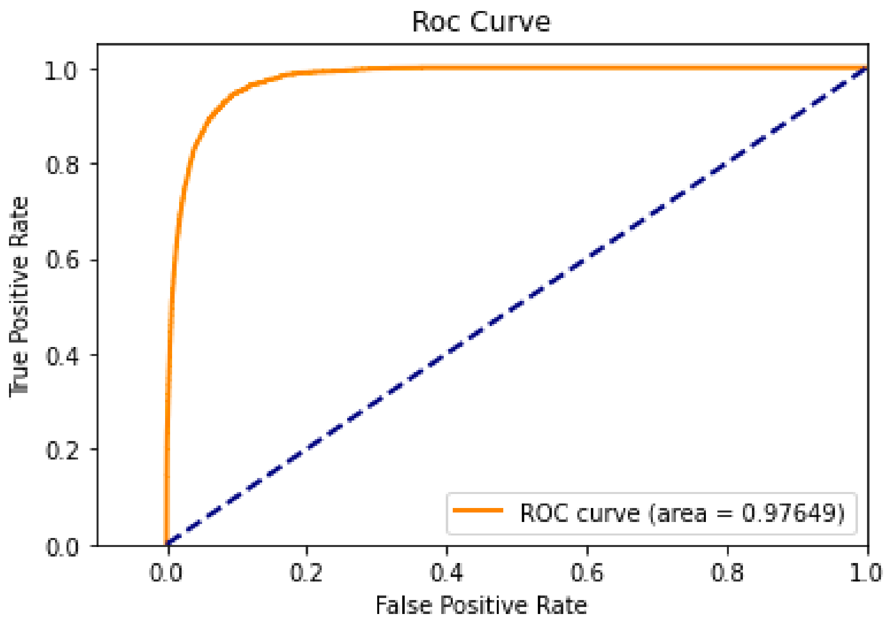


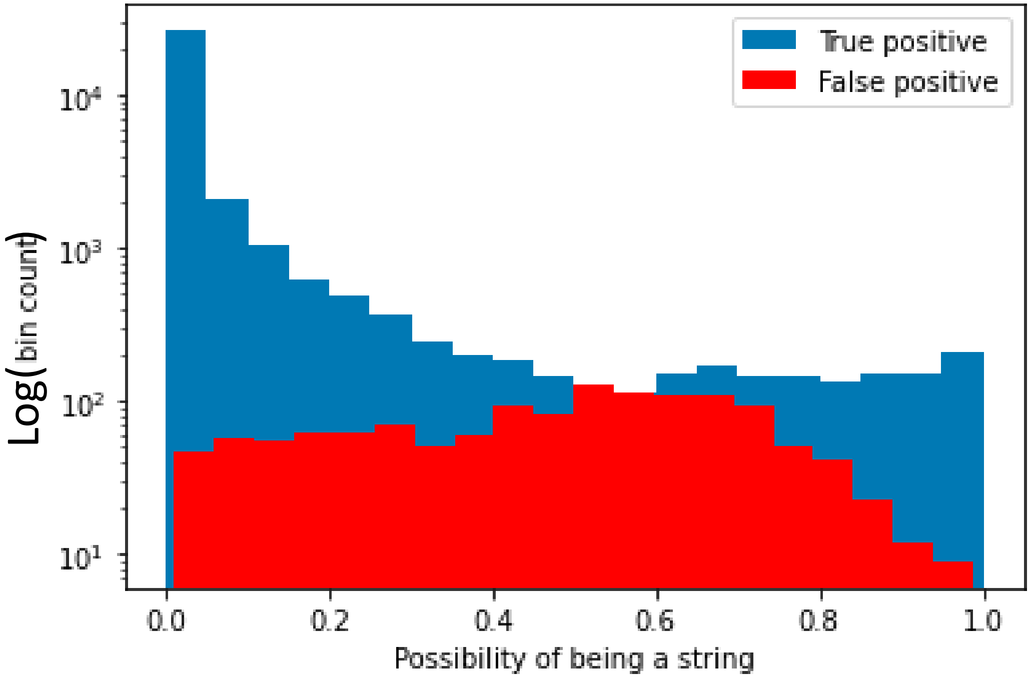
Cross validation



Roc Curve and statistics of results

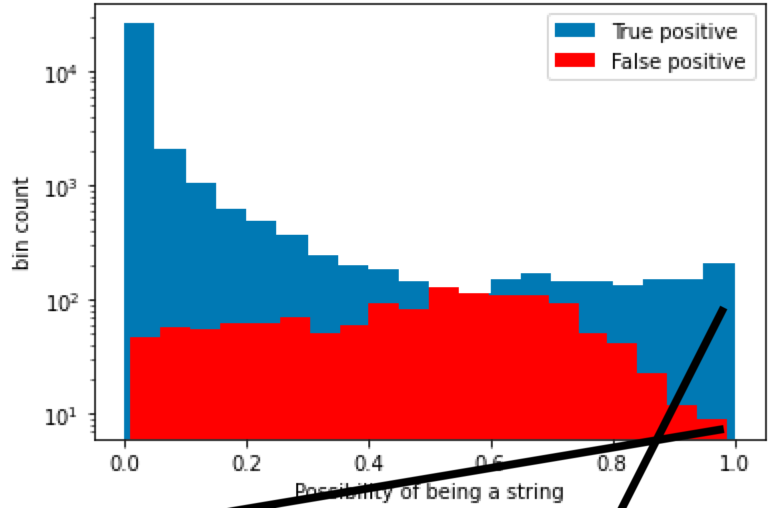
- Very nice roc curve – with a roc curve area = 0.97649
- The distribution of probabilities show that the algorithm is very certain in most of the cases (logarithmic)
- The false positives mainly occur when the algorithm is less certain of the result
- Everything looks good so far – but can it be trusted?





