

# Development of Computer Aided Multi-organ Cancer Diagnostic System from Histopathological Images

Saket Kumar

National Institute of Technology, Rourkela

Roll Number: 121EI0708

Email: 121ei0708@nitrkl.ac.in

**Abstract**—This paper introduces a model using an enhanced U-Net with Squeeze-and-Excitation (SE) blocks and Atrous Spatial Pyramid Pooling (ASPP) for nuclei segmentation in HE-stained histopathology images. The model captures multi-scale context and emphasizes critical features, providing accurate nucleus segmentation and reliable separation of overlapping nuclei, advancing medical image analysis.

## I. INTRODUCTION

The examination of histopathological images is essential for diagnosing a wide range of diseases, particularly cancer. One critical task in this process is accurately segmenting nuclei to gain insights into tissue organization and identify abnormalities. Performing this analysis manually is often time-consuming and prone to inconsistencies. As a result, there is a growing demand for automated segmentation approaches. The Multi-Organ Nucleus Segmentation Challenge (MoNuSeg) seeks to address this need by encouraging the development of models that are capable of working effectively across various organs, leveraging both advanced image processing and machine learning techniques.

## II. LITERATURE REVIEW

Several previous studies have contributed to the field of nucleus segmentation:

**A Multi-Organ Nucleus Segmentation Challenge:** The MoNuSeg 2018 Challenge took advantage of fully convolutional networks, specifically U-Net, FCN, and Mask-RCNN variants, and color normalization, as well as strong data augmentation for the purpose of developing and validating generalized nucleus segmentation methods in the field of digital pathology.

**Hybrid attention mechanism of feature fusion for medical image segmentation:** This model employed a hybrid attention architecture incorporating convolutional neural network (CNN) and a Channel Attention Enhancement Module (CAEM) and Transformer to successfully extract multi-scale features, capture long range relationships, and enhance multi-organ segmentation in the medical images

**The Squeeze-and-Excitation (SE) network, introduced by Hu et al. (2018)** enhances convolutional neural networks by dynamically recalibrating channel-wise feature responses. Unlike traditional convolutions that treat all channels equally,

SE blocks prioritize important features, improving overall representational power.

## III. DATASET DESCRIPTION

The Multi-Organ Nucleus Segmentation Challenge (MoNuSeg) dataset, in particular, contains histopathological (HE) stained images of tissues taken from various organs. These images are essential for the understanding of tissue structure and for the diagnosis of different types of cancers, through the study of the morphology of cell nuclei, which can differ in the presence of cancers. The boutique problem of segmenting nuclei from H&E-stained pathology slides is an important task for improving disease diagnosis. The MoNuSeg dataset is designed, among other things, to foster the development of nucleus segmentation algorithms that are generalisable across different types of organs and pathological conditions.

### Training Dataset:

The MoNuSeg training dataset consists of 30 high-resolution HE whole-slide images, with characteristic dimensions of 1000 × 1000 pixels, and downscaled versions from seven different sub-organs: breast, liver, kidney, prostate, bladder, colon and stomach. It was obtained from The Cancer Genome Atlas (TCGA), a test collection of diverse tissue images from patients with different conditions. It consists of a total of 21,623 manually annotated nuclei. These annotations include both epithelial and stromal nuclei that were labelled with great care using cutting edge annotation tools. epstopdf .tif

### Testing Datasets:

The MoNuSeg test dataset contains 14 images (1000 × 1000 pixels each) taken from the same seven organs, indeed each organ having two images. The test annotated dataset comprises of about 7,223 nuclei whose boundaries were delineated accurately. Like test and train data annotated in the previous stage, the training data was annotated in a sort of a comprehensible manner.

## IV. METHODOLOGY

The nucleus segmentation task was approached using fully convolutional networks (FCNs) inspired by U-Net and Mask-RCNN. Key techniques included:

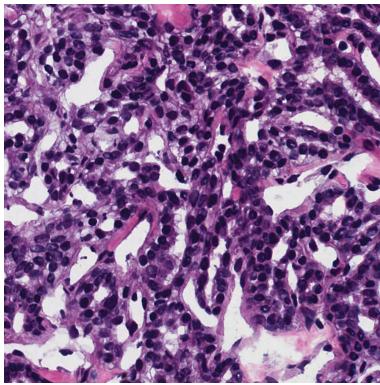


Fig. 1. H&E Strained Image of Kidney cells (Training Dataset)

