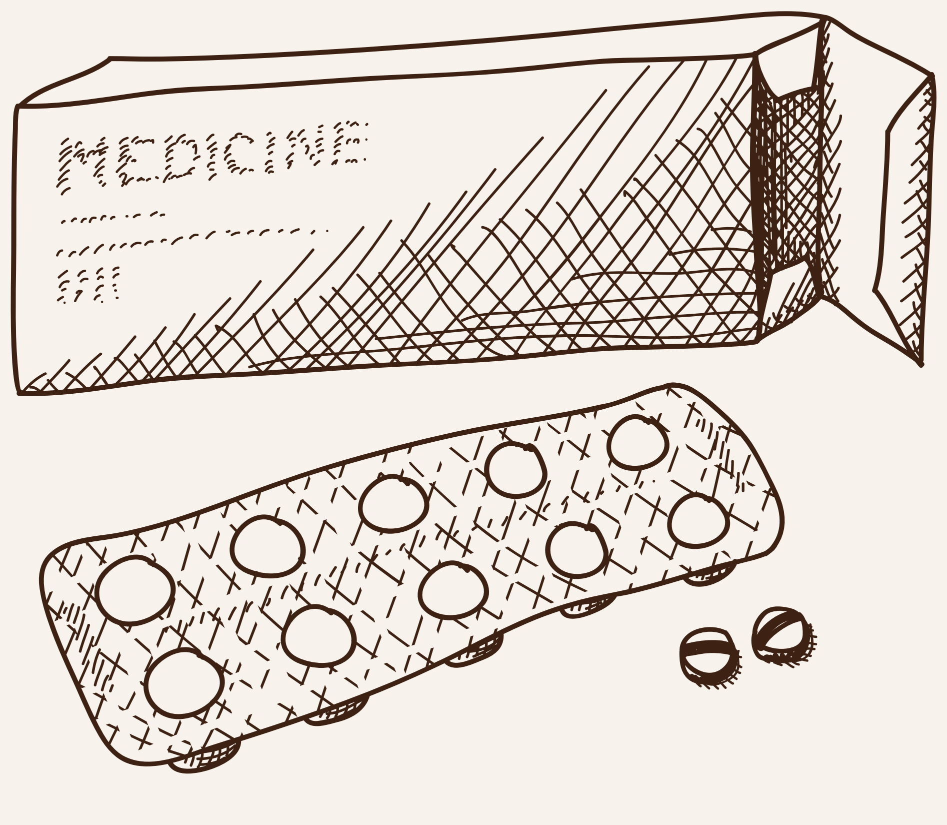
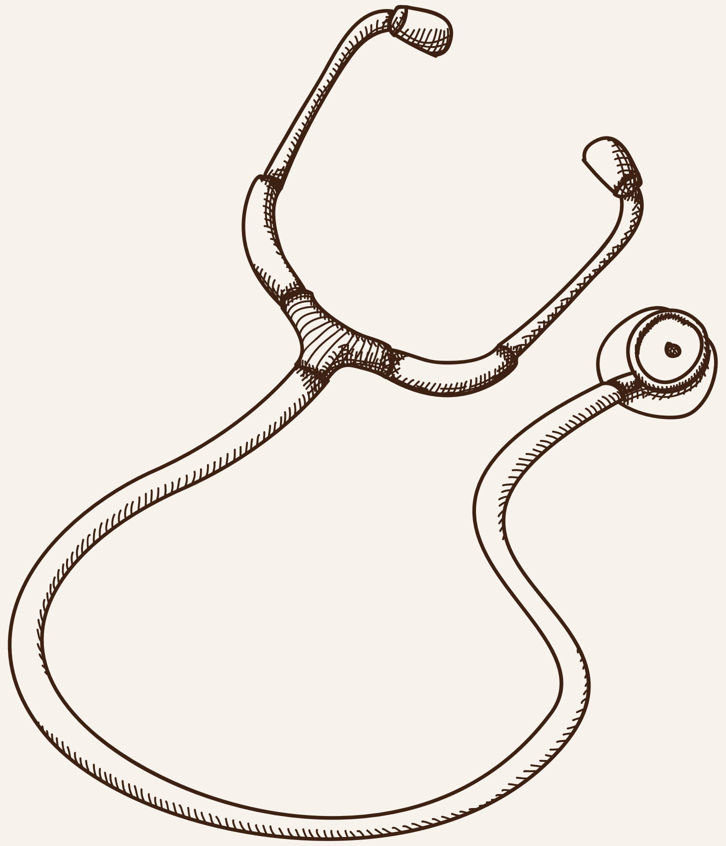
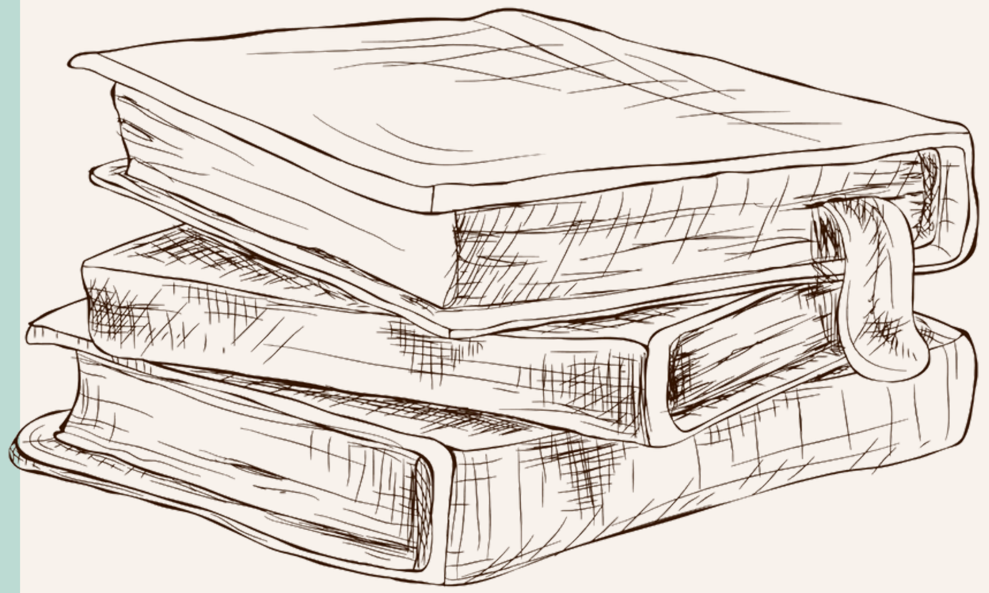
SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE

SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

Presented By:
Nikitha Matta (21481A4245)
Venkata Rami Reddy (21481A4219)
Boyi Gurunadh (22485A4202)
Kodali Naga Vamsi (22485A4205)



