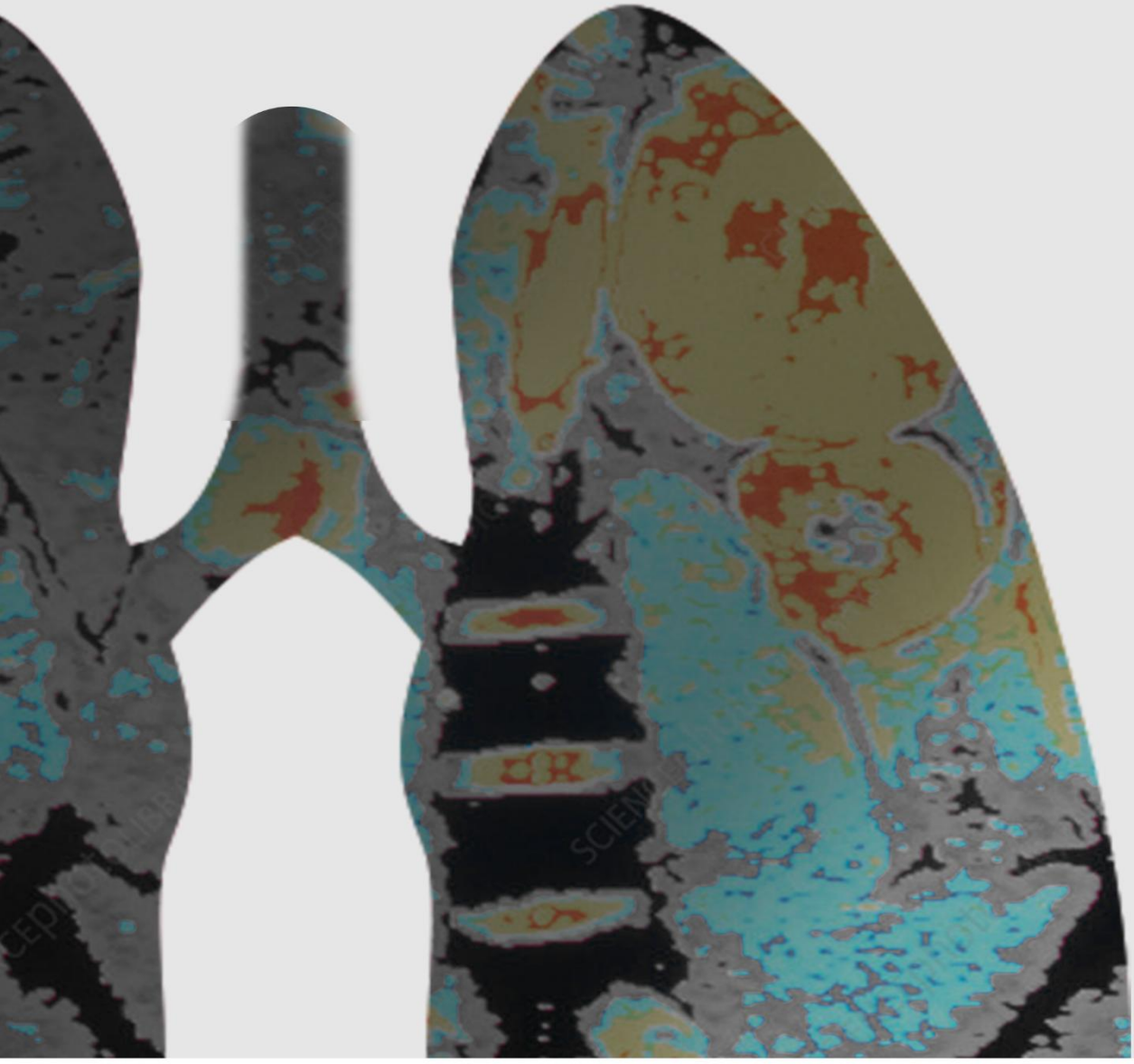
Presented to: Prof. Nashaw El bendary

2024/12/18

Lung Cancer

Deep Learning-Based Classification of Lung Cancer Types Using CNN Models



Presented by:
Amr Ashraf
Abdulrahman Ahmed

Arab Academy for Science and Technology
College of Computer and Information Technology

Table of Contents

Table of figures

Deep Learning-Based Histopathological Classification of Lung Cancer Types Using Augmented CNN Models

Problem Statement

Lung cancer is one of the leading causes of cancer-related deaths globally, accounting for 1.8 million deaths in 2020 alone [1]. Early and accurate diagnosis is crucial for improving patient outcomes and survival rates [2]. Histopathological examination of lung tissue remains the gold standard for diagnosing and classifying lung cancer types [3]. However, this manual process is time-consuming and subject to inter-observer variability, relying heavily on the expertise of pathologists [4].

Advancements in deep learning have the potential to revolutionize medical imaging by providing automated, accurate, and efficient diagnostic tools [5]. In this project, we aim to develop robust convolutional neural network (CNN) models to classify different types of lung cancer from histopathological images, specifically distinguishing between:

Adenocarcinoma

Squamous Cell Carcinoma

Benign Lung Tissue

Significance:

Implementing such models could significantly reduce the diagnostic workload on pathologists and expedite the treatment planning process for patients. Early detection and precise classification can lead to timely interventions, potentially improving survival rates and quality of life for patients battling lung cancer.

