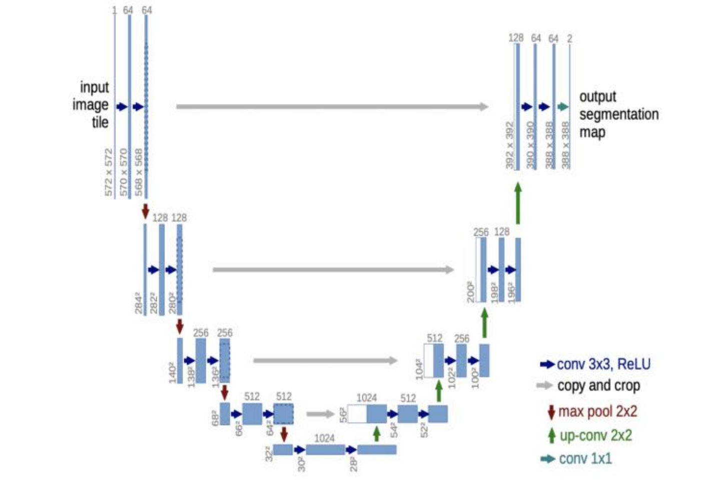


CSE 598: Bio-Inspired AI and Optimization

Using ABC Algorithm as a Hyperheuristic for Optimal UNet Hyperparameter Search

Yue Zhao
Kylel Scott

Motivation

```
Model: "U-Net"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 128, 128, 3  0           []
                                )]
```

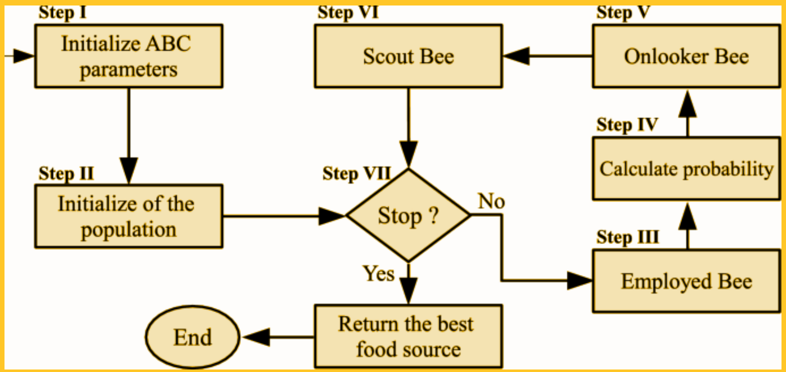
```
 dropout_6 (Dropout)            (None, 64, 64, 256)  0           ['concatenate_2[0][0]']

 conv2d_14 (Conv2D)             (None, 64, 64, 128)  295040      ['dropout_6[0][0]']

 conv2d_15 (Conv2D)             (None, 64, 64, 128)  147584      ['conv2d_14[0][0]']

 conv2d_transpose_3 (Conv2DTran  (None, 128, 128, 64  73792      ['conv2d_15[0][0]']
 spose)                         )

 concatenate_3 (Concatenate)    (None, 128, 128, 12  0           ['conv2d_transpose_3[0][0]',
                                8)                                'conv2d_1[0][0]']

 dropout_7 (Dropout)            (None, 128, 128, 12  0           ['concatenate_3[0][0]']
                                8)

 conv2d_16 (Conv2D)             (None, 128, 128, 64  73792       ['dropout_7[0][0]']
                                )

 conv2d_17 (Conv2D)             (None, 128, 128, 64  36928       ['conv2d_16[0][0]']
                                )

 conv2d_18 (Conv2D)             (None, 128, 128, 3)  195         ['conv2d_17[0][0]']

==================================================================================================
Total params: 15,646,286
Trainable params: 15,646,286
Non-trainable params: 0
__________________________________________________________________________________________________
```

The motivation for this research focused on the following:

