Deep Dive into U-Net Architecture for segmentation

Detailed Explanation of U-Net Architecture

Arthur Cartel Foahom Gouabou, PhD | https://cartelgouabou.github.io/

Clarity in Complexity

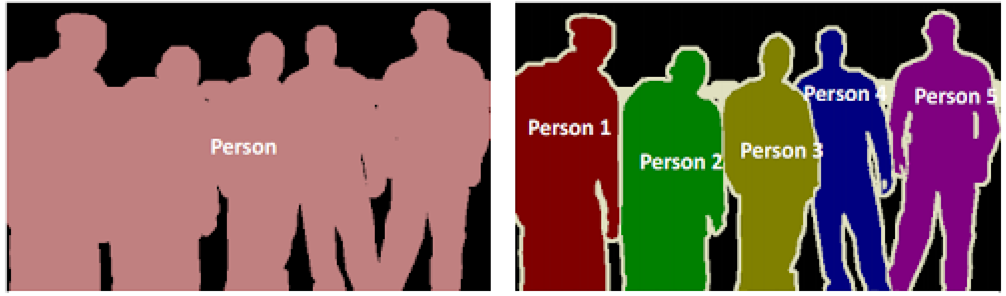
1. Introduction to image segmentation

What is Image Segmentation ?

Definition: Image segmentation is the process of partitioning an image into multiple meaningful segments or regions to simplify or change the representation of an image into something more meaningful and easier to analyze.

Types of Image Segmentation:

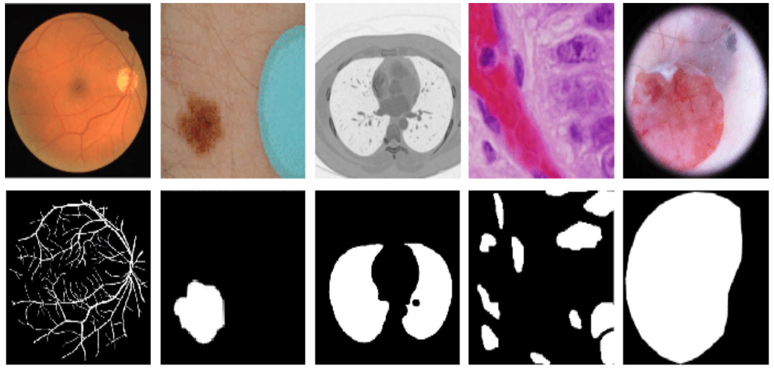
- **Semantic Segmentation**: Classifying each pixel into a predefined class (e.g; identifying all cars in an image).
- **Instance Segmentation**: Identifying and segmenting each object instance separately (e.g., labeling each car individually).



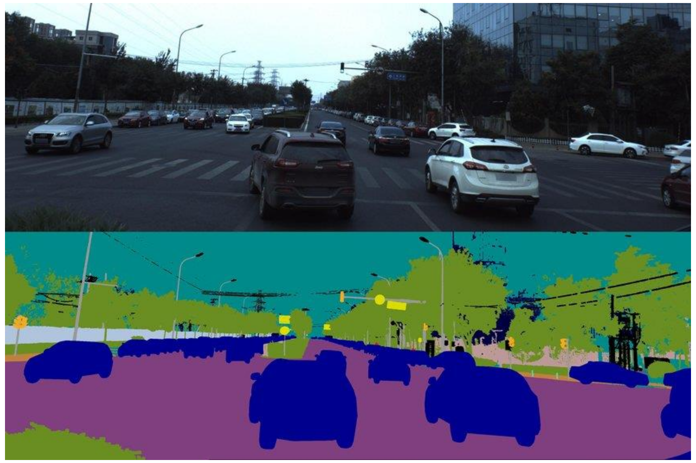
Semantic vs Instance

1. Introduction to image segmentation

Importance of Image Segmentation



Medical Imaging
(tumor detections)



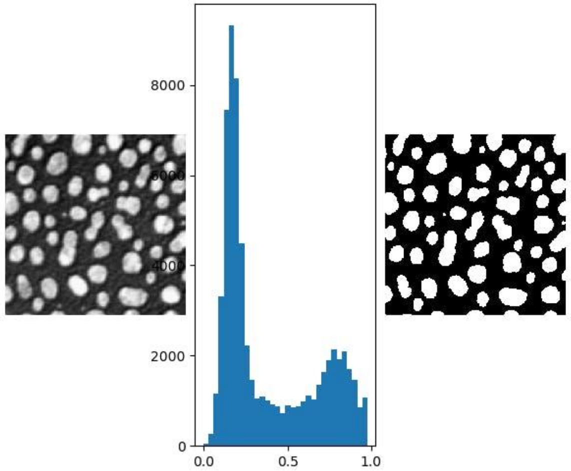
Autonomous driving
(road and pedestrain
detection)

Satellite imagery
(land use
classification)

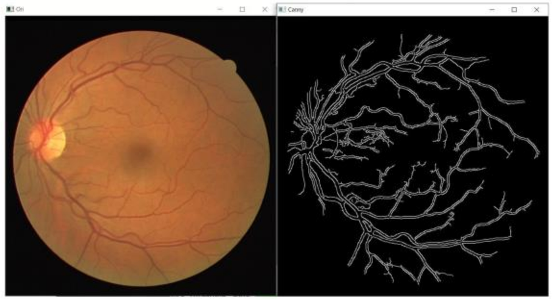
Clarity in Complexity

1. Introduction to image segmentation

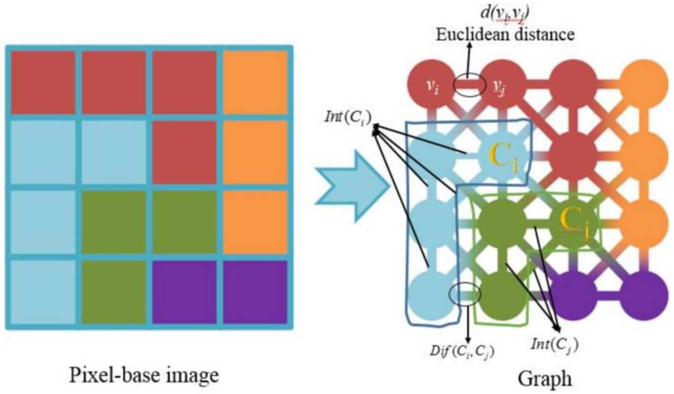
Traditional Techniques Before Deep Learning



Thresholding



Edge detection
(Sobel, canny)



Graph-Based Methods

Others:

- Region-Based Segmentation
- Clustering-Based Methods (K-means)

2. The Emergence of Deep Learning and Convolutional Neural Networks

Challenges with Traditional Techniques

- Limited ability to handle complex, high-dimensional data.
- Sensitivity to noise and variations in lighting.
- Difficlty in capturing global context and intricate pattterns.

Rise of CNNs in Image Processing

- **Advantages of CNNs**: Ability to learn hierarchical feature representation, robustnes to variations, and scalability.
- **Early CNN-Based Segmentation Methods**: Path-wise classification, Fully Convolutional Networks (FCNs).

3. U-Net Architecture for Image Segmentation

The U-Net Paper

Ronneberger, O. et al. (2015). U-net: Convolutional networks for biomedical image segmentation. In–MICCAI 2015, October 5-9, 2015, proceedings, part III 18 (pp. 234-241). Springer International Publishing.

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de
http://lmb.informatik.uni-freiburg.de/

Abstract. There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong

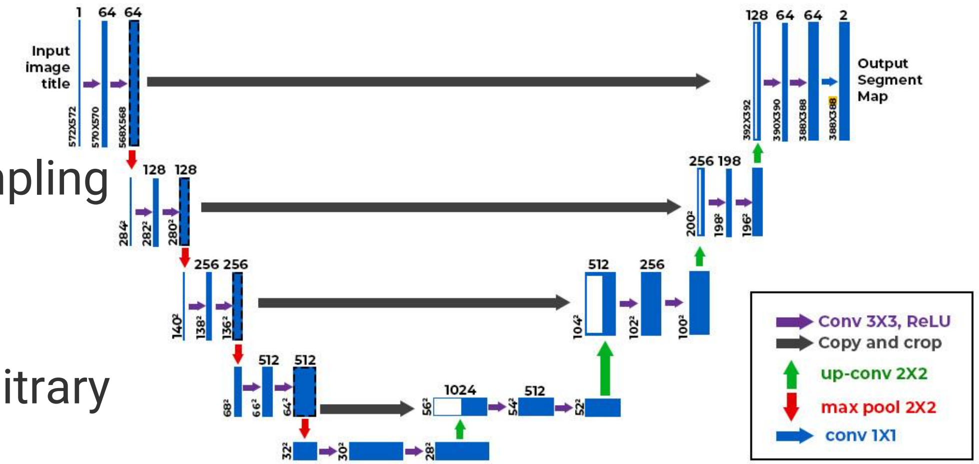
Key Contributions of U-Net

- **Encoder-Decode Structure**: Combines feature extraction with spatial localization.
- **Skip Connections**: Bridges the contracting and expanding paths to retain spatial information.
- **Designed for Biomedical Applications**: Specifically tailored for tasks with limited training data.

