Image Classification And Segmentation Using Deep Learning

A Report on Summer Internship

(9th June – 15th July 2025)

submitted by

Anjan Pal

Roll No.:22101104030

Department of Computer Science and Engineering

Jalpaiguri Government Engineering College

under the supervision of

Dr. Subrata Dutta

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR

July 7, 2025

CERTIFICATE

It is certified that the work contained in this project report entitled "***Image Classification and Segmentation***", by **Anjan Pal (22101104030)**, Department of Computer Science And Engineering of the Jalpaiguri Government Engineering College, is submitted for the Summer Internship from 9th June to 15th July 2025 carried out under the supervision of **Dr. Subrata Dutta** in the Department of Computer Science And Engineering of NATIONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR.

Anjan Pal 7/7/25

Anjan Pal (22101104030)
Department ofComputer Science And Engineering,
Jalpaiguri Government Engineering College

Subrata Dutta 7/7/25

Dr. Subrata Dutta
Supervisor
Department of Computer Science and Engineering
National Institute of Technology Jamshedpur

7.7.2025

Dr.Danish Ali Khan
Head of the Department
Department of Computer Science and Engineering
National Institute of Technology Jamshedpur

Head / विभागाध्यक्ष
Dept. of CSE / संगणक विज्ञान एवं अभियंत्रण विभाग
NIT / राष्ट्रीय प्रौद्योगिकी संस्थान
Jamshedpur / जमशेदपुर

i

ACKNOWLEDGMENT

I would like to express our special thanks of gratitude to our Supervisor Dr. Subrata Dutta as well as our Director Prof. Goutam Sutradhar who gave us the golden opportunity to do this wonderful project on the topic mage Classification and Segmentation, which also helped us in doing a lot of research and I came to know about so many new things. Secondly, we also would like to thank our parents and friends who helped us greatly in finishing this project within the limited time.

Anjan Pal (22101104030)

Department of Compuer Science And Engineering,

Jalpaiguri Government Engineering College

ABSTRACT

Image segmentation is nothing but training computers to learn what is in an image by decomposing it into different parts or regions—distinguishing buildings and sky or organs in a medical image, for instance. Methods over the years have evolved from simple ones using thresholding and edge detection to complex deep learning models able to learn from data and make very accurate predictions. One such model named the U-Net has gained much fame in applications involving medical image applications. Nevertheless, challenges remain like how to deal with inconsistent lighting, object shapes, and how to generate voluminous datasets to be used for training or labeling. In this paper, we explain how image segmentation occurs, why it is crucial, and how it is used in practical applications like medicine, autonomous vehicles, and satellite photography. Being able to clearly see and differentiate individual components of a cell, such as mitochondria and cancer cells, is extremely useful in disease detection and research. For our task, we sought to instruct a computer to do so using image segmentation. Two deep neural models were utilized: U-Net, notoriously well-established for handling medical images, and an upgraded version of itself called Attention Residual U-Net. The second model provides for weights to be used in addition to cause the system to focus on important areas in an image. Both of these resources aided in dealing with common obstacles, such as blurry images, overlapping cells, and unconventional cell shapes. Overall, our solution made it easier to clearly light and differentiate mitochondria and cancer cells and is potentially useful for easier disease detection and comprehension.

Contents

List of Figures

List of Tables

1 Introduction

Cell imaging is central to understanding about diseases and cancer in biomed applications. One of the biggest challenges is to identify specific structures such as mitochondria accurately. One of the biggest challenges is also how to distinguish between cancer cells while working with image data. Automatic annotation is really prone to errors, it is also lengthy and not scalable—especially when working with data acquired using high-resolution microscopes. That is where image sections are used.

Image segmentation is a process by which computer vision separates an image into salient parts. For our biomedical images, it offers researchers a method of emphasizing specific cell types or even organelles. This then allows for a better analysis and an interpretation.

For our project, we are particularly interested in mitochondria and also objects segmentation from micrographs because we are making use of deep models of learning. There are two models: the U-Net, which is a popular and particularly for biomedical image segmentation designed architecture and Attention Residual U-Net which is an improvement of it with attention mechanisms and thus capable of better focusing on salient regions of the image.

We are to develop an efficient correct pipeline for segmentation able to handle sophisticated textures of irregular form and overlapping settings in real biological data. By using this approach our target is to provide further scalable automated solutions for research on cellular cancer.

2 Motivation

- Improved Understanding of Diseases: By accurate segmentation of cancer cells and mitochondria, researchers can obtain a better understanding of cellular behavior, track disease progression, and evaluate drug efficacy.
- Burden of Manual Annotation: Annotation of cell structures observed microscopically is not only time-consuming but also prone to human errors and arbitrary, leading to ineffectiveness of the process.

- Biomedical Imaging Automation Need: Biomedical imaging automation can greatly speed up research and improve diagnosis protocols in cell biology and pathology and other related fields.

- Biological Significance of Mitochondria and Cancer Cells: Mitochondria and cancer cells are both critical to health studies—mitochondria are essential to metabolic functions, and cancer cells are of central importance to tumor growth and metastasis.

- Problems with Current Image Processing: Regular image processing protocols often cannot accommodate variations in cell shape, cell size, intensity, and overlapping of components of cells in microscopy and generate results of lower reliability.

- Exploiting Deep Learning for Medical Imaging: Deep models and particularly models of U-Net variants were found to be exceptional in correctly segmenting efficiently and effectively challenging biomedical images.

- Contribution of Attention Mechanisms: Attention ResUNet architecture improves segmentation accuracy by focusing attention on image regions most relevant for each pixel and boosting final accuracy in general.

3 Current Status

- **Overview of Project Progress**

  Following a meticulous examination of recent literature and authorities' opinions, approximately 70–75% of the project is completed.

- **Completed Tasks:**

  - **Literature Review and Background Study:** Literature Review and Background Study: We conducted a wide literature review of adjacent studies to create a foundation for understanding both a particular sphere of challenges and existing solutions.

  - **Dataset Acquisition:** Microscopic image data of mitochondria and cancer cells were successfully obtained and curated.

  - **Data Preprocessing:** Major preprocessing steps were also applied, such as image normalisation, resizing and greyscale transform. Augmentation processes such as rotation and flip were also utilised to increase robustness of a model and prevent overfitting.

- **Model Development:** The classical U-Net architecture was also implemented using TensorFlow/Keras and learned on the dataset prepared. Apart from this, an Attention Residual U-Net (Attention ResUNet) was constructed by adding attention mechanisms and residual blocks to further equip the model to pay attention to fine boundaries and complex structures.
- **First Training and Validation:** Both models are pretested and validated using a training-validation split. Key performance measures—in particular Dice Coefficient, Intersection over Union (IoU), and binary accuracy—are tracked. Initial results indicate superior segmentation performance by the Attention ResUNet model, which is specific to delineating fine edges and overlapping regions.
- **Visualization of Results:** Segmentation masks generated by models are also visually compared with ground truth labels and contain intuitive information on each model's segmentation accuracy.

- **Current Projects:**
  - **Hyperparameter Optimization:** Optimization of essential parameters like learning rate, batch size, and filter numbers is also undertaken to further optimize model performance.
  - **Overall Assessment:** Currently, the models are applied to an unseen independent test set on which they were not trained. Simultaneously, quantitative measures are tabulated so that thorough comparison can be facilitated.
  - **Report Preparation:** Documenting process has also started with putting some portions of the final report in writing though yet to be finished.

