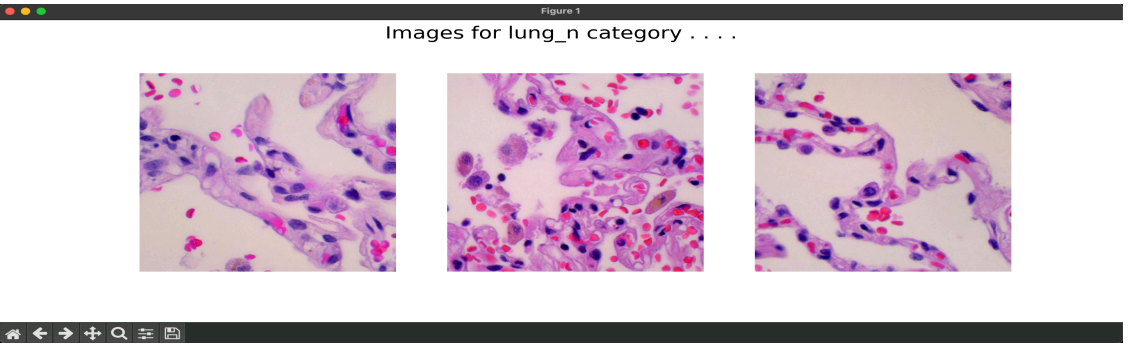
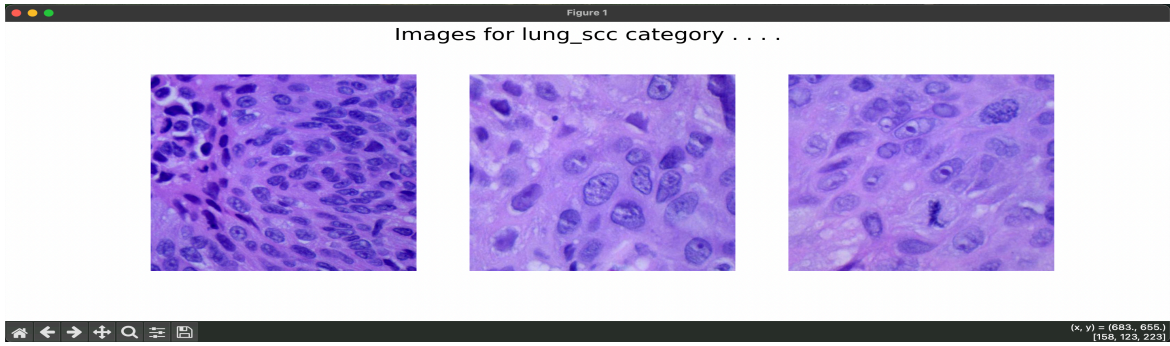
Project

Lung Cancer Detection (with CNN)

- This project aims to develop a deep learning model based on Convolutional Neural Networks (CNNs) for accurate and early detection of lung cancer from medical imaging data. The CNN will analyze lung tissue to automatically identify and classify whether it is cancerous or not, supporting timely diagnosis and treatment planning.
- Computer Vision is an application of deep neural networks that we have applied and used here since it helps us to automate tasks, hence easing the process of predicting the presence of cancerous cells among 3 classes.





Project

Workflow

- Data Collection
- Data Visualization
- Preprocessing
- Model Selection
- Architecture
- Model Training
- Validation
- Evaluation
- Deployment
- Maintenance

Project

Modules and Libraries

- **Pandas:** This library helped to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
- NumPy: NumPy arrays are very fast and can perform large computations in a very short time.
- **Matplotlib:** This library was used to draw visualizations.
- **Sklearn:** This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development and evaluation.
- **OpenCV:** This is an open-source library mainly focused on image processing and handling.
- **TensorFlow:** This is an open-source library that is used for Machine Learning and Artificial intelligence and provides a range of functions to achieve complex functionalities with single lines of code.

Project

Data Collection

- The dataset was taken from Kaggle.