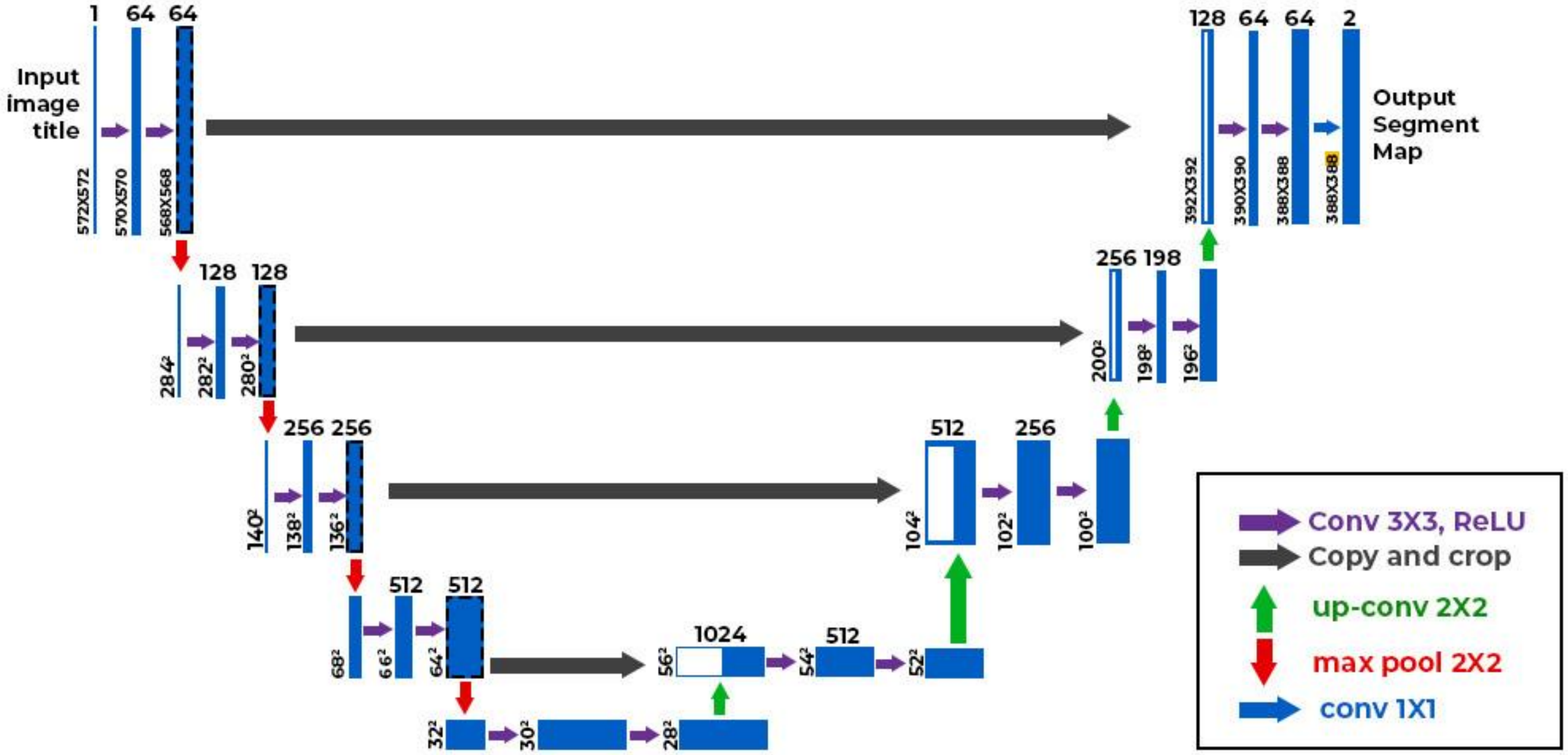
3. U-Net Architecture for Image Segmentation

Overview of U-Net Architecture

- **Contracting Path (Encoder):** Captures context via downsampling
- **Expanding Path (Decoder):** Enables precise localization via updampling and skip connections
- **Fully convolutional**: No fully connected layers, allowing for arbitrary input sizes
- **Symmetric Design**: Mirror-like structure facilitating effective feature fusions.Sensitivity to noise and variations in lighting.
- **Skip Connections**: The feature maps from the encoder are concatenated with the upsampled feature maps in the decoder, allowing for better localization.



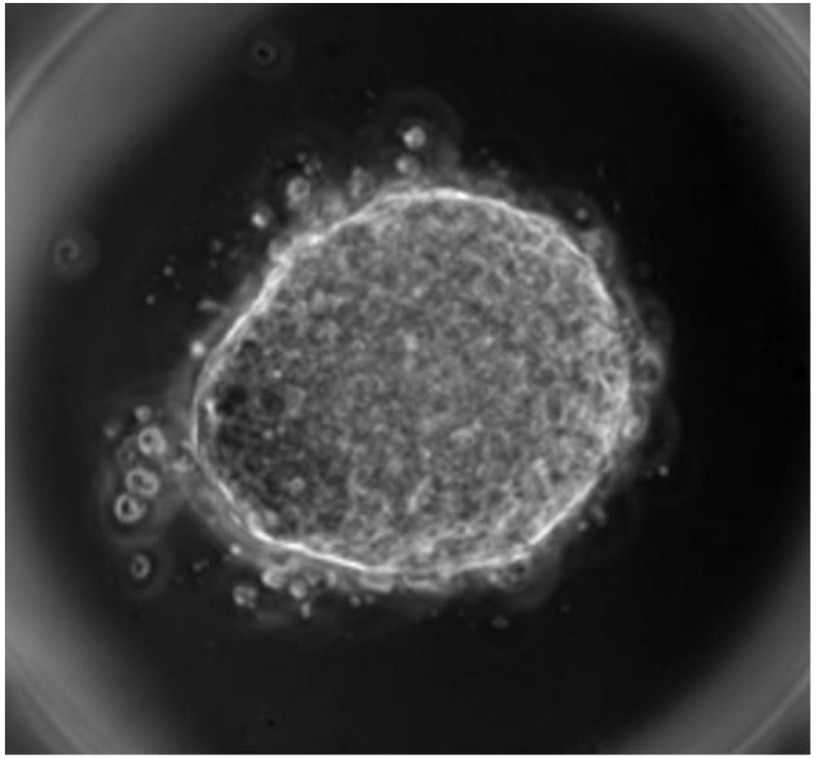
4. Detailed Explanation of U-Net Architecture