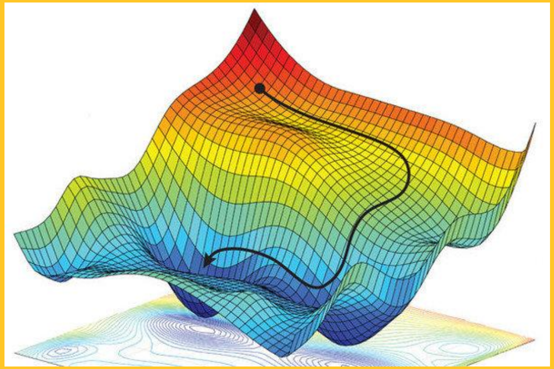
- Hyperparameter tuning by hand is **time-consuming**
- Unoptimized and/or unintelligent automated processes for training models to evaluate model accuracy is **resource intensive** and **wasteful**

Problem Definition

In order to save time and resources during the training and tuning of a **U-Net Autoencoder** architecture for **image segmentation**, the team employed **Artificial Bee Colony Optimization** to intelligently automate hyperparameter tuning through iterative selection.



https://link.springer.com/article/10.1007/s00500-018-3299-2



https://www.researchgate.net/figure/Non-convex-optimization-We-utilize-stochastic-gradient-descent-to-find-a-local-optimum_fig1_325142728

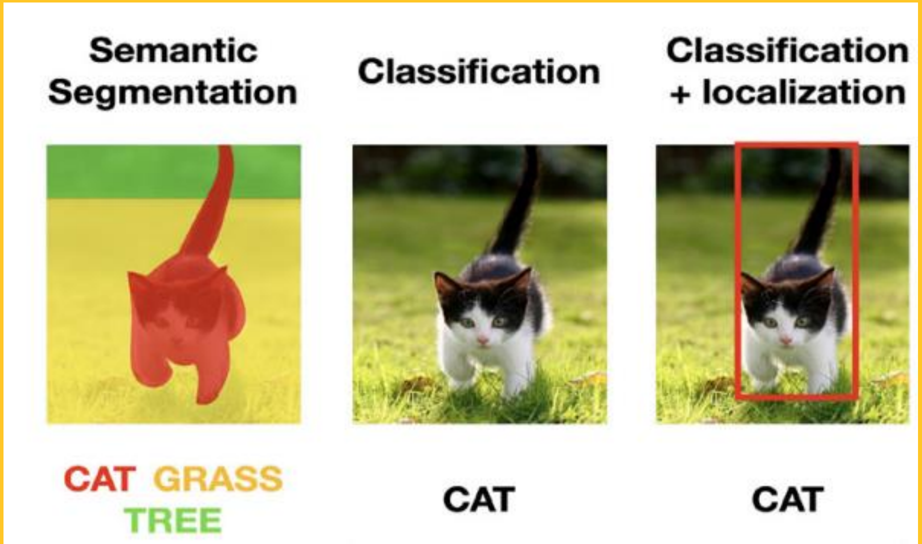
Implementation

The team considered the set of parameters shown below, and optimized these hyperparameters through the lifecycle of the algorithm. The **search space** of hyperparameters was explored in efforts to find the optimal set that warranted the highest training accuracy. **Learning rate, training epochs, batchsize, pooling type, convolutional strides, and dropout rate** were the composed the hyperparameter search space.

	Training centered			*Architecture Centered*		
Hyperparameters	Learning Rate	Training Epochs	Batch size	Convolutional Stride	Dropout Rate	Pooling type

Background: Semantic Segmentation

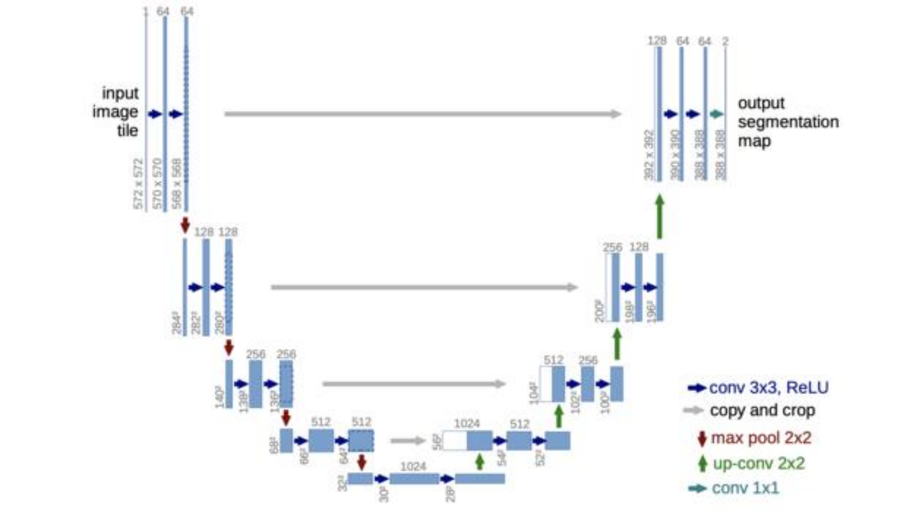
- The process of identifying pixels that belong to **specific groups** of objects in an image.
- A pixel-wise analogue to **Classification and Localization**
- Allows for more **refined** analysis of image, usually for systems that rely on precise comprehension of **object boundaries** (Medical Scanning, Robotic Autonomy, etc.)