- It had 5000 images of 3 classes of lung conditions: Normal Class, Lung Adenocarcinomas and Lung Squamous Cell Carcinomas.

```
\9'
  print('\n Validation accuracy has reached upto \90% so, stopping further training.')
['lung_aca', 'lung_n', 'lung_scc']
(12000, 256, 256, 3) (3000, 256, 256, 3) (12000, 3) (3000, 3)
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 32)	2,432
max_pooling2d (MaxPooling2D)	(None, 128, 128, 32)	0
conv2d_1 (Conv2D)	(None, 128, 128, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 64, 64, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 128)	0
flatten (Flatten)	(None, 131072)	0
dense (Dense)	(None, 256)	33,554,688
batch_normalization (BatchNormalization)	(None, 256)	1,024
dense_1 (Dense)	(None, 128)	32,896
dropout (Dropout)	(None, 128)	0
batch_normalization_1 (BatchNormalization)	(None, 128)	512
dense_2 (Dense)	(None, 3)	387

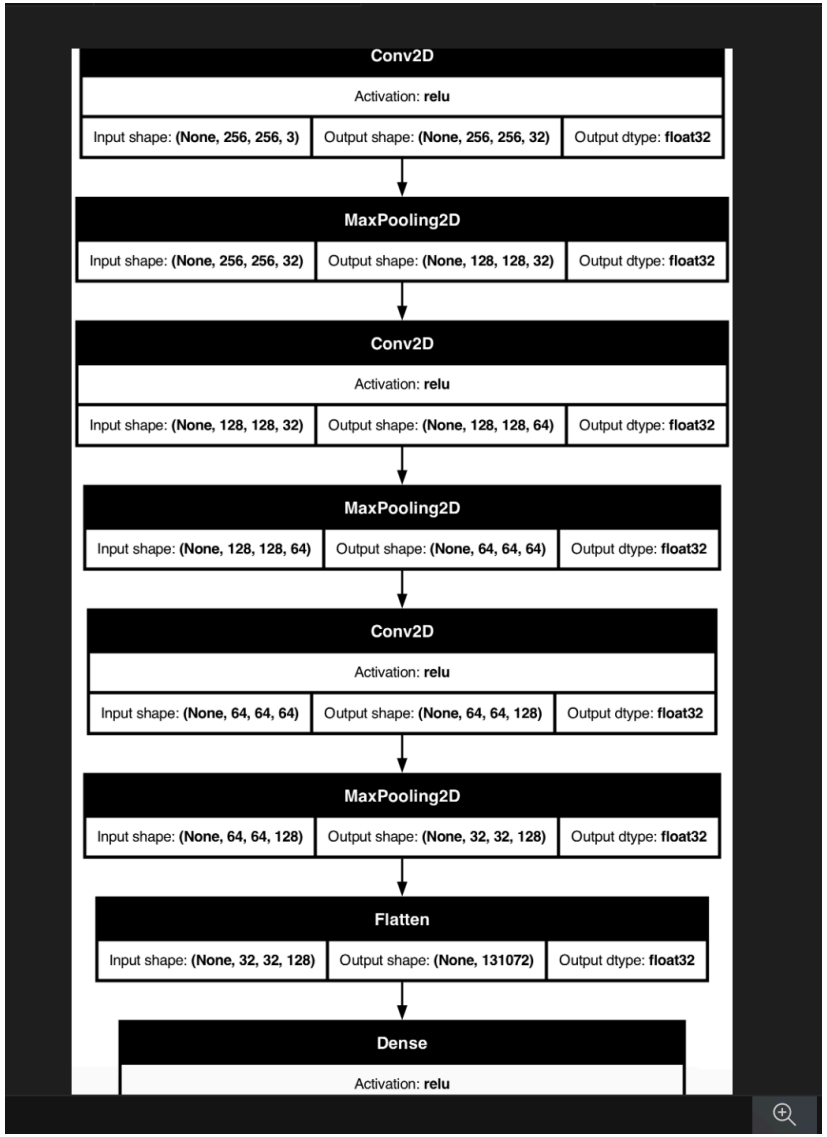
```
Total params: 33,684,291 (128.50 MB)
Trainable params: 33,683,523 (128.49 MB)
Non-trainable params: 768 (3.00 KB)
```

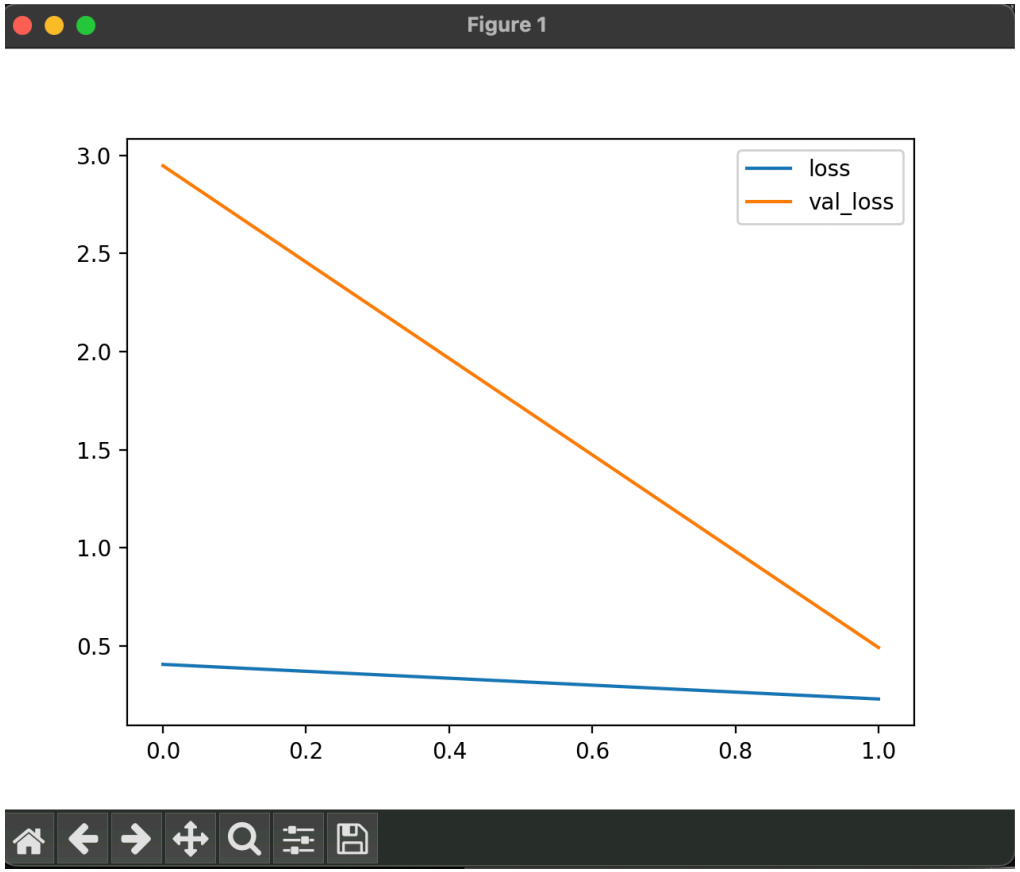

- **A sequential model was developed for the classifier.**

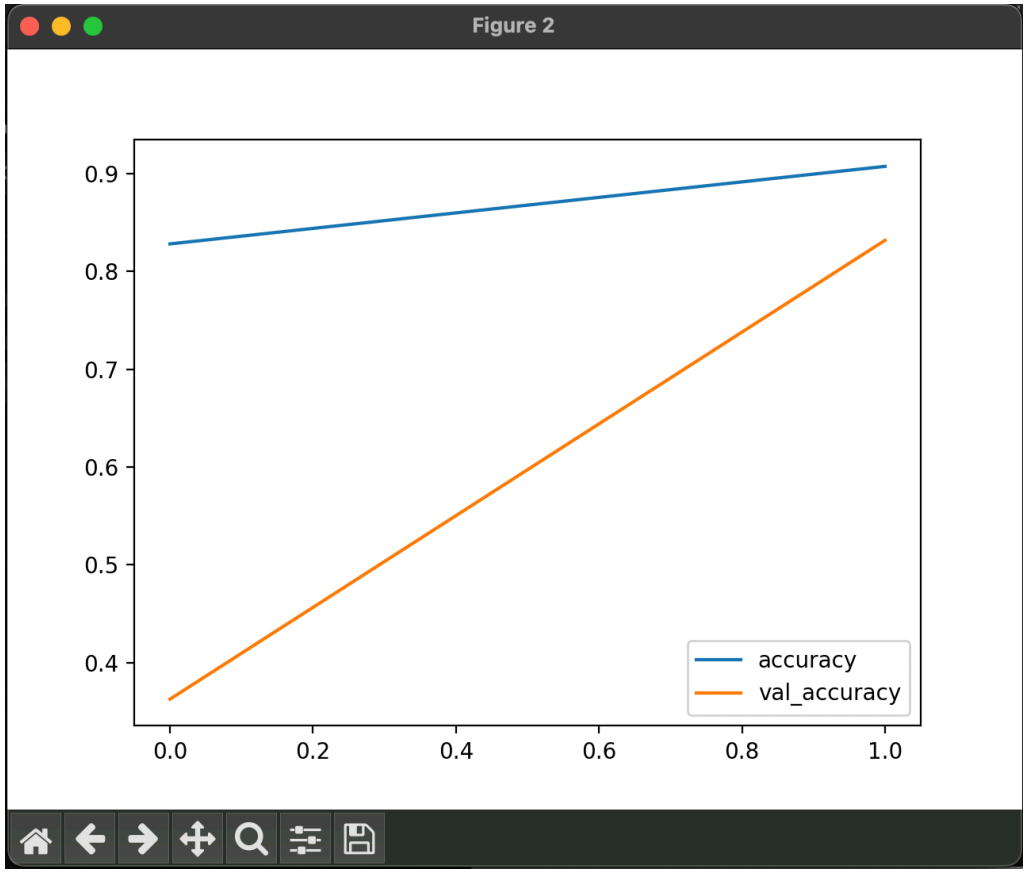
Project

Architecture

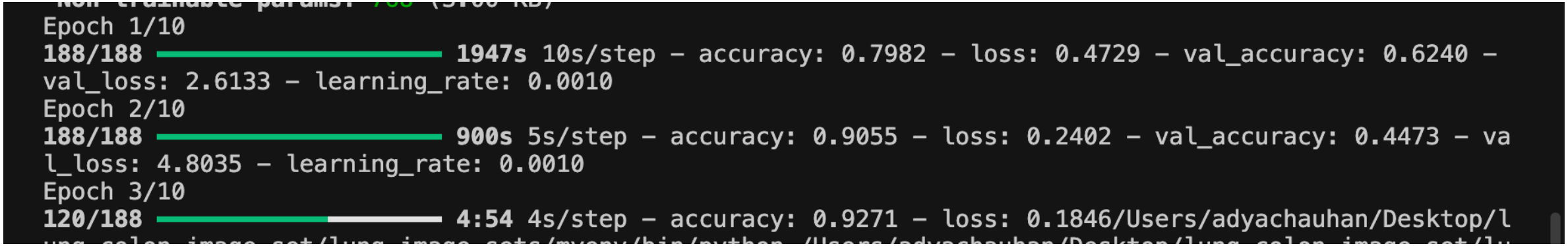