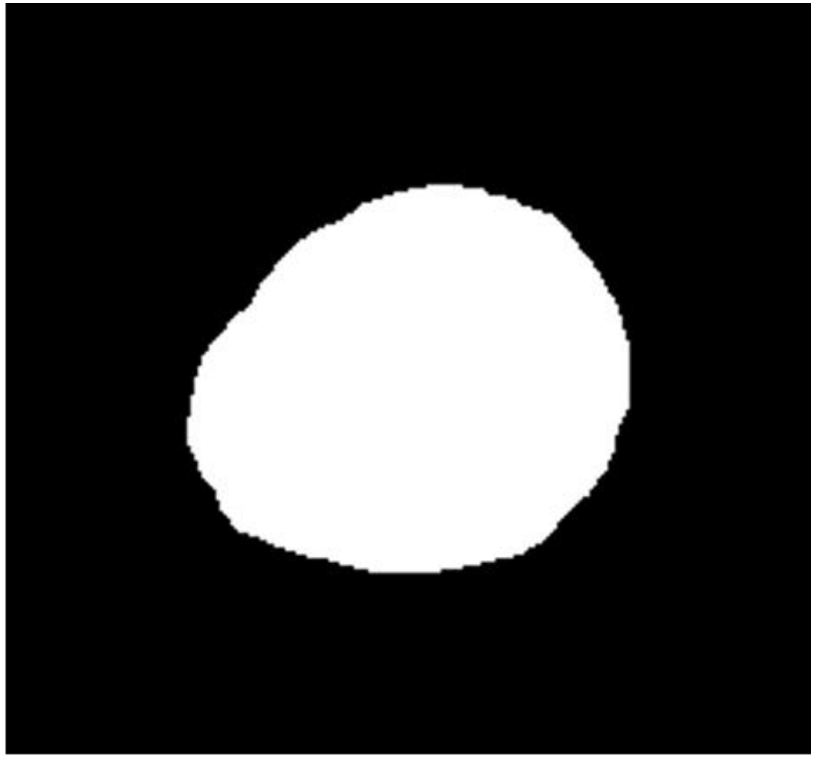
U-Net Architecture

Clarity in Complexity

4. Detailed Explanation of U-Net Architecture



Input Image



Output Mask

Sample grayscale image with it mask

4. Detailed Explanation of U-Net Architecture

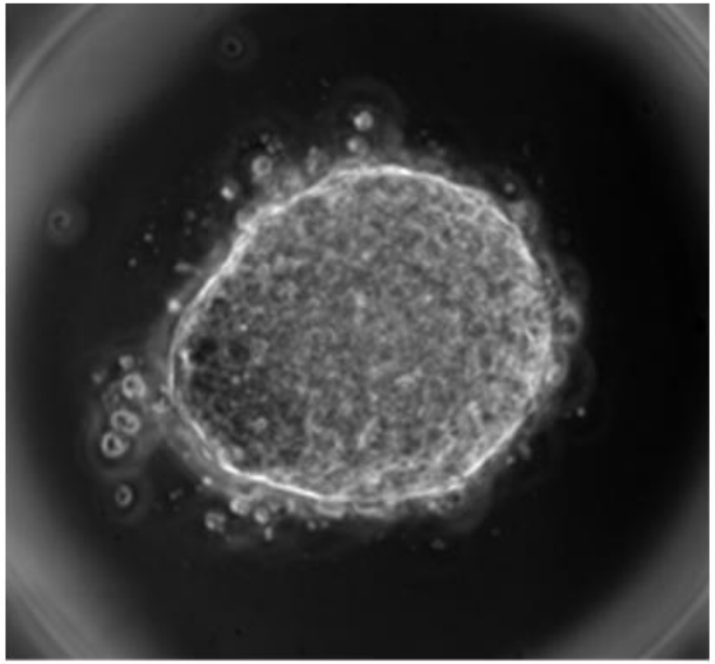
Step 1: Input Image

C = 1 for grayscale image
C= 3 for RGB image

Width

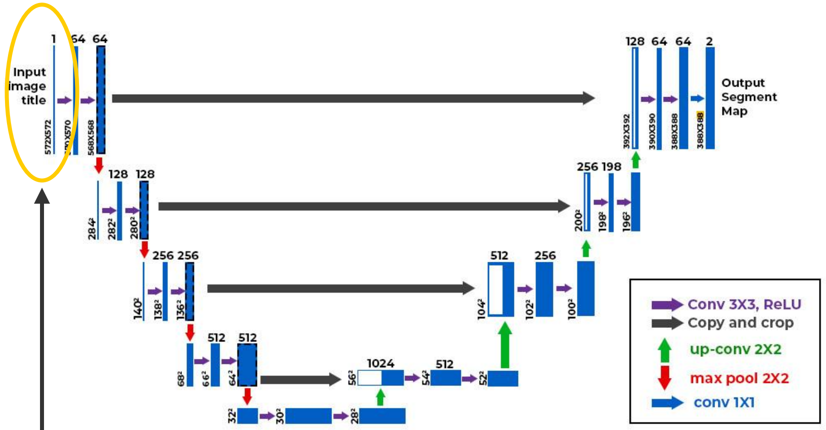
A 2D image of size H × W × C.

Height



Example: A 512 × 512 × 1

Channels



Input Image

Clarity in Complexity

4. Detailed Explanation of U-Net Architecture

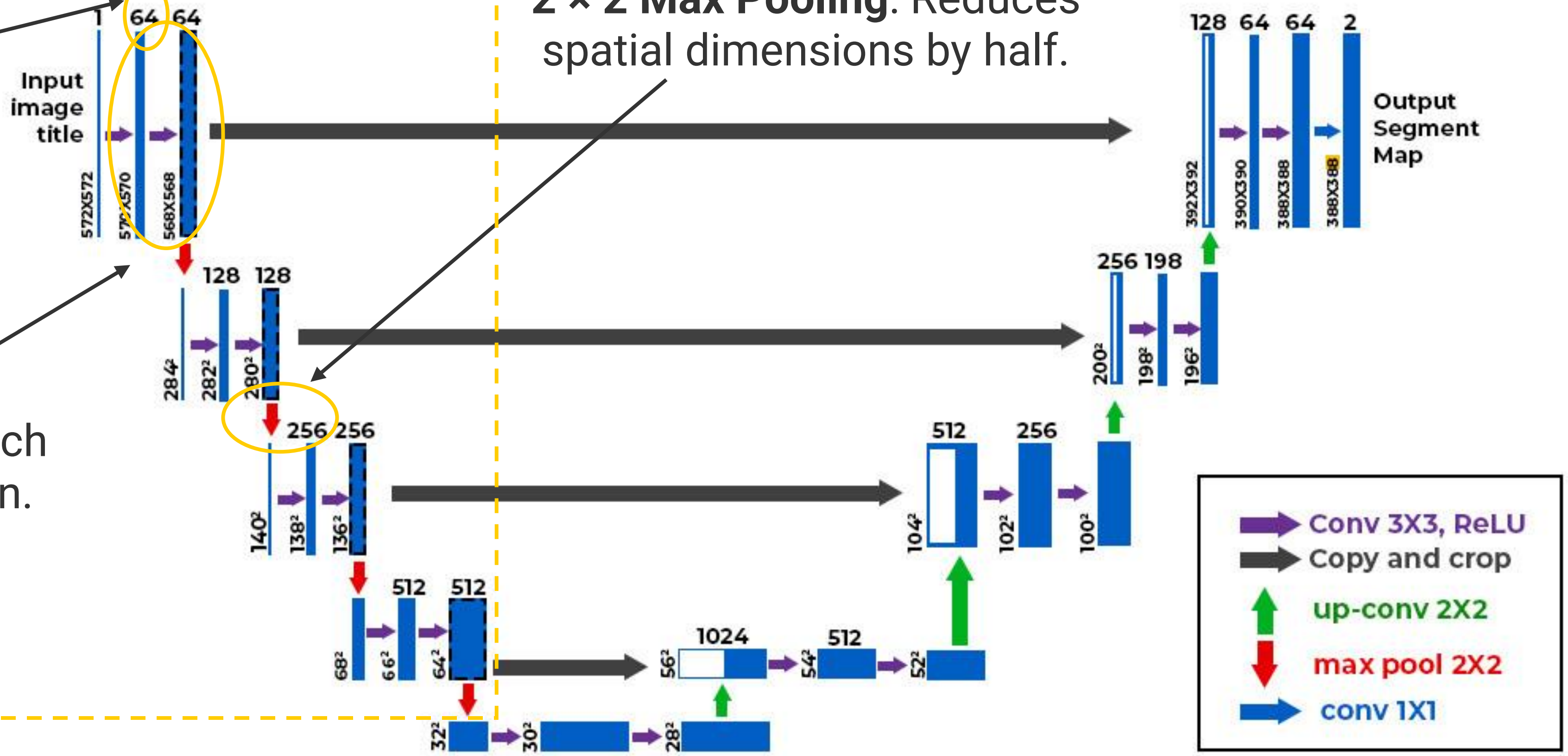
Step 2: Contracting Path (Encoder)

As we progress through the layers, the number of feature maps increases, while the spatial resolution decreases.

Feature Maps: Applying 64 filters of the same size allows learning different patterns. Increases while going deeper (64, 128, 256, etc…)

2 × 2 Max Pooling: Reduces spatial dimensions by half.

Two 3 × 3 Convolutions: Each followed by ReLU activation.

