Deep Learning for Cancer Cell Classification: A CNN Approach to Identifying Adenocarcinoma, Benign, and Squamous Cell Carcinoma

1. Introduction

Cancer detection and classification are crucial tasks in medical diagnosis and research. Accurate classification of cancer types can significantly enhance diagnosis and treatment planning. In this study, we developed a convolutional neural network (CNN) model to classify three types of cancer cells:

- Adenocarcinoma
- Benign
- Squamous Cell Carcinoma

The primary objective of this research is to build an efficient and robust deep learning model that can classify different cancer cell types from microscopic images**.**

---

2. Dataset Description

The dataset used for this study contains labeled images of three types of cancer cells. The dataset was divided into training and testing sets in an 80:20 ratio to ensure model robustness.

- Image Size: 128x128 pixels
- Number of Classes: 3 (Adenocarcinoma, Benign, Squamous Cell Carcinoma)
- Batch Size: 32
- Data Preprocessing:
    - Image normalization by dividing pixel values by 255.
    - Batch generation using Keras image_dataset_from_directory.
    - Image resizing to a fixed size of 128x128 pixels.

---

3. Model Architecture

A Convolutional Neural Network (CNN) architecture was employed for this classification task. The model's architecture is as follows:

- Input Layer: (128, 128, 3)

- Conv2D Layer (64 filters, kernel size 3x3, ReLU activation)
- MaxPooling Layer (2x2)
- Conv2D Layer (32 filters, kernel size 3x3, ReLU activation)
- MaxPooling Layer (2x2)
- Conv2D Layer (16 filters, kernel size 3x3, ReLU activation)
- MaxPooling Layer (2x2)
- Flatten Layer
- Fully Connected Layer (128 neurons, ReLU activation, Dropout 0.3)
- Fully Connected Layer (64 neurons, ReLU activation, Dropout 0.3)
- Fully Connected Layer (32 neurons, ReLU activation, Dropout 0.3)
- Fully Connected Layer (16 neurons, ReLU activation, Dropout 0.3)
- Output Layer (3 neurons, Softmax activation)

---

4. Model Training

The model was compiled with the following configurations:

- Loss Function: Sparse Categorical Cross-entropy
- Optimizer: Adam (Learning Rate: 0.001)
- Evaluation Metric: Accuracy

Training was performed for 20 epochs with a batch size of 32, using early stopping and model checkpointing to prevent overfitting.

---

5. Model Performance

- **Training Accuracy: 100%**
- **Training Loss: 0.0261**
- **Testing Accuracy: 70.83%**
- **Testing Loss: 0.8976**

---

6. Results and Observations

- The model achieved high accuracy on the training data (100%), but the test accuracy was around 70.83%.

- **The significant difference between training and testing accuracy indicates overfitting, likely due to:**
    - Insufficient data volume for generalization.
    - Complexity of the model architecture relative to the dataset size.
    - Lack of data augmentation to improve generalization.

---

7. Challenges and Limitations

- Overfitting Issue: Despite achieving perfect training accuracy, the test accuracy was significantly lower, indicating that the model memorized training data rather than learning robust patterns.
- Data Imbalance: The class distribution may not be equal, leading to model bias.
- Image Variability: Differences in lighting and image quality might have affected model performance.

---

8. Suggestions for Improvement

1. **Data Augmentation:**
    - Techniques like rotation, zooming, flipping, and shifting could be applied to increase dataset diversity**.**
2. **Model Regularization:**
    - Use techniques like Batch Normalization and L2 Regularization to improve generalization.
3. **Transfer Learning:**
    - Use pre-trained models such as VGG16, ResNet50, or InceptionV3 to leverage existing knowledge.
4. **Cross-Validation:**
    - Implement k-fold cross-validation to assess model stability and robustness.
5. **Model Tuning:**
    - Experiment with learning rate adjustments and early stopping based on validation loss.

---

9. Conclusion

The CNN model demonstrated high accuracy during training but struggled with generalization, as reflected in the testing accuracy. This discrepancy highlights the need for data augmentation and regularization. Future work should focus on leveraging transfer learning techniques and optimizing the model architecture to improve generalization and robustness.