- **Input Layer**: *receives the images of the tissue.*
- Convolutional Layer: *extracts features.*
- Activation Functions: *introducing non-linearity*
- Pooling Layers: *reduces spatial dimensions.*
- Flattening Layers: *converts 2D feature maps into 1D vectors.*
- Output Layer: *provides classification.*
- Loss Function: *to measure prediction accuracy.*
- Optimizer: *adjusting any necessary parameters.*







- **From the above graphs, we can see that we have not overfitted the data as the difference between validation accuracy and validation loss is low**



- **The dataset will complete 10 epochs in total. Here it has done two and is working on the third one.**

Project

Conclusion

- The performance of the model is apt as the f1-score is above 0.90 which indicates that the model is correct about 90% of the time.

- Since the CNN model was a simple way to build this project, by using the complex Transfer Learning Technique a larger, more advanced model can be created which trains on millions of datasets and is eased with pre-trained parameters.

Project

Challenges

- Data quality issue: variations in image quality of tissue as well as inconsistencies in annotations can affect model accuracy and reliability.
- Limits data: difficult to have access to large and diverse datasets of medical images, which poses a challenge in training a good model.
- Complexity: CNN requires expertise and significant computational resources i.e., GPU and memory.
- Overfitting: preventing the model from memorizing the training data and providing unseen data from diverse sources.
- User acceptance and trust: for the healthcare industry to welcome this model, through performance tests and other valid parameters, as healthcare providers rely on these systems for critical diagnostics.

Project

Future Scope

- Advanced Model Enhancements: Improve CNN accuracy using advanced architectures and transfer learning techniques.
- Integration of AI Technologies: Incorporate explainable AI and learning for transparent model development.
- Multimodal Imaging Expansion: Extend to integrate other scans for comprehensive diagnostic capabilities.
- Global Deployment: Scale the project for global deployment, particularly in underserved regions using cloud-based solutions.
- Ethical Compliance and Education: Address ethical considerations, ensure patient privacy, and promote AI education among healthcare professionals.

Project

Learnings

- Deep Learning Foundations.
- Data Handling and Data Preprocessing.
- Problem Solving and Optimization.
- Documentation.