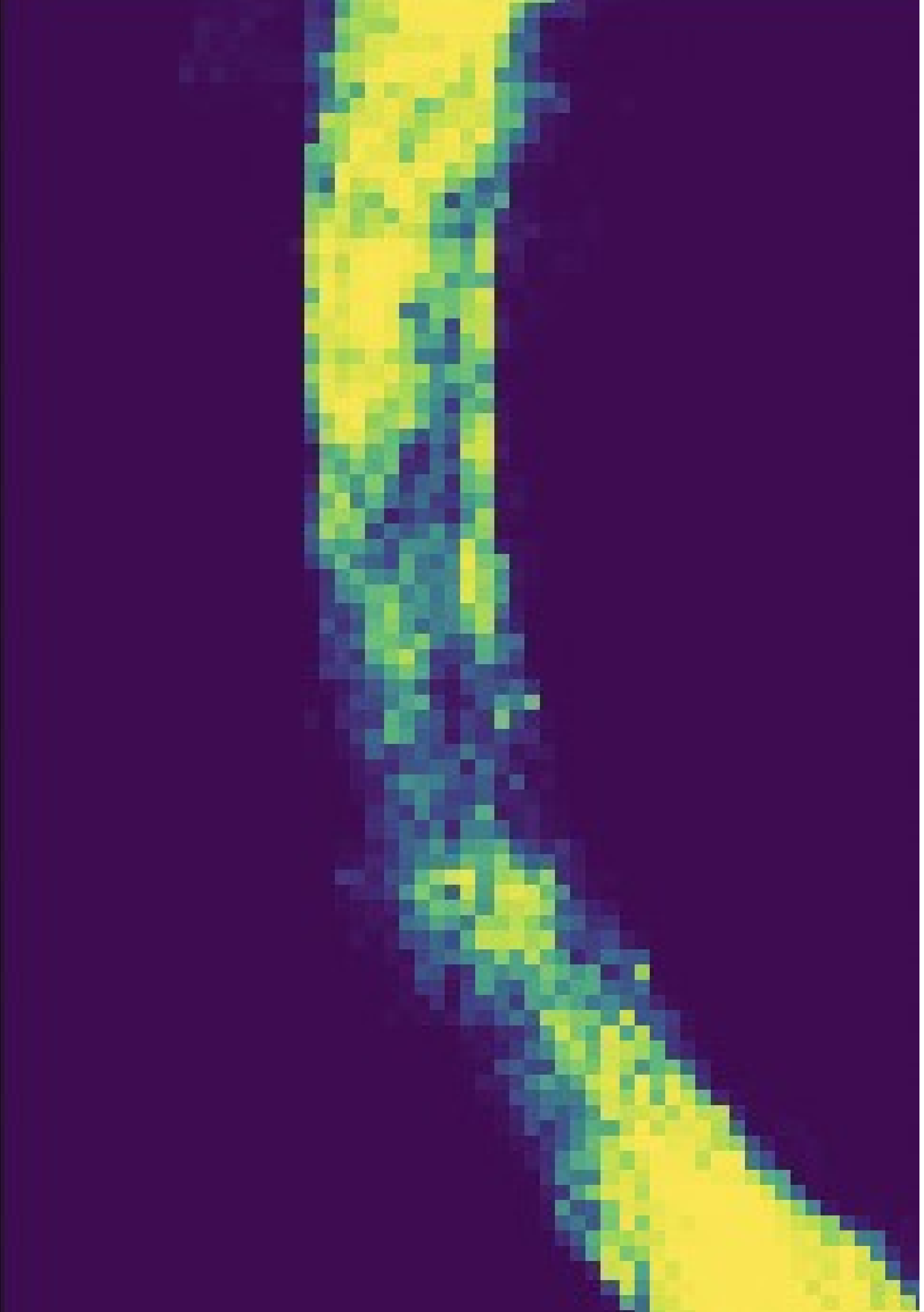


# Improvement in Interval Analysis based Localization through Neural Networks

EDWIN TOMY, DR. LUC JAULIN

# Localization through Interval Computation

The correct localization of mobile robots is necessary, even with the existence of imprecise
sensors which may record geospatial data with a certain degree of uncertainty.

These mobile robots may follow heavily non-linear and not easily represented relations.

Interval Computation is a numerical tool that allows us to solve these non-linear problems in a reliable way, guaranteeing correct results and avoiding false negatives.

# Issue with Interval Computation

An issue with Interval Computation is the time it takes to compute results, making real-time observations computationally expensive.

Average computation time using Codac is around ~0.15s to create an image.

Thus, we propose using neural networks, which could supplant the role of Interval Computation and potentially reduce the time to localize mobile robots, especially in cases where a robot may follow a not easily represented relation.

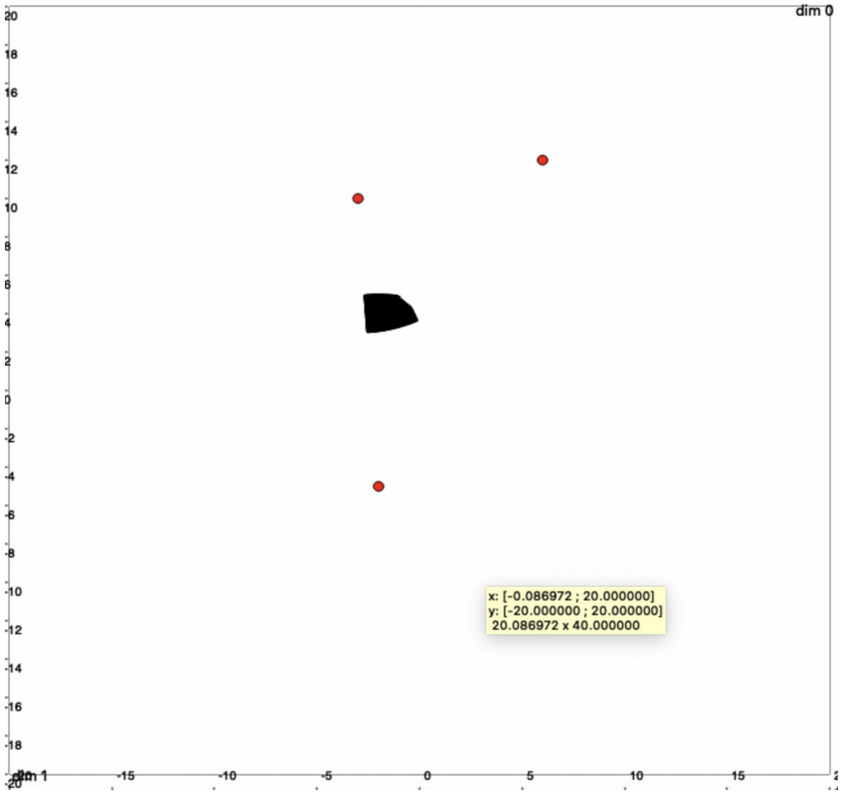
| | |
|---|---|
| Processor | 2 GHz Quad-Core Intel Core i5 |
| Graphics | Intel Iris Plus Graphics 1536 MB |
| Memory | 16 GB 3733 MHz LPDDR4X |

*computer used to run tests*

# Problem Definition

Given coordinates of landmarks and interval distances between landmarks and robot, predict the area where the robot is localized.