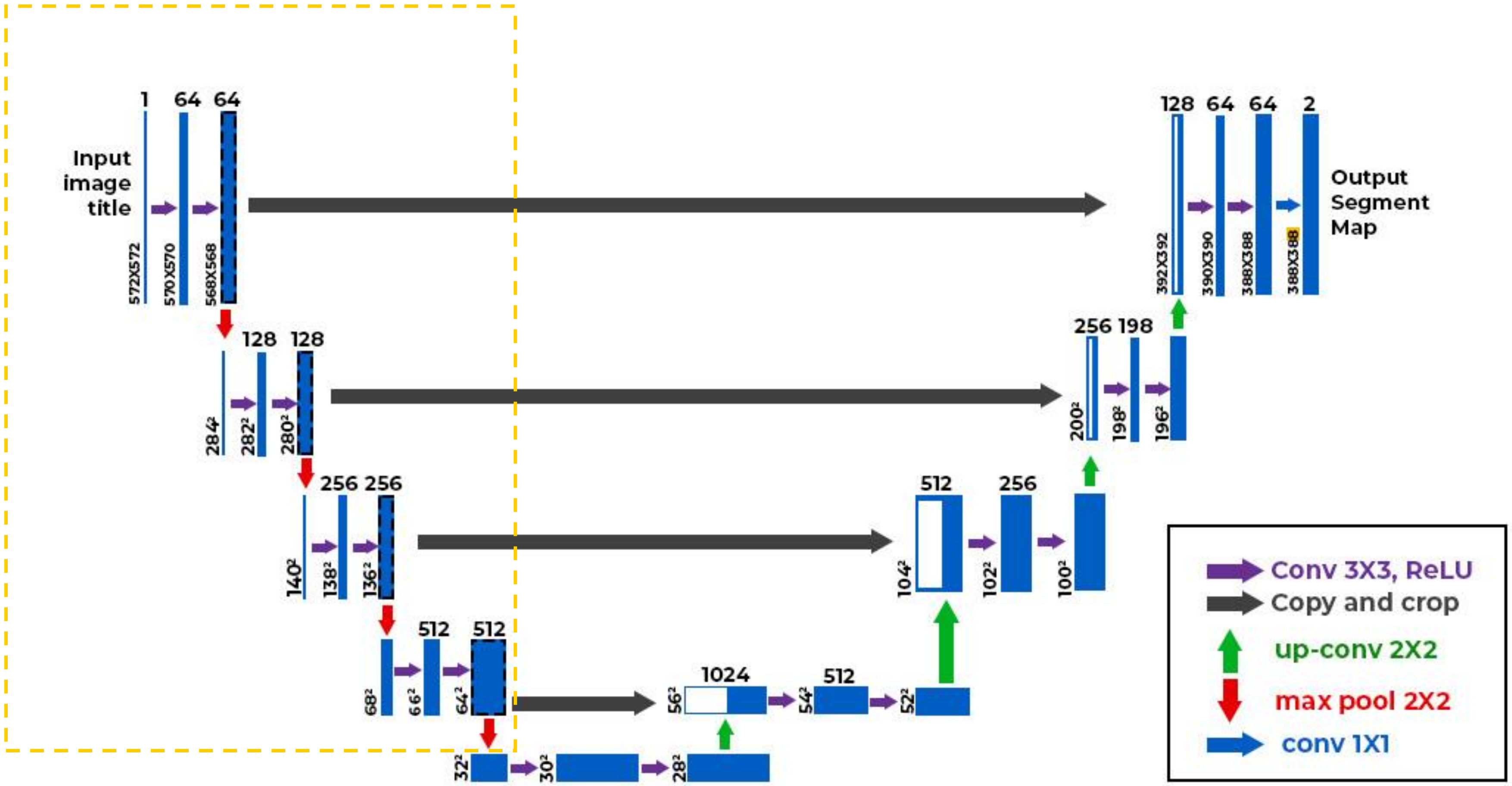


Clarity in Complexity

4. Detailed Explanation of U-Net Architecture

Step 2: Contracting Path (Encoder)

Purpose: Extract hierarchical features and capture context.

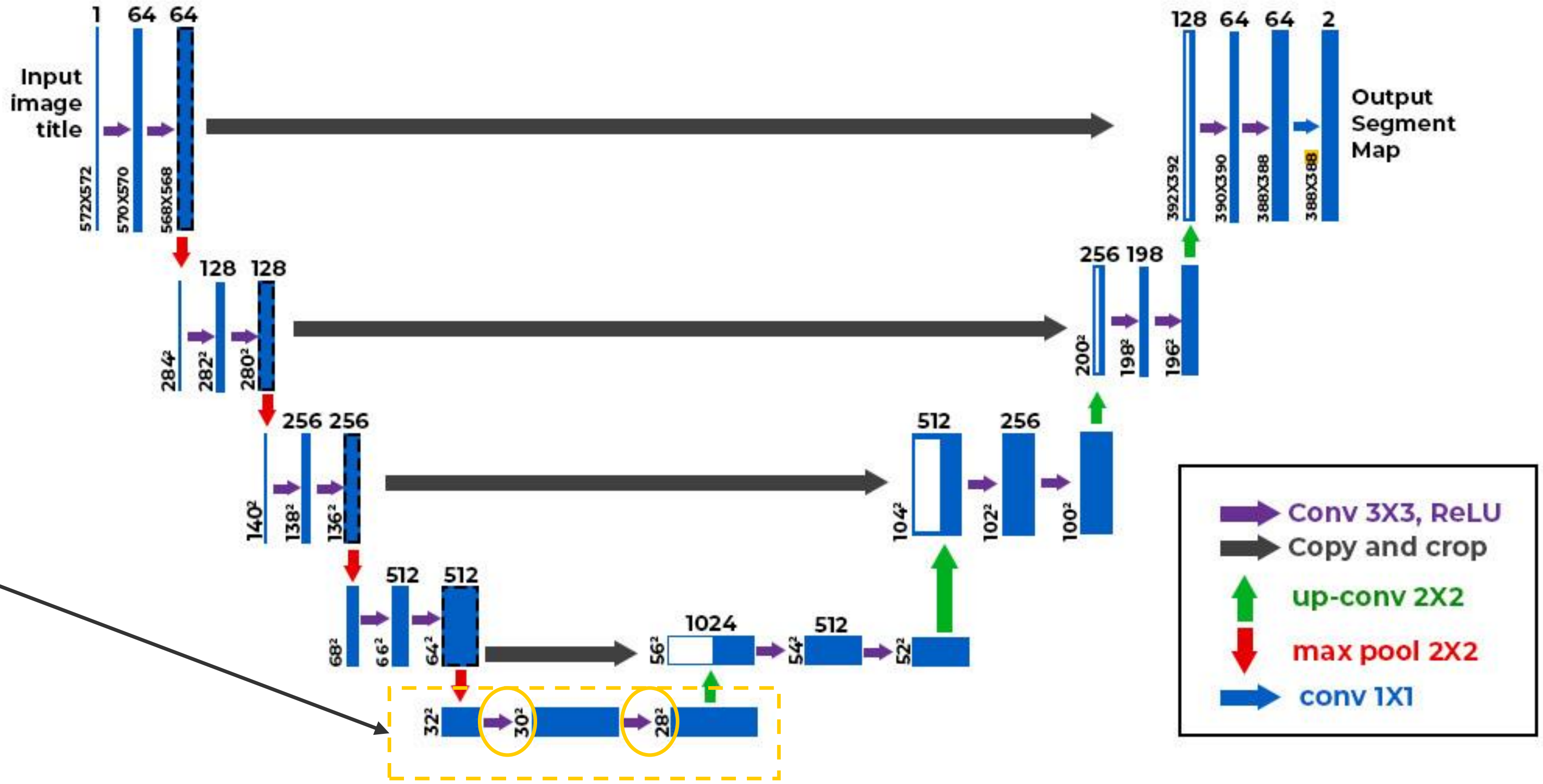


4. Detailed Explanation of U-Net Architecture

Step 3: Bottleneck

Bottleneck:
Represent the most compressed feature representation in the network.

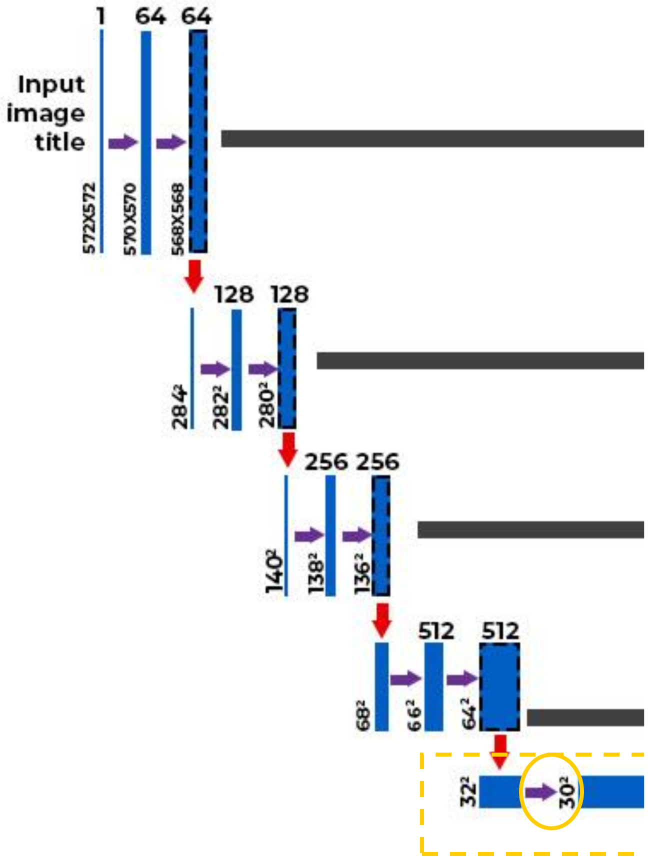
Two 3 × 3 Convolutions
: Applying two 3 × 3 convolutions filters without any pooling.



Clarity in Complexity

4. Detailed Explanation of U-Net Architecture

Step 3: Bottleneck



Purpose:

- **Captures high-level, abstract features** with minimal spatial detail, focusing on global context.
- **Provides global context** to aid in distinguishing large regions.
- **The features generated** in the bottleneck are later combined with spatial details from the encoder (via skip connections), enabling the network **to merge semantic information (from the bottleneck) with localization information** (from the encoder).
- **Reduces computational load** by compressing features, making the network more efficient.
- **Filters out redundant information**, ensuring only essential features are passed forward.