[1] https://www.implantology.or.kr/articles/pdf/RvNO/kaomi-2020-024-03-5.pdf

Background: U-Net

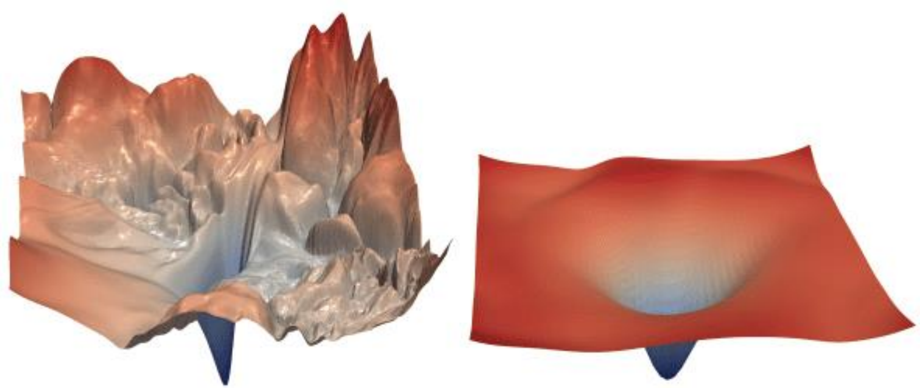
- Improvement upon the then-popular FCN
- It is an **encoder-decoder network** with **skip connections** between identical resolution contracting-expansive layers
- Commonly used for applications that require pixel-wise accuracy (segmentation, image refinement)
- Excels from its ability to **iteratively refine** low-resolution and concatenate latent-space data sequentially to **generate precise outputs**
- Help with **vanishing gradient**, and has been observed to change loss landscape [1].



[1] https://arxiv.org/abs/1712.09913
[2] https://arxiv.org/pdf/1505.04597.pdf

Background: U-Net

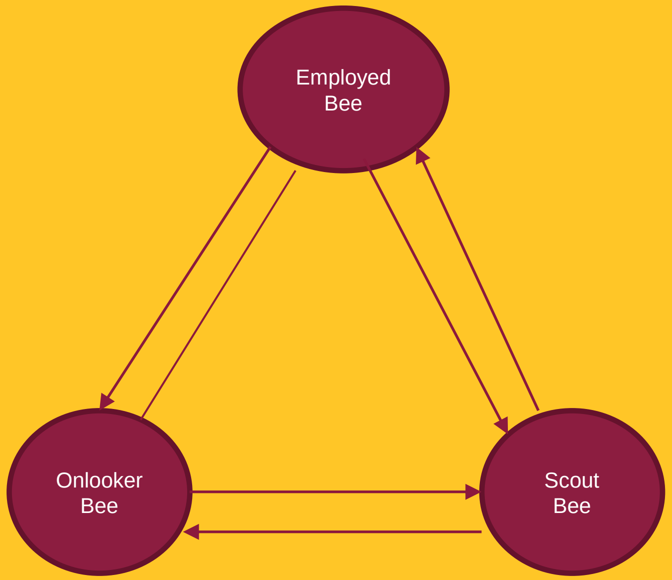
- Improvement upon the then-popular FCN
- It is an **encoder-decoder network** with **skip connections** between identical resolution contracting-expansive layers
- Commonly used for applications that require pixel-wise accuracy (segmentation, image refinement)
- Excels from its ability to **iteratively refine** low-resolution and concatenate latent-space data sequentially to **generate precise outputs**
- Help with **vanishing gradient**, and has been observed to change loss landscape [1].