ULTRASOUND NERVE SEGMENTATION

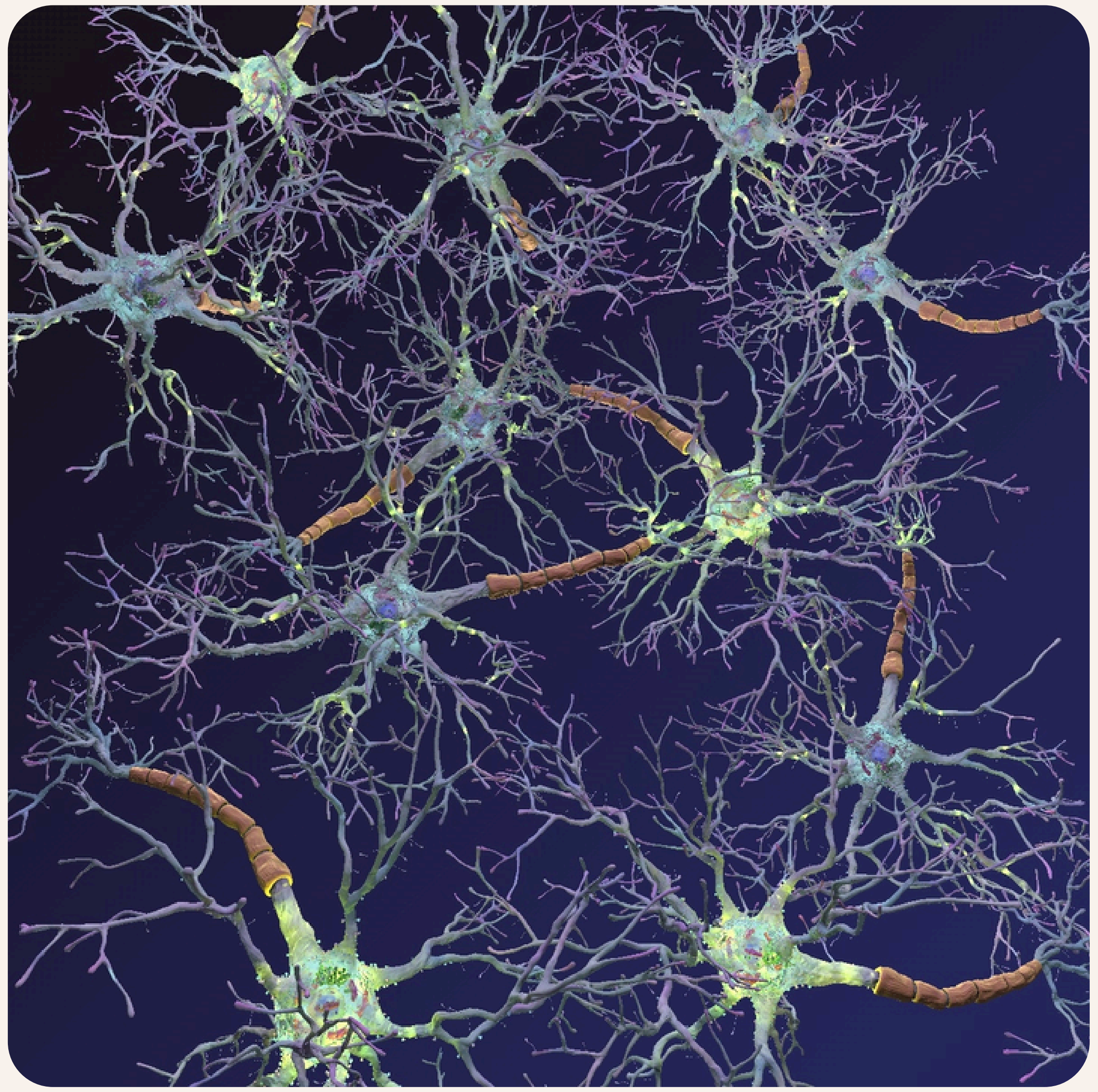
TABLE OF CONTENTS

Abstract

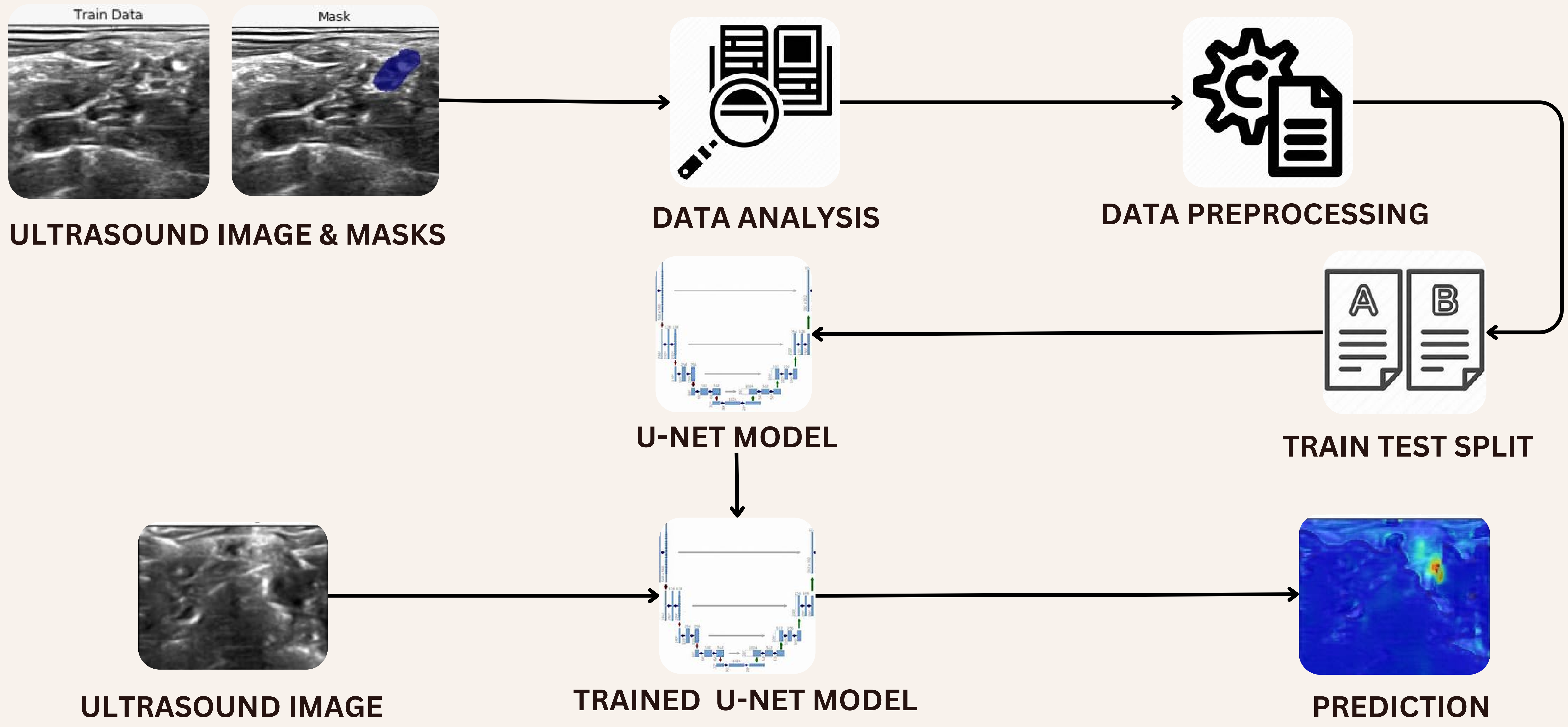
This project utilizes a U-Net model to segment nerves in ultrasound images, enhancing medical diagnosis and treatment planning. By leveraging deep learning and image processing techniques on annotated datasets, it achieves accurate segmentation, promising significant advancements in automated ultrasound image analysis for neurological conditions.