Dataset Description

The dataset used in this project is the **Lung and Colon Cancer Histopathological Image Dataset (LC25000)**, published by Borkowski et al. [6]. The original dataset is available at GitHub [7]. It consists of 25,000 color images divided into five classes:

- Lung Adenocarcinoma
- Lung Squamous Cell Carcinoma
- Lung Benign Tissue
- Colon Adenocarcinoma
- Colon Benign Tissue

Dataset samples:

Adenocarcinoma

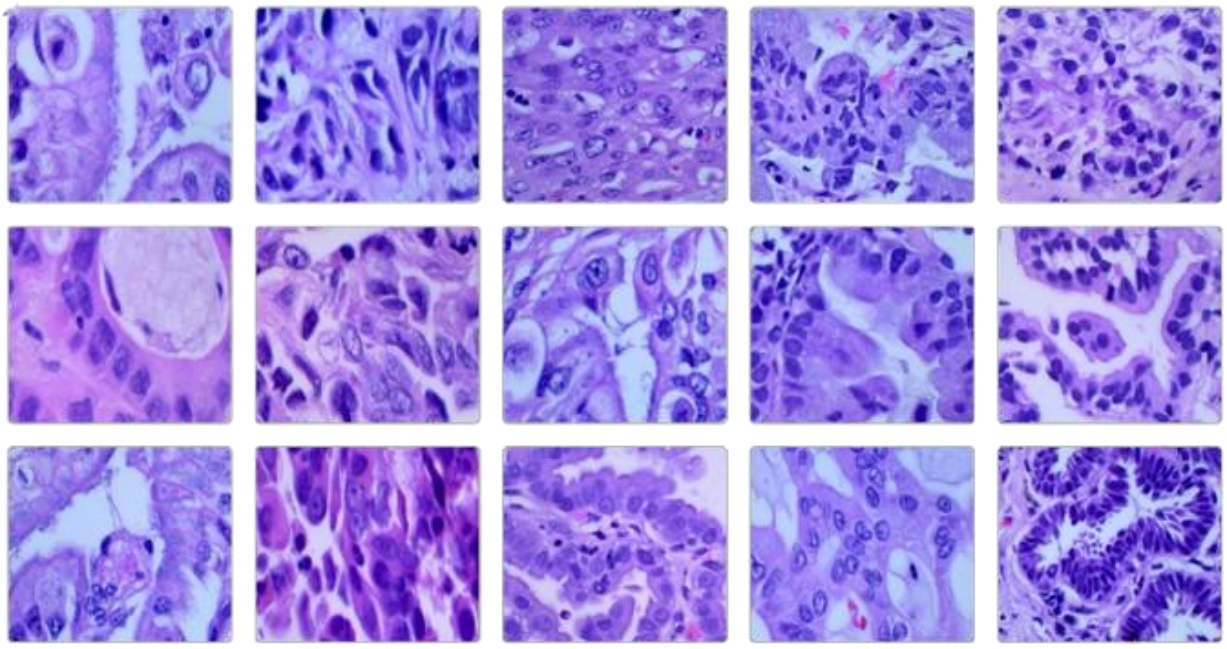


Figure 1. sample 15 images of adenocarcinoma lung cancer.

Benign

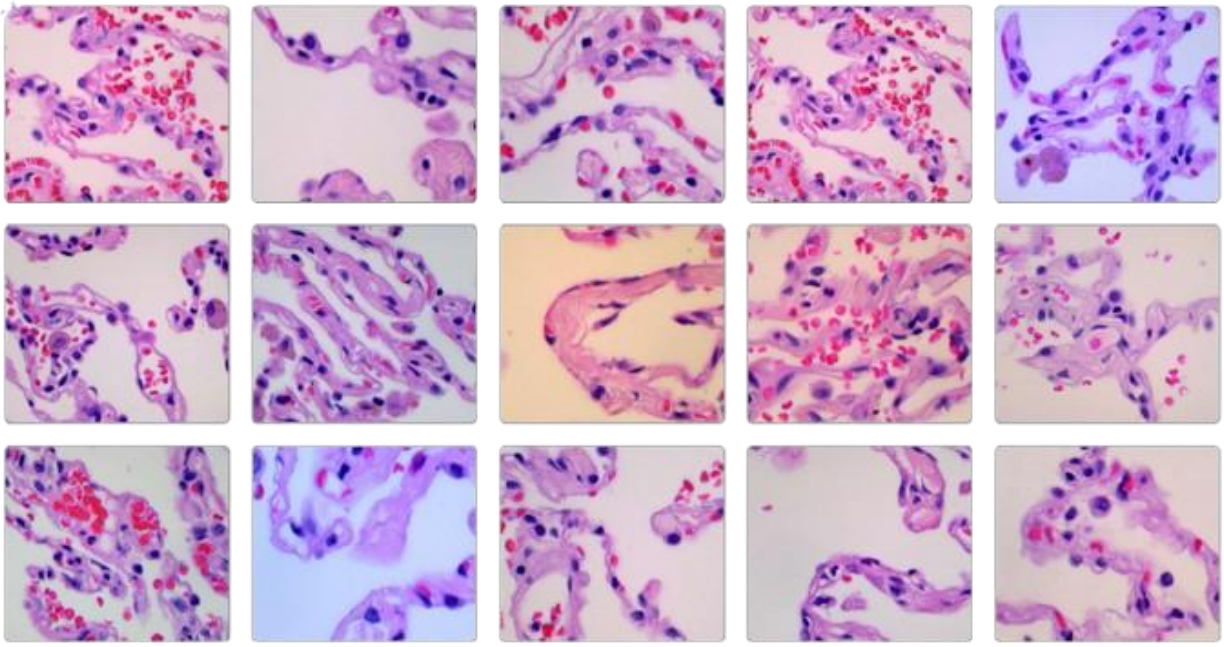


Figure 2.sample 15 images of benign lung cancer.