Objective - Replicate outputs produced by interval computation methods, using neural networks.



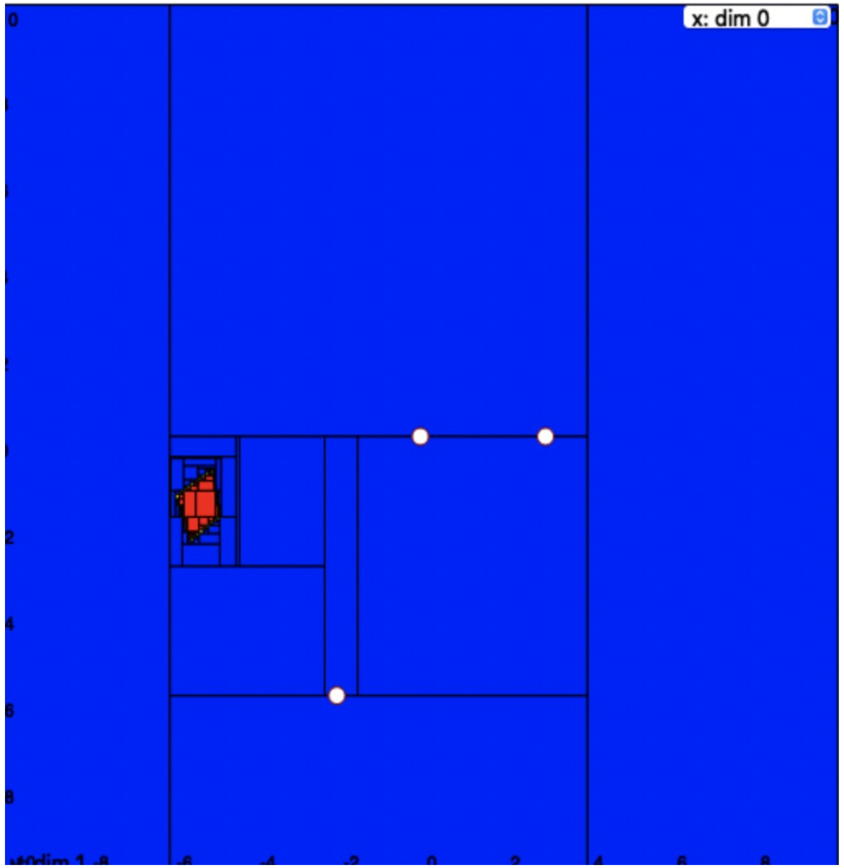
# First Approach

Train neural network with the input being 12 integers representing the coordinates and the interval distances.

We use sigmoid activation and train during 50 epochs with 90,000 images generated with interval computation.

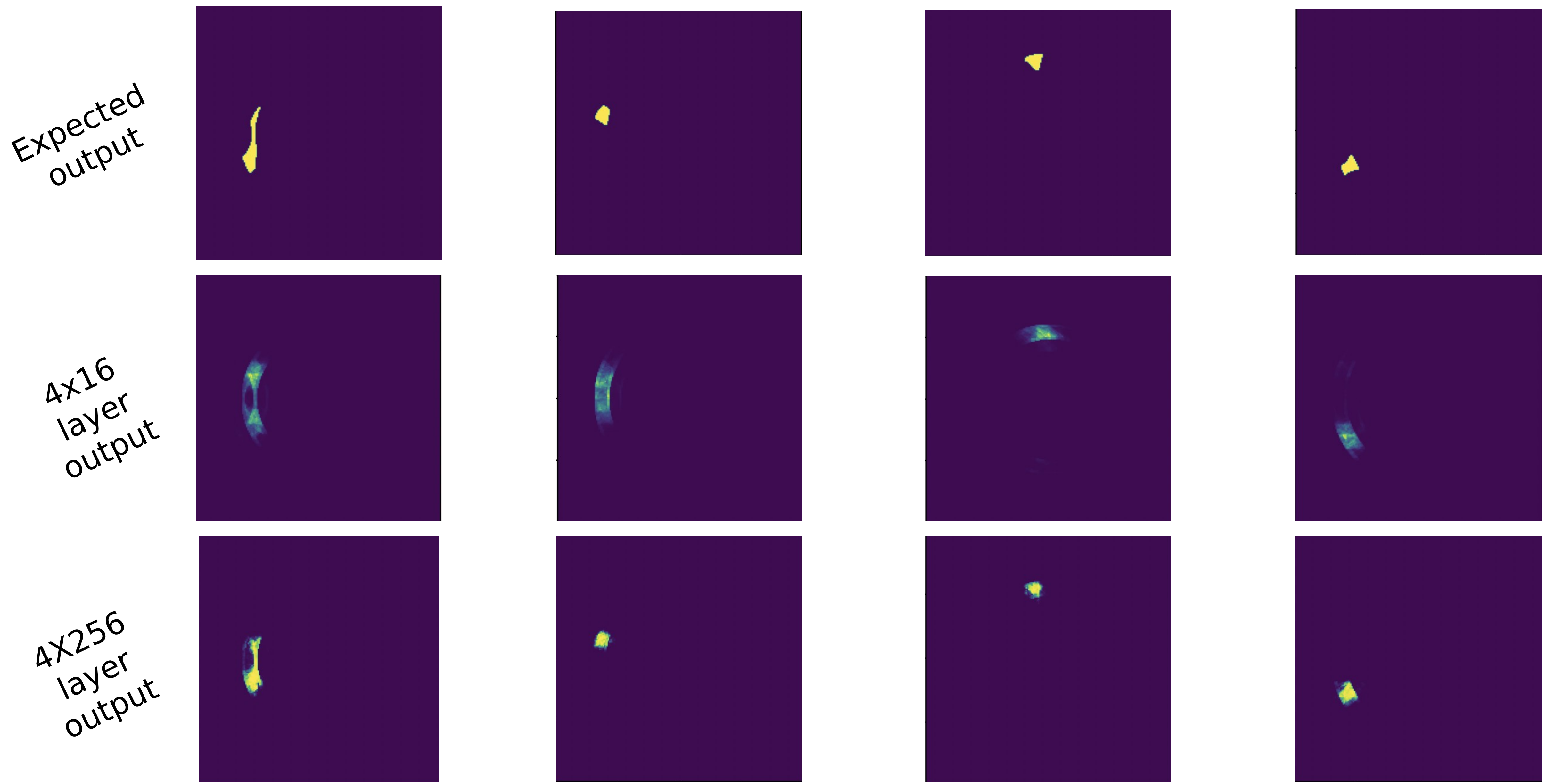
Test different layers widths to compare accuracy, recall and model size.