Can the results be trusted?

- No initial labelling was provided – need to investigate the results to verify
- Is the classification correct?
- Where and why does it fail?
- It looks like its good at finding strings that are labelled as particles!



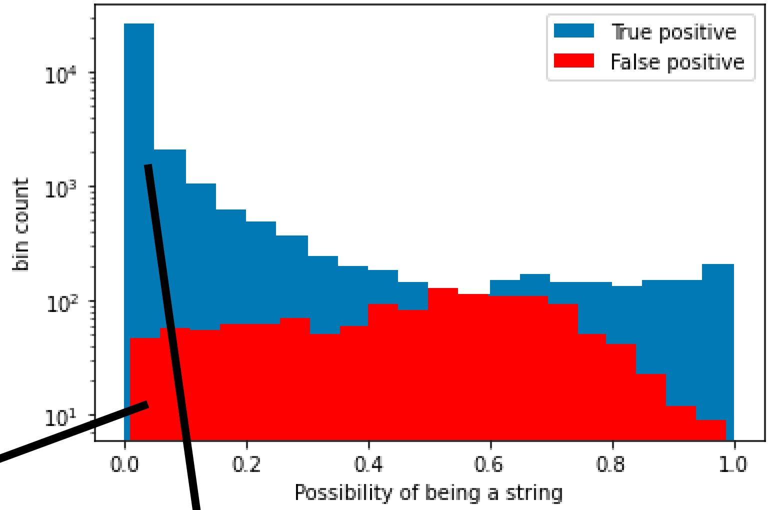
False positive strings – with a large probability of being a string

True positive strings

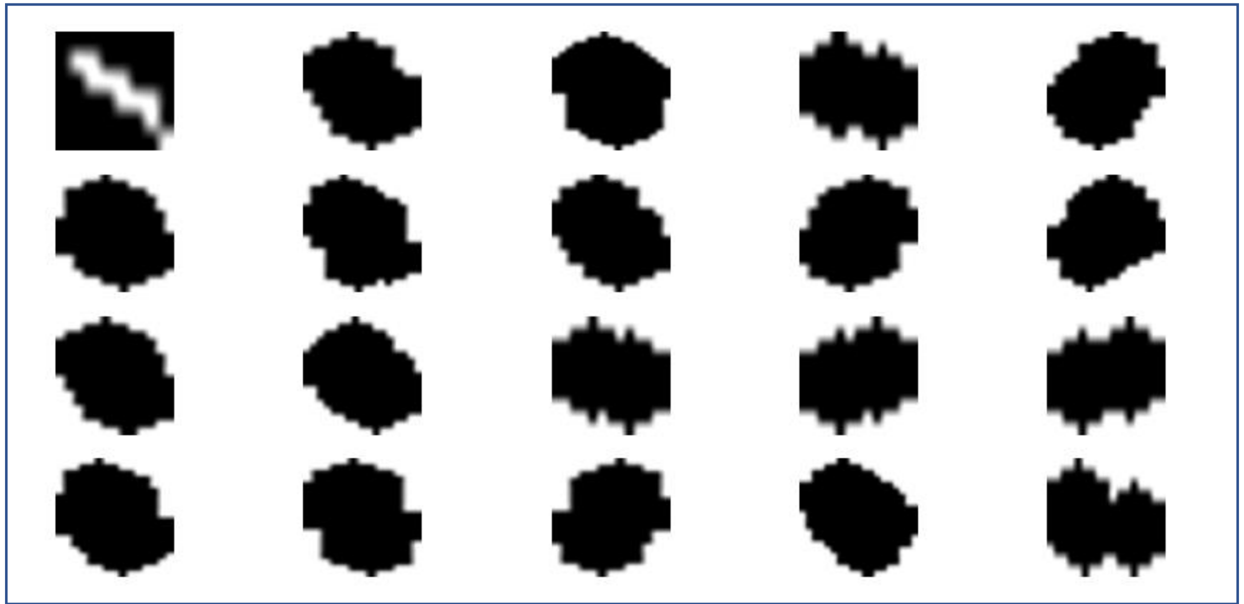
Can the results be trusted?

