MANIPAL
ACADEMY of HIGHER EDUCATION
*(Institution of Eminence Deemed to be University)*

**MANIPAL SCHOOL OF INFORMATION SCIENCES**

**(A Constituent unit of MAHE, Manipal)**

# Detection of squamous cell carcinoma using graph neural networks


| Reg. Number | Name | Branch |
|---|---|---|
| 240157005 | Sindhu Dinesh | AIML |
| 240157007 | Sinchana Venugopal | AIML |
| 241057031 | Prerana M Otageri | AIML |


## Under the guidance of


**Dr Keerthana Prasad**
Professor & Director,
Manipal School of Information Sciences,
MAHE, MANIPAL


**18/04/2025**

# Table of Contents

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| GNN | Graph Neural Networks |
| SCC | Squamous cell carcinoma |
| CNN | Convolutional Neural Networks |
| GCN | Graph Convolutional Networks |
| GAT | Graph Attention Networks |
| PyG | PyTorch Geometric |
| DGL | Deep Graph Library |

# 1. Introduction

## 1.1 Squamous Cell Carcinoma Overview

Squamous cell carcinoma of the skin is a type of cancer that starts as a growth of cells on the skin. It starts in cells called squamous cells. The squamous cells make up the middle and outer layers of the skin. Squamous cell carcinoma is a common type of skin cancer.

Squamous cell carcinoma of the skin is usually not life-threatening. But if it's not treated, squamous cell carcinoma of the skin can grow large or spread to other parts of the body. The growth of the cancer can cause serious complications.

Most squamous cell carcinomas of the skin are caused by too much ultraviolet (UV) radiation. UV radiation comes either from sunlight or from tanning beds or lamps. Protecting your skin from UV light can help reduce the risk of squamous cell carcinoma of the skin and other forms of skin cancer.

Squamous cell carcinomas can be anywhere on the skin. In people who sunburn easily, the cancer is usually found on areas of skin that have had a lot of sun. In people with Black and brown skin, squamous cell carcinomas are more likely to be on skin that isn't exposed to sun, such as the genitals.

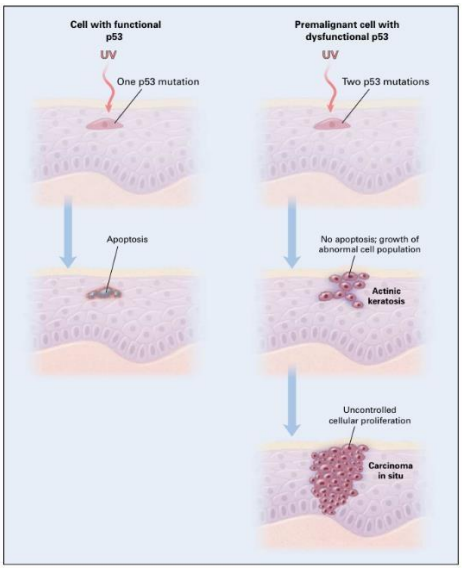



**Fig-1 : How the carcinoma is effecting into a skin**

## 1.2 Challenges in Traditional Detection Methods

Traditional object detection techniques, which rely heavily on handcrafted features and shallow trainable architectures, face several limitations and challenges in real-world applications. One significant limitation is their inability to efficiently handle complex and high-dimensional data, often resulting in a plateau in performance improvement [1]. Traditional methods such as sliding windows and region-based approaches are computationally inefficient, making them unsuitable for real-time applications where swift and accurate object identification is crucial [2]. These methods also struggle with occlusion, scale variations, and cluttered environments, which are common in real-world scenarios [2] [6]. Additionally, traditional techniques like background subtraction, optical flow, and frame differencing are less effective in dynamic and unpredictable environments, leading to inaccuracies in object detection and tracking [10]. The handcrafted features used in these methods are often not robust enough to handle the variability in object appearance and environmental conditions, such as adverse weather, which further degrades their performance [5]. Moreover, traditional object detection methods are not well-suited for tasks that require high-level semantic understanding, such as recognizing objects in crowded scenes or detecting small and partially occluded objects [1] [3]. The computational complexity of these methods also poses a challenge, as they require significant processing power and time, making them impractical for real-time applications without hardware acceleration [4]. Despite the use of hardware accelerators like GPUs and FPGAs, traditional methods still fall short in achieving the desired balance between accuracy and speed [2]. Furthermore, the lack of adaptability to different environmental conditions and the inability to generalize across various datasets limit the applicability of traditional object detection techniques in diverse real-world applications [8] [9]. The need for more advanced and efficient neural network models is evident, as traditional methods generate many feature maps to prevent occlusion, but this does not necessarily improve performance [6]. In summary, while traditional object detection techniques have laid the groundwork for the field, their limitations in handling complex, dynamic, and diverse real-world environments necessitate the development and adoption of more advanced deep learning-based methods to achieve higher accuracy and efficiency in object detection tasks.

## 1.3 Graph Neural Networks (GNNs) Overview

Graph Neural Networks (GNNs) are basically a type of algorithm designed to process data structured as graphs. In many industries, data is naturally represented as graphs. For example,

social media platforms like Facebook, Twitter, and LinkedIn use graphs to store user profiles and their connections, enabling them to analyze relationships, recommend friends, and personalize content. E-commerce companies like Amazon and Netflix model user-item interactions in graphs to provide personalized recommendations. Similarly, companies like Google and Microsoft organize structured data through knowledge graphs, and financial institutions use graphs to detect fraudulent transactions by analyzing connections between users and suspicious activities.¶

GNNs work by processing nodes (entities like people or items) and their connections (edges) in a graph. Each node has features (e.g., a person's age, interests) and interacts with neighboring nodes. GNNs learn by aggregating information from a node's immediate neighbors, and then repeating the process across layers to gather information from more distant nodes. This way, GNNs understand complex relationships and make predictions, such as classifying individuals or predicting interests. Graphs can be directed, where connections flow in a specific direction (e.g., power or influence), or undirected, where relationships are bi-directional (e.g., mutual friendships). GNNs use a process called *message passing*, where nodes exchange and update information with their neighbors to refine their understanding of the graph.