# Localization using Codac



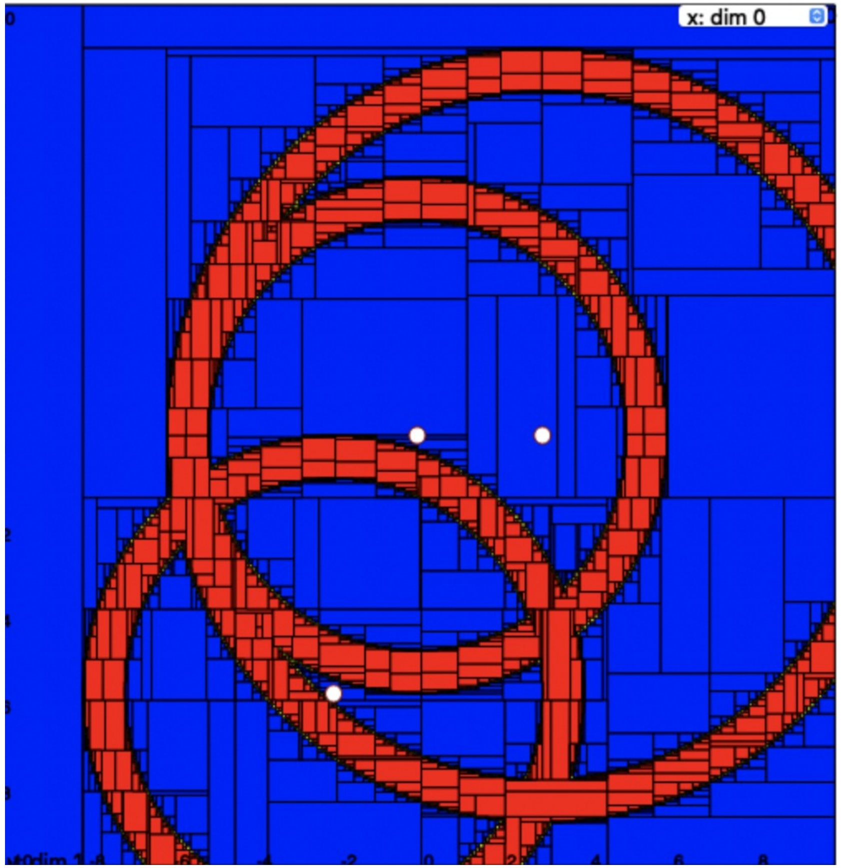
# First Approach

| **Model** | 4 x 256 units | 4 x 128 units | 4 x 64 units | 4 x 32 units | 4 x 16 units |
|---|---|---|---|---|---|
| Test Accuracy | 99.87% | 99.81% | 99.77% | 99.71% | 99.67% |
| Test Precision | 82.46% | 75.09% | 68.85% | 63.18% | 44.69% |
| **Test Recall** | **79.38%** | **62.64%** | **52.61%** | **28.42%** | **08.13%** |
| Model size | 125 MG | 64 MG | 32 MG | 17 MG | 8 MG |
| Time of prediction | ~0.015s | | | | |


Expected output
4x16 layer output
4X256 layer output

# Second Approach

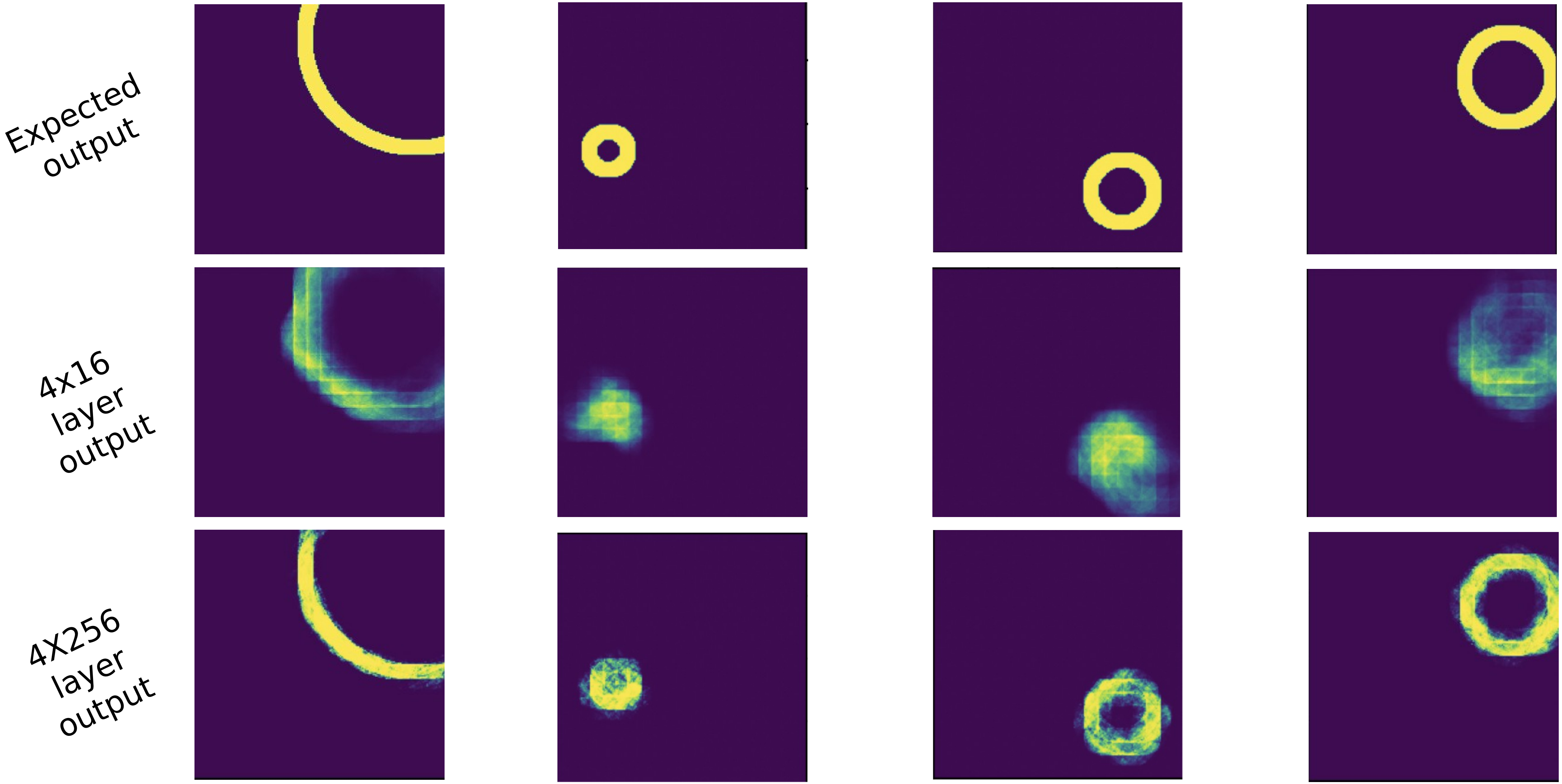
Train neural network for individual landmarks and intersect results.