(a) without skip connections (b) with skip connections

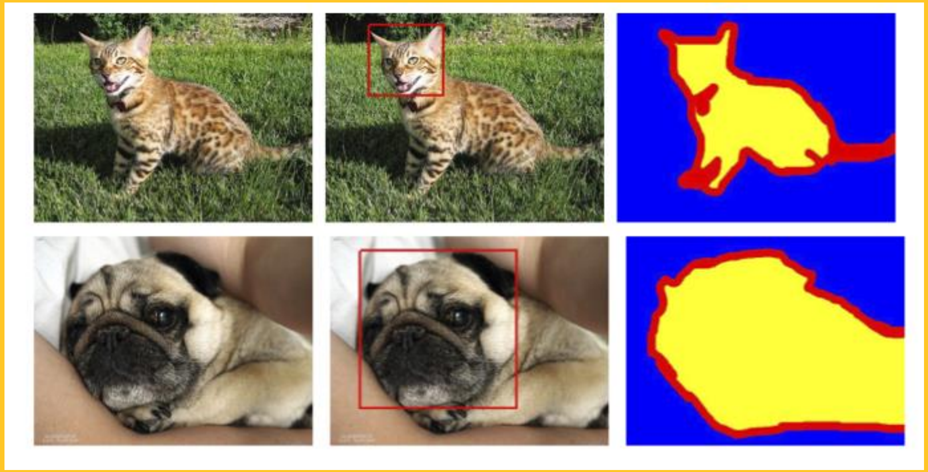
[1] https://arxiv.org/abs/1712.09913
[2] https://arxiv.org/pdf/1505.04597.pdf

Background: Artificial Bee Colony Algorithm



Food source: the feasible solution to the optimization problem to be solved, and its merit is evaluated by fitness.
Employed Bee: use a greedy criterion to compare the optimal solution in memory with the domain search solution, keeping the best solution.
Onlooker Bee: Based on information about the food source of the employed bees, neighbors close to the food source are selected with a certain probability.
Scout bee: use big step search to explore the optimal solution and avoid local optimal when the solution is trapped somewhere.

Implementation: Resources and Dataset



The Oxford-IIIT Pet Dataset

A 37 category pet dataset with roughly 200 images for each class. The images have a large variations in scale, pose and lighting. All images have an associated ground truth annotation of breed, head ROI, and pixel level trimap segmentation.

[1] https://www.robots.ox.ac.uk/~vgg/data/pets/

Steps of ABC

- Generate initial honey sources and determine their location; and initialize the algorithm parameters

$$x_{id} = L_d + rand(0,1)(U_d - L_d)$$

- The employed bee searches for a new nectar source and replaces it if it is a better one.

$$x_{id}^{new} = x_{id} + a^{*}\varphi\left(x_{id} - x_{jd}\right) \quad j \neq i$$

Steps of ABC

- Calculate the probability of the employed bee being followed.

$$p_i = \frac{fit_i}{\sum_{i=1}^{nPop} fit_i}$$

- If a nectar source does not find a better one within the threshold L, it is abandoned. The scout bee is left to find a better nectar source.

$$x_i = \begin{cases} L_d + rand(0,1)(U_d - L_d) & trial \geq L \\ x_i & trial < L \end{cases}$$

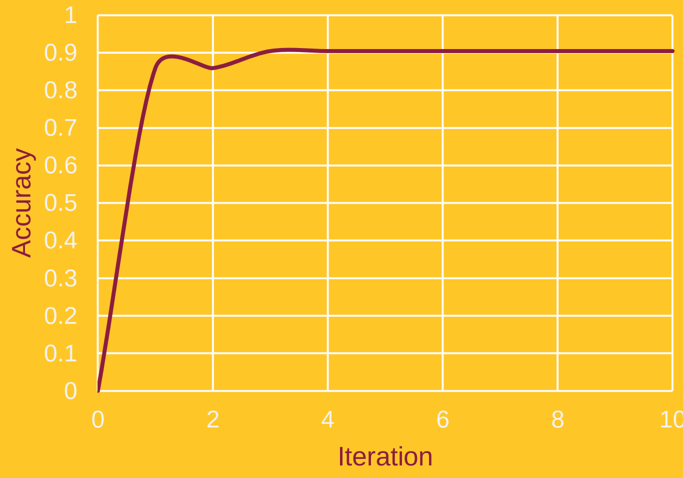
- Keep iterating until $maxgen\ th$ generation