squamous cell carcinoma

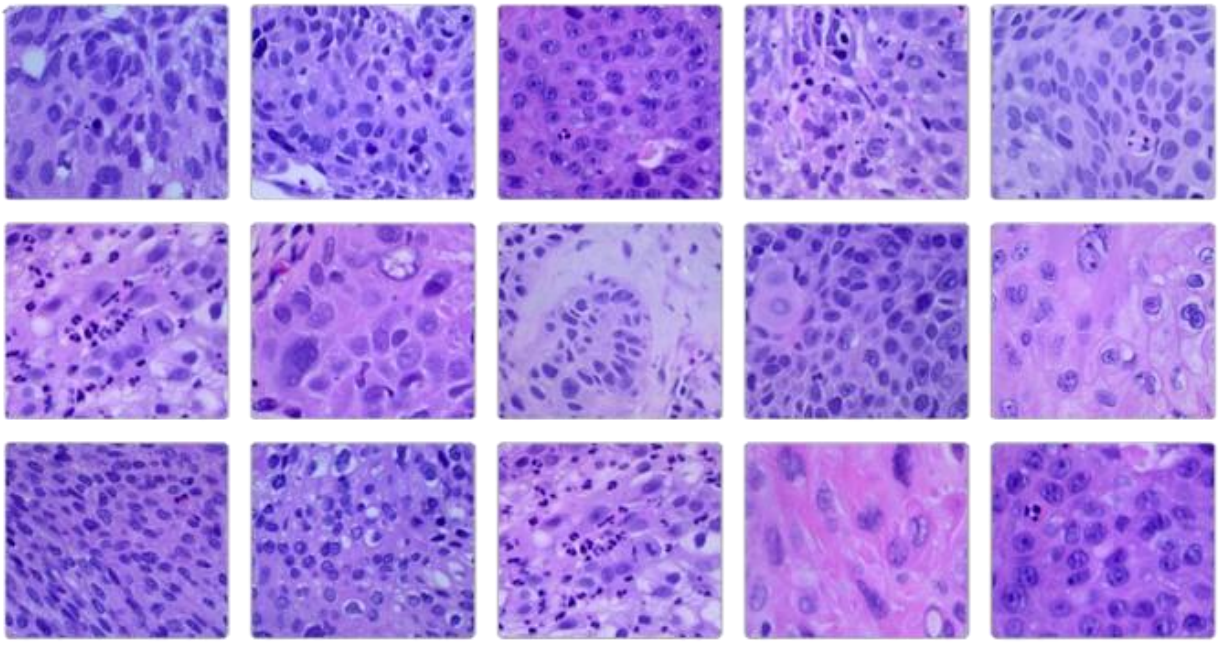


Figure 3. sample 15 images of squamous cell carcinoma lung cancer.

Project-Specific Data Selection:

For this project, we focus on the lung tissue images, specifically:

- **Adenocarcinoma**
- **Squamous Cell Carcinoma**
- **Benign Lung Tissue**

Each class contains 5,000 images, totaling 15,000 images for analysis.

Data Augmentation:

Initially, we trained our models using the original dataset without augmentation. The performance metrics were as follows:

Metric,	Value
Validation Accuracy,	0.3333333432674408
Weighted F1 Score,	0.16666666666666666
Weighted Precision,	0.1111111111111111
Weighted Recall,	0.3333333333333333

Figure 4. screen shot for F_score measures of resnet50 model on the data set without augmentation

These results indicate poor model performance, equivalent to random guessing, suggesting that the model was unable to learn meaningful patterns from the data.

To enhance the dataset and improve model performance, we applied data augmentation techniques, which have been shown to be effective in deep learning applications [8]. The augmentation methods included:

- Random rotations
- Width and height shifts
- Shear transformations
- Zoom operations
- Horizontal flips