Business Problem

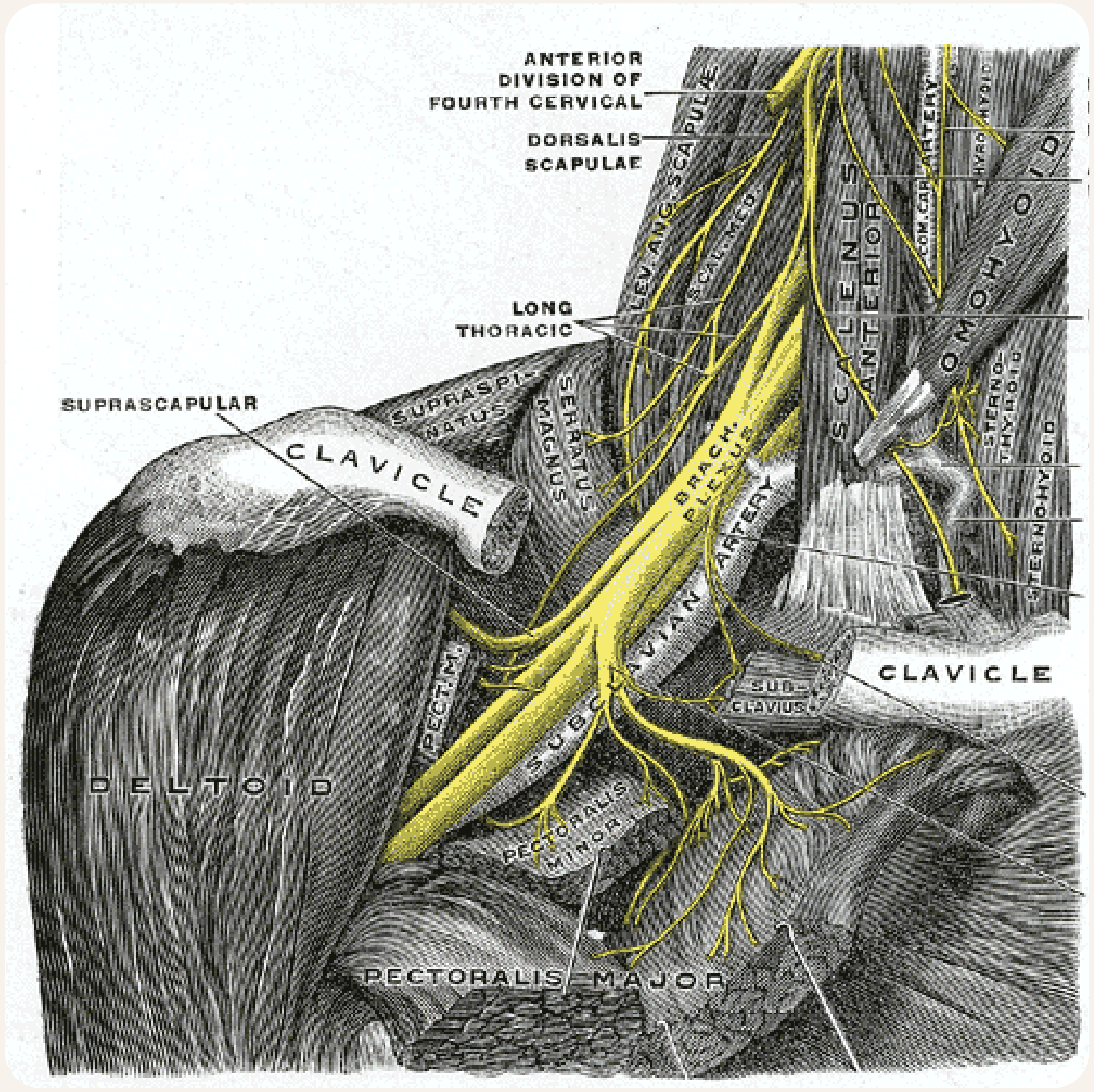
Peripheral Nerve Blocking (PNB) is crucial for post-surgery pain management, utilizing local anesthetics to numb specific areas while maintaining patient consciousness. However, identifying nerve structures via ultrasound imaging poses challenges due to echo perturbations and speckle noise, complicating precise application.



Flow Diagram



Literature Survey



01 **Location and Function:** The brachial plexus is a group of nerves found in the neck and armpit area. They help us move our arms and hands and feel things like touch and temperature.