Implementation: UNet and ABC Initialization

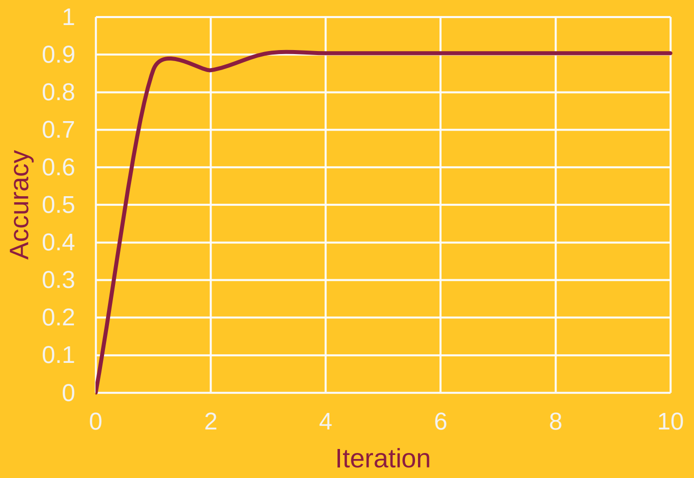
- Randomly generate multiple sets of **initial hyperparameters** as **employed bees**
- find the initial **accuracy** of each combination as a **nectar source**
- **iterates** through all hyperparameter combinations to **generate a new combination**
- analyzes whether the parameters in the new combination are **out of bounds**
- If the accuracy of **the new combination is higher**, replace it.
- Calculate the **cumulative probability**, which is the probability that the employed bee will be followed
- Select the employed bee according to the **roulette wheel rule**, **generate** a new combination, **compare** accuracy
- the combination is **discarded** if a better accuracy is **not found within the threshold L**. **Regenerate** the combination as a scout bee to find a better nectar source.

Implementation: Training

Iteration	Learning Rate	Epochs	Dropout Rate	Batch Size	Pooling Type	Accuracy
1	0.001	10	0.1	128	AP	0.85878
2	0.001	10	0.1	128	AP	0.85878
3	0.001	10	0.0939	64	MP	0.90419
4	0.001	10	0.093866	64	MP	0.904196
5	0.001	10	0.093866	64	MP	0.904196
6	0.001	10	0.093866	64	MP	0.904196
7	0.001	10	0.093866	64	MP	0.904196
8	0.001	10	0.093866	64	MP	0.904196
9	0.001	10	0.093866	64	MP	0.904196
10	0.001	10	0.093866	64	MP	0.904196



Discussion



- The increase is **rapid** because we choose **a small range of epoch**.
- The **tradeoff** between accuracy and computation
- using **validation datasets**, which means training on a small number of data.
- The **onlooker bee** can find a **better solution in the neighborhood of the nectar source** where the employed bee is located, **converging the iterative results toward the optimal value**.
- The **scout bees** can **avoid the local optimal** by **skipping the neighborhood** where the optimal solution is available and searching without being bound by the existing results.

References

[1] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, pages 234–241. Springer, 2015
[2] Kumar A.,Kumar D. and Jarial S. (2017) A Review on Artificial Bee Colony Algorithms and Their Applications to Data Clustering. Cybernetics and Information Technologies, Vol.17 (Issue 3), pp. 3-28.
[3] Karaboga, D. and Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39(3), pp.459-471.
[4] Karaboga, D. and Akay, B., 2009. A comparative study of artificial bee colony algorithm. Applied mathematics and computation, 214(1), pp.108-132.
[5] Wang, S.G. and Jiang, S., 2022. Optimal Hyperparameters and Structure Setting of Multi-Objective Robust CNN Systems via Generalized Taguchi Method and Objective Vector Norm. arXiv preprint arXiv:2202.04567.