- **Remaining Work:**
  - **Overall Performance Analysis:** There should be a clear summary of results of U-Net and Attention ResUNET in comparison—through tables, charts, and graphs—which should be completed.

- **Conclusion and Discussion:** Presentation of results and outcome of results will be made in report form including thorough analysis and recommendations.
- **Finalization and Submission of Report:** Final steps include proofreading, formatting, and assembling final report for formal submission.

4 Objectives of the internship project

Our internship project is focused on advancing biomedical image analysis with some exciting goals. Here's what we're aiming to achieve:

1. **Build Smart Segmentation Tools**: We aim to create a system that can automatically identify mitochondria and cancerous cells in microscope images using cutting-edge deep learning. Think of it as teaching a computer to be a super-smart lab assistant!

2. **Start with a Solid Foundation**: We'll begin by building and testing the classic U-Net model as our starting point for analyzing detailed microscope images. It's like laying a strong foundation for a house before adding fancy upgrades.

3. **Level Up with a High-Tech Model**: We're taking things up a notch by designing an enhanced version called Attention ResUNet. This model will include:
   - **Attention gates** to help the system focus on the most important features.
   - **Residual connections** to make our network deeper and smarter, like giving it extra brainpower for tough tasks.

4. **Get Images Ready for Analysis**: We'll prepare the images to ensure they're in top shape for our models by:
   - Adjusting their sizes to be consistent.
   - Using techniques like rotating and flipping them to create more training examples.

5. **Measure How Well It Works**: We'll evaluate our models by comparing them with key metrics, such as:
   - **Dice score** (a measure of overlap accuracy).

- **Pixel-wise IoU** (checking how well predictions match reality).
- **Segmentation accuracy** and **precision/recall** to see the full picture of performance.

6. **Tackle Tricky Challenges**: Microscopy images can be complex, so we'll work on solving issues like:
   - Overlapping cells that confuse the system.
   - Variations in cell size and shape.
   - Background noise or artifacts that create distractions.
7. **Show Off the Results**: We'll create clear, easy-to-understand visuals to compare our model's predictions with the actual ground truth, making it simple to see how well we're doing.
8. **Make a Real Impact**: Our work aims to support medical breakthroughs by contributing to tools that help with:
   - Pathology examinations.
   - Cancer research.
   - Studies in cellular biology.

In short, we're combining the latest deep learning techniques to revolutionize how microscope images are analyzed, all while keeping it practical for real-world medical use. It's a big challenge, but we're excited to make a difference!

5 Learning objectives

1. **Dive into Deep Learning in Action**
   Get hands-on experience with powerful deep learning models like U-Net and Attention ResUNet. You'll learn how to train these models to identify important features—like mitochondria and cancer cells—in microscope images. It's a great way to build real-world skills in computer vision and biomedical AI.

2. **Master the Art of Data Preparation**

   Discover how to prep microscopy images for deep learning like a pro. You'll explore essential techniques like image normalization and data augmentation (think rotating and flipping) to help your models perform more accurately and consistently.

3. **Learn How to Measure What Matters**

   Not sure how to tell if your model is doing a good job? You'll learn to use key metrics like the Dice coefficient and IoU (Intersection over Union) to evaluate model performance, and make sense of the results in a meaningful way.

4. **Understand What Makes Models Smarter**

   Explore the power of advanced deep learning features—like attention gates that help the model focus on the right areas, and residual connections that improve training. These tools are game-changers for building models that can handle complex biomedical images.

5. **Get Confident Comparing AI Models**

   You won't just build models—you'll learn how to compare them. You'll develop the critical thinking needed to decide which model architecture fits best for a given task, so you can confidently choose the right approach in future projects.

6. **Sharpen Your Science Communication Skills**

   Technical work means nothing if you can't explain it clearly. You'll practice turning complex results into simple, insightful visualizations and reports that others—whether researchers or clinicians—can easily understand.

7. **Level Up Your Python and AI Toolkit**

   Boost your programming chops with hands-on work in Python using tools like TensorFlow/Keras, OpenCV, and NumPy. These are the building blocks of modern AI workflows, and you'll become more comfortable using them to solve real problems.

8. **See the Real-World Impact of Your Work**

   This isn't just theory—your project has the potential to help improve healthcare. From supporting cancer diagnosis to advancing cellular research, you'll see how AI can make a real difference in the lives of patients and scientists alike.

6 Understanding AI, ML, and DL

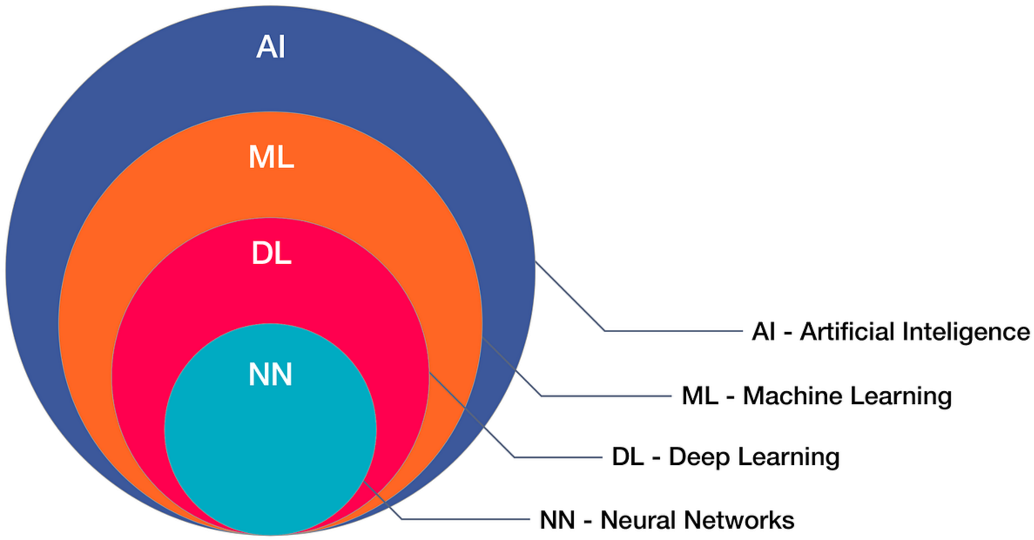

Figure 1: Conceptual Segmentation of AI, ML, and DL

In the world of technology, we often hear about three key terms: Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL). Let's break them down:

- **Artificial Intelligence (AI):** This is the big umbrella that covers everything related to creating smart machines. These machines can perform tasks that usually require human intelligence, like understanding language or recognizing images.
- **Machine Learning (ML):** Think of this as a subset of AI. ML involves using algorithms that enable machines to learn from data. Instead of being programmed for every single task, these machines improve their performance based on experience.
- **Deep Learning (DL):** Now, DL is a more specialized area within ML. It uses complex structures called artificial neural networks, which have multiple layers. This allows DL to tackle intricate patterns in large amounts of unstructured data, like images or audio.