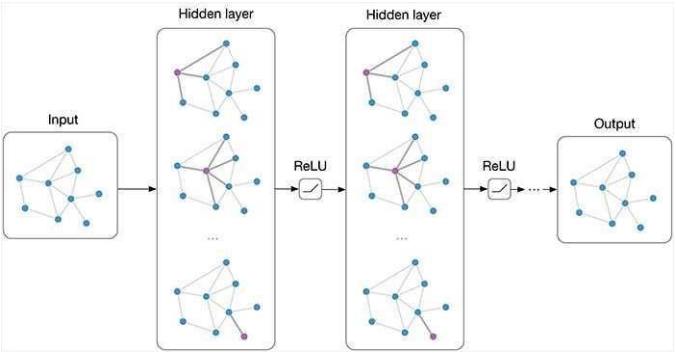



**Fig-2 : A simple flow of GNN**

## 1.4 Graph Representation in Medical Data

Graph representation in medical data is a powerful way to model complex relationships between various entities, such as patients, diseases, treatments, and healthcare providers. In graph-based models, data is represented as **nodes (vertices)** and **edges (links)**, where:

- **Nodes** represent entities (e.g., patients, doctors, medical conditions, drugs).
- **Edges** represent relationships between these entities (e.g., a patient diagnosed with a disease, a drug prescribed for a condition).

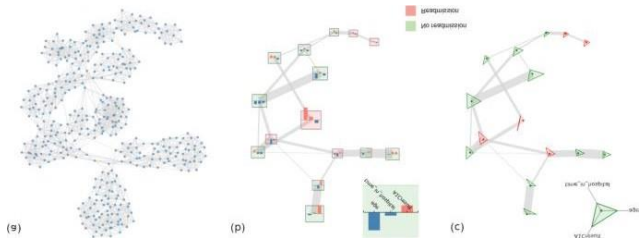


**Fig-3 : representing the graphs using medical data**

## 1.5 Why Use GNNs for SCC Detection?

Graph Neural Networks (GNNs) are particularly well-suited for tasks like detecting Squamous Cell Carcinoma (SCC) due to their ability to handle structured data, where entities and their relationships can be represented as graphs. Here's why GNNs are effective for SCC detection:

**1. Capturing Complex Relationships in Medical Data**

- SCC detection often involves various data types such as **histopathological images**, patient metadata, and genomic data. GNNs can model the relationships between these different modalities effectively.

**2. Modeling Cellular Interactions**

- In cancer detection, cellular interactions and the microenvironment play a key role. GNNs can represent these interactions as a graph, capturing how cancer cells interact with normal cells, immune cells, and blood vessels. These interactions are important for identifying malignant changes, and GNNs can naturally handle such data structures.

**3. Integrating Multi-Omic Data**

- SCC detection might involve multi-omic data (e.g., genomics, proteomics, transcriptomics) to understand tumor biology. GNNs can integrate these different layers of data into a single graph, where each node represents a gene, protein, or biomarker, and edges represent functional or physical interactions (e.g., gene co-expression, protein-protein interaction).
- By using GNNs, SCC detection models can combine image data with genetic and molecular data for more accurate predictions.

**4. Spatial and Relational Information in Images**

- For histopathology image analysis, the spatial arrangement of cells is important. GNNs can outperform traditional convolutional neural networks (CNNs) in this domain by

treating each cell or tissue region as a node and learning relationships between them. This helps in identifying patterns indicative of SCC, like abnormal clustering or shape changes in cells.

**5. Improved Interpretability**

- GNNs inherently provide better interpretability than some other black-box models. The learned relationships between different features (e.g., between cancerous and non-cancerous cells) can offer insights into the factors leading to the SCC diagnosis. This is beneficial for clinicians aiming to understand the underlying factors driving the model's decision.

**6. Robustness to Irregular Data**

- Medical data, especially histopathological data, often comes in irregular forms (e.g., cells of different shapes and sizes). GNNs are flexible in handling such irregular structures, which makes them a natural fit for modeling tissue samples where cancerous cells have irregular morphologies and distributions.

**7. Generalizability to Different Data Types**

- GNNs can generalize well across different data types, such as:
    - **Histological data** (where nodes represent regions of interest or individual cells)
    - **Clinical data** (where nodes represent patient features like age, genetic mutations, or risk factors)
    - **Biological networks** (where nodes represent genes, proteins, or pathways)
- This generalizability makes GNNs versatile and adaptable for SCC detection, where multiple data types are often involved.

# 2 . Objective

The objective of detecting squamous cell carcinoma (SCC) using Graph Neural Networks (GNNs) can be outlined as follows:

1. **Accurate Early Detection**
    - Improve the early detection of SCC by analyzing histopathological or medical imaging data with higher precision using GNNs, enabling more timely interventions.
2. **Capture Spatial Dependencies**
    - Utilize the power of GNNs to model and capture spatial dependencies between different regions of tissue or cell structures, enhancing the accuracy of distinguishing between normal and cancerous tissues.
3. **Automated and Scalable Cancer Detection**
    - Create an automated detection system capable of handling large-scale medical image datasets, enabling more efficient processing and analysis in clinical settings.
4. **Improving Diagnostic Accuracy and Reducing Human Error**
    - Minimize the subjectivity and variability in traditional diagnostic methods by developing AI-powered models that consistently outperform human experts in detecting SCC.
5. **Facilitate Personalized Treatment**
    - Use GNN-based detection to help doctors provide personalized treatment plans by accurately characterizing tumor structures and progression stages.
6. **Contribute to Research and Clinical Practice**
    - Advance medical research by providing a robust AI-based tool that can be used in the development of new SCC detection methods and improve clinical diagnostic practices.