- It looks like the results are actually trustworthy! The CNN does even seem to be good at finding strings that were labelled as particles

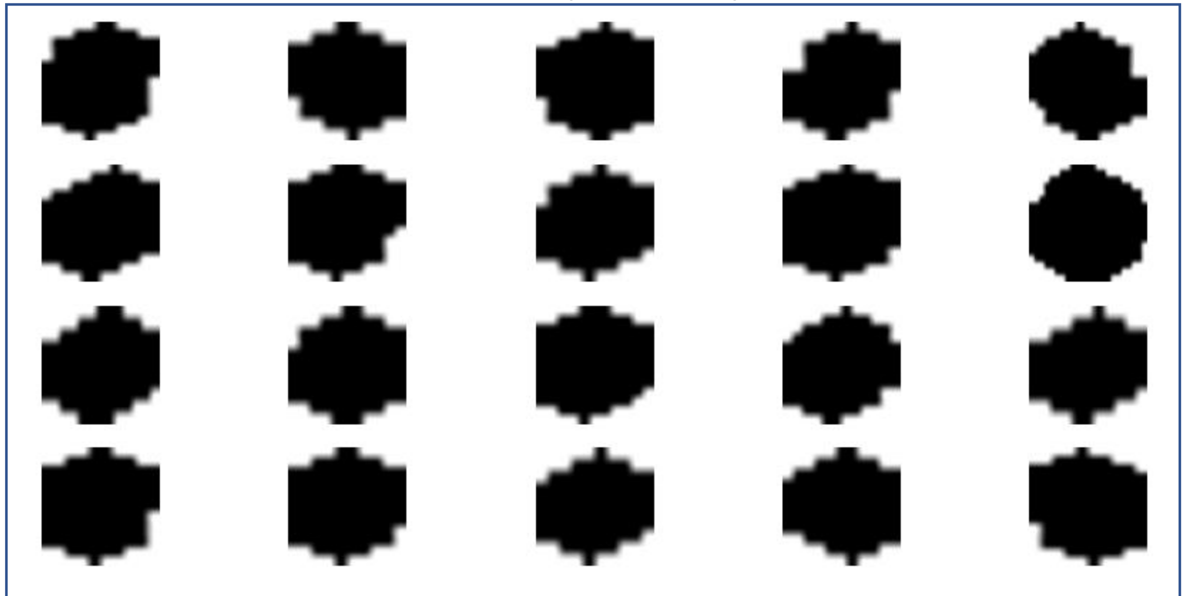
- The particles look like particles and the strings look like strings



