## Model: UNET

---

Parameters:

Backbone : backbone models
Type:list
Ex:
VGG16,
VGG19
ResNet50
ResNet101
ResNet152
ResNet50V2,
ResNet101V2,
ResNet152V2
DenseNet121,
DenseNet169,
DenseNet201
EfficientNetB[0-7]

Weights: initializing the backbone model with pretrained weights or not
Type: list
Ex : imagenet ,None

Freeze_backbone: option about freezing the weights of the pretrained backbone model
Type: Bolean
Ex: True, False

Freeze_batch_norm : freeze the inner state of all batch normalize layer
Type: Bolean
Ex: True, False

filter_num: a list that defines the number of convolutional filters per down- and up-sampling blocks.
Type: array with number of values
Min len of the array :2 , the max len of the array :6
Each number min :8 max :1024 ,
Ex: [32, 64, 128, 256, 512, 1024]

activation: the activation function of hidden layers.
Type: list
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

Output_activation : the activation function of the output layer
Type: list
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'

batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".

Type: Bolean

Ex: True, False

stack_num_down: number of convolutional layers per downsampling level.

Type: int

Ex: 1, min :1 max:500

stack_num_up: number of convolutional layers (after concatenation) per upsampling level.

Type: int

Ex: 1, min :1 max:500

pool: the configuration of downsampling (encoding) blocks.

Type:list

Ex:

False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).

True or 'max' downsampling with a max-pooling layer.

'ave' downsampling with a average-pooling layer

unpool: the configuration of upsampling (decoding) blocks.

Type:list

Ex:

False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)

True or 'bilinear' upsampling with bilinear.

'nearest' upsampling with reflective padding

---

## Model: VNET

---

Parameters:

filter_num: a list that defines the number of convolutional filters per down- and up-sampling blocks.
Type: array with number of values
Min len of the array :2 , the max len of the array :6
Each number min :8 max :1024 ,
Ex: [32, 64, 128, 256, 512, 1024]

activation: the activation function of hidden layers.
Type: list
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

Output_activation : the activation function of the output layer
Type: list
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'

batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".
Type: Bolean
Ex: True, False

res_num_max: the max number of convolutional layers within a residual block.
Type: int
Ex: 1 min :1

res_num_ini :number of convolutional layers of the first first residual block (before downsampling)
Type: int
Ex: 1

pool: the configuration of downsampling (encoding) blocks.
Type:Bolean , str
Ex:
False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
True or 'max' downsampling with a max-pooling layer.
'ave' downsampling with a average-pooling layer

unpool: the configuration of upsampling (decoding) blocks.
Type:Bolean , str
Ex:
False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)
True or 'bilinear' upsampling with bilinear.
'nearest' upsampling with reflective padding

---

## Model: UNET++

---

**Parameters:**

**Backbone** :
Type:str
Ex:
VGG16,
VGG19
ResNet50
ResNet101
ResNet152
ResNet50V2,
ResNet101V2,
ResNet152V2
DenseNet121,
DenseNet169,
DenseNet201
EfficientNetB[0-7]

**weights:** initializing the backbone model with pretrained weights or not
Type: str
Ex : imagenet ,None

**Freeze_backbone**: option about freezing the weights of the pretrained backbone model

Type: Bolean
Ex: True, False

**Freeze_batch_norm** :freeze the inner state of all batch normalize layer

Type: Bolean
Ex: True, False

**filter_num**: a list that defines the number of convolutional filters per down- and up-sampling blocks.
Type: array with number of values
Min len of the array :2 , the max len of the array :6
Each number min :8 max :1024 ,
Ex: [32, 64, 128, 256, 512, 1024]

**activation**: the activation function of hidden layers.
Type: str
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

**Output_activation** : the activation function of the output layer
Type: str
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'

batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".

Type: Bolean

Ex: True, False

stack_num_down: number of convolutional layers per downsampling level.

Type: int

Ex: 2,

stack_num_up: number of convolutional layers (after concatenation) per upsampling level.

Type: int

Ex: 2,

pool: the configuration of downsampling (encoding) blocks.

Type:Bolean , str

Ex:

False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).

True or 'max' downsampling with a max-pooling layer.

'ave' downsampling with a average-pooling layer

unpool: the configuration of upsampling (decoding) blocks.

Type:Bolean , str

Ex:

False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)

True or 'bilinear' upsampling with bilinear.

'nearest' upsampling with reflective padding

deep_supervision: deep supervision in segmentation models applies the loss function at multiple network layers to improve gradient flow and feature learning.