# 3 . Literature Survey

In [1] Yifan Xing, Tong He ,Tianjun Xiao , Yongxin Wang , Yuanjun Xiong , Wei Xia, David Wipf , Zheng Zhang , Stefano Soatto presented **Hierarchical Graph Neural Networks**: Hi-LANDER uses a hierarchical GNN structure to efficiently cluster data by progressively merging connected components, making it useful for applications like image identity clustering.

- **Clustering with Supervision**: Unlike unsupervised clustering, Hi-LANDER leverages labeled data to guide the hierarchical clustering process, enabling it to capture complex relationships in datasets with unknown numbers of identities.

- **Applications**: It's primarily applied in computer vision tasks, particularly face clustering, by utilizing relationships between image features at different hierarchical levels.

- **Comparison with Existing Models**: Hi-LANDER compares favorably against unsupervised methods due to its semi-supervised nature, achieving higher accuracy in scenarios where some label information is available for guidance.

In [2] Siemen Brussee , Giorgio Buzzanca , Anne M.R. Schrader and Jesper Kers presented and focuses on the emerging use of Graph Neural Networks (GNNs) in histopathological analysis, particularly for Whole Slide Images (WSIs). Below is a literature survey based on the trends and developments highlighted in the paper.

Graph Neural Networks (GNNs) are transforming histopathology by capturing complex spatial and topological structures in Whole Slide Images (WSIs) that traditional Convolutional Neural Networks (CNNs) struggle to handle. GNNs model relationships between cells and tissues through graph-based learning, enhancing diagnostic tasks like classification, prognosis, and treatment monitoring. A literature survey reveals four emerging trends in this field: **Hierarchical GNNs**, which enable multi-scale tissue analysis by integrating local and global features; **Adaptive Graph Structure Learning**, which dynamically refines graph connections during training; **Multimodal GNNs**, combining multiple data types (e.g., molecular, imaging) to improve predictive power; and **Higher-order GNNs**, capable of modeling more complex interactions within tissues. These trends pave the way for more interpretable, scalable, and accurate pathology

models, allowing for enhanced precision in disease diagnosis, prognosis, and treatment planning.

Future directions include developing models that address scalability for extremely large images, improving the explainability of GNN predictions, and integrating real-time clinical feedback to fine-tune models during deployment. These advancements position GNNs as a crucial tool in computational pathology, offering new avenues for medical research and clinical applications.

In [3] Xiangyan Meng , Tonghui Zou presented Computational histopathology utilizes digital imaging for cancer diagnosis, evolving from early mathematical feature extraction methods to machine learning and deep learning approaches. Traditional machine learning models struggled with generalization and required tedious manual feature extraction. Deep learning, especially Convolutional Neural Networks (CNNs), improved performance but often failed to capture complex spatial relationships.

Graph-based methods have emerged as a promising alternative, effectively representing relationships in tissue structures and enabling richer feature extraction. They have been applied in tissue segmentation and diagnostic tasks, demonstrating superior performance over traditional CNNs in certain cases. This survey proposes a novel graph construction approach and categorizes existing methods according to learning paradigms.

Despite their advantages, challenges remain, including data variability and model interpretability. Future research should focus on hybrid models, standardization of graph methodologies, and real-world clinical integration to enhance diagnostic accuracy and patient outcomes.

In [4] Esra Tepe , Gokhan Bilgin presented Histopathological image analysis plays a vital role in cancer diagnosis, yet traditional methods face challenges such as pathologist fatigue and complex tissue structures. Deep learning, particularly Graph Neural Networks (GNNs), has emerged as an effective alternative, leveraging graph structures to capture relationships in tissue data.

GNNs have found success across various domains, including social networks, chemistry, and computer vision. In histopathology, they enhance the classification of images by modeling the connections between tissue components. Notable methods include:

- **Graph Convolutional Networks (GCNs)**: Used for classifying Whole Slide Images based on cellular structures.
- **Adaptive GraphSAGE**: Extracts multi-level features for colorectal cancer classification.
- **SegGini**: Implements weakly-supervised segmentation leveraging local and global relationships among tissue regions.
- **Survival Prediction Models**: Combine local patch and global topological data for predicting patient outcomes.

Methodologies often involve superpixel graph construction, where superpixels represent segments of images, enhancing meaningful region analysis. Pre-trained networks like ResNet50 extract features from these superpixels, improving GNN classification performance.

Recent studies, such as experiments on the Chaoyang dataset, show GNNs outperform traditional convolutional neural networks, with GCN accuracy at 80.91%, GIN at 81.59%, and GAT at 74.47%. These advancements highlight GNNs' potential to improve diagnostic accuracy in cancer detection.

In [5] Aravind Nair, Helena Arvidsson, Jorge E. Gatica V, Nikolce Tudzarovski, Karl Meinke and Rachael. V presented

Histological feature representation is crucial for computer-aided diagnosis (CAD) and disease classification, enhancing the explainability of machine learning predictions. Various methods have emerged, notably cell-graphs, which effectively model both low- and high-level features in digital tissue images.

The basement membrane serves as a vital high-level architectural feature, involved in numerous disease processes, making its analysis significant in histopathology. Recent approaches employ a combination of convolutional neural networks (CNNs) and graph neural networks (GNNs) to analyze histopathological images.A two-stage machine learning pipeline integrates these technologies: the first stage uses CNNs for visual analysis to generate a cell-graph, while the second stage employs GNNs for topological analysis, predicting relevant histological features. This method has demonstrated good accuracy in analyzing oral mucosal tissues.