# Second Approach

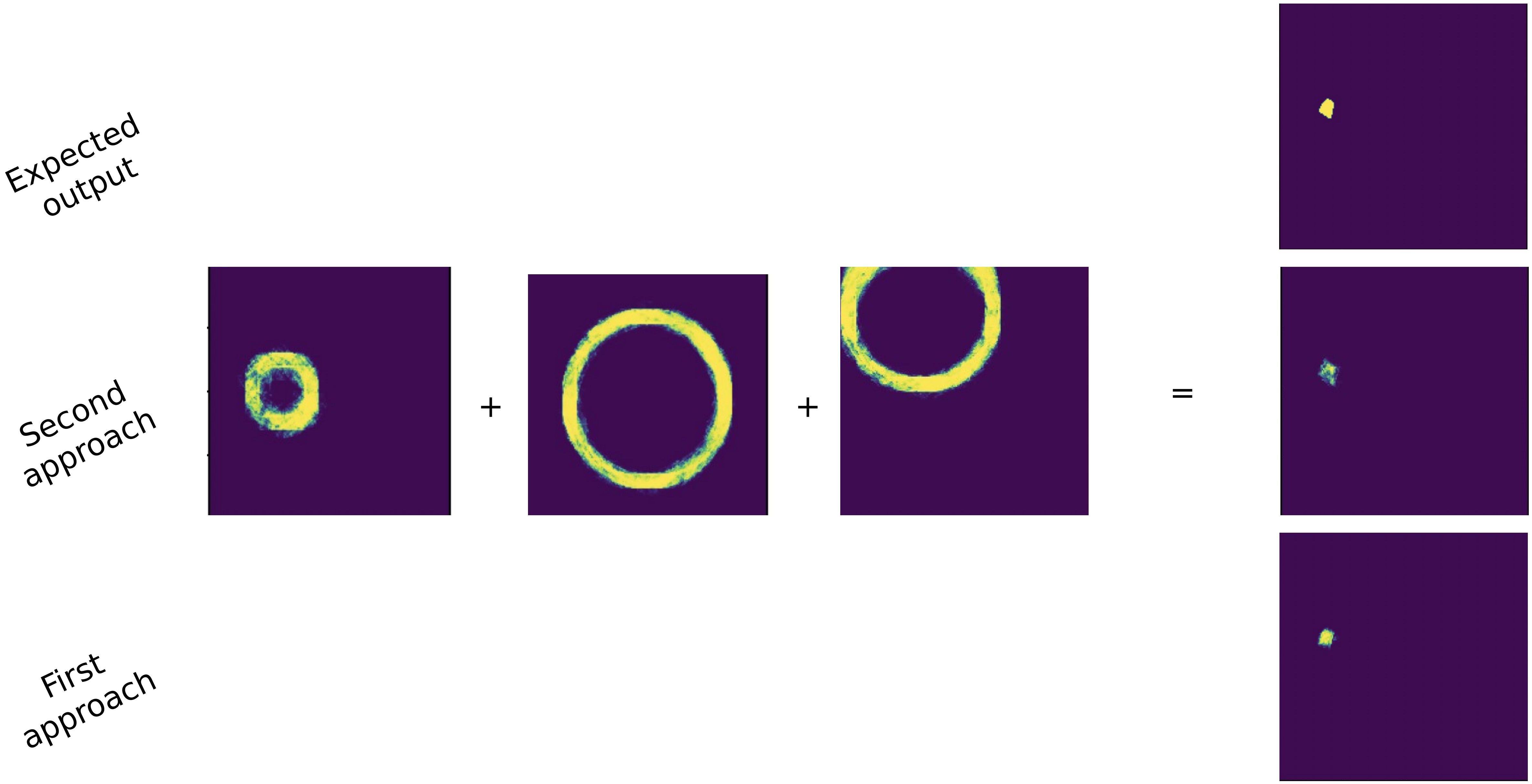
| **Model** | 4 x 256 units | 4 x 128 units | 4 x 64 units | 4 x 32 units | 4 x 16 units |
|---|---|---|---|---|---|
| Test Accuracy | 99.99% | 99.52% | 98.24% | 96.84% | 94.68% |
| Test Precision | 99.96% | 95.82% | 85.49% | 76.09% | 62.06% |
| **Test Recall** | **99.95%** | **96.65%** | **86.81%** | **72.67%** | **39.57%** |
| Model size | 125 MG | 64 MG | 32 MG | 17 MG | 8 MG |
| Time of prediction | ~0.015s | | | | |