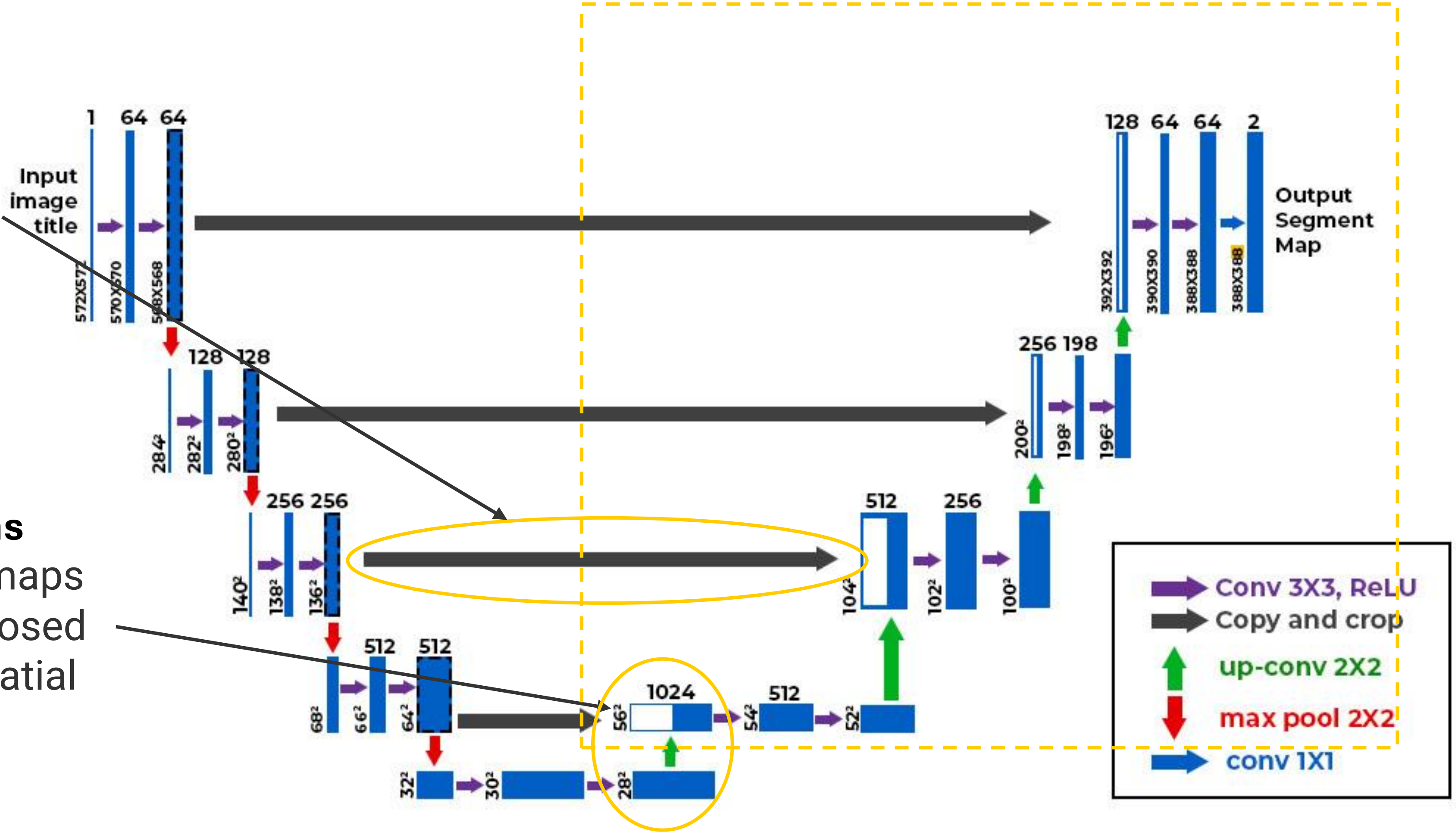
Clarity in Complexity

4. Detailed Explanation of U-Net Architecture

Step 4: Expanding Path (Decoder)

Skip connections: The upsampled feature maps are concatenated with the corresponding feature maps from the contracting path to recover fine spatial information.

Transposed Convolutions (Upsampling): The feature maps are upsampled using transposed convolutions to increase spatial resolution



Clarity in Complexity

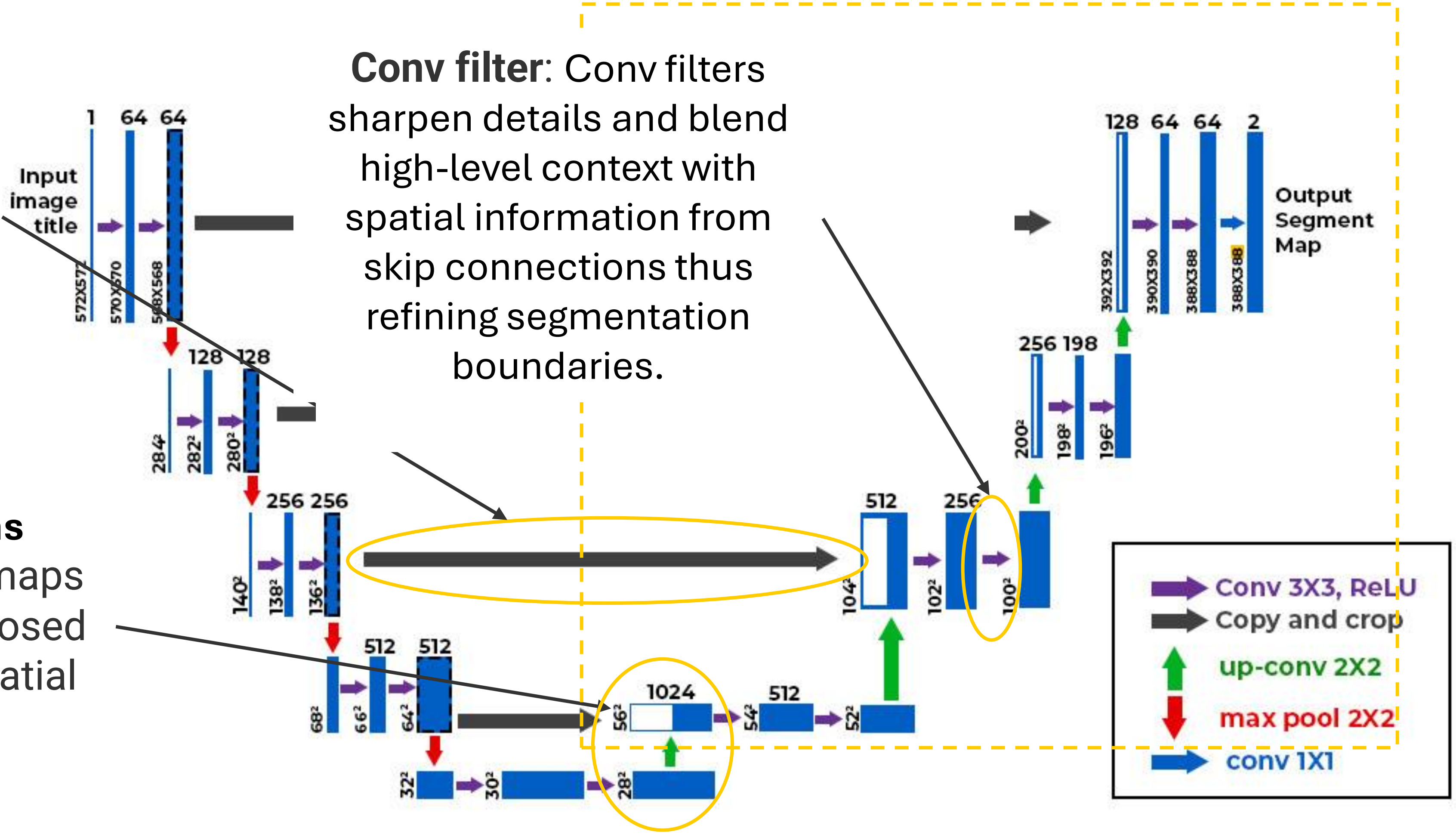
4. Detailed Explanation of U-Net Architecture

Step 4: Expanding Path (Decoder)

Skip connections: The upsampled feature maps are concatenated with the corresponding feature maps from the contracting path to recover fine spatial information.

Conv filter: Conv filters sharpen details and blend high-level context with spatial information from skip connections thus refining segmentation boundaries.

Transposed Convolutions (Upsampling): The feature maps are upsampled using transposed convolutions to increase spatial resolution