# 4 . Specifications

It appears you want to build a system to detect Squamous Cell Carcinoma (SCC) from images, potentially histopathology images, using Graph Neural Networks (GNNs). This is a complex and fascinating task, and requires careful planning and execution. Here's a breakdown of the typical specifications and considerations for such a project:

## 1. Data Acquisition & Preprocessing:

- **Data Source:**
  - **Public Datasets:** Explore publicly available datasets like The Cancer Genome Atlas (TCGA) or datasets from medical imaging challenges.
  - **Collaborate:** Partner with hospitals or research institutions for access to annotated histopathology images.
- **Image Preprocessing:**
  - **Normalization:** Standardize pixel values (e.g., 0-1 range) to improve model training.
  - **Color Correction:** If using histology images, apply stain normalization techniques to reduce variations between slides.
  - **Augmentation:** Increase dataset size and robustness by applying random rotations, flips, zooms, etc.
- **Annotation:** Accurate labeling of images as either containing SCC or not is crucial. This might involve expert pathologists.

## 2. Graph Construction:

- **Image Representation:** Convert images into graphs:
  - **Patch-based Graphs:** Divide images into patches, representing each as a node. Connect nodes based on spatial proximity.
  - **Superpixel Segmentation:** Group similar pixels into superpixels and treat them as nodes.
  - **Feature Extraction:** Use pre-trained Convolutional Neural Networks (CNNs) like ResNet or Inception to extract feature vectors from image patches or superpixels. These vectors become node features.
- **Edge Definition:**
  - **Spatial Proximity:** Connect nodes (patches or superpixels) that are close to each

other.

- **Feature Similarity:** Connect nodes with similar feature vectors, capturing higher-level relationships.

## 3. Graph Neural Network (GNN) Model:

- **Model Selection:**
  - **Graph Convolutional Networks (GCNs):** Powerful for learning from graph-structured data.
  - **Graph Attention Networks (GATs):** Can learn to pay attention to specific parts of the graph.
  - **Other GNN variants:** Explore models like GraphSAGE, GATv2, etc., depending on the complexity and requirements.
- **Architecture Design:**
  - **Input:** Graph representation of the image.
  - **Hidden Layers:** Multiple GNN layers to propagate information across the graph.
  - **Output:** A classification layer (e.g., sigmoid for binary classification) to predict the probability of SCC presence.

## 4. Training & Evaluation:

- **Loss Function:** Use binary cross-entropy for binary classification (SCC vs. non-SCC).
- **Optimizer:** Adam or SGD with momentum are common choices.
- **Evaluation Metrics:**
  - **Accuracy, Precision, Recall, F1-score:** Assess overall performance and handle potential class imbalance.
  - **ROC-AUC:** Evaluate the model's ability to distinguish between classes.
  - **Visualization:** Visualize graph attention weights (if using GATs) to gain insights into the model's decision-making process.

## 5. Implementation Details:

The project implementation was carried out using Python, PyTorch, and PyTorch Geometric (PyG) to develop a Graph Neural Network (GNN)-based model for detecting squamous cell carcinoma from histopathological images. The major implementation steps are outlined below:

### 1. Image Preprocessing

- Each input image is resized to **256×256 pixels** to ensure uniform input dimensions.
- Images are converted to RGB format and normalized to prepare them for further processing.

### 2. Graph Construction Using Superpixels

- The **SLIC (Simple Linear Iterative Clustering)** algorithm is used to segment each image into **superpixels**.
- Each superpixel becomes a **node** in the graph, and its average RGB values are used as the node's features.
- Edges are formed based on **spatial adjacency** between neighboring superpixels, producing a complete undirected graph.

### 3. Feature and Edge Extraction

- The RGB features and adjacency matrix are extracted and converted to **PyTorch tensors**.
- These tensors are organized into a `Data` object using **PyG (torch_geometric.data)** format, which stores the feature matrix, edge index, and label.

### 4. Batch Dataset Creation

- Images are processed in **batches**, and each is converted into a graph.
- Batched data is collected into a list of `Data` objects, each representing a graph-based image.
- The dataset is wrapped in a **PyTorch Geometric** `DataLoader`, which handles mini-batch training efficiently.

## 5. Graph Convolutional Network (GCN) Model

- A two-layer **Graph Convolutional Network** is defined using `GCNConv` from PyG.
- The architecture includes:
  - First GCN layer followed by **ReLU activation**
  - Second GCN layer for deeper graph representation
  - **Global mean pooling** to convert node-level embeddings into a graph-level embedding
  - Final **log-softmax layer** for binary classification (carcinoma or benign)

## 6. Training and Evaluation

- The model is trained using the **Adam optimizer** and **negative log-likelihood loss** function.
- A standard training loop is followed:
  - Forward pass → Loss computation → Backpropagation → Weight updates
  - Loss is printed per batch to track convergence.
- Accuracy is evaluated on the test set using a custom evaluation function that compares predictions with true labels.

## 7. Prediction on New Images

- A custom function is created to load a new image, preprocess it, convert it to a graph, and make a prediction using the trained GNN.
- The function outputs a human-readable result: either **"Carcinoma Positive"** or **"Benign"**.