*test results for individual outputs, not final image*


Expected output
4x16 layer output
4X256 layer output

# Comparison of Approaches

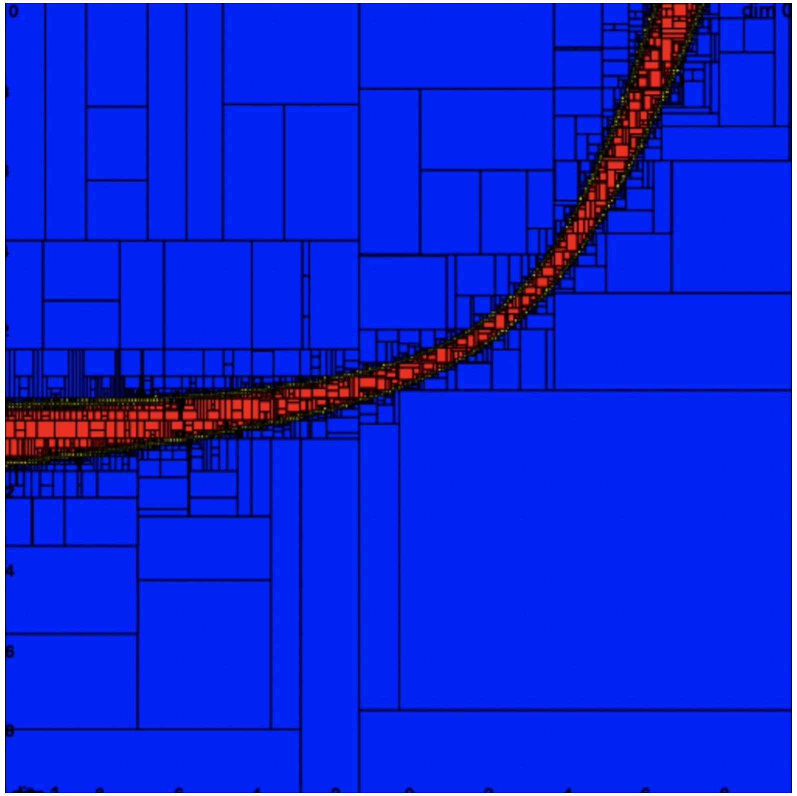
| **Model** | 4 x 256 units | 4 x 128 units | 4 x 64 units | 4 x 32 units | 4 x 16 units |
|---|---|---|---|---|---|
| Test recall of 1st approach | 79.38% | 62.64% | 52.61% | 28.42% | 08.13% |
| Test recall of 2nd approach | 97.02% | 88.40% | 63.64% | 43.32% | 11.89% |
| Model size | 125 Mo | 64 MG | 32 MG | 17 MG | 8 MG |
| Time of prediction | ~0.015s vs ~0.045s | | | | |


Expected output
Second approach
+
+
=
First approach

# $2^{nd}$ Problem Definition: TDoA

Time Difference of Arrival - Given coordinates of landmarks and interval difference of distances between any two landmarks and robot, predict the area where the robot is localized.

Objective - Replicate outputs produced by interval analysis methods, using neural networks.

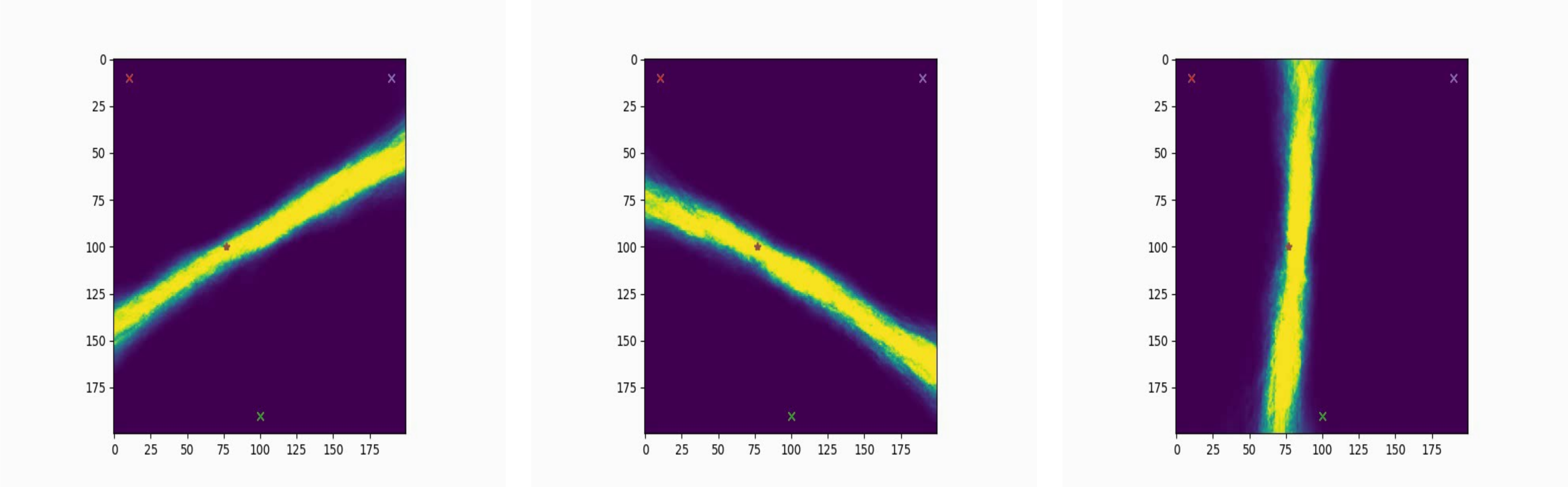


# Results

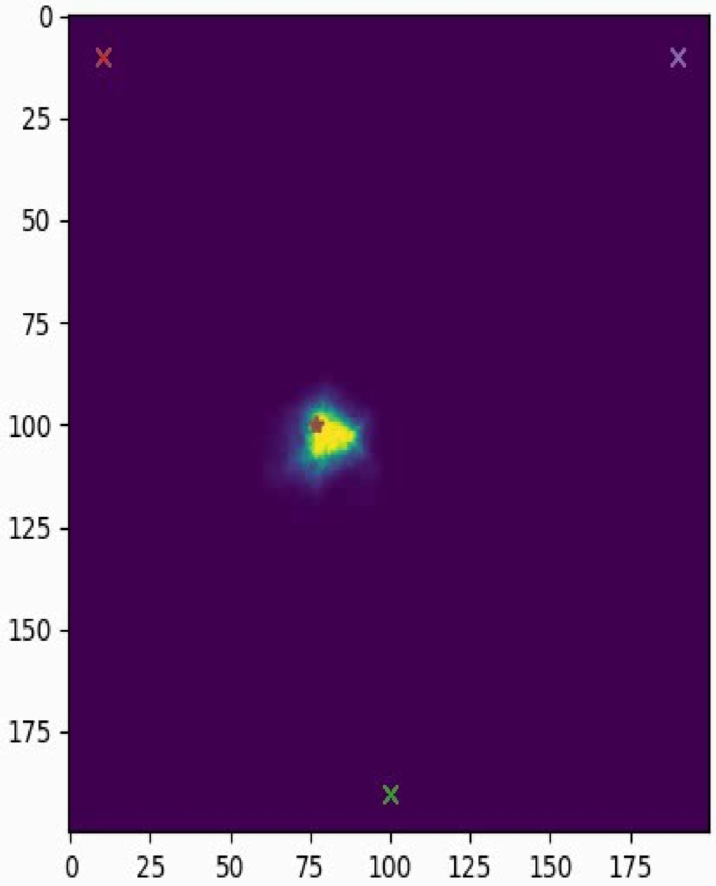
| **Model** | 4 x 256 units | 4 x 128 units | 4 x 64 units | 4 x 32 units | 4 x 16 units |
|---|---|---|---|---|---|
| Test Accuracy | 98.84% | 98.46% | 97.54% | 96.22% | 94.43% |
| Test Precision | 90.14% | 87.65% | 81.49% | 71.90% | 61.91% |
| **Test Recall** | **92.23%** | **88.66%** | **80.07%** | **68.14%** | **35.72%** |
| **Test Recall for Final Output** | **90.06%** | **77.43%** | **54.11%** | **36.41%** | **4.26%** |