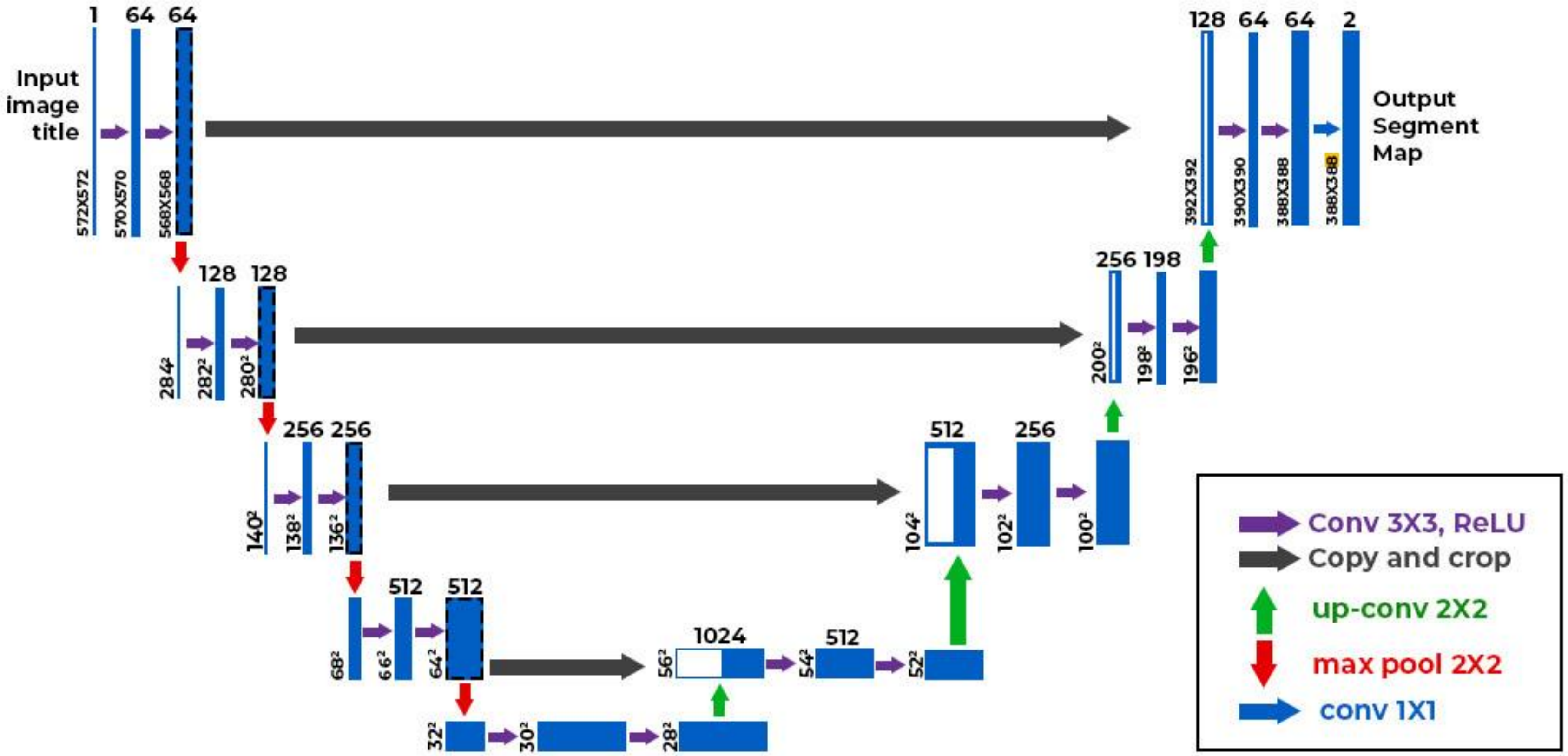


Clarity in Complexity

4. Detailed Explanation of U-Net Architecture

Step 4: Expanding Path (Decoder)

Purpose: Increase spatial dimensions until reaching original resolution

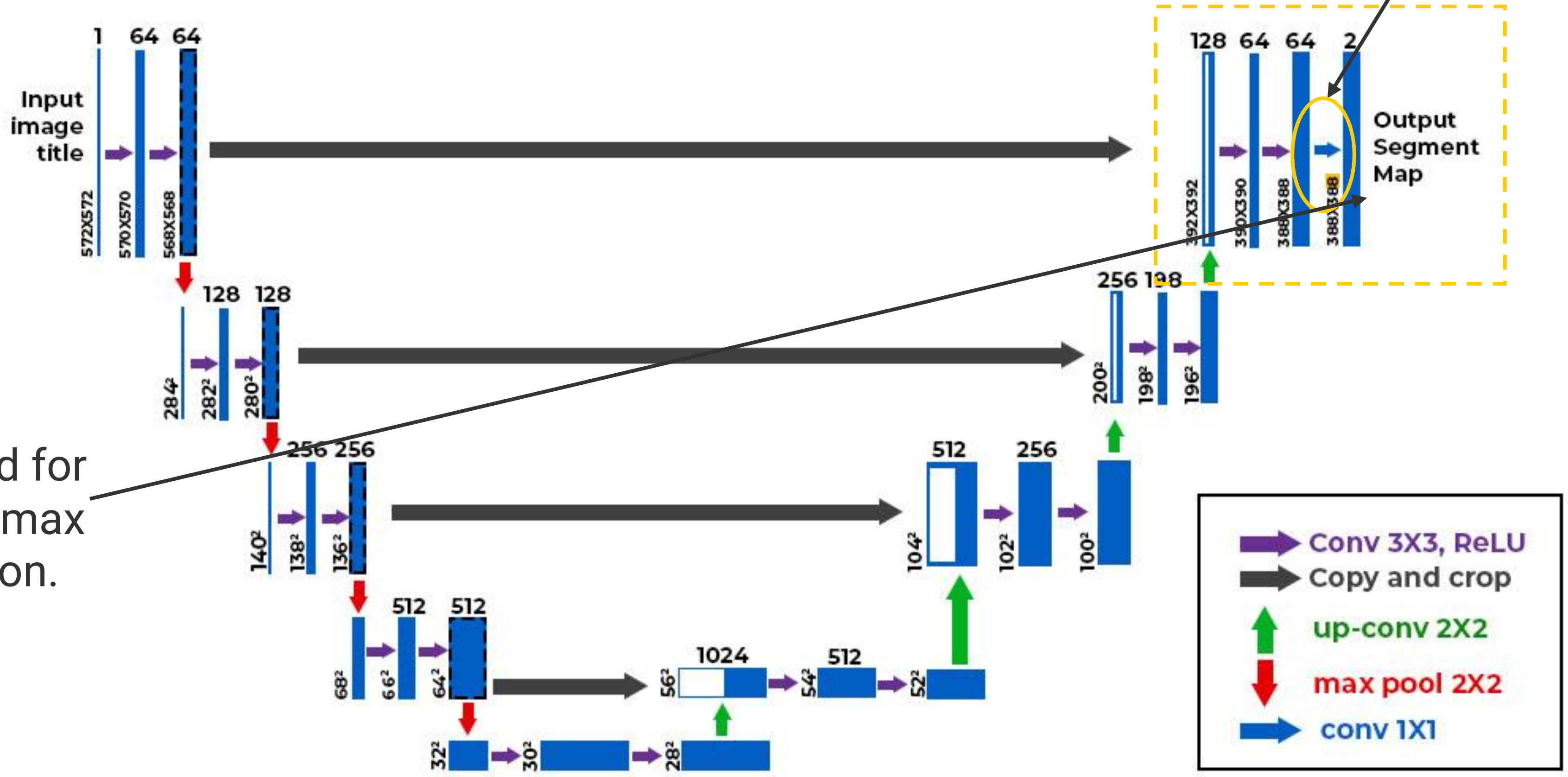


4. Detailed Explanation of U-Net Architecture

Step 5: Final Output Layer

Final output: applying **n 1 × 1** conv filter to reduces the number of channels to the desired number of classes (n = 1 for binary segmentation)

Activation function: Sigmoid for binary segmentation or softmax for multi-class segmentation.



Clarity in Complexity

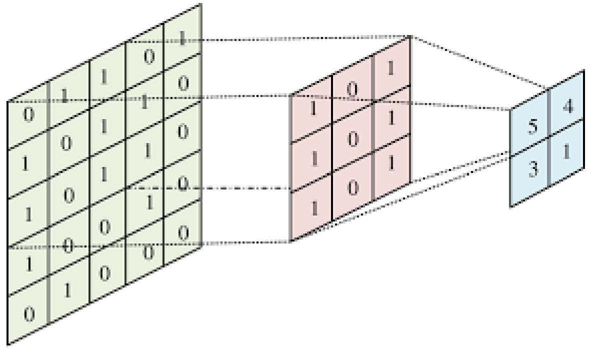
5. U-Net: Math, Metrics, and Loss Functions

Mathematical Foundation

1. **Convolution Operation**

- **Definition:** Convolution applies a small, learnable filter (kernel) across the image to extract local patterns, like edges, textures, or more complex features.

- **Purpose in U-Net**: Captures essential features of the input at each layer, enabling the network to detect patterns relevant to segmentation.



$$y_{i,j} = \sum_{p=-k}^{k} \sum_{q=-k}^{k} x_{i+p,j+q} \cdot w_{p,q} + b$$

- x Input pixel values.

- w Filter (or kernel) weights.
- b bias tem.
- y Output feature map value at position (i,j).
- k represents half the size of the filter minus one

Clarity in Complexity

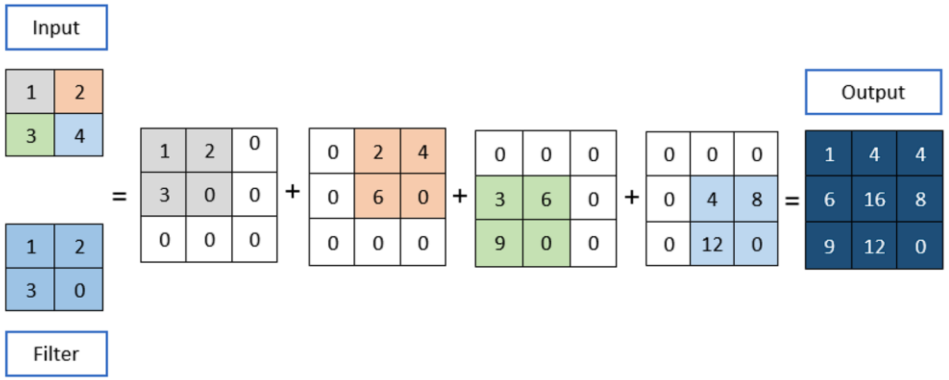
5. U-Net: Math, Metrics, and Loss Functions

Mathematical Foundation

2. Up Convolution (Transposed Convolution)

- **Definition:** An operation that increases the spatial dimensions of the input (upsampling) by applying a kernel in a way that reverses the spatial reduction effect of a regular convolution.