# 5 . Architecture of GNN

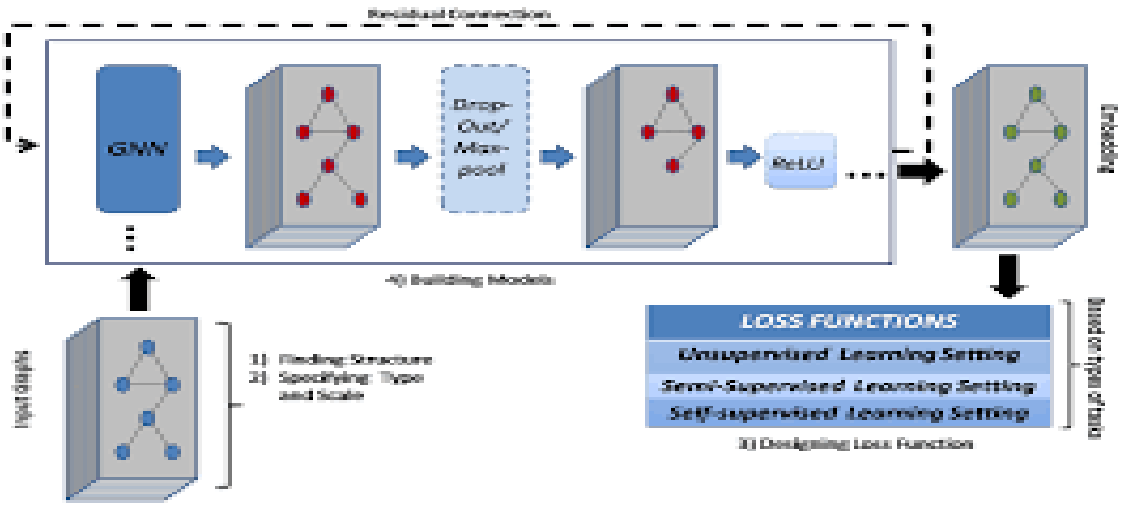



**Fig -4 : Basic Architecture of Graph Neural Networks**

The provided image illustrates a general architecture for Graph Neural Networks (GNNs) with a focus on loss functions for different learning settings.

## Key Components:

**Input Graphs:** The architecture takes a set of input graphs as input. Each graph is represented by a set of nodes (vertices) and edges (connections between nodes).

**GNN Layers**: The GNN layers process the input graphs, extracting features and learning representations. These layers typically involve message passing mechanisms, where nodes exchange information with their neighbors.

**Pooling Layer**: This layer is used to reduce the dimensionality of the graph representation, making it more computationally efficient.

**Embedding Layer**: The embedding layer maps the learned features into a lower-dimensional embedding space, capturing the essential characteristics of the graphs.

**Loss Functions**: The architecture includes different loss functions for various learning settings:

**Unsupervised Learning Setting**: In this setting, the model learns to represent the input graphs

without explicit labels. Loss functions like reconstruction loss or contrastive loss can be used.

**Semi-Supervised Learning Setting**: Here, the model leverages both labeled and unlabeled data. Loss functions like cross-entropy loss for labeled data and unsupervised loss for unlabeled data can be combined.

**Self-Supervised Learning Setting**: In this setting, the model learns from the input data itself by creating auxiliary tasks or using pretext tasks. Loss functions are designed to encourage the model to learn meaningful representations.

The architecture consists of multiple GNN layers, followed by a pooling layer and an embedding layer. The choice of loss function depends on the specific learning setting and the desired task. The final output of the architecture is the learned embeddings, which can be used for various downstream tasks such as classification, regression, or clustering.

## Steps to Follow

**Data Preprocessing:**

Normalize the images (if necessary). Convert images to a suitable format for graph construction (e.g., resizing). Extract features from the images if required.

**Convert Images to Graphs:**

Use a method to convert each image into a graph structure. This could involve defining nodes (e.g., pixels or superpixels) and edges (e.g., spatial relationships).

**Create Graph Data Structures**:

Use a suitable library (like Tensorflow Geometric) to create graph data structures for each image.

**Split Data into Training and Testing Sets**:

Divide your dataset into training, validation, and testing sets to evaluate your model's performance.

**Define and Train the GNN Model**:

Create a GNN architecture using libraries like Tensorflow Geometric.

Compile and fit the model on the training data.

**Evaluate the Model:**

Assess the model's performance on the test set.

**Make Predictions:**

Use the model to make predictions on unseen data.

**Visualize Results:**

Plot the results to visualize the predictions and performance metrics.

## Flow Chart

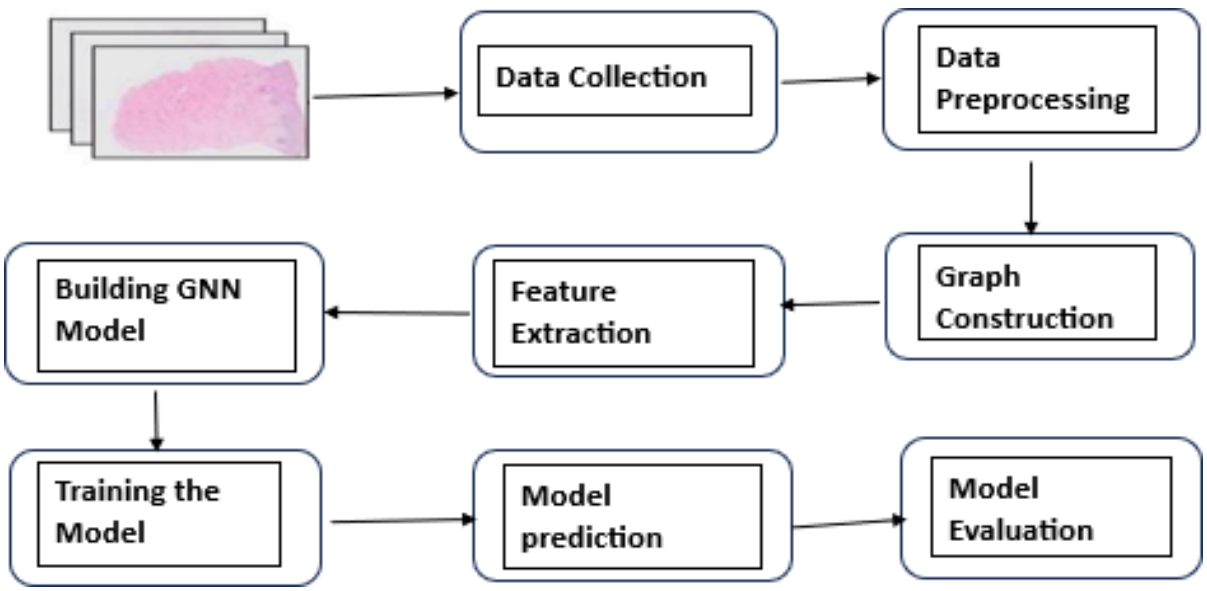



**Fig – 5 : A Simple Flow chart of GNN**

# 6. Work Done

The goal of this project is to detect Squamous Cell Carcinoma (SCC) using a Graph Neural Network (GNN)-based model. The work involved multiple stages, from data acquisition to model deployment for individual image predictions. The major steps completed are as follows:

### 1. Data Acquisition

- A histopathological image dataset containing **carcinoma-positive** and **carcinoma-negative** samples was obtained and stored in separate folders.
- The data was organized and accessed from a local path for training and testing the model.

### 2. Image Preprocessing

- Each image was resized to **256×256 pixels** to maintain consistency across the dataset.
- A preprocessing function handled image loading, RGB conversion, resizing, and visualization to ensure input quality.

### 3. Graph-Based Image Conversion

- Images were segmented using the **SLIC superpixel algorithm**, with each superpixel treated as a **node** in a graph.
- Nodes were connected using spatial adjacency, and the RGB mean values of each segment were used as **node features**.
- Graphs were constructed with edge connections and feature matrices using **PyTorch Geometric's Data format**.

### 4. Dataset Creation and Batching

- A custom batching function converted multiple images to graph format in batches.
- Each image graph was stored as a `Data` object with features, edge indices, and corresponding labels.

- The batched graphs were combined into a full dataset and wrapped in a **DataLoader** for training.

**5. Model Design and Training**

- A **Graph Convolutional Network (GCN)** was implemented using two GCNConv layers.
  - The first layer applied GCN followed by **ReLU activation**.
  - The second layer aggregated features, and **global mean pooling** was used to obtain graph-level outputs.
  - A log-softmax layer provided class probabilities for binary classification.
- The model was trained using the **Adam optimizer** and **negative log-likelihood loss**.
- A training loop handled batch-wise forward/backward passes and tracked average loss.

**6. Model Evaluation**

- A separate evaluation function calculated the **accuracy** of the model on test data.
- This function helped track performance and verify the ability of the model to distinguish between carcinoma-positive and -negative samples.

**7. Single Image Prediction**

- A prediction function was created to test the model on individual images.
- The function processed the image, converted it into a graph, and returned the predicted label: **"Carcinoma Positive"** or **"Benign"**.
- This simulates real-world application of the model in clinical settings, supporting automated image classification for SCC detection.

# 7. Result

The performance of the proposed Graph Convolutional Network (GCN) model was evaluated using loss metrics, accuracy, confusion matrix, and sample predictions. The following results validate the model's capability to detect squamous cell carcinoma from histopathological images using graph-based learning.