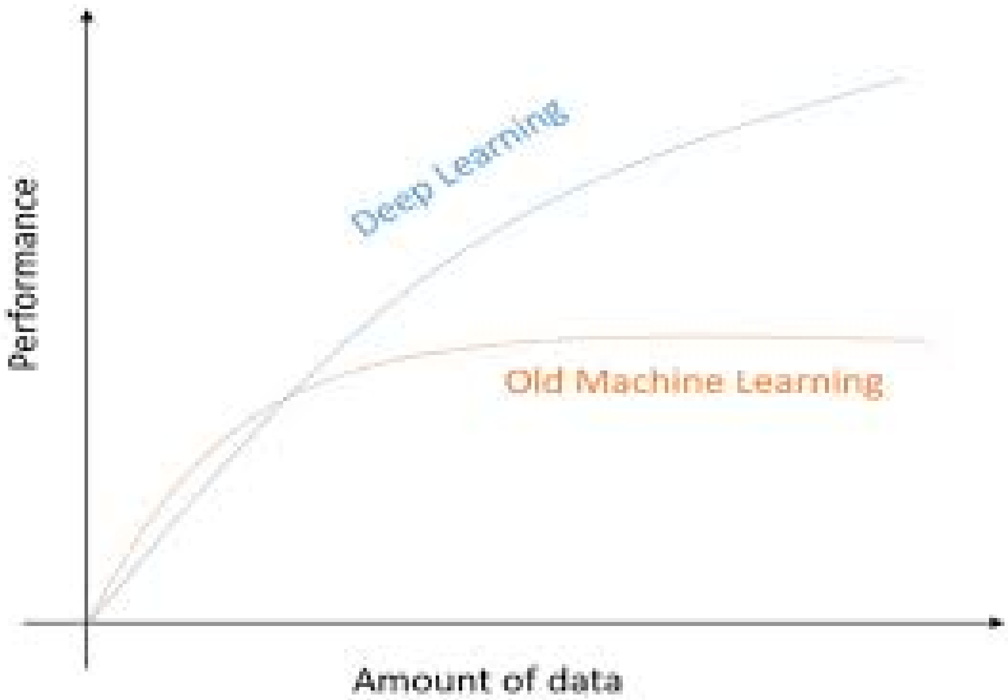

Figure 2: Hierarchical Relationship of AI, ML, and DL

When it comes to working with structured data (like spreadsheets), traditional **Machine Learning (ML)** models are often the go-to choice. They tend to be more efficient and accurate, mainly because they require less computational power and can be trained faster.

On the other hand, if you're dealing with large and unstructured datasets—think images, audio files, or natural language—**Deep Learning (DL)** becomes essential. While DL algorithms are incredibly powerful and can handle complex data, they also demand a lot of computational resources and time.

So, how do you decide whether to use ML or DL? Here are some factors to consider:

- The type of data you have (is it structured or unstructured?)
- The size of your dataset
- The computational resources you have available
- The balance you want to strike between accuracy and efficiency

Now, lest try to understand the Deep Learning Model.

6.1 Deep Learning Model

Deep Learning is a subset of Machine Learning that uses multi-layered artificial neural networks to automatically learn hierarchical representations of data. These models are particularly powerful for processing unstructured data such as images, audio, and text.

A typical deep learning model consists of:

- **Input Layer:** Receives the raw input data (e.g., pixels from an image).
- **Hidden Layers:** One or more layers of neurons that perform nonlinear transformations and learn abstract features. In Convolutional Neural Networks (CNNs), these may include convolutional and pooling layers.
- **Output Layer:** Produces the final prediction or classification result.

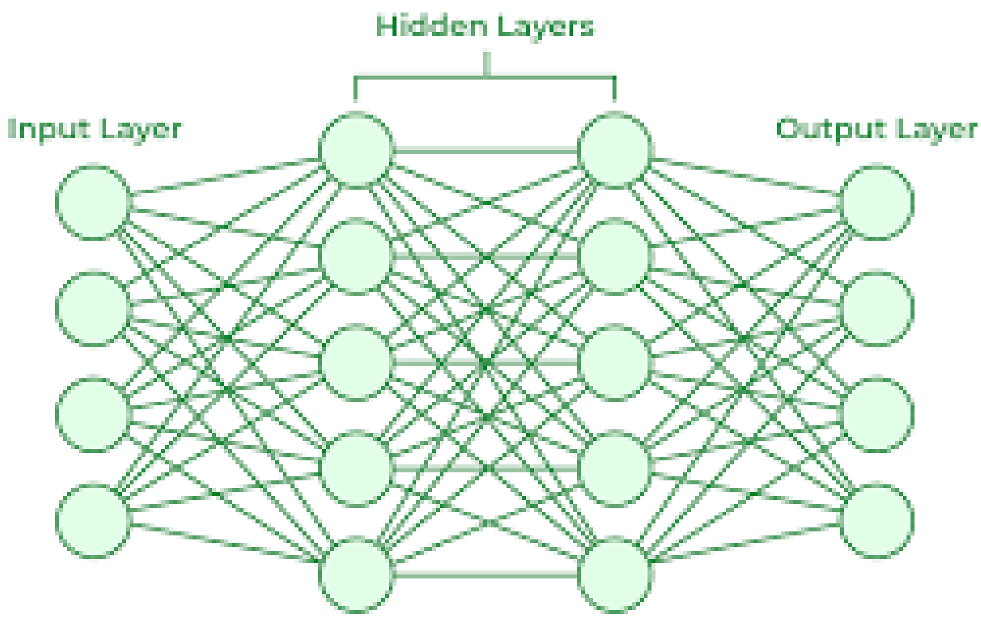

Figure 3: Basic Architecture of a Deep Learning Model

Deep learning models require large volumes of data and high computational power. They are trained using optimization techniques like backpropagation and gradient descent, and commonly rely on GPUs or TPUs for efficient training.

6.2 Neural Networks and Their Types

Types of Neural Networks

1. **Feedforward Neural Network (FNN):**
    - Information moves in one direction from input to output.
    - Example: Multilayer Perceptron (MLP).
2. **Convolutional Neural Network (CNN):**
    - Specialized for image data.
    - Includes convolutional and pooling layers.
    - Used in image classification, object detection, etc.
    - *Example: LeNet-5, ResNet*

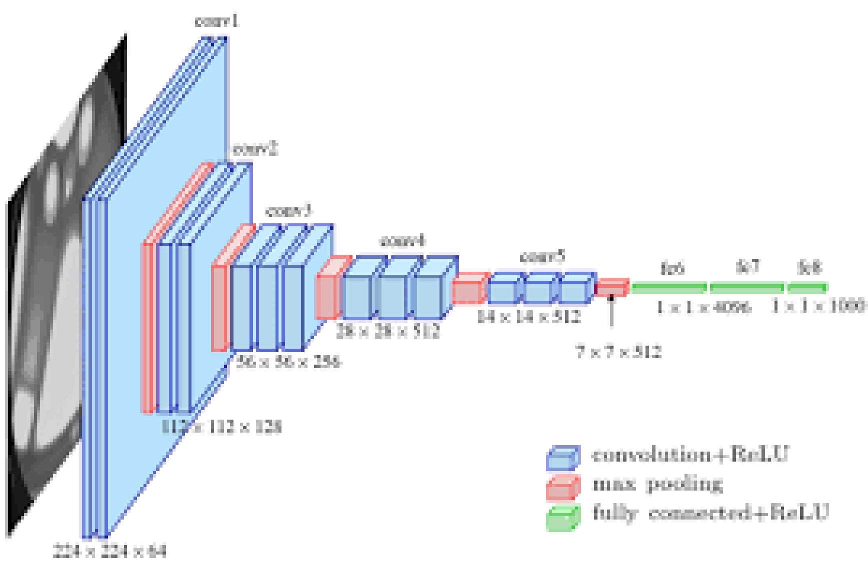

Figure 4: Typical CNN Architecture

3. **Recurrent Neural Network (RNN):**
    - Designed for sequential data like text or time series.
    - Uses feedback loops to maintain memory.
    - *Example: LSTM, GRU*