- **Purpose in U-Net**: Used in the expanding path to restore the original image resolution, allowing for precise spatial localization in the segmentation output.



Clarity in Complexity

5. U-Net: Math, Metrics, and Loss Functions

Mathematical Foundation

3. Calculation of output size of convolution

Transpose Convolution

$$H_{out} = 1 + \frac{H_{in} - k + 2p}{s}$$

- H_{in} : Input size
- H_{out} : Output size
- k : Kernel size
- p : Padding
- s : Stride

Transpose Convolution

$$H_{out} = (H_{in} - 1) \times s + k - 2p + \text{output\_padding}$$

- H_{in} : Input size
- H_{out} : Output size
- k : Kernel size
- p : Padding
- s : Stride
- **Output padding** is an additional term specific to transposed convolutions and allows fine control over the final output shape, often set to zero in standard cases.

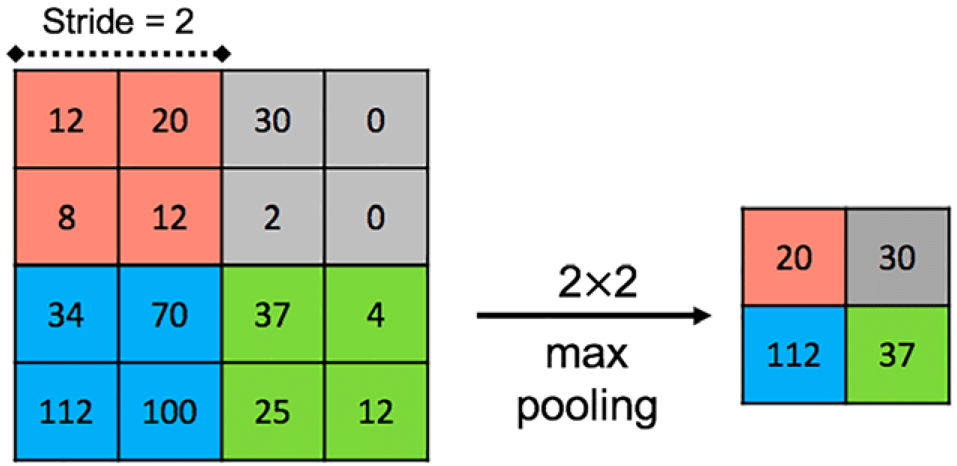
5. U-Net: Math, Metrics, and Loss Functions

Mathematical Foundation

4. Max Pooling

- **Definition:** Reduces the spatial dimensions of the input by retaining only the maximum value in each sub-region, preserving essential features while discarding less relevant data.

- **Purpose in U-Net**: Applied in the contracting path to downsample feature maps, reducing computation and helping the network focus on larger-scale features.



$$y_{i,j} = \max(x_{2i,2j}, x_{2i+1,2j}, x_{2i,2j+1}, x_{2i+1,2j+1})$$

- x Input pixel values.

- w Filter (or kernel) weights).
- b bias tem.
- y Output feature map value at position (i,j).
- k represents half the size of the filter minus one

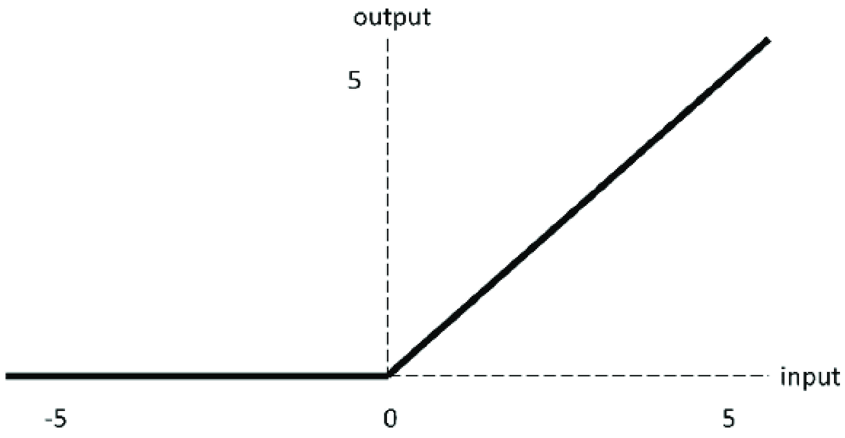
5. U-Net: Math, Metrics, and Loss Functions

Mathematical Foundation

5. ReLU(Rectified Linear Unit)

- **Definition:** An activation function that sets all negative input values to zero, introducing non-linearity while avoiding the vanishing gradient problem associated with other activation functions.

- **Purpose in U-Net**: Used after each convolution to add non-linear transformations, allowing the network to learn complex patterns without diminishing gradient flow.



$$ReLU(x) = \max(0, x)$$

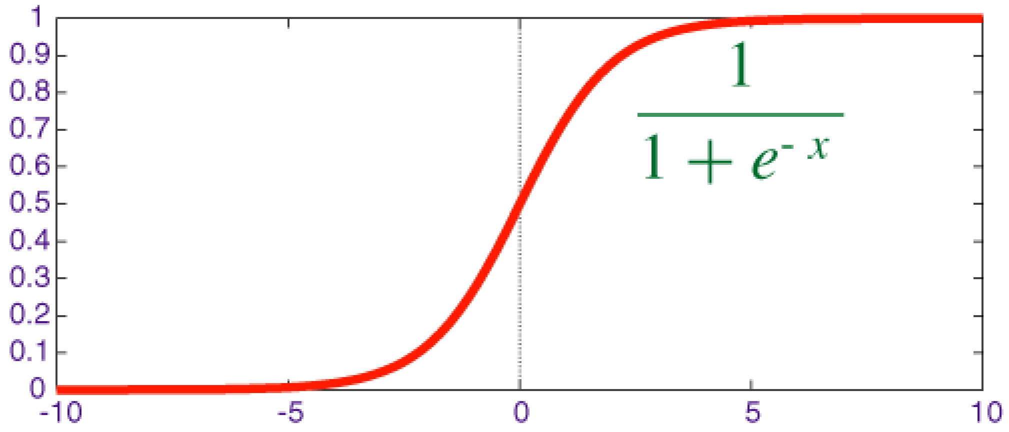
- x Input pixel values.

5. U-Net: Math, Metrics, and Loss Functions

Mathematical Foundation

6. Sigmoid Activation Function

- **Definition:** An activation function that maps input values to a range between 0 and 1, often used to represent probabilities.
- **Purpose in U-Net**: Used in the output layer for binary segmentation tasks to predict the probability of each pixel belonging to a specific class (e.g., object vs. background).



$$Sigmoid(z) = \frac{1}{1 + e^{-z}}$$

- x Input pixel values.

Clarity in Complexity

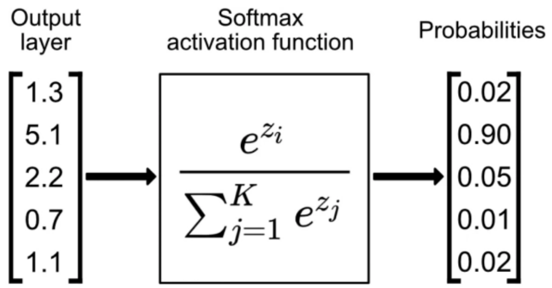
5. U-Net: Math, Metrics, and Loss Functions

Mathematical Foundation

7. Softmax Activation Function

- **Definition:** An activation function that converts a vector of scores for multiple classes into a probability distribution, where all probabilities sum to 1.

- **Purpose in U-Net**: Used in the output layer for multi-class segmentation to assign each pixel a probability for each class, enabling multi-class classification for each pixel.



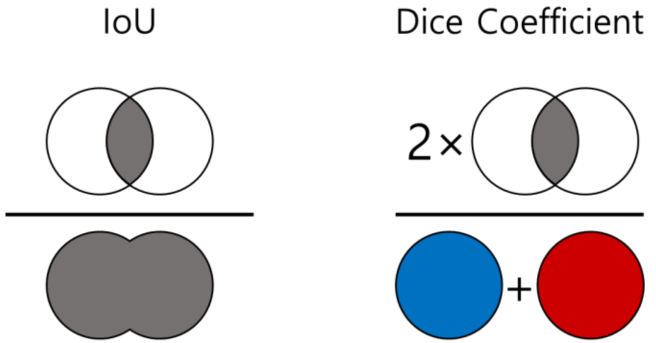
$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

- x Input pixel values.