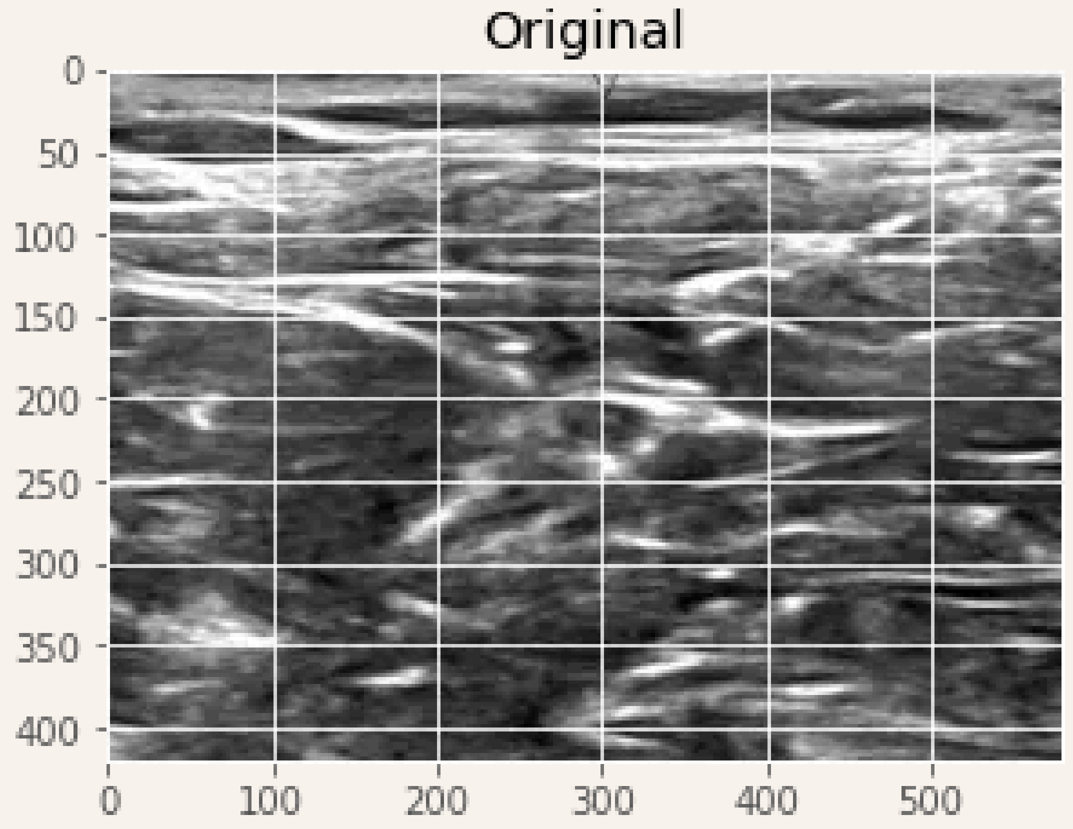
02 **Purpose of Anesthesia:** During certain surgeries or procedures on the arm or hand, doctors give anesthesia in the brachial plexus area. This makes the nerves numb, so the patient doesn't feel any pain. It's like hitting the pause button on pain signals.

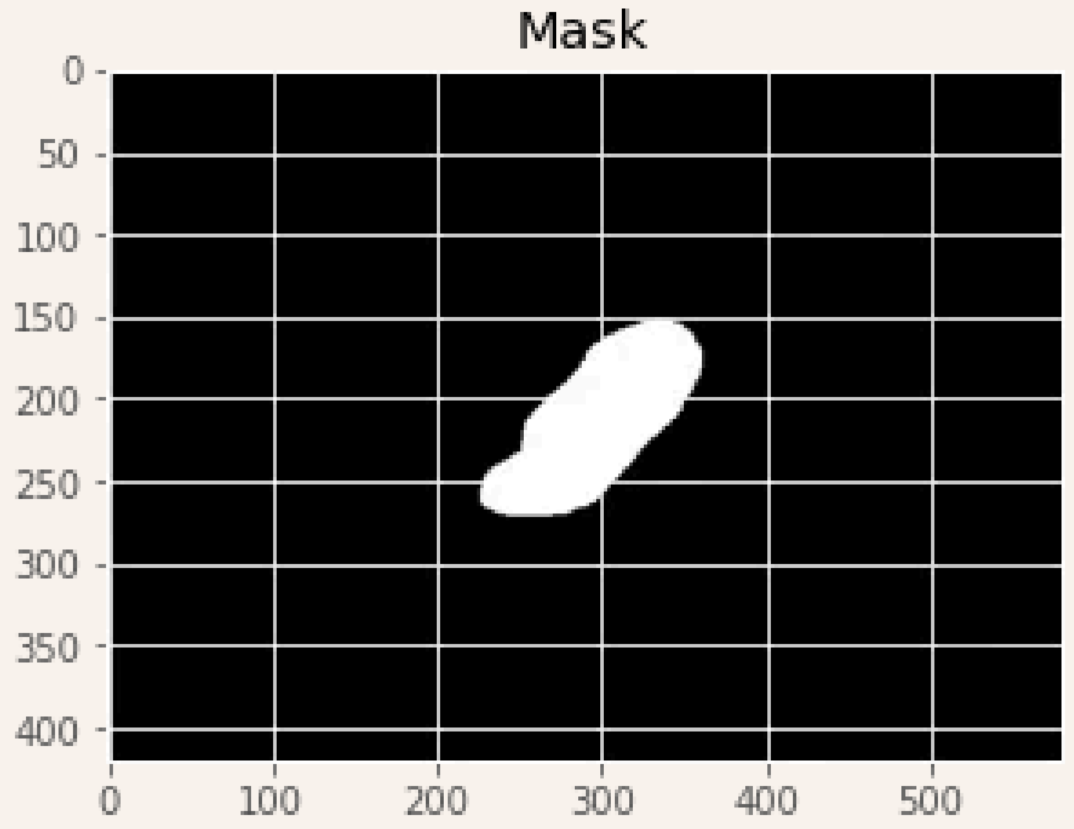
03 **Pain Relief and Surgery:** Numbing the brachial plexus helps patients feel comfortable during surgery. Since they can't feel anything in their arm or hand, surgeons can work without causing pain. This makes the surgery safer and helps patients recover better.