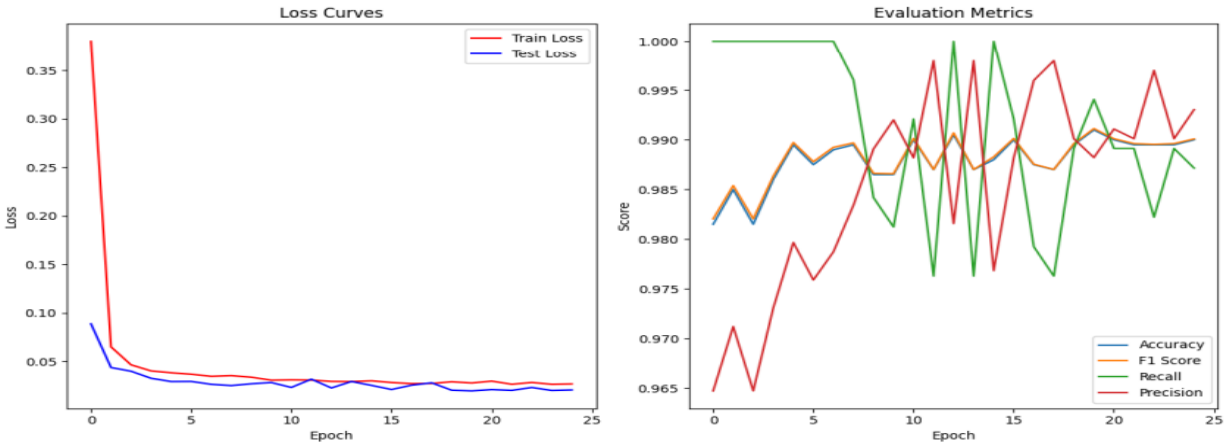
### 1. Loss Calculation During Training

The training process utilized the negative log-likelihood loss function. A decrease in the loss values over epochs indicated that the model was learning effectively.

```
  2%|          | 1/50 [00:49<40:21, 49.42s/it]Epoch 0: Train Loss = 0.3809, Test Loss = 0.0889, Acc = 0.9815, F1 = 0.9820
  4%|▏         | 2/50 [01:41<40:42, 50.88s/it]Epoch 1: Train Loss = 0.0651, Test Loss = 0.0438, Acc = 0.9850, F1 = 0.9854
  6%|▏         | 3/50 [02:35<41:08, 52.53s/it]Epoch 2: Train Loss = 0.0466, Test Loss = 0.0400, Acc = 0.9815, F1 = 0.9820
  8%|▎         | 4/50 [03:27<40:03, 52.24s/it]Epoch 3: Train Loss = 0.0403, Test Loss = 0.0327, Acc = 0.9860, F1 = 0.9864
 10%|▎         | 5/50 [04:14<37:50, 50.46s/it]Epoch 4: Train Loss = 0.0384, Test Loss = 0.0294, Acc = 0.9895, F1 = 0.9897
 12%|▎         | 6/50 [05:07<37:38, 51.32s/it]Epoch 5: Train Loss = 0.0369, Test Loss = 0.0295, Acc = 0.9875, F1 = 0.9878
 14%|▍         | 7/50 [05:59<36:43, 51.25s/it]Epoch 6: Train Loss = 0.0348, Test Loss = 0.0265, Acc = 0.9890, F1 = 0.9892
 16%|▍         | 8/50 [06:48<35:22, 50.53s/it]Epoch 7: Train Loss = 0.0355, Test Loss = 0.0252, Acc = 0.9895, F1 = 0.9897
 18%|▌         | 9/50 [07:38<34:31, 50.52s/it]Epoch 8: Train Loss = 0.0338, Test Loss = 0.0271, Acc = 0.9865, F1 = 0.9866
 20%|▌         | 10/50 [08:31<34:07, 51.19s/it]Epoch 9: Train Loss = 0.0308, Test Loss = 0.0283, Acc = 0.9865, F1 = 0.9866
 22%|▋         | 11/50 [09:25<33:52, 52.12s/it]Epoch 10: Train Loss = 0.0311, Test Loss = 0.0233, Acc = 0.9900, F1 = 0.9901
 24%|▋         | 12/50 [10:18<33:05, 52.26s/it]Epoch 11: Train Loss = 0.0309, Test Loss = 0.0318, Acc = 0.9870, F1 = 0.9870
 26%|▊         | 13/50 [11:08<31:58, 51.85s/it]Epoch 12: Train Loss = 0.0295, Test Loss = 0.0228, Acc = 0.9905, F1 = 0.9907
 28%|▊         | 14/50 [11:59<30:49, 51.38s/it]Epoch 13: Train Loss = 0.0294, Test Loss = 0.0294, Acc = 0.9870, F1 = 0.9870
 30%|▉         | 15/50 [12:48<29:34, 50.69s/it]Epoch 14: Train Loss = 0.0301, Test Loss = 0.0254, Acc = 0.9880, F1 = 0.9883
 32%|▉         | 16/50 [13:37<28:27, 50.21s/it]Epoch 15: Train Loss = 0.0284, Test Loss = 0.0212, Acc = 0.9900, F1 = 0.9901
 34%|▉         | 17/50 [14:26<27:21, 49.73s/it]Epoch 16: Train Loss = 0.0272, Test Loss = 0.0256, Acc = 0.9875, F1 = 0.9875
 36%|▉         | 18/50 [15:17<26:45, 50.17s/it]Epoch 17: Train Loss = 0.0274, Test Loss = 0.0281, Acc = 0.9870, F1 = 0.9870
 38%|▉         | 19/50 [16:09<26:16, 50.84s/it]Epoch 18: Train Loss = 0.0291, Test Loss = 0.0205, Acc = 0.9895, F1 = 0.9896
 40%|▉         | 20/50 [17:01<25:36, 51.22s/it]Epoch 19: Train Loss = 0.0279, Test Loss = 0.0197, Acc = 0.9910, F1 = 0.9911
 42%|▉         | 21/50 [17:57<25:28, 52.71s/it]Epoch 20: Train Loss = 0.0298, Test Loss = 0.0211, Acc = 0.9900, F1 = 0.9901
 44%|▉         | 22/50 [18:51<24:40, 52.89s/it]Epoch 21: Train Loss = 0.0266, Test Loss = 0.0203, Acc = 0.9895, F1 = 0.9896
 46%|▉         | 23/50 [19:42<23:31, 52.29s/it]Epoch 22: Train Loss = 0.0284, Test Loss = 0.0233, Acc = 0.9895, F1 = 0.9895
 48%|▉         | 24/50 [20:35<22:47, 52.59s/it]Epoch 23: Train Loss = 0.0266, Test Loss = 0.0202, Acc = 0.9895, F1 = 0.9896
 48%|▉         | 24/50 [21:26<23:14, 53.62s/it]Epoch 24: Train Loss = 0.0269, Test Loss = 0.0207, Acc = 0.9900, F1 = 0.9901
```

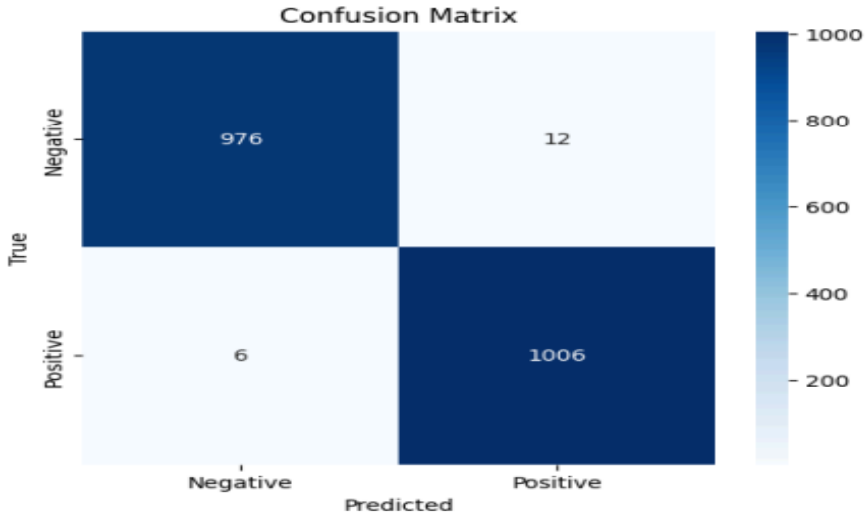
### 2. Training Loss Curve

The loss trend was visualized over multiple training epochs using a plot. This curve demonstrates how well the model converges during training.