5. U-Net: Math, Metrics, and Loss Functions

Evaluation Metrics

Metric	Pros	Cons
Dice coefficient	• Effective for imbalanced data • Sensitive to overlap	• Unstable with small regions • Less interpretable in multi-class
Intersection over union (IoU)	• Widely interpretable • Effective for overlap	• Les sensitive to boundary errors • Slower to optimize
Pixel accuracy	• Simple and intuiteve • Works well for balanced data	• Ineffective for imbalanced data • Ignores boundary precision



$$IoU = \frac{|A \cap B|}{|A \cup B|} \qquad Dice = \frac{2|A \cap B|}{|A| + |B|}$$

$$Pixel\ Accuracy = \frac{Number\ of\ correctly\ classified\ pixels}{Total\ Number\ of\ pixels}$$

5. U-Net: Math, Metrics, and Loss Functions

Loss functions

1. **Binary Cross-Entropy (BCE)**

For binary segmentation tasks

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

2. Categorical Cross-Entropy (CCE)

For multi-class segmentation tasks.

$$CCE = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$$

3. Dice Loss

Directly optimizes the Dice Coefficient.

$$Dice\ Loss = 1 - \frac{2\sum_1^n y_i\hat{y}_i}{\sum_1^n {y_i}^2 \sum_i^n \hat{y}{i}^2}$$

6. Challenges and best practices for building an effective U-Net Model

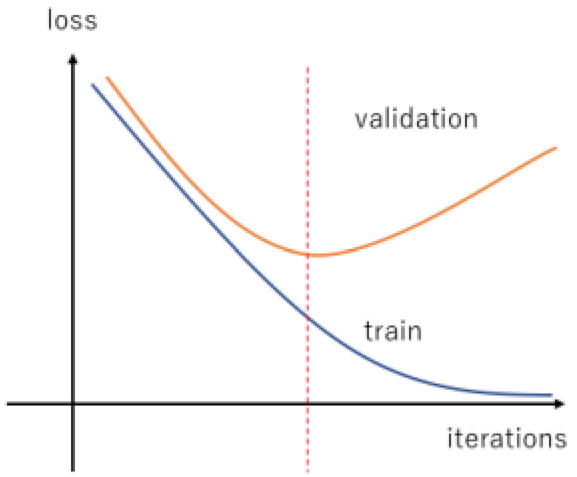
Common Issues

1. **Overfitting**

Description: Model performs well on training data but poorly on unseen data.

Causes: Limited training data, excessively complex model.

Solutions:
- Data augmentation (rotation, scaling, flipping).
- Regularization techniques (dropout, weight decay).
- Early stopping based on validation performance.



6. Challenges and best practices for building an effective U-Net Model

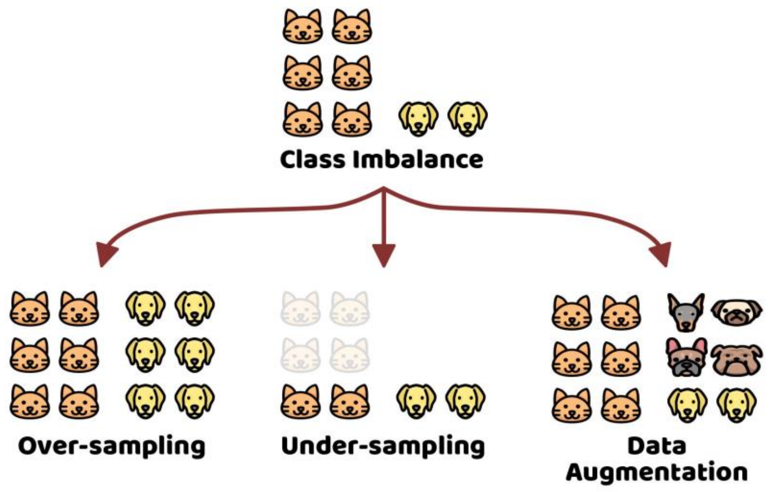
Common Issues

2. Class Imbalance

Description: Some classes (e.g., background vs. object) dominate the dataset.

Causes: Model may become biased towards majority classes.

Solutions:
- Use loss functions that handle imbalance (e.g., Dice Loss, Focal Loss).
- Apply class weighting in the loss function.
- Resample the dataset to balance classes.



6. Challenges and best practices for building an effective U-Net Model

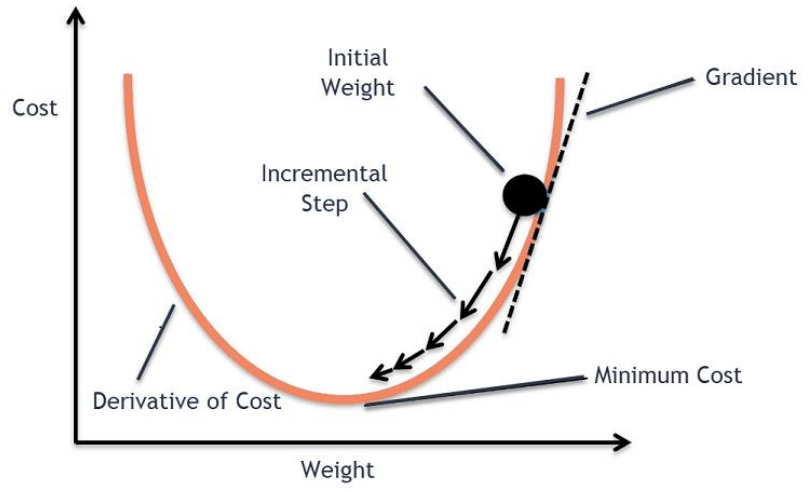
Common Issues

3. Vanishing/Exploding Gradients

Description: Gradients become too small or too large, hindering effective training.

Causes: Model may become biased towards majority classes.

Solutions:
- Use appropriate activation functions (e.g., ReLU).
- Implement batch normalization.
- Proper weight initialization (e.g., He or Xavier initialization).



6. Challenges and best practices for building an effective U-Net Model

Common Issues

4. Memory Constraints

Description: U-Net can be memory-intensive due to large feature maps and skip connections.

Causes: Model may become biased towards majority classes.

Solutions:
- Reduce input image size or use patch-based training.
- Optimize model architecture (e.g., fewer filters, smaller depth).
- Utilize gradient checkpointing or mixed precision training.

6. Challenges and best practices for building an effective U-Net Model

Common Issues

4. Difficulty in Training Stability

Description: Training may be unstable, leading to inconsistent convergence.
Causes: Model may become biased towards majority classes.

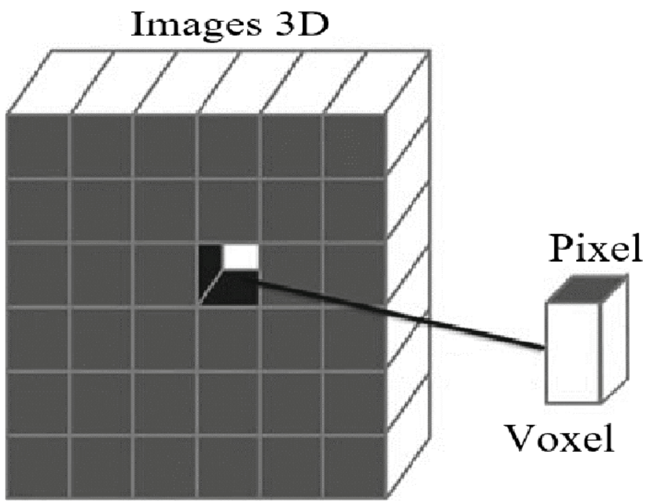
Solutions:
- Use stable optimizers (e.g., Adam, RMSprop).
- Adjust learning rates and employ learning rate schedulers.
- Normalize input data appropriately.

7. Extension to 3D U-Net

What is a Voxel?

Definition:

- A voxel is the 3D counterpart of a pixel.
- It represents a small, cube-shaped unit of volume in a 3D space, just like a pixel represents a unit of area in a 2D image.



Key Points:

- **Data Representation**: Each voxel contains data about the 3D space it occupies, such as intensity, density, or color.
- **Examples of Use**: Medical imaging (e.g., CT or MRI scans) and 3D modeling (e.g., game development, simulations).
- **Purpose in Deep Learning**: In 3D UNet, voxels represent 3D inputs (e.g., volumetric images or videos), allowing the network to process spatial relationships in all three dimensions.

7. Extension to 3D U-Net

From 2D U-Net to 3D U-Net

Key Extensions:

1. Input:

- 2D UNet takes 2D images (pixels as input).
- 3D UNet takes volumetric data (voxels as input).

2. Convolutional Kernels:

- 2D UNet uses 2D filters (height × width).
- 3D UNet extends these to 3D filters (height × width × depth), enabling spatial feature extraction in all dimensions.

3. Pooling and Upsampling:

- Adapted to operate in 3D, preserving the volumetric structure throughout downsampling and upsampling.

Purpose of the Extension:

- To process 3D data natively without flattening it into 2D slices, preserving crucial spatial information.

7. Key Concepts in 3D U-Net

Advantages of 3D UNet:

- **Spatial Context:** Explores relationships in all three dimensions, critical for accurate segmentation of volumetric data.
- **High Accuracy in Volumetric Tasks:** Perfect for medical imaging, 3D object recognition, and video processing.

Challenges:

- **Increased Computational Cost:** Larger kernels and more data lead to higher memory and processing requirements.
- **Need for 3D-Labeled Data:** Training requires annotated volumetric datasets.

Summary:

The 3D UNet builds on the 2D architecture, extending it to analyze and process 3D data effectively, unlocking powerful applications in fields requiring volumetric data understanding.

References

- Ronneberger, O., et al. (2015). U-net: Convolutional networks for biomedical image segmentation.
- Çiçek, Ö, et al. (2016). 3D U-Net: learning dense volumetric segmentation from sparse annotation.

Thank you for your attention!

Arthur Cartel Foahom Gouabou, PhD | https://cartelgouabou.github.io/

Clarity in Complexity