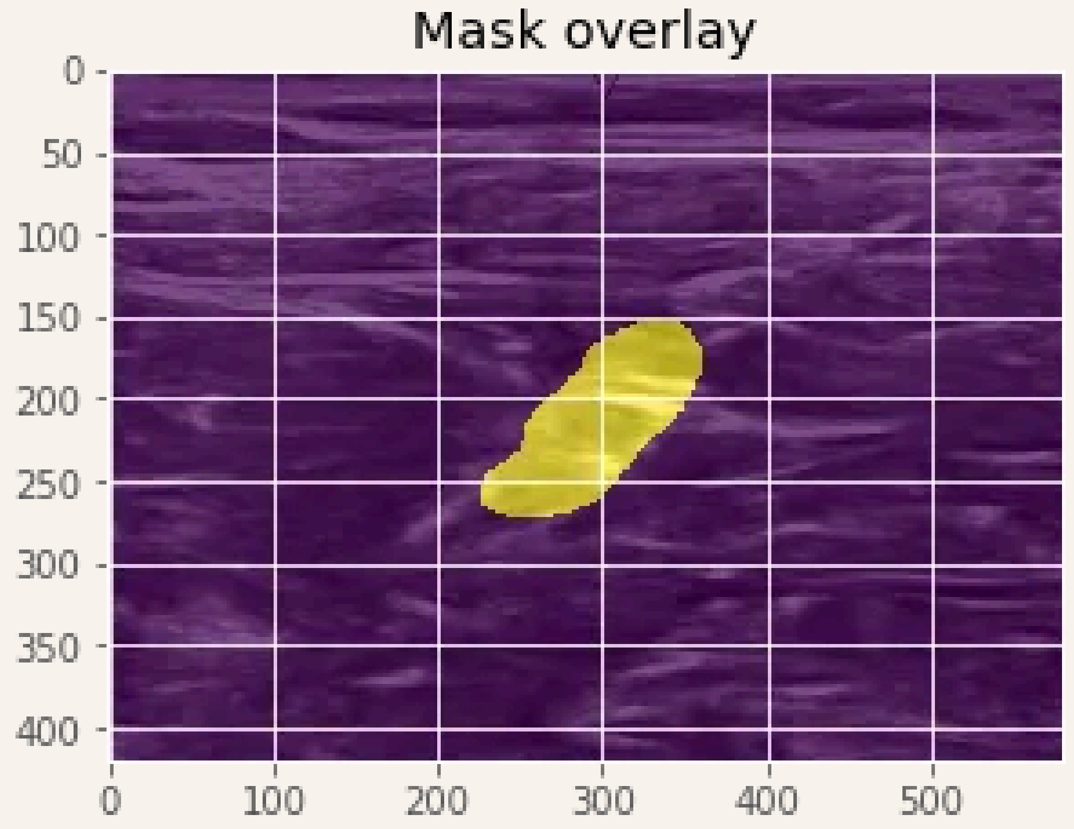
Data Preprocessing

- Dataset consists of train and test folders, along with a CSV file containing encoded pixel values for training masks.
- Train folder contains 5635 images and corresponding masks, organized by 47 subjects with approximately 120 images and masks per subject.
- Images are named 'subjectno_imageno.tif', while masks are named 'subjectno_imageno_mask.tif'.
- Masks are annotated by domain experts, accurately indicating regions corresponding to the Brachial Plexus nerve.
- Test folder contains 5508 images for model evaluation.
- Dataset exhibits variability, with some images containing masks while others do not.

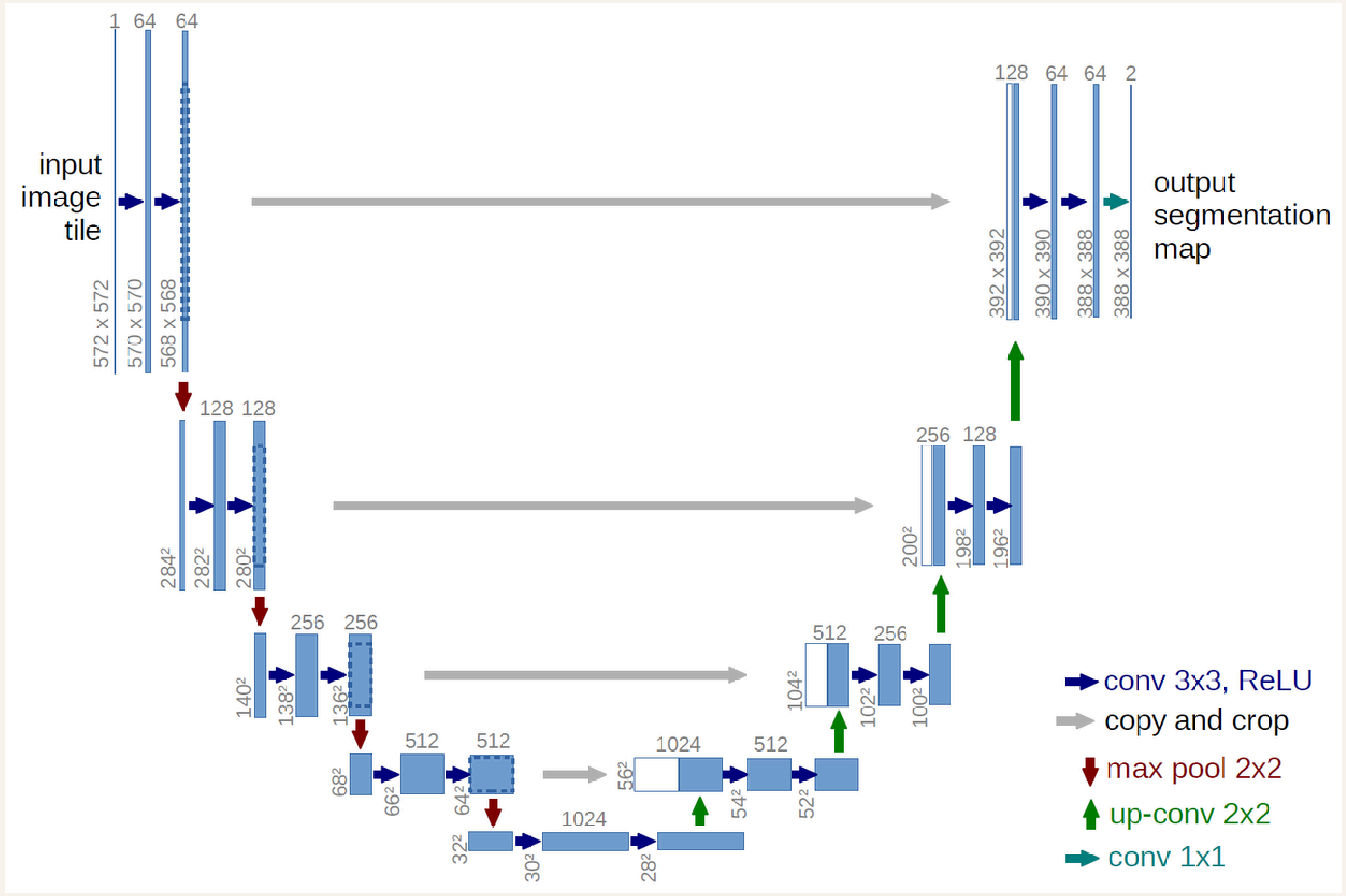
Data Preprocessing







U-Net Model



What is it?

Type of Convolutional Neural Network (CNN) architecture

Why is it used?

Commonly used for Image Segmentation tasks

How it works?