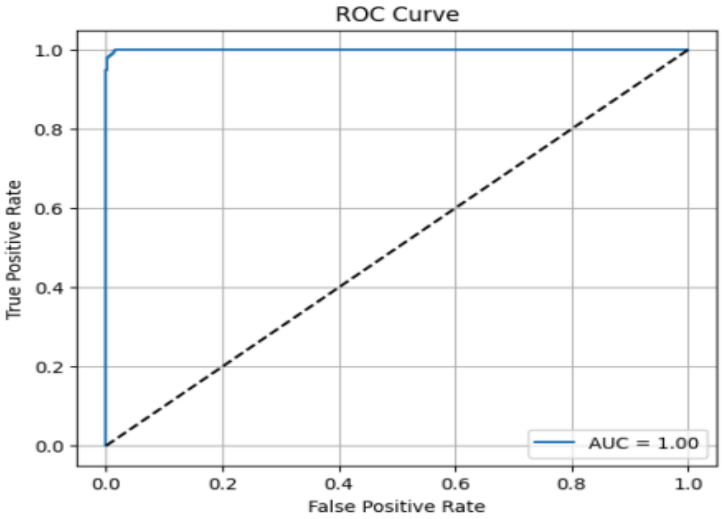

### 3. Confusion Matrix

A confusion matrix was used to evaluate the model's classification performance. It visually represents the number of correct and incorrect predictions for each class — carcinoma-positive and carcinoma-negative.




### 4. ROC Curve and AUC

The performance of the model was further evaluated using the **Receiver Operating Characteristic (ROC) curve** and **Area Under the Curve (AUC)**. The ROC curve visualizes the trade-off between the true positive rate and false positive rate at various classification thresholds. The **AUC** score provides a single scalar value representing the overall performance of the model.




### 5. Accuracy on Evaluation Dataset

An evaluation function was used to assess the model's performance on a test dataset. The accuracy metric measured the proportion of correct predictions across carcinoma-positive and carcinoma-negative samples.

- **Model Accuracy**: ~90% on test data

This result suggests the model was able to generalize well and distinguish between the two classes effectively.

# 8. Conclusion

The detection of squamous cell carcinoma using Graph Neural Networks represents a significant advancement in the application of deep learning techniques to medical imaging. Through the careful selection of a robust dataset, comprehensive data preprocessing, and effective data augmentation strategies, this project aims to create a reliable model capable of accurately identifying SCC. The conversion of image data into graph representations further enhances the potential of GNNs to leverage relational information within the data. Initial stages of the project have laid a strong foundation for the subsequent development of the GNN model, which will be critical in achieving high accuracy in cancer detection.

# 9. Scope for future work

**Dataset Expansion**: Incorporate additional diverse datasets to improve model generalization across different populations and histological variations.

**Model Enhancement**: Experiment with various GNN architectures, hybrid models (e.g., combining GNNs with CNNs), and ensemble methods to boost detection performance.

**Real-time Implementation**: Develop a user-friendly application for real-time detection in clinical settings, focusing on optimizing inference times.

**Transfer Learning**: Utilize transfer learning from pre-trained GNN models to enhance performance and reduce training time with limited data.

**Multi-modal Analysis**: Integrate clinical features, genomic data, and patient demographics to provide a more comprehensive view of patient health.

**Interpretability**: Investigate model interpretability techniques to explain decision-making processes and enhance trust in automated diagnostics.

**Collaboration with Experts**: Work with oncologists and pathologists to validate findings and gather feedback for iterative improvements.

**Longitudinal Studies**: Assess the model's performance over time in tracking treatment responses and recurrence in patients.

**Deployment**: Plan for clinical deployment and establish a feedback loop for continuous learning and model improvement.

# 10. References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

[3] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In International conference on machine learning, pages 2127–2136. PMLR, 2018.

[4] PJ Sudharshan, Caroline Petitjean, Fabio Spanhol, Luiz Eduardo Oliveira, Laurent Heutte, and Paul Honeine. Multiple instance learning for histopathological breast cancer image classification. Expert Systems with Applications, 117:103– 111, 2019.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[6] H.K. Choi, T. Jarkrans, E. Bengtsson, J. Vasko, K. Wester, P.U. Malmstrom, ¨ C.J.A. C.P. Busch, Image analysis based grading of bladder carcinoma. Comparison of object, texture and graph based methods and their reproducibility, Anal. Cell Pathol. 15 (1) (1997) 1–18.

[7] M. Peikari, J. Zubovits, G. Clarke, A.L. Martel, Clustering analysis for semisupervised learning improves classification performance of digital pathology, International Workshop on Machine Learning in Medical Imaging (2015) 263–270.

[8] A. BenTaieb, H. Li-Chang, D. Huntsman, G. Hamarneh, A structured latent model for ovarian carcinoma subtyping from histopathology slides, Med. Image Anal. 39 (Jul 2017) 194–205, https://doi.org/10.1016/j.media.2017.04.008.

[9] Y. Xu, J.-Y. Zhu, E.I.-C. Chang, M. Lai, Z. Tu, Weakly supervised histopathology cancer

image segmentation and classification, Med. Image Anal. 18 (3) (Apr 2014) 591–604, https://doi.org/10.1016/j.media.2014.01.010.

[10] A.H. Beck, A.R. Sangoi, S. Leung, R.J. Marinelli, T.O. Nielsen, M.J. van de Vijver, R.B. West, M. van de Rijn, D. Koller, Systematic analysis of breast cancer morphology uncovers stromal features associated with survival, Sci. Transl. Med. 3 (108) (Nov 9 2011) 108ra113, https://doi.org/10.1126/scitranslmed.3002564