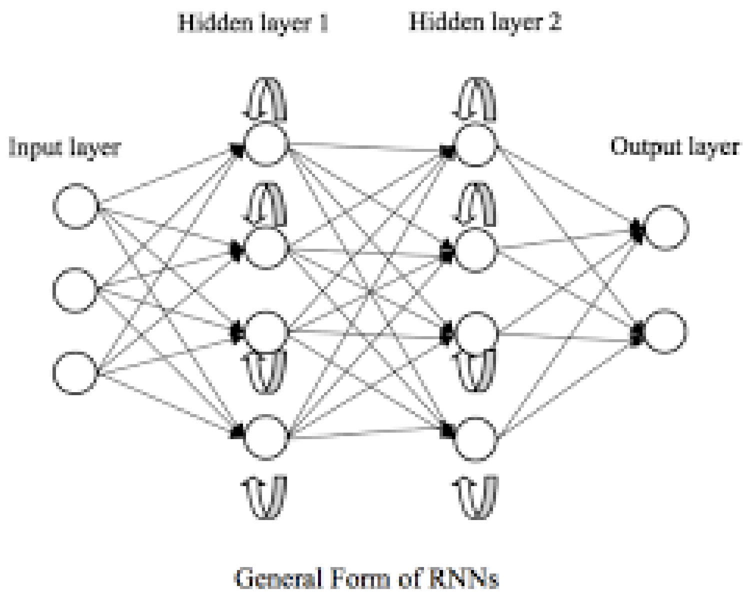

Figure 5: Recurrent Neural Network Structure

4. **Generative Adversarial Network (GAN):**
    - Consists of two networks: a generator and a discriminator.
    - Used to generate realistic data like images and audio.
5. **Autoencoder:**
    - Used for unsupervised learning and dimensionality reduction.
    - Consists of encoder and decoder components.

Applications of Neural Networks

- Image and speech recognition
- Natural language processing (e.g., sentiment analysis, translation)
- Fraud detection
- Medical diagnosis
- Recommendation systems

6.3 Core Architectures of Deep Learning

Deep Learning includes a variety of neural network architectures, but two of the most fundamental and widely used are:

1. **Artificial Neural Networks (ANN):** Think of ANNs as the building blocks of neural networks. They consist of interconnected nodes, or neurons, organized into layers: an input layer, one or more hidden layers, and an output layer. Each connection between neurons has a weight, and during training, these weights are adjusted to help the network learn. ANNs are great for handling structured data and are often used for straightforward classification and regression tasks.

2. **Convolutional Neural Networks (CNN):** CNNs are a specialized type of neural network designed specifically for processing grid-like data, such as images. They utilize convolutional layers to automatically learn and recognize spatial hierarchies of features, making them incredibly efficient. CNNs have transformed the field of computer vision and are commonly used for tasks like image classification, object detection, and segmentation.

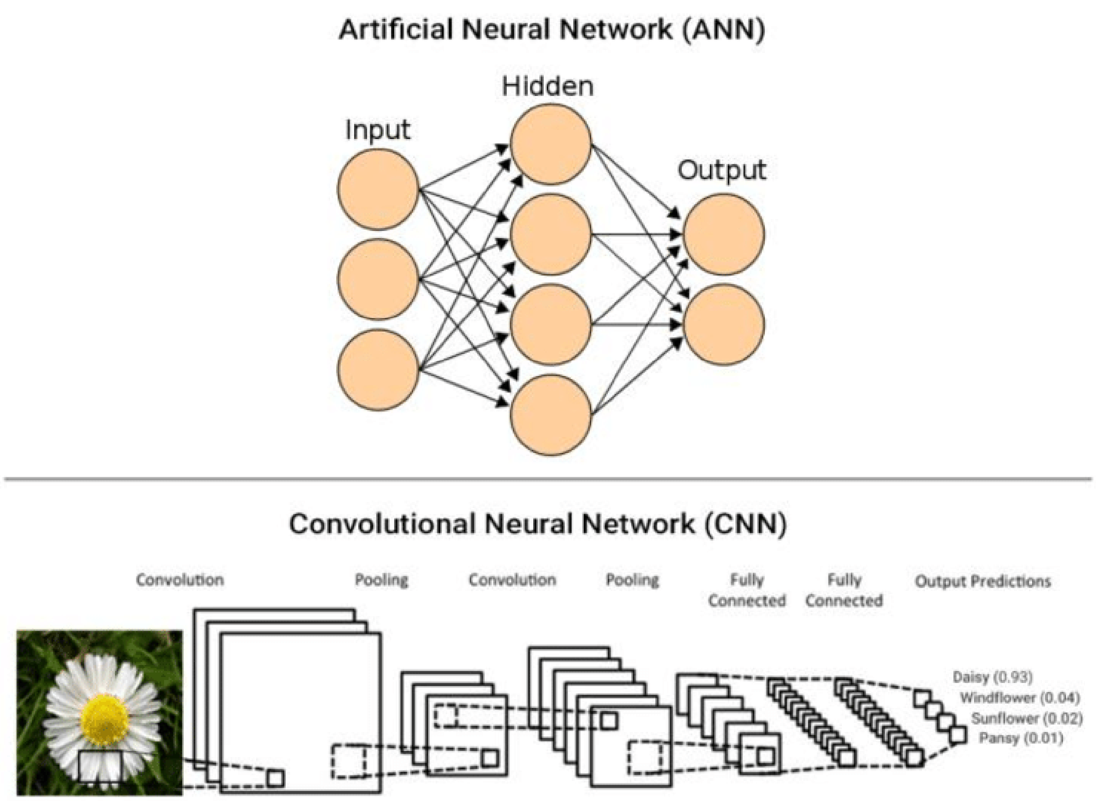

Figure 6: Comparison between ANN and CNN in Deep Learning

While ANNs serve as the foundation for many deep learning models, CNNs have truly revolutionized how machines understand and interpret visual data. Both architectures are essential and play significant roles in various applications, each suited to different types of tasks.

6.4 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a specialized type of deep learning model that excel in tasks related to image processing and computer vision. Unlike traditional neural networks, CNNs leverage a mathematical operation known as convolution, which enables them to learn spatial hierarchies of features from the input data.