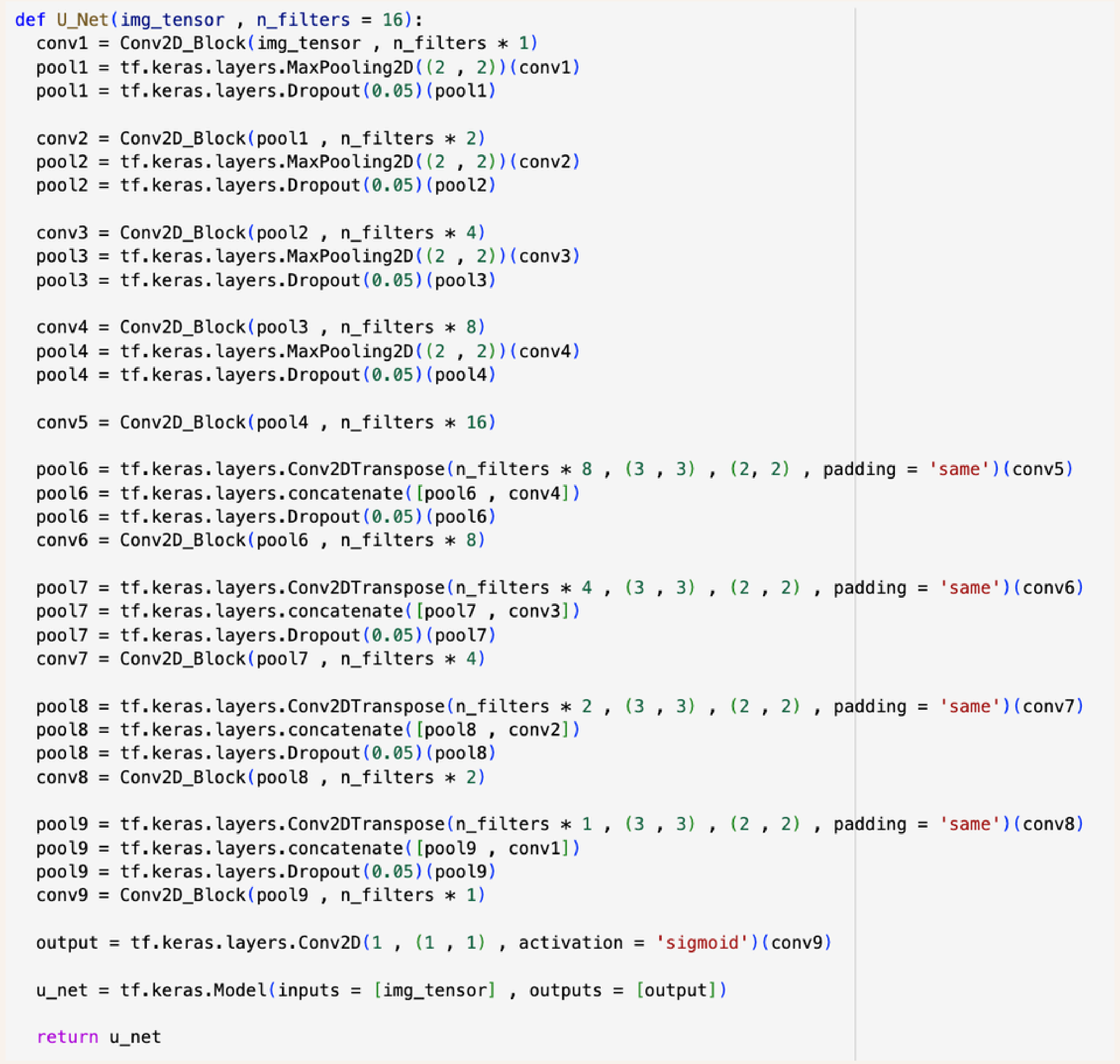
Encoder-Decoder Architecture

- Contracting Path (Encoder)
- Expanding Path (Decoder)

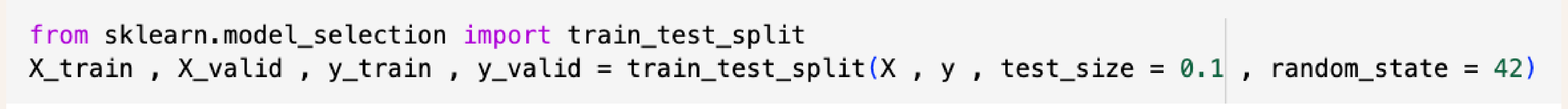
Skip Connections

Final Layer

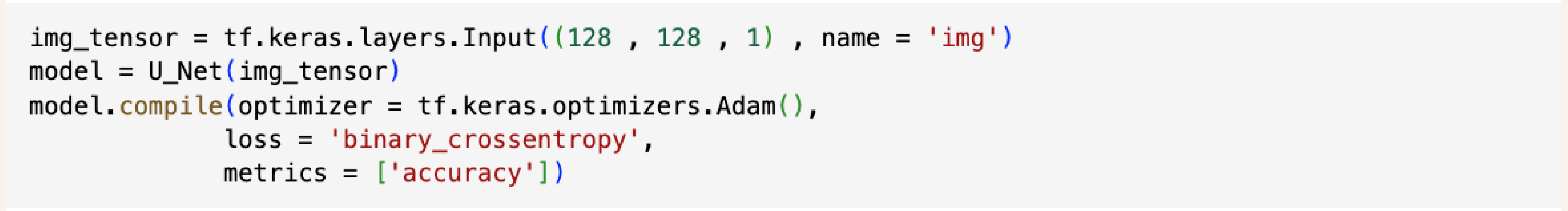
Methodology



```python
def U_Net(img_tensor , n_filters = 16):
  conv1 = Conv2D_Block(img_tensor , n_filters * 1)
  pool1 = tf.keras.layers.MaxPooling2D((2 , 2))(conv1)
  pool1 = tf.keras.layers.Dropout(0.05)(pool1)

  conv2 = Conv2D_Block(pool1 , n_filters * 2)
  pool2 = tf.keras.layers.MaxPooling2D((2 , 2))(conv2)
  pool2 = tf.keras.layers.Dropout(0.05)(pool2)

  conv3 = Conv2D_Block(pool2 , n_filters * 4)
  pool3 = tf.keras.layers.MaxPooling2D((2 , 2))(conv3)
  pool3 = tf.keras.layers.Dropout(0.05)(pool3)

  conv4 = Conv2D_Block(pool3 , n_filters * 8)
  pool4 = tf.keras.layers.MaxPooling2D((2 , 2))(conv4)
  pool4 = tf.keras.layers.Dropout(0.05)(pool4)

  conv5 = Conv2D_Block(pool4 , n_filters * 16)

  pool6 = tf.keras.layers.Conv2DTranspose(n_filters * 8 , (3 , 3) , (2, 2) , padding = 'same')(conv5)
  pool6 = tf.keras.layers.concatenate([pool6 , conv4])
  pool6 = tf.keras.layers.Dropout(0.05)(pool6)
  conv6 = Conv2D_Block(pool6 , n_filters * 8)

  pool7 = tf.keras.layers.Conv2DTranspose(n_filters * 4 , (3 , 3) , (2 , 2) , padding = 'same')(conv6)
  pool7 = tf.keras.layers.concatenate([pool7 , conv3])
  pool7 = tf.keras.layers.Dropout(0.05)(pool7)
  conv7 = Conv2D_Block(pool7 , n_filters * 4)

  pool8 = tf.keras.layers.Conv2DTranspose(n_filters * 2 , (3 , 3) , (2 , 2) , padding = 'same')(conv7)
  pool8 = tf.keras.layers.concatenate([pool8 , conv2])
  pool8 = tf.keras.layers.Dropout(0.05)(pool8)
  conv8 = Conv2D_Block(pool8 , n_filters * 2)

  pool9 = tf.keras.layers.Conv2DTranspose(n_filters * 1 , (3 , 3) , (2 , 2) , padding = 'same')(conv8)
  pool9 = tf.keras.layers.concatenate([pool9 , conv1])
  pool9 = tf.keras.layers.Dropout(0.05)(pool9)
  conv9 = Conv2D_Block(pool9 , n_filters * 1)

  output = tf.keras.layers.Conv2D(1 , (1 , 1) , activation = 'sigmoid')(conv9)

  u_net = tf.keras.Model(inputs = [img_tensor] , outputs = [output])

  return u_net
```



```python
from sklearn.model_selection import train_test_split
X_train , X_valid , y_train , y_valid = train_test_split(X , y , test_size = 0.1 , random_state = 42)
```



```python
img_tensor = tf.keras.layers.Input((128 , 128 , 1) , name = 'img')
model = U_Net(img_tensor)
model.compile(optimizer = tf.keras.optimizers.Adam(),
              loss = 'binary_crossentropy',
              metrics = ['accuracy'])
```