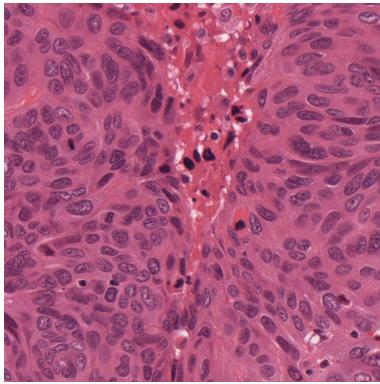


Fig. 2. H&E Strained Image of Breast cells (Testing Dataset)

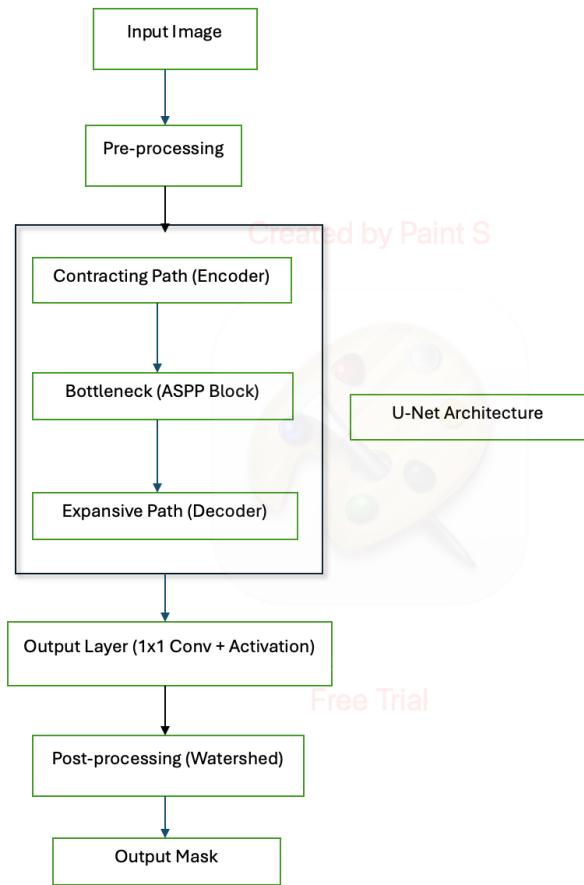


Fig. 3. The architecture of network : U-net

#### A. Input Image (H&Strained)

The input is a stained histopathology image using Hematoxylin and Eosin (H&E). These images are widely used in pathology for detecting nuclei and tissue structures. Hematoxylin stains the nuclei, making them appear dark purple/blue, while Eosin stains the cytoplasm and extracellular matrix in varying shades of pink. This input contains critical structural information about the tissue and nuclei, which is essential for segmentation.

#### B. Pre-processing

In the U-Net-based architecture (as in the MoNuSeg Challenge), pre-processing becomes a very important step in making the input images more accurate and consistent, which allows the model to generalize well across datasets and cope well with the variations in the intensity, staining, and structure of the HE-stained histology images of nuclei.

##### **Image Normalization:**

Purpose: Histopathological images often have variations in color intensity due to differences in staining protocols. Normalization ensures that the pixel values lie within a consistent range, which makes the model robust against variations in brightness and contrast.

Types of Normalization:

1. Min-Max Normalization: Rescale pixel values to the range

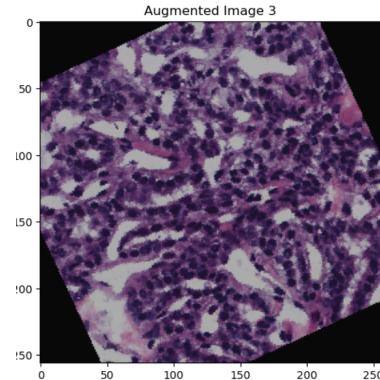


Fig. 4. Pre-processed image of Kidney Tissue

[0, 1].

$$\text{Formula: } I_{norm} = \frac{I - I_{min}}{I_{max} - I_{min}}$$

This makes the input data more uniform and easier for the neural network to process.

2. Z-score Normalization: Rescale pixel values to have zero mean and unit variance.

$$\text{Formula: } I_{norm} = \frac{I - \mu}{\sigma}$$

##### **Data Augmentation:**

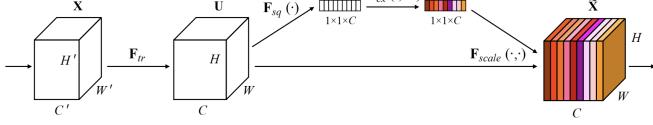


Fig. 5. Squeeze-and-Excitation block

Methods like rotation, flipping, and scaling were used to increase the diversity of the training data and improve model generalization.