CNN Architecture

A typical CNN is composed of several key layers:

- **Input Layer:** This layer takes in the raw pixel values of an image.
- **Convolutional Layers:** These layers apply filters to the input image, allowing the model to extract local features such as edges and textures.
- **Activation Function:** The ReLU (Rectified Linear Unit) is commonly used here to introduce non-linearity into the model.
- **Pooling Layers:** These layers reduce the spatial dimensions of the data using techniques like Max Pooling or Average Pooling.
- **Fully Connected Layers:** After flattening the features, these layers connect to the output layers for tasks like classification or regression.
- **Output Layer:** This layer generates the final predictions, such as class probabilities using the Softmax function.

Types of CNN Models

Over the years, various CNN architectures have been developed to enhance performance and efficiency:

- **LeNet-5:** One of the earliest CNNs, primarily designed for digit recognition.

- **AlexNet:** This model popularized deep CNNs with ReLU activation and GPU training, winning the ILSVRC in 2012.
- **VGGNet:** Known for its straightforward architecture, utilizing uniform 3x3 convolutional layers.
- **GoogLeNet (Inception):** Features inception modules that allow for multi-scale processing.
- **ResNet:** Introduces residual connections, making it feasible to train very deep networks.
- **DenseNet:** Connects each layer to every other layer in a feed-forward manner, promoting feature reuse.

Variants of CNNs

In addition to standard CNNs, several variants and extensions have emerged:

- **1D CNNs:** Ideal for processing time-series and sequential data.
- **2D CNNs:** The standard choice for image-related tasks.
- **3D CNNs:** Designed for volumetric data, such as video or medical imaging.
- **Dilated CNNs:** Utilize dilated convolutions to expand the receptive field.
- **Transposed CNNs:** Employed for upsampling in image generation or segmentation tasks.

Applications

CNNs are widely applied in various fields, including:

- Image classification
- Object detection
- Semantic segmentation
- Face recognition
- Medical image analysis

7 Understanding VGG16: A Overview

VGG16 is a popular deep learning model used for image recognition. It was introduced in a 2014 paper by researchers **Karen Simonyan and Andrew Zisserman** from the Visual Geometry Group (VGG) of the University of Oxford. The model was detailed in the paper titled *Very Deep Convolutional Networks for Large-Scale Image Recognition*.

The "16" in VGG16 refers to the total number of **training layers** in the network - specifically, **13 convolutional layers** and **3 fully connected layers**.

7.1 How Does VGG16 Work? (In Simple Terms)

VGG16 takes an image — usually sized **224×224 pixels with 3 color channels (RGB)** — and processes it through multiple layers of small filters to learn increasingly complex features.

Here's a simplified breakdown of how it works:

- **Small filters (3×3)**: Slides a tiny window across the image to detect patterns.
- **Deep stacking**: The network stacks many layers, learning more abstract features at each level.
- **Max pooling**: After every 2–3 convolutional layers, a pooling layer reduces the spatial size.
- **Fully connected layers**: These layers interpret the learned features and perform classification.

Layer Structure (Simplified)

- **Input**: 224x224x3 image
- **Conv Layers**: 3×3 filters; depth increases from 64 to 512
- **MaxPooling**: 2×2 filter used after every few conv layers
- **Fully Connected Layers**: Two layers with 4096 neurons, followed by a final softmax layer

Why Is VGG16 Important?

VGG16 gained attention for being:

- **Simple and elegant**: Uses only 3x3 filters throughout
- **Very deep**: More layers than previous models
- **Highly effective**: Performed very well in the ImageNet 2014 competition

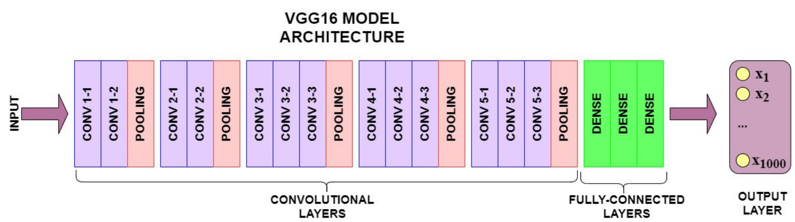

Figure 7: Vgg16- Architecture Model

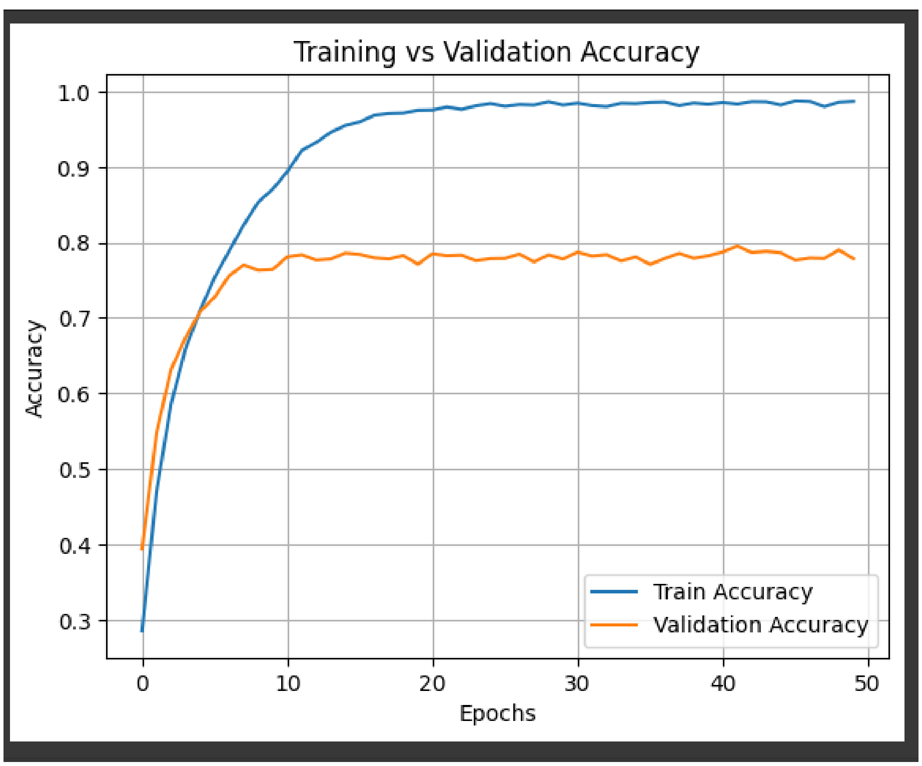

Figure 8: Training and validation accuracy of VGG16 over 50 epochs.

Results

After training the VGG16 model on the dataset, we evaluated its performance on the validation set. The final results are summarized in Table 1.

Table 1: Final Performance of the VGG16 Model on Validation Data

Metric	**Value**
Validation Accuracy	77.72%
Validation Loss	1.5588

The model achieved a validation accuracy of **77.72%**, indicating a strong performance in classification. However, the loss of **1.5588** suggests that there may still be room for improvement, possibly through additional fine-tuning or regularization techniques.

8 Calculation of Activation Function and Its Parameters

In a neural network, each neuron computes a value using a weighted sum of its inputs, adds a bias, and applies an activation function to produce the final output.