- Random contrast adjustments

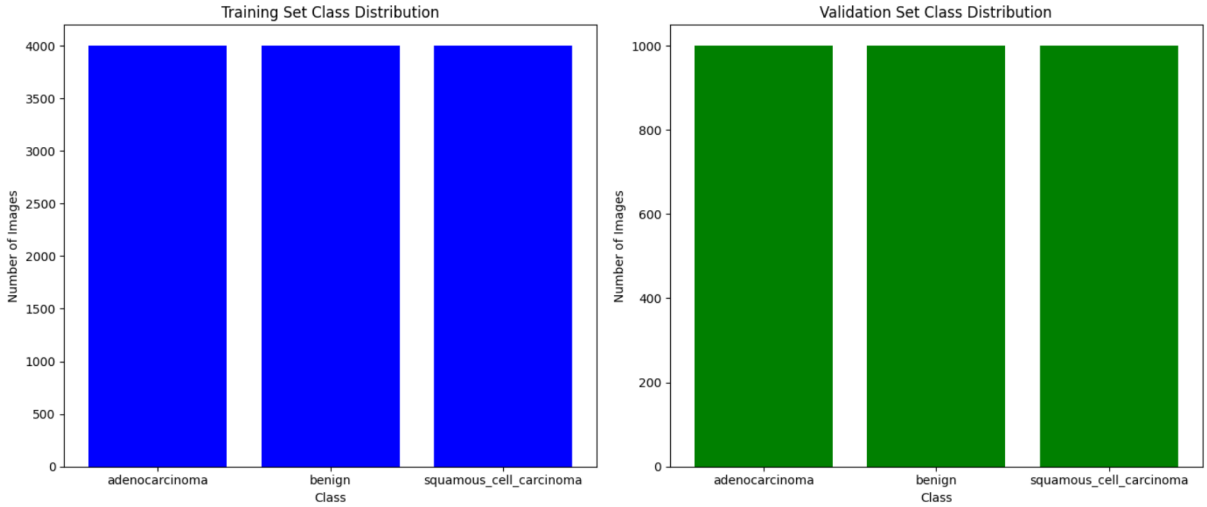


Figure 5. Data class distribution after augmentation

Reason for Augmentation:

Data augmentation increases the diversity of the training data without collecting new data, helping the model generalize better to unseen images and reducing overfitting , [8] [9].

Proposed Models

We implemented and compared two CNN architectures based on pre-trained models:

1. **ResNet50-based Model**
2. **DenseNet121-based Model**

These architectures utilize transfer learning, leveraging models pre-trained on the ImageNet dataset [10], which allows for faster convergence and better performance due to the use of previously learned features.

ResNet50-based Model

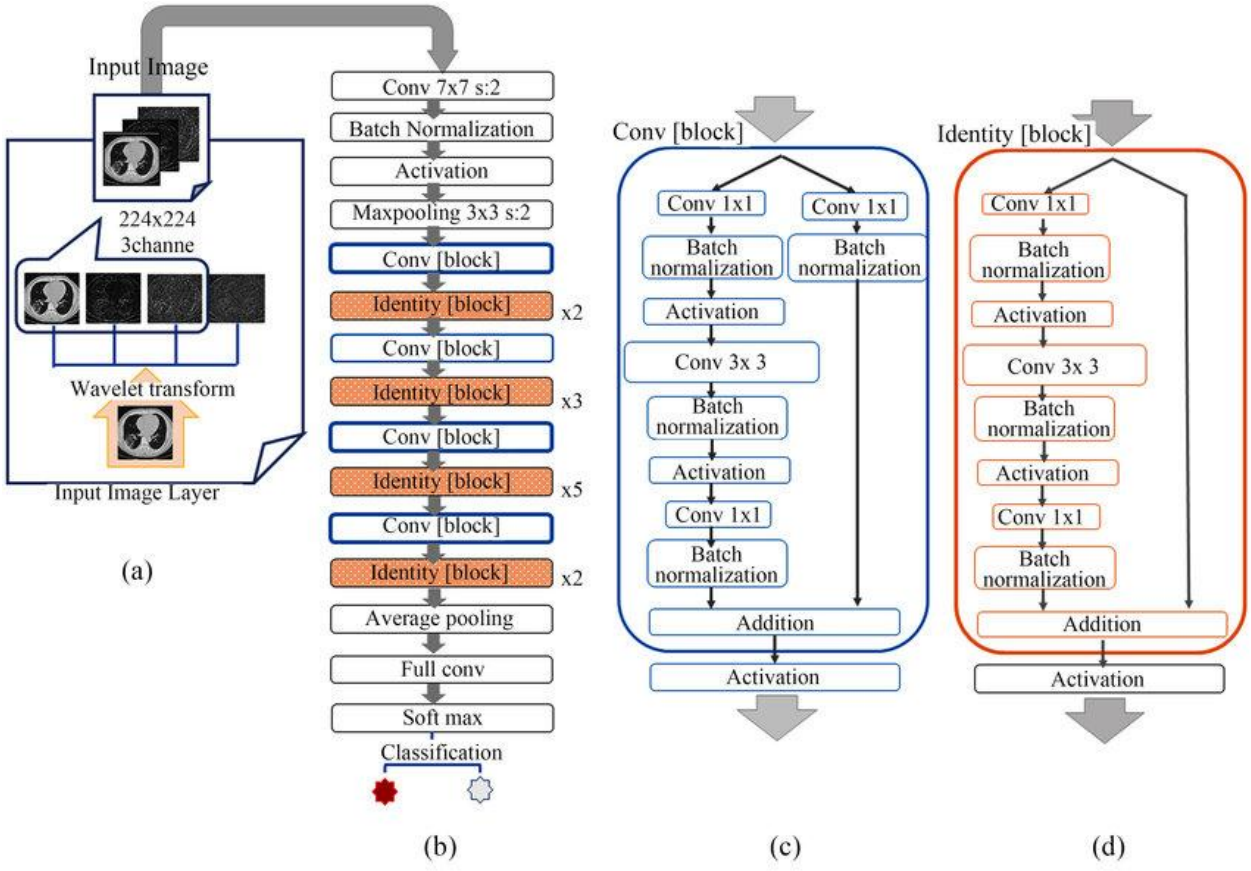


Figure 6. Outline-of-ResNet-50-architecture-a-A-3-channel-image-input-layer-The-LL-LH-and-HH [11]

Architecture Description:

Base Model: ResNet50 without the top classification layers (include_top=False).
Input Shape: (150, 150, 3).
Custom Top Layers:
Global Average Pooling Layer: Reduces the dimensions before the fully connected layers.
Dense Layer: 1,024 units with ReLU activation.
Dropout Layer: 50% rate to mitigate overfitting.
Dense Layer: 512 units with ReLU activation.
Dropout Layer: 30% rate.
Output Layer: 3 units with softmax activation for multiclass classification.