Type:Bolean

Ex:

False .True

---

## Model: UNET3+

---

Parameters:

Backbone : backbone models
Type:str
Ex:
VGG16,
VGG19
ResNet50
ResNet101
ResNet152
ResNet50V2,
ResNet101V2,
ResNet152V2
DenseNet121,
DenseNet169,
DenseNet201
EfficientNetB[0-7]
weights: initializing the backbone model with pretrained weights or not
Type: str
Ex : imagenet ,None

Freeze_backbone: option about freezing the weights of the pretrained backbone model

Type: Bolean
Ex: True, False

Freeze_batch_norm :freeze the inner state of all batch normalize layer

Type: Bolean
Ex: True, False

filter_num_down: a list that defines the number of convolutional filters per downsampling blocks.
Type: list
Ex: [32, 64, 128, 256, 512, 1024]

activation: the activation function of hidden layers.
Type: str
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

Output_activation : the activation function of the output layer
Type: str
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'

batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".
Type: Bolean
Ex: True, False

stack_num_down: number of convolutional layers per downsampling level.
Type: int
Ex: 2,

stack_num_up: number of convolutional layers (after concatenation) per upsampling level.

Type: int
Ex: 2,

pool: the configuration of downsampling (encoding) blocks.
Type:Bolean , str
Ex:
False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
True or 'max' downsampling with a max-pooling layer.
'ave' downsampling with a average-pooling layer

unpool: the configuration of upsampling (decoding) blocks.
Type:Bolean , str
Ex:
False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)
True or 'bilinear' upsampling with bilinear.
'nearest' upsampling with reflective padding

deep_supervision: deep supervision in segmentation models applies the loss function at multiple network layers to improve gradient flow and feature learning.

Type:Bolean
Ex:
False .True

filter_num_skip:a list that defines the number of filters after each full-scale skip connection ,note the length of the list must be the same and the length of number of filter option
Type:list or string
Ex:
'auto'
Or [32, 32, 32, 32]

filter_num_aggregate:an integer that sets the channel number for aggregating multi-scale features, critical for enhancing the network's segmentation performance
Type:int or string
Ex:
'auto'
Or 5 min :1 max:64

---

## Model: R^2UNET

---

Parameters:

filter_num: a list that defines the number of convolutional filters per down- and up-sampling blocks.
Type: array with number of values
Min len of the array :2 , the max len of the array :6
Each number min :8 max :1024 ,
Ex: [32, 64, 128, 256, 512, 1024]

activation: the activation function of hidden layers.
Type: str
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

Output_activation : the activation function of the output layer
Type: str
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'

batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".
Type: Bolean
Ex: True, False

rstack_num_down: number of convolutional layers per downsampling level.
Type: int
Ex: 2,

stack_num_up: number of convolutional layers (after concatenation) per upsampling level.

Type: int
Ex: 2,

pool: the configuration of downsampling (encoding) blocks.
Type:Bolean , str
Ex:
False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
True or 'max' downsampling with a max-pooling layer.
'ave' downsampling with a average-pooling layer

unpool: the configuration of upsampling (decoding) blocks.
Type:Bolean , str
Ex:
False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)
True or 'bilinear' upsampling with bilinear.
'nearest' upsampling with reflective padding

recur_num: number of recurrent iterations. Type: int Ex: 2

---

## Model: AttentionUNET

---

Parameters:

Backbone : backbone models
Type:str
Ex:
VGG16,
VGG19
ResNet50
ResNet101
ResNet152
ResNet50V2,
ResNet101V2,
ResNet152V2
DenseNet121,
DenseNet169,
DenseNet201
EfficientNetB[0-7]
weights: initializing the backbone model with pretrained weights or not
Type: str
Ex : imagenet ,None

Freeze_backbone: option about freezing the weights of the pretrained backbone model

Type: Bolean
Ex: True, False

Freeze_batch_norm :freeze the inner state of all batch normalize layer

Type: Bolean
Ex: True, False

filter_num: a list that defines the number of convolutional filters per down- and up-sampling blocks.
Type: array with number of values
Min len of the array :2 , the max len of the array :6
Each number min :8 max :1024 ,
Ex: [32, 64, 128, 256, 512, 1024]

activation: the activation function of hidden layers.
Type: str
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

Output_activation : the activation function of the output layer
Type: str
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'

batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".
Type: Bolean
Ex: True, False

stack_num_down: number of convolutional layers per downsampling level.
Type: int
Ex: 2,

stack_num_up: number of convolutional layers (after concatenation) per upsampling level.