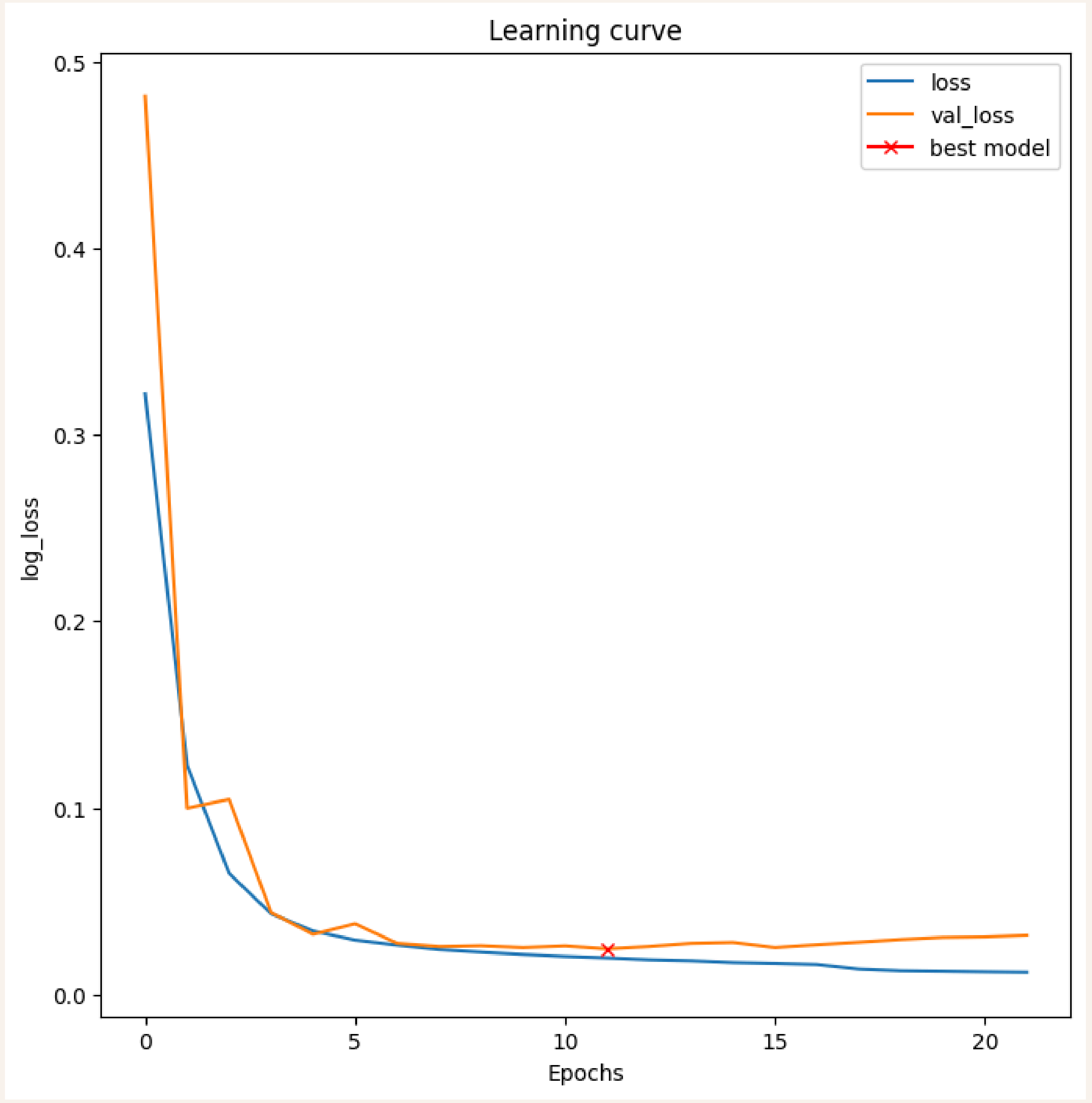
Contracting Path Functions:

- **Conv2D_Block:** A function for creating a convolutional block with specified number of filters.
- **MaxPooling2D:** A function for downsampling feature maps using max pooling with a window size of (2, 2).
- **Dropout:** A function for applying dropout regularization with a dropout rate of 0.05.

Expansive Path Functions:

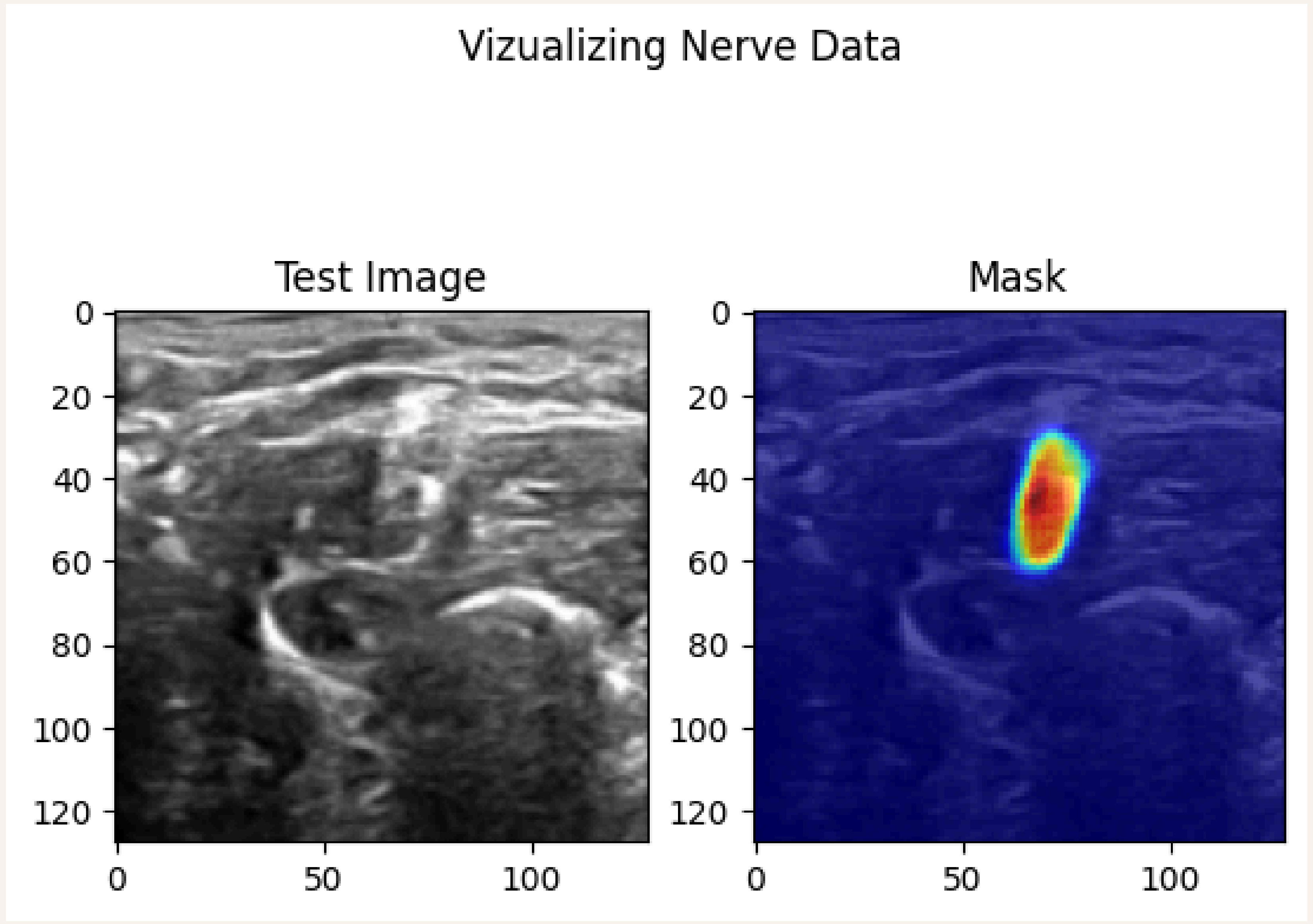
- **Conv2DTranspose:** A function for upsampling feature maps using transpose convolution with a kernel size of (3, 3) and a stride of (2, 2).
- **concatenate:** A function for concatenating feature maps from the contracting path with corresponding feature maps in the expansive path.
- **Dropout:** A function for applying dropout regularization with a dropout rate of 0.05.

Results



- Epochs: 50
- Early Stopping: 22
- Optimizer: Adam
- Accuracy: 98.5%
- Loss: Binary Crossentropy (0.02)

Results


Vizualizing Nerve Data
Test Image
Mask
0
20
40
60
80
100
120
0
50
100

Future Scope

01

Improved Accuracy and Efficiency

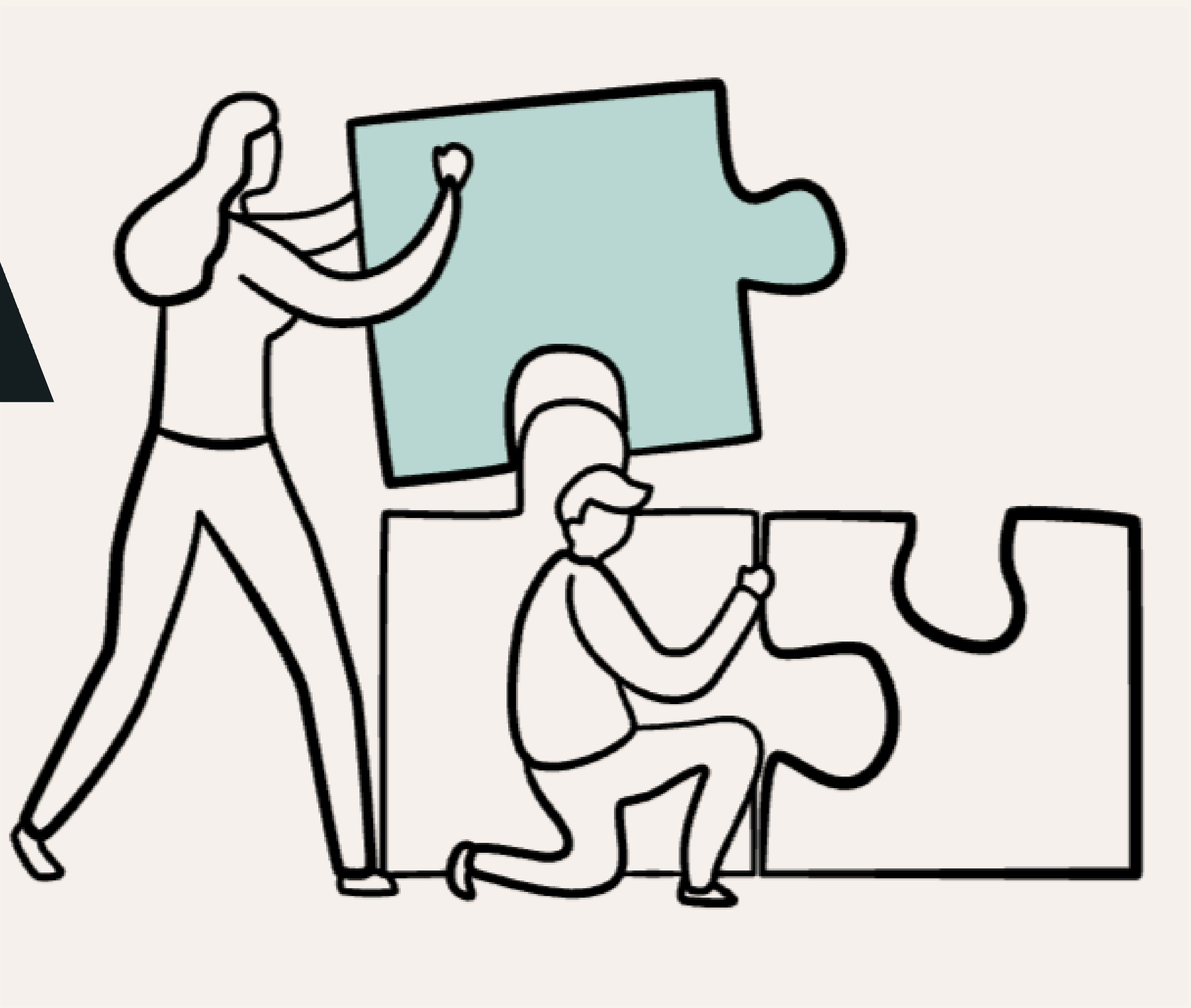
02

Clinical Integration and Decision Support

03

Telemedicine and Remote Healthcare

Q&A



Thank you!