Fine-Tuning:

The last 30 layers of the ResNet50 base model were unfrozen to allow the model to learn more specific features related to our dataset.

Model Compilation and Training:

Optimizer: Adam with a learning rate of 0.0001.
Loss Function: Categorical Crossentropy.
Metrics: Accuracy.
Callbacks: EarlyStopping and ModelCheckpoint based on validation loss.
Epochs: 10.

ResNet50 Overview:

ResNet50, introduced by He et al. [12], is a 50-layer deep CNN that uses residual learning to ease the training of networks substantially deeper than previous networks, by adding shortcut connections that skip one or more layers.

DenseNet121-based Model

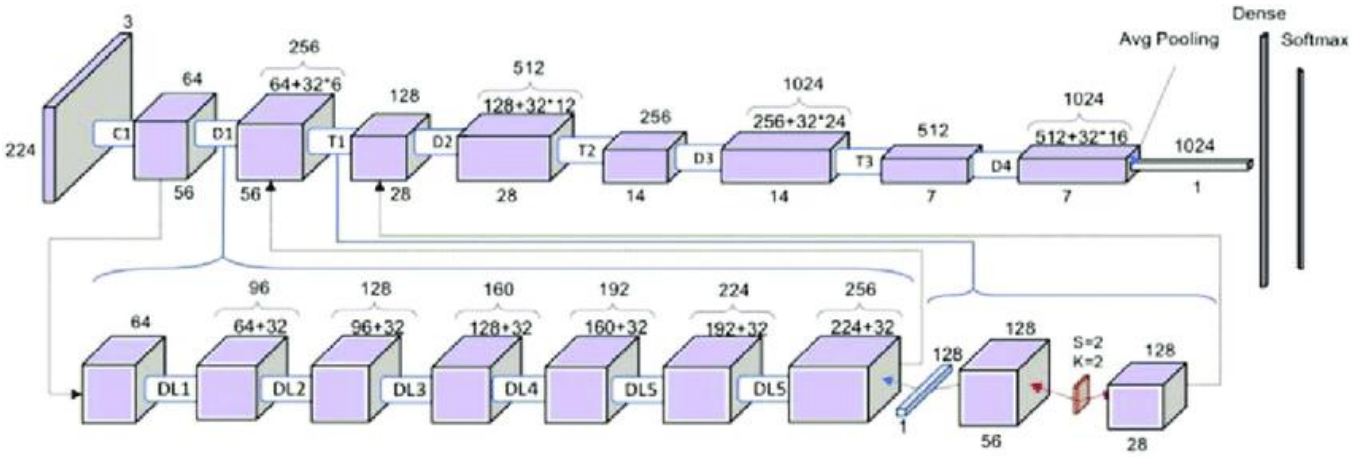


Figure 7. The-architecture-of-DenseNet121-with-Dense-block-D-Transition-blocks-T-and-Dense [13]

Architecture Description:

Base Model: DenseNet121 without the top classification layers (include_top=False).
Input Shape: (150, 150, 3).
Custom Top Layers:
Global Average Pooling Layer.
Dense Layer: 1,024 units with ReLU activation.
Dropout Layer: 50% rate.
Dense Layer: 512 units with ReLU activation.
Dropout Layer: 30% rate.
Output Layer: 3 units with softmax activation.

Fine-Tuning:

Unfrozen the last 30 layers of the DenseNet121 base model.

Model Compilation and Training:

Optimizer: Adam with a learning rate of 0.0001.
Loss Function: Categorical Crossentropy.
Metrics: Accuracy.
Callbacks: EarlyStopping and ModelCheckpoint.
Epochs: 10.

DenseNet121 Overview:

DenseNet121, proposed by Huang et al. [14], introduces densely connected layers where each layer is connected to every other layer in a feed-forward fashion. This design alleviates the vanishing-gradient problem, strengthens feature propagation, and substantially reduces the number of parameters.