False positive particles – with a large probability of being a particle

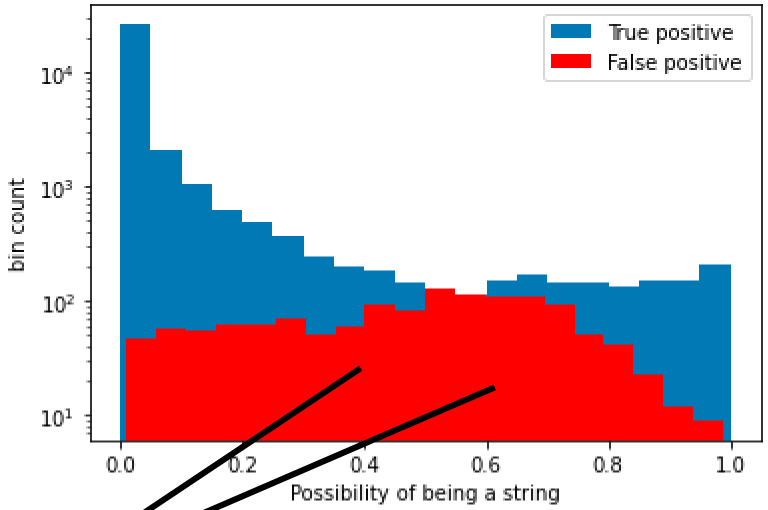


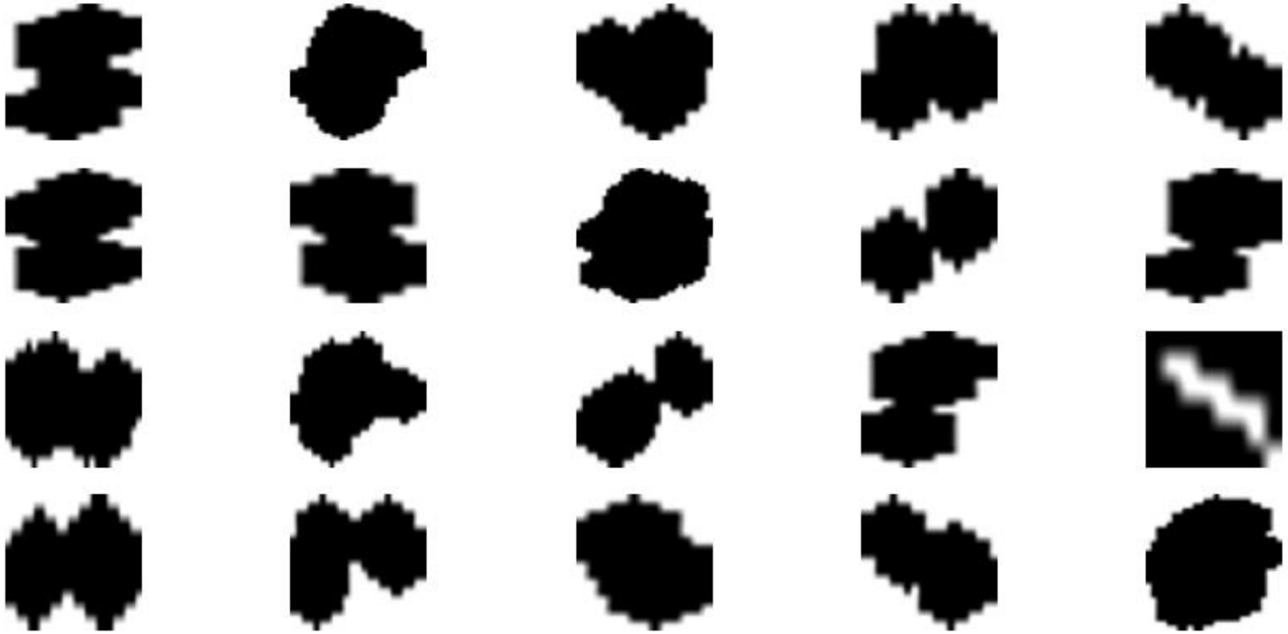
True positive particles



When is it uncertain?

- In the cases where the algorithm is less certain we find a couple of weird results
- Some look like 2 particles very close to each other – it makes sense that the CNN struggles here!