Type: int
Ex: 2,

pool: the configuration of downsampling (encoding) blocks.
Type:Bolean , str
Ex:
False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
True or 'max' downsampling with a max-pooling layer.
'ave' downsampling with a average-pooling layer

unpool: the configuration of upsampling (decoding) blocks.
Type:Bolean , str
Ex:
False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)
True or 'bilinear' upsampling with bilinear.
'nearest' upsampling with reflective padding

atten_activation: specifies the type of nonlinear activation function used in the attention
Type: str
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

Attention: determines the style of the attention mechanism. 'add' (additive attention) and 'multiply' (multiplicative attention) are options that describe how attention weights interact with the feature maps
Type: str
Ex: 'add', 'multiply'

---

## Model: U^2NET

---

Parameters:

filter_num_down: list that defines the number of RSU output filters for each downsampling level
Type: list ,str Ex: [16, 32, 64, 512] ,'auto'

filter_mid_num_down: list that defines the number of RSU intermediate filters for each downsampling level
.
Type: list ,str Ex: [16, 32, 64, 128] ,'auto'

filter_mid_num_up: list that defines the number of RSU intermediate filters for each upsampling level
.
Type: list ,str Ex: [16, 32, 64, 128] ,'auto'

filter_4f_num: a list that defines the number of RSU-4F output filters for each downsampling and bottom level

Type: list ,str Ex: [512, 512] ,'auto'

filter_4f_mid_num: a list that defines the number of RSU-4F intermediate filters for each downsampling and bottom level

Type: list ,str Ex: [512, 512] ,'auto'

activation: the activation function of hidden layers.
Type: str
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

Output_activation : the activation function of the output layer
Type: str
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'
batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".
Type: Bolean Ex: True, False

stack_num_down: number of convolutional layers per downsampling level.
Type: int
Ex: 2,

stack_num_up: number of convolutional layers (after concatenation) per upsampling level.

Type: int
Ex: 2,

pool: the configuration of downsampling (encoding) blocks.
Type:Bolean , str
Ex:
False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
True or 'max' downsampling with a max-pooling layer.
'ave' downsampling with a average-pooling layer

deep_supervision: deep supervision in segmentation models applies the loss function at multiple network layers to improve gradient flow and feature learning.

Type:Bolean
Ex:
False .True

unpool: the configuration of upsampling (decoding) blocks.
Type:Bolean , str
Ex:
False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)
True or 'bilinear' upsampling with bilinear.
'nearest' upsampling with reflective padding

---

## Model: ResUNET

---

Parameters:
filter_num: a list that defines the number of convolutional filters per down- and up-sampling blocks.
Type: array with number of values
Min len of the array :2 , the max len of the array :6
Each number min :8 max :1024 ,
Ex: [32, 64, 128, 256, 512, 1024]

aspp_num_up: defines the number of filters (e.g., 128) in the ASPP layer after the final upsampling block in ResUNet, enhancing high-resolution feature capture.
Type: int Ex: 128

aspp_num_down: sets the number of filters (e.g., 256) in the ASPP layer following the last downsampling block, improving multi-scale context extraction before upsampling.

Type: int Ex: 256

activation: the activation function of hidden layers.
Type: str
Ex: 'ReLU', 'LeakyReLU', 'PReLU', 'ELU', 'GELU', 'Snake'

output_activation : the activation function of the output layer
Type: str
Ex: 'Sigmoid', 'Softmax', None (linear), 'Snake'
batch_norm: if specified as True, all convolutional layers will be configured as stacks of "Conv2D-BN-Activation".
Type: Bolean Ex: True, False

pool: the configuration of downsampling (encoding) blocks.
Type:Bolean , str
Ex:
False: downsampling with a convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation).
True or 'max' downsampling with a max-pooling layer.
'ave' downsampling with a average-pooling layer

unpool: the configuration of upsampling (decoding) blocks.
Type:Bolean , str
Ex:
False: upsampling with a transpose convolutional layer (2-by-2 convolution kernels with 2 strides; optional batch normalization and activation)
True or 'bilinear' upsampling with bilinear.
'nearest' upsampling with reflective padding

---

## Notes:

---

- **For parameters lacking defined min/max values or list size constraints, their configuration depends on the user's discretion and the capabilities of the device hosting the ARK container .**

- **While values must exceed one, there's no set maximum, allowing customization based on the device's computational capacity and specific user requirements.**