#### Adding Channels:

adding channels allows the model to effectively represent and process complex image features, leading to improved segmentation and classification performance, especially for tasks like nucleus segmentation in histopathology images.

### C. U-Net Architecture

#### 1. Contracting Path (Encoder):

In the first convolutional layer (c1), the input image, with dimensions  $H \times W \times C$ , is processed by applying a 3 convolutional filter. This is followed by BatchNormalization to stabilize the learning process and ReLU activation to introduce non-linearity. A skip connection is used from the input to the output of this first convolutional block, which preserves spatial information for use later in the expansive path.

The **Squeeze-and-Excitation (SE)** block is then applied to recalibrate the channel-wise features before further convolution layers. In the Squeeze step, Global Average Pooling (GAP) is applied to aggregate the spatial information across each channel, producing a channel descriptor. This descriptor is passed through two fully connected layers (FC), with ReLU and Sigmoid activations, in the Excitation stage, to generate attention weights for each channel. In the Recalibration step, the output of the SE block is multiplied with the input tensor, recalibrating the feature map based on the learned channel-wise attention. The resulting output feature map c1 is then passed through MaxPooling2D for downsampling by a factor of 2 and Dropout is applied for regularization.

In the subsequent convolutional layers (c2, c3, c4), the contracting path continues with additional convolutional blocks. Each block follows the same structure as the first, applying convolutions, SE blocks, BatchNormalization, and ReLU activations, where the number of filters doubles at each step (e.g., 16 - 32 - 64 - 128). Each of these blocks also includes MaxPooling2D for spatial downsampling and Dropout for regularization, progressively reducing the spatial dimensions while increasing the depth of the feature maps.

#### 2. Bottleneck (ASPP Block):

After the contracting path, the **ASPP block** is applied to capture multi-scale contextual information. The ASPP block uses dilated convolutions with varying dilation rates to capture features at different scales, allowing the model to effectively learn both fine-grained and broader contextual information.

**Global Average Pooling (GAP)** is also applied within the block to capture global features from the entire image. The outputs of the dilated convolutions and the GAP are then concatenated, combining features from multiple scales, and passed through a final convolution to produce a unified set of features. The output of the ASPP block, which now contains multi-scale features, is then forwarded to the expansive path for further processing and reconstruction.

#### 4. Expansive Path (Decoder)

The expansive path of the model consists of **Transposed Convolutions** (also known as deconvolutions), which are used to upsample the feature maps. For instance, the first upsampling operation utilizes a **Conv2DTranspose** layer with strides of (2, 2), effectively doubling the spatial dimensions of the feature map. After upsampling, the corresponding skip connection from the contracting path, such as from c4, is concatenated with the upsampled feature map. This **skip connection** helps preserve spatial details from the contracting path, which is crucial for accurate segmentation. Additionally, **Dropout** is applied to regularize the network and prevent overfitting.

Each upsampled and concatenated feature map then passes through a series of convolutional layers similar to those in the contracting path. These layers consist of **Conv2D** operations, followed by **BatchNormalization**, **ReLU activation**, and **Squeeze-and-Excitation (SE) blocks**, which recalibrate the feature maps to enhance their representational power. As the network upscales the feature maps, the number of filters decreases progressively (e.g., 128 - 64 - 32 - 16) to refine the features.

Finally, after the expansive path, the output feature map is passed through a **1x1 convolution** with a **sigmoid activation** to produce the final segmentation mask. This mask has dimensions  $H \times W \times 1$  for **binary segmentation** or  $H \times W \times C$  for **multi-class segmentation**, where  $H$  and  $W$  are the spatial dimensions and  $C$  is the number of classes. The resulting segmentation mask represents the model's final output.

### D. Competition Metric and Loss

#### 1. Jaccard Distance Loss:

The Jaccard distance (also known as the Intersection over Union (IoU)) measures the overlap between two sets (in this case, the predicted segmentation mask and the ground truth mask). It is especially useful for segmentation tasks where the dataset is imbalanced (i.e., there is a disproportionate amount of foreground and background pixels).

#### Formula:

$$\text{Jaccard} = \frac{|X \cap Y|}{|X \cup Y|}$$

Where:

- $|X \cap Y|$  is the intersection of the predicted mask and ground truth mask.
- $|X \cup Y|$  is the union of the predicted mask and ground truth mask.

#### Loss Calculation:

- The intersection between the ground truth and the predicted mask is computed using  $K.sum(K.abs(y_{true} \times y_{pred}), axis = -1)$ .
- The sum of both masks is calculated with  $K.sum(K.abs(y_{true}) + K.abs(y_{pred}), axis = -1)$ .
- The Jaccard score is computed and then the loss is given by:

$$\text{Loss} = (1 - \text{Jaccard}) \times \text{smooth}$$

## 2. Dice Loss

The Dice coefficient is another metric used for measuring the overlap between two sets (similar to the Jaccard index). It is often used in segmentation tasks because it is well-suited for handling imbalanced datasets. The Dice loss function is derived from the Dice coefficient.