Models Implementation

Code Snippet for ResNet50-based Model:

```
# Load the ResNet50 model, excluding the top layers
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(150, 150, 3))

# Add custom layers on top of the base model
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
predictions = Dense(3, activation='softmax')(x)

# Create the final model
model = Model(inputs=base_model.input, outputs=predictions)

# Unfreeze some layers of the base model for fine-tuning
for layer in base_model.layers[-30:]:
    layer.trainable = True

# Compile the model with a lower learning rate
optimizer = Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

# Define callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
check_point = ModelCheckpoint('../models/best_model.h5', save_best_only=True)

# Train the model with callback
history = model.fit(
    train_generator,
    validation_data=validation_generator,
    epochs=10,
    callbacks=[early_stopping, check_point]
)
```

Figure 8. Code Snippet for ResNet50-based Model

Code Snippet for DenseNet121-based Model:

```
# Load the ResNet50 model, excluding the top layers
base_model = DenseNet121(weights='imagenet', include_top=False, input_shape=(150, 150, 3))

# Add custom layers on top of the base model
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
predictions = Dense(3, activation='softmax')(x)

# Create the final model
model = Model(inputs=base_model.input, outputs=predictions)

# Unfreeze some layers of the base model for fine-tuning
for layer in base_model.layers[-30:]:
    layer.trainable = True

# Compile the model with a lower learning rate
optimizer = Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

# Define callbacks
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
check_point = ModelCheckpoint('../models/best_model.h5', save_best_only=True)

# Train the model with callback
history = model.fit(
    train_generator,
    validation_data=validation_generator,
    epochs=10,
    callbacks=[early_stopping, check_point]
)
```

Figure 9. Code Snippet for DenseNet121-based Model

Results and Discussion

Performance Metrics After Augmentation:

Metric	ResNet50 Model	DenseNet121 Model
Validation Accuracy	0.996	0.998
Weighted F1 Score	0.996	0.998
Weighted Precision	0.996	0.998
Weighted Recall	0.996	0.998

Confusion matrix:

Resnet50-based model:

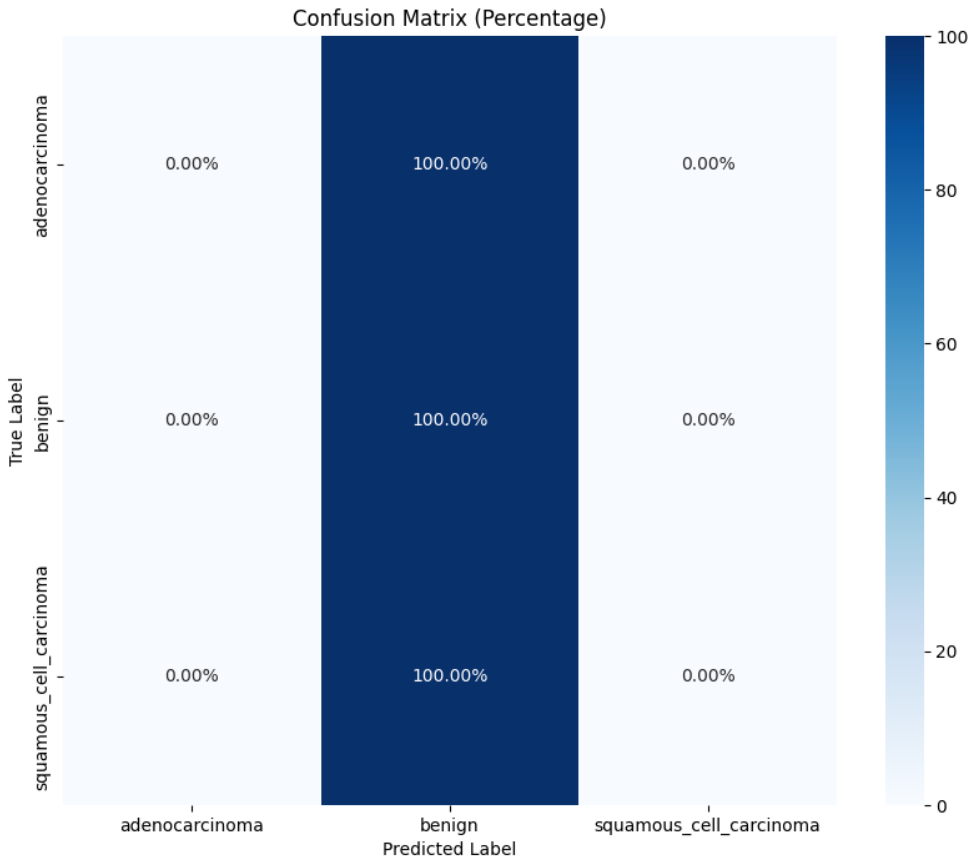


Figure 10. lung_cancer-resnet50 not augmented

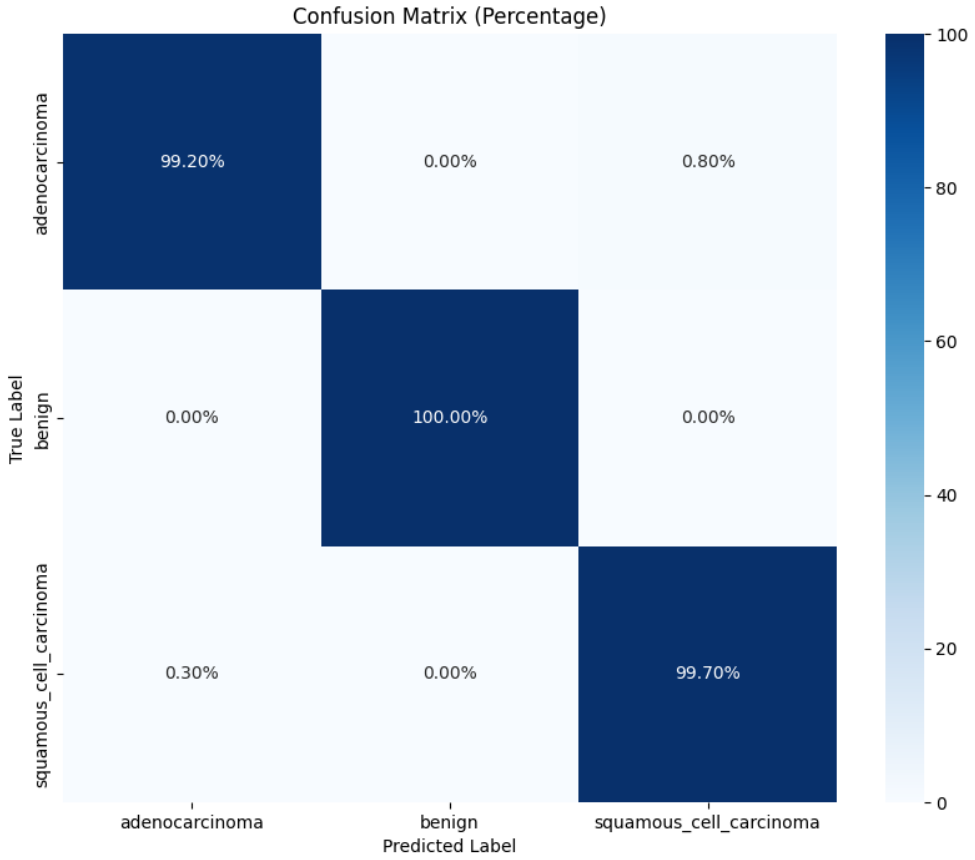


Figure 11. lung_cancer-resnet50-augmneted

Densenet121-based model:

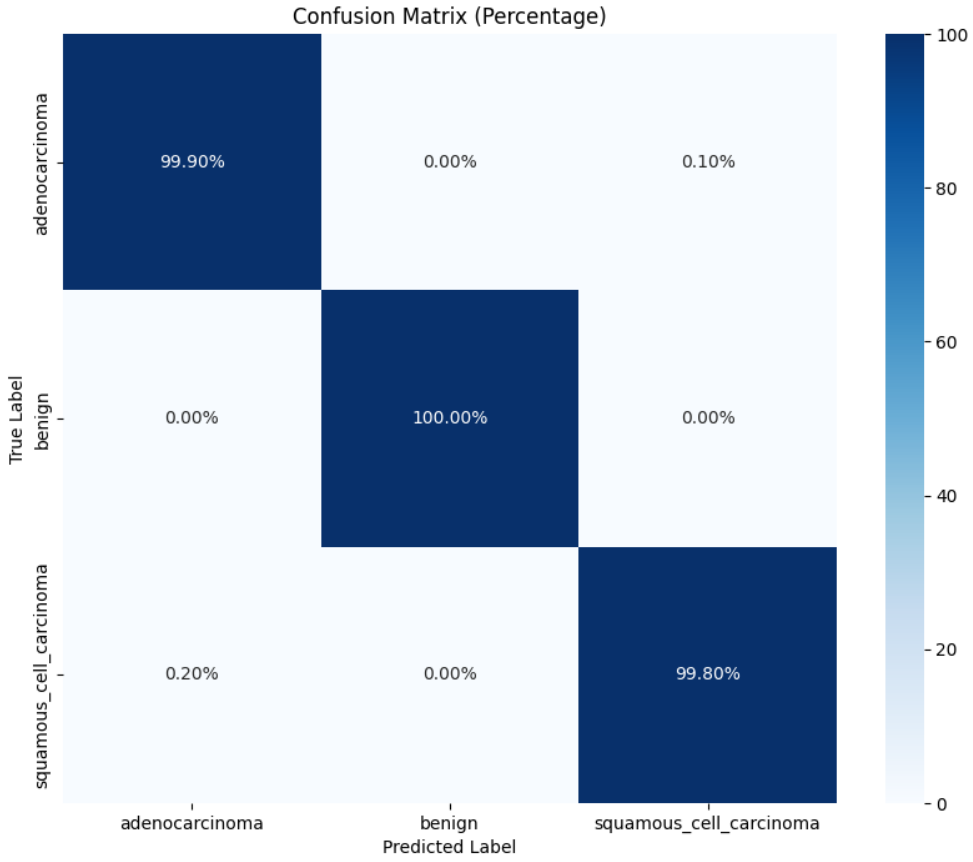


Figure 12. lung_cancer-densenet121 not augmented

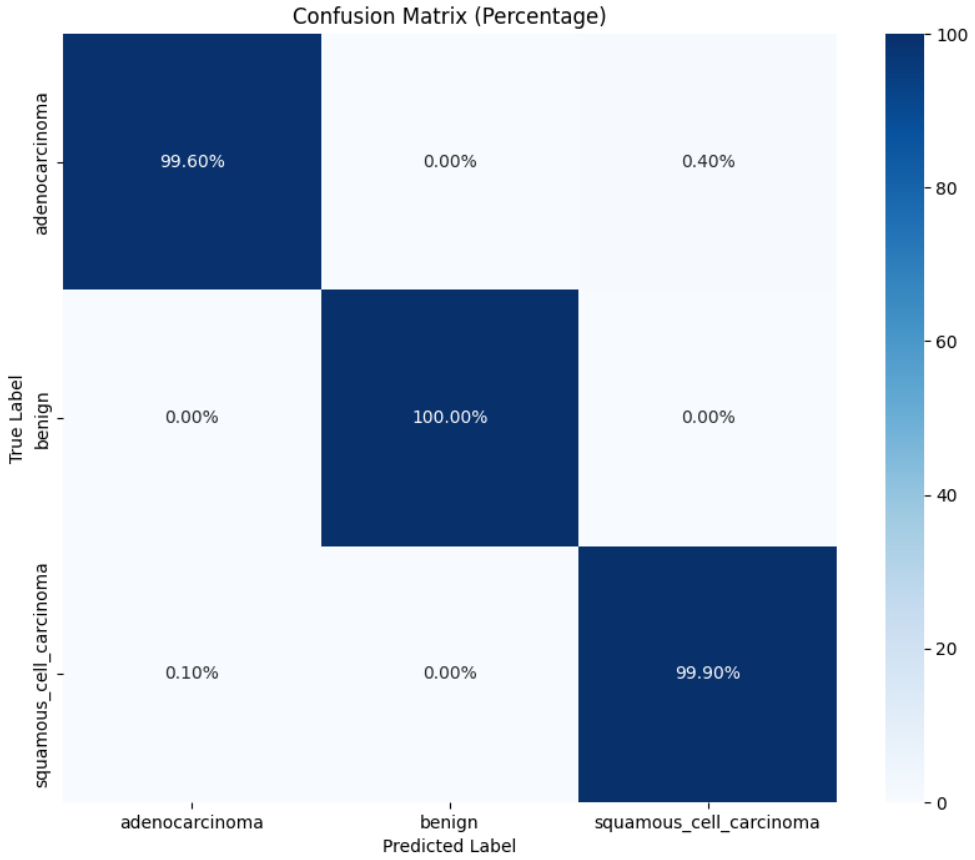


Figure 13. lung_cancer-densenet121-augmented

Observations:

- The DenseNet121-based model slightly outperformed the ResNet50-based model in terms of validation accuracy and other metrics.
- Data augmentation significantly improved the models' abilities to generalize to the validation dataset.
- Both models achieved substantial improvements over the initial unaugmented results.

Resources

Full code found here:

https://github.com/DevAbdoTolba/lung-cancer-classification [15]

Full dataset source:

https://www.kaggle.com/datasets/rm1000/lung-cancer-histopathological-images/ [16]

References

[1] W. H. Organization, "WHO," 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer. [Accessed 18 12 2024].

[2] A. C. Society, "cancer," 2021. [Online]. Available: https://www.cancer.org/cancer/types/lung-cancer/detection-diagnosis-staging.html. [Accessed 18 12 2024].

[3] J. C. a. T. Allen, "vol. 136 no.12," in *Lung Cancer Biomarkers: Present Status and Future Developments*, 2012, pp. 1478-1486.

[4] D. C. D. M. R. a. J. S. S. G. A. Müller, "vol. 9, no. 8," in *Inter-observer Variability in Lung Cancer Diagnosis*, 2014, p. e47–e48.

[5] A. E. e. al., "Nature, vol. 542, no. 7639," in *Dermatologist-level classification of skin cancer with deep neural networks*, 2017, p. 115–118.

[6] M. M. B. L. B. T. C. P. W. L. A. D. a. S. M. M. A. A. Borkowski, Lung and Colon Cancer Histopathological Image Dataset (LC25000), 2019.

[7] tampapath, "github," 18 12 2024. [Online]. Available: https://github.com/tampapath/lung_colon_image_set. [Accessed 18 12 2024].

[8] C. S. a. T. M. Khoshgoftaar, "J. Big Data, vol. 6, no. 1," in *A survey on Image Data Augmentation for Deep Learning*, 2019, p. 60.

[9] L. P. a. J. Wang, The Effectiveness of Data Augmentation in Image Classification using Deep Learning, 2017.

[10] I. S. a. G. E. H. A. Krizhevsky, ImageNet Classification with Deep Convolutional Neural Networks, 2012.

[11] researchgate, "researchgate," researchgate, jan 2020. [Online]. Available: https://www.researchgate.net/figure/Outline-of-ResNet-50-architecture-a-A-3-channel-image-input-layer-The-LL-LH-and-HH_fig3_343233188. [Accessed 18 12 2024].

[12] X. Z. S. R. a. J. S. K. He, "in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition," in *Deep Residual Learning for Image Recognition*, 2016, pp. 770-778.

[13] researchgate, "researchgate," researchgate, jan 2022. [Online]. Available: https://www.researchgate.net/figure/The-architecture-of-DenseNet121-with-Dense-block-D-Transition-blocks-T-and-Dense_fig3_358042365. [Accessed 18 12 2024].

[14] Z. L. L. v. d. M. a. K. Q. W. G. Huang, "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition," in *Densely Connected Convolutional Networks*, 2017, p. 4700–4708.

[15] DevAbdoTolba, "github," 18 12 2024. [Online]. Available: https://github.com/DevAbdoTolba/lung-cancer-classification. [Accessed 18 12 2024].

[16] R. Mandal, "kaggle," 9 2024. [Online]. Available: https://www.kaggle.com/datasets/rm1000/lung-cancer-histopathological-images/. [Accessed 18 12 2024].