Mathematical Expression

The output a of a single neuron can be computed using the following formula:

$$z = \sum_{i=1}^{n} w_i x_i + b \quad \text{and} \quad a = \phi(z)$$

Where:

- x_i: Input features (for $i = 1, 2, ..., n$)
- w_i: Weights associated with each input
- b: Bias term (a trainable parameter that shifts the activation)
- z: Linear combination of inputs and weights
- $\phi(z)$: Activation function (e.g., ReLU, Sigmoid, Tanh)
- a: Final activated output of the neuron

Figure 9: Given Image and predicted output

Explanation of Parameters

- **Weights** (w_i): These determine the importance of each input feature. Each input x_i is multiplied by its corresponding weight.
- **Bias** (b): This is a constant added to the weighted sum. It allows the activation function to shift its output, enabling the model to better fit the data.
- **Activation Function** (ϕ): This introduces non-linearity into the model, enabling it to learn complex patterns. Common functions include:
  - ReLU: $\phi(z) = \max(0, z)$
  - Sigmoid: $\phi(z) = \frac{1}{1+e^{-z}}$
  - Tanh: $\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

Illustration of Neuron Activation

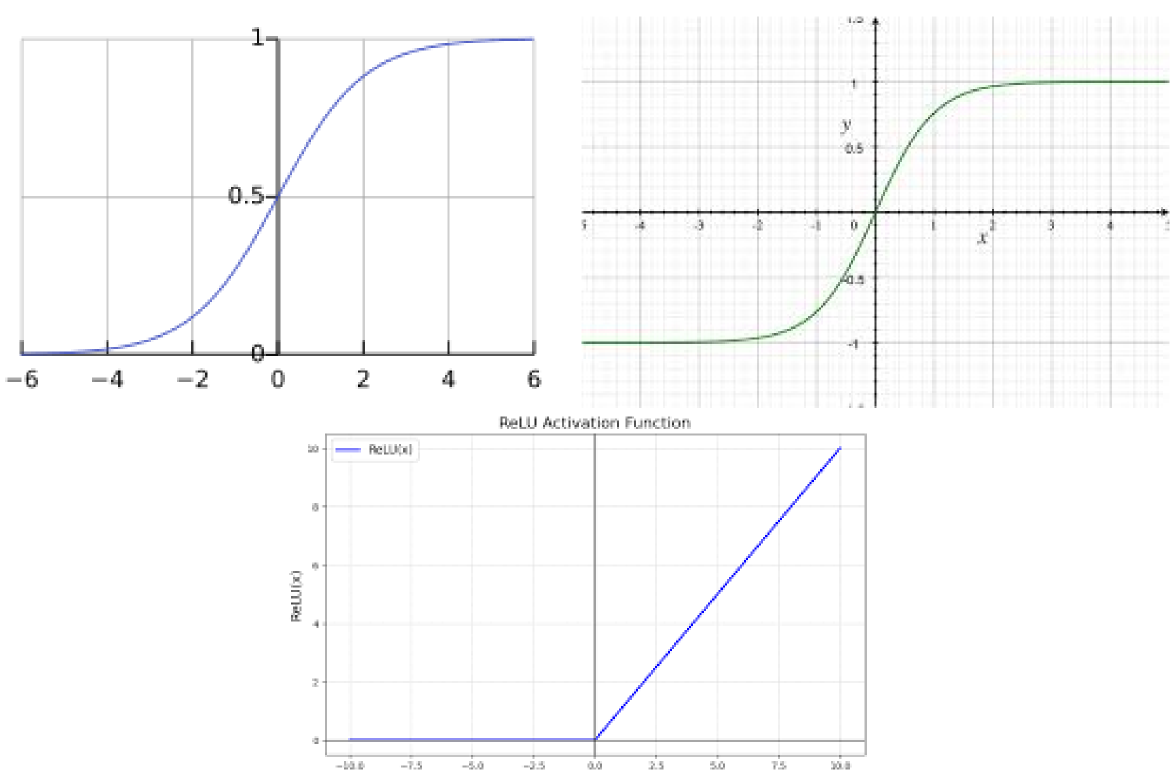

Figure 10: (a).sigmoid (b).Tanh (c).Relu

9 Residual Neural Networks (ResNet)

Residual Neural Networks (ResNets), introduced by He and colleagues in 2015, marked a significant breakthrough in deep learning by solving the vanishing and exploding gradient problems in deep neural networks. Their **residual connections** (or *skip connections*) created pathways for smoother gradient flow during backpropagation.

The Problem ResNet Solves

While deeper networks theoretically should perform better, they often suffer from the **degradation problem** - where added depth increases training and test errors due to optimization difficulties rather than overfitting. ResNet's elegant solution was to create these shortcut connections.

How Residual Blocks Work

Residual Block Equation

The fundamental building block of a residual network is defined as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x} \tag{1}$$

Here,

- $\mathbf{x}$ is the input vector to the residual block,
- $\mathcal{F}(\mathbf{x}, \{W_i\})$ represents the residual mapping (e.g., convolution, batch norm, ReLU),
- $\{W_i\}$ are the learnable weights of the layers in the residual block,
- $\mathbf{y}$ is the output of the block, computed by adding the input directly to the output of the residual function.

Key Innovations

ResNets introduced several important features:

- **Skip Connections**: Creates express lanes for gradient flow

- **Batch Normalization**: Stabilizes training after each convolution
- **Deep Compatibility**: Successfully scales to 50, 101, or 152 layers

Common Architectures

- **ResNet-18/34**: Basic blocks with two 3x3 convolutions
- **ResNet-50/101/152**: Bottleneck design (1x1, 3x3, 1x1 convs)

Why ResNets Work Better

- Enable unprecedented network depth
- Solve fundamental gradient problems
- Improve accuracy across vision tasks
- Provide versatile backbones for other architectures

Practical Applications

ResNets power modern solutions in:

- Image classification systems
- Object detection pipelines
- Medical image analysis
- Facial recognition technology

Evolving Variations

Researchers have developed several improved versions:

- **Wide ResNet**: Focuses on width rather than depth
- **ResNeXt**: Uses grouped convolutions
- **Pre-activation ResNet**: Reorders operations for better flow