### Formula:

$$\text{Dice} = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

Where:

- $|X \cap Y|$  is the intersection of the predicted and ground truth masks.
- $|X|$  and  $|Y|$  are the areas of the predicted and ground truth masks, respectively.

### Loss Calculation:

- The numerator is calculated as  $2 \times \text{sum}(y_{true} \times y_{pred})$ , which computes the intersection of the two masks.
- The denominator is the sum of both masks,  $\text{sum}(y_{true} + y_{pred})$ .
- The loss is defined as:

$$\text{Loss} = 1 - \text{Dice coefficient}$$

## 3. F1 Score:

The F1 score is a metric that combines precision and recall into a single value, providing a balanced evaluation of the model's performance. It is particularly useful when the classes are imbalanced, as it accounts for both false positives and false negatives.

**Precision:** Precision measures how many of the predicted positives are actually true positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where:

- $TP$  is the number of true positives.
- $FP$  is the number of false positives.

### Recall:

Recall measures how many of the actual positives were correctly identified. It answers the question: "Out of all the true positive cases, how many were predicted correctly?"

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where:

- $TP$  is the number of true positives.

- $FN$  is the number of false negatives.

### F1 Score:

The F1 score is the harmonic mean of precision and recall, and it is particularly useful in scenarios where you need a balance between precision and recall, especially when the dataset is imbalanced.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

This formula balances the trade-off between precision and recall, giving a higher score when both precision and recall are high.

## E. Post-Processing

Watershed segmentation was employed to handle overlapping nuclei and improve instance segmentation.

## V. TRAINING AND TESTING

### A. Training

#### 1. Data Loading:

- The images are loaded and resized to a fixed size ( $im\_width \times im\_height$ ), and the pixel values are normalized between 0 and 1 by dividing by 255.0.

#### 2. Tiles/Patches Generation:

- Both images and masks are split into smaller patches using `view_as_windows`, a function that divides the image into overlapping tiles of size  $patch\_width \times patch\_height$ . This is done to enable training on smaller portions of large images and to reduce memory usage.

#### 2. Model Definition:

- The model is compiled using the Adam optimizer and a custom loss function (`jaccard_distance_loss`) with evaluation metrics such as accuracy, `dice_coefficient`, and `F1 score`.

#### 3. Model Training:

- The model is trained using the `fit()` method, with training data (`X_train, y_train`) and validation data (`X_test, y_test`).
- The model training includes:

- **EarlyStopping:** Stops training if the validation loss doesn't improve for a given number of epochs (`patience=10`).

- **ReduceLROnPlateau:** Reduces the learning rate by a factor of 0.1 if the validation loss doesn't improve after a specified number of epochs (`patience=10`).

- **ModelCheckpoint:** Saves the best model based on the validation `dice_coef`.

## Testing Process

### 1. Tile/Patch-based Testing

- The test image is loaded, resized, and normalized in the same way as the training images. The image is split into tiles using the `createTiles` function, which ensures that the image size is compatible with the model's input size.
- The model then performs prediction on these tiles, which are processed in batches.

### 2. Prediction Merging

- After the model generates predictions for each tile, these predictions are merged back into a single image using the `mergeTiles` function. This step reconstructs the full-size predicted image from the smaller patches.

### 3. Testing Process Visualization

- The output segmentation mask is saved and displayed with the filename indicating the predicted output.