# Circular Path Test

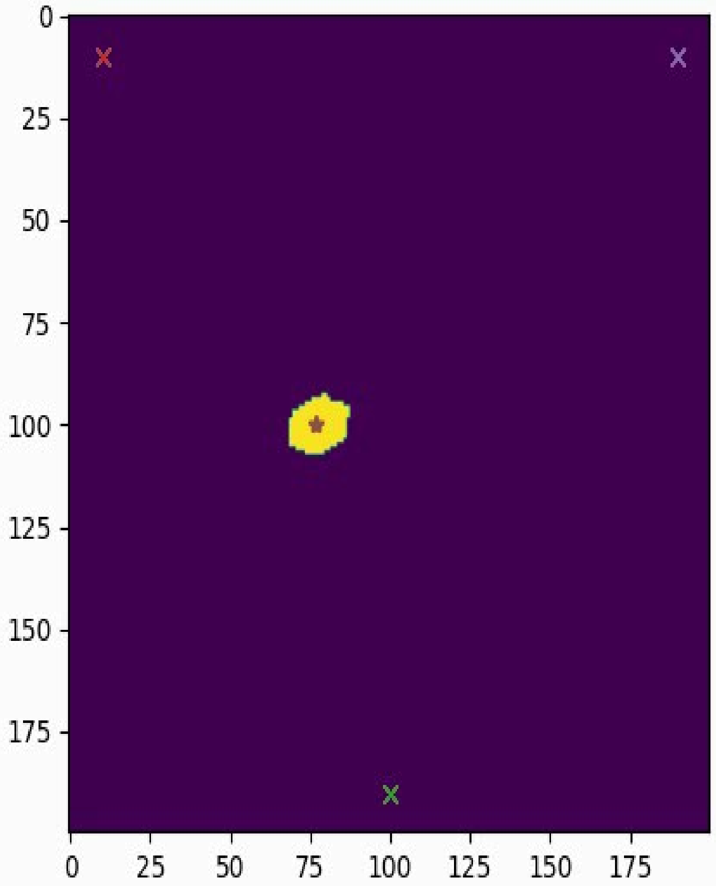
# Per pair of landmarks



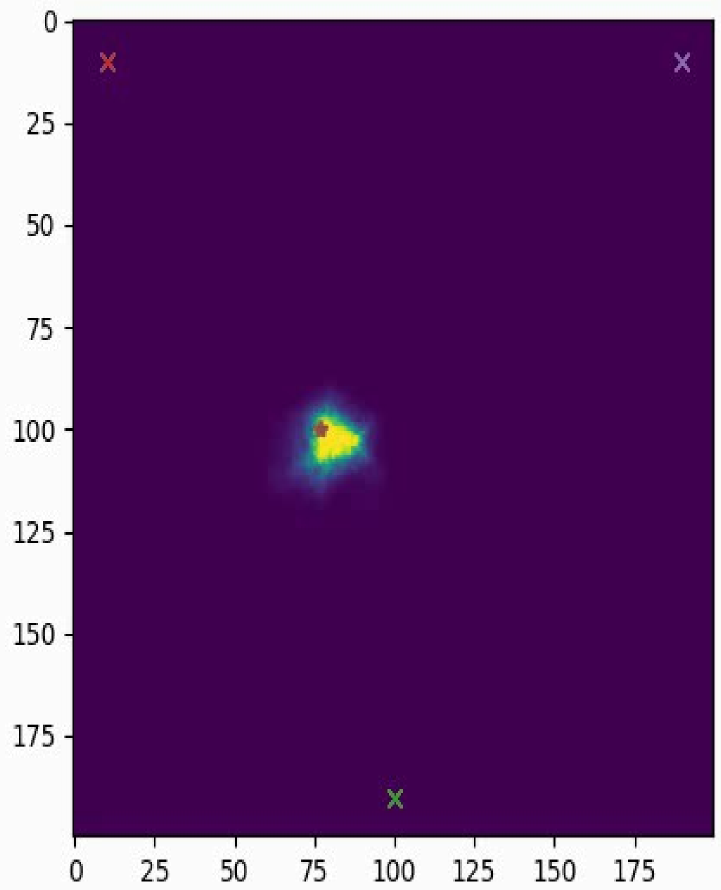
## Neural Network



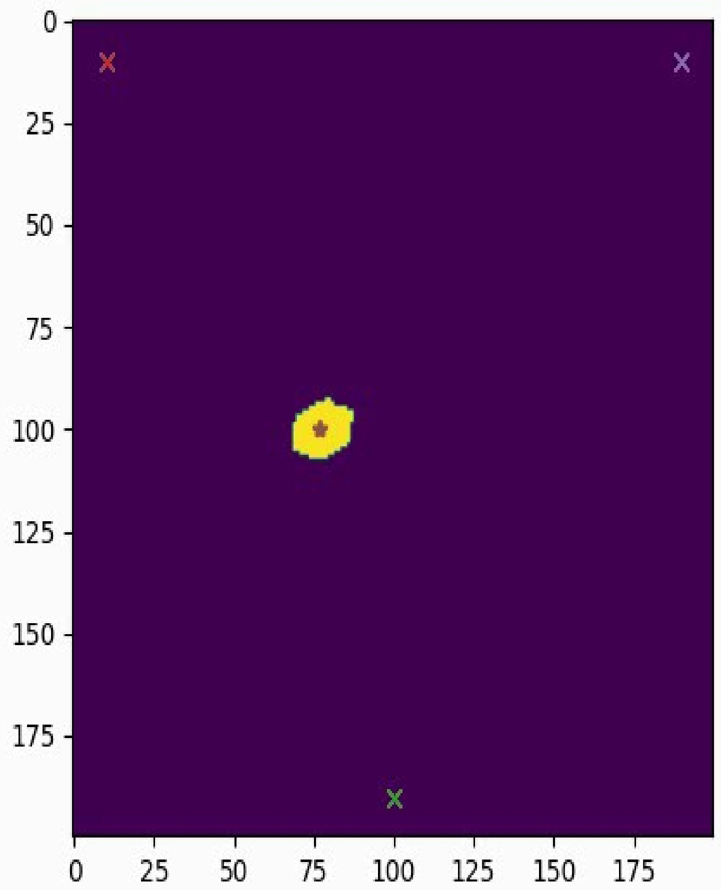

## Interval Computation



Neural Network

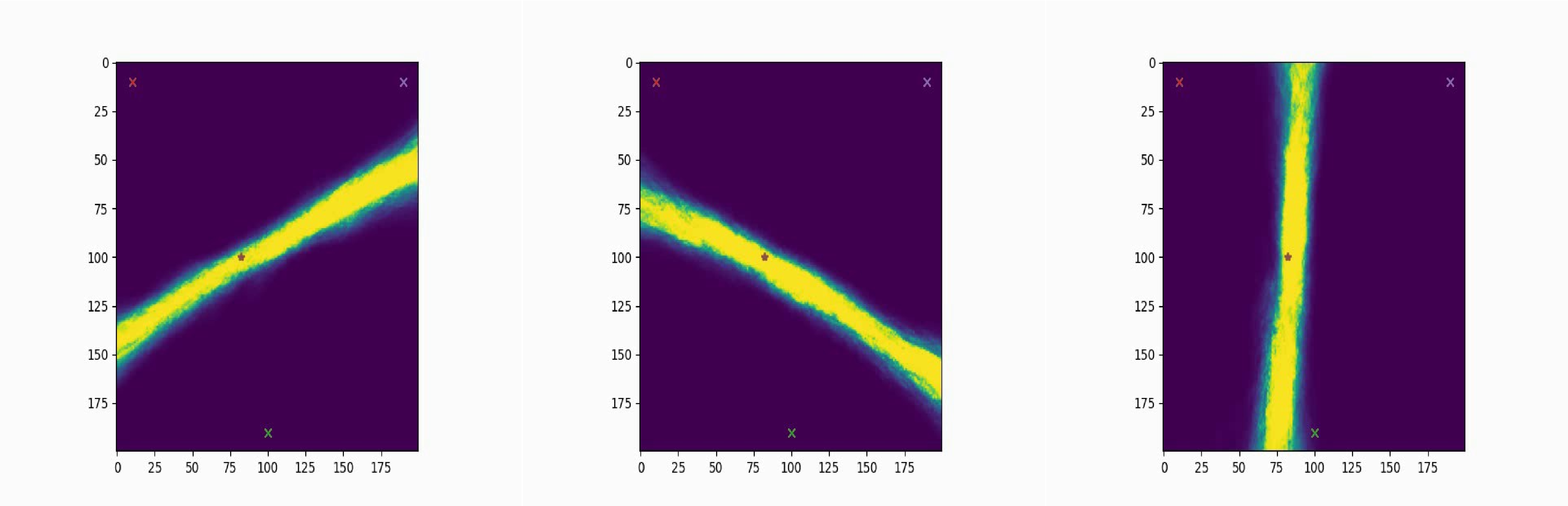



Interval Computation

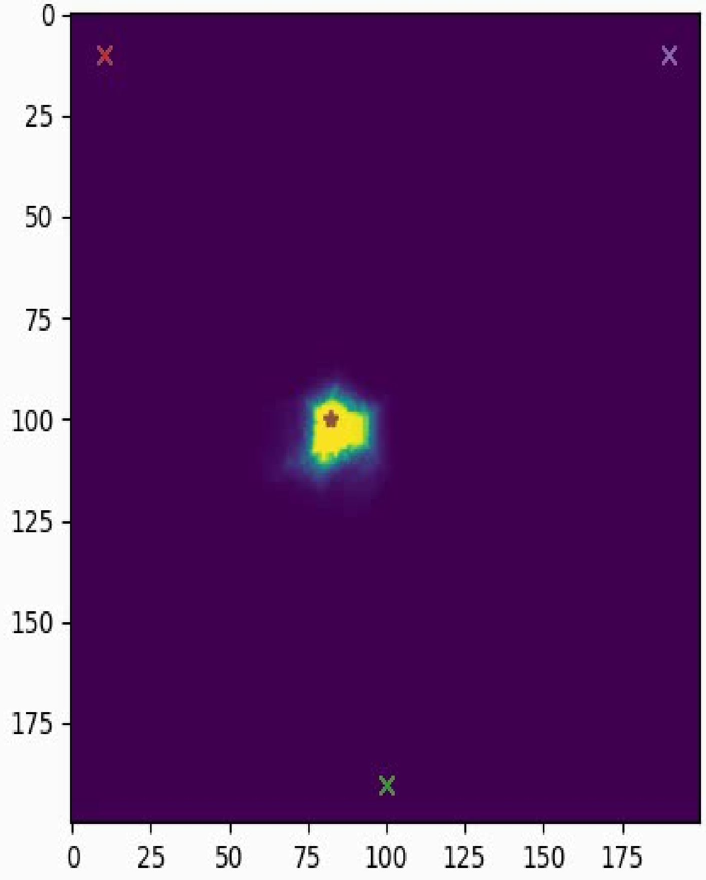


# Straight Path Test

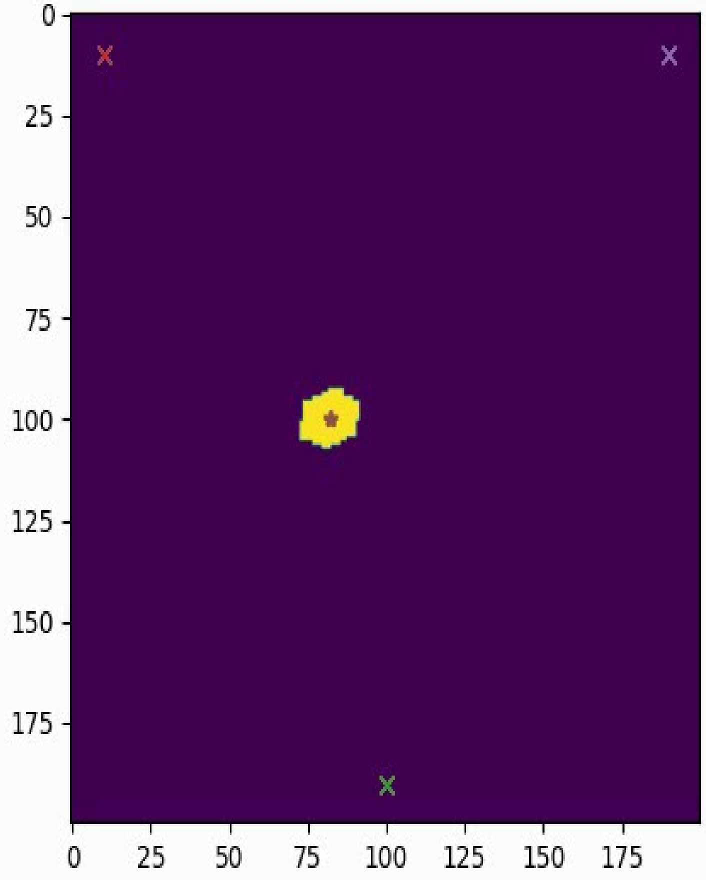
# Per pair of landmarks



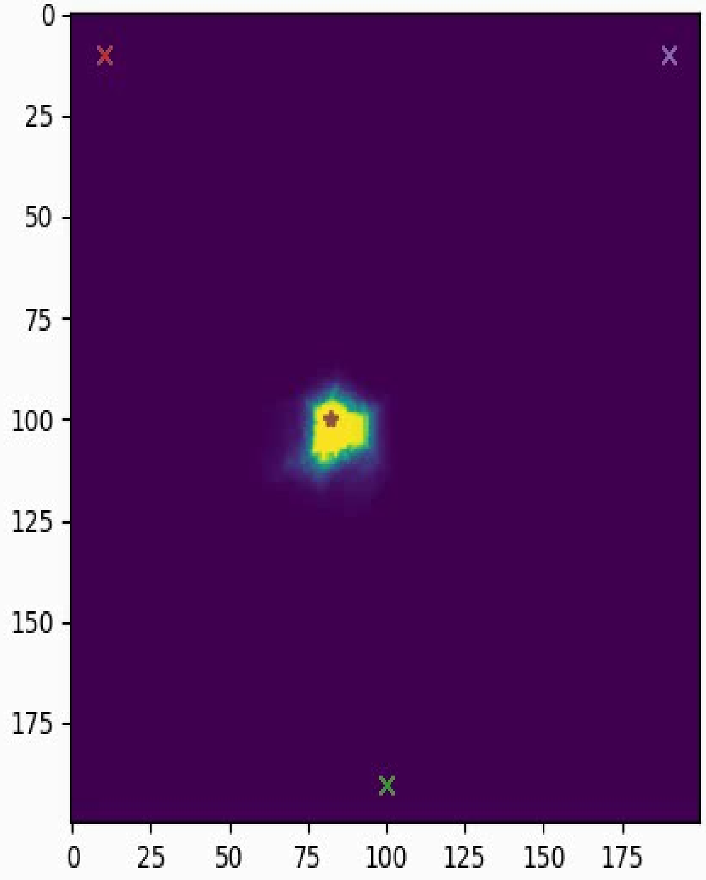
Neural Network



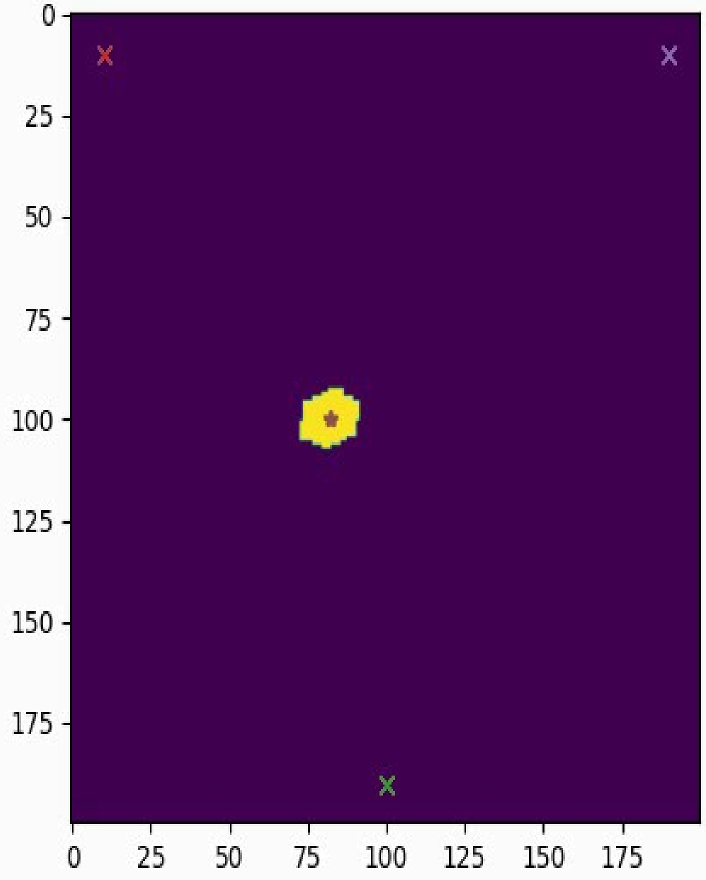

Interval Computation



## Neural Network




## Interval Computation



# Conclusion

Neural Networks reduce time to localize objects, although not significantly. In our examples, they reduced computation time by a third approximately.