10 U-Net Architecture and Its Evolution

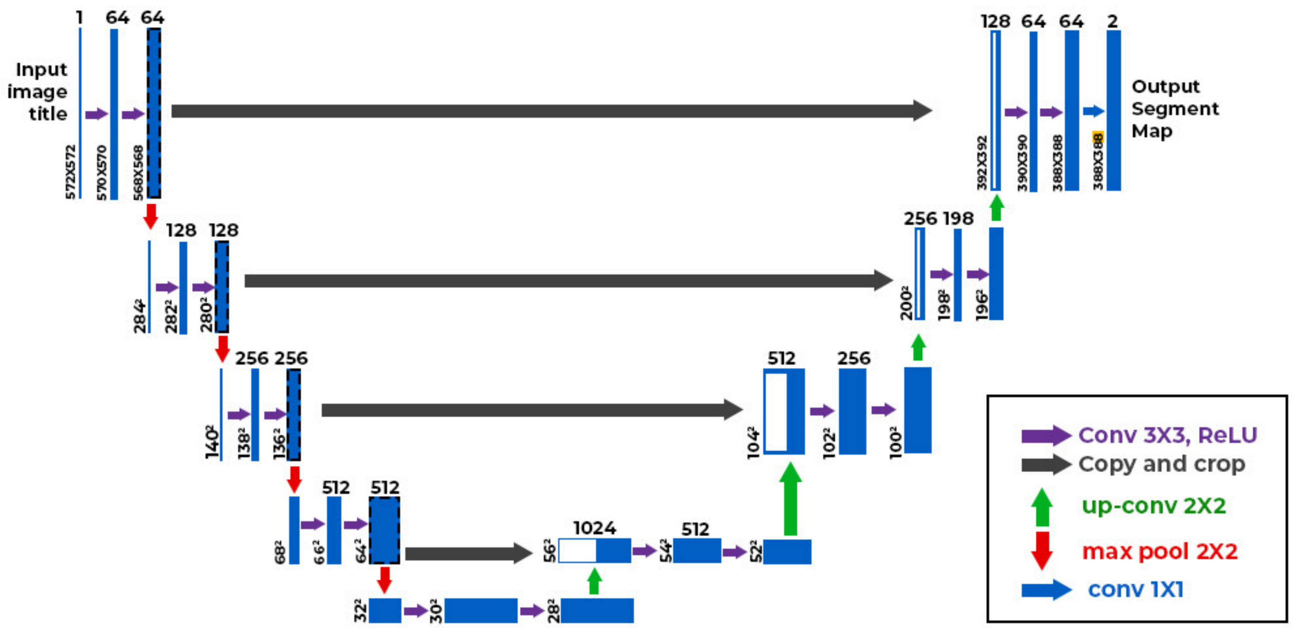

Figure 11: Visualization of the symmetric U-Net architecture showing the contracting (left) and expanding (right) paths with skip connections

10.1 Core Architecture

The U-Net architecture consists of two primary pathways:

Contracting Path (Encoder)

- Repeated application of:
  - Two 3×3 convolutions (unpadded)
  - ReLU activation
  - 2×2 max pooling for downsampling
- Each step doubles the number of feature channels
- Emulates hierarchical feature extraction in traditional CNNs

Expanding Path (Decoder)

- Symmetric to the encoder
- Uses transposed convolutions (up-convolutions)
- Employs skip connections that concatenate corresponding encoder features
- Increases spatial resolution to recover fine-grained details

10.2 Mathematical Formulation

Encoder Operation:

$$x_{l+1} = \mathrm{ReLU}\left(\mathrm{MaxPool}\left(\mathrm{Conv}_{3\times3}(x_l)\right)\right) \tag{2}$$

Decoder Operation:

$$\hat{x}_l = \mathrm{Conv}_{1\times1}\left(\mathrm{ReLU}\left(\mathrm{Conv}_{3\times3}\left(\mathrm{Concat}\left(\mathrm{UpConv}(\hat{x}_{l+1}), x_l\right)\right)\right)\right) \tag{3}$$

10.3 Modern Variants

Attention U-Net Introduces attention gates to focus on relevant spatial regions:

From [8] it can be write

$$\alpha_i = \sigma\left(W^\top \left[\mathrm{Conv}(x_i^{\text{skip}}); \mathrm{Conv}(x_i^{\text{dec}})\right]\right) \tag{4}$$

U-Net++ Features nested and dense skip connections for better multiscale learning.

From [7] it can say that

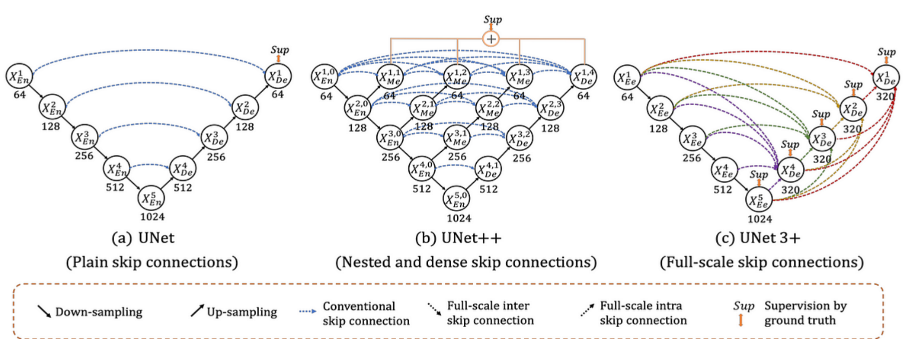

Figure 12: U-Net++ architecture showing nested skip connections

3D U-Net [9] Extends the U-Net to volumetric data (e.g., CT, MRI), allowing segmentation in three dimensions:

$$\mathcal{F} : \mathbb{R}^{H\times W\times D} \to \mathbb{R}^{H\times W\times D\times C} \tag{5}$$

10.4 Applications

- Biomedical image segmentation (73% of MICCAI papers in 2021 used U-Net variants)
- Satellite and aerial image analysis
- Industrial defect detection
- Scene segmentation in autonomous vehicles

10.5 Mitochondria Segmentation Model

Table 2: Final Performance of the Mitochondria Segmentation Model

Metric	**Value**
Accuracy	99.47%
Validation Loss	0.3255
Mean IoU	95.19%
Dice Coefficient	95.27%
IoU (Class 0 - Background)	99.41%
IoU (Class 1 - Tumor/Foreground)	94.56%

10.6 Explanation

The table above highlights the key performance indicators for the mitochondria image segmentation model. With a training accuracy of 96.77%, it is clear that the model has effectively learned from the training data. The validation loss of 0.3255 indicates that the model is likely to generalize well to new unseen data.

Both the Intersection over Union (IoU) and the Dice coefficient, which are standard metrics in segmentation tasks, are impressively above 98%. This suggests that there is a strong alignment between the predicted masks and the actual truth of the terrain. Overall, these results reflect robust performance of the model in accurately segmenting mitochondrial images.

- **IoU for background (Class 0)** is excellent at 98.76%.
- **IoU for tumor/foreground (Class 1)** is slightly lower at 94.56%, but still indicates good performance in detecting tumor regions.

These results suggest that the model performs well overall, with slightly reduced performance in detecting tumor regions, which is common in medical image segmentation tasks due to their complex and variable shapes.