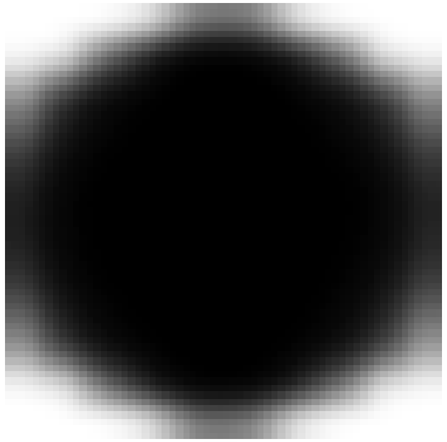




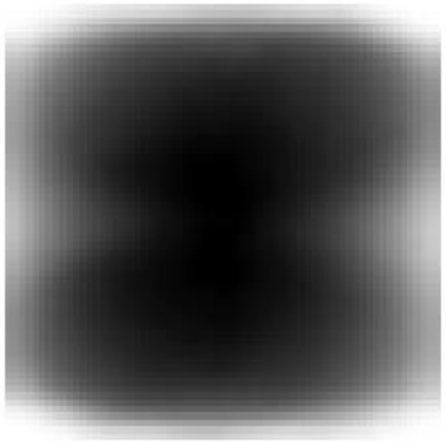
Mistake in the image

Improvement and outlook

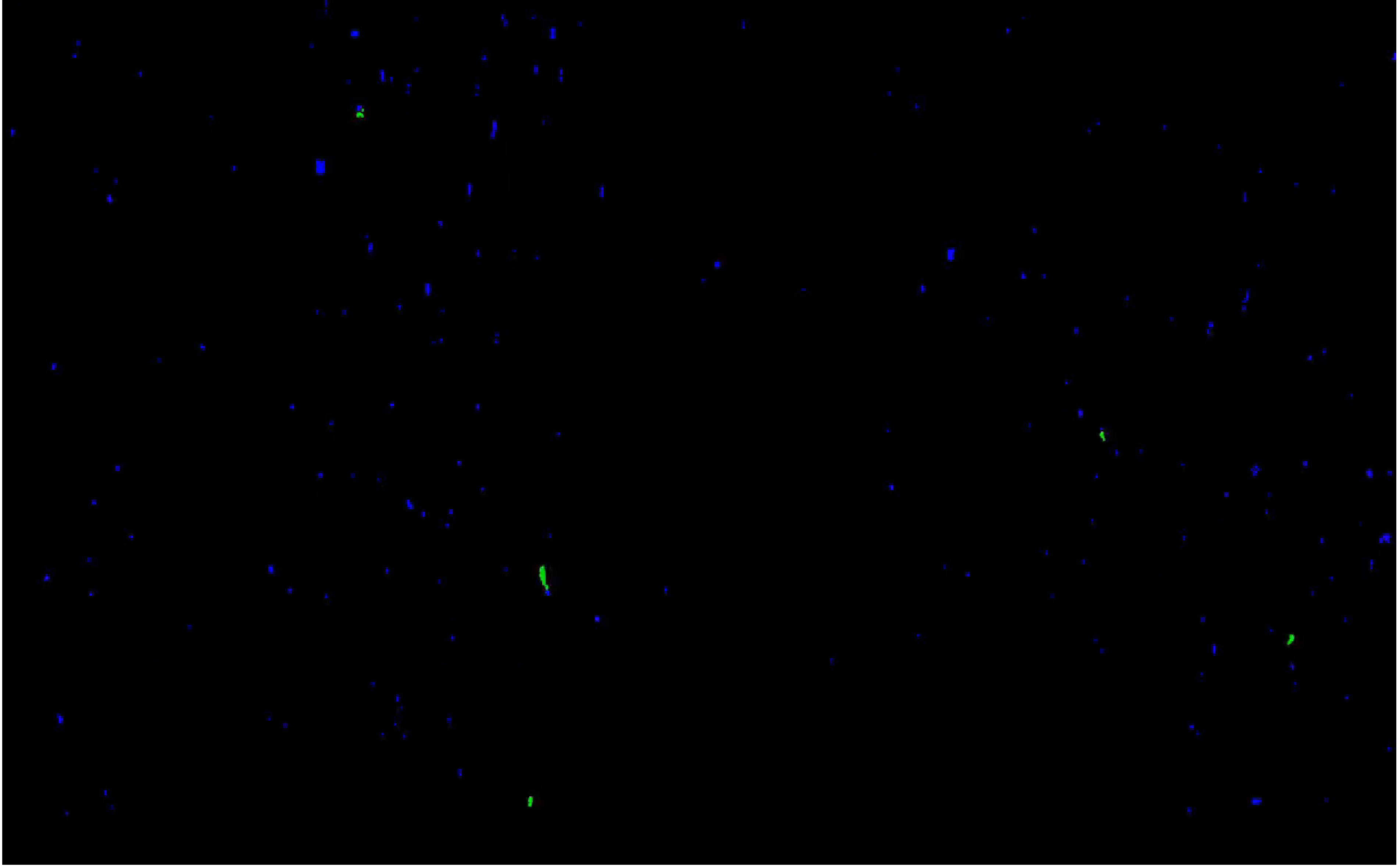
- It seems that the CNN is better at locating strings than the initial labelling with symmetry – It actually makes sense to do the CNN
- Could be an idea to find a way of sorting out “double particles” or even classify them as 2 particles
- Should be fairly easy to implement on top of Carls work – in fact this is already done!



The average of all particles



The average of all strings



Some ”flickering”

Able to always find the obvious string!

Conclusion

- Was able to classify the strings and particles with a large accuracy – even able to find strings that were initially labelled as particles

- Relative short training time for the CNN

- The problem is mainly to initially make a good labelling from which the CNN is able to learn and even make improvements in the results