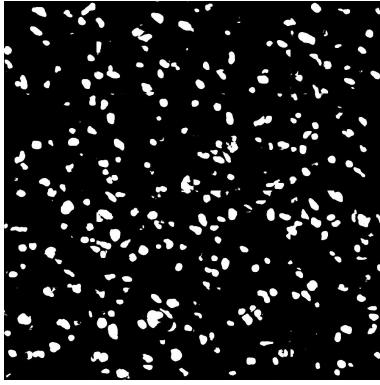


Fig. 6. Output: Segmentation Mask

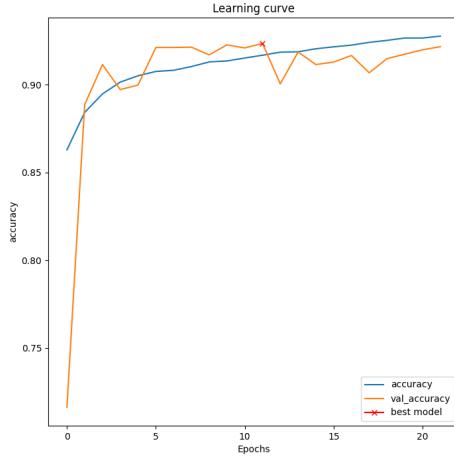


Fig. 7. Train Accuracy

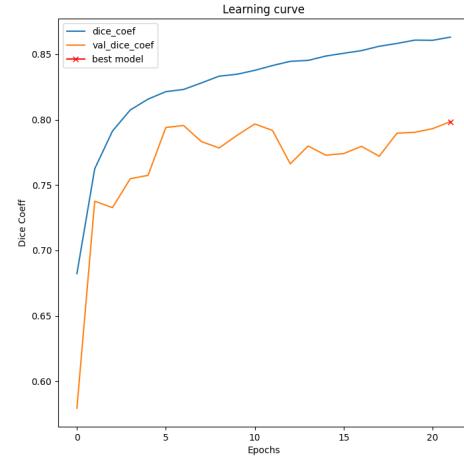


Fig. 8. Train Dice

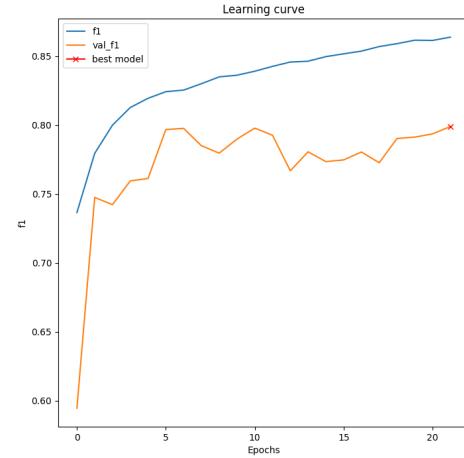


Fig. 9. Train F1

## CONCLUSION

In this work, we have implemented and demonstrated the effectiveness of a modified U-Net architecture, incorporating Squeeze-and-Excitation (SE) blocks for improved feature recalibration. This approach has shown significant potential in addressing the challenges of nucleus segmentation in histopathological images by dynamically enhancing channel-wise feature representations. The model effectively balances precision and recall, achieving robust segmentation performance on challenging datasets. Additionally, the use of advanced loss functions such as Jaccard distance and Dice loss ensures the model's adaptability to imbalanced data distributions.

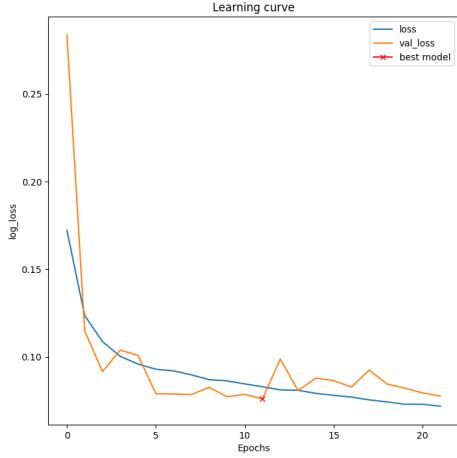


Fig. 10. Train Loss

#### FUTURE SCOPE

The proposed model can be extended and improved in several ways:

- **Integration with Transformers:** Incorporating attention mechanisms, such as Vision Transformers (ViT), could further enhance the model's ability to capture global contextual information.
- **Multi-class Segmentation:** Adapting the model for multi-class segmentation tasks could expand its applicability to broader domains, such as organ segmentation in medical imaging.
- **Self-Supervised Learning:** Employing self-supervised learning techniques could reduce dependency on large annotated datasets, making the model more versatile for diverse applications.

The exploration of these future directions will further enhance the applicability and performance of the model in medical image analysis and related fields.

#### VI.

#### REFERENCES

- [1] N. Kumar et al., "A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology," *IEEE Transactions on Medical Imaging*, vol. 36, no. 7, pp. 1550-1560, 2017.
- [2] S. Ali et al., "An Integrated Approach for Multiple Object Overlap Resolution in Histological Imagery," *IEEE Transactions on Medical Imaging*, vol. 31, no. 7, pp. 1448-1460, 2012.
- [3] J. H. Xue and D. M. Titterington, "t-Tests, F-Tests and Otsu's Methods for Image Thresholding," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2392-2396, 2011.
- [4] S. Kumar, "Development of Computer Aided Multi-organ Cancer Diagnostic System from Histopathological Images," RP4, National Institute of Technology, Rourkela, 2020.