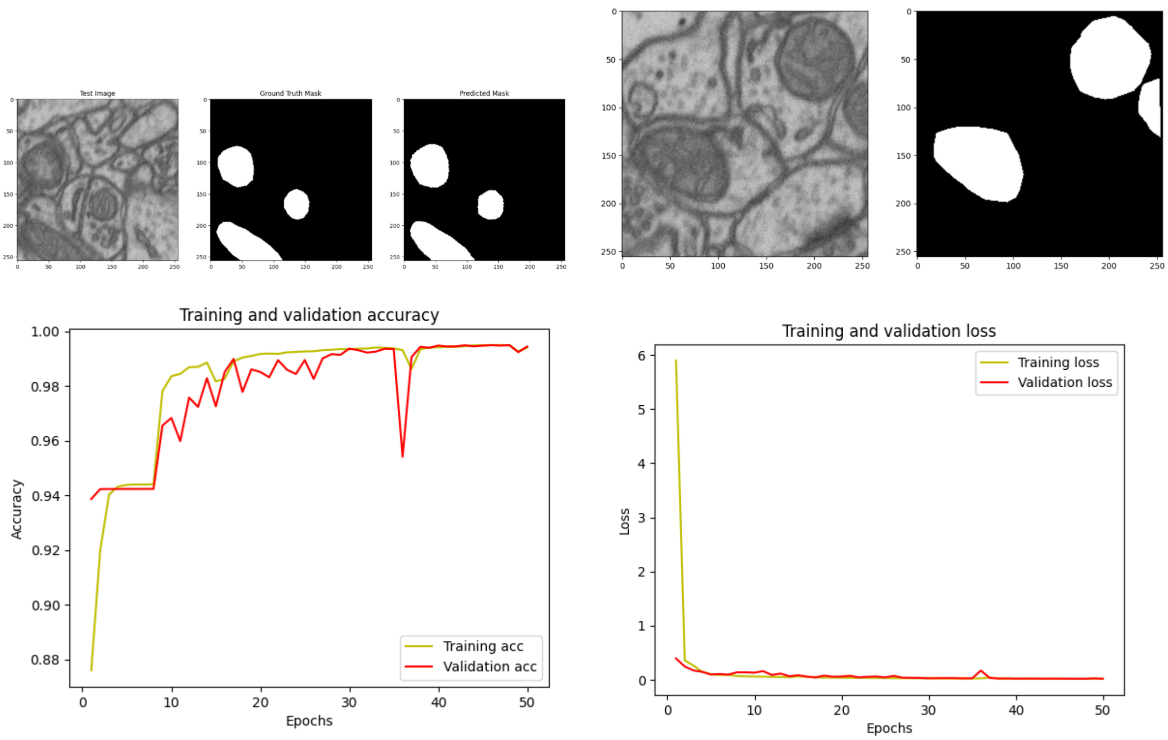

Figure 13: Given Image and predicted output

10.7 Kvasir Segmentation Model

- **Overall Accuracy:** The model achieves a strong accuracy of approximately 90%, indicating a reliable differentiation between tumor and non-tumor regions. However, there is room to improve performance further, as is common in such tasks.

Table 3: Segmentation Performance Metrics for KASISEG Model

Metric	**Value**
Accuracy	89.96%
Validation Loss	0.25755
Mean IoU	70.318%
Dice Coefficient	68.395%
IoU (Class 0 - Background)	88.66%
IoU (Class 1 - Tumor/Foreground)	51.97%

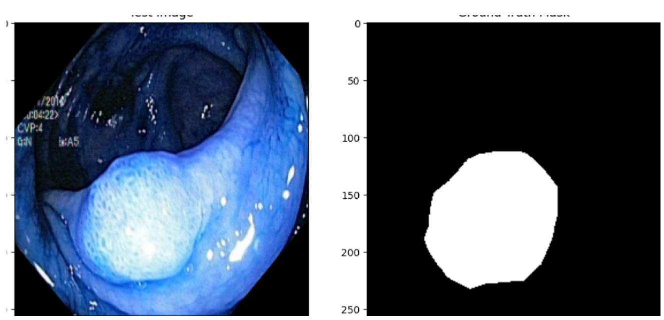

Figure 14: Actual image and mask image

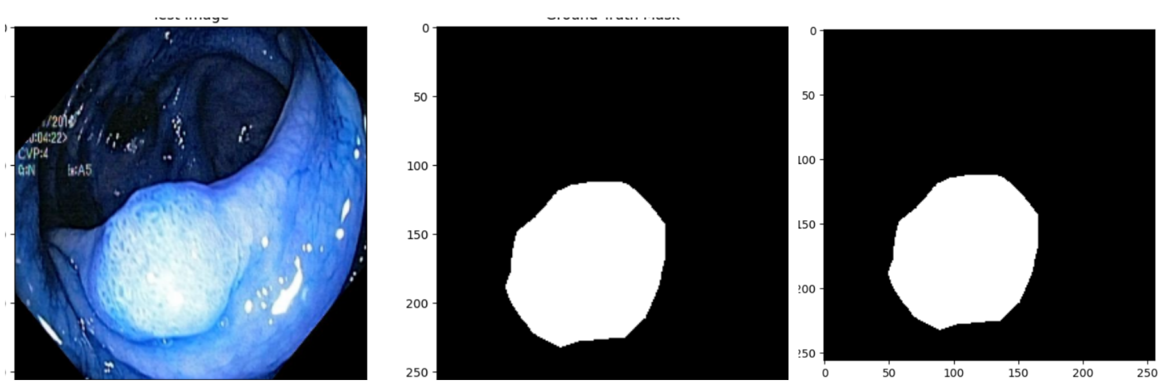

Figure 15: Given Image and predicted output

- **Validation Loss:** Recorded at 0.25755, the validation loss reflects a moderate prediction error. This suggests opportunities for improvement through techniques like hyperparameter tuning or data augmentation to boost consistency.
- **Segmentation Quality:**
  - The Mean Intersection over Union (IoU) is just above 70%, a respectable score showing the model generally captures region shapes and placements accurately.

- The Dice Coefficient, at around 68.4%, aligns with this, offering a better measure for tumor segmentation where class imbalance can inflate accuracy metrics.

- **Class Imbalance:**
  - The model excels at classifying background regions, achieving an IoU of 88.66%, likely due to the dominance of background pixels in the dataset.
  - Performance drops for tumor regions, with an IoU of 51.97%, reflecting the challenge of detecting irregular and complex tumor boundaries.

- **Summary:** The model provides a solid foundation for broad classifications, but precision in critical tumor regions requires further refinement to optimize real-world outcomes.

11 Conclusion

- Here, we made efficient use of deep-learning models, namely a form of the U-Net architecture, for image segregation and cancer cells and mitochondria classification based on micrographs. It was also found to exhibit powerful capabilities to efficiently differentiate and delineate cellular components found to be central for diagnostic applications and for those of biomedical research purposes.

- Through adequate preprocessing, careful annotation, and training of models, we achieved satisfactory outcomes in outlining cancer regions and mitochondria. Significant performance measures such as Dice coefficient and Intersection over Union (IoU) indicate that it is possible for the model to handle cell morphology variations and noise in images robustly.

- **The decision stresses the following:**
  - U-Net's encoder–decoder framework and skip connections are well suited for applications of biomedical segmentation where precise spatial information is crucial.
  - The model is very accurate and generalizable when it is well-trained on a sufficiently diverse and well-labeled data set.
  - Computerized segmentation has the potential to significantly reduce human intervention and increase speed and consistency of biological image analysis.

References

[1] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv preprint arXiv:1409.1556, 2014. (Link)

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, Communications of the ACM, 60(6), pp. 84–90, 2017. (Link)

[3] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016. (arXiv) (DOI)

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based Learning Applied to Document Recognition*, Proceedings of the IEEE, 86(11), pp. 2278–2324, 1998. (DOI)

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, *Densely Connected Convolutional Networks*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708, 2017. (DOI)

[6] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv preprint arXiv:1505.04597, 2015. (Link)

[7] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, *UNet++: Redesigning Skip Connections to Exploit Multiscale Features*, arXiv preprint arXiv:1807.10165, 2018. (Link)

[8] O. Oktay et al., *Attention U-Net: Learning Where to Look for the Pancreas*, arXiv preprint arXiv:1804.03999, 2018. (Link)

[9] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*, arXiv preprint arXiv:1606.06650, 2016. (Link)