Neural Networks may be used to represent more complex relations that may be difficult to graph with simple equations.

Neural Networks can offer highly accurate localization although there will still be false negatives present, unlike in interval computation.

# Future Developments

Add more data where robot is localized near boundaries created by landmarks.

Test different architecture depths.

Train models on fixed landmarks and use geometric transformations to generalize outputs to different landmark locations.

Use pruning methods and TensorFlowLite to reduce the size of neural networks.

Search and test on more complex relations.

# References

Drevelle, V., & Nicola, J. (2014, July 31). *Vibes: A visualizer for intervals and Boxes - Mathematics in Computer Science*. SpringerLink. https://link.springer.com/article/10.1007/s11786-014-0202-0

Jaulin, L. (n.d.). *Karush-Kuhn-Tucker conditions to build efficient contractors ...* https://www.ensta-bretagne.fr/jaulin/paper_ctctdoa.pdf. https://www.researchgate.net/publication/371684678_Karush-Kuhn-Tucker_conditions_to_build_efficient_contractors_Application_to_TDoA_localization

Jaulin, L., Desrochers, B., Rohou, S., & Rinin, J. (n.d.). *Introduction*. IAMOOC. https://www.ensta-bretagne.fr/jaulin/iamooc.html

Rohou, S., & Desrochers, B. (2022). The Codac library -- Constraint-programming for robotics. http://codac.io


Thank you
for your
attention!